Process Monitoring and Fault Diagnosis using Random Forests

by

Lidia Auret

Dissertation presented for the Degree

of

DOCTOR OF PHILOSOPHY

(Extractive Metallurgical Engineering)

in the Department of Process Engineering

at the University of Stellenbosch

Promoter

Prof. Chris Aldrich

STELLENBOSCH

December 2010

DECLARATION

I, the undersigned, hereby declare that the work contained in this dissertation is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

…….. ……

Signature Date

SUMMARY

Fault diagnosis is an important component of process monitoring, relevant in the greater context of developing safer, cleaner and more cost efficient processes. Data-driven unsupervised (or feature extractive) approaches to fault diagnosis exploit the many measurements available on modern plants. Certain current unsupervised approaches are hampered by their linearity assumptions, motivating the investigation of nonlinear methods. The diversity of data structures also motivates the investigation of novel feature extraction methodologies in process monitoring.

Random forests are recently proposed statistical inference tools, deriving their predictive accuracy from the nonlinear nature of their constituent decision tree members and the power of ensembles. Random forest committees provide more than just predictions; model information on data proximities can be exploited to provide random forest features. Variable importance measures show which variables are closely associated with a chosen response variable, while partial dependencies indicate the relation of important variables to said response variable.

The purpose of this study was therefore to investigate the feasibility of a new unsupervised method based on random forests as a potentially viable contender in the process monitoring statistical tool family. The hypothesis investigated was that unsupervised process monitoring and fault diagnosis can be improved by using features extracted from data with random forests, with further interpretation of fault conditions aided by random forest tools. The experimental results presented in this work support this hypothesis.

An initial study was performed to assess the quality of random forest features. Random forest features were shown to be generally difficult to interpret in terms of geometry present in the original variable space. Random forest mapping and demapping models were shown to be very accurate on training data, and to extrapolate weakly to unseen data that do not fall within regions populated by training data.

Random forest feature extraction was applied to unsupervised fault diagnosis for process data, and compared to linear and nonlinear methods. Random forest results were comparable to existing techniques, with the majority of random forest detections due to variable reconstruction errors. Further investigation revealed that the residual detection success of random forests originates from the constrained responses and poor generalization artefacts of decision trees. Random forest variable importance measures and partial dependencies were incorporated in a visualization tool to allow for the interpretation of fault conditions.

A dynamic change point detection application with random forests proved more successful than an existing principal component analysis-based approach, with the success of the random forest method again residing in reconstruction errors.

The addition of random forest fault diagnosis and change point detection algorithms to a suite of abnormal event detection techniques is recommended. The distance-to-model diagnostic based on random forest mapping and demapping proved successful in this work, and the theoretical understanding gained supports the application of this method to further data sets.

OPSOMMING

Foutdiagnose is ’n belangrike komponent van prosesmonitering, en is relevant binne die groter konteks van die ontwikkeling van veiliger, skoner en meer koste-effektiewe prosesse. Data-gedrewe toesigvrye of kenmerkekstraksie-benaderings tot foutdiagnose benut die vele metings wat op moderne prosesaanlegte beskikbaar is. Party van die huidige toesigvrye benaderings word deur aannames rakende liniariteit belemmer, wat as motivering dien om nie-liniêre metodes te ondersoek. Die diversiteit van datastrukture is ook verdere motivering vir ondersoek na nuwe kenmerkekstraksiemetodes in prosesmonitering.

Lukrake-woude is ’n nuwe statistiese inferensie-tegniek, waarvan die akkuraatheid toegeskryf kan word aan die nie-liniêre aard van besluitnemingsboomlede en die bekwaamheid van ensembles. Lukrake-woudkomitees verskaf meer as net voorspellings; modelinligting oor datapuntnabyheid kan benut word om lukrake-woudkenmerke te verskaf. Metingbelangrikheidsaanduiers wys watter metings in ’n noue verhouding met ’n gekose uitsetveranderlike verkeer, terwyl parsiële afhanklikhede aandui wat die verhouding van ’n belangrike meting tot die gekose uitsetveranderlike is.

Die doel van hierdie studie was dus om die uitvoerbaarheid van ’n nuwe toesigvrye metode vir prosesmonitering gebaseer op lukrake-woude te ondersoek. Die ondersoekte hipotese lui: toesigvrye prosesmonitering en foutdiagnose kan verbeter word deur kenmerke te gebruik wat met lukrake-woude geëkstraheer is, waar die verdere interpretasie van foutkondisies deur addisionele lukrake-woude-tegnieke bygestaan word. Eksperimentele resultate wat in hierdie werkstuk voorgelê is, ondersteun hierdie hipotese.

’n Intreestudie is gedoen om die gehalte van lukrake-woudkenmerke te assesseer. Daar is bevind dat dit moeilik is om lukrake-woudkenmerke in terme van die geometrie van die oorspronklike metingspasie te interpreteer. Verder is daar bevind dat lukrake-woudkartering en -dekartering baie akkuraat is vir opleidingsdata, maar dat dit swak ekstrapolasie-eienskappe toon vir ongesiene data wat in gebiede buite dié van die opleidingsdata val.

Lukrake-woudkenmerkekstraksie is in toesigvrye-foutdiagnose vir gestadigde-toestandprosesse toegepas, en is met liniêre en nie-liniêre metodes vergelyk. Resultate met lukrake-woude is vergelykbaar met dié van bestaande metodes, en die meerderheid lukrake-woudopsporings is aan metingrekonstruksiefoute toe te skryf. Verdere ondersoek het getoon dat die sukses van res-opsporing op die beperkte uitsetwaardes en swak veralgemenende eienskappe van besluitnemingsbome berus. Lukrake-woude-metingbelangrikheidsaanduiers en parsiële afhanklikhede is ingelyf in ’n visualiseringstegniek wat vir die interpretasie van foutkondisies voorsiening maak.

’n Dinamiese aanwending van veranderingspuntopsporing met lukrake-woude is as meer suksesvol bewys as ’n bestaande metode gebaseer op hoofkomponentanalise. Die sukses van die lukrake-woudmetode is weereens aan rekonstruksie-reswaardes toe te skryf.

’n Voorstel wat na aanleiding van hierde studie gemaak is, is dat die lukrake-woudveranderingspunt- en foutopsporingsmetodes by ’n soortgelyke stel metodes gevoeg kan word. Daar is in hierdie werk bevind dat die afstand-vanaf-modeldiagnostiek gebaseer op lukrake-woudkartering en -dekartering suksesvol is vir foutopsporing. Die teoretiese begrippe wat ontsluier is, ondersteun die toepassing van hierdie metodes op verdere datastelle.

ACKNOWLEDGEMENTS

I would like to lift and twirl my imaginary hat to a number of
people and entities who have made this work easier, better and worth it:

For a never-ending source of what-ifs and why-nots,
for unrelenting, infectious belief in the potential of data-based modelling,
and for overall support and guidance, thank you Professor Chris Aldrich.

For technical and theoretical assistance,
thank you to Doctor Gorden Jemwa and Doctor JP Barnard,
and also to all of those who have elucidated
some programming trick or statistical concept.

For financial assistance, a big thank you to
the Wilhelm Frank scholarship trust,
the Stellenbosch University merit bursary scheme
and AngloPlatinum.

For emotional support, I am very grateful to my friends and family.
Thank you Marietjie, Jeanette, Corné and all the rest of you.

Finally, I dedicate this work to my parents.

"There is no true interpretation of anything; interpretation is a vehicle in the service of human comprehension. The value of interpretation is in enabling others to fruitfully think about an idea."

- Andreas Buja

TABLE OF CONTENTS

CHAPTER 1 - INTRODUCTION

Fault diagnosis is an important component of process monitoring, in the greater context of developing safer, cleaner and more cost efficient processes. Unsupervised feature extraction approaches to fault diagnosis exploit the many measurements available on modern plants. Certain current feature extraction approaches are hampered by linearity assumptions. The diversity of data structures also motivates the investigation of novel feature extraction methodologies in process monitoring. The purpose of this study is to investigate the feasibility of a new feature extraction method based on random forests as a viable contender in the process monitoring statistical tool family.

1.1 Monitoring and control in process engineering

Process engineering is concerned with the conversion of raw materials to valuable products. This conversion entails physical and chemical processes, with the tasks of the process engineer being the design, maintenance and optimization of said processes. Once specific flow sheets and process units have been designed and commissioned, the extent of possible optimization is automatically constrained. In order to improve the efficiency of existing systems, the options available are limited to controlling process variables to minimize variance around a known optimal state, or determining other optimal states.

A vital element of controlling processes is that of process monitoring. Process monitoring involves the collection of data in the form of observations from process variables, the detection and identification of abnormal process conditions, and the recovery of the process to a specified optimal state. Abnormal process conditions have adverse effects on process plants in the shape of equipment failure resulting in downtime, reduced product quality, increased emissions and possible catastrophic events. The early detection of these events is thus beneficial economically, environmentally and in terms of personnel safety (Himmelblau, 1978; Russell et al., 2000).

The nature of modern process plants adds complexity to the task of process monitoring. Not only are plants large scale operations with a myriad of manipulated variables, but flow sheets contain intricate recycle configurations and convoluted control systems that may conceal faults (Himmelblau, 1978; Russell et al., 2000). On the point of control systems, a clear distinction between process control and process monitoring is identified: process control involves the design of controllers to compensate for the effect of process disturbances on important variables, in effect transferring system variability to less significant manipulated variables. In contrast, process monitoring aims to identify abnormalities and diagnose the causes of disturbances in order to improve a process (MacGregor & Kourti, 1995).

1.1.1 Fault diagnosis: The core of process monitoring

The key elements of process monitoring are fault detection, fault identification and process recovery. Here, a fault is defined as “a departure from an acceptable range of an observed variable or calculated parameter associated with equipment” (Himmelblau, 1978). The detection and identification components of process monitoring are grouped as fault diagnosis. Detection requires ascertaining whether a fault has occurred, while identification pinpoints the process variables involved in the fault (Russell et al., 2000). Fault diagnosis can be seen as the inverse of process simulation. Where process simulation models system behaviour given certain structural and functional features, diagnosis extracts the structural and functional aspects from system behaviour (Venkatasubramanian et al., 2003a).

Process system behaviour is monitored through a possible multitude of sensors on observed variables and the process control inputs to actuators. When a fault occurs, the relationships among observed variables are

altered, causing process performance specifications to be violated (Himmelblau, 1978; Venkatasubramanian et al., 2003b). These faults may be due to disturbance parameter changes, process parameter changes or malfunctioning actuators and sensors (Russell et al., 2000).

An ideal fault diagnostic system would be able to rapidly detect, identify and explain any of aforementioned faults, while being robust to system noise. Not only must a distinction be made between different failures, but the case of simultaneous faults should be distinguishable. Such a system would also be adaptable and able to identify novel faulty conditions. An accurate estimate of the probability of false and missed alarms would allow sound engineering decisions, while modelling and computational expenses should be kept to a minimum (Venkatasubramanian et al., 2003c).

1.1.2 Various approaches to fault diagnosis

A grouping variable for all fault diagnostic schemes is that of the nature of the a priori knowledge used in their development (Venkatasubramanian et al., 2003c). When a fundamental, first-principles analytical modelling approach is utilized, it is referred to as quantitative model-based fault diagnosis. Qualitative model-based fault diagnosis extracts fundamental understanding of the process from expert knowledge. Process history-based methods (so called data-driven) utilize a priori knowledge in the form of process measurement databases of normal operating conditions only.

The main drawback of fundamental models (whether quantitative or qualitative) is the large cost and time investment required to obtain models of sufficient accuracy and robustness for complex plants (Russell et al., 2000). In contrast, data-driven methods require little modelling effort and, on the surface, a low-level of expert knowledge on specific systems (Venkatasubramanian et al., 2003b). The proliferation of sensors in process plants have led to the availability of large databases suitable for data-driven techniques. However, the proficiency of data-driven fault diagnosis is sensitive to the quantity and quality of available process data (Russell et al., 2000).

1.1.3 Feature extraction in fault diagnosis

The large number of measured variables in process databases is both help and hindrance. Various diverse measurements increase the detectability of faults, while data overload may lead to correlated and redundant information in heavily instrumented processes. Data-driven fault diagnostics can then be seen as a problem of measurement compression and feature extraction (Wise & Gallagher, 1996). Compression is required to retain only useful information, while extraction is necessary where covariation of variables obscure this useful information. This information extraction procedure produces a number of features, which can be considered as concealed combinations of observation variables that express available process information in the lowest dimensionality.

Once informative features have been extracted, the feature space is used for fault detection. The translation of the feature space to a decision space of fault / no-fault membership can be regarded as a search strategy or learning algorithm. Two schemes can be employed in this transformation: supervised and unsupervised learning. For supervised learning, representative measurements are available for all expected abnormal conditions, with fault detection and identification reducing to a pattern recognition problem. This approach requires the availability of fault data. The unsupervised learning method, however, only requires normal operating condition data, with faults detected as mismatches to the normal operation template (Venkatasubramanian et al., 2003b).

1.1.4 Room for improvement in unsupervised fault diagnosis

A major limitation of current linear feature extraction benchmarks is their linear nature. It has been postulated that using a linear method to extract features from nonlinear data can be inadequate (Dong & McAvoy, 1996).

Another incentive for exploring novel nonlinear feature extraction methods originates from the varied nature of processes and process data. The diversity of possible process data structures motivates the exploration of diversity of feature extraction methods. In light of this, state-of-the-art statistical inference techniques such as neural networks (Kramer, 1991; Jia et al., 1998; Antory et al., 2008; Dong & McAvoy, 1996; Zhu & Li, 2006), kernel methods (Lee et al., 2004a; Choi et al., 2005; Cho et al., 2005) and more have been investigated in feature extractive fault diagnosis.

1.2 A novel idea: Application of random forests to unsupervised fault diagnosis

A recent development in statistical learning is the emergence of ensembles of learning machines. An ensemble is simply the combination of a collection of classifiers in order to enhance the performance of the overall classifier (Valentini & Masulli, 2002). It has been shown (Valentini & Masulli, 2002) that ensembles of classifiers regularly perform better than the individual classifiers, even when the base classifiers are considered weak. With limited data sets, a number of different training hypotheses can be reasonably accurate, even though they do not reflect the true hypothesis. By constructing an ensemble of classifiers, more thorough exploration of hypotheses can be accomplished (Valentini & Masulli, 2002).

The random forest model is an example of ensemble methods, with the base classifiers consisting of unpruned decision tree classifiers (Breiman, 2001). A decision tree is a recursive subspace partitioning classifier, in such a way as to reduce the class impurity of successive subsets (Breiman et al., 1993). The current widespread use of the random forest algorithm can be attributed to its high accuracy, robustness to outliers and noise, fast computations and internal estimates of error and variable importance (Breiman & Cutler, 2003).

Tree ensembles such as random forests not only provide predictions and proximities; but, through the complex structure of its ensemble members, can further provide interpretive functionality through variable importance (Breiman & Cutler, 2003) and partial dependence analysis (Friedman, 2001).

1.2.1 Random forest feature extraction

Recently, research has been conducted by (Shi & Horvath, 2006) in using random forests for unsupervised learning. The training data set for this application consists of unlabeled process data and an additional synthetic class to act as contrast. The synthetic data can be generated from the process data distributions, or as a uniform hyper-rectangle in the measurement space. An ensemble of decision trees is constructed, and similarities between the unlabeled process data points can be determined by calculation of a proximity matrix for only unlabeled process data. This proximity matrix is derived from the agreement amongst ensemble members with respect to the subspace allocation of data point pairs. Multidimensional scaling can be employed to extract features from this proximity data (Shi & Horvath, 2006).

1.2.2 A candidate random forest fault diagnostic scheme

In the field of process monitoring and fault diagnosis, decision trees and ensembles of trees have not been extensively used. Notable exceptions include the inductive learning of process conditions from process features by decision trees (Bakshi & Stephanopoulos, 1994; Jemwa & Aldrich, 2005; Ma & Wang, 2009) as well as the application of supervised random forest classification to gas turbine (Maragoudakis et al., 2008) and induction motor (Yang et al., 2008) fault diagnosis.

The use of random forests in unsupervised fault diagnosis is an unexplored direction. Such a method could consist of adding a contrast to the normal process data, and training the random forest on this new data set of two classes. The multidimensional scaling features of the proximities of the normal and synthetic classes could feasibly be used to construct a confidence limit to the normal process data, with added unseen faults then detected as outliers to the confidence threshold. Variable reconstruction may be used to identify and diagnose faults. Fault identification and interpretation can be further enhanced with random forest variable importance and partial dependencies.

1.3 Hypothesis

Random forests constitute a viable basis for the development of nonlinear process monitoring and fault diagnosis methods.

1.4 Objective and scope

The overall objective of this study is to investigate the advantages, if any, of random forest feature extraction in fault diagnosis applications, specifically as applied to process engineering applications.

This global objective will be pursued by completion of the following tasks:

- A critical literature survey on feature extraction techniques in fault diagnosis; as well as on the random forest modelling and feature extraction approach
- The assessment of the quality of features extracted with random forests
- The development of an unsupervised fault diagnostic scheme using random forest feature extraction
    - The implementation of the random forest fault diagnostic scheme as robust code
    - The testing of the random forest fault diagnostic scheme on simulated data, a benchmark process engineering problem and a real-world mineral processing data set
- The development of interpretive tools for identifying and interpreting important process variables involved in process changes and faults
    - The implementation of interpretive variable importance and visualization tools based on random forests as robust code
    - The testing of random forest interpretive tools on simulated data and process faults
- The development of a dynamic random forest change point detection scheme
    - The implementation of the dynamic random forest change point detection as robust code
    - The testing of the dynamic random forest change point detection scheme on simulated static and dynamic data sets

In terms of the scope of this study, only the application of random forest feature extraction to *unsupervised* fault diagnosis and dynamic change point detection for continuous process systems will be developed and tested on a proof-of-concept basis. This explicitly excludes random forest classification applied to supervised fault diagnosis, application to batch processes and the development of process recovery schemes after fault identification.

1.5 Thesis layout

Chapter 2 presents the critical literature survey on unsupervised fault diagnosis, with chapter 3 presenting the background of random forests and random forest feature extraction. Chapter 4 gives a brief summary of the methodology followed in developing and testing fault and change detection frameworks. Chapter 5 presents a summary of the assessment of random forest feature extraction. Chapters 6, 7 and 8 provide details on the random forest fault detection framework, its application and performance analysis. Chapters 9 and 10 further exploit random forest interpretation tools for fault identification. Chapter 11 covers a random forest change

detection framework and its application to case studies. Finally, chapter 12 closes with the most salient conclusions and recommendations. The appendices include additional information on random forest feature assessment; a more complex methodology for extracting random forest features based on tree structures and a list of peer-reviewed references for this work.

CHAPTER 2 - UNSUPERVISED FAULT DIAGNOSIS

Statistical process monitoring originated with the univariate monitoring of key performance indicators. Extending the monitoring task beyond univariate upper and lower control limits to multivariate confidence regions enables correlation among variables to be taken into account. By further utilising data compression and feature extraction, redundancy can be minimized and inherent data structure explored. The definition of new measures, such as Hotelling's T^2 (with modifications) and squared prediction error, allows decomposition of process variance into within-process and disturbance-related causes. Principal component analysis is a much-used multivariate statistical process control technique in this mould. The linear limitation of PCA has led to the application of nonlinear feature extraction methods to fault diagnosis, and continues to be an area of active research.

2.1 Statistical process control

Statistical process control is an approach to process monitoring, with the premise that measured process variables are subject to random disturbances. Such stochastic variables are not entirely determined by past or present states of the underlying process, but are rather single realisations of an underlying stochastic process. A stochastic variable can be considered as adhering to a specific distribution when under control. If further assumptions are made on the nature of this distribution, for example that a parametric distribution is valid, faulty conditions can be expressed as a change in the underlying distribution, and may be detected as parameter changes (Venkatasubramanian et al., 2003b).

In the following subsection, the conceptual foundation of univariate statistical process control will be presented. This serves as a proper grounding of certain fault diagnosis principles, and although limitations are present in this approach, much can be discovered of the theory of statistical application to the process monitoring problem.

2.1.1 Univariate statistical process control

The quality control of product characteristics was the first application of statistical process control in online process monitoring, as demonstrated by Shewhart control charts (1931), cumulative sums charts (1954) and exponentially-weighted moving average charts (Venkatasubramanian et al., 2003b).

A Shewhart control chart is constructed by assuming an underlying parametric distribution for the specific stochastic process variable under consideration, and calculating confidence limits (upper and lower control limits) for in-control manifestations from the properties of the assumed distribution. When the confidence limits are exceeded, it is an indication that an abnormal event, presumably a fault, has occurred (Russell et al., 2000).

The control limits are calculated based on the variability of a reference set of process data, the so-called normal operating conditions (NOC) process data. The assumption is made that only natural variation occurs in the NOC data, with control limits based on NOC only exceeded in case of an abnormal event. The choice of NOC data is thus critical to the success of the control chart approach. NOC data must reflect all natural variance for process data when the process performance specifications are met (Kresta et al., 1991). An example of a univariate control chart is given in Figure 2.1.

For robust yet sensitive fault detection, a trade-off exists for the selection of control limits in terms of minimizing false alarms and missed detections. If the control limits are too stringent, the false alarm rate will be high. However, if the control limits are too accommodating, the missed detection rate will be high. The

advantage of assuming a parametric distribution for the stochastic measurements is that the false alarm and missing detection rates can be predicted as Type I and Type II errors, respectively (Russell et al., 2000).

Initially, only variables indicative of product quality were considered for control chart investigation. However, there are definite advantages in monitoring process variables as well. Any process event that has an effect on a product quality variable is likely to have "fingerprints" in process data as well. Furthermore, it may be that only a small number of the product properties are measurable, to such an extent that the product quality cannot be defined sufficiently. Conversely, process variables can often be indicative of product quality and should be exploited. As an added advantage, fault identification may be aided by directly monitoring process variables in order to determine fault causes. Aforementioned reasons led to the application of univariate control charts not only to product quality measures, but to process variables as well (MacGregor & Kourti, 1995).

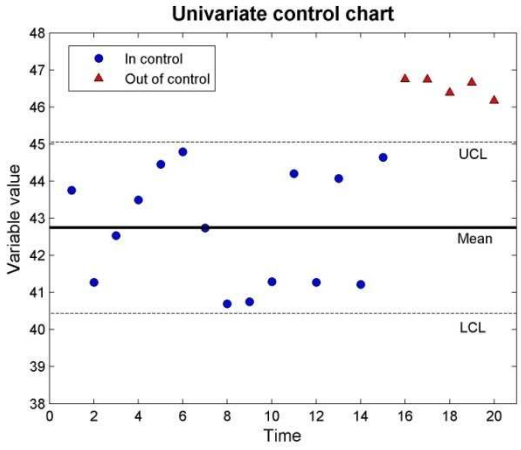

Figure 2.1: Univariate control chart with mean, upper control limits (UCL) and lower control limits (LCL)

A limitation of univariate statistical process control fault detection lies in the fact that control limits for each process variable is determined individually, disregarding the correlation that might exist between variables (Russell et al., 2000).

Fault identification as applied to univariate statistical process control consists of calculating the normalized errors for each monitored variable. The normalization is achieved by subtracting the expected value of the variable from a specific instance, and dividing by its variation. Again, correlation among process variables is not accounted for (Russell et al., 2000).

2.1.2 Multivariate statistical process control

As mentioned in the previous subsection, univariate control charts do not exploit the correlation that may exist between process variables. In the case of process data, cross correlation is present due to restrictions enforced by mass and energy conservation principles, as well as the possible existence of a large number of different sensor readings on essentially the same process variable. Another form of correlation present in process data is autocorrelation. Where sampling intervals for process variables are small, the measurements cannot be assumed to be identically and independently distributed. Rather, a variable value at a specific time value is dependent on its value at the previous value (Russell et al., 2000).

- Hotelling's T^2: Process data compression

The first attempt at incorporating cross correlation among variables in the control chart approach led to the development of the Hotelling's T^2 statistic. The control limits (confidence bounds) for Hotelling's T^2 score

distances is determined based on the assumption that the NOC process data conforms to a multivariate normal distribution. If it is further assumed that the NOC training data parameter estimates are sufficiently accurate estimates of NOC parameters in general, the Hotelling's T^2 abides by the chi-squared distribution (m degrees of freedom), which translates as a hyperelliptical in the original observation space. This elliptical confidence region is less conservative than control limits based on univariate distribution assumptions. It must be noted that the training sample size and quality have a large influence on the accuracy of the control limits (Russell et al., 2000). An example of the elliptical confidence limits is given in Figure 2.2.

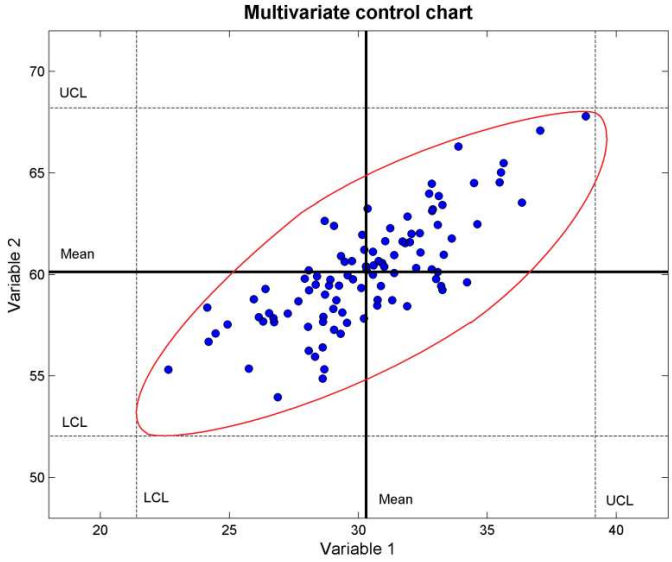

Figure 2.2: Multivariate control chart with improved confidence region

A brief outline of this method is given here to illustrate the compression of multivariate process data (Russell et al., 2000):

Hotelling's T^2 calculation

- Let **X** be NOC process data, functioning as the training set for the determination of parameters for Hotelling's approach. **X** consists of N observations of m process variables: (If required, **X** is autoscaled to zero mean and unit variance)
  - $\mathbf{X} = \begin{bmatrix} \mathbf{X}_{11} & \mathbf{X}_{21} & \dots & \mathbf{X}_{1m} \\ \mathbf{X}_{12} & \mathbf{X}_{22} & \dots & \mathbf{X}_{2m} \\ \dots & \dots & \dots & \dots \\ \mathbf{X}_{N1} & \mathbf{X}_{N2} & \dots & \mathbf{X}_{Nm} \end{bmatrix}$ ***Eqn. 1***
- **Σ**, the sample covariance of **X** is calculated:
  - $\mathbf{\Sigma} = \frac{1}{N-1}\mathbf{X}^T\mathbf{X}$ ***Eqn. 2***
- The sample covariance now undergoes eigenvalue decomposition:
  - $\mathbf{\Sigma} = \mathbf{V}\Lambda\mathbf{V}^T$ ***Eqn. 3***
- The measurement space is now deconstructed into a collection of uncorrelated variables **W** such that:
  - $\mathbf{W} = \mathbf{V}^T\mathbf{X}$ ***Eqn. 4***
- A score distance based on Hotelling's T^2 is now calculated as:
  - $s_{H,i} = \mathbf{X}_{i.}{}^T\mathbf{V}\Lambda^{-1}\mathbf{V}^T\mathbf{X}_{i.}$ ***Eqn. 5***
  - Where $\mathbf{X}_{i.}$ is a single sample i consisting of all m variable measurements

Fault identification is accomplished through Hotelling's T^2 decomposition to detect fault instances, with Λ_i and $\mathbf{V}_i$ giving an indication of the contribution of a specific process variable $\mathbf{X}_{.i}$ to the fault condition (Russell et al., 2000).

A modification of Hotelling's T^2 approach involves the retention of only a (with $a < m$) eigenvectors and eigenvalues for the rotation and scaling of process data. This translates to the retention of a lower dimensional, supposedly "information-rich" feature space, and the discarding of an "information-poor" set of eigenpairings. This selective use of features is explored further in the following subsection (Russell et al., 2000).

- Redundancy in process data and feature extraction

Feature extraction aims to exploit the presence of redundancy in the process data. Redundancy is an indication of uninformative measurements in process data, resulting in an intrinsic dimensionality of informative process data being lower than the number of process measurement variables available. This lower dimensional "information space" is defined by features or latent variables constructed as combinations and functions of measured process variables. These features aim to capture the relationship among process variables, which often arise due to fundamental aspects of physical systems (Venkatasubramanian et al., 2003c; Kresta et al., 1991).

In light of the presence of lower dimensional features, process fault diagnosis can be considered as a series of transformations (mappings) of measured process variables. The first mapping is the transformation from process measurement space to the feature space. Secondly, a learning algorithm can be utilised to map the feature space to a decision space of fault / no-fault membership. A possible further mapping is from the decision space to a class space, where a fault is characterized as a specific type. This mapping is more common in supervised fault diagnosis methods (Venkatasubramanian et al., 2003c).

The theoretical advantages of multivariate feature extraction fault diagnosis pertain to the compression and information-retention characteristics of the feature extraction transformation. By utilizing data compression, these methods can handle many correlated measurement variables. The presence of structure in the data is explicitly exploited by assuming that a lower intrinsic dimensionality is applicable. Feature extraction can contribute useful diagnostic information (Wise & Gallagher, 1996). However, certain identification methods require that explicit mappings exist between the measurement, feature and decision spaces. Where feature extraction is deterministic, model development is straightforward and computational requirements are low. Further computational and training preparation savings arise from the fact that only NOC data are required (Wise & Gallagher, 1996).

It is also stated that the division of the original measurement space into "information-rich" and "information-poor" subspaces allows higher robustness to noise (Venkatasubramanian et al., 2003b). A further advantage is the fact that no explicit process system-based model has to be developed. Product quality prediction based on extracted features is another application of this adaptation of fault diagnosis (Kresta et al., 1991).

If two or three features can be extracted which represents the process sufficiently, these features and projected process measurements can be visualized. This allows plant managers to exploit the powerful human talent of pattern recognition in order to gain an understanding of process conditions and fault patterns. Above and beyond the visualization of process data in low-dimensional feature space, the inspection of simple derived-statistic control charts enable a quick and easy way to determine whether a process is in control, and whether it shows trends towards out-of-control behaviour (Kresta et al., 1991).

In the case of implicit measurement-to-feature relation, the absence of "fingerprint" properties (explicit process variable contributions to features), fault identification is challenging (Venkatasubramanian et al., 2003b).

- Confidence bounds for features

Control limits for feature extraction-based fault diagnosis can be determined parametrically or non-parametrically. As shown with the Hotelling's T^2 approach, by assuming an underlying parametric distribution of the measurements, the distribution properties can be retained through the respective mappings and parametric distribution-based confidence limits calculated.

Another approach is to not make any parametric distribution assumptions about the process measurement or feature variables, but to rather use empirical techniques to establish the structure of the NOC data from the data itself. Such an approach was first developed by Martin and Morris (1996), who exploited kernel density estimation in the construction of so called likelihood-based confidence regions. This approach was motivated by the fact that the authors found that for many industrial process, multivariate normality tests confirmed that the scores in the feature space rarely followed a multivariate normal distribution. It was also shown that Hotelling's T^2 confidence limits for industrial data sometimes resulted in very conservative confidence regions, with large areas devoid of any NOC data (Martin & Morris, 1996).

- Fault identification in feature space and beyond

As with the Hotelling's T^2 statistic approach, fault identification can be attempted through decomposition of the T^2 statistic. However, the identification of process variables responsible for abnormal conditions is not always a simple matter in the case of feature extraction-based fault diagnosis. Complexities arise when the mapping from the measurement space to the feature space is not explicit, thus providing no direct weightings, coefficients or "fingerprints" of process variables in the calculated features.

The presence of an "information-poor" space, or residual space, can be exploited to great advantage. By calculating a residual distance r (squared prediction error, SPE) of the feature-based reconstruction of the process variables and the actual process variables, another diagnostic sequence is available for monitoring. Furthermore, by calculating the squared prediction error on individual process variable basis (Q_j for process variable $\mathbf{X}_{ij}$ at sample i), an indication of process variables contributing to or affected by the abnormal event can be obtained. However, care must be taken in the interpretation of such contribution plots, with the well-known statistics adage of "correlation does not imply causation" to be kept in mind.

An increase of the "information-rich" or score distance, without a breach of SPE limits, indicate that the model assumptions (in this case, measurement space to feature space mapping relations) are still valid, the process is merely migrating within this subspace away from the NOC standard. In any situation where the residual space diagnostic SPE increases beyond its control limits, a break-down of the relationships between process variables as encapsulated by the measurement space to feature space mapping is on the cards.

2.2 Multivariate statistical process control benchmark: Principal component analysis

The use of the deterministic variance-preserving procedure, principal component analysis (PCA), is the most widespread feature extraction fault diagnostic system applied in industrial processes (MacGregor & Kourti, 1995; Kresta et al., 1991). Due to its dominant role in process fault diagnosis, the PCA approach will be briefly illustrated here. (Refer to Chapter 4: Methodology for in-depth discussion of PCA fault detection and identification). The following snapshot serves as a platform for the discussion of selected recent feature extraction fault diagnostic methods.

PCA was proposed by Pearson in 1901 (Venkatasubramanian et al., 2003b) and developed by Hotelling in 1947 (Venkatasubramanian et al., 2003b) with the aim to define a set of principal components, consisting of linear combinations of the original measurement variables, such that the first principal component accounts for the most variance in the data set, the second principal component for the second most variance, etc. The principal components are orthogonal and preserve the correlation among the process variables. As in Hotelling's T^2 statistic approach, the principal components are calculated using eigen decomposition of the NOC process data covariance matrix.

Principal component analysis

- For data set **X** (N observations by m variables), construct the covariance matrix $\boldsymbol{\Sigma}$
  - $\boldsymbol{\Sigma} = \frac{1}{N-1}\mathbf{X}^{T}\mathbf{X}$ ***from Eqn. 2***
- Calculate the eigenvectors **V** and eigenvalues Λ for the covariance matrix $\boldsymbol{\Sigma}$ using eigenvalue decomposition
  - $\boldsymbol{\Sigma} = \mathbf{V}\Lambda\mathbf{V}^{T}$ ***from Eqn. 3***
- Determine the reduced dimensionality a which captures significant variance
- Calculate principal component scores using principal components **P** (a columns of eigenvector matrix **V**)
  - $\mathbf{T} = \mathbf{XP}$ ***Eqn. 6***

The original measurements can be located on the hyperplanes spanned by principal components subject to their scores (see Figure 2.3). In the context of feature extraction, the score vectors obtained from projecting the process measurements onto the principal components can be considered the extracted features. The number of principal components to use in calculating the features can be determined by investigating the cumulative variance accounted for by adding additional principal components to the score space.

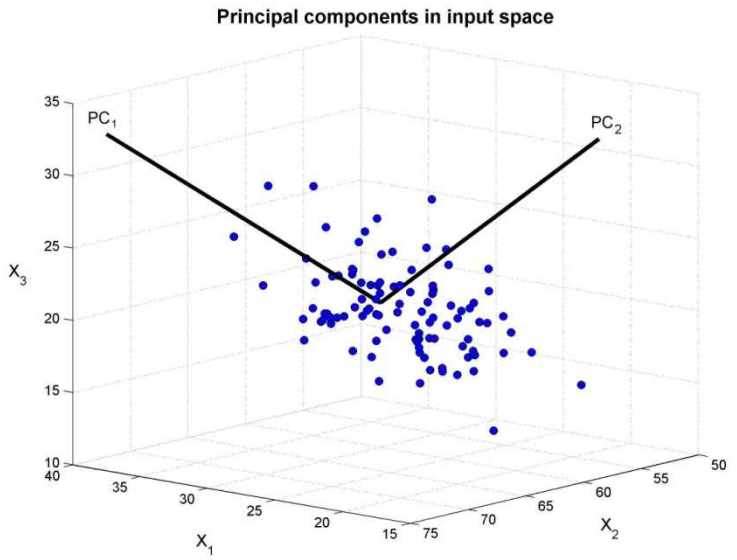

Figure 2.3 : Geometric interpretation of principal component analysis – the first principal component (PC_1) lies in the direction of most variance in the data, with the second principal component (PC_2) orthogonal to PC_1

In the case of two or three principal components accounting for significant total variation (say, 90%), the feature space can be visualized as discussed earlier. If, however, a larger number of principal components are employed, the process can still be visually summarized by the diagnostic score distance and residual distance control charts.

The separation of the score distance and residual distance control charts can be interpreted as the classification of abnormal variation into two parts; process model variation outside its NOC control limits and variation implying a break in the expected correlation of the NOC process data (Dunia & Qin, 1998). An illustration of confidence limits in the score and residual space is shown in Figure 2.4.

The PCA score distance is similar to Hotelling's T^2 statistic, where only a << m projection vectors are retained. As mentioned before, a can be determined by an inspection of the variance decomposition of the training data by principal component number.

A suggested advantage of PCA models are that the score variables obtained are linear combinations of the measurement variables, and as a consequence of the central limit theorem, should show a more normal distribution than the measurement variables themselves. From this deduction, the PCA scores should be approximately normally distributed. However, in the presence of autocorrelation, this assumption and conclusion is no longer valid, as the normal distribution assumes independently and identically distributed data (Wise & Gallagher, 1996).

Fault identification with PCA relies on score distance and residual distance decomposition contribution plots (Russell et al., 2000)[1]. The ease of calculation of these contributions can be attributed to the deterministic and explicit nature of PCA feature extraction.

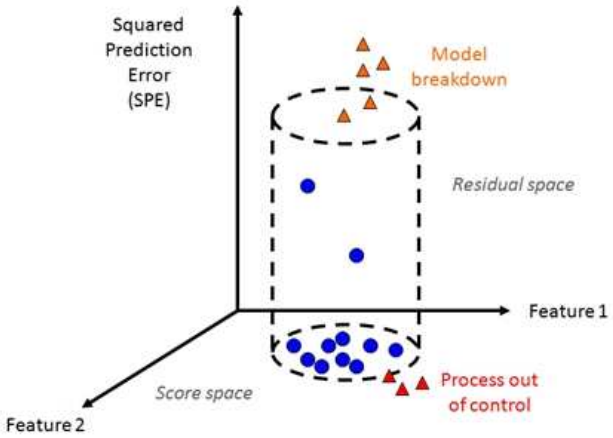

Figure 2.4: Process data decomposition into feature space and residual space, with confidence limits forming a cylindrical normal operating conditions region

Limitations of the PCA approach include its lack of exploitation of autocorrelation (Venkatasubramanian et al., 2003b), and its linear nature. It has been postulated that using linear PCA to extract features from nonlinear data can be inadequate (Dong & McAvoy, 1996). The minor principal components would normally represent insignificant variance in the data for the linear case, but this cannot be said with certainty for nonlinear data. To confidently represent a nonlinear dataset, more principal components must then be retained. This

[1] *More details on contribution plots for PCA are given in Chapter 6*

increases computational requirements. It is also difficult to discern which minor components capture nonlinearity and which represent insignificant variation (Dong & McAvoy, 1996).

The PCA assumption of steady state data is not always valid in chemical and other processes, due in part to possible high sampling frequencies and time-varying behaviour (Ku et al., 1995). In terms of exploiting autocorrelation, a number of variants to the original PCA fault diagnosis method have been suggested (Venkatasubramanian et al., 2003b). An extension of PCA incorporating lagged copies of process variables was termed dynamic PCA (Ku et al., 1995). To account for changes in the crosscorrelation structure of process variables due to time-varying behaviour such as catalyst deactivation, equipment aging and sensor drift, Li et al. (2000) suggested recursive PCA. The PCA model is adapted as new process data become available, incorporating new NOC data into the ever-growing training data set. Another approach is moving window PCA (Wang et al., 2005), where a new model is generated on a constant size training set as new data becomes available.

2.3 Developments in nonlinear feature extractive fault detection

In order to capture the nonlinear nature of process data for fault detection, various feature extraction strategies have been investigated in the last two decades. A large number of these strategies have called on and combined the power and versatility of a variety of statistical learning techniques, including neural networks, the kernel trick and manifold concepts.[2]

- Neural networks

Neural networks are essentially multi-stage regression models, defined by architectures of input, hidden and output layers. Each layer consists of nodes representing activation functions. It is here that nonlinearities can be introduced by using nonlinear activation functions, for example sigmoidal and hyperbolic tangent functions. The training of a neural network is generally achieved through backpropagation to optimize the connection weights (Hastie et al., 2009). Figure 2.5 demonstrates a typical neural network structure.

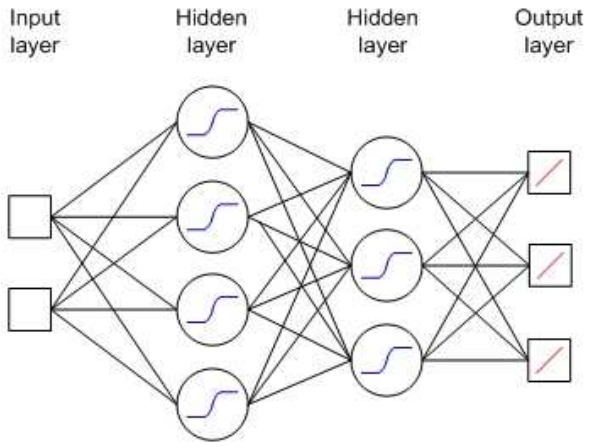

Figure 2.5: Typical artificial neural network topology (multilayer perceptron) with nonlinear hidden nodes and linear output nodes

[2] *The overview of these methods is not meant to be an exhaustive comparison of nonlinear feature extraction methodologies, but rather to highlight different approaches to nonlinear feature extraction.*

- The kernel trick and kernel feature space

The so-called kernel trick exploits the notion that nonlinear data structures can be transformed to linear structures by mapping to a higher dimensional feature space. If a certain mapping from the input space to the feature space can be expressed as a dot product of input space vectors, the explicit mapping function can be replaced by an implicit kernel function. Linear transformation in this implicit feature space translates to nonlinear transformations in the original input space (Lee et al., 2004a).

$$\mathcal{K}(\mathbf{x},\mathbf{y}) = \varphi(\mathbf{x}) \cdot \varphi(\mathbf{y}) \qquad \textit{Eqn. 7}$$

- Manifolds

The manifold concept is especially applicable to constrained, physical data. A data manifold is the lower dimensional hyper-surface to which data are often restricted in high-dimensional physical measurement space. On the manifold surface, data points situated close to each other are physically more related than data points further removed along the manifold. Euclidian distances between data points do not necessarily reflect the manifold structure. So-called geodesic distances, or distances "along the manifold", better represent similarities among data points. Feature extraction in the manifold mindset then consists of determining the dimensionality of said manifold and calculating the features to represent an "unfolded" projection of said manifold (Van der Maaten et al., 2009; Tenenbaum et al., 2000; Roweis & Saul, 2000; Carreira-Perpinan, 1997). Figure 2.6 shows a folded and unfolded two-dimensional manifold.

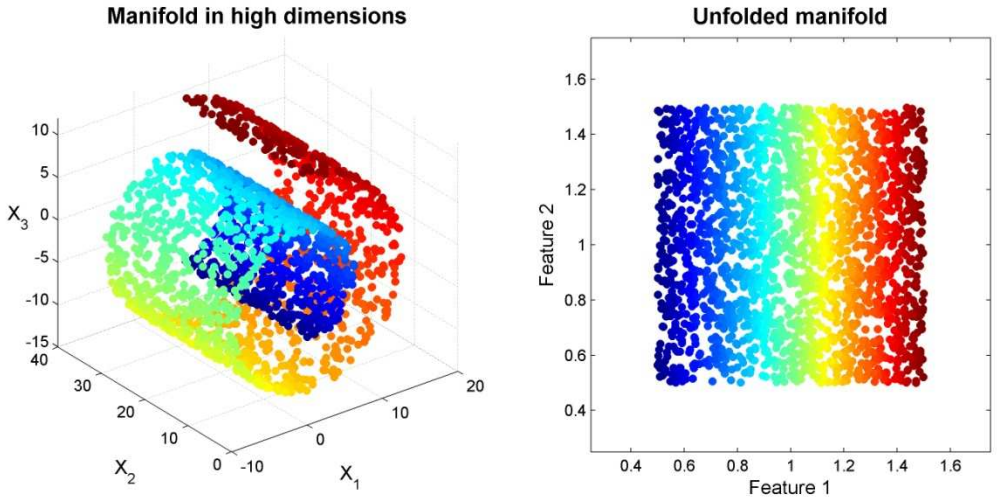

Figure 2.6: An example of a two-dimensional manifold folded in a three-dimensional space: the Swiss roll data structure

If the assumption of an inherent data manifold is valid, a further consideration is the nature of the information obtained from manifold learning techniques. Most nonlinear techniques (especially topology-preserving methods) provide only the smoothed, projected map of low-dimensional features, but no definition of the geometry of said manifold. If said feature space would be further utilised in a fault detection environment, the error between projected data points and measured data points in the original high-dimensional measurement space is required (Chen et al., 2008). Some researchers (Chigirev & Bialek, 2004) suggest manifold description as a doublet of the shape of the manifold and the projected low-dimensional map.

- General notes on feature extractive fault diagnosis

The following general notes are made in aid of the discussion of a variety of feature extraction techniques discussed hereafter:

- Process / original variables refer to the physical measurements made during normal operating conditions and used as input to feature extractive algorithms, represented by the matrix **X** (N samples of m dimensions).
- Features refer to the directional components or manifold definitions extracted by dimension reduction (or expansion, in the case of kernel methods).
- Scores are the sample-specific values along the defined features, represented by the matrix **T** (N samples of a dimensions).
- The mapping function $\mathcal{G}(.)$ calculates the scores from the process variables: $\mathbf{T} = \mathcal{G}(\mathbf{X})$. This is also referred to as projection.
- Reconstructed variables are represented by **X'** (N samples of m dimensions), and are the approximation of the original variables from the scores. This is also referred to as self-consistency.[3]
- The demapping function $\mathcal{H}(.)$ calculates the reconstructed variables from the scores: $\mathbf{X}' = \mathcal{H}(\mathbf{T})$.

Features and their corresponding scores can be extracted either sequentially or in parallel. In sequential extraction, one feature is extracted with the residuals from the reconstructed and original variables used as input to the next feature extraction step. In parallel extraction, all features are calculated simultaneously.

Only techniques that have been applied to static fault detection (and in some cases, fault identification) of process data are included in the following discussion.

2.3.2 Nonlinear PCA with neural networks

Kramer (1991) investigated a nonlinear PCA extension based on autoassociative networks. The feed forward network has an input layer of process variables, a mapping hidden layer of nonlinear activation functions, a bottleneck hidden layer representing NLPCA scores, a demapping hidden layer of nonlinear activation functions and an output layer of reconstructed process variables (see Figure 2.7). Once training of the network weights has been completed through conjugate gradient optimization, the network can be split into separate mapping and demapping networks for unseen data. This approach (Kramer, 1991) did not extend the feature extraction from process data to a fault detection scheme with confidence limits.

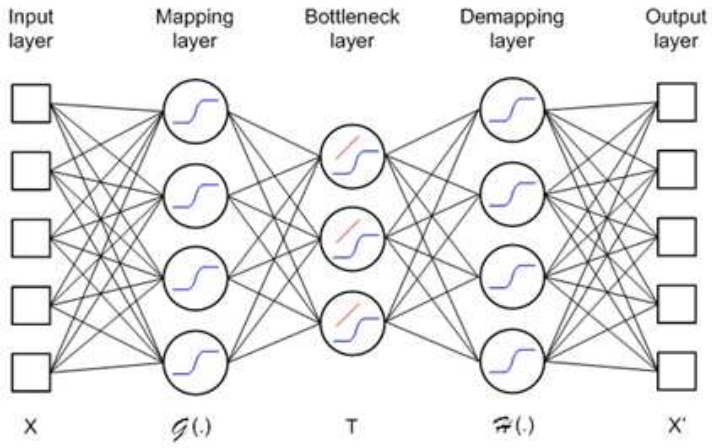

Figure 2.7: Structure of an autoassociative network (Kramer, 1991)

[3] *Note on notation: (.)' represents a reconstruction or estimation of (.); $(.)^T$ represents the transpose of (.)*

An alternative to the autoassociative network approach by Kramer was suggested by Jia et al. (1998). The Input-Training neural network is similar to the demapping section of the autoassociative network, with features as inputs, one demapping hidden layer and the reconstructed process variables making up the output layer. As the nonlinear features are unknown, network learning consists of optimizing the weights and inputs to minimize the error between reconstructed and original process variables. Once the features have been set, a three-layer mapping network is trained for unseen sample processing. Figure 2.8 shows the layout of an Input-Training neural network.

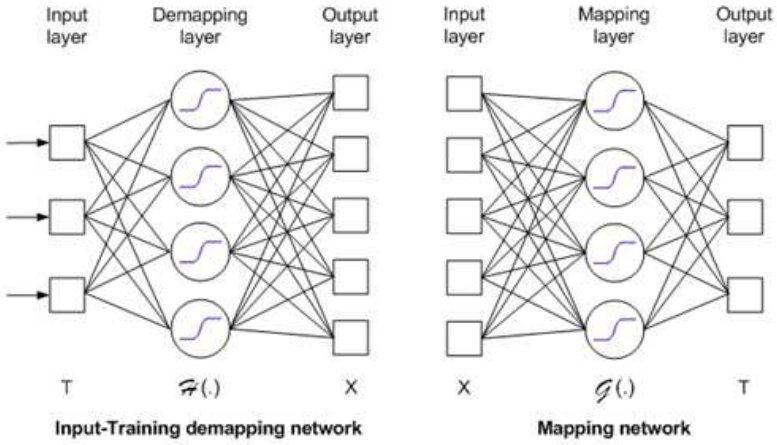

Figure 2.8: Structure of Input-Training neural network (Zhu & Li, 2006)

Autoassociative neural networks are difficult to train, owing to the presence of more than two hidden layers and several approaches have been proposed to overcome these problems. Hsieh (2007) has proposed the assignment of two hidden nodes in the bottleneck layer. The outputs of these nodes are constrained to lie in a unit circle, effectively giving one free angular variable (the nonlinear principal component).

An alternative to the autoassociative network is to use linear PCA scores as input and output for training, termed the T2T network (Antory et al., 2008). The reasoning behind this is that using a reduced number of linear PCA features excludes linearly redundant and insignificant information, as well as simplifying network training due to reduced and well-conditioned inputs. The same network architecture as for Kramer (1991) is used, with the exception of direct connections between the input layer and bottleneck layer and bottleneck layer and output layer (see Figure 2.9).

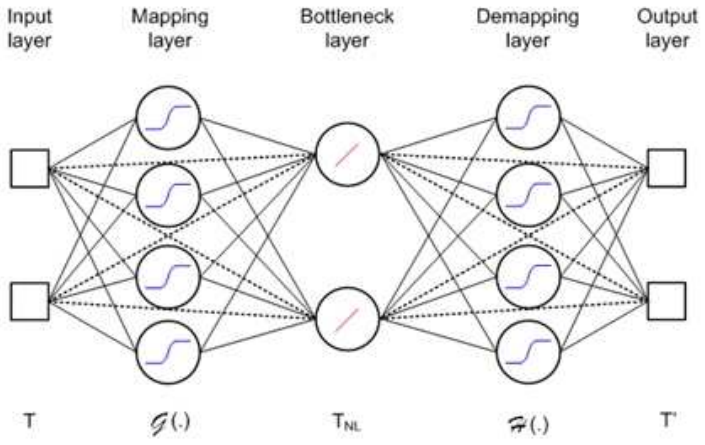

Figure 2.9: Structure of T2T neural network (Antory et al., 2008)

Dong and McAvoy (1996) presented a manifold-motivated NLPCA feature extraction technique. One-dimensional manifolds, principal curves, serve as the nonlinear principal components onto which process data are projected. A principal curve is calculated using a curve smoothing function, the scores on said curve are calculated, and neural networks utilised to construct mapping and demapping functions. The principal curve concept is demonstrated in Figure 2.10. The algorithm is applied sequentially to determine further nonlinear principal curves. A possibly restricting assumption of the principal curve NLPCA approach is that a nonlinear function can be approximated by linear combinations of one given nonlinear function (Jia et al., 1998).

Wilson et al. (1999) expanded on the manifold-motivated NLPCA concept by not extracting one-dimensional principal curve scores sequentially, but extracting higher-dimensional principal manifold scores in parallel. A radial basis function network is used for this mapping function, while m RBF networks are required for demapping.

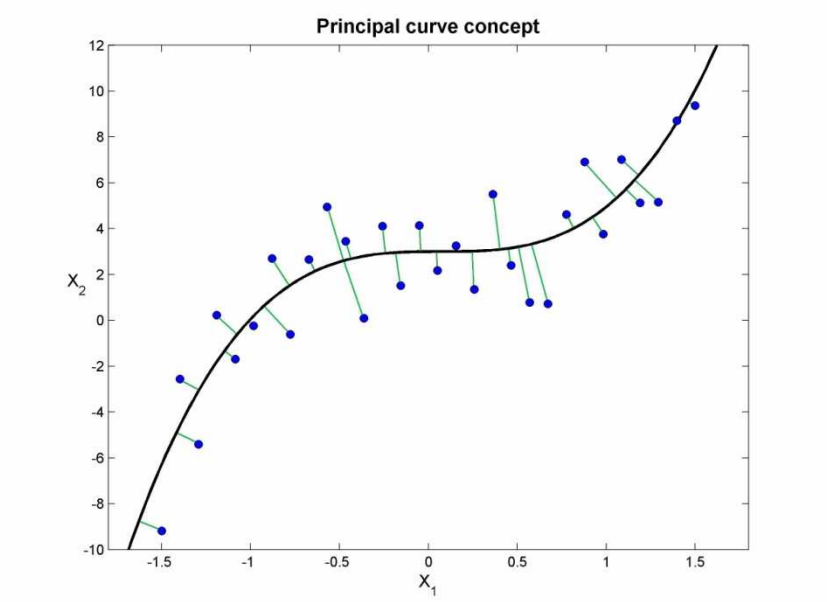

Figure 2.10: Data orthogonally projected to nonlinear principal curve

2.3.3 Kernel PCA and extensions

Identified shortcomings of the neural network methodology for feature extraction are the nonlinear network weight optimization problem and the constraints of having to predetermine the number of features to extract (Lee et al., 2004a). An alternative to network-based feature extraction that addresses these difficulties is kernel PCA. KPCA transforms the original process variables implicitly to a higher-dimensional feature space, where linear PCA can be applied and only significant components retained. The calculation of kernel principal components is an eigenvalue problem, with an explicit solution, as opposed to the nonlinear optimization problem of network-based feature extraction. The number of retained components is determined after the eigenvalue problem has been solved, based on variance decomposition (Lee et al., 2004a; Cho et al., 2005; Choi et al., 2005). Figure 2.11 illustrates the kernel principal component analysis mechanism.

One drawback of KPCA in terms of fault detection and identification is that no explicit demapping function is present to reconstruct nonlinear principal components to the original input space. However, an indication of reconstruction error can be ascertained from the high-dimensional feature space. Some limitations to the KPCA approach is the computational expense of calculating the required dot product for large sample data sets, as well as the lack of interpretability of nonlinear components in the original input space (Cho et al., 2005).

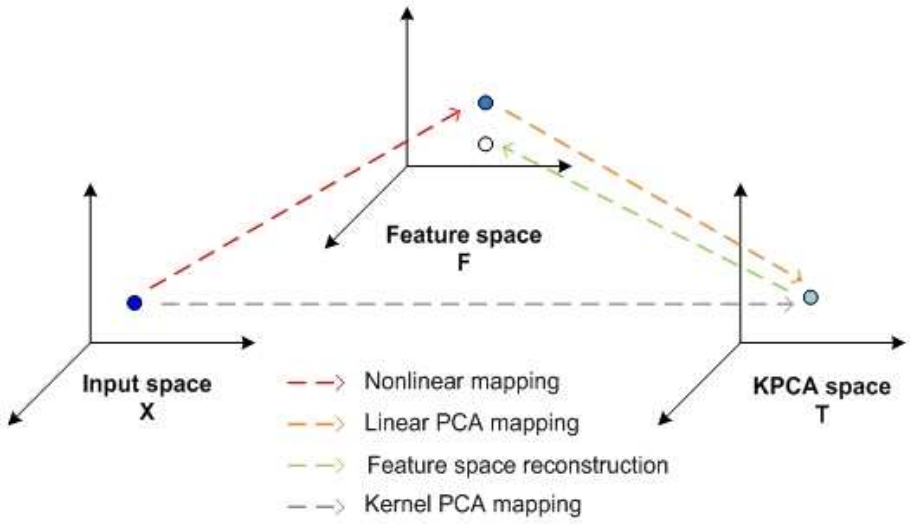

Figure 2.11: Steps of kernel principal component analysis projection and reconstruction (Lee et al., 2004a)

A recent suggested solution to the computational expense of KPCA was suggested (Zhang & Qin, 2008), using geometry-based similarity measures in the input and feature space to decrease the size of the samples retained for training.

A further development in the kernel approach to feature extraction addresses the very nature of PCA: the idea that uncorrelated components are suitable to capture data structure in lower dimensions. PCA uses first and second order statistical moments (mean and variance) to determine linear components, assuming Gaussian data distribution. By using higher order statistical moments (e.g. kurtosis), linear components can be determined to maximize non-Gaussianity. These components are independent, a stricter constraint than decorrelation (Kano et al., 2003; Lee et al., 2004b).

Independent component analysis can be extended to nonlinear feature extraction by calculating independent components in kernel space. This is achieved by first determining significant kernel principal components, and then calculating the ICA mixing matrices based on said kernel components to maximize non-Gaussianity, in this case using negentropy (Lee et al., 2007).

2.3.4 Nonlinear manifolds

A concern raised by the use of KPCA for feature extraction is the use of empirical kernels that do not consider the underlying structure of the process data. Maximum variance unfolding is a modified KPCA technique that learns a specific kernel from training data; this learnt kernel maximizes the variation in kernel space subject to local distance and angles between neighbours constraints (Shao & Rong, 2009).

Many manifold techniques only output scores of the training samples on the manifold surface; without a mapping function for unseen samples or a demapping function to the original variable space. An example of a manifold approach (Shao & Rong, 2009) uses multiple linear regression to map unseen input variables to the manifold features. The inverse of this regression function provides the demapping function. The entire mapping and demapping algorithm is known as maximum variance unfolding projection (MVUP) (Shao & Rong, 2009).

Another recently developed manifold technique aims to exploit the benefits of orthogonal components: generalized orthogonal locality preserving projections (GOLPP). Orthogonal components preserve the global data structure, as well as being well-behaved for reconstruction and contribution purposes. The linear

technique of orthogonal locality preserving projections finds a transformation matrix that minimizes proximity distortion of the projections, with the constraint of orthogonal components. The proximities among data points are computed using an adjacency graph. Extension to nonlinearity can be achieved with this technique by projecting to higher-dimensional kernel space (Shao et al., 2009).

The fundamental assumption underlying the majority of aforementioned techniques is that the data set lies on some (often continuous) lower-dimensional manifold. It has been suggested (Carreira-Perpinan, 1997; Cayton, 2005) that it may be that most data sets do not contain embedded manifolds. Yeh et al. (2005) goes further in proposing that the scenarios in which manifold learning is shown to be successful represents a small and contrived sample of data sets: artificially created dense manifolds and face recognition images where subsequent images are very closely related. Meanwhile, the discontinuous nature of a broad class of real-world data sets effectively shuts out many manifold learning techniques. Another disadvantage is related to the computational expenses in constructing neighbourhood or adjacency graphs.

Topology-preserving techniques in particular require densely populated manifolds in order to learn the manifold correctly, while noisy variables can inhibit learning as well (Yeh et al., 2005). As the dimensionality of a data set increases, the number of samples to obtain a statistically significant model rises exponentially, and aggravates aforementioned constraints (Carreira-Perpinan, 1997; Yeh et al., 2005).

2.3.5 Summary of unsupervised fault diagnosis techniques

The before mentioned techniques and their application to process fault diagnosis are summarised in Table 2.1.

As with PCA these methods can be and have been (for some) extended to incorporate autocorrelation through the dynamic and recursive concepts discussed earlier. For example, KPCA fault detection has been extended to dynamic KPCA (Choi & Lee, 2004) and to moving window KPCA (Liu et al., 2009).

In summary, the shortcomings of the discussed methods are given below:

- Neural networks require the solution of a high-dimensional nonlinear optimization problem, with the main parameter choice that of the number of mapping and demapping nodes.
- Kernel methods have high computational expenses with large training data sets, and selections to be made in terms of which empirical kernel functions and kernel parameters to use.
- Manifold methods may also be computationally expensive due to neighbourhood determination, often involve parameter selection for neighbourhood size, and may not be applicable to discontinuous data.
- All complex unsupervised techniques may suffer from the interpretability of feature scores related to the process generating the training data.

2.3.6 Suitability of feature extraction for fault diagnosis

It has been noted (Venkatasubramanian et al., 2003b) that fault diagnostic methods (whether feature extractive or not) are limited by generalization capability outside of training data. As only a finite sampling of the NOC distribution is available, generalization to unseen data may be uncertain.

Table 2.1: Overview of feature extraction techniques in process fault diagnosis (s_M = score distance based on modified Hotelling's T^2 statistic, r = residual distance, C_s = contributions from score distance decomposition, C_r = contributions from residual distance contribution)

	Model parameters	Number of features[4]	Diagnostics (confidence limits)	Fault identification	Process application
NLPCA (Neural networks) (Kramer, 1991)	mapping and demapping nodes; activation function; optimization criteria	-	-	-	simulated batch reactor
NLPCA (Principal curves) (Dong & McAvoy, 1996; Zhang et al., 1997)	curve calculation; hidden nodes; activation functions	percent variance	r and s (Guassian)	-	Tennessee Eastman; polymerisation reactor
NLPCA (ITNN) (Jia et al., 1998)	mapping and demapping nodes; activation function; optimization criteria	percent variance	r and s (KDE)	C_s and C_r	industrial reactor
NLPCA (Principal surfaces) (Wilson et al., 1999)	smoothing parameter; activation functions; nodes; optimization criteria	percent variance	r and s (Guassian)	C_r	condenser and reflux drum rig
NLPCA (T2T) (Antory et al., 2008)	mapping and demapping nodes; activation function; optimization criteria	MSE cross-validation	r (Gaussian) and s (KDE)	C_r	industrial glass melting process
KPCA (Lee et al., 2004a; Cho et al., 2005; Zhang & Qin, 2008)	kernel function; kernel parameters; similarity thresholds	cut-off at mean eigenvalue; cross-validation	r (Gaussian) and s (KDE)	C_s and C_r	waste water treatment plant; simulated non-isothermal CSTR; Tennessee Eastman; penicillin fermentation
KICA (Lee et al., 2007; Zhang & Qin, 2008)	kernel function; kernel parameters; ICA calculation;	cut-off at mean eigenvalue;	r (Gaussian) and s (KDE)	-	Tennessee Eastman; waste water treatment plant; penicillin fermentation
MVUP (Shao & Rong, 2009)	nearest neighbours; kernel optimization; ridge regression parameter	scree test	r and s (KDE)	C_s and C_r	Tennessee Eastman
GOLPP (Shao et al., 2009)	nearest neighbours; weighting parameter; kernel function; kernel parameters	MSE cross-validation for KPCA; scree test for GOLPP	r and s (KDE)	-	Tennessee Eastman

[4] *The techniques mentioned here to specify the number of features to extract are discussed in Chapter 4*

Nomenclature

$\mathcal{a}$	reduced dimensionality / number of features retained
$\mathcal{G}$	mapping function
$\mathcal{H}$	demapping function
$\mathcal{K}$	kernel function
m	number of process variables
N	number of observations
P	principal components
Q	squared prediction error per variable
r	residual distance; residual space indicator
s	score distance; score space indicator
T	feature matrix
V	matrix of eigenvectors
W	uncorrelated Hotelling's variables
X	process data
X'	reconstructed process data
Λ	matrix of eigenvalues
φ	nonlinear mapping function
Σ	covariance matrix

CHAPTER 3 - RANDOM FORESTS

Random forests are recently proposed statistical inference tools, deriving their predictive accuracy from the nonlinear nature of their constituent decision tree members and the power of ensembles. Random forest committees of models provide more than just a prediction: although not easily interpretable, measures such as variable importance, partial dependence, out-of-bag error estimates and proximities add attractive analysis opportunities. Random forest feature extraction exploits proximity information to enable data structure characterization, in the presence or absence of an associated response variable. The success of random forest models in especially genetic and ecological case studies merits the further investigation of random forest features in process applications.

3.1 Decision trees

A decision tree is a statistical model learnt from a given training data set to perform a classification or regression task. The training data set consists of a number of input vectors **X** and a corresponding response vector Y. Where Y can have a discrete class membership, the tree model is known as a classification tree, while continuous response values are obtained from a regression tree. An algorithm for the learning of classification and regression trees (CART) was first proposed by Breiman and others in 1984 (Breiman et al., 1993).

The essence of the CART algorithm is binary recursive partitioning of the input space. Starting with the entire input space **X**, CART attempts to find a binary partition to increase the response purity in the subspaces formed by the partition. The partition is defined as a hyperplane perpendicular to one of the coordinate axes of **X**. The purity of the resulting subspaces depends on the homogeneity of the response classes (for classification) or the mean squared error from the response subspace average (for regression). Binary partitioning is repeated in each new subspace until subspace response homogeneity is achieved (Breiman et al., 1993).

The successive binary partitions can be expressed as a tree diagram. When training commences, the entire input space **X** is represented as a root node. The first binary split is shown as two branches leading to left and right children nodes. The split is defined as a specific value of a certain variable: if an input has a value smaller than the split, it reports to the left child node, if it is larger, it reports to the right child node. These children nodes now contain only the inputs relevant to their defined subspaces. Repeated partitioning produces a tree with one root node and a number of non-terminal and terminal nodes (Breiman et al., 1993). Figure 3.1 gives an example of a simple classification problem and a classification tree structure associated with this problem.

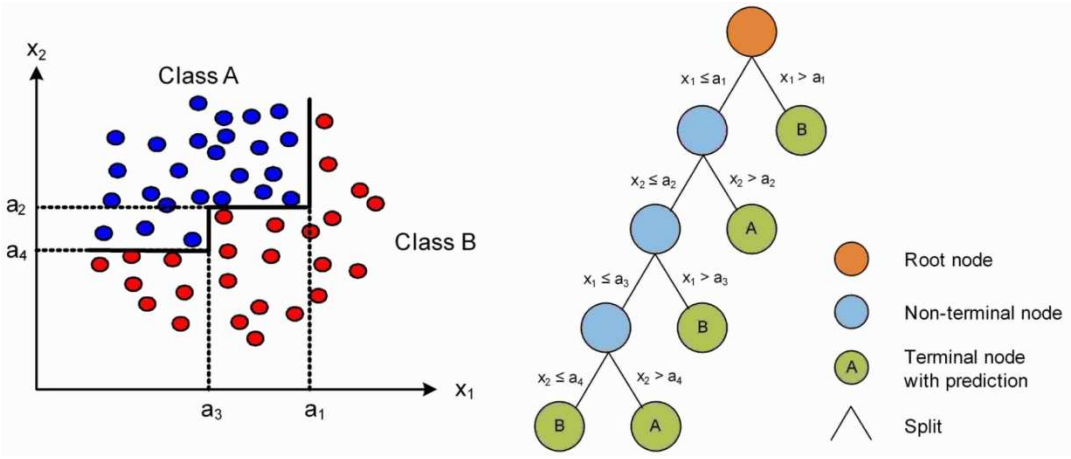

Figure 3.1: Classification problem with decision boundaries and corresponding classification tree

Training of a decision tree consists of three tasks: selecting the split position (partition) at each new node; determining whether a node is terminal or not; and assigning a predicted response to terminal nodes (Breiman et al., 1993).

3.1.1 Split selection

The partition at each non-terminal node is optimized to result in the largest increase in child node purity, or conversely, the largest decrease in impurity. Classification impurity is based on an impurity measure i(η) calculated from the class proportions present at node η.

The impurity measure must be at a maximum where all classes are present in equal proportion, and a minimum where only one class is present. CART utilises the Gini index of diversity as a classification impurity measure (Breiman et al., 1993). An example of the Gini index for a two-class problem is given in Figure 3.2.

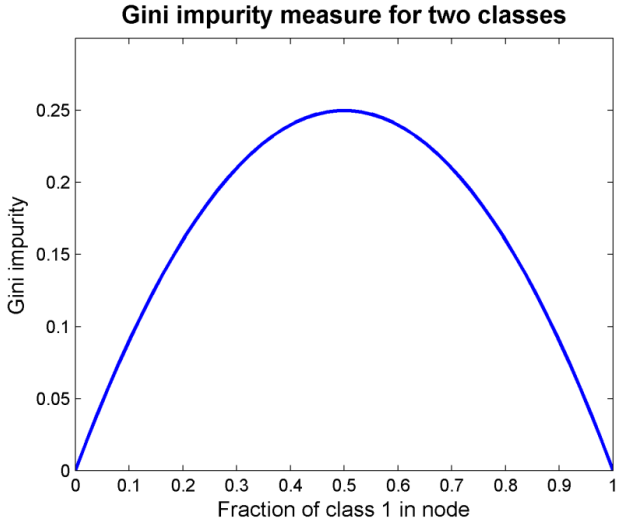

Figure 3.2: Example of Gini impurity measure for two-class problem: the more mixed the classes (i.e. proportion close to 0.5), the higher the impurity measure

The decrease in impurity Δi(ς,η) for a candidate split position ς depends on the impurity measures of the current node, the candidate child nodes and the proportion of samples reporting to the left and right child nodes (p_L and p_R).

$$i(\eta) = \sum_{i \neq j} p(i|\eta)p(j|\eta) \qquad \textit{Eqn. 8}$$

Above, J is number of classes in response Y and p(i|η) is the proportion of class i in node η.

$$\Delta i(\varsigma, \eta) = i(\eta) - p_R i(\eta_R) - p_L i(\eta_L) \qquad \textit{Eqn. 9}$$

At each node, the best split position ς* is selected from the best split positions from each variable that results in the largest decrease in impurity. As the training sample is of a finite size, there are not an infinite number of possible split positions for each variable, but only a maximum of N-1 options per variable (Breiman et al., 1993).

Regression trees do not utilize the Gini index for split selection, but rather minimizes the mean squared error in the children nodes, where the error is the departure from the child node response average (Breiman et al., 1993).

3.1.2 Termination of splitting

There are three motives for declaring a node a terminal node. If all samples in a node have the same response value, there is no rationale to create further partitions. A similar situation arises when only one sample is present in a node. One could also specify a minimum number of samples to be present in all terminal nodes. Another termination criterion requires splitting to stop if a specified minimum decrease in impurity is no longer possible (Breiman et al., 1993).

3.1.3 Terminal node prediction

For classification models, each terminal node prediction will be that of the class majority present. For regression models, the average value of learning sample responses in terminal nodes are assigned as predictions (Breiman et al., 1993).

To predict the response of a new input, the sample is simply "fed" down the tree from the root node, reporting to successive child nodes according to the learnt splits. When the new input reaches a terminal node, the terminal node prediction is assigned (Breiman et al., 1993).

3.1.4 Decision tree characteristics

Decision trees can be seen as models that partition the input space into rectangular subspaces, and assign a simple prediction (subspace class majority or response average) to each subspace. In this sense, a regression tree can be thought of as a histogram estimate of the response regression surface (Breiman et al., 1993). This discontinuous nature can be beneficial in terms of capture of nonlinear and local effects, but detrimental when smooth-function approximation is required (Hastie et al., 2009). Figure 3.3 illustrates the discontinuous nature of decision tree predictions.

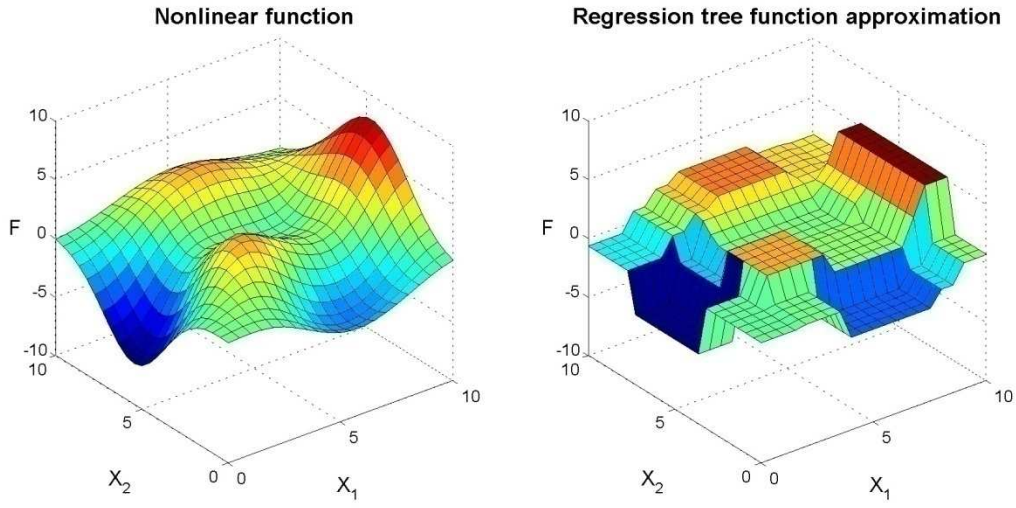

Figure 3.3: Example of regression tree prediction for a nonlinear function:
$F = 3.(1-X_1)^2.\exp[-(X_1^2) - (X_2+1)^2] - 10.(X_1/5 - X_1^3 - X_2^5).\exp(-X_1^2-X_2^2) - \exp[-(X_1+1)^2 - X_2^2]/3$

CART models are conceptually simple, as only an impurity measure and stopping criteria have to be specified. Training is computationally inexpensive, and making predictions for new inputs are even faster, as the tree structure can be summarised as a set of simple if-then rules. These rules and tree diagrams also make the models easily interpretable (Breiman et al., 1993; Ho, 1995).

As splitting is only rank-dependent, decision tree training is invariant to monotone transformations of variables. Both continuous ordered variables and categorical discrete variables can be accommodated as input (Breiman et al., 1993).

However, as terminal nodes can be defined up to a single training point, a large tree might overfit the data. Another drawback is the instability of decision trees. A small change in the learning sample can result in a completely different tree. The reason for this is its hierarchical approach to learning. An error early in the tree construction can change the entire tree structure (Hastie et al., 2009).

As partitions are defined parallel to the coordinate axes, capturing additive structure requires a complicated tree with many levels. If the learning sample is not large enough, such a "diagonal" partition may not be captured satisfactorily and may require more complex tree structures (Hastie et al., 2009). This decision tree characteristic is illustrated in Figure 3.4.

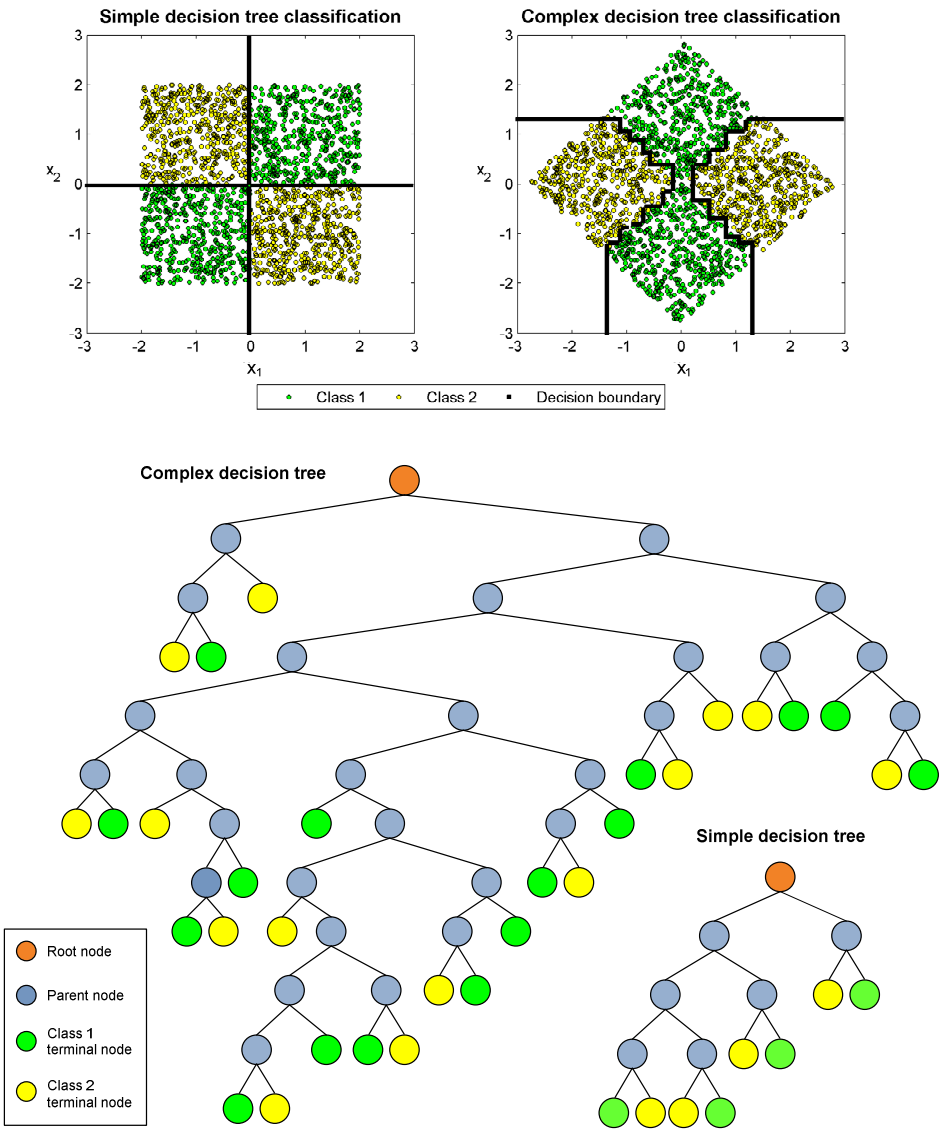

Figure 3.4: Illustration of parallel decision boundary characteristics of decision trees, showing the complex nature of a decision tree trained on data with diagonal decision boundaries (tree structure and terminal node classification are given, with split decisions omitted)

The tree structure of the simple decision tree in Figure 3.4 further shows the limitations of greedy splitting. As entropy decrease is optimized locally over each new split, the most parsimonious tree structure will not necessarily be found with the CART algorithm.

3.2 Ensemble theory and application to decision trees

Decision trees, similar to neural networks and subset selection linear regression, are unstable statistical models, in the sense that a small change in the learning set or model initialization can dramatically affect the final model (Breiman, 1996). This instability can be exploited using ensemble theory.

3.2.1 Combining statistical models

An ensemble algorithm constructs a set of statistical models, and uses majority voting (for classification) or averaging (for regression) to make predictions (Dietterich, 2000). For an ensemble to be more accurate than its members, two necessary and sufficient conditions must be met: the ensemble members must have individual accuracies better than random guessing, and the ensemble members must be diverse. These conditions are known as the strength and diversity of an ensemble (Breiman, 2001). Here, two models are considered diverse if the errors they make on unseen data are uncorrelated (Dietterich, 2000). Ensemble theory is in direct contrast to Occam's Razor, which states that the simplest solution is the right solution.

Unstable statistical models are ideally suited to being collected in an ensemble, as a great variety of different models can be constructed given a finite learning data set. This can be expressed in terms of statistical, computational and representational advantages in hypothesis space (Dietterich, 2000). Figure 3.5 illustrates these concepts.

A learning algorithm attempts to find the best approximation of the true function, given a specific training set, in a space of many possible hypotheses. Statistically, without sufficient data, the learning algorithm may find many different hypotheses that have identical accuracies on the training set. By using all these hypotheses in an ensemble, the risk of choosing the wrong hypothesis is reduced. Computationally, iterative learning may get stuck in local optima in the hypothesis space. By repeating learning from different initial positions, local optima may be circumvented. Finally, the hypothesis space of a single model might fundamentally be unable to represent the true function. This representational problem may be skirted by combining models in an ensemble (Dietterich, 2000).

Decision trees are now considered in light of these hypothesis space constraints. Certain problems may require a large decision tree to give an accurate prediction. The larger the decision tree, the larger the training data set required to ensure a good fit. This represents the statistical constraint. In terms of computational constraints, it has been said that decision trees are unstable, and may arrive at different hypothesis due to small errors. Multiple trees may then explore multiple optima. Lastly, the rectangular partitions of decision trees hinder the successful representation of especially diagonal decision surfaces. However, combining many trees can approximate such structure (Dietterich, 2000).

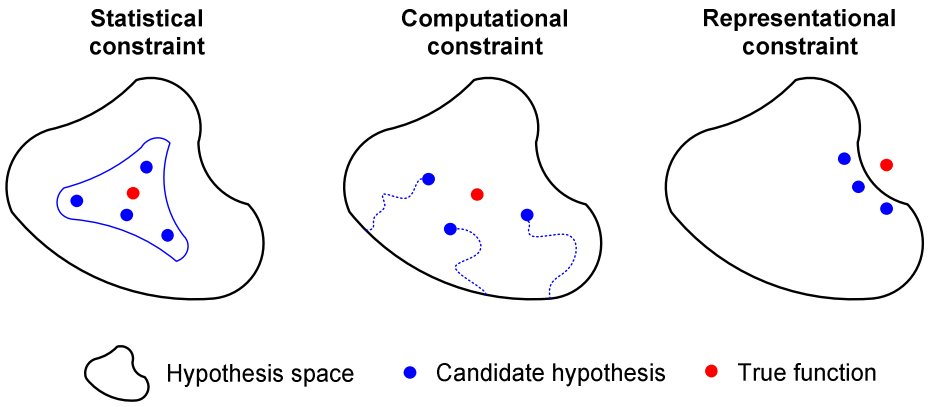

Figure 3.5: Ensemble motivation based on hypothesis space concepts (Dietterich, 2000)

3.2.2 Ensembles of decision trees

Before Breiman introduced his formulation of random forests in 2001, other attempts were made to construct ensembles of decision trees. One approach (using slightly modified diagonal-friendly trees) depends on using different randomly selected subspaces of the original learning set to construct multiple trees (Ho, 1995). Each tree is given only a subset of all input variables available in the learning set to use in split selection. Another approach (Breiman, 1996) manipulates the learning set for each tree through bagging. Bagging is the term for bootstrap aggregating: taking a different bootstrapped sample with replacement from the learning set as modified learning set for each new tree, and aggregating their predictions through voting or averaging. Yet another approach (Dietterich, 2000) was that of randomization: the top twenty splits for each node is calculated, from which a random choice is made regarding which one to use.

All these procedures can be expressed as combinations of decision trees such that the structure of each tree $\mathcal{T}_k$ depends on a random vector θ_k that is independent of past θ, but belong to the same distribution. The k^{th} tree predictior is then $\mathcal{T}_k(\theta_k,\mathbf{X},Y)$ (Breiman, 2001). For bagging trees, θ represents the bootstrap sample identifier; for random subspace trees, θ represents the selection of input vectors available for training, etc.

It has been shown that the generalization error for a decision tree ensemble converges asymptotically to a limit as the number of trees in the ensemble increases, subject to accuracy and diversity constraints as mentioned before. This ensemble limit can be improved by minimizing correlation among members while maintaining their accuracy (Breiman, 2001).

3.3 Random forests

Breiman suggested an improvement on bagging decision tree ensembles by decreasing correlation through further randomization: employing random split selection. This ensemble, known as a random forest, is constructed by not only using different bootstrapped training sets for each tree, but also by restricting the available split variables at each node to M randomly drawn input variables (Breiman, 2001).

3.3.1 Random forest algorithm

The construction of a random forest follows the following algorithm (Breiman, 2001; Hastie et al., 2009):

Random forest classification and regression algorithm

- For k = 1 to K (size of ensemble)
  - Construct a bootstrap sample with replacement $\mathbf{X}_k$ from the learning set **X**, of the same size as the learning set
  - Grow a random forest tree $\mathcal{T}_k$ on $\mathbf{X}_k$ by employing the CART tree growing algorithm, with the following modification at each node:
    - Select m input variables from $\mathbf{X}_k$ to use as possible split variables
    - Calculate best split position ς* from these M variables
    - Split the node into two child nodes
  - Repeat the above tree growing algorithm until the following stopping criteria is achieved:
    - A node size of one (for classification) or five (for regression)
    - A node with homogenous class membership or response values
- Output the ensemble of trees $\{\mathcal{T}_k\}_1^K$
- A new prediction is made as follows:
  - The prediction of each tree h_k is calculated
  - For classification, the majority vote over all K trees is assigned
  - For regression, the average response value over all K trees is assigned

The only model parameters to be specified is the number of trees to grow (K) and the number of random split variables to consider at each split (M).

3.3.2 Model accuracy, strength and correlation

The accuracy of the random forest can be candidly estimated using so-called out-of-bag data. As each tree is grown on a bootstrapped sample $\mathbf{X}_k$ of the original training set **X**, a certain portion (about one-third) of samples will not have been used in training tree $\mathcal{T}_k$. Aggregating votes over tree for only this out-of-bag (oob) data can be used to calculate an honest estimate of generalization error (Breiman, 2001).

Breiman (2001) derives measures of the strength and correlation of trees in a forest based on the margin function of a tree and forest (as applied to classification problems). The strength of a forest relates to the average tree accuracy, while correlation is a measure of tree diversity. The larger the correlation among trees, the lower their diversity. The raw margin function of a tree indicates whether a prediction for an input was right or wrong. The margin function of a forest indicates to what extent a majority vote was achieved for a certain class membership.

The strength of a forest is the average of the margin function for all inputs in the learning set, while the average correlation of a forest is a function of the average correlations and standard deviations of the raw margin functions between all trees (Breiman, 2001).

These measures can be used to ascertain the generalization error, member accuracy and member diversity of a random forest ensemble for different data sets and parameter values.

3.3.3 Parameter selection

Only two model parameters are required when constructing random forests: the number of trees K and the number of random split variables M.

As the generalization error approaches a limit asymptotically in terms of increasing number of trees (Breiman, 2001), this parameter can be selected to be sufficiently large so that oob error no longer improves significantly

with additional trees. Breiman and Cutler (2003) suggest using large numbers of trees (1000 to 5000), especially if additional model properties are to be assessed.

In terms of the number of split variables, it is intuitive that choosing smaller values for m will increase the diversity (at least structural, if not in predictions) of trees (Hastie et al., 2009). However, the accuracy of a tree may be decreased for small m. Breiman (2003) suggests using m equal to the square root of the number of inputs for classification.

3.3.4 Model interpretation

Although decision trees are easily interpreted due in part to their rule-based construction, this characteristic does not extend to random forests. As a forest consists of many different decision trees, a simple model structure is not evident (Breiman, 2001).

However, random forests do contain a wealth of information that can be exploited. Two facets of this are variable importance measures and proximities.

- Variable importance and partial dependence

The relative importance of the different input variables in the construction of a random forest model, and by extension, in describing the structure of the learning set, can be obtained from a random forest in one of two ways (Breiman & Cutler, 2003; Archer & Kimes, 2008):

- For each decision tree, at each split, the decrease in the Gini measure is recorded for the relevant split variable. The average of all Gini decreases over the forest for variable j gives the Gini variable importance measure for that variable.
- For each variable and each decision tree, create a testing set $\mathbf{X}_{j,oob}$ that is identical to the oob input values $\mathbf{X}_{oob}$ for that tree, with the exception that variable j has been randomly permuted. Predict class membership for $\mathbf{X}_{oob}$ and $\mathbf{X}_{j,oob}$, and sum over the entire forest the number of incorrect votes due to permutation of variable j. This yields the oob accuracy variable importance measure.

It has been shown that random forest variable measures are affected by the scale and number of categories of an input variable (Strobl et al., 2007). As an example, the Gini variable importance measure inflates importance for continuous variables or categorical variables with many levels (Archer & Kimes, 2008). However, for standardized, continuous and correlated input variables, the random forest variable importance measure is considered useful (Archer & Kimes, 2008).

Once important variables have been identified, the nature of its influence on the response is considered. Friedman has suggested an approach to investigating the influence of a variable on a response (given any predictive model): the partial dependence plot (Friedman, 2001). It is useful to inspect a plot of the predicted response for the range of values of a specific input variable, as averaged over all training values of the other input variables (known as the marginal average for a specific variable). Identification of important variables, combined with partial dependence plots for these important variables, provides a power visualization and interpretation tool for random forest models.[5]

[5] *Variable importance and partial dependence methodologies will be extensively discussed, evaluated and applied in Chapters 9 and 10.*

- Proximities

Another aid to data interpretation arising from random forests is found by inspecting the terminal node characteristics of all trees. By investigating whether samples report to the same leaf nodes, one can construct a proximity measure for all points. A leaf node essentially spans a hyperrectangle in the input space; if two samples report to the same hyperrectangle, they must be proximal. The algorithm for constructing a proximity matrix is summarized below (Breiman & Cutler, 2003):

Random forest proximity algorithm

- Construct a random forest model on learning set **X** with response Y
- Create an empty similarity (proximity) matrix **S**
- For each tree k = 1 to K
    - For each sample combination (i,j), determine the terminal nodes to which they report
    - If a sample combination (i,j) report to the same terminal node, increase $S_{i,j}$ by one
    - Repeat for all possible sample combinations
- Scale the similarity matrix **S** by dividing by the number of trees K; the similarity matrix is symmetric and positive definite, with entries ranging from 0 to 1, and diagonal elements equal to 1

This similarity matrix can be converted to a dissimilarity matrix: **D** = 1 – **S [*Eqn. 10*]** (Breiman & Cutler, 2003) or **D** = √(1 – **S**) **[*Eqn. 11*]** (Shi & Horvath, 2006). The dissimilarity matrix may be considered as Euclidian distances in high dimensional space, and multidimensional scaling employed to obtain a lower dimensional representation of the distances between data points (Breiman & Cutler, 2003).

Given a dissimilarity matrix, multidimensional scaling (Cox & Cox, 2001) attempts to find coordinates for data points that preserve the dissimilarity as Euclidian distances in the found coordinate space. The preservation of dissimilarities is measured by a stress function, the sum of squared differences between point dissimilarities and the new coordinate distances (Hastie et al., 2009). Classical multidimensional scaling finds an explicit solution to the stress function by utilizing eigenvalue decomposition. The eigenvectors are the MDS coordinates, while the eigenvalues give an indication of every coordinate's contribution to the squared distance between points. Smaller eigenvalues thus indicate less significant coordinates, and can be inspected in order to select only significant coordinates (Cox & Cox, 2001). Classical MDS is not considered a manifold technique, as it explicitly attempts to minimize all pairwise distances (Hastie et al., 2009). These MDS coordinates can be considered as random forest features to represent the data in low dimensions.

Hastie et al. (2009) have noted that random forest proximity plots look very similar for different classification data sets. Each class seem to be represented by one arm of a star, with points in pure class regions mapping to the extremities of the star, while points closer to the decision surface maps near the centre of the start. They deduce that this is due to data points in pure class regions reporting to the same leaf nodes, while the uncertainty close to a decision surface reduces the likelihood of proximity between points. Even so, proximity plots are useful in classification and regression problem data visualization, especially in identifying cluster structure and separation (Cutler, 2009).

Proximities generated using random forests do not only measure similarity, but incorporate variable importance as well. As an example, samples with different values on certain input variables may still report to the same terminal nodes, if said differences occur only in unimportant variables (Cutler, 2009).

Decision trees in random forests are not pruned. This implies that proximities based on all data may overfit, as samples from different classes will have proximities of zero for a tree where they served in the training data set. A manner to circumvent this problem may be to only use oob data to calculate proximities (Cutler, 2009).

3.3.5 Strength and correlation of random forests

As the predictive performance of an ensemble is dependent on the accuracy and diversity of its members, it may be insightful to inspect accuracy and diversity measures of random forests. In the original formulation of random forests (Breiman, 2001), measures for strength and correlation are defined based on so-called margin functions. These measures as defined for two-class classification problems will now be discussed.

The raw margin function m_k for a tree $\mathcal{T}_k(\theta_k,\mathbf{X},Y)$ in a random forest is determined from the tree prediction $h_k(\mathbf{X},\theta_k)$ defined as:

$$m_k(\theta_k,\mathbf{X},Y) = I(h_k(\mathbf{X},\theta_k) = Y) - I(h_k(\mathbf{X},\theta_k) \neq Y) \qquad \textit{Eqn. 12}$$

Above, I(.) is an indicator function. The raw margin function is then +1 for an input that is correctly classified and -1 for an input that is incorrectly classified.

The margin function $\mathcal{M}$ for an entire forest of trees is the average of all raw margin functions in that forest:

$$\mathcal{M}(\mathbf{X},Y) = \frac{\sum_{i=1}^{K} m_i}{K} \qquad \textit{Eqn. 13}$$

The strength Ω of a random forest is the average of the margin function for all training inputs:

$$\Omega(\mathbf{X},Y) = \frac{\sum_{i=1}^{N} \mathcal{M}(\mathbf{X_i},Y_i)}{N} \qquad \textit{Eqn. 14}$$

The correlation $\rho(\theta_i,\theta_j)$ between two trees $\mathcal{T}_i(\theta_i,\mathbf{X},Y)$ and $\mathcal{T}_j(\theta_j,\mathbf{X},Y)$ is the correlation between the raw margin functions of these two trees, m_i and m_j, for a given input and response set ($\mathbf{X}$,Y). The average correlation P for all trees in a random forest is defined as:

$$P(\mathbf{X},Y) = \frac{\sum_{j=i+1}^{K}\sum_{i=1}^{K}[\rho(\theta_i,\theta_j)sd(m_i)sd(m_j)] \Big/ sd(m_i)sd(m_j)}{K(K-1)} \qquad \textit{Eqn. 15}$$

Here, sd(.) indicates a standard deviation calculation. The correlation to strength ratio of a random forest is defined as the forest correlation divided by the square of the forest strength, P/Ω^2.

3.3.6 Strengths and weaknesses of random forests

Random forest classification models are a surprisingly powerful technique (Hastie et al., 2009). This is the paraphrased response of the writers (Hastie et al., 2009) of *The Elements of Statistical Learning* to the performance of random forests on the Neural Information Processing Systems (NIPS) classification competition. The challenge required five data sets with 500 to 100 000 variables and 100 to 6 000 samples to be classified. Random forests obtained an average rank of 2.7 for classification accuracy and 1.9 for computational time, competing with the likes of Bayesian neural networks, boosted trees, boosted neural networks and bagged neural networks (Hastie et al., 2009). The creator of random forests (Breiman, 2001) is the first to acknowledge that random forest regression is less successful than its classification counterpart.

The strengths of random forests is not limited to its classification accuracy, but also extends to the very little tuning required (Hastie et al., 2009); its automatic calculation of an generalization error; an ability to handle

missing data; variable importance measures and the application possibilities of proximities (Breiman & Cutler, 2003).

However, variable importance measures do not substitute entirely for model interpretability. It has also been shown that random forest performance may be reduced when a large number of noisy variable are present, and a small number of random split variables are used (Hastie et al., 2009).

3.4 Random forest feature extraction: the unsupervised approach

A feature extraction technique must be able to compute significant features for data sets where no response variables are present. This forms part of unsupervised learning, whereas supervised learning attempts to fit input variables to a given response. Random forest classification and regression are supervised learning methods, and require some adaptation to be employed to the unsupervised case.

3.4.1 Unsupervised learning as supervised learning

The application of supervised methods to unsupervised problems is discussed by Hastie et al. (2009). In their discussion, they use density estimation as an example of an unsupervised learning. The concepts discussed can be extended to the characterization of multivariate structure as an unsupervised problem.

If g is the unknown data density to approximate, let g_0 be any known reference distribution. A synthetic data set can be constructed by sampling g_0. A supervised task can now be assigned to build a model that can distinguish between data generated by g and g_0. The model predictions can be inverted to provide an estimate for g. [6] In essence, the supervised model provides estimates on the departure of g from g_0. The selection of g_0 is dictated by what type of departure is considered interesting. Departures from uniformity or multivariate normality can be investigated by choosing the corresponding distributions for g_0, while departures from independence between variables can be ascertained by choosing the product of marginal distributions for g_0. (The product of marginal distributions is simply the resampling, for each variable, from its own values, independent from other input variables).

3.4.2 Unsupervised random forests

The concept of unsupervised learning is further extended to random forests (Breiman & Cutler, 2003), with a further study (Shi & Horvath, 2006) investigating its properties extensively.[7] The unlabeled original data is given a class label, say 1. A synthetic contrast data set is created, and given a different class label, say 2. A classification random forest can now be constructed which attempts to separate the two classes. From this forest, proximities can be calculated for the original unlabeled data, using the aforementioned proximity procedure. Obtaining dissimilarities from the similarity matrix, multidimensional scaling is applied to produce a set of features (coordinates) and their corresponding eigenvalues. These features now represent a projection of the original unlabeled data. An example of a proximity matrix and random forest features plot can be found in Figure 3.6.

The random forest feature extraction algorithm is summarized below (Shi & Horvath, 2006):

---

[6] *See 'The Elements of Statistical Learning' (Hastie et al., 2009) for details.*

[7] *Shi and Horvath (2006) focused on the clustering utility of random forest proximities, a subtle difference to general feature extraction applications. Here, clustering refers to the ability of a feature extraction method to generate projections where known clusters are separate, without using cluster information in training.*

Random forest feature extraction algorithm

- For an unlabeled learning set **X**, create a synthetic data set $\mathbf{X}_0$ by random sampling from the product of marginal distributions of **X**
- Label the response of **X** inputs as class 1 and the response of $\mathbf{X}_0$ inputs as class 2
- Concatenate input matrices **X** and $\mathbf{X}_0$ as **Z**, with concatenated response Y
- Construct a random forest classification model to predict Y given **Z**
- Create an empty similarity (proximity) matrix **S**
- For each tree k = 1 to K
    - For each sample combination of $\mathbf{X}_{i.}$ and $\mathbf{X}_{j.}$, determine the terminal nodes to which they report
    - If a sample combination of $\mathbf{X}_{i.}$ and $\mathbf{X}_{j.}$, report to the same terminal node, increase $S_{i,j}$ by one
    - Repeat for all possible sample combinations
- Scale the similarity matrix **S** by dividing by the number of trees K; the similarity matrix is symmetric and positive definite, with entries ranging from 0 to 1, and diagonal elements equal to 1
- Determine the dissimilarity matrix **D** = 1 – **S**
- Use the dissimilarity matrix **D** as input to classical multidimensional scaling, retaining a scaling coordinates **T** as random forest features

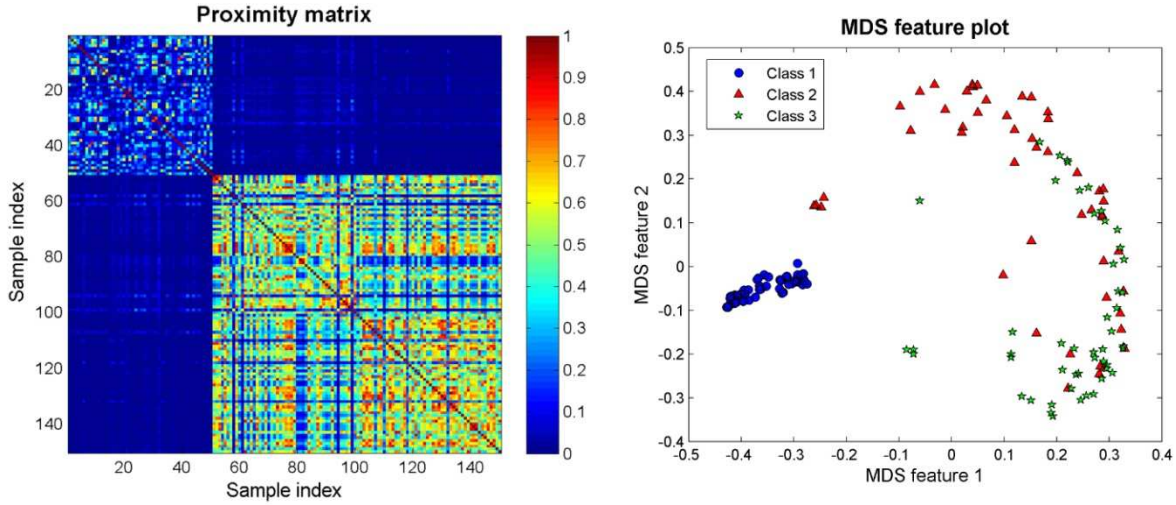

Figure 3.6: Example of random forest proximity matrix and corresponding multidimensional scaling features for unsupervised three-class data set (class information added post priori to features)

3.4.3 Specifying random forest feature extraction models

A number of decisions must be made when constructing an unsupervised random forest in order to extract features: what type of contrast distribution to use, the size of the forest (K) and the number of random split variables (M).

- Contrast distribution

As with unsupervised learning applied to density estimation, the choice of the reference distribution for the synthetic data set determines what is considered as interesting structure. Shi and Horvath (2006) considered

the uniform and product of marginal densities. If a uniformly distributed synthetic class is used, variables that show great departure from uniformity are often selected during tree construction. If however, the product of marginal densities is used, trees will select splitting variables that are dependent on other variables.

It was found that generating a synthetic class using the product of marginal densities provided the most insightful scaling coordinates for a variety of data sets. However, redundant collinear variables may be detrimental to interpreting scaling plots (Shi & Horvath, 2006).

Using the product of marginal densities to create a synthetic class destroys the multivariate structure of the original data. If the original data had little multivariate structure to start with, the synthetic data will be similar to the original data, and a random forest model will be similar to random guessing, with an accuracy close to 50 %. Higher accuracies may be indicative that the original data contained dependent multivariate structure (Cutler & Stevens, 2006). This leads to the observation that, in general, there is an inverse relationship between the oob error rate of an unsupervised random forest and the cluster signal in its proximity plot (Shi & Horvath, 2006).

- Size of forest

The elements of the similarity matrix are averaged over all trees in the forest, suggesting intuitively that the more trees, the more statistically sound the similarity. As previously discussed, random forest performance does not deteriorate as the number of trees increase. The only constraint on forest size is determined by computational resources. Shi and Horvath (2006) typically used 2000 trees per forest for random forest feature extraction. However, they recommended that more than one forest should be grown, and the resulting similarity matrices averaged. This ensures that the similarities are not dependent on a single manifestation of the synthetic contrast.

- Number of random split variables

The clustering performance of random forest feature extraction was shown to be robust to the number of random split variables. Very low and very high values for M were not conducive to good clustering, while selecting M which corresponds to a low oob error rate did not improve clustering (Shi & Horvath, 2006).

3.4.4 Interpreting random forest features

The presence of clusters in a random forest feature plot may not be indicative of the presence of clusters in the original data. It is also difficult to provide a geometric interpretation of random forest features (Shi & Horvath, 2006). This may be due, in part, to the disjoint nature of decision tree partitions.

As proximities are linked with variable importance (Shi & Horvath, 2006; Cutler, 2009), an aid to interpretation may be to investigate corresponding variable importance measures.

As with feature extractive methods such as maximum variance unfolding, random forest feature extraction does not provide a direct mapping function from input variables to features. Separate mapping and demapping regression functions must be learnt to project new data points.

3.5 Application of random forests

Random forests have been widely applied in classification, somewhat utilized as a regression method, with a small number of studies focusing on unsupervised feature extraction (mainly in aid of clustering). A selection of these applications is discussed below.

- Classification

The use of random forest as a classification tool has become wide spread since its inception in 2001, with a diversity of applications and overall good performance compared to other classification techniques.

A popular and successful application of random forest classification is in the prediction of species distribution for future climate scenarios (Cutler et al., 2007; Garzón et al., 2006; Peters et al., 2007). In these studies, random forest classification performance was shown to be the same, and often better, than other classification techniques (such as linear discriminant analysis, logistic regression and neural networks).

Random forest classification has also been applied to predicting chemical properties such as reactivity and biological activity from molecular structure (Svetnik et al., 2003; Sakiyama et al., 2008; Gupta et al., 2006). Again, comparable and improved classification performance was achieved with random forests as compared to decision trees, support vector machines, neural networks and quantum modelling.

A major application of random forest classification has been to genetics, especially to microarray and DNA sequence data (Diaz-Uriarte & Alvarez de Andres, 2006; Pang et al., 2006; Cummings & Segal, 2004). Random forests were compared to decision trees, bagging, linear discriminant analysis, support vector machines, neural networks, k-nearest neighbours and others, and delivered the best or equivalent performances.

- Regression

Random forest regression has not found as wide-spread application as its classification counterpart, but also shows good performance in species distribution prediction (Prasad et al., 2006), biological activity prediction (Svetnik et al., 2003) and genetic applications (Bureau et al., 2003) compared to partial least squares, neural networks, multivariate adaptive splines and other techniques.

- Variable importance and supervised feature extraction

In most of the above mentioned studies, random forest variable importance measures were calculated to aid in interpretation of the given problem. Proximity plots are rarely utilised to aid in data visualization when a supervised model has been trained, with few exceptions (Svetnik et al., 2003; Pang et al., 2006; Granitto et al., 2007; Finehout et al., 2007; O'Riordan et al., 2004; Raza et al., 2006).

- Unsupervised feature extraction

Unsupervised random forest feature extraction is also in its infancy. Most applications utilize random forest proximities and features in clustering algorithms. One application was the visualization of DNA microarray data with random forest features to aid in breast cancer classification (Abba et al., 2007). Amaratunga et al. (2008a) compared unsupervised random forest based clustering to other clustering techniques on a variety of microarray data, and showed ordinary results for the random forest method. Bonissone and Iyer (2007) suggest using unsupervised random forest feature extraction for anomaly detection in aircraft flight event logs.

Curnow et al. (2005) used random forest feature extraction to cluster molecular information to distinguish eye disease severity. Qi et al. (2005) used unsupervised random forest similarities in a clustering algorithm to classify protein interactions, and showed promising results for the random forest and k-nearest neighbour approach, compared to an equivalent Euclidian distance based approach. Seligson et al. (2009) determined important factors affecting prostate cancer recurrence by identifying clusters in an unsupervised random forest feature plot. Shi et al. (2005) did a similar analysis on factors correlated to kidney cancer.

Nomenclature

a	reduced dimensionality / number of features
D	dissimilarity matrix
g	data density distribution
h	tree prediction
i	impurity measure
J	number of classes
K	ensemble size
k	ensemble member indicator
L	left node indicator
M	number of random split variables
$\mathcal{M}$	margin function
m	dimensionality of **X** / number of input variable
m	raw margin function
N	number of samples
p	proportion of samples
R	right node indicator
S	similarity / proximity matrix
T	feature matrix
$\mathcal{T}$	tree function
X	input data matrix
Y	output / response vector
Z	concatenated matrix of **X** and its permutation
η	node
θ	random vector
P	average correlation of forest
ρ	correlation between trees
ς	split position
ς*	best split position
Ω	strength of forest

CHAPTER 4 - OVERVIEW OF METHODOLOGY

This chapter summarizes the methodology applied in this work to develop, test and analyse fault and change point detection frameworks based on random forest feature extraction and other random forest interpretive tools.

The first proposed study aims to validate the feature extractive performance of random forests, comparing random forest feature extraction to a selection of diverse feature extraction techniques on a variety of data sets. The second study will introduce and apply fault detection and identification algorithms based on random forest feature extraction. The third study will focus on the identification and interpretation of process variables involved in detected faults, by exploiting the variable importance and partial dependence tools of supervised random forests. The fourth and final application study considers dynamic process monitoring, and will show the development and application of a random forest method analogous to a linear approach to change point detection.

4.1 Overview

The hypothesis of this work is restated here:

Random forests constitute a viable basis for the development of nonlinear process monitoring and fault diagnosis methods.

With the following key objectives:

- A critical literature survey on feature extraction techniques in fault diagnosis; as well as the random forest modelling and feature extraction approach
- The assessment of the quality of features extracted with random forests
- The development of unsupervised fault diagnostic schemes using random forest feature extraction
    - The implementation of random forest fault diagnostic schemes as robust code
    - The testing of the random forest fault diagnostic schemes on simulated data, a benchmark process engineering problem and a real-world mineral processing data set
- The development of interpretive tools for identifying and interpreting important process variables involved in process changes and faults
    - The implementation of interpretive variable importance and visualization tools based on random forests as robust code
    - The testing of random forest interpretive tools on simulated data, a benchmark process engineering problem and a real-world mineral processing data set
- The development of a dynamic random forest change point detection scheme
    - The implementation of the dynamic random forest change point detection as robust code
    - The testing of the dynamic random forest change point detection scheme on simulated static and dynamic data sets

The literature survey in Chapters 2 and 3 gave details on feature extractive fault diagnosis and random forests. In terms of the remaining objectives, this chapter gives a brief overview of the methodologies applied to the feature quality, fault diagnosis and change point detection studies.

It was shown in Chapter 2 that fault diagnosis considered in a feature extractive framework entails the extraction of information-rich features to segment a process data set into feature and residual spaces, as shown in Figure 4.1.

Changes in the feature and residual spaces can be monitored by diagnostics summarizing the normal operating condition data distribution in the respective spaces. Fault detection is achieved when new process data exceeds thresholds of these diagnostics. With principal component analysis, the feature space diagnostic is a modified Hotelling's T^2 statistic, while the residual space diagnostic is a squared prediction error. Identifying the original process variables involved in the fault is achieved by looking at the contribution of individual variables through decomposition of the feature and residual space diagnostics, with so-called contribution plots. The limitations of linear methods to nonlinear data were discussed in Chapter 2, motivating the investigation of prospective nonlinear feature extraction techniques.

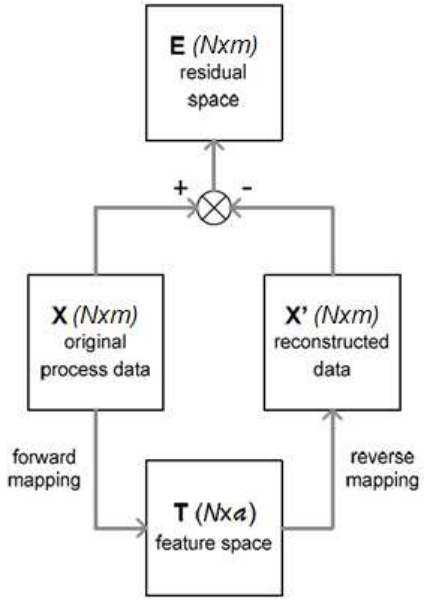

Figure 4.1: Schematic of feature extraction approach to segmenting process data into feature and residual spaces

With the view on applying random forests to this framework, Chapter 3 presented the background of random forests. The various aspects of random forests are summarized in Figure 4.2. It is clear that random forests provide more than merely features; tree structures can potentially capture complex nonlinearities, with interpretation of the model's nonlinear nature available in the form of proximities, variable importance and partial dependence.

In order to apply random forests to a feature extractive fault diagnosis framework, a number of design decisions must be considered. As a precursor, the quality of random forest features is compared to PCA and a selection of nonlinear mapping techniques. A fault detection framework is then developed incorporating unsupervised random forest proximities and supervised random forest regression. The interpretive tools of random forests are then exploited in fault identification frameworks. Finally, random forest feature extraction is applied to the specialized fault detection problem of change detection.

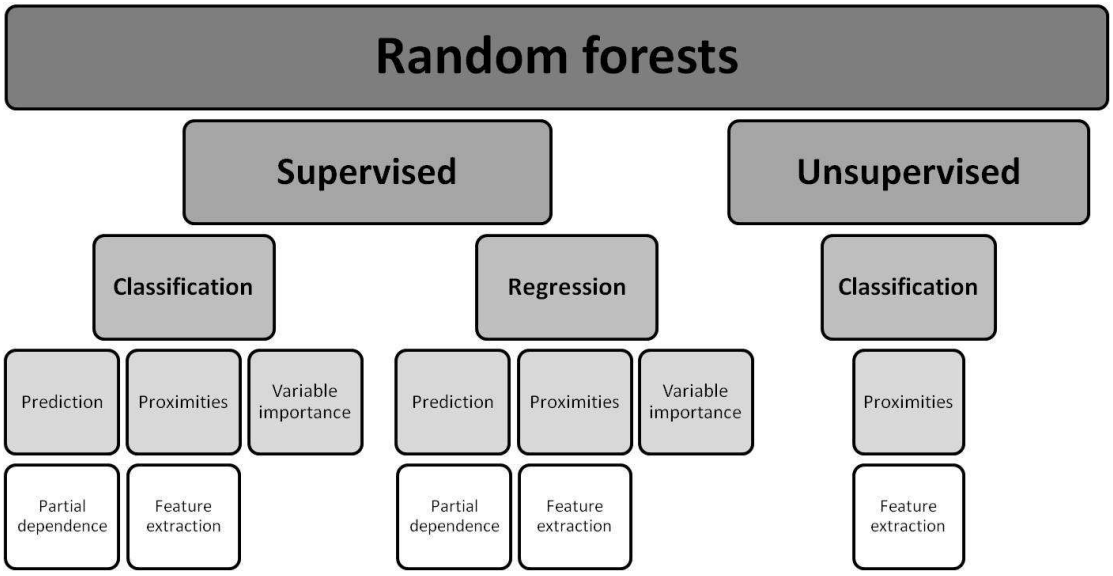

Figure 4.2: Schematic showing random forest functionality for supervised and unsupervised approaches

4.2 Quality of random forest features

To assess the quality of random forest features, random forest feature extraction is compared to one linear (PCA) and four nonlinear (kernel PCA, Sammon mapping, Isomap and locally linear embedding) feature extraction methods, as applied to seven data sets. These mapping methods, and feature extraction considerations in general, are discussed in Chapter 5.

Analyzing the performance of random forest features is done in terms of quantitative and qualitative comparison to other mapping techniques. Quantitative comparison is made in terms of local structure preservation (as embodied by the 1-nearest neighbour error in feature space) and global structure preservation (as represented by linear reconstruction correlation, nonlinear reconstruction correlation and pairwise distance correlation). Qualitative comparison is possible through inspection of two-dimensional feature plots, with class information added post priori.

Random forest feature extraction is further investigated by determining the effect of parameters M (random split variables) and K (number of ensemble members) on aforementioned quantitative performance measures. The effect of these parameters on the strength-correlation ratio of the ensembles is also investigated.

Finally, a simple simulated data set is used to evaluate the nature of random forest feature extraction mapping and demapping models based on random forest regression.

Chapter 5 presents all details and results for the study on the quality of random forest features.

4.3 Fault detection and identification

In order to use random forest features in a feature extractive fault diagnosis framework (as shown in Figure 4.1), a number of modifications are required to the general PCA fault diagnosis methodology summarized in Chapter 2. These modifications are related to the implicit nature of random forest mapping, the definition of a feature space diagnostic suited to non-Gaussian distributions, and the interpretation of detected faults.

4.3.1 Fault diagnosis framework

Chapter 6 provides the details on the development of a fault diagnosis framework, with the above design decisions incorporated. The problem of implicit random forest mapping is circumvented by employing random forest regression as mapping and demapping functions. One-class support vector machines modelling is introduced as a data density estimation technique suitable to non-Gaussian data, and thus providing a feature space diagnostic for the random forest approach. A detailed comparison to the PCA framework is provided, as PCA will act as the benchmark process monitoring technique in the application study. Performance indicators in the form of false alarm rates, missing alarm rates and detection delays are introduced. Three data sets are presented that will form the basis for the application study: a simple nonlinear data set, the benchmark Tennessee Eastman problem, and a real-world mineral processing data set.

4.3.2 Fault diagnosis application

The results of the application of PCA and random forest fault diagnosis to three case studies are presented in Chapter 7. For the benchmark Tennessee Eastman problem, performance indicators for other nonlinear feature extraction techniques (kernel PCA and kernel independent component analysis) are included for comparison. The random forest fault diagnosis method is also evaluated in terms of ideal characteristics of fault diagnostic algorithms (Venkatasubramanian et al., 2003c).

4.3.3 Feature space investigation of fault diagnosis framework

Further investigation into the nature of feature space detections is continued in Chapter 8. Two-dimensional feature spaces for the Tennessee Eastman process and real-world mineral processing data are considered, in order to visualize the projections of known faults. Simulations are done to show the influence of correlation, sample size and dimensionality on two-dimensional random forest features, to mimic the results found in the previous case studies.

4.3.4 Fault identification with variable importance and partial dependence

Fault identification is further developed in Chapter 9, with the introduction of random forest variable importance measures. The unsupervised problem of fault detection can be transformed to a supervised problem once samples have been classified as faulty. Random forest variable importance, an interpretive tool for supervised applications, is then a viable option for fault identification. In Chapter 9, random forest variable importance measures are validated on simulated case studies and the Tennessee Eastman process. The random forest approach is also compared to other tree ensemble techniques (conditional inference forests and boosted trees).

As an extension to Chapter 9, the interpretation of the influence of identified important variables is considered in Chapter 10, as well as the definition of a simple threshold for variable importance measures. To this end, random forest partial dependence is introduced and validated on simulated data, and an information visualization method developed. Finally, this interpretive visualization method is applied to the Tennessee Eastman process and real-world mineral processing data set.

4.4 Change point detection

As a final application of random forest feature extraction to process monitoring, random forest mapping and demapping is applied to the problem of change point detection. Change point detection is similar to that of fault detection, with the exception that the so-called normal operating conditions data are continually updated. The focus of change point detection is on the early detection of changed process states, whether it be indicative of a possibly catastrophic change, changing feed properties, deteriorating equipment and or other dynamic events.

An algorithm to detect change in univariate or multivariate time series using random forest feature extraction is developed in Chapter 11, based on moving base and test matrices and the calculation of a distance diagnostic. This distance diagnostic reflects the reconstruction error on test data, where reconstruction is based on RF models trained on base matrix data. The RF change point detection algorithm is conceptually similar, excluding the nature of the feature extraction, to a previously developed algorithm; that of singular spectrum analysis change point detection. These two algorithms are compared on simple simulated time series, and more complex dynamic reaction time series. The sensitivity of the diagnostic to an important model parameter and to noise is also investigated.

4.5 Conclusions and recommendations

Finally, the main conclusions of the feature quality study, the fault detection studies and change detection studies are summarized, and a final conclusion made on the applicability of random forest feature extraction to process monitoring applications. Recommendations for relevant future research opportunities are also presented.

Nomenclature

a	reduced dimensionality / number of features retained
E	residual matrix
m	number of process variables
N	number of observations
T	feature matrix
X	process data
X'	reconstructed process data

CHAPTER 5 - QUALITY OF RANDOM FOREST FEATURES

Random forest feature extraction is compared to five other techniques on seven data sets, with performance measures incorporating local and global structure preservation. From this comparative study, random forest features are shown to be generally difficult to interpret in terms of geometry present in the original variable space. However, this lack of interpretation does not hinder the accurate reconstruction of the original variable space from random forest features using random forest regression, as applied to seen training data.

A further investigation on the effect of the RF parameters on the performance of feature extraction is done. The overall conclusion from the overview on parameter sensitivity is that performance measures are not very sensitive to the random forest parameters.

Finally, a simple data set is used to demonstrate random forest regression mapping and demapping performance. From this example, it is concluded that random forest mapping and demapping models are very accurate on training data, and extrapolate weakly to unseen data that do not fall within regions populated by training data.

5.1 Overview

In chemical engineering and other disciplines, data bases of high dimensionality are often collected or constructed. These data sets consist of many observations for a myriad of variables. The presence of duplicate, noise and correlated variables result in an artificially high-dimensional measurement space. However, most processes may be thought of as being driven by only a few underlying events. By removing dependencies among variables through decorrelation and other methods, the measurement space can be reduced to a lower dimensional subspace, retaining useful information on the process driving forces (MacGregor & Kourti, 1995).

The basic assumption of feature extraction in the guise of dimension reduction is that measured data is constrained to a lower-dimensional manifold in the measurement space (Carreira-Perpinan, 1997), where a manifold is the essential shape of the data cloud. By constructing a modified plane onto which to project higher-dimensional data, the data manifold can be unfolded for investigation, excluding “empty” spaces.

The aims of dimension reduction include compressing datasets for more efficient processing and possible supervised learning tasks, data visualization and determining underlying trends driving the process characterized by the data set (Cayton, 2005). By dividing the measurement space into an information-rich feature space and an information-poor residual space, parameters can be devised to indicate process drift, as applied in fault detection (MacGregor & Kourti, 1995).

To validate the applicability of random forest feature extraction to the task of process monitoring, a study was set up to evaluate random forest feature extraction in terms of certain performance criteria, as compared to other feature extraction methods on a selection of data sets.

Random forest feature extraction is to be compared to a selection of other feature extraction techniques, as applied to a variety of data sets. The comparative techniques are selected to represent a diversity of dimension reduction approaches, in terms of linear / nonlinear and global / local focus. The data sets are selected to represent a variety of real-world and simulated situations, with continuous manifold and discontinuous cluster

structures. The performance of the techniques is evaluated in terms of local and global preservation characteristics.

5.2 Feature extraction validation data sets

Seven data sets are used in the feature extraction validation study, including real-world process engineering data sets, a simulated process data set and simulated manifold data sets. A brief description of each data set is now given.

5.2.1 Copper flotation data set (datacop)

The copper flotation data sets consists of ten inputs extracted from digitized images of froths obtained from a copper flotation plant, with four different observed operating regimes function as class label. This data set contains 490 samples (Jemwa & Aldrich, 2006).

5.2.2 Platinum group metals flotation data set (datapgm)

The platinum group metals flotation data set consists of five inputs from digitized images of froths obtained from a PGM flotation plant, with three different observed operating regimes present and a sample size of 294 (Jemwa & Aldrich, 2006).

5.2.3 Simulated phase flow data set (data3p)

The simulated phase flow data set consists of modelled data representing non-intrusive measurements on a pipe-line that transports a mixture of oil, water and gas. Three flow regimes are simulated, namely horizontally stratified, nested annular or homogeneous flow. Twelve variables are measured and 1000 samples recorded, but for each flow regime there is only two degrees of freedom: the fractions of the oil and the water[8].

5.2.4 Hollow sphere and centre cluster data set (datasphere)

The datasphere data set consists of two clusters: 300 samples forming a uniform sphere as centre, and 700 samples forming a uniform shell outside the centre. The data set thus has 1000 samples in three dimensions. Figure 5.1 shows the shell and centre clusters of this data set.

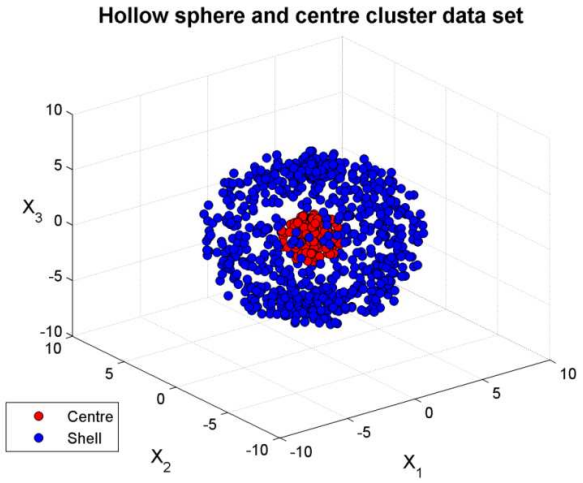

Figure 5.1: Hollow sphere and centre cluster data set with class distinction

[8] *The simulated phase flow data can be downloaded from http://www.ncrg.aston.ac.uk/GTM/3PhaseData.html.*

5.2.5 Swiss roll data set (dataswiss)

The Swiss roll data set consists of 2000 samples of a two-dimensional manifold embedded in three-dimensional space as a rolled-up structure. The class response divides the two-dimensional manifold into a checkerboard pattern. Figure 5.2 presents the manifold structure of this data set.

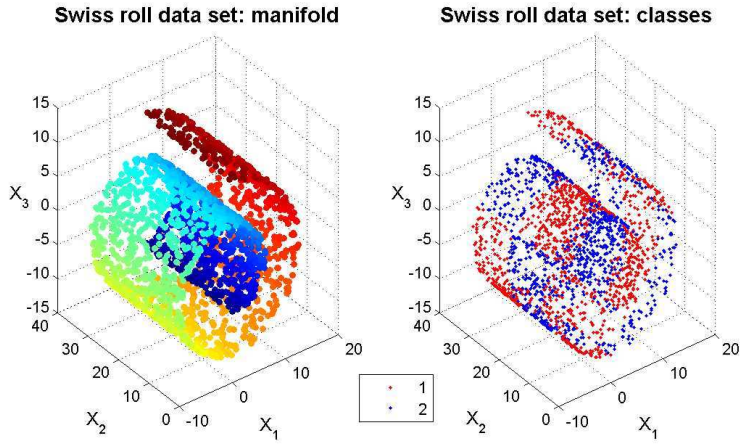

Figure 5.2: Swiss rall data set with manifold gradation and class distinction

5.2.6 Broken Swiss roll data set (databroken)

The broken Swiss roll data set is similar to the Swiss roll data set, but with a discontinuity in the two-dimensional manifold. Three inputs for 2000 samples make up the data set, with the class response consisting of nine divisions of the manifold. Figure 5.3 depicts the broken manifold structure of this data set.

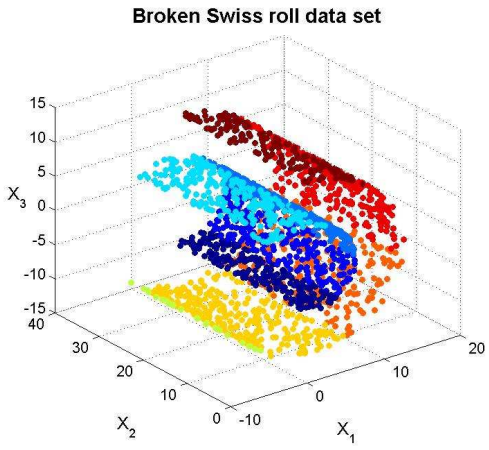

Figure 5.3: Broken Swiss roll data set with class distinction

5.2.7 Twin peaks data set (datatwin)

The twin peaks data set is a two-dimensional manifold embedded in three dimensions as a nonlinear surface. The class response for the three-input, 2000 sample data describes a checkerboard pattern on the two-dimensional manifold. Figure 5.4 shows the structure of this data set.

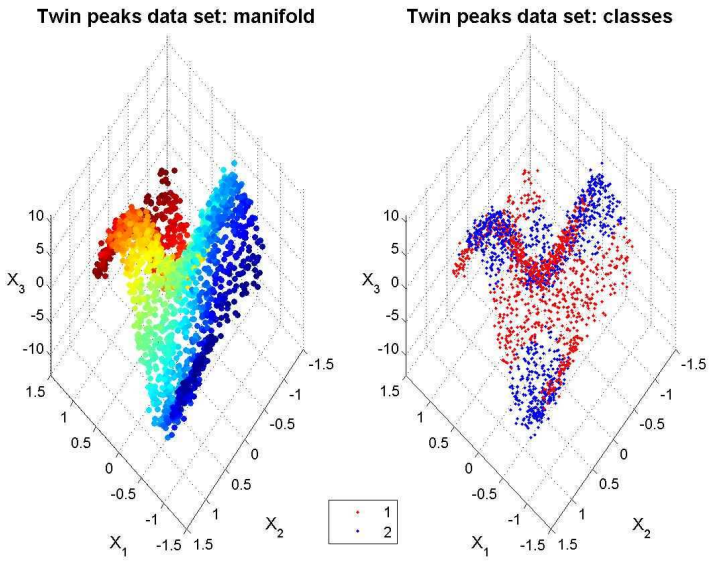

Figure 5.4: Twin peaks data with manifold gradation and class distinction

5.3 Feature extraction validation techniques

Various feature extraction algorithms exist, with principal component analysis arguably the simplest and most widely used method (Cayton, 2005). These methods can be categorized in several ways: linearity of projections, nature of criterion to be explicitly or implicitly optimized, local or global emphasis and more. In order to validate the use of random forest feature extraction, a diverse array of techniques were selected for comparison.

5.3.1 Principal component analysis

Principal component analysis (PCA), a linear technique, is the most widely used of all dimension reduction techniques (Cayton, 2005), operating by finding orthogonal vectors in the measurement space onto which data is projected. The criterion for determining these principal components (vectors) is to maximize variance of data projected onto said components (MacGregor & Kourti, 1995). Each principal component is linear combination of measured variables, uncorrelated to all other components. The first component accounts for the most variance, the second component for the next most variance, etc. The determination of these components is an optimality guaranteed problem in the form of spectral decomposition of the covariance matrix of the data (Carreira-Perpinan, 1997). The first a (a being intrinsic dimension) principal components can be thought of as a hyperplane onto which data are projected to capture the most variance present. Advantages of PCA include its general computational efficiency, guaranteed optimality and interpretability (Tenenbaum et al., 2000), while a main disadvantage is its limitation to linear subspaces (Carreira-Perpinan, 1997; Cayton, 2005).

The PCA algorithm is expounded on here, as PCA will also be used in fault diagnosis algorithms, and is the de facto leading linear feature extraction method applied to process monitoring.

Principal component analysis algorithm

- For data set **X** (N observations by m variables), construct the covariance matrix $\mathbf{\Sigma}$
  - $\mathbf{\Sigma} = \frac{1}{N-1}\mathbf{X}^{T}\mathbf{X}$ ***from Eqn. 2***
- Calculate the eigenvectors **V** and eigenvalues Λ for the covariance matrix $\mathbf{\Sigma}$ using eigenvalue decomposition
  - $\mathbf{\Sigma} = \mathbf{V}\Lambda\mathbf{V}^{T}$ ***from Eqn. 3***
- Determine the reduced dimensionality a which captures significant variance
- Calculate principal component scores using principal components **P** (a columns of eigenvector matrix **V**)
  - $\mathbf{T} = \mathbf{XP}$ ***from Eqn. 6***

The Matlab™ Statistics Toolbox function **`princomp`** is used to compute principal components, scores and variances.

5.3.2 Kernel principal component analysis

Kernel principal component analysis (KPCA) extends the concept of PCA to the maximization of projected variance on a nonlinear manifold. The measurement space is enlarged to higher dimensions by the calculation of kernel functions (e.g. Gaussian or polynomial kernels) of data point pairs. The kernel matrix can then be spectrally decomposed, resulting eigenvectors used to calculate eigenvectors of the nonlinear covariance matrix (Van der Maaten et al., 2009). A low-dimensional manifold is not explicitly constructed, but rather linear correlations are searched for in an increased dimensional space (Chigirev & Bialek, 2004). Although this technique accounts for nonlinear manifolds, the size of the kernel matrix (Van der Maaten et al., 2009) and difficulty of interpretation (Chigirev & Bialek, 2004) are disadvantages.

More details on KPCA are given in Chapter 2. KPCA features are calculated using Van der Maaten's Matlab™ Dimension Reduction Toolbox (Van der Maaten et al., 2009). A Gaussian kernel is utilized, with the Gaussian kernel parameter calculated as the average of all interpoint distances between 10 % of samples.

5.3.3 Sammon mapping

Multidimensional scaling (MDS) is a method of translating proximities between data points into Euclidian distances in a low-dimensional space, often with the purpose of visualization (Fodor, 2002). By equating the proximities between points as Euclidian distances, MDS can be used as a dimension reduction technique (Van der Maaten et al., 2009). MDS aims to minimize a cost function of differences between squared pairwise distances in the higher dimensional measurement space and lower dimensional feature space (Fodor, 2002). By modifying the cost function to normalize distances, i.e. place greater emphasis on proximal points, another variant of MDS is obtained: Sammon mapping (Sammon, 1969). The original MDS method (classic multidimensional scaling) is considered to be linear (Lee et al., 2004; Geng et al., 2005), while Sammon mapping is a nonlinear technique.

The Sammon cost function emphasizes small interpoint distances:

$$\phi(\mathbf{T}) = \frac{1}{\sum_{ij} d_{ij}} \sum_{i \neq j} \frac{d_{ij} - \|\mathbf{t}_i - \mathbf{t}_j\|}{d_{ij}} \qquad \textit{Eqn. 16}$$

With $\phi(\mathbf{T})$ the Sammon cost function of the low-dimensional mapping $\mathbf{T}$; d_{ij} the Euclidian distance between points i and j in the original input space; and $\mathbf{t}_i$ the i^{th} sample score vector.

A drawback of MDS lies in the fact that only global interpoint distances are considered, and not the distribution of nearest neighbours of data points. This can lead to insensitivity to the underlying data manifold (Van der Maaten et al., 2009).

The Sammon mapping dimension reduction function of Van der Maaten's Matlab™ Dimension Reduction Toolbox is used in this study (Van der Maaten et al., 2009). The default parameters of the DR Toolbox are used (maximum of 500 iterations, and a tolerance function limit of 1×10^{-9}).

5.3.4 Isometric feature mapping

Isometric feature mapping, or Isomap, (Tenenbaum et al., 2000) is one of the better known and most applied of the nonlinear manifold learning methods (Van der Maaten et al., 2009). Isomap aims to characterize the manifold by determining the geodesic between points. Geodesic distances are distances measured along a manifold. The term isometric indicates that interpoint distances are preserved during embedding, subject to constraints (Cayton, 2005). The Euclidian distances between all data point pairs are calculated, and a specified number of nearest neighbours for each point defined according to these distances. A neighbourhood graph can then be constructed for geodesic distance estimation. The geodesic distances between neighbours are equated to their Euclidian distances, while non-neighbouring geodesic distances are the summations of Euclidian distances along the shortest path between points, "hopping" along neighbours. By applying MDS to the obtained geodesic matrix, a lower-dimensional feature space can be estimated (Tenenbaum et al., 2000).

An advantage of the Isomap technique is the guarantee of optimality for certain types of data distributions (Tenenbaum et al., 2000), while disadvantages include sensitivity to noise (Geng et al., 2005) and inability to accommodate discontinuous manifolds (Van der Maaten et al., 2009). This property may result in some data points simply left out of the low-dimensional projection, as said points were not considered to form part of the Isomap data manifold (Van der Maaten et al., 2009).

The details of Isomap are given below to illustrate the typical steps in this class of manifold-unfolding algorithms (Tenenbaum et al., 2000):

Isometric feature mapping algorithm

- Calculate all Euclidian pairwise distances ($d_{\mathbf{X}}$) between data points in the high-dimensional space.
- Determine the specified number of nearest neighbours of each data point.
- Construct neighbourhood graph, with data points connected to their nearest neighbours by distances equal to d_X.
- Determine geodesic distances (d_G) between all points from the neighbourhood graph, by "hopping along" nearest neighbours between far-off points.
- Use geodesic distances d_G as input to multidimensional scaling.

Again, Van der Maaten's Matlab™ Dimension Reduction Toolbox is used to implement Isomap feature extraction (Van der Maaten et al., 2009). The optimum number of nearest neighbours is estimated by calculating the 1-nearest neighbour generalization error for nearest neighbour values ranging from 5 to 15, and choosing the parameter that results in the lowest generalization error. The out-of-sample estimation function

of the Dimension Reduction Toolbox is used to determine the feature coordinates for data points that are excluded from the Isomap algorithm, due to points not deemed as connected to the main manifold.

5.3.5 Locally linear embedding

Locally linear embedding (Roweis & Saul, 2000), or LLE, is a manifold learning technique which incorporates the assumption of local linear regions in the manifold by constructing overlapping linear neighbourhood hyperplanes around all points (Cayton, 2005). These local hyperplanes are summarized as weight matrices of contributions of nearest-neighbours to each point. In the process of lower-dimensional embedding, the main constraint is in maintaining the weight matrices (Roweis & Saul, 2000). This is an example of conformal embedding, which aims to preserve local angles between data points, as opposed to local interpoint distances in isometric embedding (Roweis & Saul, 2000). Although LLE is less sensitive to short-circuiting than Isomap, disadvantages of this method include inability to successfully embed manifolds with holes, as well as a tendency to collapse a large group of points to a single projection (Van der Maaten et al., 2009).

Van der Maaten's Matlab™ Dimension Reduction Toolbox provides LLE feature extraction functionality (Van der Maaten et al., 2009). Optimal nearest neighbour determination and out-of-sample extension for LLE are identical to the Isomap approach mentioned before.

5.3.6 Random forest feature extraction

RF feature extraction can be considered a mixed global and local nonlinear method. The local nature of this method arises from the subspace characterization of data points by leaf nodes. By calculating the proximity of data points, the global structure of the data is characterized, and further globally defined by multidimensional scaling.

The unsupervised random forest feature extraction discussed in Chapter 3 is repeated here:

Random forest feature extraction algorithm

- For an unlabeled learning set **X**, create a synthetic data set $\mathbf{X}_0$ by random sampling from the product of marginal distributions of **X**
- Label the response of **X** inputs as class 1 and the response of $\mathbf{X}_0$ inputs as class 2
- Concatenate input matrices **X** and $\mathbf{X}_0$ as **Z**, with concatenated response Y
- Construct a random forest classification model to predict Y given **Z**
- Create an empty similarity (proximity) matrix **S**
- For each tree k = 1 to K
    - For each sample combination of **X** (i,j), determine the terminal nodes to which they report
    - If a sample combination of **X** (i,j) report to the same terminal node, increase $S_{i,j}$ by one
    - Repeat for all possible sample combinations
- Scale the similarity matrix **S** by dividing by the number of trees K; the similarity matrix is symmetric and positive definite, with entries ranging from 0 to 1, and diagonal elements equal to 1
- Determine the dissimilarity matrix $\mathbf{D} = 1 - \mathbf{S}$
- Use the dissimilarity matrix **D** as input to classical multidimensional scaling, retaining a scaling coordinates **T** as random forest features

In terms of implementation, Andy Liaw's **randomForest** package in the statistical programming language R has an unsupervised mode operation which allows calculation of a proximity matrix (Liaw & Wiener, 2002). The Matlab™ Statistics Toolbox function **cmdscale** is used to compute multidimensional scaling coordinates and corresponding eigenvalues from the random forest proximities.

5.4 Determining intrinsic dimensionality

The intrinsic dimensionality of a data set is the minimum number of parameters required to identify the essential structure of the data, translating into the dimensionality of the manifold (Van der Maaten et al., 2009). For example, a straight line, in how ever many dimensions, has an intrinsic dimension of one.

5.4.1 Data visualization approach

If the purpose of feature extraction is to obtain an informative visualization of the data, a maximum of three features must be extracted. However, data visualization may be hampered in three dimensions due to obstruction of points. In this study, two features are extracted for the purpose of data visualization.

5.4.2 Maximum likelihood intrinsic dimensionality estimator

There exists a correlation between the number of data points in a hypersphere of a certain radius r and r^a, where a is the intrinsic dimension of a data structure (Van der Maaten et al., 2009). By modelling the number of datapoints in a growing hypersphere as a Poisson process (Levina & Bickel, 2004), the intrinsic dimensionality can be estimated.

Where the intrinsic dimensionality of a data set is not explicitly known, Van der Maaten's Matlab™ Dimension Reduction Toolbox implementation of maximum likelihood estimation is used to determine an approximate intrinsic dimensionality (Van der Maaten et al., 2009).

5.5 Feature quality performance measures

5.5.1 Local structure preservation measure

The evaluation of the success of a feature extraction algorithm is no simple task. Theoretical evaluation of techniques is possible, though complex for some techniques, and may not result in definitive conclusions (Cayton, 2005). Experimental evaluation of manifold learning techniques require knowledge of the inherent data structure, thus limiting application (Cayton, 2005), while visual inspection for useful information (Cayton, 2005) is highly subjective.

One approach is to determine how well features can represent the local structure present in the original high-dimensional input space. Van der Maaten et al. (2009) uses 1-nearest-neighbour classification error (κ) as a numeric evaluation criterion in their review of manifold learning methods, which also requires a priori knowledge in terms of class membership. The 1-nearest neighbour classification method classifies each data point to the same class as its nearest neighbour. If nearest neighbours in the original input space remain nearest neighbours in the reduced feature space, it serves as an indication that local structure has been preserved.

The Matlab™ Pattern Recognition Toolbox (Duin et al., 2007) is used to calculate the 1-nearest neighbour generalization error.

5.5.2 Global structure preservation measures

To evaluate how well global structure is preserved in reduced feature space, one can attempt to reconstruct the original input variables from features, or one can compare the sample distances in input and feature space.

- Input variable reconstruction

Using multiple linear regression and random forest regression, the original input variables can be modelled using the features. The goodness-of-fit of the predictions then give an indication as to the global structure information present in the reduced feature space. Pearson's linear correlation between the original and predicted variables is indicative of the fit of reconstruction. The linear reconstruction correlation λ_L (from multiple linear regression) and nonlinear reconstruction correlation λ_N (from random forests regression) are defined to characterize global structure preservation.

- Input and feature distance comparison

The correlation of sample Euclidian distances λ_d in input and feature space gives another indication as to the preservation of global structure. Again, Pearson's linear correlation coefficient can be used to determine correlation between feature space distances and input space distances.

5.6 Methodology of assessment of quality of features

The procedure followed in the assessment of quality of features study is summarized here:

Methodology of assessment of quality of features

- Scaling:
  - Scale an input data set to **X** by subtracting variable means and dividing by variable standard deviations
- Intrinsic dimensionality:
  - Determine the intrinsic dimension a of **X** for visualization, from prior knowledge or with maximum likelihood estimation
- Feature extraction:
  - For each feature extraction algorithm, calculate a features
- Local structure measure:
  - Calculate the 1-nearest neighbour generalization error κ in the feature space given post priori class information
- Global structure measure:
  - Input variable reconstruction
    - Use multiple linear regression to reconstruct scaled input variables from calculated features: $\mathbf{X'}_L$ linear reconstructed scaled input variables
    - Calculate λ_L the scaled inputs and scaled input predictions
  - Use random forest regression to reconstruct scaled input variables from calculated features: $\mathbf{X'}_{NL}$ nonlinear reconstructed scaled input variables
  - Calculate λ_N for the scaled inputs and scaled input predictions
- Comparison:
  - Use t-tests to compare performance measures of different techniques

5.7 Results of feature extraction comparisons

5.7.1 Feature plots

The results of the two-dimensional feature extraction component of this validation study are presented in the following figures. Here, the two-dimensional features for each data set as obtained by each feature extraction method are shown, including the local structure preservation measure, κ.[9]

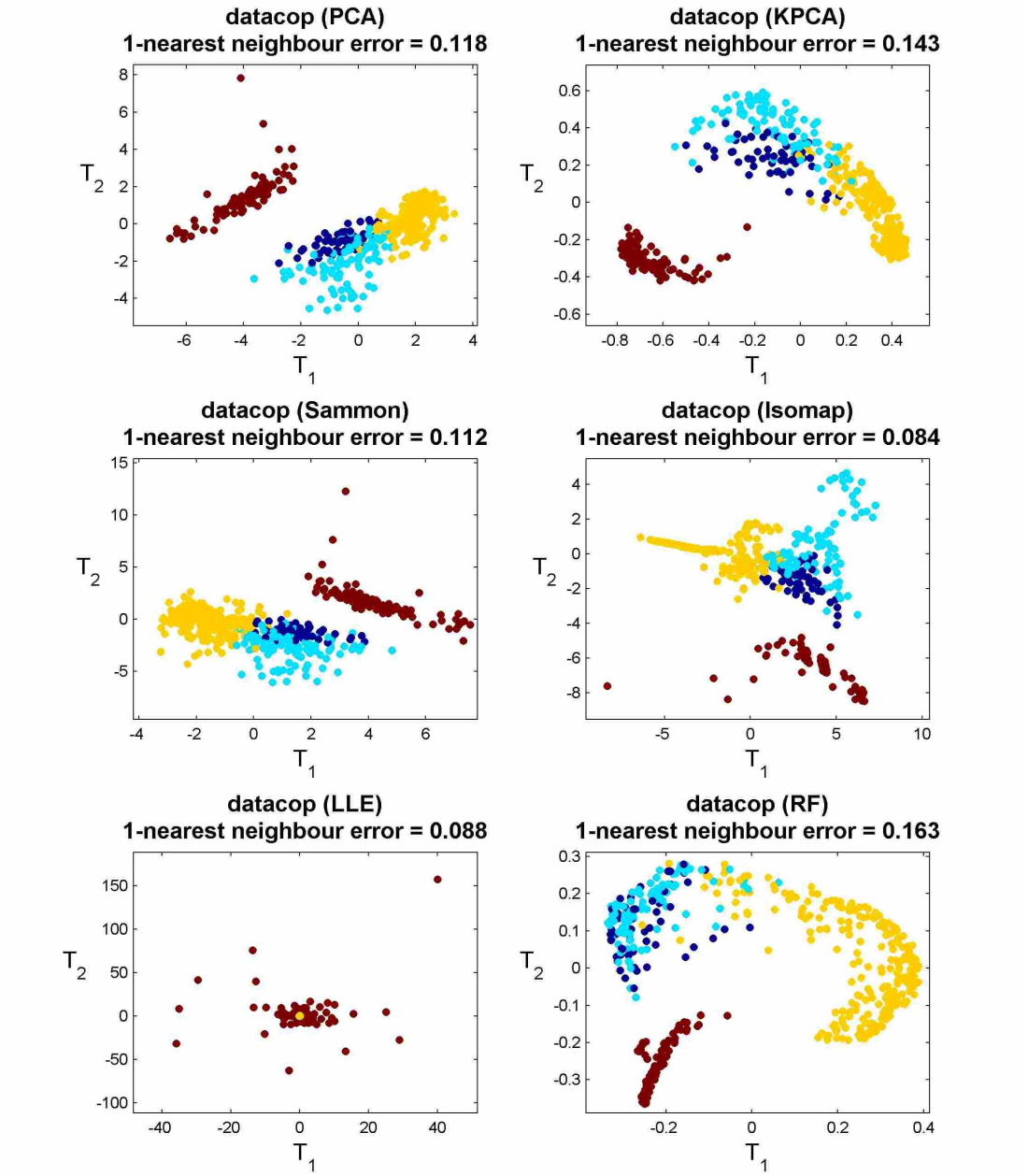

Figure 5.5: Two-dimensional features obtained for copper flotation data set, with class distinction and local structure preservation measures shown

[9] *Features were extracted in ten independent runs, in order to determine the stability of the different techniques. The feature plots shown here were constructed based on the features obtained from a single run, arbitrarily chosen as the fifth run of each technique.*

For the datacop data (Figure 5.5), the random forest features show reasonable separability, with class overlap for two classes indicated by cyan and dark blue. From the κ-values, Isomap shows best local structure preservation.

Isomap again shows the best local structure preservation for datapgm (Figure 5.6), with the random forest features again showing overlap for two classes.

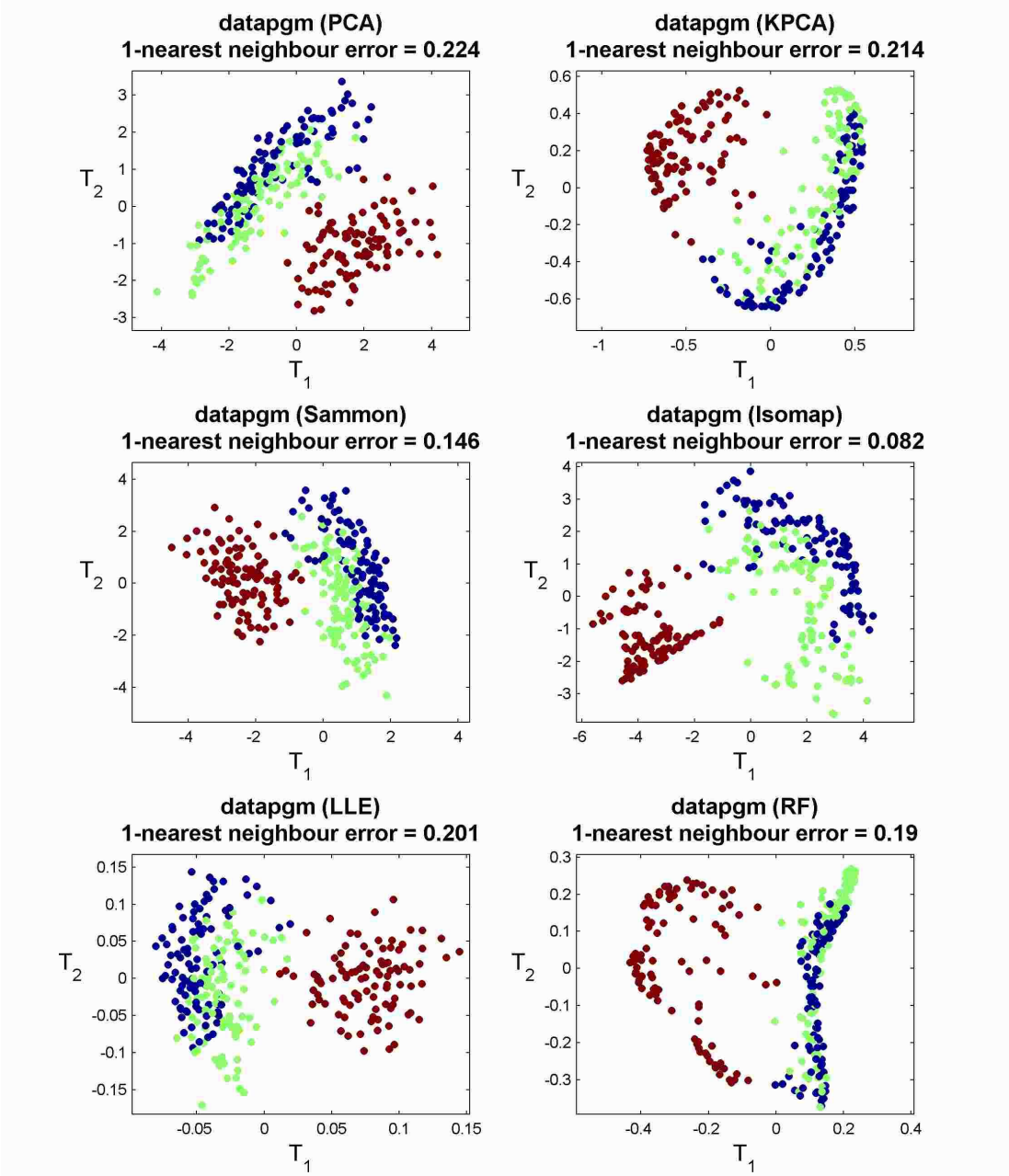

Figure 5.6: Two-dimensional features obtained for platinum flotation data set, with class distinction and local structure preservation measures shown

The Sammon mapping shows the best class distinction for data3p (Figure 5.7), closely followed by Isomap. Although there is overlap with the random forest features, a layering of classes can be observed.

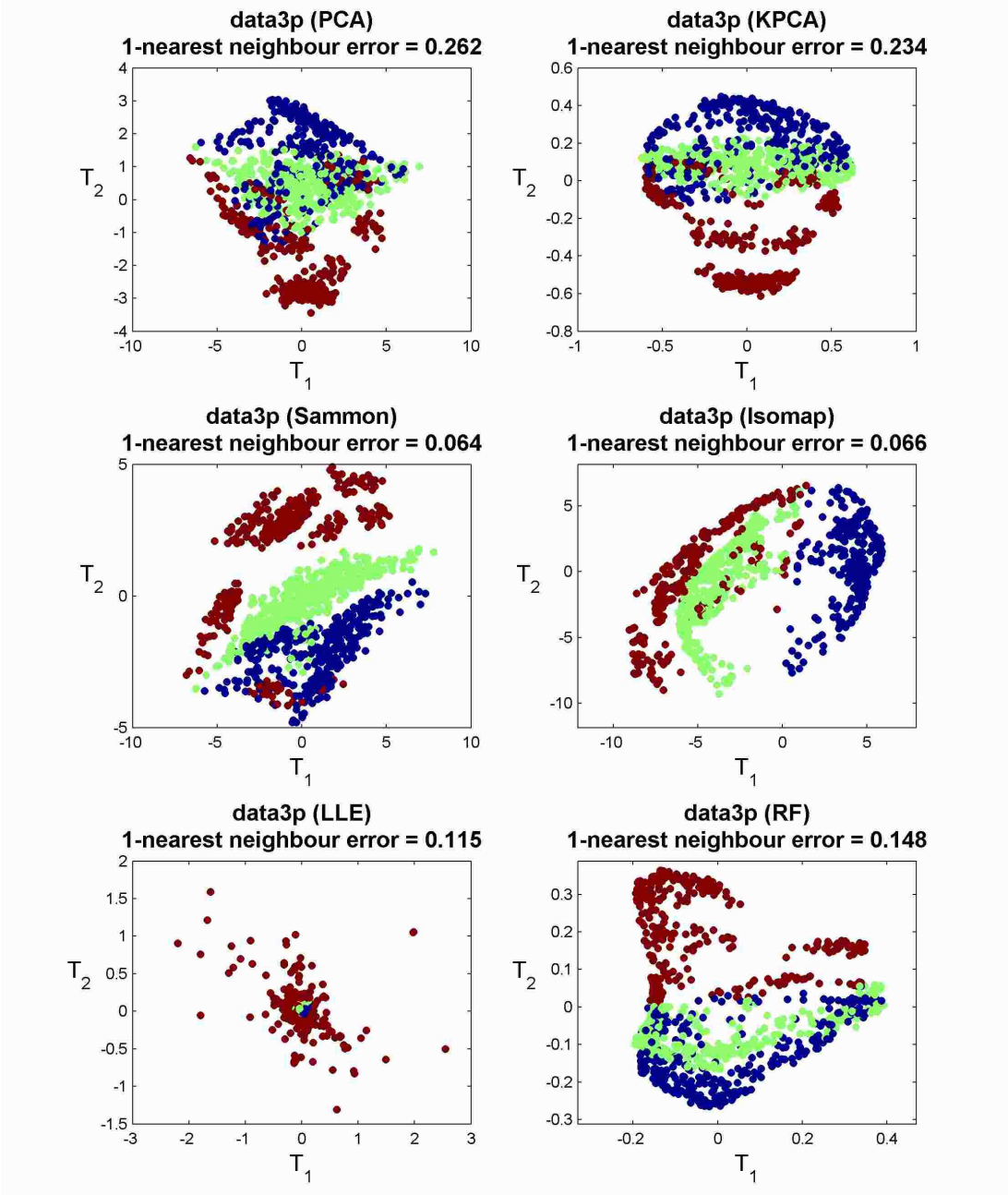

Figure 5.7: Two-dimensional features obtained for simulated multiphase flow data set, with class distinction and local structure preservation measures shown

For datasphere (Figure 5.8), Sammon mapping is able to completely separate the hollow outer shell and inner cluster. The random forest features are next best at local structure preservation (in terms of κ). It is interesting to note that the structure presented by the random forest features does not correspond to the geometry of the original hollow shell and inner cluster.

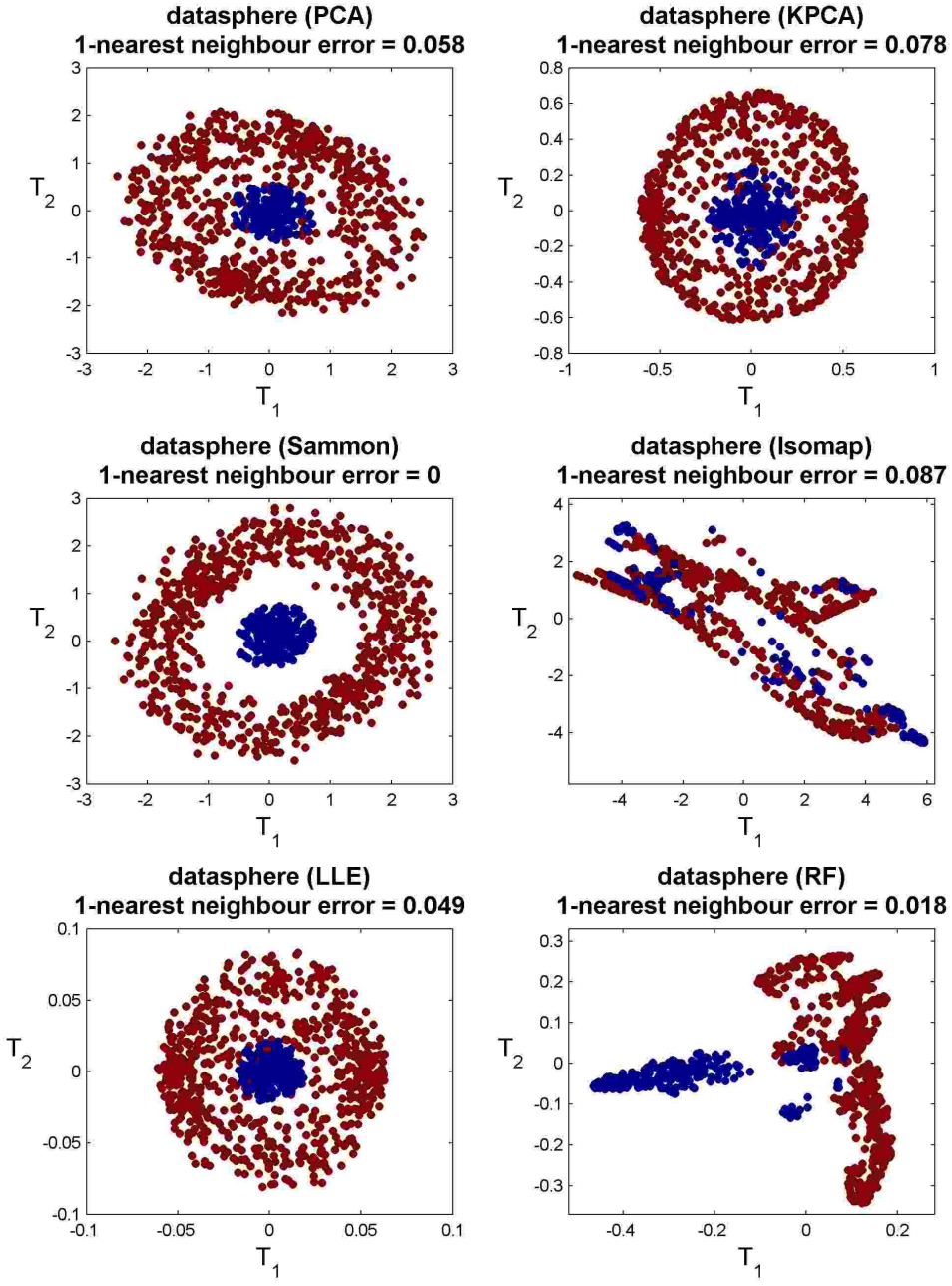

Figure 5.8: Two-dimensional features obtained for hollow sphere and centre data set, with class distinction and local structure preservation measures shown

With dataswiss (Figure 5.9), LLE and Isomap are able to "unfold" the two-dimensional manifold embedded in the three dimensions of the original data. The random forest features do not seem to relate to an unfolded manifold at all, and result in an even worse κ-value than the linear projection of PCA.

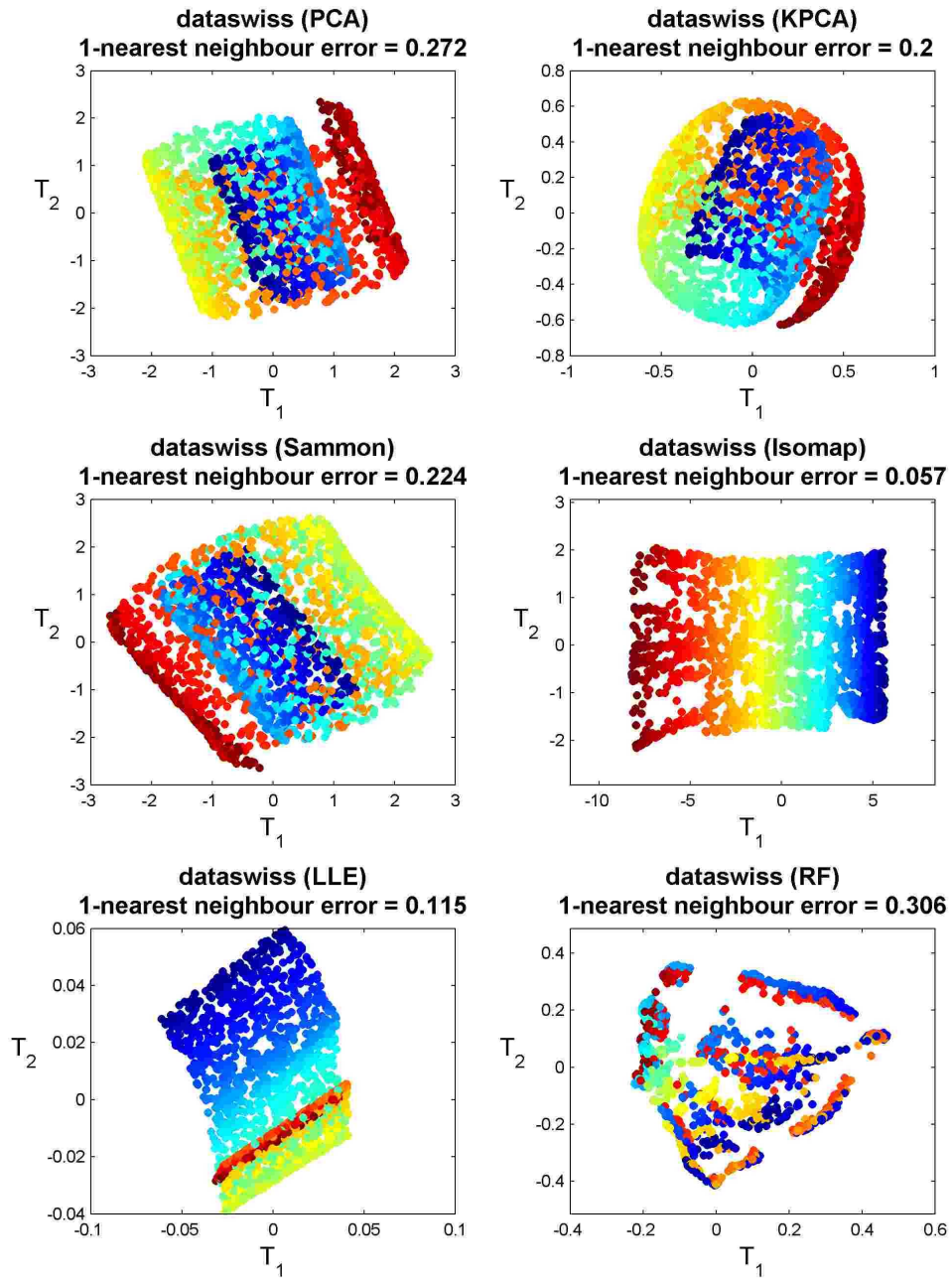

Figure 5.9: Two-dimensional features obtained for Swiss roll data set, with manifold colouration and local structure preservation measures shown

For databroken (Figure 5.10), Sammon mapping appears able to unfold both manifolds, albeit projecting these manifolds on top of each other. Isomap results in a lower κ-value, while LLE fails. Random forest features correspond to the second best κ-value, with features suggesting two overlapping manifolds, as with the Sammon features.

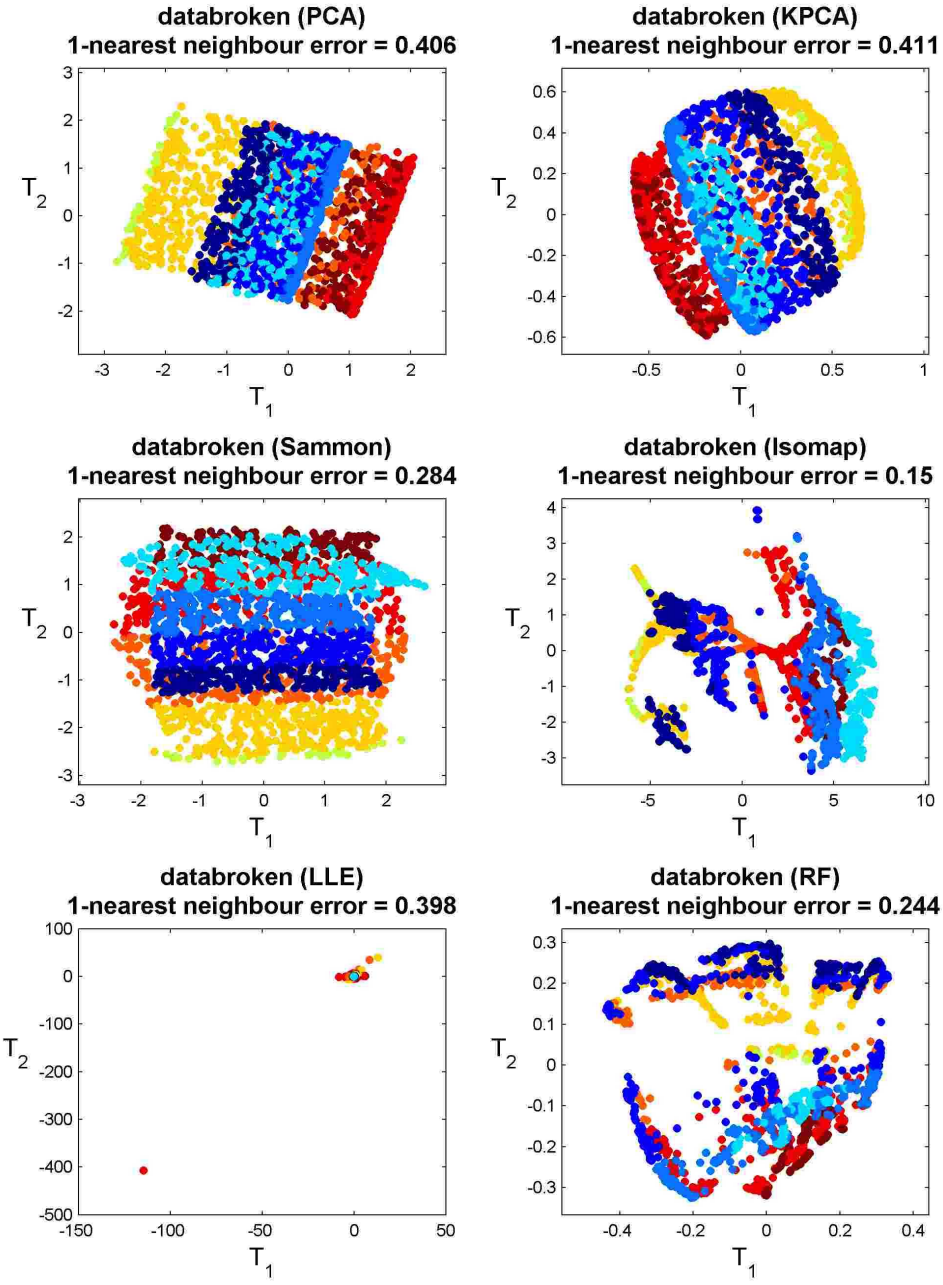

Figure 5.10: Two-dimensional features obtained for broken Swiss roll data set, with manifold colouration and local structure preservation measures shown

The datatwin data set (Figure 5.11) is "flattened" rather convincingly by Isomap. The random forest features seem to suggest three extremes in the data, which is not apparent from the original geometry.

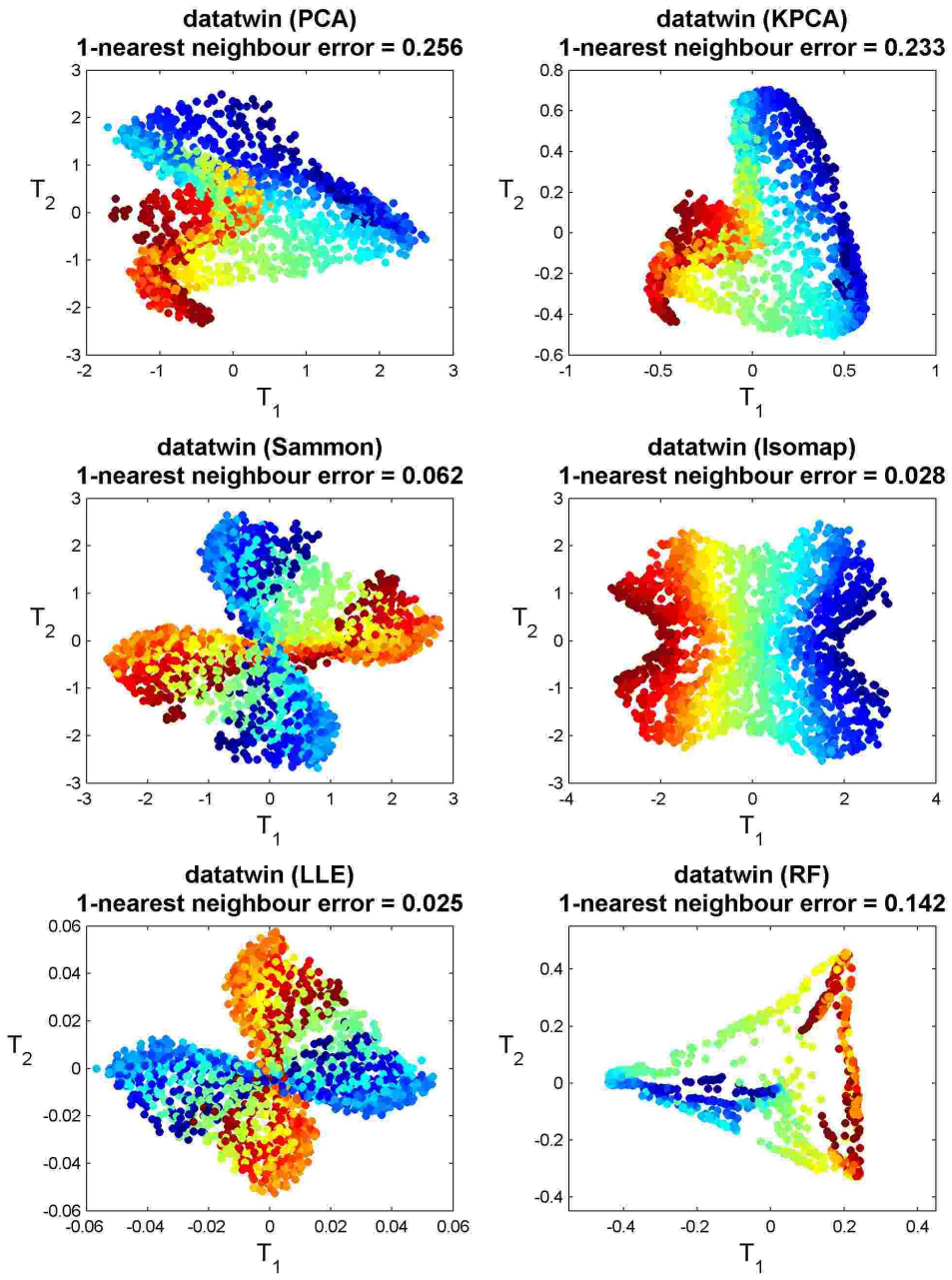

Figure 5.11: Two-dimensional features obtained for twin peaks data set, with manifold colouration and local structure preservation measures shown

5.7.2 Performance measures

The local structure preservation and global structure preservation performance measures are presented graphically for the different techniques, data sets and intrinsic dimensionalities. The results for the comparative two-sample t-tests are also presented in tabular form. One-sided t-tests were conducted to determine whether the mean of κ-values for random forest feature extraction is significantly smaller than that of a competing technique. If so, random forest feature extraction is designated better (b) than that technique. Similar one-sided t-tests were conducted to test the opposite, i.e. whether the random forest feature extraction mean is significantly larger than the competing technique. If so, random forest feature extraction is designated worse (w). If no significant difference is apparent random forest feature extraction is designated the same as (s) the competing technique. A significance level was chosen to correspond to 95 % confidence in the designations.[10]

The MLE-based intrinsic dimensions are 5, 4, 5 and 3 for datacop, datapgm, data3p and datasphere, respectively. Selecting an intrinsic dimensionality of three for the datasphere data is not especially informative, as the original variable space is three-dimensional.

- Local structure preservation

From Table 5.1, random forest feature extraction is shown to perform (based on local structure preservation) significantly better than PCA and KPCA for four data sets, and better at least once for one data set compared to Sammon mapping, Isomap and LLE.

Table 5.1: Result of t-tests comparison of random forest feature extraction to other techniques based on local structure preservation measure κ (95% confidence level; b = better, s = same as, w = worse)

a	κ	PCA	KPCA	Sammon	Isomap	LLE
2	datacop	w	w	w	w	w
	datapgm	s	s	w	w	w
	data3p	**b**	**b**	w	w	w
	datasphere	**b**	**b**	w	**b**	**b**
	dataswiss	w	w	w	w	w
	databroken	**b**	**b**	**b**	w	**b**
	datatwin	**b**	**b**	w	w	w
MLE	datacop	w	w	w	w	w
	datapgm	w	w	w	w	w
	data3p	w	w	w	w	w
	datasphere	w	w	w	**b**	w

- Linear reconstruction correlation

From Table 5.2, it is concluded that random forest feature extraction never performs (in terms of linear reconstruction correlation) better than PCA or KPCA, only better once than Sammon mapping and better most of the time than LLE.

[10] *Additional figures to show the distribution of the performance measures are given in Appendix A*

Table 5.2: Result of t-tests comparison of random forest feature extraction to other techniques based on global structure preservation measure λ_L (95% confidence level; b = better, s = same as, w = worse)

a	λ_L	PCA	KPCA	Sammon	Isomap	LLE
2	datacop	w	w	w	w	**b**
	datapgm	w	w	w	w	w
	data3p	w	w	w	w	**b**
	datasphere	w	w	w	w	w
	dataswiss	w	w	w	**b**	**b**
	databroken	w	w	w	s	**b**
	datatwin	w	w	**b**	**b**	**b**
MLE	datacop	w	w	w	w	**b**
	datapgm	w	w	w	w	w
	data3p	w	w	w	**b**	**b**
	datasphere	w	w	w	w	w

- Nonlinear reconstruction correlation

From Table 5.3, random forests outperform LLE for the majority of data sets, while performing better than PCA and KPCA (given two features) for a minimum of four data sets.

Table 5.3: Result of t-tests comparison of random forest feature extraction to other techniques based on global structure preservation measure λ_N (95% confidence level; b = better, s = same as, w = worse)

a	NCC	PCA	KPCA	Sammon	Isomap	LLE
2	datacop	w	w	w	w	**b**
	datapgm	w	w	w	w	w
	data3p	**b**	w	w	w	**b**
	datasphere	**b**	**b**	w	w	s
	dataswiss	**b**	**b**	**b**	w	**b**
	databroken	**b**	**b**	**b**	w	**b**
	datatwin	**b**	**b**	w	w	**b**
MLE	datacop	w	w	w	w	**b**
	datapgm	w	w	w	w	s
	data3p	w	w	w	w	**b**
	datasphere	w	w	w	w	w

Random forest feature extraction performs better on at least one data set compared to all techniques except Isomap, and for the majority of data sets as compared to PCA and LLE. This better performance with nonlinear reconstruction may indicate that linear regression is not able to reconstruct nonlinear features as obtained by random forests.

- Pairwise distance correlation

From Table 5.4, it is apparent that only LLE (and Isomap for one data set) is worse than random forest feature extraction in terms of pairwise distance correlation.

Table 5.4: Result of t-tests comparison of random forest feature extraction to other techniques based on global structure preservation measure λ_d (95% confidence level; b = better, s = same as, w = worse)

a	λ_d	PCA	KPCA	Sammon	Isomap	LLE
2	datacop	w	w	w	w	**b**
	datapgm	w	w	w	w	w
	data3p	w	w	w	w	**b**
	datasphere	w	w	w	w	w
	dataswiss	w	w	w	**b**	**b**
	databroken	w	w	w	w	**b**
	datatwin	w	w	w	w	w
MLE	datacop	w	w	w	w	**b**
	datapgm	w	w	w	w	s
	data3p	w	w	w	s	**b**
	datasphere	w	w	w	w	w

5.7.3 Discussion

Two-dimensional feature plots for the real-world data (datacop, datapgm and data3p) show that random forest feature extraction is fairly successful in class separation. For the synthetic data sets (datasphere, dataswiss, databroken and datatwin), the random forest features are not obviously related to the original structure of the manifolds in the original variable space. This may be an artefact of the distinct local nature of random forest models.

Neighbouring regions defined by leaf nodes are represented by a constant for prediction (be it class membership or regression response), with these local constants being independent of surrounding neighbourhoods. This disjoint, locally constant nature of random forest models may be responsible for the unexpected feature configurations observed in aforementioned feature plots.

In terms of local structure preservation, it was shown that random forest feature extraction performs significantly better than PCA and KPCA for four data sets, and better at least once for one data set compared to Sammon mapping, Isomap and LLE. Even though no absolute statement can be made in terms of random forest feature extraction being superior to other methods as measured by local structure preservation, it is significant that this technique does perform better on some data sets.

The failure of random forest features to enable successful reconstruction of the original variable space using linear regression underlines the fact that random forests are nonlinear methods. The success of nonlinear reconstruction of the original variables from random forest features suggests that, regardless of the seemingly strange structure of random forest features, these features capture sufficient information to enable successful reconstruction.

The inferior performance of random forest feature extraction in terms of pairwise distance correlation underlines the fact that global structure preservation is not an explicit criterion of random forest feature extraction. The fact that pairwise distances in feature space do not relate well to pairwise distances in the original variable space may be another result of the disjoint, local nature of random forest models.

An interesting observation from the performance measures results is that random forest feature extraction is more competitive for two-dimensional feature spaces than for higher dimensional feature spaces. This suggests that random forests may better capture structure in lower dimensions.

Three main conclusions arise from this feature extraction comparison study:

- Random forest features do not necessarily lend themselves to simple, geometric interpretation. This is a direct result from the disjoint, local nature of random forest models.
- Pairwise distances in the original variable space are neither explicitly, and from these experimental results, nor implicitly maintained in the random forest feature space.
- Even though no simple geometric interpretations are obvious from random forest features, sufficient information are contained in said features to enable successful reconstruction to the original feature space using nonlinear random forest regression, for training data.

Random forest feature extraction did not outperform all other feature extraction methods on all data sets. In the spirit of no free lunch theorems, this should not be expected. The old adage of horses for courses should be applied in feature extraction applications: certain techniques are suited better to certain types of data structures. For example, an explicit manifold unfolding method does better on the Swiss roll data set, where a linear method such as principal component analysis would fail. Here, "better" implies better unfolding of the manifold.

It can then be assumed that data structures exist for which random forest feature extraction will outperform other feature extraction techniques. The investigation of novel feature extraction techniques is then validated by the consideration that data may be structured in an infinite number of configurations.

A separate study has been conducted to investigate the exploitation of tree structure information when extracting features with random forests. The main conclusion of this study was that geometrically more representative features can be obtained with the proposed tree structure based method. However, the proposed method has very high computational expenses due to the required traversal of trees, as well as multiple sample-pair evaluations. This makes it currently unsuitable for application in fault diagnostic schemes. The details and results of this study are presented in Appendix B as an example of future possibilities of random forest feature extraction.

5.8 Feature extraction performance sensitivity

As an extension of the previous study, an investigation was done on the sensitivity of the local structure preservation (κ), global structure preservation (λ_L, λ_N and λ_d) and ensemble performance measures to changes in the random forest parameters, K and M (number of trees and random split variables). Results are summarized here, with detailed results in the Appendices.

The influence of changes in K on performance measures was investigated for forest sizes from 100 to 1000 trees, on all seven data sets and an intrinsic dimensionality of two. The influence of changes in M was restricted to two data sets (data3p and datacop) as all other data sets are of low dimensionality. The dimensionality of the data sets (ten and twelve, respectively) determine the range in which M was changed. Again, two features were extracted.[11]

[11] *These parameter changes refer to the number of trees and random split variables of the unsupervised feature extraction random forest, and not to the parameters of the regression forests used in calculating nonlinear reconstruction correlation.*

The overall conclusion from the overview on parameter sensitivity is that performance measures are not very sensitive to the random forest parameters. Where trends are present, there is scant evidence that an increase in the number of trees reduces the 1NN error as measure of local structure preservation. Another possible trend may be an improvement in global structure preservation (as measured by nonlinear reconstruction correlation) with an increase in number of trees. Finally, an increase in the number of random split variables increase the strength-correlation ratio of the forest, probably due to the increased correlations between trees.

From the above, the parameter selection for random forest feature extraction remains as suggested: M equal to the square root of m, and 1000 trees per forest.

5.9 Mapping and demapping with random forest regression

As random forest feature extraction is proposed to be used where offline training and online application are required, the explicit mapping and demapping functions are now considered. As mentioned before, the random forest feature extraction algorithm as developed by Shi and Horvath (2006) does not provide explicit mapping or demapping functionality for unseen data. For the purpose of this study, a random forest regression models are trained to map the original variables **X** to the feature space **T**, and m random forest regression models are trained to demap the features **T** to reconstruct the original variables **X**.

The nature of this regression mapping and demapping is the focus of this section. To investigate the mapping and demapping characteristics, a simple simulated data set is created, consisting of an outer ring and inner filled circle in two dimensions. (See Figure 5.13). Random forest feature extraction is applied to this data set to obtain two features. Random forest regression models are trained on these features for mapping and demapping purposes. The accuracy of the mapping is apparent from Figure 5.12. The data set is now expanded by adding uniform data between the outer ring and inner circle, as well as outside the ring. This new data set is mapped and demapped using the previously trained forests.

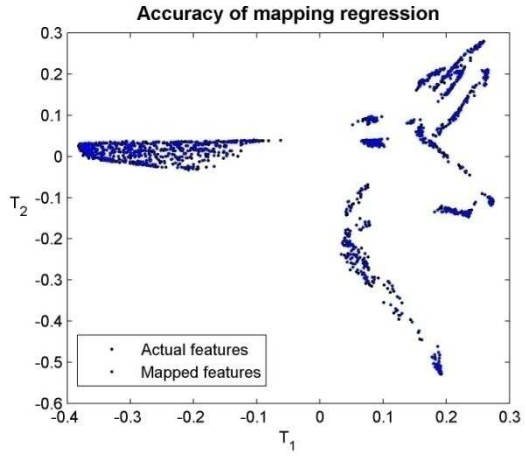

Figure 5.12: Random forest feature space showing actual and regression mapping features

The separation of the ring and circle in the training data is evident from the feature plot (κ of 0.001), with an excellent nonlinear reconstruction correlation (λ_N of 0.9815) achieved in demapping. This is apparent, as the reconstructed variable space closely resembles the original variable space. However, when unseen data are introduced, the local structure preservation deteriorates (κ of 0.2135 for unseen data). The green samples (between ring and circle) fall between the ring and circle samples in the feature space, but the black samples (outside ring) overlap with the ring samples in feature space. When using the trained demapping forests, the unseen data result in a lower nonlinear reconstruction correlation (λ_N of 0.7689). The green samples are

scattered nonuniformly on and within the ring in reconstructed variable space, while the black samples do not report outside the ring as in the original variable space.

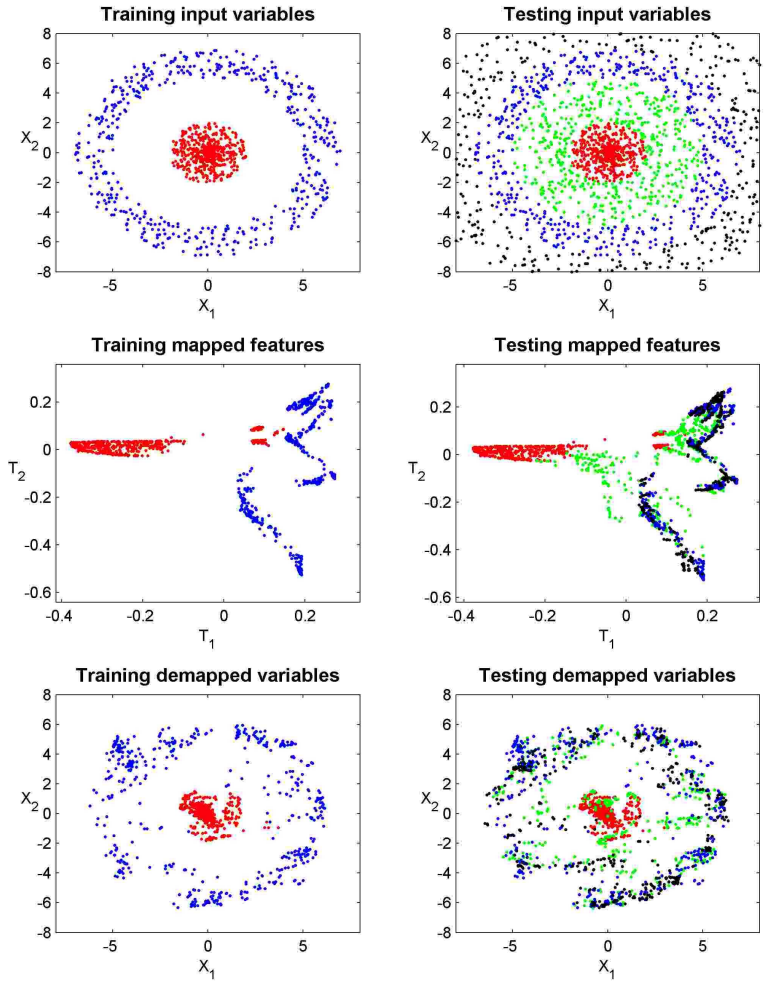

Figure 5.13: Random forest feature extraction demonstrating generalization characteristics

From this example, it is concluded that random forest mapping and demapping models are very accurate on training data, and extrapolate weakly to unseen data that do not fall within regions populated by training data. This is most likely a result of the disjoint, local nature of random forest models.

Nomenclature

a	reduced dimensionality / number of features
D	dissimilarity matrix
d	Euclidian distance
G	geodesic distance indicator
K	number of ensemble members
k	ensemble member
m	dimensionality of **X**
N	number of samples
P	principal components
r	hypersphere radius
S	similarity / proximity matrix
T	feature matrix
t	feature vector for one sample
V	eigenvector matrix
X	input data matrix / process data
Y	response vector
Z	concatenation of **X** and its permutation
κ	local structure preservation measure
$\mathbf{\Lambda}$	eigenvalues
Λ	global structure preservation measure
$\mathbf{\Sigma}$	covariance matrix
ϕ	Sammon cost function

CHAPTER 6 - FAULT DIAGNOSIS: FRAMEWORK

A fault diagnosis framework based on random forest feature extraction is developed in this chapter, analogous to principal component feature extraction. The developed framework for random forest feature extraction consists of an offline training algorithm on normal operating conditions data, and an online application algorithm to new data. Fault detection is based on comparing feature and residual space diagnostics to diagnostic thresholds. Fault identification is done through contribution plots in the residual space.

Design decisions include using random forest regression models to map unseen data to the feature space, and another set of random forest regression models to map features to the original variable space. Another design decision sees the introduction of one-class support vector machine probabilities as a diagnostic in the random forest feature space.

For application purposes, fault diagnosis performance measures are introduced in terms of false alarm rates, missing alarm rates and detection delays. Details of three data sets are given, viz. a simple nonlinear data set, the benchmark Tennessee Eastman process and a real-world calcium carbide process. Fault diagnosis with principal component analysis and random forest feature extraction will be applied to these data sets in the next chapter.

6.1 Overview

It was shown in Chapter 2 that feature extraction can be employed in fault diagnosis by segmenting a process data set into feature and residual spaces, and monitoring changes in these spaces with diagnostics. Chapter 5 compared the features obtained with random forests to other linear and nonlinear technique. Although random forests features were not distinctly superior to all techniques on all data sets (based on certain performance criteria), the information captured in the random forest features and their nonlinear nature make them viable as fault diagnosis fodder.

The aim of the fault detection and identification study is two-fold: to develop offline and online strategies employing random forest feature extraction, and to compare the performance of the developed strategies to existing techniques. The random forest approach will be compared to principal component analysis directly and indirectly to other feature extractive techniques on the basis of reported performance a benchmark process monitoring problem. The performance of the fault detection methods can be quantitatively assessed with false alarm rates, missing alarm rates and detection delays, while fault identification can be qualitatively assessed based on correct ranking.

A general outline of fault diagnosis with feature extraction is now presented, with the aim of designing a framework based on random forest feature extraction. Process fault detection and identification with unsupervised methods consists of offline and online routines. In the offline routine, normal operating conditions data are used to specify the process at hand. The online routine involves the testing of unseen data against the specified process features in order to detect whether a fault has occurred, and identifying process variables involved in a fault.

Fault detection and identification: general offline algorithm

For normal operating conditions data:

- Feature extraction:
  - Scaled original input variables **X** representing normal operating conditions are transformed to features **T**. The number of features to extract may need to be specified beforehand, or can be determined once all features are known.
- Feature characterization:
  - The score distance (s) is calculated to summarize the information in the feature space.
  - A detection threshold for the score distance (s_α) is determined using parametric or nonparametric methods. It is noted that this threshold is determined by some density estimation of the one-dimensional score distance sample.
  - Optionally, a nonparametric confidence region can be specified in the feature space. This corresponds to density estimation of the multidimensional feature space.
- Variable reconstruction:
  - Features are demapped to the original input space to obtain reconstructed variable values **X**'.
- Reconstruction characterization:
  - A statistic is calculated to summarize the information not captured in the feature space: the residual distance (r).
  - A detection threshold for the residual distance (r_α) is determined using parametric or nonparametric methods.
- Contribution calculations:
  - Feature space contributions (C_s) can be determined by decomposing the score distance into input variable contributions
  - Residual space contributions (C_r) can be determined by decomposing the residual distance into input variable contributions

Fault detection and identification: general online algorithm

For unseen data:

- Feature calculation:
  - The process inputs for unseen data are scaled using the scaling model for the normal operating conditions data. The feature scores for the unseen data are determined using the appropriate mapping models.
- Feature characterization:
  - Using the calculated scores, score distances can be calculated.
- Detection in feature space:
  - Score distances are compared to its detection threshold (s_α), and a detection is indicated if this statistic exceeds the detection threshold.
- Reconstruction calculation:
  - With the feature scores of the unseen data, reconstructed input variables can be obtained, and squared reconstruction errors (Q) calculated.
- Reconstruction characterization:
  - Reconstructed variable values are used to calculate the residual distance (r)

- Detection in residual space:
    - Residual distances are compared to its detection threshold (r_α), and a detection is indicated if this statistic exceeds the detection threshold.
- Contribution calculations:
    - Feature space and residual space contributions (C_s and C_r) are calculate as with the offline algorithm.
    - For detected faults, these contributions can be inspected to identify the fault. The upper limits of contributions for normal operating conditions may provide a useful comparison.

6.2 Design issues of random forest fault diagnosis

Fault diagnosis requires more than simply a feature extraction step, as shown in the general fault detection and identification algorithms. The design issues as applicable to creating a random forest feature extraction fault diagnostic method are presented here.

6.2.1 Reduced feature space dimension

Selecting the number of features that captures significant information is a challenge with no clear solution. One approach, the crossing criterion (Russell et al., 2000), makes use of the multidimensional scaling eigenvalues. An additional eigenvalue set is generated based not on the random forest proximities of the original data, but on random forest proximities of a randomly permuted version of the original data. By plotting both the structured and unstructured eigenvalues, the reduced feature space dimension can be approximated where the structured eigenvalues cross the unstructured eigenvalues.

6.2.2 Mapping and demapping functions

As mentioned in Chapter 3, random forest feature extraction only provides feature scores for training data, and cannot explicitly calculate scores for data which did not appear in the training set. This problem is also present in other nonlinear feature extraction techniques such as Isomap and Sammon mapping.

To calculate the scores and reconstructions of unseen data, explicit mapping and demapping models must be constructed. Multiple linear regression has the advantage that once the mapping model is defined, the demapping is automatically obtained as its inverse. However, linear regression may not be able to capture nonlinear relationships between the input variables and features. Random forest regression can model these nonlinear relationships, but are computationally expensive compared to multiple linear regression. Random forest regression is used for mapping and demapping in this study. Mapping requires as many random forests as there are feature variables, while demapping requires as many random forests as there are input variables. Classic multidimensional scaling features are independent of each other (Cox & Cox, 2001), which suggests that using one forest per feature should be sufficient.

6.2.3 Feature characterization

The computation of a modified Hotelling's T^2 statistic (as applied in traditional PCA fault diagnosis approaches) implies a uniform, hyperspherical shape to the normal operating conditions score distributions. This may not be a valid assumption for non-Gaussian data and scores (Martin & Morris, 1996). Another approach to characterizing the normal operating conditions region of feature space is to employ one-class support vector machines (Jemwa & Aldrich, 2006).

A one-class support vector machine (1SVM) finds a function which is positive in dense data regions, and negative where there is no data, by mapping the scores to a high-dimensional space and finding a hyperplane in this space that separates the data from the origin. Such a model requires a parameter *v* that controls the

trade-off between the complexity of the decision surface and the number of training data lying within the estimated region.

$$\min_{\mathbf{w}\in F,\rho\in\mathbb{R}^1,\boldsymbol{\xi}\in\mathbb{R}^m} \quad -v\rho+\frac{1}{m}\sum_{i=1}^{m}\xi_i \qquad \text{subject to} \begin{cases} \|\mathbf{w}\|_1=1 \\ (\mathbf{w}\cdot\varphi(\mathbf{x}))\geq\rho-\xi_i \\ \xi_i\geq 0,\ i=1,\ldots,m \end{cases} \qquad \textit{Eqn. 17}$$

With **w** defining a separating hyperplane in high-dimensional space $\mathbb{F}$; ρ some threshold or significance level; **ξ** the margin error; v a decision surface complexity parameter and $\varphi(\mathbf{x})$ a nonlinear mapping function on the vector **x**.

A simple grid search with cross-validation can be employed to automatically determine v (Hsu et al., 2003). The score distance (s) of a data point is then the negative one-class support vector machine probability of classification to the region defined by the training data. A detection threshold s_α for the one-class support vector machine score distances can be determined by a simple percentile approach, with a 1-α confidence.

The simple grid search approach involves five-fold cross-validation using normal operating conditions features, testing a range of v-values from 2^{-19} to 2^9. The cross-validation criterion is the average one-class support vector machine prediction accuracy of the left-out data for each fold. The maximum v-value that still ensures 70% average prediction accuracy is selected as model parameter. (The 70% criterion proved visually successful in accounting for irregular decision boundaries for various data sets, and reflects the irregularity of the decision boundary, and not the fault detection confidence limit). The percentile approach of defining s_α ensures a user-definable confidence limit.

The detection threshold for the one-class support vector machine score distances can again be determined by a simple percentile approach, as for the modified Hotelling's T^2 statistic. A drawback of the one-class support vector machine score distance approach is that decomposition for contribution purposes is no straight-forward task.

One-class support vector machines are employed to estimate score distances for random forest features, using the Matlab™ interface of the LIBSVM library (Chang & Lin, 2001). A radial basis function kernel is used, with v selected by cross-validation accuracy using a simple grid search approach (Hsu et al., 2003).

Figure 6.1 shows an example of a two-dimensional feature space with a non-Gaussian distribution. A 1SVM model was trained on NOC data in the feature space, using the above mentioned grid search approach. A grid of testing data was generated to show the range of 1SVM probabilities over the extent of the feature space. The negative 1SVM probabilities mimic the Hotelling's T^2 statistic in a PCA feature space, with increasing values as samples leave the support (distribution) of the original NOC training data. A threshold based on the 99% percentile is also shown in Figure 6.1.

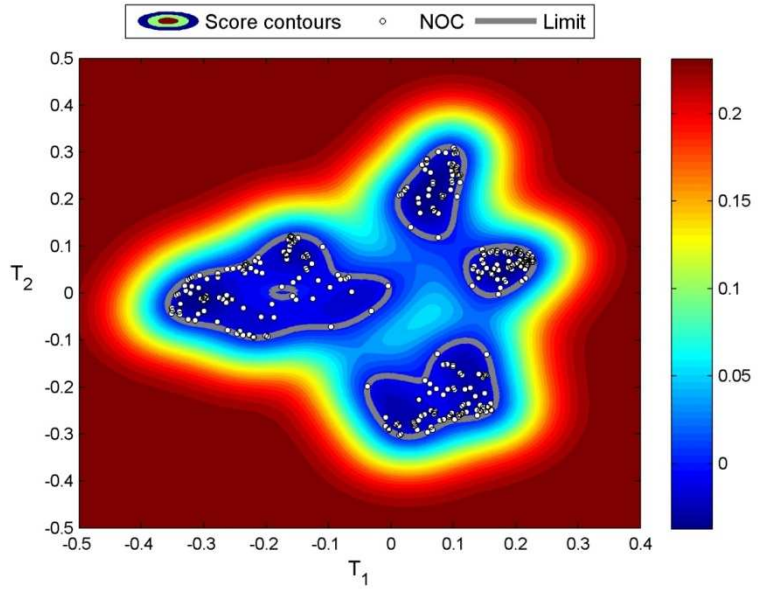

Figure 6.1: Example of one-class support vector machine probability surface for non-Gaussian distribution. Colouring indicates negative 1SVM probability.

6.2.4 Contribution calculations

Process variables related to a fault condition could be determined by investigating the difference between actual values and reconstruction values, as obtained by demapping forests. The residual based contribution of variable j ($C_{r,j}$) is calculated from the actual variable value X_j and the reconstructed variable value X'_j:

$$C_{r,j} = \left(X_j - X_j'\right)^2 \qquad \textit{Eqn. 18}$$

6.3 Fault detection and identification techniques

The proposed random forest technique is to be compared to principal component analysis fault detection and identification. Results on the Tennessee Eastman process for other nonlinear techniques, as discussed in Chapter 2, will also be provided. More details on these techniques can be found in literature.

6.3.1 Principal component analysis fault diagnosis

The feature extraction methodology of principal component analysis was discussed in the previous study. The application of principal component analysis to fault diagnosis is summarized here.

PCA fault diagnosis: offline training algorithm

For normal operating conditions data:

- Feature extraction:
  - For a scaled data set **X** of normal operating conditions (N observations by m variables), construct the covariance matrix $\boldsymbol{\Sigma}$
    - $\boldsymbol{\Sigma} = \frac{1}{N-1}\mathbf{X}^T\mathbf{X}$ ***from Eqn. 2***
  - Calculate the eigenvectors **V** and eigenvalues Λ for the covariance matrix $\boldsymbol{\Sigma}$ using eigenvalue decomposition
    - $\boldsymbol{\Sigma} = \mathbf{V}\Lambda\mathbf{V}^T$ ***from Eqn. 3***
  - Determine the reduced dimensionality a which captures significant variance; i.e. a is the

number of components accounting for 90 % of cumulative variance
    - Define the loading matrix (principal components) **P** as the first a eigenvectors of **V**
    - Calculate principal component scores
        - $\mathbf{T} = \mathbf{XP}$ ***from Eqn. 6***
- Feature characterization:
    - Calculate the score distance as the modified Hotelling's T^2 value:
        - $s = \mathbf{x}^{T} P \Sigma_a^{-2} P^{T} \mathbf{x}$ ***Eqn. 19***
    - Determine detection thresholds for the score distance:
        - $s_\alpha = \frac{a(N-1)}{N-a} F_\alpha(a, N-a)$ ***Eqn. 20***
- Variable reconstruction:
    - Calculate the reconstructed input variables **X'**
        - $\mathbf{X}' = \mathbf{TP}^{T}$ ***Eqn. 21***
    - Calculate the squared prediction errors Q
        - $Q_j = \left(X_j - X'_j\right)^2$ ***Eqn. 22***
    - Calculate the residual distance r
        - $r = \sum_{j=1}^{m} Q_j$ ***Eqn. 23***
    - Determine detection thresholds for the squared prediction errors:
        - $r_\alpha = \theta_1 \left[\frac{h_0 c_\alpha \sqrt{2\theta_2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2}\right]^{1/h_0}$ ***Eqn. 24***
        - With $\theta_i = \sum_{j=a+1}^{n} \sigma_j^{2i}$, $h_0 = \frac{2\theta_1\theta_3}{3\theta_2^2}$ and c_α is a normal deviate
- Contribution calculations:
    - Calculate the score distance contributions:
        - $C_{s,j} = \mathbf{T}\Lambda^{-1} P_j^{T} X_j$ for variable j ***Eqn. 25***
    - Calculate the residual distance contributions:
        - $C_{r,j} = Q_j$ for variable j ***Eqn. 26***

PCA fault diagnosis: online application algorithm

For unseen data:

- Feature calculation:
    - Scale new data using scaling model of normal operating conditions data
    - Calculate principal component scores
- Feature characterization:
    - Calculate score distances
- Detection in feature space:
    - Compare score distances to detection threshold s_α, indicate detection if value exceeds threshold
- Reconstruction calculation:
    - Calculate reconstructed variables
    - Calculate squared prediction errors
    - Calculate residual distances
- Detection in residual space:
    - Compare residual distances to detection threshold r_α, indicate detection if value exceeds threshold
- Contribution calculations:

- Calculate score distance contributions
  - For identified faults, compare with upper limits of score distance contributions of normal operating conditions
- Calculate residual distance contributions
  - For identified faults, compare with upper limits of residual distance contributions of normal operating conditions

6.3.2 Random forests fault diagnosis

The random forest fault diagnosis methodology is summarized here:

Random forest fault diagnosis: offline training algorithm

For normal operating conditions data:

- Feature extraction:
  - For a scaled data set **X** of normal operating conditions, calculate random forest features with random forest feature extraction algorithm
  - Determine the reduced dimensionality a which captures significant variance using the crossing criterion
  - Define the random forest feature scores **T** as a first score vectors of random forest features
  - Train mapping model with **X** as input and **T** as output
    - Random forest regression, a models
- Feature characterization:
  - Train a 1SVM model, calculate the probability values for all data points and define the score distance (s) as negative 1SVM probability
  - Determine a detection threshold for the score distances using the percentile approach
- Variable reconstruction:
  - Train demapping model with **T** as input and **X'** as output
    - Random forest regression, m models
  - Calculate the squared prediction errors Q
    - $\mathrm{Q_j} = \left(\mathrm{X_j} - \mathrm{X'_j}\right)^2$ ***from Eqn. 22***
  - Calculate the residual distance r
    - $\mathrm{r} = \sum_{\mathrm{j}=1}^{\mathrm{m}} \mathrm{Q_j}$ ***from Eqn. 23***
  - Determine detection threshold r_α for the residual distances using the percentile approach
- Contribution calculations:
  - Calculate the residual distance contributions:
    - $\mathrm{C_{r,j}} = \left(\mathrm{X_j} - \mathrm{X_j}'\right)^2$ ***from Eqn. 18***

Random forest fault diagnosis: online application algorithm

For unseen data:

- Feature calculation:
  - Scale new data using scaling model of normal operating conditions data
  - Calculate random forest feature scores
- Feature characterization:
  - Calculate score distances as negative 1SVM probabilities

- Detection in feature space:
  - Compare score distance to detection threshold r_α, indicate detection if value exceeds threshold
- Reconstruction calculation:
  - Calculate reconstructed variables
  - Calculate squared prediction error
- Detection in residual space:
  - Compare squared prediction error to detection threshold r_α, indicate detection if value exceeds threshold
- Contribution calculations:
  - Calculate squared prediction error contributions
    - For identified faults, compare with average squared prediction error contributions of normal operating conditions

A schematic comparison of the PCA and RF offline training algorithms is given in Figure 6.2.

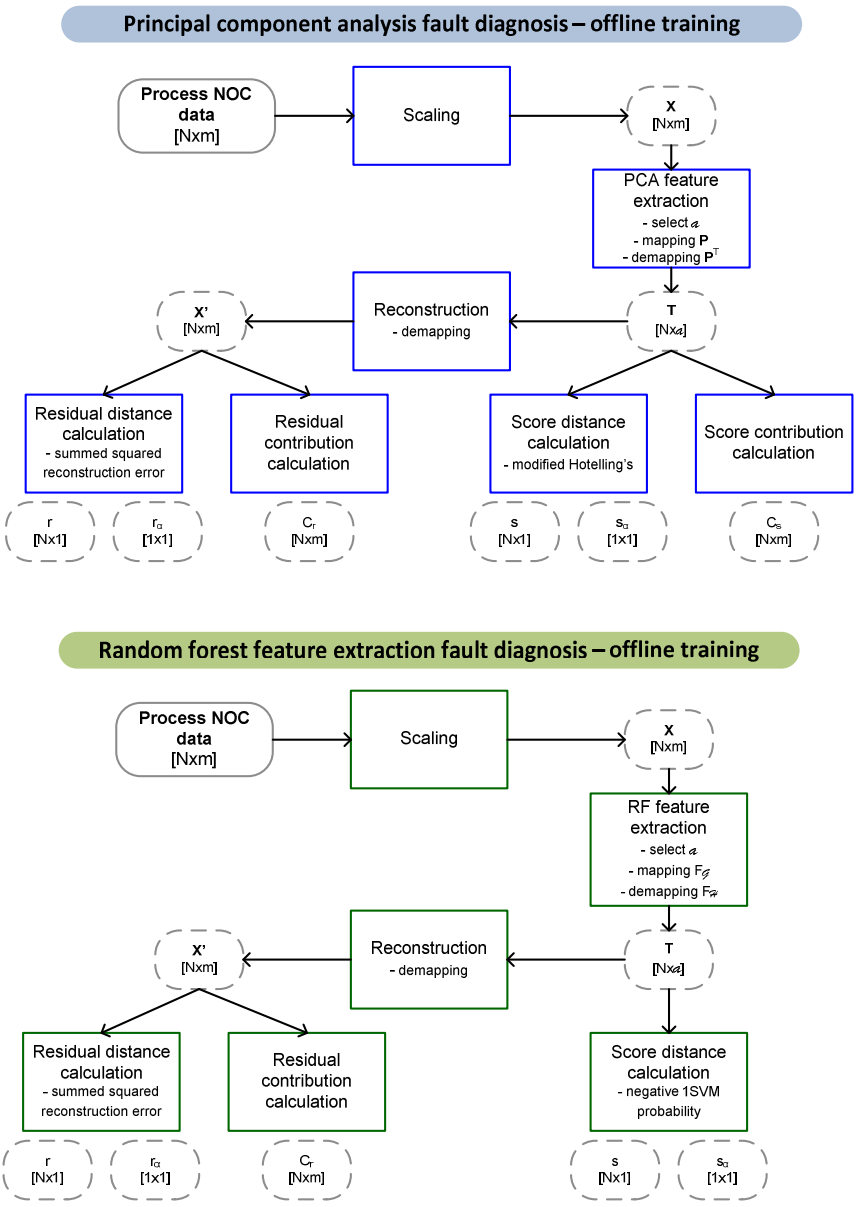

Figure 6.2: Schematic of PCA and RF fault diagnosis training algorithms

6.4 Performance measures

The performance of fault detection algorithms can be compared based on the number of faults that are correctly and incorrectly identified, as well as the delay in detecting a fault.

6.4.1 False alarm rate

The false alarm rate ζ is the number of detections that are logged for a validation normal operating conditions data set, i.e. a data set that was unseen during training, but is known *a priori* consists only of normal operating condition data. As thresholds are specified for the score and residual distances, the overall false alarm rate ζ is the maximum of the score distance false alarm rate ζ_s and the residual distance false alarm rate ζ_r.

6.4.2 Missing alarm rate

The missing alarm rate δ is the fraction of known fault samples that are not detected for a given data set. The overall missing alarm rate δ is the minimum of the score distance missing alarm rate δ_s and the residual distance missing alarm rate δ_r.

6.4.3 Detection delay

The detection delay γ is the number of consecutive faulty samples missed before a detection is logged, with the overall detection delay γ the minimum of the score distance detection delay γ_s and the residual distance detection delay γ_r. For detection delay calculation purposes, a detection is defined as three consecutive alarms.

6.5 Fault diagnosis methodology

For each of the three data sets described, PCA and RF fault diagnosis algorithms are trained offline on the relevant normal operating conditions data. If a validation normal operating conditions data set is available, the false alarm rates for the different algorithms are calculated. In order to ensure fair comparison, the detection thresholds for all algorithms are aligned to correspond to a 1% false alarm rate on the validation normal operating conditions data using the percentile approach for limits correction. The online implementations of the PCA and RF algorithms are then applied to the different faults. Missed alarm rates, detection delays and variable contributions can then be compared.

6.6 Fault detection and identification data sets

Three data sets are considered in this study: a simple nonlinear system, the benchmark process monitoring problem: the Tennessee Eastman process, as well as a real-world mineral process data set from a calcium carbide furnace operation.

- Simple nonlinear system

A simple simulated nonlinear system (Shao et al., 2009) consists of three variables and two degrees of freedom. Normal operating conditions are defined by the following set of equations:

Equation set: Simple nonlinear system

$$x_1 = t_1 + e_1$$
$$x_2 = t_1^3 - 4.5t_2^2 + 6t_1 + t_2 + e_2$$
$$x_3 = 3t_1^2 - t_2^3 + 3t_2^2 + e_3$$

Above, t_1 and t_2 are elements of [0.01,3] and e_1, e_2 and e_3 are independent Gaussian noise variables with mean of 0 and variance of 0.01. Two faults are introduced by changing certain parameters, with two hundred samples generated for each fault. (The normal and fault data are shown in Figure 6.3.)

- A linear increase of 0.1(k-50) is added to x_3 from sample 51 to sample 150, where k is the sample number. (Fault 1)
- The coefficient of the t_1^2 term in the expression for x_3 is linearly increased from 3 at sample 50 to 4 at sample 150. (Fault 2)

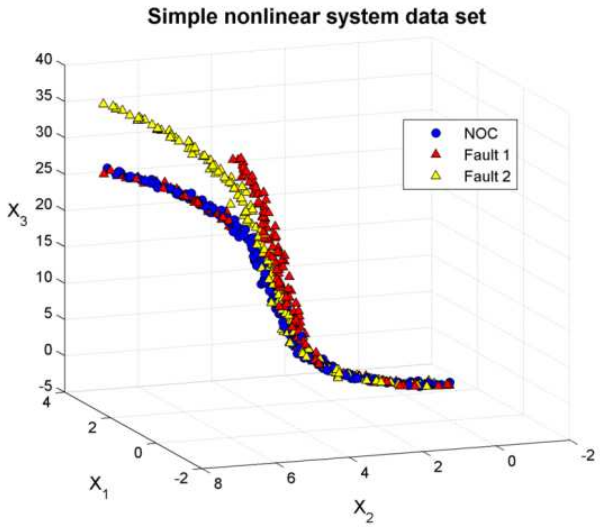

Figure 6.3: Normal operating condition and fault data for the simple nonlinear system data set

6.6.2 Tennessee Eastman process

The Tennessee Eastman process is a simulated chemical process developed to provide a realistic industrial process on which to evaluate the performance of process monitoring methods (Russell et al., 2000). The Eastman Chemical Company created a simulation of an actual chemical process with five major units and eight components (Downs & Vogel, 1993). Simulation data of the Tennessee Eastman process (with plant-wide control based on proportional and proportional-integral) control are available for normal operating conditions and 21 fault conditions (http://brahms.scs.uiuc.edu).

The flow sheet for the process is given in Figure 6.4. Gaseous reactants *A*, *C*, *D* and *E*, with inert *B*, are fed to a water-cooled reactor, where liquid products *G* and *H* and liquid byproduct *F* are formed.

Equation set: Tennessee Eastman process

$$A(g) + C(g) + D(g) \rightarrow G(l)$$
$$A(g) + C(g) + E(g) \rightarrow H(l)$$
$$A(g) + E(g) \rightarrow F(l)$$
$$3D(g) \rightarrow 2F(l)$$

The reactions are irreversible, exothermic and approximately first-order in terms of reactant concentrations, and have Arrhenius reaction rates in terms of temperature dependence, with the activation energy of *G* higher than that of *H*.

The reactor product is cooled in a condenser, and then separated in a vapour liquid separator. The vapour phase from the separator is recycled via a compressor to the reactor, with a portion purged to avoid accumulation of inert *B* and byproduct *F* in the system. The liquid from the separator is pumped to a stripper to remove remaining reactants in the stream for recycle. Liquid products *G* and *H* report to the product stream.

Process data consists of 41 measured variables (process and composition measurements) and 11 manipulated variables. Normal operating conditions are represented by 500 samples, with a further 960 samples available as validation data for normal operating conditions. Each of the 21 fault data set consists of 960 samples, with the fault occurring after 161 time steps within these 960 samples. The input variables and faults are summarized in Table 6.1 and Table 6.2, respectively.

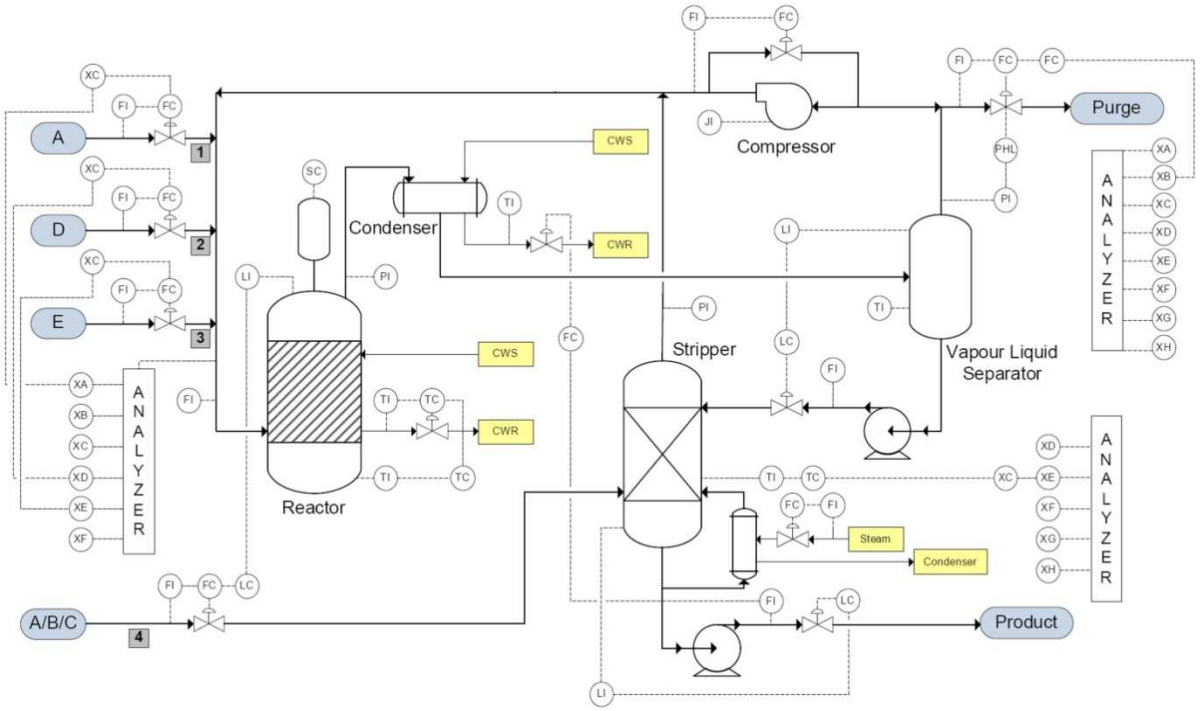

Figure 6.4: Process flow diagram for Tennessee Eastman process (Russell et al., 2000)

Table 6.1: Process faults of Tennessee Eastman process

Fault	Description	Type
1	*A*/*C* feed ratio (*B* composition constant) – Stream 4	Step change
2	*B* composition (*A*/*C* feed ratio constant) – Stream 4	Step change
3	*D* feed temperature – Stream 2	Step change
4	Reactor cooling water inlet temperature	Step change
5	Condenser cooling water inlet temperature – Stream 2	Step change
6	*A* feed loss – Stream 1	Step change
7	*C* header pressure loss (reduced availability) – Stream 4	Step change
8	*A*, *B*, *C* feed composition – Stream 4	Random variation
9	*D* feed temperature – Stream 2	Random variation
10	*C* feed temperature – Stream 4	Random variation
11	Reactor cooling water inlet temperature	Random variation
12	Condenser cooling water inlet temperature	Random variation
13	Reaction kinetics	Slow drift
14	Reactor cooling water valve	Sticking
15	Condenser cooling water valve	Sticking
16 - 20	Unknown	Unknown
21	Valve – Stream 4	Constant position

Table 6.2: Process variables of the Tennessee Eastman process (MV = manipulated variable, PM = process measurement, CM = composition measurement)

Variable	Description	Variable	Description
1	*A* feed – Stream 1 (PM)	***27***	Reactor feed component *E* (CM)
2	*D* feed – Stream 2 (PM)	***28***	Reactor feed component *F* (CM)
3	*E* feed – Stream 3 (PM)	***29***	Purge component *A* (CM)
4	Total feed – Stream 4 (PM)	***30***	Purge component *B* (CM)
5	Recycle flow (PM)	***31***	Purge component *C* (CM)
6	Reactor feed rate (PM)	***32***	Purge component *D* (CM)
7	Reactor pressure (PM)	***33***	Purge component *E* (CM)
8	Reactor level (PM)	***34***	Purge component *F* (CM)
9	Reactor temperature (PM)	***35***	Purge component *G* (CM)
10	Purge rate (PM)	***36***	Purge component *H* (CM)
11	Separator temperature (PM)	***37***	Product component *D* (CM)
12	Separator level (PM)	***38***	Product component *E* (CM)
13	Separator pressure (PM)	***39***	Product component *F* (CM)
14	Separator underflow (PM)	***40***	Product component *G* (CM)
15	Stripper level (PM)	***41***	Product component *H* (CM)
16	Stripper pressure (PM)	***42***	D feed flow – Stream 2 (MV)
17	Stripper underflow (PM)	***43***	E feed flow – Stream 3 (MV)
18	Stripper temperature (PM)	***44***	A feed flow – Stream 1 (MV)
19	Stripper steam flow (PM)	***45***	Total feed flow – Stream 4 (MV)
20	Compressor work	***46***	Compressor recycle valve (MV)
21	Reactor cooling water outlet temp. (PM)	***47***	Purge valve (MV)
22	Separator cooling water outlet temp. (PM)	***48***	Separator product liquid flow (MV)
23	Reactor feed component *A* (CM)	***49***	Stripper product liquid flow (MV)
24	Reactor feed component *B* (CM)	***50***	Stripper steam valve (MV)
25	Reactor feed component *C* (CM)	***51***	Reactor cooling water flow (MV)
26	Reactor feed component *D* (CM)	***52***	Condenser cooling water flow (MV)

6.6.3 Calcium carbide process

The third case study involves process data from a submerged arc furnace from a commercial calcium carbide facility (Aldrich & Reuter, 1999). A key performance indicator of the process is the carbide product tonnage. A fault diagnosis problem was created by segmenting a set of 240 samples on 9 process variables into NOC and fault classes, based on a cut-off value for the carbide product response (which was not included in the process variable data set). The NOC data represent high product tonnage, while the fault data represent low product tonnage. 98 samples are available as training NOC data, with 25 NOC samples for validation. The fault data set consists of 117 samples. The process variables are presented in Table 6.3.

Table 6.3: Process variables of the calcium carbide process data set

Variable	Description
1	Furnace load
2	Power consumption
3	Average resistance underneath three electrodes
4	Lime consumption
5	Charcoal consumption
6	Coke consumption
7	Anthracite consumption
8	Lime under-burnt
9	Lime over-burnt

From process knowledge of the calcium carbide furnace (Jemwa & Aldrich, 2006; Aldrich & Reuter, 1999), the following is known: A high overall furnace load (variable 1), combined with high loads of lime (variable 4), charcoal (variable 5) and coke (variable 6) produced high quantities of high grade product. These four variables are also highly correlated. In contrast, the power consumption (variable 2), electrode resistance (variable 3) and anthracite (variable 7) were weakly correlated with the product grade and quality, while the lime quality (variables 8 and 9) appeared to have a negligible influence on the performance of the furnace.

Nomenclature

a	Autocatalytic dimensionless number
A	Tennessee Eastman reactant
c	normal deviate
C	contribution
C	Tennessee Eastman reactant
D	Tennessee Eastman reactant
e	independent Gaussian noise
E	Tennessee Eastman reactant
F	Tennessee Eastman byproduct
H	Tennessee Eastman product
m	number of process variables
N	sample size
$\mathbf{P}$	principal components
Q	residuals on individual variables
r	residual distance and indication of residual space
s	score distance and indication of score space
$\mathbf{T}$	feature matrix
$\mathbf{w}$	1SVM separating hyperplane
$\mathbf{V}$	eigenvectors
$\mathbf{X}$	process data
$\mathbf{X}'$	reconstructed process data
a	reduced dimensionality / number of features
α	significance level
γ	detection delay
δ	missing alarm rate
ζ	false alarm rate
θ	detection threshold parameter
λ	global structure preservation measure
$\mathbf{\Lambda}$	eigenvalues
ν	1SVM decision surface complexity parameter
$\boldsymbol{\xi}$	1SVM margin error
ρ	1SVM threshold
$\mathbf{\Sigma}$	covariance matrix

CHAPTER 7 - FAULT DIAGNOSIS: APPLICATIONS

Random forest feature extractive fault diagnosis, as developed in Chapter 6, is compared to principal component analysis fault diagnosis on a simple nonlinear system, the benchmark Tennessee Eastman process and the real-world calcium carbide process.

For the simple nonlinear system, the random forest method shows positive results compared to the principal component method. Random forests result in lower detection delays and accurate contribution estimates, while false alarm rates are similar to the principal component method, and missing alarm rates are slightly worse. Random forest results for the Tennessee Eastman process show similar missing alarm rates to the principal component approach, but worse rates than kernel principal component analysis and kernel independent component analysis approaches. Random forest fault detection proves to be slightly better than principal component analysis for the calcium carbide process, with an additional 8% of the fault data set detected by the random forest approach. For all three of these applications, the random forest residual space diagnostic proves to be more successful than its score space counterpart.

A conclusion that arises from all three case studies is that further investigation of the nature of the random forest feature space, especially for fewer features, is merited. This will then form the basis of the next chapter.

Overall, from the fault diagnosis performance criteria, the random forest approach developed in this study can be considered a suitable option for fault diagnosis.

7.1 Simple nonlinear system

The simple nonlinear system consists of normal operating conditions data of 200 samples in three dimensions, and two faults with the same number of samples. For both faults, a change in the third variable is the cause of the fault condition.

7.1.1 Selecting number of model components

The selection of the number of model components a (principal components or random forest features) is analogous to the selection of the intrinsic dimensionality as applied to feature extraction. By inspecting a scree plot of the variance contributions of each component or feature, a can be estimated. Another consideration is that the original data are three-dimensional. The scree plots for PCA and RF based on NOC data are given in the following figures.

From the scree plot for principal components in Figure 7.1, it is seen that two components account for nearly 100 % of the variance present in the NOC data. Two model components will thus be used for the PCA model. The interpretation of the random forest scree plot (Figure 7.2) is less straightforward. The original and permuted data variance lines cross at four model components, which only accounts for 60 % of variance present in NOC data. However, the NOC data are only three-dimensional. For the purpose of this study, two components will also be used for the random forest model. This will allow comparison with PCA in terms of information captured by the same number of components.

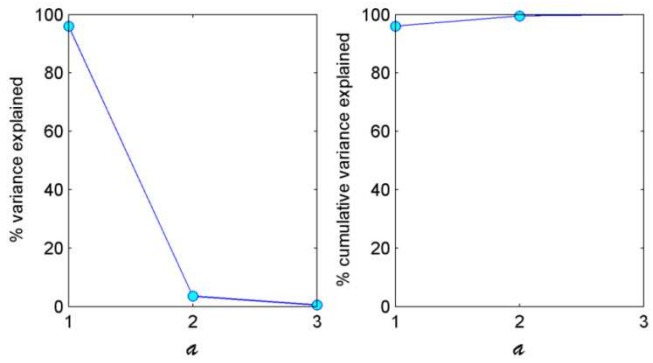

Figure 7.1: Scree plot and cumulative variance for simple nonlinear system principal components

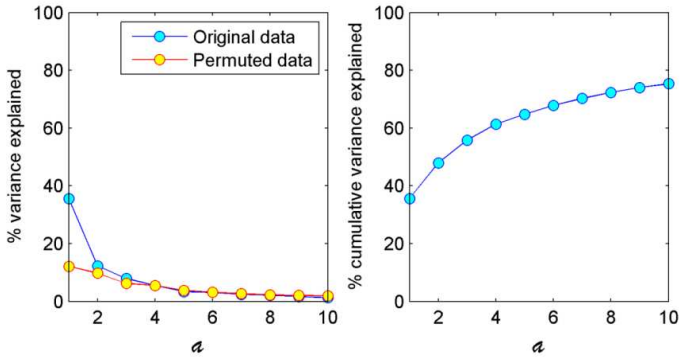

Figure 7.2: Scree plot and cumulative variance for simple nonlinear system random forest features

7.1.2 Reconstruction of NOC and fault data

The squared correlations between the original variables **X** and reconstructed variables **X'** for the NOC training data are 0.996 (λ_L) for the PCA method and 0.995 (λ_N) for the RF method. Both methods are thus very capable in reconstructing the original NOC data.

Figure 7.3 presents the NOC and fault data as reconstructed by the PCA and RF algorithms. Note that the fault data remained unseen to these algorithms during training. From the PCA reconstruction, the essence of the fault structures can be seen, with the fault data deviating from the reconstructed NOC data. Opposed to this, the RF reconstructions are less illuminating with regard to the original fault structure. The RF reconstructed NOC data also appears noisy, as compared to the PCA reconstruction.

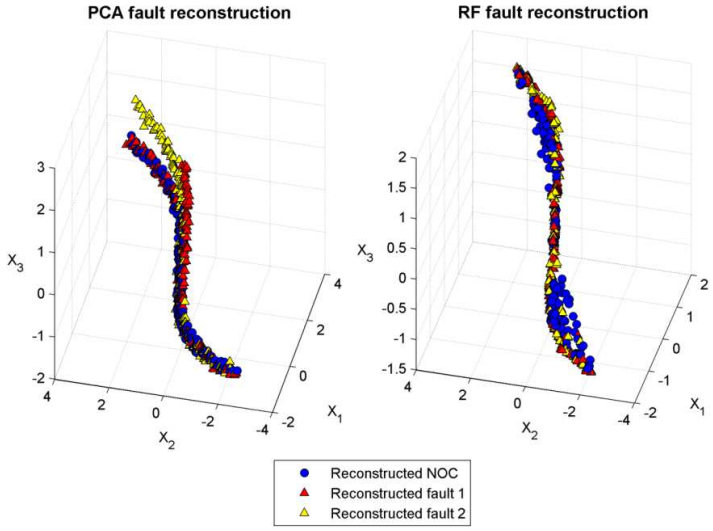

Figure 7.3: Reconstructed NOC and fault data with PCA and RF models for simple nonlinear system

7.1.3 Alarm rates and detection delays

A confidence level of 99 % was used in this study. No validation NOC data were used in this study, so no false alarm rates for NOC are available. False alarm rates, missing alarm rates and detection delays were calculated for faults 1 and 2. Of the 200 samples, 50 samples for fault 1 and 150 samples for fault 2 are known to represent abnormal conditions. The remaining 150 and 50 samples for fault 1 and 2, respectively, can then be used to calculate false alarm rates.

Table 7.1: False alarm rates (ζ), missing alarm rates (δ) and detection delays (γ) for the simple nonlinear system

	ζ (PCA)	ζ (RF)	δ (PCA)	δ (RF)	γ (PCA)	γ (RF)
Fault 1	**0.02**	0.04	**0.47**	0.55	55	**28**
Fault 2	**0.04**	**0.04**	**0.63**	0.73	105	**61**

From Table 7.1, PCA achieves a better false alarm rate only for fault 1, and the difference is marginal. PCA missing alarm rates are lower for both faults, while RF detection delays are considerably shorter.

Table 7.2: Score and residual distance based missing alarm rates for the simple nonlinear system

	δ_s (PCA)	δ_r (PCA)	δ_s (RF)	δ_r (RF)
Fault 1	0.97	**0.47**	0.56	0.55
Fault 2	1.00	**0.63**	0.89	0.73

Inspecting the nature of the fault detections (Table 7.2), it is apparent that PCA score distance based detection is a failure for both faults 1 and 2. Residual distance based detection are superior for both PCA and RF for these faults, although only marginally so for RF. RF score based detection is better (in terms of lower missing alarm rates) than PCA scored based detection for this data set. A possible explanation for better RF score distance detection may be the nonlinear nature of its features and confidence limits.

The influence of the number of model components a on the various rates and delays are depicted in Figure 7.4, Figure 7.5 and Figure 7.6. A maximum of three components can be investigated for PCA (as the dimensionality of the system is three), while five components were tested for RF.

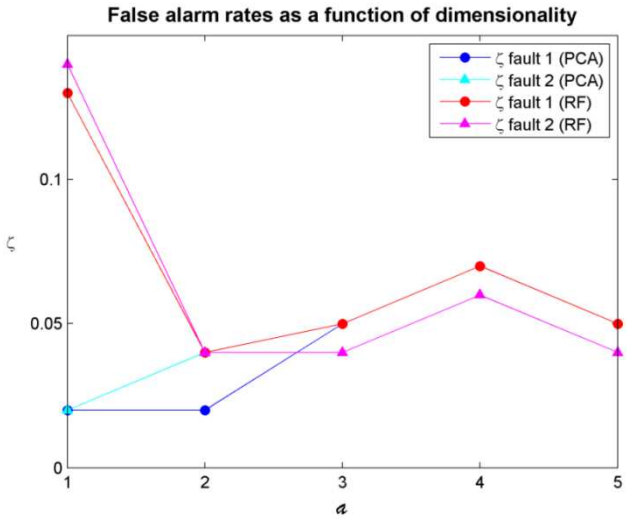

Figure 7.4: Simple nonlinear false alarm rates as function of number of model components

The false alarm rates increase with number of model components for both faults in the case of PCA, while an initial decrease and subsequent increase is present for RF rates on faults 1 and 2.

As the number of model components increase for the PCA model, the score space may incorporate more noise rather than more signals. The simple nonlinear system NOC is essentially a one-dimensional manifold. Incorporating more components may obscure information.

Conversely, too few components may not accurately capture all information. This is evident from the RF false alarm rate profile: using only one model component results in a higher false alarm rate than two components or more.

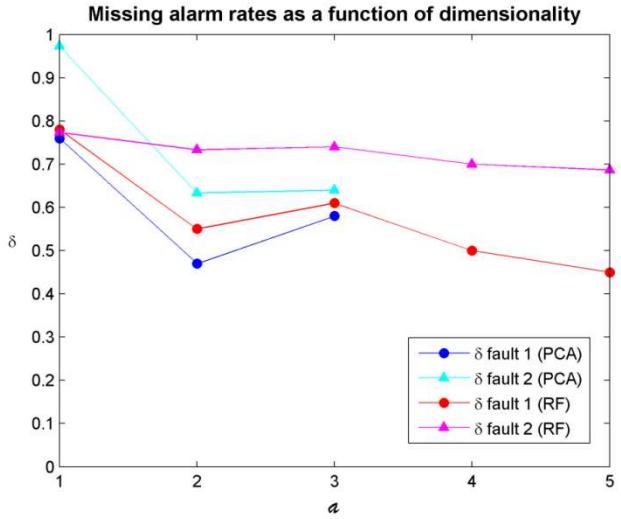

Figure 7.5: Simple nonlinear missing alarm rates as function of number of model components

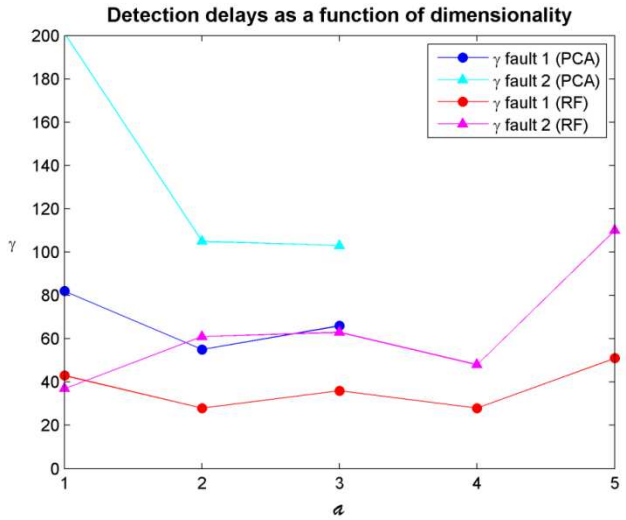

Figure 7.6: Simple nonlinear detection delays as function of number of model components

From Figure 7.5, missing alarm rates for both fault 1 and 2 generally decrease with increase in model components for RF models, while PCA missing alarm rates decrease with two components and stabilize or increase with the third and final model component. This increase may be due to the addition of an information-poor component as the third component, seeing that the system is essentially one-dimensional, but linearly characterized in two dimensions. The slow but steady decrease in RF missing alarm rates indicate that more informative RF features may be available than only the two considered in the original model, or the five investigated here.

The PCA detection delays (Figure 7.6) as function of number of model components exhibit a similar pattern to that of PCA missing alarm rates, and probably due to the same underlying information-noise trade-off. The RF detection delays (Figure 7.6), however, do not decrease consistently with increased dimensionality. This may indicate that the distribution of detections are independent of number of components. This statement is made in lieu of the specification that three consecutive detections are needed to calculate detection delay.

7.1.4 Variable contributions

Variable contributions can be determined from the PCA model from score and residual distance contributions, while only residual distance contributions are available for the RF model. It is known that variable 3 is associated with both faults, as only changes to the parameters of the model generating variable 3 were made. Figure 7.7 and Figure 7.8 depict average contributions, where the average is calculated over samples that are detected.

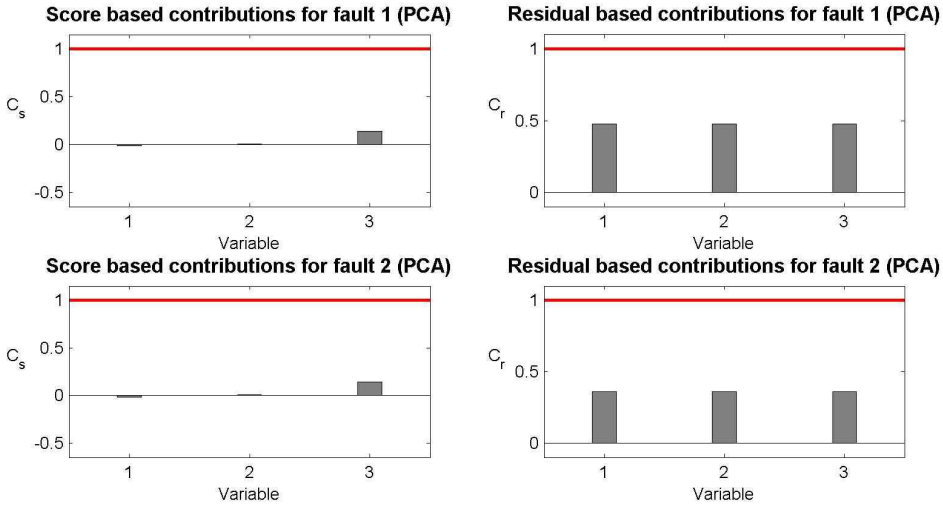

Figure 7.7: Scaled PCA variable contributions for simple nonlinear system (red lines signify the 99% percentile of contributions calculated from NOC training data)

The PCA contributions fail to exceed the 99% percentile level of the NOC data, which gives no further insight as to the cause of the faults.

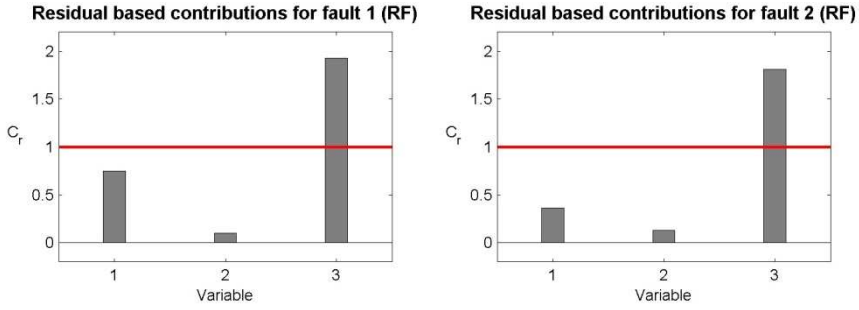

Figure 7.8: Scaled random forest variable contributions for simple nonlinear system (red lines signify the 99% percentile of contributions calculated from NOC training data)

The RF residual based contributions for both faults successfully rank variable 3 as being the only variable associated with the fault condition; exceeding the NOC 99% percentile contribution levels.

7.1.5 Discussion of simple nonlinear system results

The PCA and RF fault diagnosis algorithms require the specification of two parameters: the number of model components a (analogous to intrinsic dimensionality) and the confidence level for limits. A confidence level of 99% was selected for this study, representing an expected false alarm rate on normal operating conditions data of one falsely alarmed sample for every hundred samples. The number of model components is the most crucial selection, and was chosen as two for the initial PCA and RF models.

The ability of the PCA and RF models to adequately reconstruct the training NOC data was shown to be high, with reconstruction correlations exceeding 0.99. Reconstruction of unseen fault data shows interesting aspects of the two models: the PCA reconstructions capture the known structure of the faults, while the RF reconstructions do not appear to resemble the original fault structures. This may be another demonstration of the failure of random forest demapping models to extrapolate (in terms of variable space reconstruction) to unseen data, as expounded upon in Chapter 5.

The performance measures for fault detection sketch an overall positive result for the RF method. False alarm rates are similar to the PCA model, missing alarm rates are worse (but only in the order of 10% worse), while RF improves on detection delays: nearly half that of the PCA detection delays.

In dissecting the nature of the detections, it is found that RF score based detection does better than PCA score based detection. This may be a result of the nonlinear features and confidence limits associated with the RF method. However, for both PCA and RF, the residual based detections are the dominant accurate detective measures.

In terms of the influence of the crucial model parameter (number of components) on performance measures, the only confident conclusion that can be made is that the more RF features are used, the lower the missing alarm rate (for the range of components investigated). However, increased RF components seem to increase detection delay and false alarm rates (to a lesser degree).

RF variable contribution successfully identifies variable 3 as the variable most closely associated with the faults. PCA contribution measures fail by not indicating any variable as having a significant contribution.

7.2 Tennessee Eastman process

The Tennessee Eastman process NOC data consists of 500 samples of 52 measured variables of a simulated chemical process with an implemented control scheme. Further data consists of 960 samples each for 21 fault conditions. The inception of each fault is at sample number 160 for these fault data sets, and various variables may achieve steady-state values due to the control scheme adjustments, although not necessarily steady-state values corresponding to normal operating conditions.

7.2.1 Selecting number of model components

Variance and cumulative variance plots were generated to aid in the selection of the number of model components to select for both the PCA and RF algorithms.

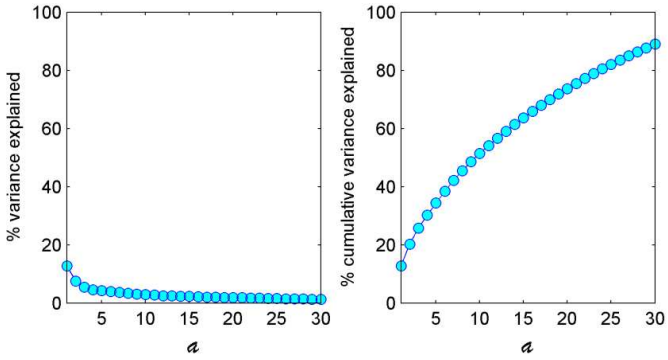

Figure 7.9: Scree plot and cumulative variance for Tennessee Eastman process principal components

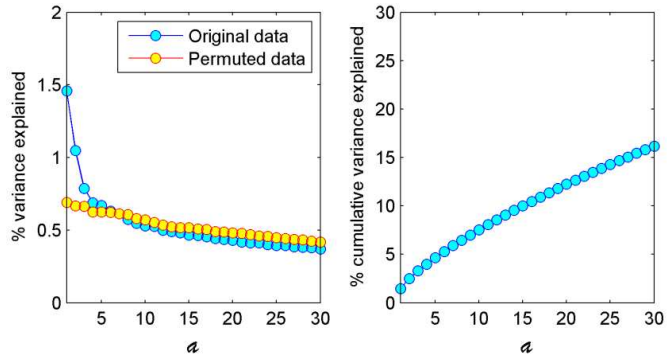

Figure 7.10: Scree plot and cumulative variance for Tennessee Eastman process random forest features

From Figure 7.9, to account for 90% cumulative variance in the training NOC data, 30 principal components are required. This number was thus selected as the number of model components for the PCA algorithm. From Figure 7.10, crossing occurs at around seven random forest features, even though this accounts for less than 8% of cumulative variance. Seven components were used in the RF algorithm.

The effect of dimensionality on various fault detection performance measures are included where these measures are discussed in the following subsections.

7.2.2 Reconstruction of NOC data

As with the simple nonlinear system, the correlation between the original variables **X** and their model reconstructions **X**' can be measured. For PCA models, this reconstruction correlation corresponds to λ_L as used in Chapter 5, while for RF models reconstruction correlation is equivalent to λ_N. The reconstruction correlations for each of the original variables are given in Figure 7.11. RF shows very accurate reconstruction for all variables of the training NOC (correlations of at least 0.9), but erratic and mostly low correlations for the unseen NOC data. PCA shows lower correlations for the seen NOC data (ranging from 0.6 to 1), but unseen NOC correlations are much better than compared to the RF case. This shows that PCA generalizes better to unseen data (in terms of reconstruction), as well as that the training NOC data set may not have been representative enough for the entire extent of normal operations.

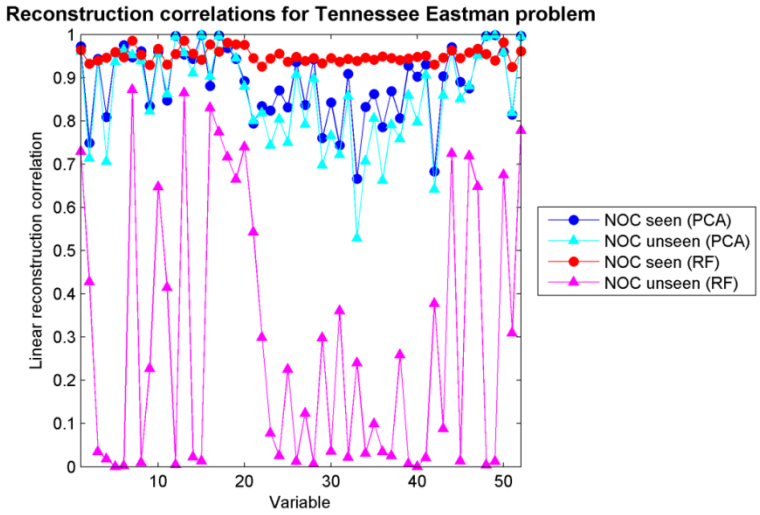

Figure 7.11: Variable space reconstruction correlations for the Tennessee Eastman process, calculated for training (seen) NOC data and unseen NOC data

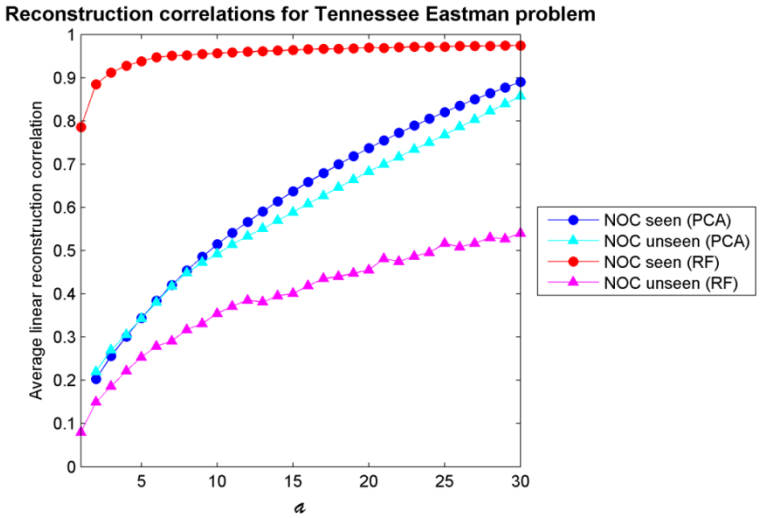

Figure 7.12: Average variable space reconstruction correlations for the Tennessee Eastman process for a range of model components (calculated for training / seen NOC data and unseen NOC data)

Figure 7.12 depicts the average variable space reconstruction correlations for a range of model components. As in the previous figure, it is clear that RF reconstructions have high accuracies for seen NOC and low accuracies for unseen NOC. PCA reconstructions have similar accuracies for seen and unseen data. As the number of model components increase, there is an increase in all reconstruction correlations.

7.2.3 Alarm rates and detection delays

A confidence level of 99% was used in this study. The initial false alarm rates on the validation / unseen NOC data were 0.15 and 0.997 for the PCA and RF models, respectively. The influence of the number of model components on the false alarm rate of unseen NOC data is shown in Figure 7.13. The most distinctive conclusion from this plot is the very large (approximately 1) false alarm rates for the RF models, irrespective of the number of model components. This is due to the very low accuracy in reconstruction of the unseen NOC data, which leads to high residual distances, and thus detections. As mentioned before, this low accuracy in reconstruction may be due to lack of representative NOC data used in training.

The low false alarm rates for PCA models may suggest that the training data is sufficiently representative for a linear feature extraction technique. These low false alarm rates for PCA may also suggest that the score and

residual space diagnostics (based on assumed distributions) are more conservative than the percentile approach employed by the RF framework. Another conclusion that arises from this is that RF models, for this data set, may require more; and more representative, samples of NOC training data to achieve generalization accuracy comparable to PCA with smaller sample sizes.

Another illuminated trait from the plot is the initial increase and eventual plateau of the PCA false alarm rates for unseen data. With more model components, one may be liable to add less informative / noisier components, which increases false alarms.

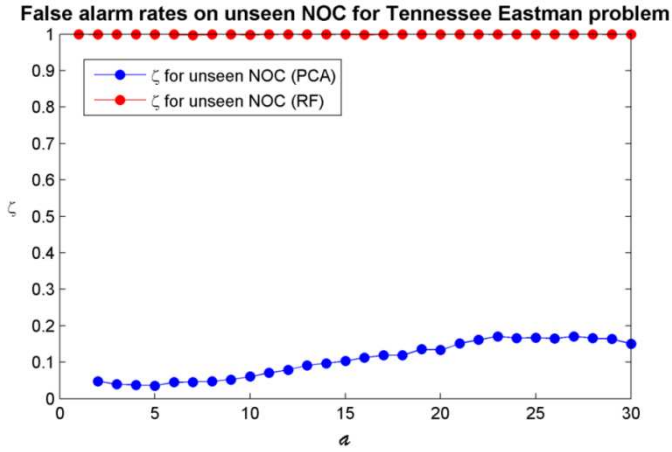

Figure 7.13: False alarm rates on unseen NOC data for the Tennessee Eastman process for various number of model components

In order to compare fault detection performance on fault data on an equal footing, the score and residual distance confidence limits are adjusted for both models, incorporating the previously unseen NOC data. Confidence limits are set using the percentile approach to the 99% percentile level of the previously unseen NOC data.

False alarm rates, missing alarm rates and detection delays were calculated for the 21 faults of the Tennessee Eastman process data set. Each fault data set consists of 960 samples, with each fault only commencing from the 161th sample. The first 160 samples can thus be used to calculate false alarm rates.

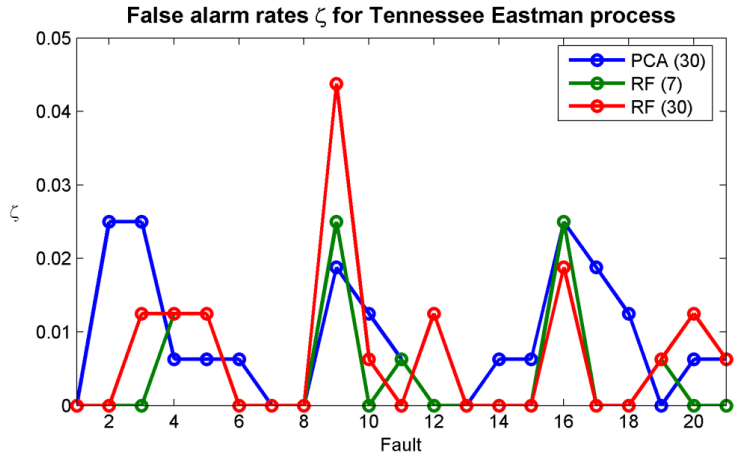

Figure 7.14: False alarm rates on fault data for the Tennessee Eastman process (number of model components in brackets)

From Figure 7.14, PCA has worse (higher) false alarm rates than RF (for 7 and 30 components) for eight faults, and better false alarm rates for five faults. RF with seven components has similar or better false alarm rates than RF with thirty components for all but two faults. In general, it seems that more RF components lead to worse false alarm rates.

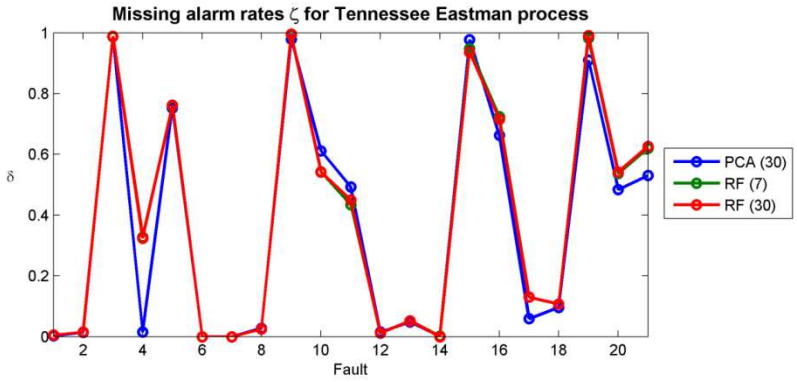

Figure 7.15: Missing alarm rates on fault data for the Tennessee Eastman process (number of model components in brackets)

From Figure 7.15, the overall impression is that all three models achieve very similar missing alarm rates for all faults (evident from the level of overlap of the line plots). PCA has better (lower) missing alarm rates for six faults, and RF (7 and 30 components) have better alarm rates for three faults. The only difference in missing alarm rates exceeding 10 % occurs at fault 4. The missing alarm rates for RF with 7 or 30 components are nearly identical.

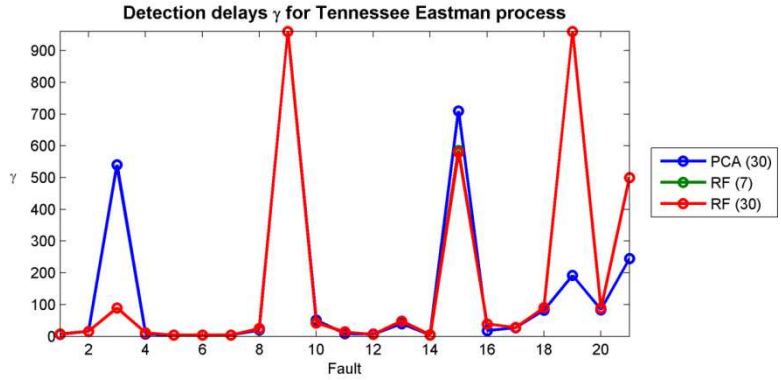

Figure 7.16: Detection delays on fault data for the Tennessee Eastman process (number of model components in brackets)

From Figure 7.16, PCA has significantly better (shorter) detection delays for two faults, and RF (7 and 30 components) has better detection delays for two faults. Detection delays for RF with 7 or 30 model components are nearly identical.

As seen from the previous figures, missing alarm rates and detection delays are nearly identical for both 7 and 30 component RF models. The only significant difference in terms of model components is in false alarm rates, where fewer components gave better (lower) false alarm rates for most faults. A possible suggestion from these results is then that fewer model components are suitable for RF fault diagnosis.

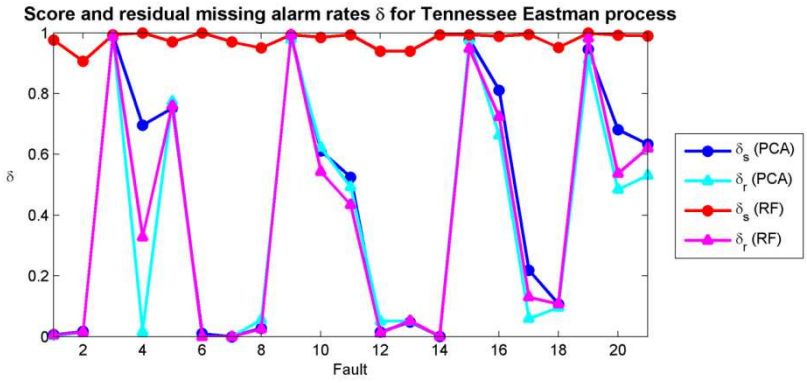

Figure 7.17: Score and residual distance missing alarm rates on fault data for the Tennessee Eastman process (PCA with 30 components and RF with 7 components)

A comparison of the score and residual distance missing alarm rates for PCA with 30 components and RF with 7 components are given in Figure 7.17. Score distance missing alarm rates for RF fault diagnosis is clearly the worst (highest) for all faults. RF residual distance missing alarm rates are comparable to, and sometimes better than (for two faults), both PCA score and residual distance missing alarm rates. PCA score and residual distance alarm rates are similar, with score distance missing alarm rates better (lower) than their residual distance counterpart for five faults.

A comparison of PCA and RF missing alarm rates to results obtained in literature for KPCA and KICA are shown in Figure 7.18 and Figure 7.19 (Zhang & Qin, 2008).

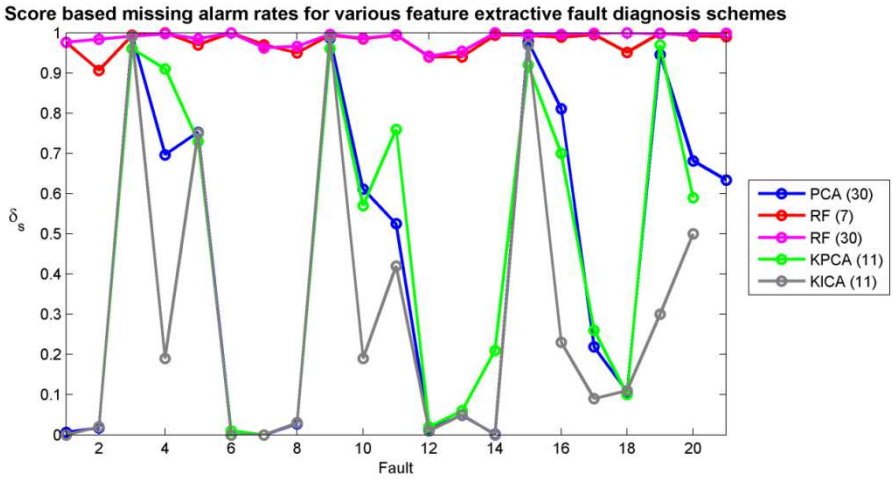

Figure 7.18: Score distance missing alarm rates on fault data for the Tennessee Eastman process, including KPCA and KICA results (number of model components in brackets)

From Figure 7.18, KICA score distance missing alarm rates are the overall best, and RF score distance missing alarm rates are the overall worst, irrespective of number of RF model components.

From Figure 7.19, KICA and KPCA have better (lower) residual distance missing alarm rates than PCA and RF for five faults. RF residual distance missing alarm rates are more competitive than its score distance counterpart (see previous figure). For twelve faults, the residual distance missing alarm rates for all methods are very similar.

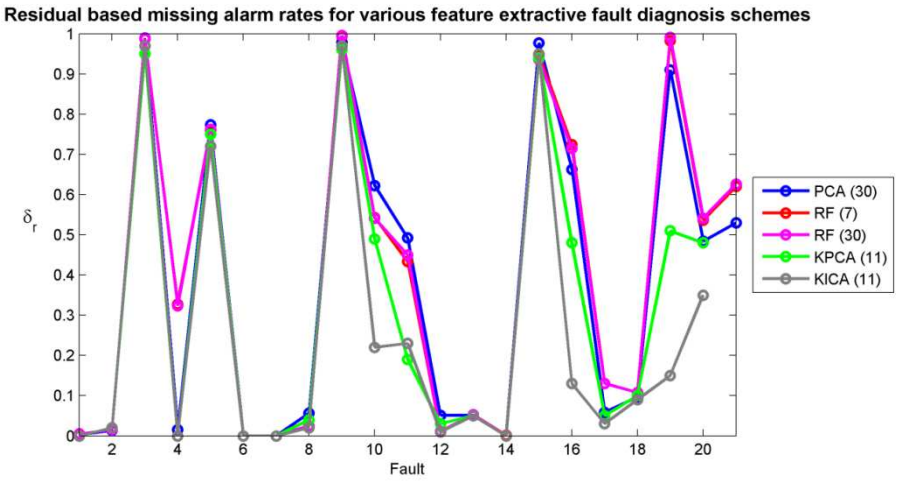

Figure 7.19: Residual distance missing alarm rates on fault data for the Tennessee Eastman process, including KPCA and KICA results (number of model components in brackets)

- Influence of number of model components

The following figures present the influence of the number of PCA or RF model components on the average over all faults for false alarm rates, missing alarm rates and detection delays for the Tennessee Eastman process. For each model manifestation, the confidence limits for score and residual distances were reset using the percentile approach to a 99 % level based on unseen NOC validation data.

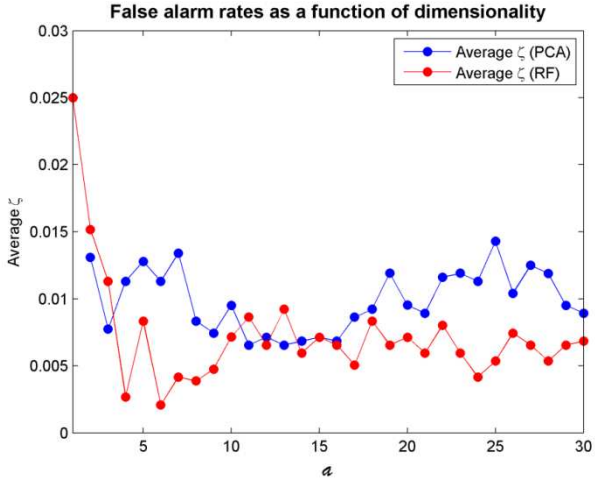

Figure 7.20: Average false alarm rates on fault data for the Tennessee Eastman process for various numbers of model components

Initially, as the number of model components increase, there is a corresponding decrease on RF false alarm rates for fault data. For more than 6 components, the false alarm rates increase again until 13 components, then remain stable (with oscillation) up to 30 components. This may be an indication of the addition of informative / noisy component trade-off. This trade-off is less visible for the PCA case.

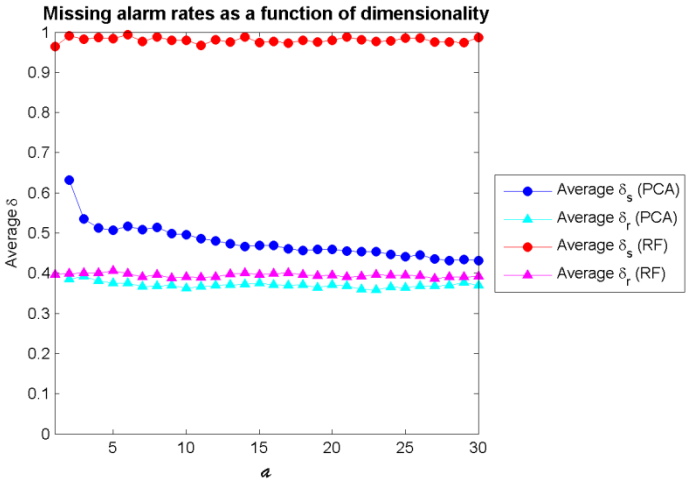

Figure 7.21: Average missing alarm rates on fault data for the Tennessee Eastman process for various numbers of model components

From Figure 7.21, RF missing alarm rates (based on score and residual distances) are insensitive to the number of model components. This confirms an earlier observation that RF with 7 or 30 components gave nearly identical results for missing alarm rates. In contrast, score distance missing alarm rates for PCA shows a steady decrease as model components increase. PCA residual distance missing alarm rates remain relatively constant. Again, it is apparent that RF score distance missing alarm rates are the worst (highest).

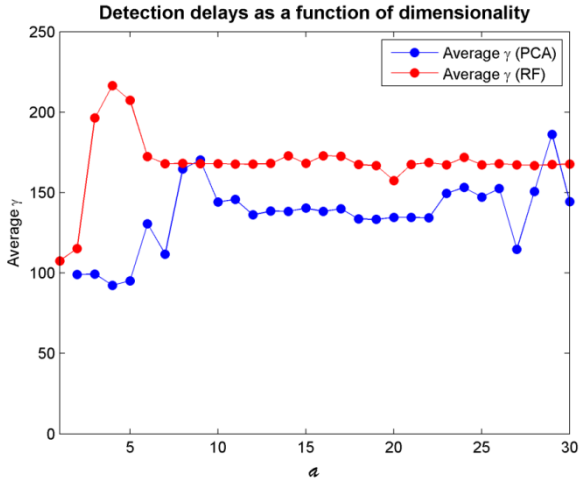

Figure 7.22: Average detection delays on fault data for the Tennessee Eaastman process for various numbers of model components

From Figure 7.22, RF shows longer average detection delays than PCA for most number of model components. Whereas PCA shows an erratic, but mostly monotone, increase in detection delay with number of model components, RF shows an initial increase for the first four components, followed by a decrease and then constant level. This suggests that, with fault information available, one may be able to tune the number of model components in order that the fault diagnosis model delivers a low average detection delay, with conversely higher false alarm rates (see Figure 7.16).

7.2.4 Contributions

To assess the ability of PCA and RF fault diagnosis methods to determine the causal or affected process variables, five of the twenty-one faults for which causal or affected variables are known were chosen to assess variable contributions. These faults are fault 4, 5, 11, 12 and 14, and the causal process variables are identified based on observations made in fault diagnostic literature on the Tennessee Eastman process. The

contributions are calculated as the average contributions of samples that are both indicated as faulty and known to be faulty.

The identified variables may not be causal, but may be closely related to the fault. For example, when the reactor temperature increases due to some external disturbance, the cooling water flow rate to the reactor will increase due to closed loop control. Cooling water flow rate is then an affected variable, and not a causal variable.

- Fault 4

Fault 4 is simulated by introducing a step change in the reactor temperature (variable 9). As control loops compensate for the temperature increase, a step change in reactor cooling water flow rate (variable 51) is induced, while all other variables return to steady state. Variable 51 is assumed to be most closely associated with the fault (Russell et al., 2000). From the following contribution plots[12], it is clear that both PCA and RF can successfully identify variable 51 for this fault.

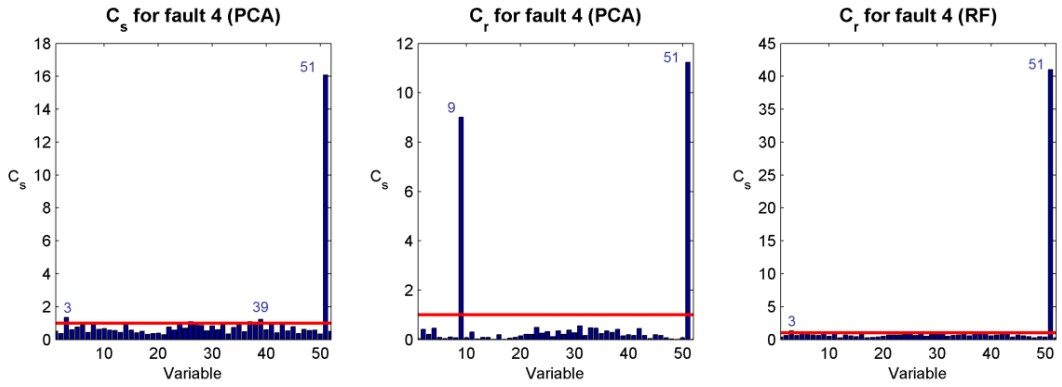

Figure 7.23: Scaled contribution plots for fault 4 of the Tennessee Eastman process (PCA with 30 components, RF with 7 components; 99% percentile NOC contributions shown in red)

- Fault 5

A step change in condenser cooling water inlet temperature is the defining condition for fault 5. This induces a step change in condenser cooling water flow rate (variable 52), and increases the flow rate of the vapour liquid separator feed, which subsequently results in an increase in the vapour liquid separator temperature (variable 11). This temperature increase further induces an increase in the separator cooling water outlet temperature (variable 22) (Russell et al., 2000; Shao & Rong, 2009). The contributions plot based on PCA scores show variable 11 with the most significant contribution, and ranks variable 22 fourth. The contribution plot based on PCA residuals shows no significant contributing variables. The contribution plot based on RF residuals rank variable 11 as most significant, with more than twenty other variables also shown as significant. Variable 52 is not shown as significant in any of the contribution plots.

[12] *Contribution plots for this study are scaled by dividing the variable contributions for the fault conditions by the 99% percentile variable contributions of the normal operating conditions. Scaled contributions with values above 1 are thus considered to be significant.*

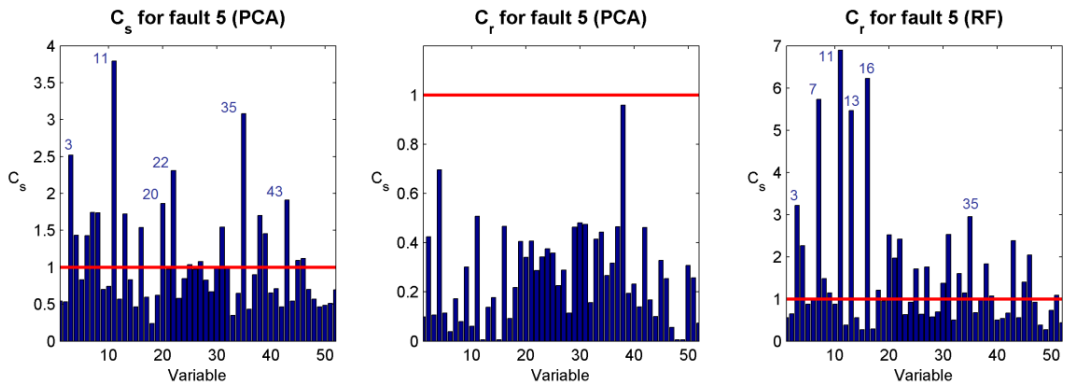

Figure 7.24: Contribution plots for fault 5 of the Tennessee Eastman process (PCA with 30 components, RF with 7 components; 99% percentile NOC contributions shown in red)

- Fault 11

Fault 11 is simulated as random variation in reactor cooling water inlet temperature. This causes large oscillations in the reactor cooling water flow rate (variable 51) and the reactor temperature (variable 9) (Russell et al., 2000). From the following contribution plots, it is clear that both PCA and RF can successfully identify variables 9 and 51 for this fault.

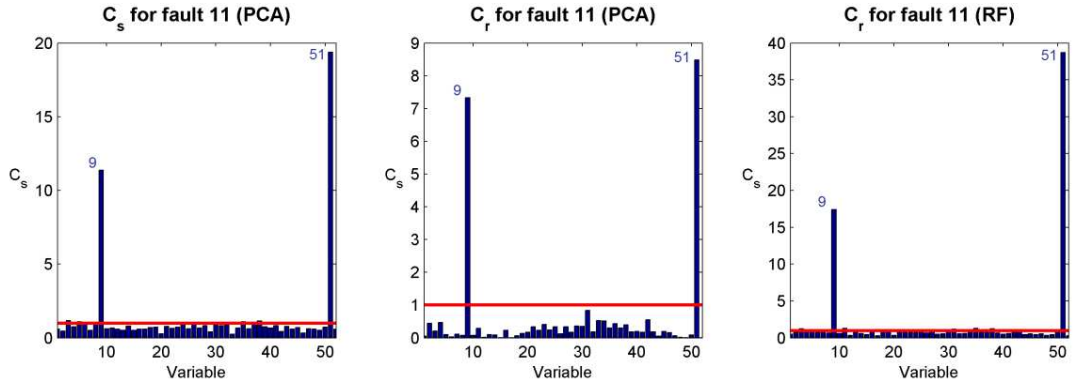

Figure 7.25: Contribution plots for fault 11 of the Tennessee Eastman process (PCA with 30 components, RF with 7 components; 99% percentile NOC contributions shown in red)

- Fault 12

Fault 12 is simulated as random variation in the condenser cooling water temperature, which induces abnormal behaviour in many variables, including the separator temperature (variable 11), separator pressure (variable 13) and separator outlet cooling water temperature (variable 22) (Russell et al., 2000; Shao & Rong, 2009). PCA score contributions rank variables 11, 13 and 22 as second, fourth and third most significant, respectively. PCA residual contributions rank variable 11 fifth most significant, while RF contributions rank variable 11 and 13 fifth and first most significant.

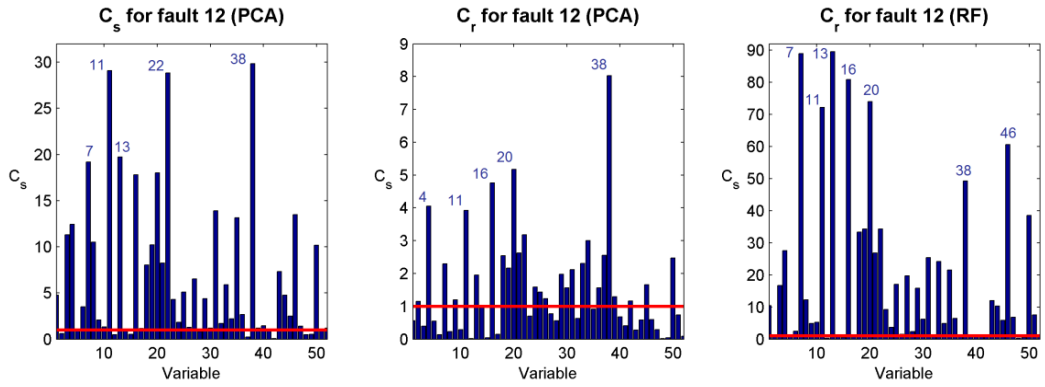

Figure 7.26: Contribution plots for fault 12 of the Tennessee Eastman process (PCA with 30 components, RF with 7 components; 99% percentile NOC contributions shown in red)

- Fault 14

A sticking valve for reactor cooling water is simulated for fault 14, causing large fluctuations in reactor temperature (variable 9), the reactor cooling water outlet temperature (variable 51) and the reactor cooling water flow rate (variable 21) (Russell et al., 2000; Shao & Rong, 2009). PCA score contributions and RF contributions successfully rank these three variables as the three most significant variables, while PCA residual contributions do not identify variables 9 and 51 as related to this fault.

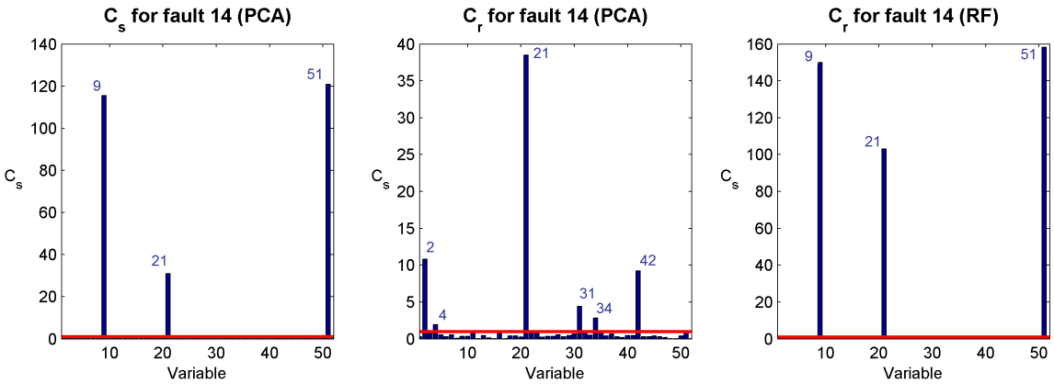

Figure 7.27: Contribution plots for fault 14 of the Tennessee Eastman process (PCA with 30 components, RF with 7 components; 99% percentile NOC contributions shown in red)

7.2.5 Computational aspects

Figure 7.28 gives the computational times for the two fault diagnostic algorithms applied to the Tennessee Eastman process data in this study. These computational times include the offline training on NOC data, the updating of confidence limits based on unseen NOC data, the online implementation of the trained models on twenty-one fault conditions, and the calculation of performance measures.

The RF fault diagnosis method has computational times of at least an order of a magnitude larger than that of PCA fault diagnosis. This is due to the computational expenses of constructing a N x N proximity matrix, training a regression forests for mapping, training m regression forests for demapping, training 1SVM models by cross-validation and calculating 1SVM-based score distances.

A trend is also observed in that the computational times increase with an increase in number of model components. This is expected, as for RF models, more mapping regression forests must be trained.

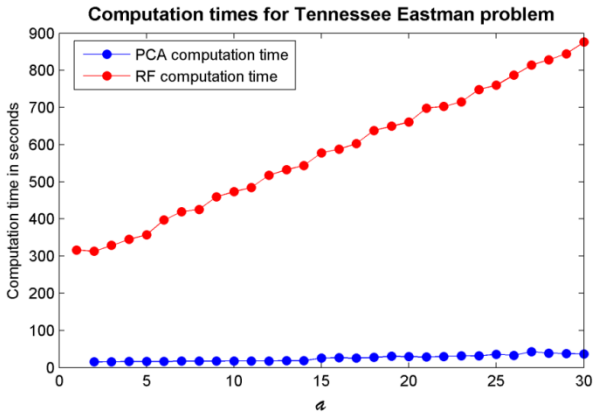

Figure 7.28: Computation times for different number of model components for the Tennessee Eastman process

7.2.6 Repeatability aspects

As the random forest fault diagnostic approach involves stochastic models, the repeatability of this method was investigated. Table 7.3 presents the results from these repeatability tests, based on ten replicates of random forest fault diagnosis runs on the Tennessee Eastman process. For a specified number of model components, the average false alarm rates, average missing alarm rates and average detection delays over all 21 faults were calculated. Standard deviations for ten replicates of these averages could then be calculated.

Table 7.3: Standard deviations of average false alarm rates (ζ), missing alarm rates (δ) and detection delays (γ) for ten replicates of random forest fault diagnosis on the Tennessee Eastman process

	a = 5	a = 10	a = 15	a = 20	a = 25
Standard deviation for average ζ	0.0021	0.0020	0.0029	0.0025	0.0018
Standard deviation for average δ	0.0043	0.0031	0.0048	0.0046	0.0033
Standard deviation for average γ	25.1	5.2	20.2	14.1	17.6

Given that the range of possible false and missing alarm rates is [0,1]; the standard deviations for the selected number of model components are acceptably small. The range for detection delays is [0 960]; and the detection delay standard deviations are also small. For this case study, it is then stated that random forest fault diagnosis shows good repeatability.

7.2.7 Discussion of Tennessee Eastman process results

The PCA and RF fault diagnosis algorithms require the specification of two parameters: the number of model components a and the confidence level for limits. A confidence level of 99 % was selected for this study, representing an expected false alarm rate on unseen normal operating conditions data of one falsely alarmed sample for every hundred samples. The number of model components is the most crucial selection, and was chosen as 30 for the initial PCA model and 7 for the initial RF model. This selection is based on at least 90 % cumulative variance explained for PCA, and the crossing criterion for RF. Sensitivity of the various performance measures to the number of model components was also investigated. [13]

[13] *Note: When trends are highlighted for performance measures based on number of model components, these trends are related to number of model components ranging from one to thirty, as this was the extent of model component sensitivity investigated in this case study.*

Reconstruction of process variables from features for the PCA and RF models were expressed as linear reconstruction correlations (λ_L) for PCA and nonlinear reconstruction correlations (λ_N) for RF. RF showed very high correlations for seen NOC, and very low correlations for unseen NOC, while PCA showed lower correlations than RF on seen NOC and better correlations than RF on unseen NOC. These results suggest that, for this data set, PCA generalizes better to unseen NOC. The low generalization ability of RF may be due to not enough (or representative enough) training data for NOC conditions. However, as PCA generalizes well, this may suggest that linear generalization is superior for this data set, given the same training data.

An increase in the number of model components increased the reconstruction correlations, for RF and PCA models, on seen and unseen NOC data. In terms of the false alarm rates on the unseen NOC, RF shows very high values. This is a result of the low reconstruction correlations on unseen NOC data. As the RF reconstructions are inaccurate, the residual distances will be large, leading to false detections. PCA false alarm rates on unseen NOC data are much lower in comparison to RF, and are a result of its better generalization.

An increase in the number of model components does not influence the RF false alarm rates on unseen NOC, but worsens the false alarm rate on unseen NOC for PCA models. This reflects the trade-off of adding informative variables versus adding noise variables. The same situation may be present for RF false alarm rates on unseen NOC, but since the RF false alarm rates are nearly at a maximum from the minimum number of model components, this cannot be observed.

A fair comparison on false alarm rates, missing alarm rates and detection delays for the twenty-one faults was ensured by adjusting the detection thresholds based on 99% percentiles of the score and residual distances for unseen NOC.

The false alarm rates for fault data showed a mixed bag for RF and PCA models, with RF models showing better rates for eight of the twenty-one faults, and PCA models showing better rates for five of the twenty-one faults. The average false alarm rates are generally lower for RF than PCA for a variety of model component quantities. By inspecting the influence of number of model components on RF false alarm rates, it appears that a trade-off between informative and noise features can be found around four to six RF model components. For PCA false alarm rates, this trade-off zone (in terms of false alarm rates) lies between ten and fifteen components. This suggests that optimization in terms of false alarm rate performance may provide a model component selection criterion, assuming that fault data are available.

The missing alarm rates of PCA and RF models are very similar, and the missing alarm rates for RF models with either seven or thirty components are nearly identical. This suggests that RF missing alarm rates are insensitive to number of model components selected. An inspection of the missing alarm rates for a variety of model component numbers confirms this statement. This may suggest that the crossing criteria approach may not be the optimal method for selecting the number of RF model components. In contrast, PCA missing alarms decrease with an increase in model components.

In terms of detection delays, performance is similar for PCA and RF models. PCA models have better detection delays for two faults, and RF models have better detection delays for two faults. Again, detection delays for RF models with either seven or thirty components are nearly identical. However, from an inspection of the detection delays for a variety of model component numbers, the RF detection delays show an initial increase, followed by a decrease and eventual stabilisation. PCA detection delays show steady increase of detection delay as model components increase. These trends may be another indication of possible trade-off between informative and noise components.

A closer inspection on the nature of the fault detections show that RF score distance based detection is very unsuccessful. RF distance based detection is similar to PCA score and distance based detection.

Comparing PCA and RF missing alarm rates to KPCA and KICA results from literature, show overall superior performance for the KICA model, based on five faults. The other sixteen faults show very similar results for PCA, RF, KPCA and KICA models.

RF contributions are fairly successful on the five faults investigated, showing correct significance and ranking for three faults, and at least one correct variable indication or ranking for the other two faults.

Even though seven RF components only account for 8% of cumulative variance in terms of square distances of multidimensional scaling components, RF performance on the Tennessee Eastman process is promising. The near identical missing alarm rates for seven and thirty RF model components suggest that seven components are sufficient in this case.

Finally, RF fault diagnosis is much more computationally expensive than PCA fault diagnosis, and this expense increases with the number of model components included. This provides additional motivation for the use of fewer model components in the RF approach.

7.3 Calcium carbide process

The calcium carbide process consists of 98 samples of 10 measured variables of a real-world mineral processing plant. Further data consists of 25 samples for NOC validation, and 117 samples for faulty conditions, where faulty conditions were initially specified as product tonnages lower than a certain minimum. The nature of the definition of the fault data implies that NOC and fault samples may overlap in the process variable space, due to noise and process lags. I.e. it is known that the fault data is not distinctly separated from the NOC data in the process variable space. A confidence level of 99% was enforced for score and residual diagnostic thresholds.[14]

7.3.1 Selecting number of model components

Variance and cumulative variance plots were generated to aid in the selection of the number of model components to select for both the PCA and RF algorithms. From Figure 7.29, to account for 90% cumulative variance in the training NOC data, 6 principal components are required. This number was thus selected as the number of model components for the PCA algorithm. From Figure 7.30, crossing occurs at around 23 random forest features, even though this accounts for less than 50% of cumulative variance. Based on the crossing criteria, 23 components were used in the RF algorithm.

The effect of score space dimensionality on various fault detection performance measures are included where these measures are discussed in the following subsections.

[14] *To ensure fair comparison of the PCA and RF frameworks, the confidence limits were adjusted to give a maximum false alarm rate of 1% on the training and validation NOC data.*

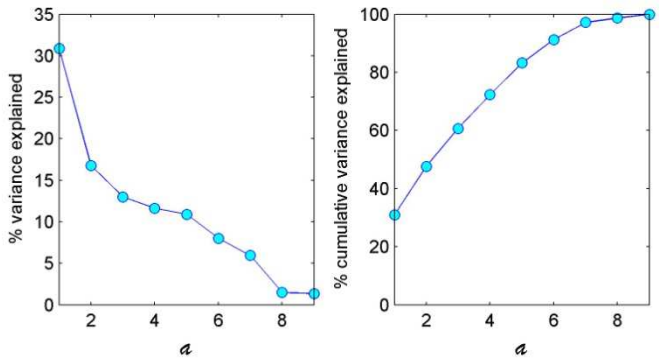

Figure 7.29: Scree plot and cumulative variance for calcium carbide process principal components

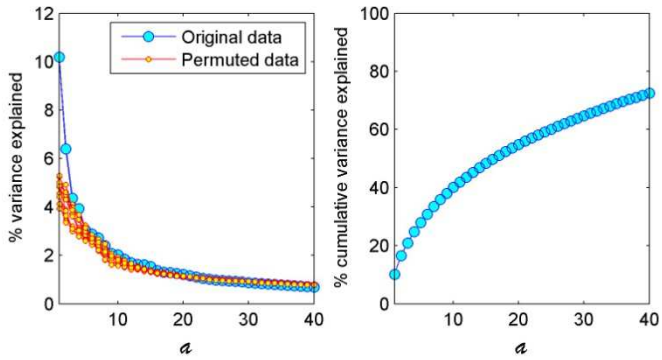

Figure 7.30: Scree plot and cumulative variance for calcium carbide process random forest features (10 different permuted scree plots are shown, to indicate the natural variability for different data permutations)

It is interesting to note that the crossing criteria suggests that 23 RF features are required to capture sufficient variance, since the original data set has a dimensionality of 9. Starting from the assumption that the crossing criteria approach correctly indicates the number of informative features, this suggests that more RF features are required to capture the data structure than the original dimensionality of the data. However, the nonlinear nature of the RF dissimilarity calculation does not explicitly take the original dimensionality into account. RF features can be considered analogous to kernel features, which may capture significant variance in higher dimensionalities than the original data.

The suitability of using the crossing criteria for selecting the number of features to use may become apparent by investigating the effect of using a range of features. Therefore, fault diagnosis with PCA and RF were also applied using a range of model components.

7.3.2 Reconstruction of NOC data

Figure 7.31 shows the range of average NOC reconstruction correlations for the PCA and RF frameworks. For PCA, these reconstructions were done with multiple linear regression models (giving λ_L), while RF regression was used for the RF framework (giving λ_N). For the crossing criteria number of model components (6 for PCA and 23 for RF), average adjusted R^2 (training NOC) values of 0.91 for PCA and 0.98 for RF were found; while average adjusted R^2 (validation NOC) values were 0.92 for PCA and 0.74 for RF.

As with the Tennessee Eastman results (Figure 7.12), it is clear that RF reconstructions have high accuracies for seen NOC and low accuracies for unseen NOC. However, compared to the Tennessee Eastman application, RF

shows better reconstruction ability for unseen NOC. As the same RF framework was applied to the Tennessee Eastman and the calcium carbide data sets, this suggests that the low unseen NOC reconstruction correlation is not exclusively symptomatic of the framework, but also dependent on the data. Better reconstruction of unseen data may imply that with the calcium carbide application, the seen and unseen NOC data have more similar distributions than is the case for the Tennessee Eastman process.[15]

PCA shows a dramatic increase in reconstruction ability as the number of model components increase. The ability of RF to reconstruct unseen NOC data initially increases with additional components, but then stabilizes and shows a decrease after 26 components, indicating overfitting past this point.

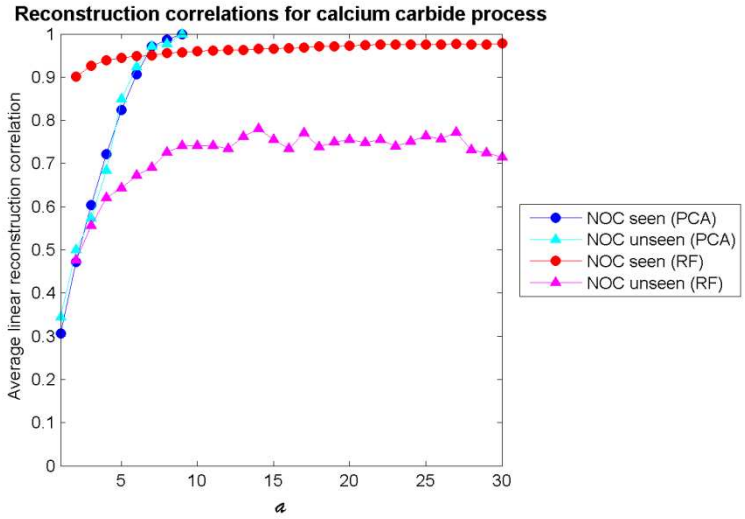

Figure 7.31: Average variable space reconstruction correlations for the calcium carbide process for a range of model components (calculated for training and valid NOC data)

7.3.3 Alarm rates

The false alarm rates on the unseen NOC data (before thresholds were adjusted for fair comparison between the PCA and RF techniques) are given in Figure 7.32. The PCA framework with 6 components gave a false alarm rate of 0, while the RF framework with 23 components gave a false alarm rate of 0.16. Again, the RF framework performed worse in terms of false alarm rates on unseen NOC data.

Compared to the false alarm rates found in the Tennessee Eastman application (Figure 7.13, where RF indicated all validation NOC data as faulty), the RF approach shows much better performance across a range of model components in not wrongly flagging unseen NOC data as faults. PCA still shows better false alarm rates than RF, over a range of model components.

[15] *This will be shown to be true in the Chapter 8.*

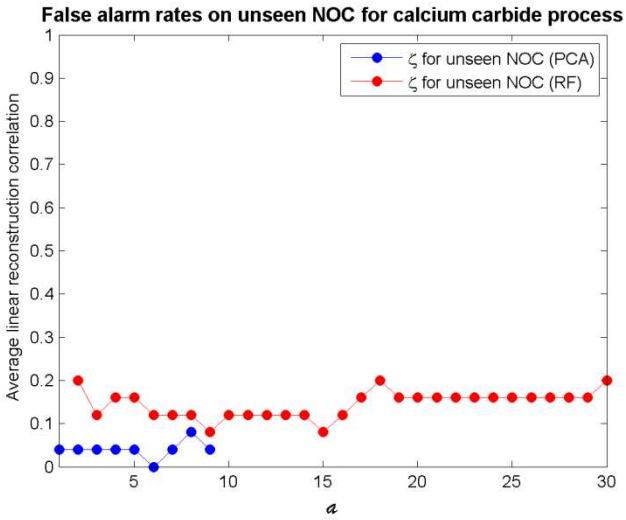

Figure 7.32: False alarm rates on unseen NOC data for the calcium carbide process for various number of model components

Better reconstruction ability, whether due to seen and unseen NOC data having similar distributions; features that capture the structure of the NOC data better; or better mapping and demapping models, imply lower squared prediction errors, and fewer detections (false alarms) in the residual space.

Once the confidence levels were adjusted to assure a maximum false alarm rate of 1% on the unseen data, the missing alarm rates were calculated for the fault data. As there is no consistent chronology in the fault data (due to the threshold method of dividing the process data into NOC and fault data), calculating detection delays are meaningless. Missing alarm rates for the selected number of model components (6 for PCA and 23 for RF) are given in Table 7.4.

Table 7.4: Score and residual distance based missing alarm rates for the calcium carbide process

	δ_s (PCA)	δ_r (PCA)	δ_s (RF)	δ_r (RF)
Fault	0.70	0.71	1	**0.62**

Inspecting the nature of the fault detections (Table 7.4), RF is the most successful at detecting the fault, with the detections made in the residual space (as with the Tennessee Eastman application). RF completely fails to detect any samples in the score space, while conversely showing improved performance in the residual space. This tendency for the RF framework to detect faults in the residual space has now been shown for the simple nonlinear system, the Tennessee Eastman process and this, the calcium carbide process data.

The influence of the number of model components a on missing alarm rates is depicted in Figure 7.33. A maximum of 9 components can be investigated for PCA (as the dimensionality of the system is 9) while up to 30 components were tested for RF.

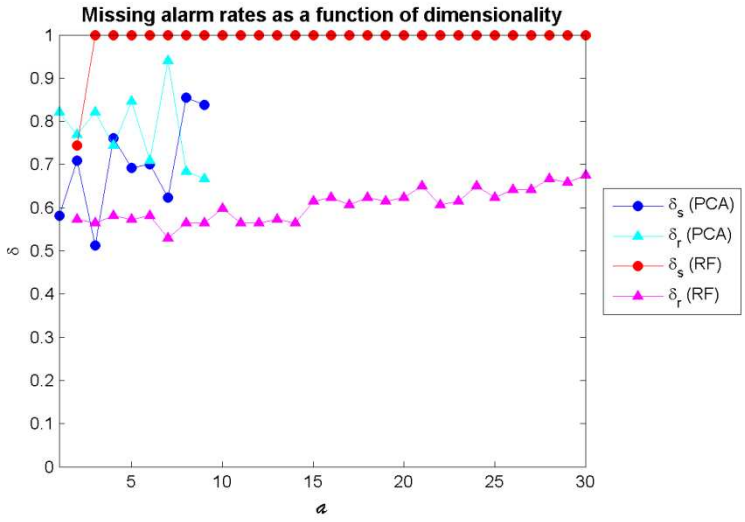

Figure 7.33: Missing alarm rates on fault data for the calcium carbide process for various numbers of model components

From Figure 7.33, the RF residual space missing alarm rate clearly outperforms the PCA diagnostics and RF score space missing alarm rates for the entire range of model components investigated, bar the PCA score diagnostic for three components. RF shows a score space missing alarm rate comparable to PCA when two model components are used. This suggests that the crossing criteria may not be the best criteria for selecting the number of model components to use for the RF framework.

7.3.4 Variable contributions

Variable contributions can be determined from the PCA model from score and residual distance contributions, while only residual distance contributions are available for the RF model. From process knowledge of the calcium carbide furnace (Jemwa & Aldrich, 2006; Aldrich & Reuter, 1999) it is known that high values for variables 1, 4, 5 and 6 ensure high product grade and tonnages, on which the definition of the fault conditions were based. Variables 2, 3 and 7 are weakly correlated with these product variables (not included in the process data), while variables 8 and 9 are considered to have a negligible influence on the performance of the furnace.

From Figure 7.34, PCA indicates variables 1, 2 and 4 contributing to the fault based on the score space; and variables 1, 2, 3, 8 and 9 involved in the fault based on the residual space. Based on the process knowledge given above, PCA identifies two variables correctly and one incorrectly in the score space, while identifying one correctly and four incorrectly in the residual space. The RF contribution plot (based on the residual space) is similar to the PCA residual space contributions, with two correct indications and three incorrect indications.

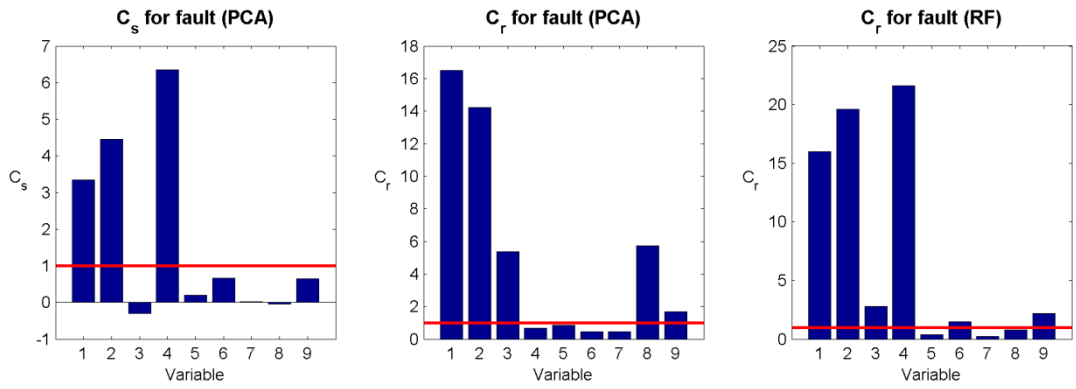

Figure 7.34: Scaled contribution plots for the fault of the calcium carbide process (99% percentile NOC contributions shown in red)

7.3.5 Discussion of calcium carbide process results

For the calcium carbide process, a confidence limit of 99% was selected, and the number of model components (features to extract) was determined as 6 for PCA (to account for 90% cumulative variance) and as 23 for RF (from the crossing criteria). As the original process data were only nine-dimensional, 23 RF features may seem too many to capture only informative NOC structure. However, RF features can be considered analogous to kernel features, which may capture significant variance in higher dimensionalities than the original data.

Using 23 RF features, the RF framework is able to detect more than 8% of the fault samples than PCA. Overall, independent of the number of model components used, the RF approach is as successful as, or better than, PCA on this real-world data set based on missing alarm rates.

As with the Tennessee Eastman process, the majority of the RF framework detections are in the residual space. This may indicated that the RF score space is not suitable for fault detections. This may be due to three reasons: the incorrect number of RF features are being used; even if an optimal number of features are used, the RF mapping functions are unable to accurately portray new data in the score space; or, capturing the nature of the NOC data structure successfully in a feature space is not related to the success of detecting new, unseen fault data in said feature space.

In terms of the first explanation, from this case study it seems that selecting a lower number of RF features would have given better detection performance. The crossing criteria may thus not be suitable for selecting the optimal number of features. To investigate the nature of the score space for a small number of model components, the next chapter will focus on investigating two-dimensional RF score spaces for the Tennessee Eastman and calcium carbide case studies. This investigation may also throw light on the second consideration, that of the suitability of the RF regression mapping functions. It was already noted in Chapter 5 that the RF mapping functions do not generalize well to unseen data regions.

The final consideration, that of the suitability of feature spaces based on NOC data for detecting new fault data, is not specific to RF fault diagnosis but is applicable to all feature extraction fault diagnosis schemes. The monitoring of the residual space should be sufficient to detect whether the structural tendencies exhibited by the NOC data, which shapes the feature space, are still valid for unseen fault data.

Both the PCA and RF contributions were not very successful in identifying the process variables that are deemed influential in this calcium carbide process fault. The extension of fault identification with RF will be taken up in Chapters 9 and 10, with the application of RF variable importance and partial dependence tools.

7.4 Fault diagnosis performance criteria

As mentioned in Chapter 1, an ideal fault diagnostic system should be able to rapidly detect, identify and explain faults, while being robust to system noise. Not only must a distinction be made between different failures, but the case of simultaneous faults should be distinguishable. Such a system should also be adaptable and able to identify novel faulty conditions. An accurate estimate of the probability of false and missed alarms should be made, with modelling and computational expenses kept to a minimum (Venkatasubramanian et al., 2003c).

From the results of the fault diagnosis applications in this chapter, the following conclusions are made for the developed random forest feature extractive fault diagnosis method, in light of these stated objectives.

- Short detection delays

In terms of rapid detection, the RF method showed shorter detection delays for the simple nonlinear system than the PCA approach. For the Tennessee Eastman process, the majority of faults had similar detection delays for both methods, with RF improving detection delay for one fault. RF fault diagnosis detection delays can then be considered on par with the PCA approach, and the first criterion satisfied.

- Accurate detection

The detection of faults by RF fault diagnosis is evaluated in terms of missing alarm rates. RF showed worse (by no more than 10 %) detection ability than PCA for the simple nonlinear system. For the Tennessee Eastman process, RF showed very similar detection ability to PCA for the Tennessee Eastman process, with improved detections for three faults. For the calcium carbide process, the RF framework showed comparable to better performance than PCA for a range of number of model components. The RF method thus exhibits the ability to detect faults, and detect some faults better than other methods, and the second criterion is satisfied.

- Fault explanation

Contribution plots can be considered as the fault explanation facility of a fault diagnostic scheme. RF fault identification was able to correctly identify the responsible variable for both faults of the simple nonlinear system, where PCA fault identification was unsuccessful. RF fault identification was also overall successful in the identification of three faults for the Tennessee Eastman process. Less success was shown for the calcium carbide process. The third criterion is mostly satisfied by the RF fault diagnosis method.

- Robustness to noise

The robustness of the RF fault diagnosis method to system noise was not explicitly tested for by increasing system noise and evaluating the change in performance of the RF method. No comment can thus be made in terms of the fourth criterion.

- Distinction between different faults

A distinction between different faults would be possible in the RF fault diagnosis framework by inspecting the contribution plots, and determining whether the same process variables were affected for different samples. This was indirectly shown when the contributions for five faults of the Tennessee Eastman process were investigated. The fifth criterion can be considered partially satisfied.

- Identification of different faults

When different fault conditions occur simultaneously, detection and identification of process variables involved should still be possible. This criterion is fairly open-ended, as the nature of the faults should be considered. Additive faults, which bolster each other in creating larger distances from normal operating conditions, can be detected with the RF fault diagnosis schemes. One would not necessarily be able to distinguish how many fault

conditions contributed to the detection, but from contribution plots one would be able to ascertain how many process variables showed significant deviation. RF fault diagnosis would then satisfy this sixth criterion for a certain class of faults.

- Adaptability

The RF fault diagnostic method, being based solely on normal operating conditions data and relying on a small number of model parameters, is easily adapted to changing conditions or new data sets. One simply retrains the RF algorithm on a new data set. However, recursive adaption such as in recursive principal component analysis, is not possible for the RF method as employed in this study. Still, RF fault diagnosis can thus be considered adaptable due to the ease of retraining and small number of model parameters required, satisfying the seventh criterion.

- Detection of novel faults

The detection of novel faults again depends on the nature of the faults considered. The RF fault diagnosis method is an unsupervised approach, relying only on normal operating conditions data. Any departure from these conditions will then be considered a fault. This would not be the case for supervised methods, where an algorithm is trained to recognize only a specified library of trained faults. Faults that do not deviate from the normal operating region, as specified by RF score and residual distances, will not be detected. In general, the eight criterion is satisfied, where the novelty of the fault is not at play, but rather the nature of the fault.

- Estimates of false and missing alarm rates

Accurate estimates of false and missing alarms are required for a good fault diagnostic method. The false alarm rate is theoretically specified by the significance level of the detection limits. The RF fault diagnostic method does not rely on distribution assumptions, but uses the percentile approach to determine detection limits. The only assumption made is that unseen normal operating conditions would exhibit the same, unknown and unspecified, distribution as the normal operating conditions on which training was conducted. The accuracy of the false alarm rate estimate is then linked to the representativeness of the training normal operating conditions data. As no fault data are used in training, estimates cannot be made as to the probability of missing alarm rates. This ninth criterion is then partially satisfied by the RF fault diagnosis algorithm.

- Modelling requirements

Another criterion is that of keeping modelling requirements to a minimum. Even though the RF diagnostic approach utilizes five feature extraction random forests, a mapping regression forests and m demapping regression forests, it has been shown in this study that these models are fairly robust to parameter selection, with default values being suitable for most applications. The automatic selection on one-class support vector machine parameters proved to be suitable in determining score distances and limits. Thus, many models are trained, but the input required by the operator is minimal. The tenth criterion of modelling requirements is thus satisfied by the RF fault diagnosis approach.

- Computational expense

The last criterion is in terms of computational expense is considered. Due to the training of a plethora of random forest classification and regression models, as well as the cross-validation exercise required for one-class support vector machine score distance modelling, the RF fault diagnostic scheme suggested in this study is not computationally cheap. RF model computational times were shown to be at least an order of magnitude larger than the PCA approach. However, comparison was not made to other computationally intensive feature extractive fault diagnosis methods such as kernel principal component analysis and kernel independent component analysis. Be that as it may, the RF approach is not considered computationally inexpensive, and thus is not considered to be successful on the twelfth and last criterion for fault diagnostic methods.

Overall, from the previous criteria, the random forest approach developed in this study can be considered a suitable option for fault diagnosis. Future investigation of robustness to noise, estimation of alarm rates and the reduction of computational expenses may further improve the application of random forest feature extraction to fault diagnosis.

7.5 Final comment

Returning to the horses for courses argument expounded on in the feature extraction validation study (Chapter 5), the lack of convincingly superior performance for the random forest fault diagnostic method on the three data sets investigated does not reject the possibility of useful application of this method. For data sets more suited to the random forest characteristics, better performance may be had. What these characteristics may be shall be further investigated in the next Chapter. What is clear is, that no matter the shortcomings of the feature space and the associated mapping / demapping models, the squared prediction error in the RF residual space is a powerful diagnostic for fault detection.

Nomenclature

a	feature space dimensionality
C	variable contributions
r	residual space indicator and residual space distance
s	score space indicator and score space distance
γ	detection delay
δ	missing alarm rate
ζ	false alarm rate
λ	global structure preservation measure

CHAPTER 8 - FAULT DIAGNOSIS: FEATURE SPACE EFFECT

Two-dimensional random forest feature spaces are investigated for the Tennessee Eastman and calcium carbide processes to determine the cause of the high missing alarm rates for the score space diagnostic found in Chapter 7. It is found that projections of fault data never exceed the range of the NOC features. The constrained response characteristic, an artefact of using regression forests for explicit mapping and demapping, is found to be responsible for both high score space missing alarm rates and low residual space missing alarm rates. It is thus concluded that the random forest fault diagnosis framework, as it was developed in Chapter 6, has a strong residual space diagnostic, under the assumption that a large and representative sample of NOC data is available for training.

The random forest feature space is investigated further by inspecting random forest features obtained from various configurations of multivariate Gaussian data. A deeper understanding is achieved of the nature of random forest proximities and feature spaces, with three common structures found: annulus-like shapes, round data clouds and horseshoes. The emergence of horseshoes ties in with results found in literature. From these results, a simple heuristic rule is suggested for selecting the number of random forest features to extract, given a random forest proximity matrix. If the assumption is further made that NOC data would consist of one continuous group of data, selecting two random forest features may be a good starting place for the random forest fault diagnosis framework.

8.1 Score space comparisons for process data

In order to gain further understanding as to the nature of especially score distance detections, the Tennessee Eastman and calcium carbide data sets are revisited, where the number of PCA and RF model components are fixed as two. This enables the visualization of the score space, which may lead to a better understanding as to the very low detection rates of faults based on RF score distances.

8.1.1 Tennessee Eastman process

From Figure 7.9 and Figure 7.10, the cumulative variance for two model components are 20% and 2.5%, for PCA and RF models, respectively. A confidence limit of 99% was chosen, with adjustment of the confidence limits based on the 99% percentile approach on unseen NOC data. The confidence limits of the PCA framework for training NOC data are based on assuming certain distributions for the NOC data, while adjusting both the PCA and RF limits on the validation data based on the percentile approach ensures a fair comparison.

Figure 8.1 shows the seen NOC and unseen NOC data with initial and corrected score distance confidence limits for the two component PCA model. The initial confidence limits are based on the assumption of Gaussian distributed data, hence the wide limits in comparison to the seen NOC data, with no data points lying outside the limits. This, in conjunction with the similarly assumed distribution-based residual diagnostic limits, may explain the low false alarm rates on unseen NOC data for the PCA approach in Chapter 7.

The projection of the unseen NOC data to the first two principal components shows that the unseen NOC data have a wider spread than the seen NOC data. This may support an earlier statement with regards to the inability of RF to generalize well to the unseen NOC data, as the training NOC data were not suitably representative of all NOC conditions. The corrected PCA score distance limit is wider than the original limit, while still excluding some of the unseen NOC samples. This is due to the fact that a distribution-based threshold is not used for the corrected limits, but rather the percentile approach. By definition, the percentile derived limits will exclude 1% of data for a 99% confidence level.

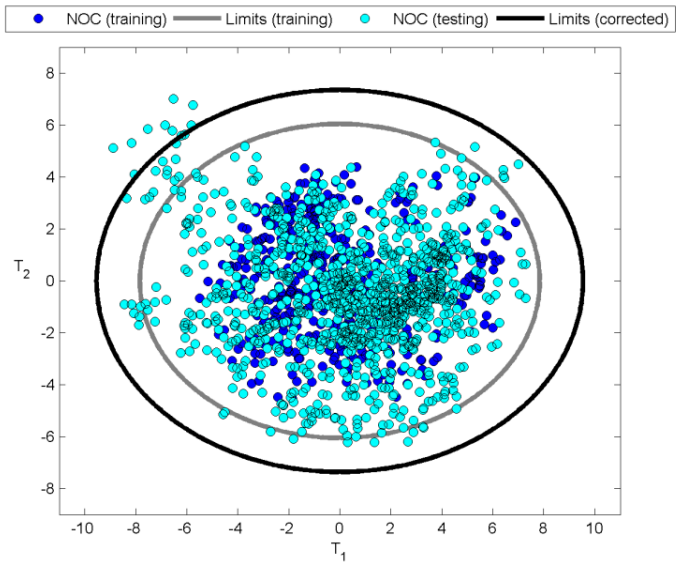

Figure 8.1: PCA NOC features with confidence limits from training and testing data for the Tennessee Eastman process

Figure 8.2 shows the seen NOC and unseen NOC data with initial and corrected score distance confidence limits for the two component RF model. The 1SVM-derived score distance limits can be seen to closely follow the shape of the seen NOC data, and excludes sparse regions within the seen NOC features range. This plot also validates the success of the heuristic cross-validation approach of selecting suitable 1SVM parameters to adequately represent the shape of the features without being too sensitive or noisy.

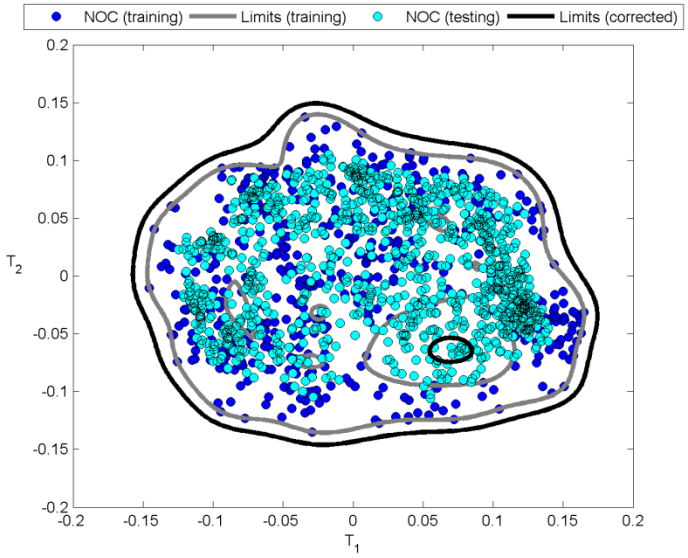

Figure 8.2: RF NOC features with confidence limits from training and testing data for the Tennessee Eastman process

The addition of the unseen NOC features shows an interesting phenomenon. The PCA scores showed that the unseen NOC data cover a wider range of PCA score values than the seen NOC PCA scores. With the RF scores, the unseen NOC scores do not cover a wider range, with some samples rather occupying a sparse region within the seen NOC score range. The adjustment to the RF score distance limit reduces the area of this no-longer sparse region, but also increases the limit outside the scores range. Since the unseen NOC samples are now

occupying a region which was sparse for the seen NOC samples, this suggests that the seen and unseen NOC data distributions are not similar, as was shown in linear projection of Figure 8.1.

In order to accommodate the additional distributional information of the unseen NOC data, the RF framework would require the recalculation of features and 1SVM probabilities, not merely the extension of the score distance threshold as shown in Figure 8.2.

Figure 8.3 and Figure 8.4 present the PCA scores for the fault data sets. Eight faults are difficult to detect due to fault scores lying primarily within the PCA score distance limits: faults 3, 4, 9, 11, 15, 16, 19 and 21. The other faults show large departures from the PCA score distance limits. This corresponds to the missing alarm rate accuracies based on score distances found in the previous Tennessee Eastman case study, and suggest that PCA scores can, for certain faults, be successfully utilised in fault detection.

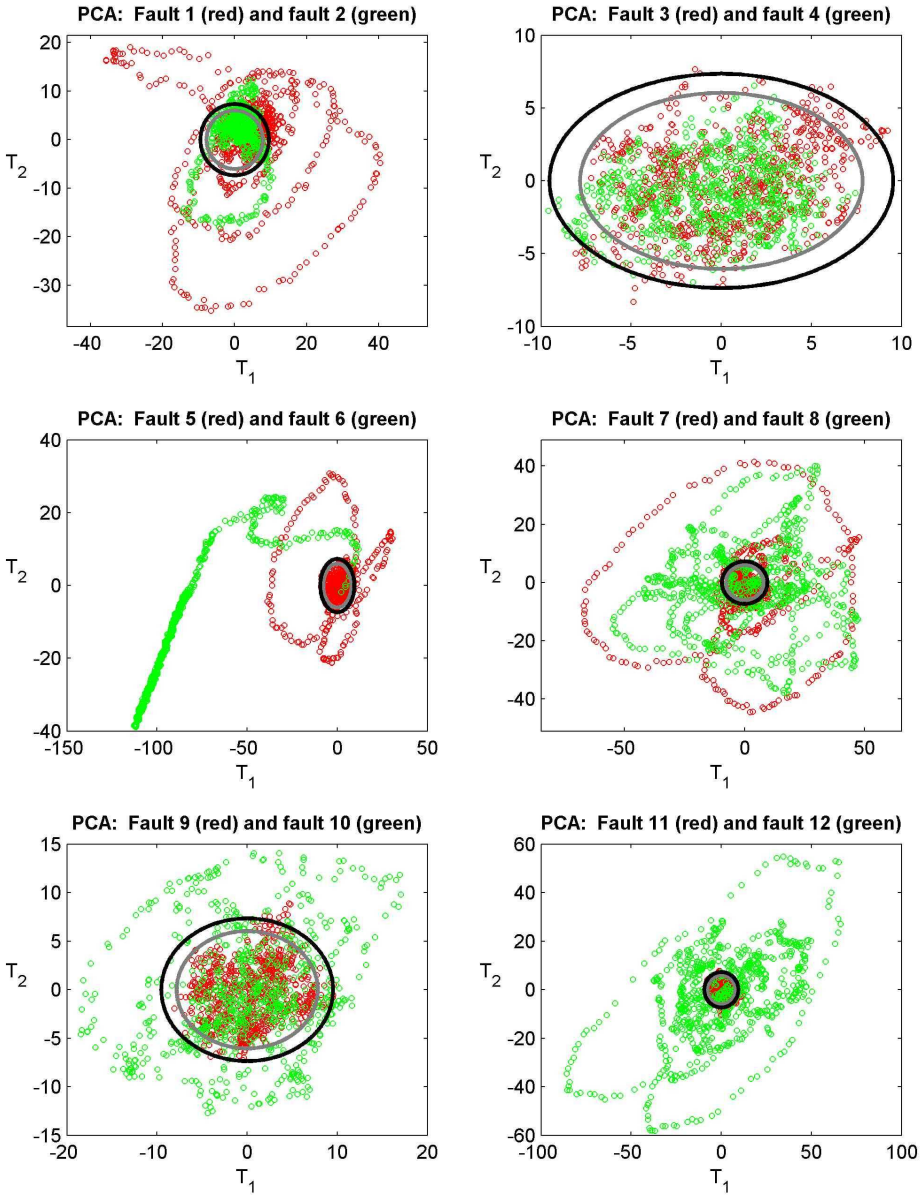

Figure 8.3: PCA features for faults 1 to 12 for the Tennessee Eastman process (confidence limits in grey based on training NOC and in black based on testing NOC)

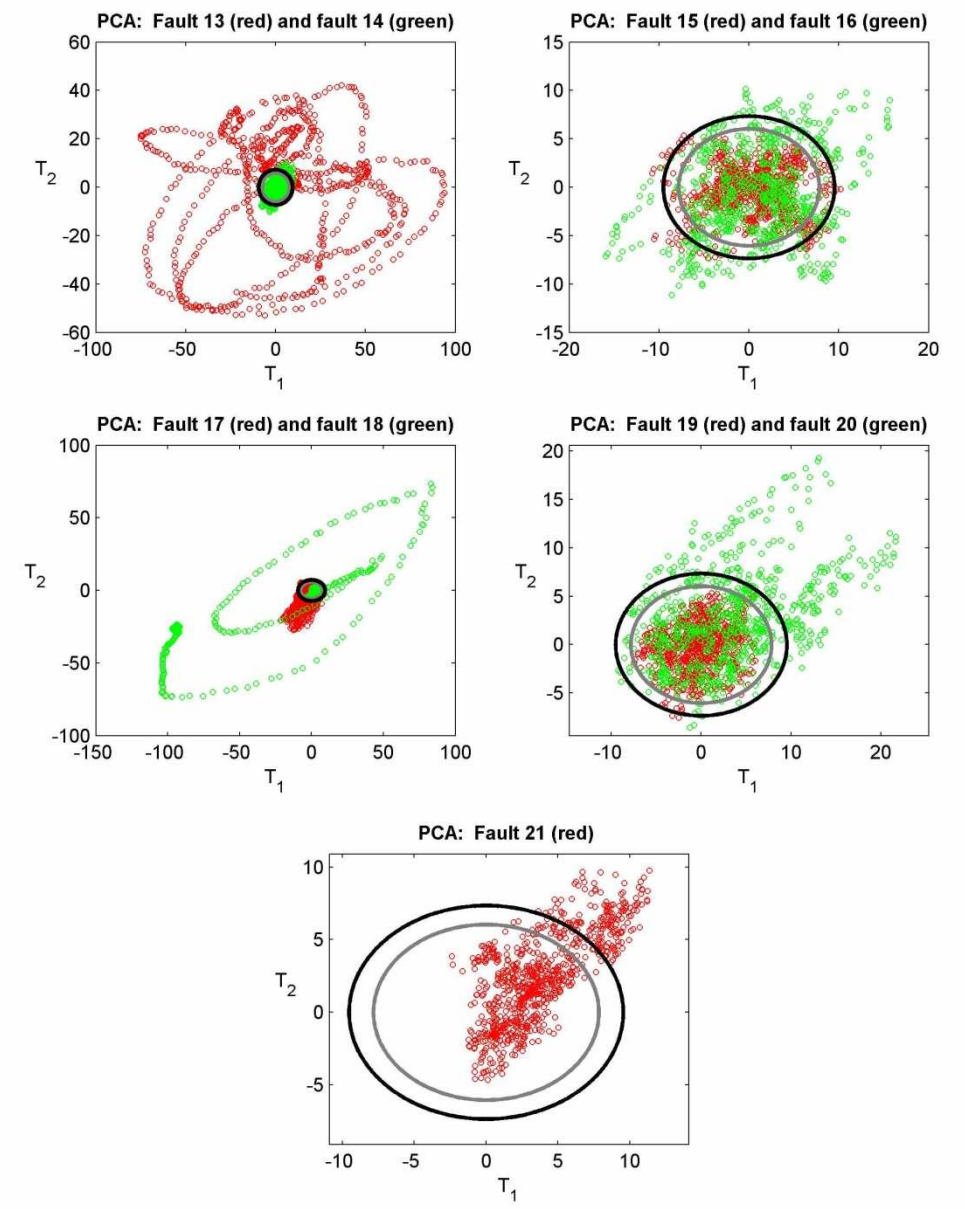

Figure 8.4: PCA features for faults 13 to 21 for the Tennessee Eastman process (confidence limits in grey based on training NOC and in black based on testing NOC)

Figure 8.5 and Figure 8.6 present the RF scores for the fault data sets. For none of the 21 faults do the fault scores exceed the outer RF score limits. From visual inspection, only faults 1 and 21 appear to have a majority of fault scores within the interior sparse region of the initial NOC limits. However, these limits were updated based on the unseen NOC data, resulting in a much smaller interior fault region, and subsequently much reduced detection.

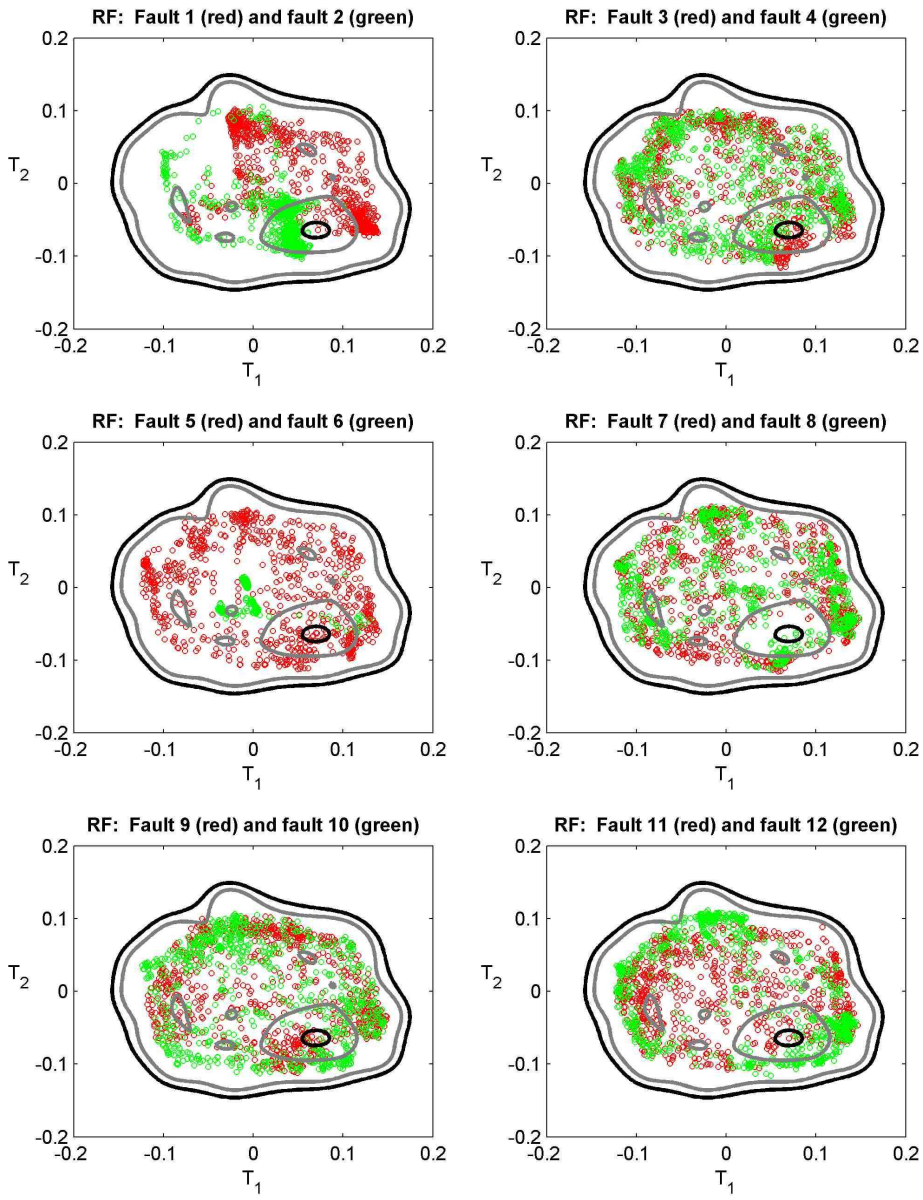

Figure 8.5: RF features for faults 1 to 12 for the Tennessee Eastman process (confidence limits in grey based on training NOC and in black based on testing NOC)

From these RF score plots, it is then apparent why score based detection is ineffective for the Tennessee Eastman problem: fault scores are generally indistinguishable from the NOC region specified by the 1SVM limits. The nature of the score limit definition (1SVM-based in this study) does not seem to be responsible for this low score detection ability, as the fault scores generally overlaps with the NOC scores, irrespective of defined limits. The random forest features or random forest feature mapping must then be to blame for poor score-based detection performance.

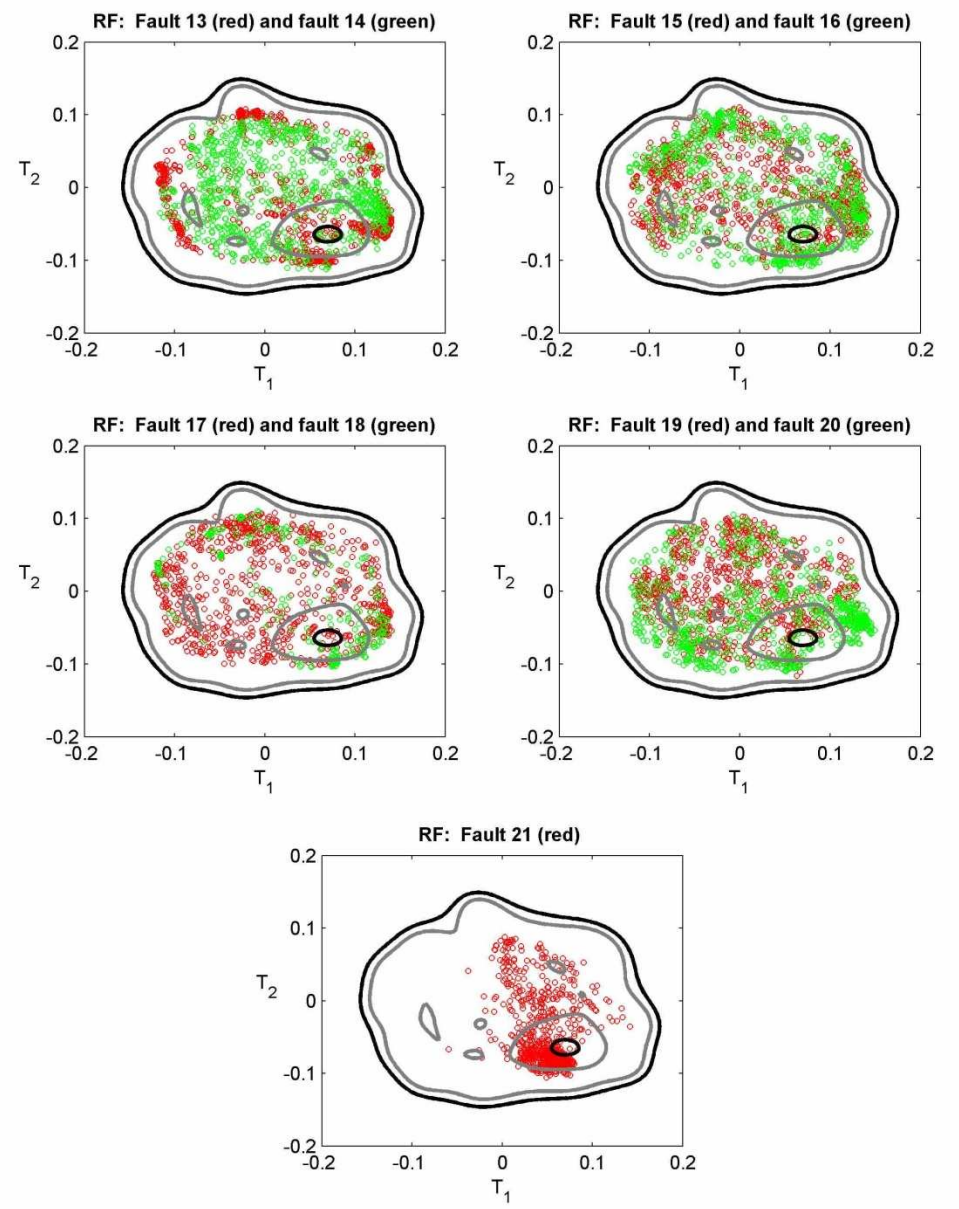

Figure 8.6: RF features for faults 13 to 21 for the Tennessee Eastman process (confidence limits in grey based on training NOC and in black based on testing NOC)

8.1.2 Calcium carbide process: comparison with two components

From Figure 7.29 and Figure 7.30, the cumulative variances for two model components are less than 50% and 20%, for PCA and RF models, respectively. An initial confidence limit of 99% was chosen, adjusted to a maximum false alarm rate of 1% on unseen NOC data. The confidence limits of the PCA framework for training NOC data are based on assuming certain distributions for the NOC data, while adjusting both the PCA and RF limits on the validation data based on the percentile approach ensures a fair comparison.

Figure 8.7 shows the seen NOC, unseen NOC and fault data with initial and corrected score distance confidence limits for the two component PCA model. The initial confidence limits are based on the assumption of Gaussian distributed data, hence the wide limits in comparison to the seen NOC data, with no data points lying outside the limits. In contrast, a 99% percentile limit based on the unseen NOC data would have result in a smaller confidence area, as shown. The larger of the two confidence regions were kept to ensure that false alarms would not be made on the seen NOC. As opposed to the Tennessee Eastman NOC data discussed before, the unseen NOC data do not have larger extents in the score space than the seen NOC data. This

implies that the distributions of the seen and unseen NOC data are similar, at least more so than for the Tennessee Eastman case.

The projected fault scores lie within and without the NOC confidence region. This agrees with the observed missing alarm rates obtained in Chapter 7, as well as the previously stated observation that the nature of the definition of the calcium carbide fault data (a threshold on a key performance indicator) implies that NOC and fault samples may overlap in the process variable space, due to noise and process lags. I.e. it is known that the fault data is not distinctly separated from the NOC data in the process variable space.

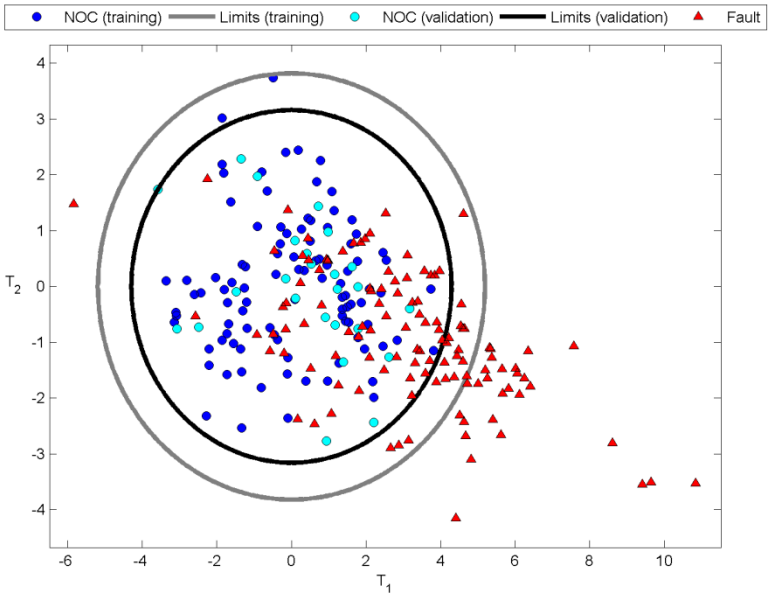

Figure 8.7: PCA features for NOC and fault data for the calcium carbide process (confidence limits in grey based on training NOC)

Figure 8.8 shows the seen NOC, unseen NOC and fault data with initial and corrected score distance confidence limits for the two component RF model. The 1SVM-derived score distance limits can be seen to closely follow the shape of the seen NOC data, and excludes sparse regions within the seen NOC features range.

The addition of the unseen NOC scores reiterates the observation from Figure 8.7: the seen and unseen NOC data for the calcium carbide process appear to have similar distributions in the feature space (unlike the dissimilar seen and unseen NOC distributions of the Tennessee Eastman data). The score space threshold was increased to incorporate the unseen NOC outlier in the region of (0.2, 0.1) in the score space. The corrected threshold maintained the character of the seen NOC data, with the most of the sparse region in the centre remaining outside the confidence limits.

The addition of the fault scores shows overlap with the NOC confidence region (as with the PCA approach), while detections are made in the sparse centre region. Detecting fault samples in the RF feature space is then shown to be possible, given the right number of features and consistent, representative NOC training data.

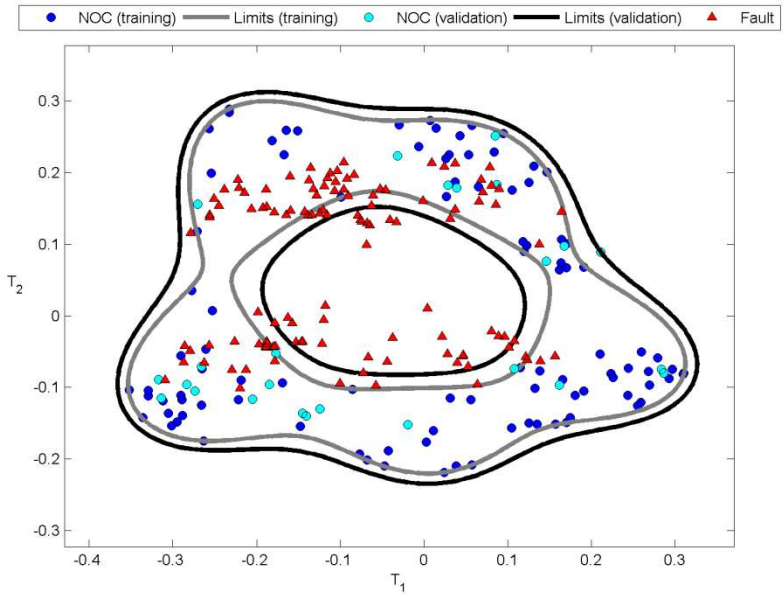

Figure 8.8: RF features for NOC and fault data for the calcium carbide process (confidence limits in grey based on training NOC)

As with all the fault projections of the Tennessee Eastman process (Figure 8.5 and Figure 8.6), the calcium carbide fault scores do not exceed the score ranges of the training NOC scores. This peculiar characteristic is investigated in the next section.

8.2 Constrained score spaces and random forest generalization

To investigate the apparent constrained behaviour of RF fault diagnosis framework mapping technique, a two-dimensional uniform ring-shaped data set was generated as NOC data. RF and PCA models with two components were trained on this NOC data. The NOC scores and 95% confidence limits for these two methods are shown in Figure 8.9 and Figure 8.10.

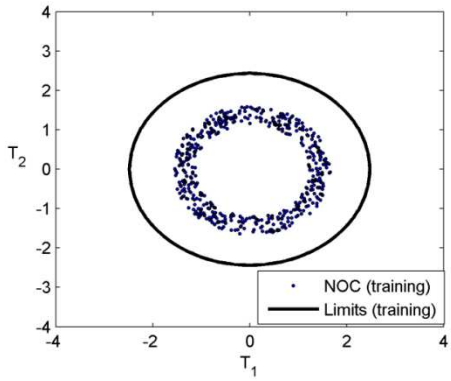

Figure 8.9: PCA scores of simulated NOC data and PCA 95% confidence limits based on Gaussian distribution assumption

PCA is a linear feature extraction method, as can be seen from the retention of the shape of the uniform ring in the PCA feature space. The PCA confidence limits are some distance from the NOC data, due to the assumption of Gaussian distribution of the scores.

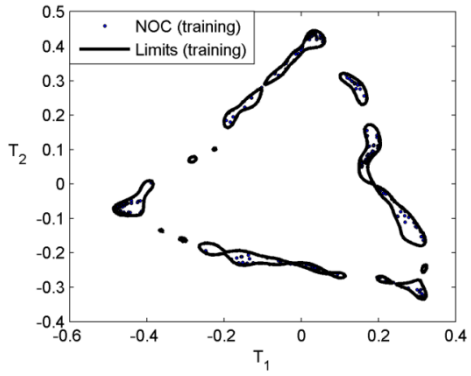

Figure 8.10: RF scores of simulated NOC data and RF 95 % confidence limits based on 1SVM probabilities

The RF scores do not retain the exact shape of the uniform ring in RF feature space. Whether the distorted annular shape of the NOC scores correspond to the annularity of the uniform ring in the original input space is not apparent. The 1SVM confidence limits follow the NOC scores closely, possibly in too disjointed a fashion.

Figure 8.11 and Figure 8.12 present simulated fault conditions and their associated PCA and RF fault scores. Given the Gaussian distribution based confidence limits, the PCA model would only be able to detect a portion of the third fault. However, if 1SVM limits were employed, more fault samples would be detected due to the "data wrapping" nature of the 1SVM limits.

The majority RF fault scores fall outside the 1SVM limits for fault 1, suggesting a low (good) missing alarm rate. RF fault scores for fault 2 overlap to a larger extent with the NOC confidence region, but should still show a low missing alarm rate. Fault 3, with input variable values beyond the range of the NOC data input variables, exhibits interesting RF scores. Fault 3 scores show the greatest overlap with the NOC confidence region. As with the previous studies on two-dimensional Tennessee Eastman and calcium carbide process features, the fault scores do not show values that exceed the range of the NOC features.

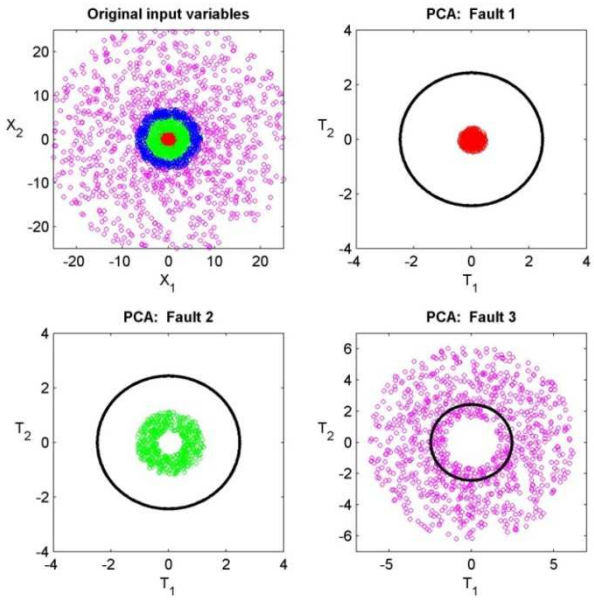

Figure 8.11: Input variables and PCA fault scores for various faults for the simulated data set (blue ring represents NOC data)

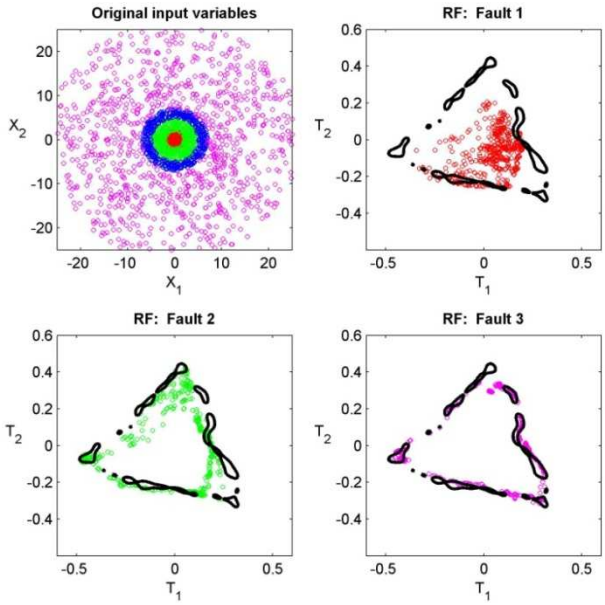

Figure 8.12: Input variables and RF fault scores for various faults of the simulated data set (blue ring represents NOC data)

A possible reason for the constrained score space described in the previous two studies is presented here: Once the features have been extracted with unsupervised random forests, a regression random forest is trained to predict each feature, independent of the other features, based on only the input variables. The training data set for these mapping forests consists of only the training data for the unsupervised random forest, and only for the original unlabelled data, i.e. excluding the synthetic contrast class.

A regression decision tree creates local hyperrectangular neighbourhoods in the input space, and fits the average of responses in a particular neighbourhood as the prediction of that neighbourhood. The possible predictions for a decision tree can then only be the average of some subsample of the data, i.e. nothing greater than the largest response or smaller than the smallest response.

When a decision tree is faced with a previously unseen input vector, with input variable values outside the range of the variable values of the training input vectors, generalization to this unseen data is very simple, and possibly restrictively so. A local neighbourhood is defined in terms of split positions on input variables. The neighbourhoods lying on the outer ranges of the training input space will then be responsible for out-of-range input predictions. However, these predictions are based only on the average of responses of training inputs in that local region. If the real underlying response function is not the same constant as in the outer neighbourhoods, generalization will be poor. If the real underlying response function is monotonically increasing or decreasing outside the training region, the decision tree generalization will be increasingly poor with distance from the original training region.

This poor generalization in terms of extrapolation for decision trees extend to random forests, as the regression response is the average of the decision tree predictions. The average of average responses can still not be larger than the maximum response in the training data or smaller than the minimum response in the training data. Random forest regression will then also show this poor extrapolation to input data outside the range of the training input data.

The concept of constrained responses can be extended to demapping regression forests as well. Demapping regression forests (for fault diagnosis applications) are trained with only NOC features as input, and the different NOC input variables (one forest for each variable) as response. The possible response values are

again limited to the minima and maxima values present in the NOC training data. As enumerated on before, fault scores will necessarily lie within the minima and maxima of the NOC scores. Poor generalization for extrapolation, therefore, is not at play here. However, the reconstructed fault input variables can, due to the constrained responses concept, only assume values within the limits of the NOC variable values. If the fault variables are outside the range of the NOC variables, the reconstruction will not be accurate, resulting in large residual distances. This may indicate poor generalization in terms of interpolation.

In effect, the constrained responses characteristic and associated poor generalization (for extrapolation and interpolation) of random forest regression enables the detection of fault conditions based on residual distances. To ensure confidence in the ability of random forest fault diagnosis to not indicate false alarms on normal operating conditions, a large, representative training data set characterizing normal operating conditions is vital.

To improve the generalization of mapping to and from the RF feature space, other mapping / demapping techniques may be considered, e.g. feed-forward multilayer perceptrons or other regression models, or even robust triangulation from training NOC dissimilarities[16].

8.3 Interpreting random forest feature space projections

In order to better understand the nature of the random forest feature space, an investigation of the random forest features of simple multivariate Gaussian data will now be considered. To this end, two-dimensional data sets were generated by randomly sampling from a multivariate Gaussian distribution, with zero means and specified crosscorrelation values. Random forest features were extracted from these data sets, based on the average proximities of five unsupervised forests. The first two RF features are inspected for each data set. Figure 8.13 presents the 9 data sets investigated. Three sample sizes (N) of 50, 100 and 500 were used, with crosscorrelations (ρ) of 0.3, 0.6 and 0.9. Figure 8.14 presents the features extracted from these 9 data sets.

From Figure 8.14, a number of interesting effects can be seen. The reason for these effects may stem from the synthetic contrast that is added to the data to allow the training of a classification forest for feature extraction, as well as the nature of the decision tree nodes.[17]

[16] *The application of neural networks and triangulation for explicit mapping of Sammon features (see Chapter 5) is an example of using these techniques (De Ridder & Duin, 1997) .*

[17] *Note: The random forest proximities are calculated only for the original data.*

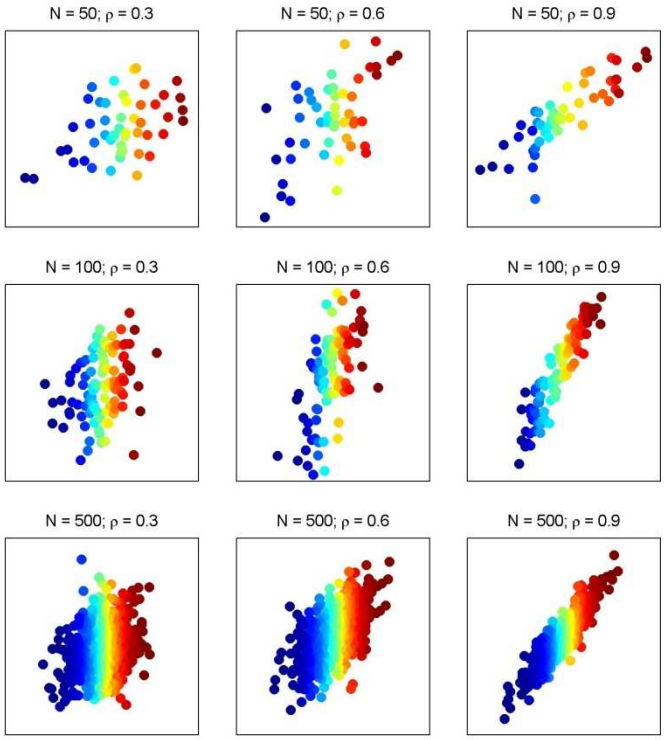

Figure 8.13: Multivariate Gaussian distributions investigated, with the abscissa axes X_1 and the ordinate axes X_2. Colouring is added to indicate the ordering of the samples. Crosscorrelation increases from left to right; sample size increases from top to bottom.

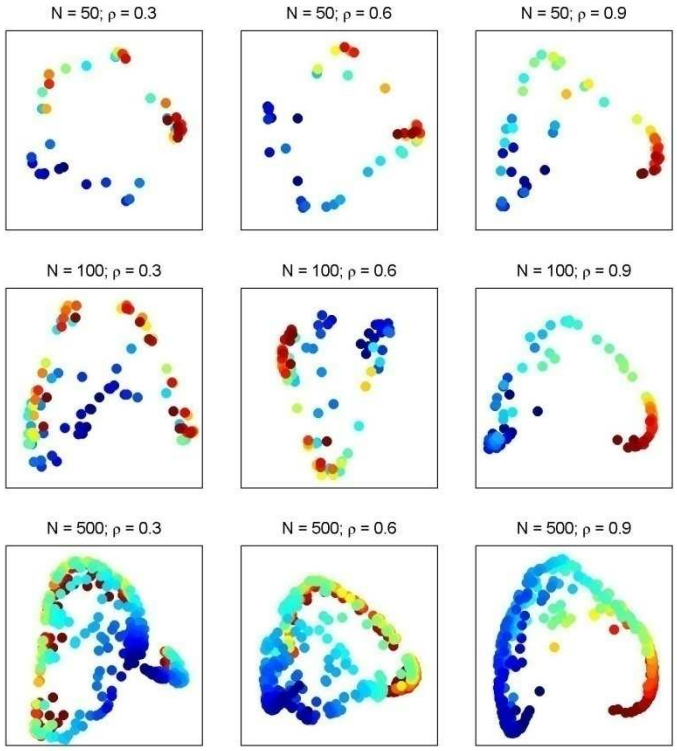

Figure 8.14: Random forest features extracted from multivariate Gaussian distributions investigated, with the abscissa axes T_1 and the ordinate axes T_2. Colouring is added to indicate the original ordering of the samples in X. Crosscorrelation increases from left to right; sample size increases from top to bottom.

To recap: the synthetic class is the product of the marginal distributions of the original data, with the same number of samples as the original data (Chapter 3). The product of the marginal distributions is in effect each variable permuted, independent of other variables. If the original data structure is a multivariate Gaussian distribution with specific crosscorrelations, the synthetic data structure will be a multivariate Gaussian distribution with close to zero crosscorrelations, as per-variable permutation destroys the correlation between variables. In two dimensions, this synthetic class has the appearance of a "round cloud" of samples, with increasing sample density of points towards the middle.

When decision trees are trained to classify the original and synthetic data, the size and stability of leaf nodes will depend on amongst others, two things: the number of samples in the training data sets, and the overlap of the original data and the synthetic data. The fewer the samples, the lower the sample density, the larger the leaf nodes will be. And, the higher the overlap of the original and synthetic data, the smaller the leaf nodes will be, as leaf nodes attempt to span "pure" subspaces. When trees are grown from different bootstrapped samples of the same training data set, leaf nodes will differ. For small sample sizes, bootstrapped aggregating has a large effect on the leaf node subspaces. By removing a few samples from the training data set for a new tree, the subspace partitioning may look very different from one tree to the next. For large sample sizes, this instability of leaf nodes is not as severe, as the distribution of training data is less sparse. A different bootstrapped training data set will induce only a slightly different subspace partitioning.

For low samples sizes and low crosscorrelation, the random forest feature space shows an annulus-like shape (Figure 8.14; first row, first column). For data with low crosscorrelation, the shape of the synthetic class will be very similar to the shape of the original data, giving a large degree of overlap and no large "pure spaces" containing only the original data. However, because of the low sample sizes, leaf nodes will be somewhat unstable. When calculating the original data proximity from these trees, it will be found that each sample will have a small neighbourhood of samples it is proximal to, due to the unstable leaf node spans. The proximity matrix will then show each sample has high similarity to a small number of other samples, and high dissimilarity to all other samples. When extracting multidimensional scaling coordinates, the result is as given in the first row, first column of Figure 8.14: points lying on a ring. The ring shape is due to the repulsive forces of many dissimilar samples (Van der Maaten & Hinton, 2008). This ring shape is also similar to the two-dimensional random forest feature space of the calcium carbide NOC data (Figure 8.8).

For larger sample sizes and low crosscorrelation, the annulus-shape disappears, with the random forest feature space rather resembling a round cloud of data (Figure 8.14; third row, first column). Again, due to low crosscorrelation, there will be great overlap between the original data and synthetic data. Now, however, the sample size is large, suggesting that the leaf nodes will be smaller than for small sample sizes. Also, bootstrapping will have a less destabilizing effect on the final leaf node spans. Less of the original data samples will then end up in the same leaf nodes. The proximity matrix will then tend to have relatively constant, low values. When multidimensional scaling is applied to a dissimilarity matrix of almost equal dissimilarities, an interesting effect occurs: the two-dimensional feature space will show samples lying on concentric circles, or a "round cloud" (Borg & Groenen, 2005). This round cloud is similar to the two-dimensional random forest feature space of the Tennessee Eastman process NOC data (Figure 8.2).

The most interesting phenomenon of Figure 8.14 is the occurrence of horseshoes at high correlations (last columns). At high correlations, there will be less overlap between the linear-like clusters of the original data (Figure 8.13, last column), and the round cloud of the synthetic data. In particular, the ends of the linear-like clusters (dark blue and dark red regions of Figure 8.13 and Figure 8.14) will be free of overlap from the synthetic data. The ends of the linear-like clusters will result in larger leaf nodes, incorporating more of the original data samples into these nodes. The effective neighbourhood of the samples at the ends of the linear-like clusters are thus larger than the effective neighbourhoods of the samples in the middle of the linear-like

clusters. The proximity matrix will now show a latent ordering: dark blue samples similar to dark blue samples, dark blue samples less similar to green samples, dark blue samples completely dissimilar to red samples. It has been shown than when multidimensional scaling is applied to a dissimilarity matrix that exhibits latent ordering, horseshoe shapes arise: *"In general, a latent ordering of the data gives rise to these patterns [horseshoes] when one only has local [own emphasis] information... ...that is when only the interpoint distances for nearby points are known accurately"* (Diaconis et al., 2008). The random forest proximity measure is a local method of assigning similarities, as similarities are increased only if sample pairs report to the same leaf nodes. The local nature of the random forest proximity measures suggests that these findings for simple multivariate Gaussian data may be extendable to more complex nonlinear data.

To investigate the effect that the input data dimensionality has on the random forest feature space, simulated multivariate Gaussian data sets were again generated, consisting of 500 samples, crosscorrelations of 0.6, and dimensionalities of 2, 10 and 30. The extracted features are shown in Figure 8.15.

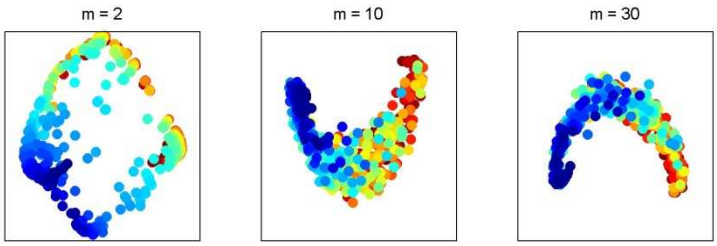

Figure 8.15: Random forest features extracted from multivariate Gaussian distributions investigated in terms of the effect of dimensionality, with the abscissa axes T_1 and the ordinate axes T_2. Colouring is added to indicate the original ordering of the samples in X. Dimensionality of X increases from left to right.

From Figure 8.15, as the dimensionality increases, the horseshoe shape becomes more pronounced. This can be explained from the increasing sparsity of data as dimensionality increases. The higher the dimensionality, the further the ends of the linear-like cluster of the original data will "stick out" from the "round cloud" of the synthetic data, increasing the effect of latent ordering discussed before.

Another interesting aspect that Diaconis et al. (2008) discussed with regards to horseshoes in multidimensional scaling is the appearance of multiple horseshoes when there are more than one group with latent ordering in the data set. Diaconis and co-workers showed that data that exhibit a partitioned proximity matrix (with two groups of samples having high intra-cluster similarities and low inter-cluster similarities) will result in two horseshoes in a three-dimensional multidimensional scaling space (Diaconis et al., 2008).

From these preliminary findings, a heuristic rule for determining the number of random forest features to extract presents itself: by inspecting the proximity matrix to determine the number of clusters, the minimum number of features to show the horseshoe groups can be found. One group with latent ordering can be visualized in two dimensions; two groups with latent ordering can be visualized in three dimensions, etc. The ordering of the proximity matrices to show clusters [e.g. reordering proximity data for block diagonalizing and using cluster count algorithms (Sledge et al., 2009)] is the topic of much research that could be applied to this heuristic rule. An example of two clusters apparent from the random forest proximity matrix (and three random forest features) is given in Figure 8.16. From Figure 8.16, two groups are apparent from the random forest proximity matrix. The feature plot also show two groups, albeit rather noisy horseshoes.

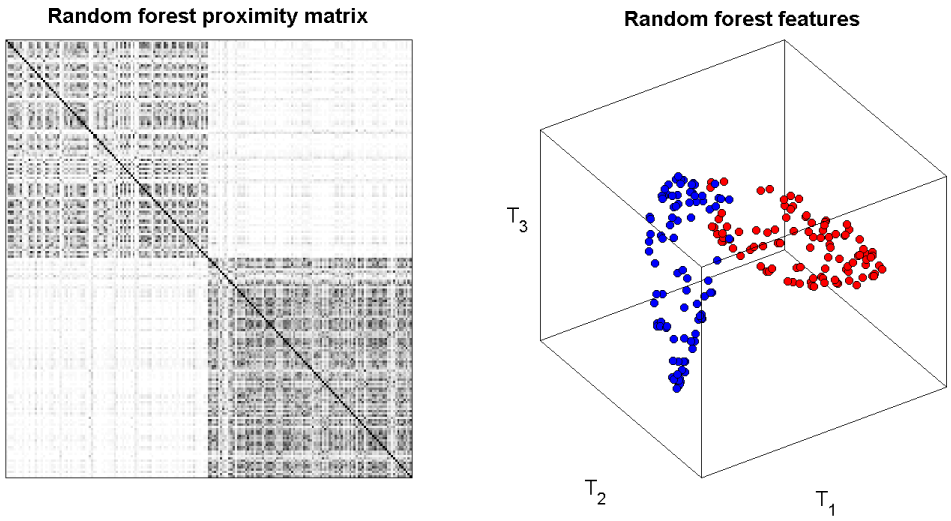

Figure 8.16: Random forest proximity matrix and three random forest features for two classes of the PGM flotation data (datapgm, see Chapter 5). Proximities are shaded from white (zero proximity) to black (proximity of one).

Where normal operating conditions data will normally consist of only one group (except where the process operates in several states), two random forest features should be representative of its structure. From the three types of structures found in Figure 8.14, two random forest features capture their structure: For the case where proximities show small neighbourhoods of communal similarity, an annulus-like shape is found (these proximities would result on samples on the outer edge of a sphere for three features, and samples on the outer edge of a hypersphere for more features). For the case where proximities are mostly equal, a round cloud is found (for more features, this would be a hypercloud). For the case of one group with latent ordering based on locally accurate similarities, a horseshoe is found (which remains a horseshoe in higher dimensions). From this, selecting two random forest features initially may be a good strategy for the random forest fault diagnosis framework developed in Chapter 6.

Using two random forest features would have given an improved missing alarm rate for the calcium carbide process application discussed in Chapter 7, without reducing residual space detections. No advantage, but also no disadvantage, in terms of alarm rates would have resulted for the simple nonlinear system or Tennessee Eastman process had two random forest features been used (Chapter 7). Fewer random forest features would reduce the computational expenses for the random forest fault diagnosis framework. Another consideration is the curse of dimensionality and its effect on the 1SVM score distance calculations. As more features are used, the dimensionality of the feature space increases, and data points are increasingly sparse. The 1SVM estimation of data density may then lead to overfitting around local, small populations of data points, causing disjointed confidence regions for normal operating conditions. Overfitting of the normal operating conditions confidence regions would then induce high false alarm rates.

Verily, when it comes to random forest features, less is more.

Nomenclature

a	feature space dimensionality
m	input space dimensionality
N	sample size
ρ	crosscorrelation

CHAPTER 9 - FAULT IDENTIFICATION: VARIABLE IMPORTANCE

This chapter investigates the application of random forest variable importance as a fault identification scheme. To this end, supervised random forests can be trained once normal operating conditions and fault conditions have been identified by a fault detection technique. The level of association of each process variable with the fault condition (its variable importance) can be determined from this supervised random forest by calculating the degradation in performance when each process variable is permuted.

Additional tree ensembles employing the same variable importance scheme as random forests are introduced. These methods are applied to validation case studies, to illuminate the effect of correlation on the variable importance measures. Finally, these tree ensembles are applied to a fault identification problem on several Tennessee Eastman faults. From these studies, it is shown that correlation has a significant effect on variable importance measures. From this, process expert intervention (whether at the data preprocessing or interpretation stages) is essential to differentiate causation, meaningful association and correlation.

It is concluded that random forests with random split selection can provide informative, computationally efficient indicators that can aid fault identification.

9.1 Overview

The random forest fault diagnosis framework developed in Chapter 6 only employs contribution charts from the residual space for fault identification purposes. As shown in Chapters 3 and 4, random forests can provide variable importance measures, a potential tool for indicating variables associated with a detected fault.

The unsupervised problem of fault detection can be transformed to a supervised problem once samples have been classified as faulty. Random forest variable importance, an interpretive tool for supervised applications, is then a viable option for fault identification. It is proposed that by calculating random forest variable importance, the process variables that were involved in the transition from normal operating conditions to fault conditions can be identified. Whether this involvement is due to correlation or causation requires the input of process experts. For now, the purpose of random forest variable importance is to provide some form of ranked importance scores for process variables, based on their involvement in a fault condition.

In the work presented in this chapter, random forest variable importance measures for fault identification application are introduced and validated on simulated case studies and the Tennessee Eastman process. The random forest approach is also compared to other tree ensemble techniques (conditional inference forests and boosted trees). In Chapter 10, random forest variable importance will be further exploited in fault identification and interpretation, with applications to the Tennessee Eastman and calcium carbide processes.

9.2 Additional tree ensemble methods

Classification modelling and variable importance measures based on ensembles of tree models have become popular in a number of fields, including species distribution for future climate scenarios (Garzón et al., 2006; Cutler & Stevens, 2006; Peters et al., 2007); predicting chemical properties such as reactivity and biological activity from molecular structure (Svetnik et al., 2003; Svetnik et al., 2005; Gupta et al., 2006; Sakiyama et al., 2008); chemometrics data (Zhang et al., 2005); as well as microarray and DNA sequence data (Cummings & Segal, 2004; Diaz-Uriarte & Alvarez de Andres, 2006; Pang et al., 2006).

Conditional inference forests, an extension to random forests, and boosted trees are briefly introduced here.

9.2.1 Conditional inference forests

CART trees select optimal splits based on the Gini impurity measure (Chapter 3). However, it has been shown that this measure is biased (Breiman et al., 1993), favouring splits on variables with more categories or distinct numerical values. To address bias in decision trees, recursive partitioning in a conditional inference framework has been suggested (Hothorn et al., 2006). For each split, the significance of a global null hypothesis (that the response is independent of all input variables) is determined through permutation-based multiple significance testing. If the global null hypothesis is supported, no further partitioning occurs. If the global null hypothesis is rejected, the optimal splitting variable is selected from the permutation test results. The optimal splitting position is determined by calculating permutation test statistics for all possible binary splits on the optimal splitting variable, and selecting the split that maximizes this statistic.

A conditional inference forest (CF) is an ensemble of conditional inference trees, where each tree is trained on a bootstrapped learning sample, subject to random split selection. As with random forests, the final predictor is the majority vote or average over all trees.

9.2.2 Boosted trees

While random and conditional inference forests aggregate parallel models, boosting involves sequentially learning models, modifying successive training data sets to place emphasis on misclassified examples, and finally combining all models weighted according to their accuracies (Hastie et al., 2009). The boosted trees (BT) technique also uses the CART method, but the method is applied sequentially, with stochastic gradient boosting to decrease the overall loss function. The details of Friedman's gradient boosting algorithm (based on a Bernoulli loss function for classification) are given in these references: (Friedman, 2001; Ridgeway, 2007b).

9.3 Tree ensemble variable importance

Although they provide a powerful framework for capturing nonlinear relationships between variables, ensembles of tree models are black box models that cannot be interpreted directly. However, tree ensemble methods can be used to generate variable importance measures that give a ranked indication of the relative significance of input variables to the classification or regression problem at hand.

Embedded estimates of variable importance can be made from model parameters, for example investigating the connection weights in artificial neural networks (Olden & Jackson, 2002) or using perturbation analysis with other black box models. For tree methods, a variety of internal estimates exist. Variable importance can be derived from the number of times a variable is selected as a split variable; the summation over all nodes of impurity decreases for relevant split variables; or the summation over all nodes of impurity decreases for all input variables (Breiman et al., 1993; Hastie et al., 2009)**.** The variable importance measure for variable j $\varepsilon_j(\mathcal{T}_L)$ in tree $\mathcal{T}_L$ can be expressed in terms of the decrease in node impurity from parent node to children nodes Δi, for all non-terminal nodes J:

$$\varepsilon_j(\mathcal{T}_{\mathbf{L}}) = \sum_{J \in \mathcal{T}_L} \Delta i\,(J) \qquad \textit{Eqn. 27}$$

These variable importance measures can be expanded to ensembles of trees by averaging individual tree importance measures:

$$\varepsilon_j = \frac{1}{K}\sum_{k=1}^{K} \varepsilon_j\left(\mathcal{T}_{\mathbf{L}(\boldsymbol{\theta}_k)}\right) \qquad \textit{Eqn. 28}$$

Wrapper estimates of variable importance involve the modification of the input space and subsequent retraining of models to evaluate the change in model accuracy. One such an approach is leave-one-out variable importance. This approach compares the accuracy of a model trained using all predictors to the accuracies of models trained leaving out one predictor each time. The rationale behind this is that a strongly relevant predictor cannot be left out of a model without loss in accuracy. For induction models, the optimal predictors may not necessarily reflect the relevance of predictors, but be related to the nature of the model (Kohavi & John, 1997). This motivates the investigation of variable importance measures for different model types.

A second wrapper approach is to permute the values of the predictor variables, one at a time, and determine the decrease in model accuracy for each variable. The permutation of a variable destroys its association with the response. If the prediction accuracy of the new model is significantly lower than for the original model, it implies that the association between predictor and response is significant. For random forests, the oob samples can be permuted, without having to train new forests (Breiman & Cutler, 2003; Archer & Kimes, 2008). The variable importance measure based on permutation $\omega_j(\mathcal{T}_\mathbf{L})$ is calculated according to the following equation (where a is the accuracy of the model, and $\mathbf{L}^j_{\text{OOB}}(\boldsymbol{\theta})$ is the OOB learning sample with variable j permuted):

$$\omega_j(\mathcal{T}_\mathbf{L}) = a\left(\mathcal{T}_{\mathbf{L}(\boldsymbol{\theta})}\right) - a\left(\mathcal{T}_{\mathbf{L}^{\text{j}}_{\text{OOB}}(\boldsymbol{\theta})}\right) \qquad \textit{Eqn. 29}$$

Again, these variable importance measures can be expanded to ensembles of trees by averaging individual tree importance measures:

$$\omega_j = \frac{1}{K}\sum_{k=1}^{K} \omega_j\left(\mathcal{T}_{\mathbf{L}(\boldsymbol{\theta}_k)}\right) \qquad \textit{Eqn. 30}$$

The advantage of wrapper variable importance measures such as ω_j is that, compared to univariate screening methods, ω_j considers multivariate interactions with other input variables.

Variable importance measures based on permutation may vary, thus necessitating repeatedly running algorithms to obtain a distribution of importance measures (Nicodemus & Malley, 2009). Moreover, it has been shown that scaling variable importance measures by their standard errors produces irregular statistical characteristics, which is undesirable (Strobl & Zeileis, 2008).

9.3.1 Sensitivity of variable importance to correlation

Both embedded and wrapper variable importance measures (ε_j and ω_j) are sensitive to the correlation structure of the predictor variables (Archer & Kimes, 2008; Strobl et al., 2008; Nicodemus & Malley, 2009). The reason for the bias in the original formulation of ω_j is an artefact of the permutation process. A variable $\mathbf{X}_j$ is permuted to destroy the association between $\mathbf{X}_j$ and the response, $\mathbf{y}$. However, this permutation also destroys the association between $\mathbf{X}_j$ and the other input variables $\mathbf{X}_i$ ($i \neq j$). A decrease in model accuracy can then imply dependence of $\mathbf{y}$ on $\mathbf{X}_j$ or dependence of $\mathbf{X}_j$ on any of $\mathbf{X}_i$ ($i \neq j$), the latter not being useful in the context of variable importance measures (Strobl et al., 2008). Strobl et al. (Strobl et al., 2008) have suggested a conditional permutation framework to reduce this effect, and applied this conditional permutation to their conditional inference forests.

From a comparative study of random forest and conditional inference forest variable measures in regression, Grömping (2009) showed that RF variable importance measures are more sensitive to correlated variables than CF. With increasing correlation, RF equalized variable importance measures towards correlated variables. In contrast, for high values of m (the number of random split variables) CF variable importance measures were less equalizing to correlated variables at high correlations.

9.3.2 Sensitivity of variable importance to model parameters

In terms of selection of tree ensemble parameters, it is noted that the optimal random split parameter m for model accuracy is not necessarily optimal for variable importance purposes. Genuer et al. (2010) have shown for one case study that larger values of m increase the magnitude of variable importance for truly important variables. Smaller values of m may allow correlated variables higher prominence than merited, as fewer potentially stronger competing variables are present at each split (Strobl et al., 2008).

Grömping (2009) showed that CF variable importance was more sensitive to M (number of random split variables) than RF. A hypothesis for this effect is presented in terms of tree size and splitting competition. Due to the more stringent hypothesis testing stopping condition for conditional inference trees, CF will contain smaller trees than RF. Smaller trees contain fewer splits where weak or correlated variables can be dominant. The higher M, the more likely it is that a weak or correlated variable competes with a truly strong variable, and the less likely that these weak or correlated variables will be included in the tree.

The number of trees K should be sufficiently large to reduce systematic variance, and it has been noted that the more trees an ensemble contains, the higher the stability of the variable importance measures (Genuer et al., 2010).

9.3.3 Sensitivity of variable importance to complex interactions

Strobl et al. (2009) suggest that RF and CF with random split selection may allocate higher importance scores to variables that are involved in complex interactions, as compared to single trees and bagged forests. Random split selection allows variables to appear in various diverse contexts, and thus potentially reveal complex effects on a response variable.

9.3.4 Calculation of tree ensemble variable importance

The implementations of random forests, conditional inference forests and gradient boosting methods with variable importance measures are available in the R system for statistical computing (R Development Core Team, 2010) as the add-on packages randomForest (Liaw & Wiener, 2002), party (Strobl et al., 2007) and gbm (Ridgeway, 2007a). Wrapper-based methods (i.e. permutation-based variable importance) were investigated, as these methods are less sensitive to scale and number of discrete values. Variable importance measures were not scaled by standard errors. Permutation of variables was done unconditionally for CF models, as conditional permutation proved computationally expensive, as noted by (Nicodemus et al., 2010). The model training parameters for the case studies discussed hereafter are given in Table 9.1.

Table 9.1: Tree ensemble model training parameters. (5-CV refers to five-fold cross-validation; NRS refers to no random splits: bagging)

	RF	**RF (NRS)**	**CF**	**CF (NRS)**	**BT**
Number of trees *K*	1000	1000	1000	1000	1000
Optimal number of trees	-	-	-	-	5-CV
Random split variables *M*	floor(√*m*)	*m*	floor(√*m*)	*m*	*m*
Minimum node size	5	5	-	-	5
Significance level	-	-	0.05	0.05	-
Bootstrapping replacement?	no	no	no	no	no
Bootstrap sample size *n*	0.632*N*	0.632*N*	0.632*N*	0.632*N*	0.5*N*
VIM scaled by standard error?	no	no	no	no	no

9.4 Variable importance case studies

Wrapper-based tree ensemble variable importance methods were applied to three case studies, viz. a set of multivariate Gaussian distributions with mean and variance shifts, a simulation of block correlated data with various strengths of association to a response and the Tennessee Eastman process.

9.4.1 Case study 1: Multivariate Gaussian data

In the first case study, a reference condition was generated by randomly sampling from the multivariate Gaussian distribution with parameters $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$, with 500 samples and three variables.

Table 9.2: Multivariate Gaussian parameters for case study 1 (reference condition).

Parameter	$\boldsymbol{\mu}_0$	$\boldsymbol{\Sigma}_0$
Reference	[0,0,0]	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.5 \\ 0 & 0.5 & 1 \end{bmatrix}$

Fault conditions were then generated by randomly sampling from the multivariate Gaussian distribution with parameters $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$, again with 500 samples and three variables. Ten fault conditions were generated, representing mean and variance shifts. The parameters for the reference and fault conditions are given in Table 9.3.

Table 9.3: Multivariate Gaussian parameters for case study 1 (fault conditions).

Parameter	$\boldsymbol{\mu}_i$	$\boldsymbol{\Sigma}_i$	Parameter	$\boldsymbol{\mu}_i$	$\boldsymbol{\Sigma}_i$
Fault 1	[0.4,0,0]	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.5 \\ 0 & 0.5 & 1 \end{bmatrix}$	Fault 6	[0,0,0.8]	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.5 \\ 0 & 0.5 & 1 \end{bmatrix}$
Fault 2	[0,0.4,0]	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.5 \\ 0 & 0.5 & 1 \end{bmatrix}$	Fault 7	[0,0,0]	$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0.5 \\ 0 & 0.5 & 1 \end{bmatrix}$
Fault 3	[0,0,0.4]	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.5 \\ 0 & 0.5 & 1 \end{bmatrix}$	Fault 8	[0,0,0]	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0.5 \\ 0 & 0.5 & 1 \end{bmatrix}$
Fault 4	[0.8,0,0]	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.5 \\ 0 & 0.5 & 1 \end{bmatrix}$	Fault 9	[0,0,0]	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.5 \\ 0 & 0.5 & 2 \end{bmatrix}$
Fault 5	[0,0.8,0]	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.5 \\ 0 & 0.5 & 1 \end{bmatrix}$	Fault 10	[0,0,0]	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.9 \\ 0 & 0.9 & 1 \end{bmatrix}$

For each of the tree ensemble techniques, 100 replicates were generated for each of the faults, and the average variable measures reported.

9.4.2 Case study 2: Correlated data

The second case study was inspired by Archer and Kimes (2008), and involves 100 samples generated from block correlated random variables. Samples are drawn from multivariate Gaussian variables in $J = 10$ blocks, with each block consisting of $Q = 10$ variables. Within the j^{th} block of variables, the correlation between variables were $\rho_j = 0.1j\text{-}0.1$. The block-diagonal variance matrix can be expressed as:

$$\boldsymbol{\Sigma} = \begin{bmatrix} \mathbf{V}_1 & \mathbf{0}_{10x10} & \mathbf{0}_{10x10} & \cdots & \mathbf{0}_{10x10} & \mathbf{0}_{10x10} \\ \mathbf{0}_{10x10} & \mathbf{V}_2 & \mathbf{0}_{10x10} & \cdots & \mathbf{0}_{10x10} & \mathbf{0}_{10x10} \\ \mathbf{0}_{10x10} & \mathbf{0}_{10x10} & \mathbf{V}_3 & \cdots & \mathbf{0}_{10x10} & \mathbf{0}_{10x10} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0}_{10x10} & \mathbf{0}_{10x10} & \mathbf{0}_{10x10} & \cdots & \cdots & \mathbf{0}_{10x10} \\ \mathbf{0}_{10x10} & \mathbf{0}_{10x10} & \mathbf{0}_{10x10} & \cdots & \mathbf{0}_{10x10} & \mathbf{V}_{10} \end{bmatrix} \qquad \textit{Eqn. 31}$$

Here, each block $\mathbf{V}_j$ is a QxQ covariance matrix:

$$\mathbf{V}_j = \begin{bmatrix} 1 & \rho_j & \rho_j & \cdots & \rho_j & \rho_j \\ \rho_j & 1 & \rho_j & \cdots & \rho_j & \rho_j \\ \rho_j & \rho_j & 1 & \cdots & \rho_j & \rho_j \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \rho_j & \rho_j & \rho_j & \cdots & \cdots & \rho_j \\ \rho_j & \rho_j & \rho_j & \cdots & \rho_j & 1 \end{bmatrix} \quad \textit{Eqn. 32}$$

The means of the multivariate Gaussian variables were sampled from a uniform distribution on the interval [6,12]. The first variable within each block of correlated variables ($\mathbf{x}_{j(1)}$) was used in generating a response variable, with the strength of the association determined by parameter β. After scaling the true predictor variable, the response was generated, with random noise induced by a uniform variable **u** sampled from the interval [0,1]:

$$y_i = \begin{cases} 1 & \text{if } \pi_i < u_i \\ 0 & \text{if } \pi_i \geq u_i \end{cases} \quad \textit{Eqn. 33}$$

The threshold is calculated from:

$$\pi_i = \frac{e^{\beta x_{ij(1)}}}{1+e^{\beta x_{ij(1)}}} \quad \textit{Eqn. 34}$$

The effect of correlation and strength of association on variable importance measures can be tested by selecting true predictors from different correlated blocks and adjusting parameter β. For each of the tree ensemble techniques, 100 replicates were generated for 10 different correlation levels (j from 0 to 0.9 in increments of 0.1) and three levels of association strength (β of 0.5, 1 and 1.5). The number of times the correct predictor variable was ranked as most important was recorded, as well as the number of times the highest ranked variable was in the correlation group of the correct predictor variable.

9.4.3 Case study 3: Tennessee Eastman process

In the final case study, the identification of important variables is cast as a realistic fault identification problem. The tree ensemble methods are first used to classify process data representing normal operating and fault conditions on a chemical process plant.[18] Seven faults of the Tennessee Eastman process (see Chapter 7 for more details on the nature of the faults) were selected for investigation: faults 4, 5, 6 ,7, 10, 11 and 14.

For each of the tree ensemble techniques, the average variable importance measures for 10 ensemble initiations for each of the 7 selected faults were recorded. Training data sets were constructed by concatenating the 500 samples of normal operating conditions and 800 samples of fault conditions, for each fault, and generating a corresponding class label vector.

Reported variable importance measures were calculated as the average of variable importance measures from 10 trained forests for each technique.

$$\bar{\omega}_j = ave(\omega_j)_{10\ forests} \quad \textit{Eqn. 35}$$

[18] *This assumes that a fault detection framework, or a process expert, made initial classification of process samples into normal operating conditions and fault conditions categories.*

To determine a threshold for variable importance, a distribution of classification accuracies was generated from 30 trained forests. The threshold was specified as half the range of classification accuracies. This threshold only assumes that the classification accuracies are symmetrically distributed.

$$\tau = \frac{max\left(a\left\{\gamma_{L(\theta_k)}\right\}_1^K\right)_{30\ forests} - min\left(a\left\{\gamma_{L(\theta_k)}\right\}_1^K\right)_{30\ forests}}{2} \qquad \textit{Eqn. 36}$$

9.5 Variable importance case studies results

The results for the application of random forests, conditional inference forests and boosted trees are presented in this section.

9.5.1 Case study 1: Multivariate Gaussian data

The results for case study 1 are presented in Figure 9.1 and Table 9.4. For each fault, the variable importance measures were scaled to the range [0, 100], with the lowest importance at 0 and the highest importance at 100. To reflect the overall performance of a tree ensemble on the 10 faults, the sum of absolute deviances from the control variable importance measures were calculated.

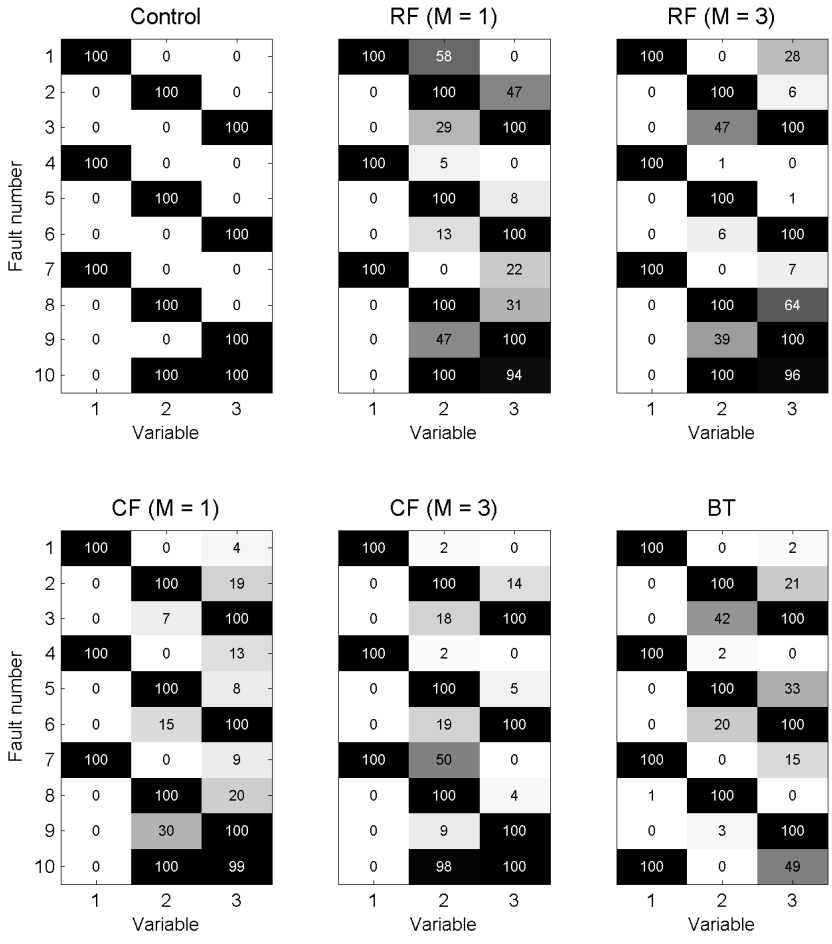

Figure 9.1: Results of case study 1 – simple Gaussian data. Elements represent scaled variable importance measures, with maximum of 100 (black) and minimum of 0 (white). The control graph indicates expected variable importance measures.

Table 9.4: Results of case study 1 – simple Gaussian data. The sum of absolute deviances from the expected variable importance measures for the different tree ensemble techniques

	RF (M = 1)	RF (M = 3)	CF (M = 1)	CF (M = 3)	BT (M = 3)
Sum of absolute deviance	266	197	126	125	290

The CF ensembles performed best, and the BT ensemble worst. Faults 1 to 3 showed higher deviances than faults 4 to 6, reflecting the larger mean change for the latter faults. The boosted trees method was unable to identify both true important variables for fault 10, while all other instances showed the highest importance for the true variables for all faults. However, the change in correlation is not an additive effect, which may explain why boosting showed low success on fault 10.

Specifying a higher value for m (number of random split variables) increases the accuracy of the variable importance measure for RF, and marginally for CF. As all variables are available for splitting at each node when M = m = 3, the input variable most closely associated with the response will be selected each time, taking fluctuations due to bootstrap sampling into account. As faults 1 to 9 involved only one important variable, bagging would be the most successful strategy for selecting the true variable. For fault 10, where two variables are important, selecting M = 1 gave similar results to M = M.

9.5.2 Case study 2: Correlated data

The results for case study 2 are presented in Figure 9.2, as the proportion of times the true variable or true group was identified for different levels of correlation and association.

For all of the tree ensemble methods, three trends are apparent. Firstly, as the association between the true variable and the response increases (larger values of β), the proportion of times the true variable is correctly identified increases. Secondly, as the correlation of within-group variables (ρ) increases, the proportion of times the true variable is identified decreases. Thirdly, as the correlation of within-group variables (ρ) increases, the proportion of times the true group is identified increases. These trends agree with the findings by Archer and Kimes (Archer & Kimes, 2008) and the discussion in the previous section.

In terms of a distinction between the different types of tree ensembles, CF shows fractions up to 0.2 points higher than the other methods for true variables identified. For the fraction of times the true group is identified, CF generally has lower values, as the true important variable is excluded in calculating these fractions. The results obtained with RF and BT do not appear to differ significantly.

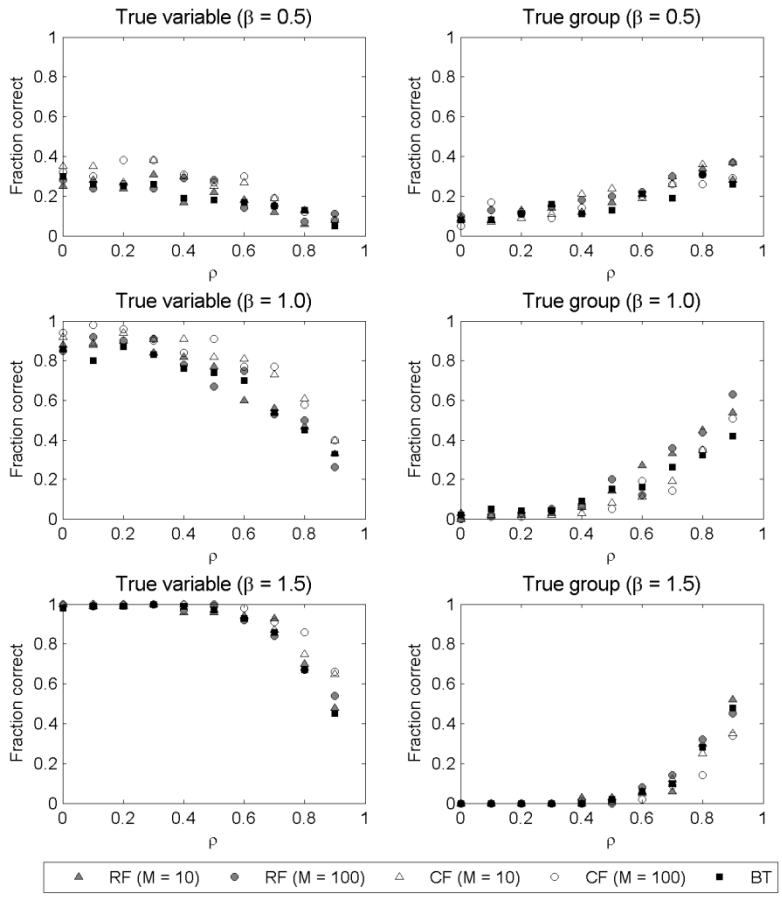

Figure 9.2: Results of case study 2 – correlated data. Each row of graphs represents a different level of association (β) of the important variable to the response. The first column of graphs indicates the number of times the true important variable was successfully ranked, for different correlation structures. The second column of graphs indicates the number of times the correct group (excluding the true important variable) was successfully ranked, for different correlation structures.

9.5.3 Case study 3: Tennessee Eastman process

The results for case study 3 are presented Figure 9.3. Figure 9.3 shows the tree ensemble variable importance measures. As before, for each fault, the variable importance measures were scaled to the range [0, 100], with the lowest importance at 0 and the highest importance at 100. Variable importance measures that exceeded thresholds are indicated with 'x' and 'o' symbols.

For most faults, the number of variables and identities of variables flagged by boosted trees was similar to that of bagged RF and CF. This correspondence arises from the fact that boosted trees do not employ random split selection, but has all variables available as candidate splits at each tree. Many of the effects of bagging will then also be applicable to boosting.

A general trend from the results in Figure 9.3 is the large number of variables that are flagged as important for random split selection forests of RF and CF, compared to the smaller number of variables that are flagged for bagged RF, CF and boosted trees. However, even though large numbers of variables are flagged as important for the random split selection forests, one can gauge from scaled variable importance magnitudes (represented by density of shading in this case) that only a small number of variables have much higher importance measures that the other flagged variables.

The large number of flags for random split selection forests will be considered in terms of correlation, accuracy in greedy model building, and complex interactions.

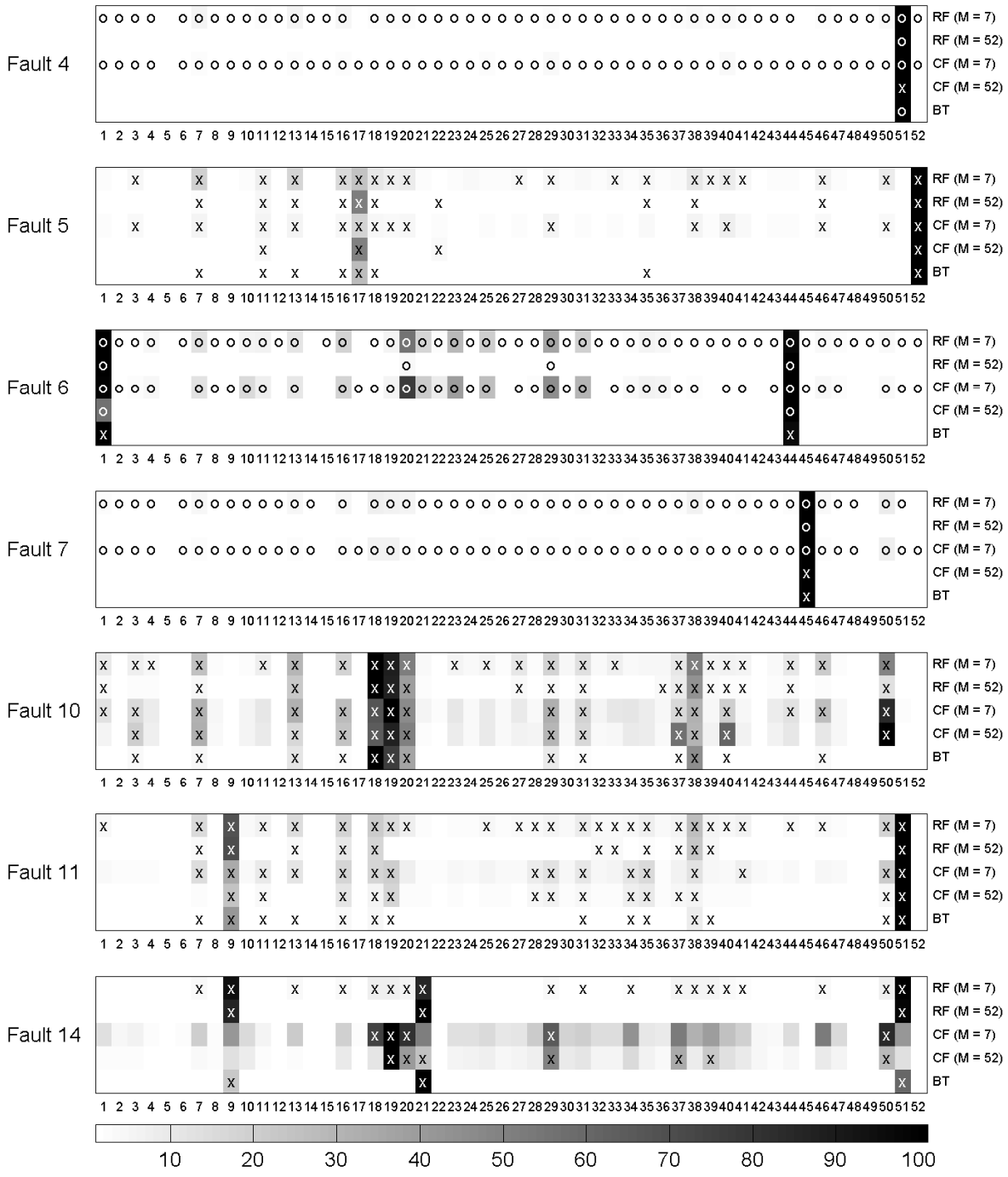

Figure 9.3: Results of case study 4 – Tennessee Eastman process. Scaled variable importance is represented by the gray scale of the graph elements, with maximum 100 (black) and minimum 0 (white). The 'x' symbol indicates variables with variable importance measures exceeding the threshold defined by the range of model errors. The 'o' symbol indicates variables with variable importance measures higher than 0, where model error rates were constant over 30 ensemble initiations for a specific data set.

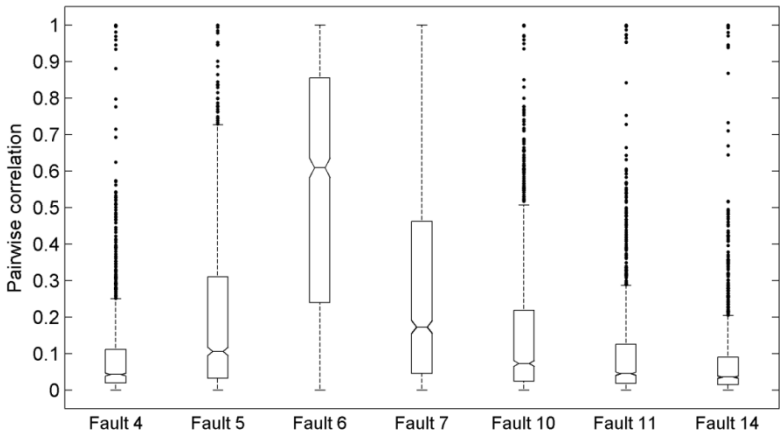

Figure 9.4: Boxplots of pairwise correlation for Tennessee Eastman process data. Boxes represent 25% and 75% percentiles, notches represent medians, with outliers shown as points.

Figure 9.4 shows the distributions of pairwise correlations for the 52 Tennessee Eastman process variables. Faults 5, 6 and 7 have medium to high correlations, and from the results of case study 3, it can be inferred that variable importance performance from random split selection forests would be poor, especially in terms of high numbers of false alarms. This may explain the high number of flagged variables for these methods in Figure 9.3 for faults 5, 6 and 7.

Another consideration is the average error rates from repeated forest iterations. As indicated in Figure 9.3, faults 4, 6 and 7 had constant model error rates over 30 ensembles for the indicated techniques. The average error rates and half ranges for all techniques and all fault data sets are presented in Table 9.5 and Table 9.6. The half ranges for faults 4, 6 and 7 are zero for a number of tree ensembles, suggesting no variation in highly accurate models classifying the normal operating condition and fault data.

Table 9.5: Average model error rates for Tennessee Eastman process data. Calculated over 30 iterations.

	Fault 4	**Fault 5**	**Fault 6**	**Fault 7**	**Fault 10**	**Fault 11**	**Fault 14**
RF (M = 7)	0	0.0404	0	0	0.0233	0.0344	0.0028
RF (M = 52)	0	0.0097	0	0	0.0322	0.0426	0.0030
CF (M = 7)	0.0015	0.0681	0	0.0015	0.0670	0.0735	0.166
CF (M = 52)	0.0010	0.0203	0.00077	0.0010	0.0672	0.0808	0.154
BT	0.0001	0.366	0.00014	0.0001	0.520	0.453	0.128

Table 9.6: Half model error ranges for Tennessee Eastman process data. Calculated over 30 iterations.

	Fault 4	**Fault 5**	**Fault 6**	**Fault 7**	**Fault 10**	**Fault 11**	**Fault 14**
RF (M = 7)	0	0.005	0	0	0.003462	0.002692	0.001154
RF (M = 52)	0	0.000769	0	0	0.001923	0.001923	0.000385
CF (M = 7)	0	0.004231	0	0	0.005	0.005	0.012692
CF (M = 52)	0.000385	0.001923	0	0.000385	0.006538	0.003462	0.009615
BT	0	0.002106	0.000514	3.27E-09	0.001489	0.0015	0.00075

Closer inspection of the variables flagged by bagged RF and CF are merited to investigate the lack of variation in model accuracies. Variable time series plots are presented in Figure 9.5.

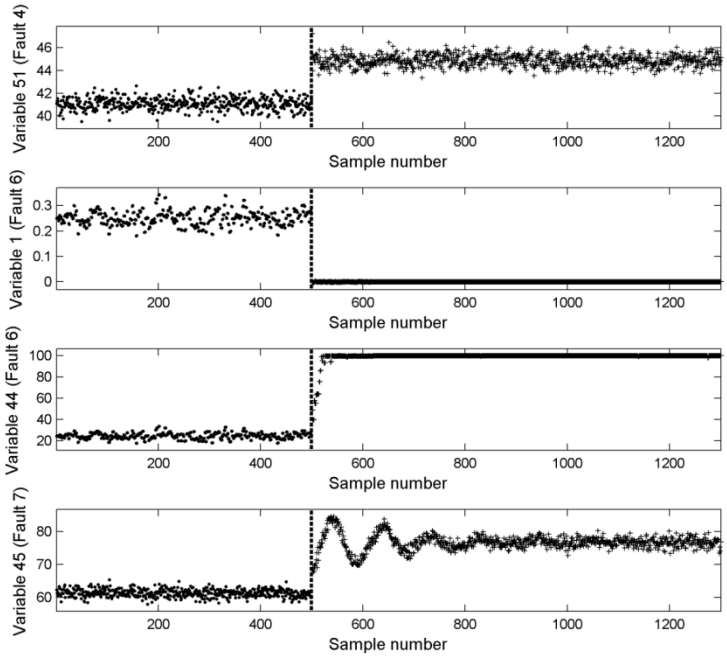

Figure 9.5: Selected variables for certain faults of the Tennessee Eastman process data. The broken lines at sample number 500 represent the start of fault conditions, '•' represents normal operating conditions and '+' represents fault conditions.

A characteristic of all the variables shown in Figure 9.5 is the complete separation of the distribution of normal operating condition values and fault conditions values. In any decision tree where all variables are available as split candidates, these variables will naturally be selected as the first, and only, splitting variable. Splitting on these variables will ensure high, if not complete, accuracy on the training data. This greedy data splitting approach does not consider variable interactions. It is evident that bagged forests trained on these data would consist of nearly identical trees (given randomization introduced by bootstrap sample selection), all splitting on the variables shown, as all variables compete against these strongest candidates.

For certain fault conditions such a single strong candidate may be the sole important variable. For example, in fault 4, the reactor cooling water temperature undergoes a step change (see Chapter 7). Given the control system in place, this would lead to a corresponding change in the cooling water flow rate (variable 51). Whether only the cooling water flow is affected, without disturbances in the reactor temperature, pressure and product concentration, is dependent on the process dynamics. If the process dynamics are fast enough, the only perceptible change may well be the cooling water flow rate, and hence it is the sole important variable. Fault 6, which involves loss of component A in the feed stream, would involve wide ranging effects across process units, with changed steady state values in many process variables. A variable importance technique that only singles out a few variables (e.g. variables 1 and 44) would not capture the full extent of causation and association. Again, it is noted that tree ensembles, as with many statistical inference techniques, cannot distinguish between causation and association through correlation.

It has been suggested by Strobl et al. (Strobl et al., 2009) that random split selection forests would be more sensitive to complex interactions than bagged forests or single decision trees, and this aspect may be at play for faults affecting many process units, with complex process dynamics. Boosted trees, with its stepwise fitting of residuals, could also be limited in its distinction of complex interactions that are not additive.

The variable importance measures for fault 14 show a departure from those of the other faults, in that variables flagged by bagged CF (and random split selection CF) show very little correspondence with the variables flagged by the other tree ensemble models. Fault 14 involves the sticking of the reactor cooling water flow valve (see Chapter 7), with RF and boosted trees indicating that the reactor cooling water flow (variable 51) has high importance for this fault, which the CF methods do not flag. From Table 9.5, CF ensembles show the highest error rates for detection of this fault. It has been said that low error rates do not necessarily correspond to good interpretation, but that does not suggest that the reverse is true. The conditional inference scheme may not be suitable to this fault, with insignificant splitting variables potentially resulting in early stopping, small trees, and inadequate variable coverage for importance calculations.

In terms of computational expenses for the Tennessee Eastman process, random split selection RF, random split selection CF and BT require on average 11, 226 and 21 seconds to construct one forest and calculate variable importance measures. CF is then the most computationally expensive, with a computation cost of at least an order of a magnitude larger than RF and BT. Since several forests must be grown to obtain stable measures, this can be a limiting factor for conditional inference forests.

9.6 Discussion

Case study 1 indicated that boosted trees struggle to identify the true variables when a classification problem has a nonadditive structure. In case study 2, the sensitivity of all tree ensemble techniques to correlated input data was shown. As within-group correlation between variables increased, the accuracy of tree ensembles to indicate true variables decreased. Conditional inference forests showed better performance than random forests and boosted trees in identifying the true variable for high correlations.

In the final case study, application of tree ensemble variable importance was cast as a fault diagnostic problem, via the Tennessee Eastman process. This study illuminated some practical considerations. Random split selection forests showed high numbers of variables flagged as important, which may by symptomatic of highly correlated data (as present in some of the faults), or the presence of one or a few variables with distinctly different distributions for normal operating and fault conditions, or indicative of complex interactions. As the Tennessee Eastman process simulation includes various PID process controllers, complex interactions may be at play here. This suggests that the flagging of many important variables is correct, as many process variables are affected by the fault conditions. Inspecting the magnitude of the variable importance measures would then be the next stop in interpreting flagged important variables.

Considerations in using tree ensembles for variable importance indicators include the following: Including expert knowledge of the process under consideration in data preprocessing or the interpretation of results would be beneficial. In terms of data preprocessing, representative or latent variables can be selected to characterize specific process units, thereby reducing the presence of correlated variables.

When results are interpreted, the list of flagged variables indicated as important by tree ensembles can be incorporated into an expert view on the process and its dynamics, in order to distinguish causation and association. The selection of the correct threshold for the flagging of important variables is a tuneable parameter that can be considered in further studies. For exploratory purposes, an inspection of variable rankings and importance magnitudes may prove insightful as well. Using bagged ensembles would then not be ideal, as all other (less important) variables would be overshadowed by a few strong variables.

As a conclusion from these validation studies, variable importance measures from random forests with random splits can be viewed as striking a good balance between incorporating sensitivity to complex interactions, not overemphasizing a few strong variables and being computationally efficient.

Nomenclature

i	impurity function
J	non-terminal nodes
K	number of ensemble members
k	class indicator
$\mathbf{L}$	learning set of input variables $\mathbf{X}$ and output variable $\mathbf{y}$
m	dimensionality of input data $\mathbf{X}$
M	number of random split variables
Q	size of block covariance matrix
$\mathcal{T}$	decision tree function
$\mathbf{u}$	uniformly distributed variable
$\mathbf{V}$	block covariance matrix
$\mathbf{X}$	input variables
$\mathbf{y}$	response variable
β	influence parameter
ε	embedded variable importance
$\boldsymbol{\theta}$	random vector
$\boldsymbol{\mu}$	mean vector
π	threshold for classification
$\boldsymbol{\Sigma}$	covariance matrix
τ	variable importance threshold
ω	wrapper variable importance

CHAPTER 10 - FAULT IDENTIFICATION: PARTIAL DEPENDENCE

The interpretation of identified process variables is investigated in this chapter, using the approach of random forest based partial dependence. The partial dependence methodology is introduced, and together with random forest variable importance, incorporated into a data visualization tool for fault identification and interpretation. A variable importance thresholding technique based on the addition of a dummy variable is also introduced.

The partial dependence methodology is validated on linear and nonlinear simulation studies. Results show that random forest variable importance is able to distinguish important variables in both linear and nonlinear cases. The random forest partial dependence approach is able to capture and present nonlinear structure, even though only low predictive performance is achieved.

The partial dependence methodology is then applied to faults of the Tennessee Eastman and calcium carbide processes. The dummy variable threshold for variable importance is shown to be less successful where data are correlated. Partial dependence plots for the NOC / fault probabilities show variable thresholds between these two classes.

Overall, the partial dependence methodology is a useful approach to visualizing and interpreting so-called black box models such as random forests, as well as for fault identification and interpretation.

10.1 Overview

Statistical process monitoring often involves the construction of a process model to predict a certain key performance indicator. Cast as a fault diagnosis problem, the performance indicator may be class membership to either normal operating conditions or fault conditions. An empirical process model incorporates process variables as input data, the key performance indicator as response data, and is trained using some statistical inference method. The utility of process models lies not only in their predictive capacity, but also in the qualitative interpretation of the model. Two key areas of interpretation reside in the identification of important process variables, and the nature of the influence that these important process variables have on the key performance indicator.

When considering multiple linear regression as a process model, these interpretations can be summarized in the values, signs and confidence limits of the linear regression coefficients. If the input data were scaled in a preprocessing step, the relative coefficient magnitudes give an indication of variable importance. The signs of the coefficients, coupled with their magnitude, give an indication of the direction and magnitude of influence of the specific process variable on the response. However, linear models have restricted use, since most physical processes are nonlinear and exhibit interaction between variables. Not only does the predictive power of linear methods decrease for nonlinear systems, but interpretation in terms of variable importance and influence on the response deteriorates. In nonlinear systems with interactions, the influence of a process variable on a response is not only conditional on its own value, but also on the values of other process variables.

To overcome the restrictions of linear models, a variety of nonlinear models have been implemented in process modelling, including neural networks (Massinaei & Doostmohammadi, 2010; Jorjani et al., 2007) and support vector regression (Wang et al., 2010; Cai et al., 2009). Application of random forest modelling and associated interpretation via variable importance measures and partial dependence plots is a new data mining tool, not yet wide spread in the applied science and engineering fields. Notable applications are found in the fields of

climatology (Deloncle et al., 2007), ecology (Furlanello et al., 2003; Carlisle et al., 2008; Lennert-Cody et al., 2008; Oppel et al., 2009; Girardello et al., 2010), genetics (Eller et al., 2007) and sociology (Berk et al., 2009). These investigations showed several nonlinear dependencies, some with apparent variable thresholds, which could be interpreted by appropriate experts. A simple application of this interpretive brand of random forest modeling to process engineering was recently shown by Berrado and Rassili, in the modeling of steel thixoforming (Berrado & Rassili, 2010).

10.2 Partial dependence

Nonlinear models do not lend themselves to simple direction of influence analysis as with linear models. The reason for this is inherent in the definition of a nonlinear model: interactions and transformations of variables are taken into consideration. One cannot then simply say (as for linear models) that as a certain variable increases, the response would necessarily increase at a constant rate, for all possible values of all other variables. Correlation and interactions of variables add intricacies to the interpretation of influence.

The interpretation of the dependence of a response variable to a single variable in the case of tree-based methods is now considered. Tree-based methods and their ensemble extensions partition the input space into subspaces such that each subspace has a relatively homogenous response. The locally fitted prediction values are independent of their neighbouring regions, with no continuity constraints. This allows for flexible modelling, but destroys the idea of single-direction influence. By further averaging over an ensemble of diverse members, direct interpretation is a non-trivial task.

Friedman has suggested an approach to investigating the influence of a variable on a response (given any predictive model): the partial dependence plot (Friedman, 2001). It is useful to inspect a plot of the predicted response for the range of values of a specific input variable, as averaged over all training values of the other input variables.

Suppose $\mathbf{X}$ is the input data, and $f(\mathbf{X})$ is the predictor model for the response. Let $\mathbf{X}_S$ be a subset of the input variables of interest, and $\mathbf{X}_C$ all other input variables not included in $\mathbf{X}_S$. The model depends on all variables: $f(\mathbf{X}) = f(\mathbf{X}_S,\mathbf{X}_C)$, but the partial dependence of the predicted response to a subset of interest can be defined as the marginal average of the approximation function over $\mathbf{X}_C$ (Friedman, 2001):

$$\bar{f}(\mathbf{X}_S) = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{X}_S, x_{iC}) \qquad \textit{Eqn. 37}$$

With x_{iC} the values of samples in $\mathbf{X}_C$ in the training sample.

For classification problems (e.g. classification to NOC and fault classes), a continuous response for each class label can be generated by using a modified logit function of the portion of tree votes for each class (Friedman, 2001):

$$f_k(\mathbf{X}) = log\big(p_k(\mathbf{X})\big) - \frac{1}{K}\sum_{j=1}^{K} log\left(p_j(\boldsymbol{X})\right) \qquad \textit{Eqn. 38}$$

With p_k the proportion of votes for class k among the K trees of the ensemble of trees. Values greater than zero suggest that the majority of trees voted for class k.[19] The modified logit function for a two-class classification

[19] *Eqn. 38 is undefined for $p_k = 0$ or $p_k = 1$. In practice, this is circumvented by assigning $p_k = 0.5/K$ when $p_k = 0$, and $p_k = 1-0.5/K$ when $p_k = 1$. This gives the logit function a resolution of half a tree vote at the left and right edges of $p_k \in [0,1]$.*

problem is shown in Figure 10.1. Eqn. 37 can then be applied to this modified logit function to determine partial dependencies on selected variables.

Friedman suggested that the partial dependence function is especially adept at capturing the true influence of input variables where said influence is additive of multiplicative (Friedman, 2001).

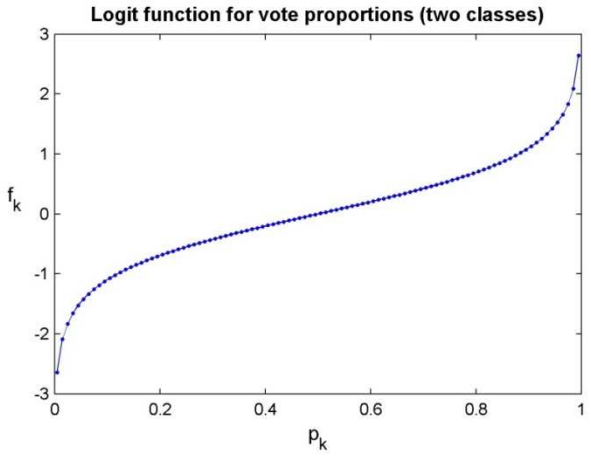

Figure 10.1: Modified logit function for one class of a two-class classification problem

10.3 Visualization tool based on variable importance and partial dependence

Variable importance information can add value to partial dependency results, as the direction and magnitude of influence of a less important variable is not necessarily as insightful as the direction and magnitude of influence of an important variable. Visualization of the partial dependencies for single important variables and combinations of two important variables will enhance the qualitative understanding of the approximation function.

Once variable importance measures (ω_j) are obtained (as discussed in Chapter 9, using a random forest with random split selection), a significance threshold would be useful in determining whether variables are significant in describing the response. For random forest variable importance, Breiman suggested that ω_j be scaled by its standard errors, as this would result in a z-score with an asymptotically standard normal distribution (Breiman & Cutler, 2003). However, it has been shown that scaling ω_j in this way gives rise to irregular statistical characteristics, which is undesirable (Strobl & Zeileis, 2008).

In this study, a simple approach is used to generate a variable importance threshold based on an input variable that is known to not be influential or correlated to the response. The input variable matrix **X** is augmented with a dummy variable, $\mathbf{X}_D$, randomly sampled from the standard normal distribution, and the random forest trained on this new input matrix. As the dummy variable $\mathbf{X}_D$ has no influence on, as well as being uncorrelated to, the response **y**, the variable importance measure for the dummy variable (ω_D) serves as a threshold for importance.

A data visualization tool based on the three most important variables (as identified by random forests) is presented here. The data visualization tool consists of a matrix of nine plots. For the three important variables, single variable partial dependence plots, as well as combination two-variable partial dependence surfaces are shown. A further aid to interpretation is presented in the form of scatterplots for the considered variable, superimposed with response variable. The scatterplots give qualitative indications of the correlations between variables, as well as the support (distributions) of the different variables. This matrix of plots gives an overview for qualitative interpretation of the data set under consideration.

10.4 Linear and nonlinear regression examples

To demonstrate the concept of partial dependence, two simulated linear and nonlinear regression systems are investigated. Random forest variable importance measures ω_j are compared to linear regression model coefficients β_j and their confidence intervals. Linear regression was done on input variables scaled to zero mean and unit variance, to allow comparison of linear regression coefficients. Random forest variable importance measures were calculated over 30 random permutations for each variable. For two-variable partial dependences, a grid of 30x30 input variable values were evaluated, ranging in each direction across the minima and maxima of each variable.

A measure of the goodness-of-fit for random forest regression models is the pseudo-R^2 value, calculated from the OOB mean squared error (*mse*) of the ensemble and the variance (*var*) of the response variable:

$$R^2 = 1 - \frac{mse\left(\left\{\gamma_{\mathbf{L}_{\mathrm{OOB}}(\boldsymbol{\theta}_k)}\right\}_1^K\right)}{var(\mathbf{y})} \qquad \textit{Eqn. 39}$$

10.4.1 Linear simulation data

A simple linear system was investigated to show that random forest regression and associated interpretive tools are also applicable to linear problems. The input data **X** consisted of 10 uniformly distributed variables, U[0,1]; with the response **y** a linear function of the first three variables:

$$\mathbf{y} = 10\mathbf{X}_1 + 5\mathbf{X}_2 + 2.5\mathbf{X}_3 + \varepsilon \qquad \textit{Eqn. 40}$$

With ε random Gaussian noise with zero mean and unit variance. A data set with 500 samples was generated.

Both models showed high predictive ability for the linear simulation study, with the linear regression model delivering a R^2 of 0.926 and the random forest model giving a pseudo-R^2 of 0.853. The lower R^2 for the random forest model is still indicative of a good fit, but performs worse than the linear model.

From Figure 10.2 and Figure 10.3, both the linear and random forest model was successful in identifying only the applicable variables as important; with the threshold of the dummy variable only exceeded by the three important variables. The magnitudes of the linear coefficients further reflect the true influence of the three important variables on the response, with the coefficient for $\mathbf{X}_1$ exceeding that of $\mathbf{X}_2$, which in turn exceeds that of $\mathbf{X}_3$. The same ranking is observed from the random forest importance measures, but the decrease in importance measures is not linear, which would reflect the true values of the coefficients (10, 5 and 2.5, respectively, for $\mathbf{X}_1$, $\mathbf{X}_2$ and $\mathbf{X}_3$). It is noted that the random forest variable importance measures represent the change in mean squared errors after variable permutation; this squaring of residuals is responsible for the nonlinear ratio between importance measures with a linear ratio of effect size.

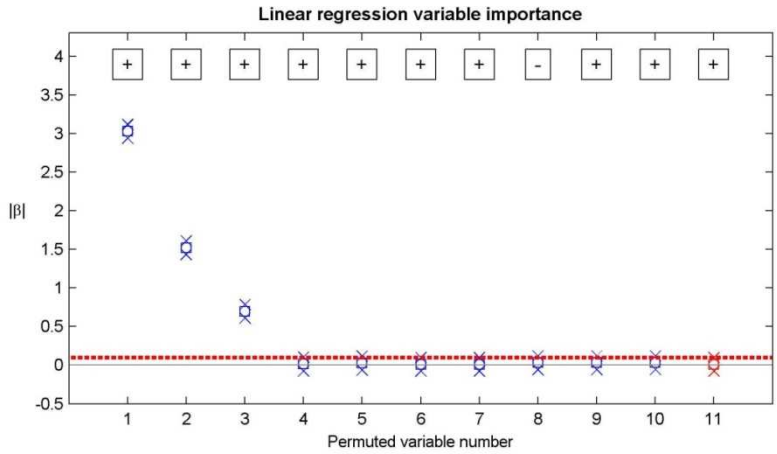

Figure 10.2: Linear regression variable importance measures for linear simulation data. Squares indicate coefficient values, with crosses indicating upper and lower 95% confidence limits. The dummy variable is shown in red, with the dashed red line indicating the upper 95% confidence limit of the dummy variable coefficient.

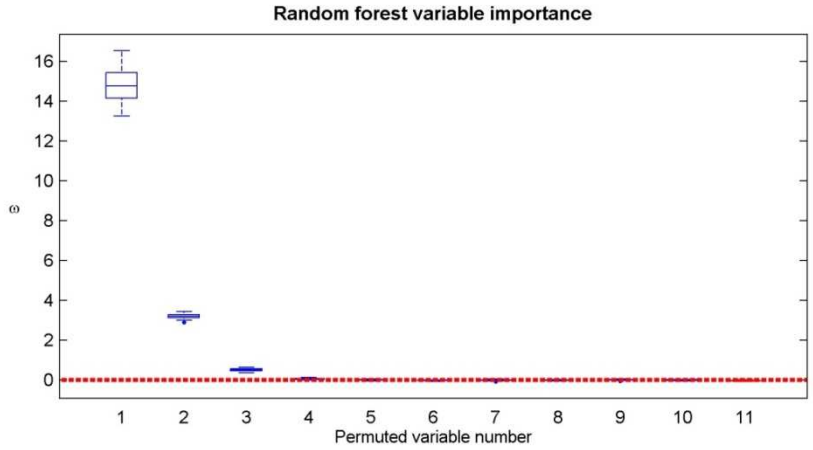

Figure 10.3: Random forest variable importance measures for linear simulation data. The dummy variable is shown in red, with the dashed red line indicating the upper 95% confidence limit of the dummy variable importance measure.

From Figure 10.4, the partial dependences obtained from the random forest model indicate generally linear trends for all considered variables, with no interactions evident from the two-variable partial dependences. The range of the partial dependences corresponds to the ratio of the true coefficients of the generating model.

The flattening of the one-variable partial dependencies at extreme values of variables (see graphs on the diagonal) may be an artefact of random forest regression: Regression trees cannot deliver predictions outside the range of training response values, as node predictions are the average of samples within that node. Due to bootstrap aggregated sampling during random forest construction, the omission of samples with extreme response values further flattens the predictions at the extremes of input variables.

From these results, the statement is made that random forest modelling, variable importance measures and partial dependences generalize well to linear systems as well.

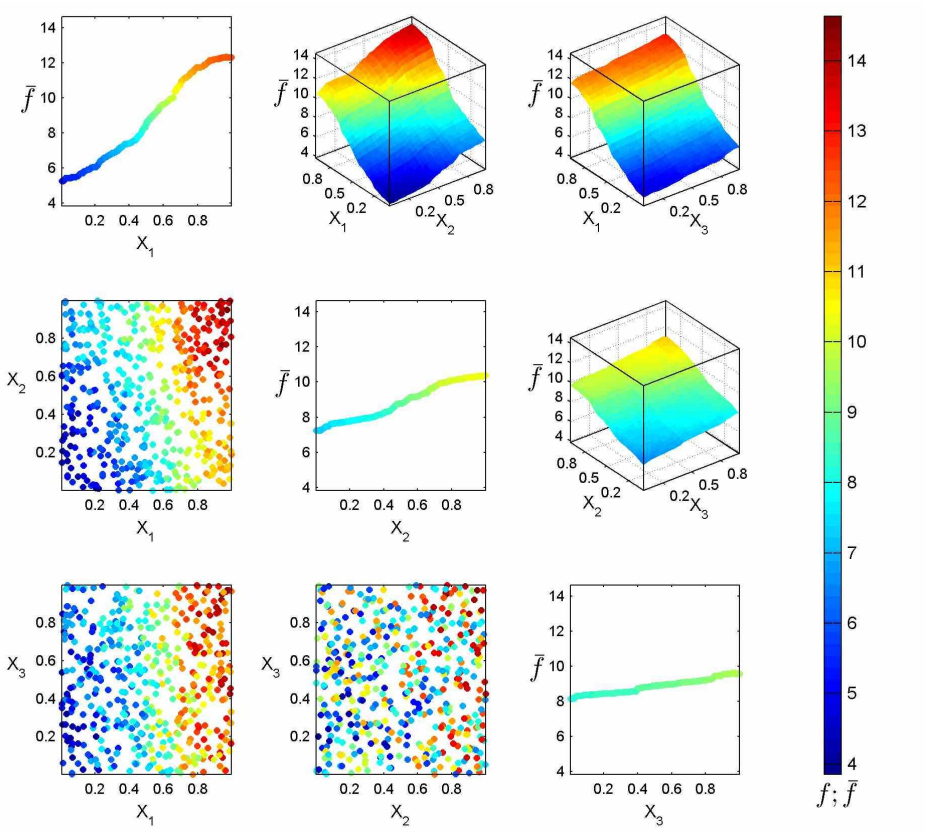

Figure 10.4: Visualization of linear simulation data with partial dependences of the three most important variables. Diagonal plots show one-variable partial dependences, above-diagonal plots show two-variable partial dependences and below-diagonal plots show two-variable scatterplots. The colour range represents the extent of function approximation values for the response.

10.4.2 Nonlinear simulation data

A nonlinear system was investigated to compare random forest and linear regression on a nonlinear system, and to show the ability of random forests to estimate known two-variable partial dependence surfaces. The input data **X** consisted of 10 uniformly distributed variables, U[-3,3]; with the response **y** a nonlinear function of the first three variables, including interaction between $\mathbf{X}_1$ and $\mathbf{X}_2$:

$$\mathbf{y} = 3(1-\mathbf{X}_1)^2 e^{-\mathbf{X}_1^2-(\mathbf{X}_2+1)^2} - 10\left(\mathbf{X}_1/5 - \mathbf{X}_1^3 - \mathbf{X}_2^5\right)^2 e^{-\mathbf{X}_1^2-\mathbf{X}_2^2} - 1/3\, e^{-(\mathbf{X}_1+1)^2-\mathbf{X}_2^2} + 2sin(\pi\mathbf{X}_3) + 0.2\varepsilon 1)$$

Eqn. 41

With ε random Gaussian noise with zero mean and unit variance. A data set with 500 samples was generated. The effect of $\mathbf{X}_1$ and $\mathbf{X}_2$ on the response, ignoring the effect of $\mathbf{X}_3$ and noise, is shown in Figure 10.5.

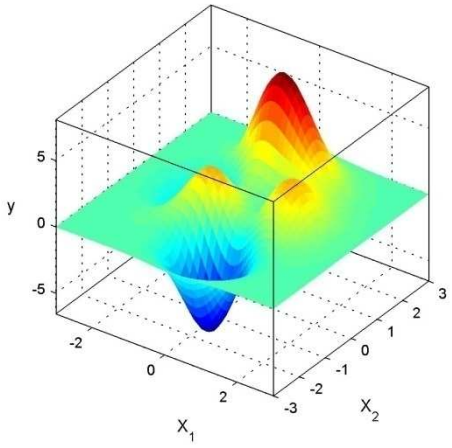

Figure 10.5: True partial dependence of response on X1 and X2 for nonlinear simulation. (Ignoring other variables and noise.)

Both models showed low predictive ability for case study 2, with the linear regression model delivering a R^2 of 0.114 and the random forest model giving a pseudo-R^2 of 0.382. Random forest showed a higher prediction power, as expected, due to the nonlinear nature of the data.

From Figure 10.6 and Figure 10.7, the linear regression model could only identify one of the important variables, while the random forest model correctly identified all three important variables ($\mathbf{X}_1$, $\mathbf{X}_2$ and $\mathbf{X}_3$), regardless of the low predictive ability of said model. The dummy variable threshold was again successful in identifying only the truly important variables.

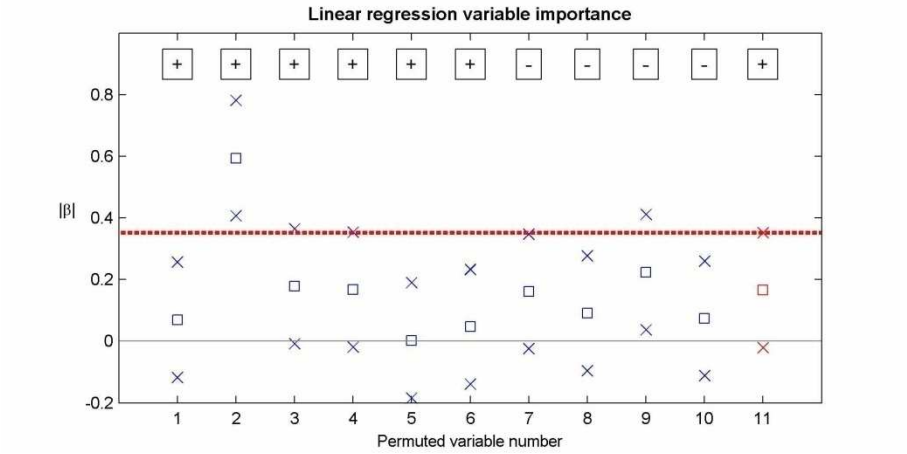

Figure 10.6: Linear regression variable importance measures for nonlinear simulation data. Squares indicate coefficient values, with crosses indicating upper and lower 95% confidence limits. The dummy variable is shown in red, with the dashed red line indicating the upper 95% confidence limit of the dummy variable coefficient.

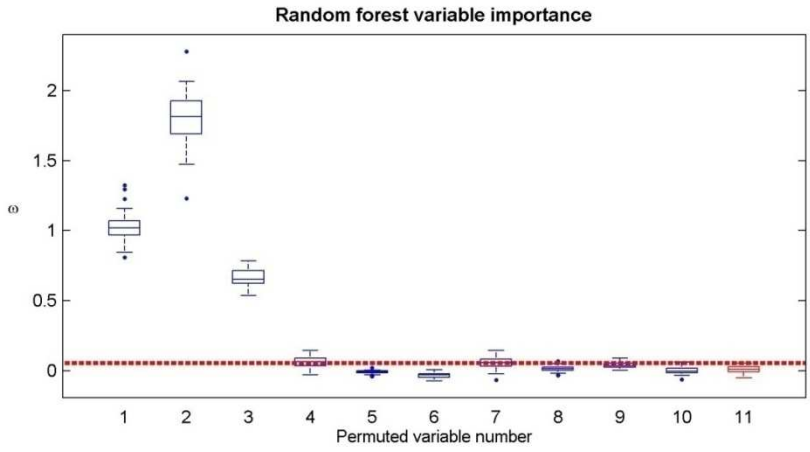

Figure 10.7: Random forest variable importance measures for nonlinear simulation data. The dummy variable is shown in red, with the dashed red line indicating the upper 95% confidence limit of the dummy variable importance measure.

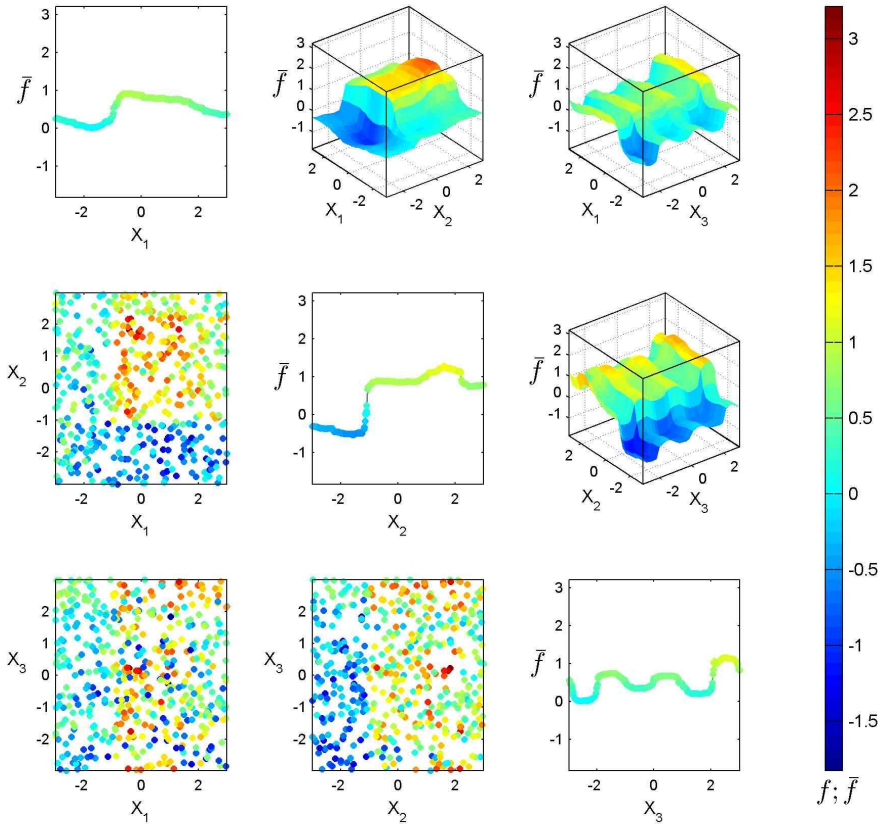

Figure 10.8: Visualization of nonlinear simulation data with partial dependences of the three most important variables. Diagonal plots show one-variable partial dependences, above-diagonal plots show two-variable partial dependences and below-diagonal plots show two-variable scatterplots. The colour range represents the extent of function approximation values for the response.

From Figure 10.8, random forest partial dependences closely approximate the true dependence of the response on the three important variables. The two-variable partial dependence surface of variables $\mathbf{X}_1$ and $\mathbf{X}_2$ depicted in Figure 10.8 resembles the noise-free, $\mathbf{X}_3$-independent relation between $\mathbf{y}$, $\mathbf{X}_1$ and $\mathbf{X}_2$ shown in Figure 10.5, with a similar peak and valley structure. The one-variable partial dependence of the response on $\mathbf{X}_3$ is

convincingly captured in Figure 10.8, resembling a sine wave, which is the true relationship between **y** and $\mathbf{X}_3$. As Friedman suggested, partial dependences are successful in capturing additive structure (Friedman, 2001).

The single direction coefficients of the linear regression models do not capture the nonlinear nature of the dependence of the response on the input variables, as shown by the random forest partial dependence plots. Even though the random forest model was not a powerful predictor (reflected by its low pseudo-R^2), variable importance measures and partial dependences still gave valid qualitative information on the data structure.

10.5 Application to Tennessee Eastman and calcium carbide process data

The visualization tool (based on random forest variable importance and partial dependence) developed and validated in the previous section is now applied as fault identification and interpretation tool to two fault conditions of the Tennessee Eastman process (faults 4 and 14) and to the calcium carbide process data. (See Chapter 6 for details of these data sets, and Chapter 7 for the results of the application of the random forest fault diagnostic framework.)

For each fault investigated, the fault identification problem is cast as a classification problem: once a fault detection scheme (such as a process expert or the random forest fault diagnosis framework) has detected fault conditions, a data set is created with labelled NOC and fault data. A classification random forest can now be trained, and this forest evaluated in terms of variable importance and partial dependence. For this work, it is assumed that all fault samples are detected.

Random forest variable importance measures were calculated over 30 random permutations for each variable. For two-variable partial dependences, a grid of 30x30 input variable values were evaluated, ranging in each direction across the minima and maxima of each variable.

10.5.1 Tennessee Eastman process: fault 4

Fault 4 is simulated by introducing a step change in the reactor temperature (variable 9). As control loops compensate for the temperature increase, a step change in reactor cooling water flow rate (variable 51) is induced, while all other variables return to steady state. Variable 51 is assumed to be most closely associated with the fault (Russell et al., 2000).

Figure 10.9 gives the random forest variable importance measures for fault 4, while Figure 10.10 shows the one-variable and two-variable partial dependencies of the three most important variables (as identified by random forest variable importance). From the variable importance plot, variable 51 is clearly the most important variable, with all other variables much less important. More than 15 variables exceed the dummy variable threshold, indicating low-level correlation of variables to the fault condition.

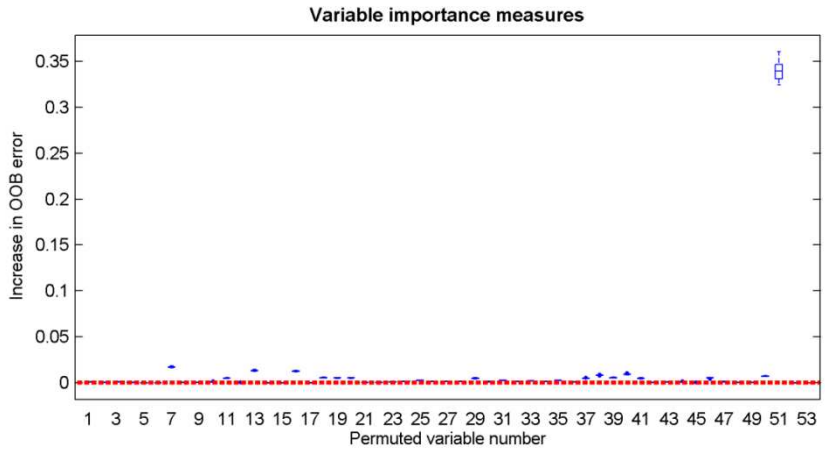

Figure 10.9: Random forest variable importance measures for fault 4 of the Tennessee Eastman process. The dummy variable is shown in red, with the dashed red line indicating the upper 95% confidence limit of the dummy variable importance measure.

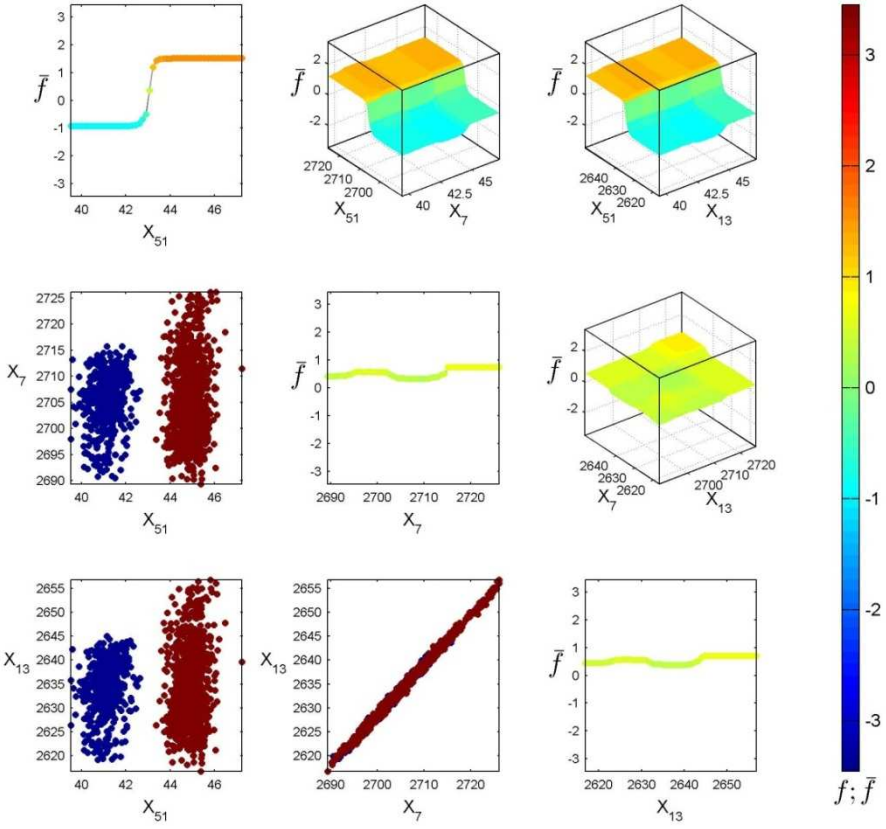

Figure 10.10: Visualization of NOC of fault 4 of the Tennessee Eastman process with partial dependences of the three most important variables. Diagonal plots show one-variable partial dependences, above-diagonal plots show two-variable partial dependences and below-diagonal plots show two-variable scatterplots. The colour range represents the log-odds approximation for the fault class. Negative log-odds suggest classification to the NOC class, while positive log-odds suggest classification to the fault class.

From the one-variable partial dependence plots for fault 4 (Figure 10.10), it is apparent that there is a distinct threshold for variable 51 that is associated with the crossover of NOC conditions to fault 4. Comparing the ranges of the one-variable partial dependence plots for fault 4, it is clear that variable 51 has a much greater influence (association) than variables 7 and 13, the second and third most important ranked variables. This

agrees with their comparatively low variable importance scores in Figure 10.9. The two-variable partial dependence plots are not particularly insightful, as variable 51 dominates the effect on the modified logit function. From the scatterplots in Figure 10.10, the threshold value of variable 51 is also apparent, as the NOC data and fault data form two distinct clusters. The dominance of variable 51 was discussed in Chapter 9.

10.5.2 Tennessee Eastman process: fault 14

A sticking valve for reactor cooling water is simulated for fault 14, causing large fluctuations in reactor temperature (variable 9), the reactor cooling water outlet temperature (variable 51) and the reactor cooling water flow rate (variable 21) (Russell et al., 2000; Shao & Rong, 2009).

Figure 10.11 gives the random forest variable importance measures for fault 14, while Figure 10.12 shows the one-variable and two-variable partial dependencies of the three most important variables (as identified by random forest variable importance). From the variable importance plot, variables 9, 21 and 51 are the three most important variables. More than 17 variables exceed the dummy variable threshold, indicating low-level correlation of variables to the fault condition.

The scatterplots from Figure 10.12 show an interesting relationship between variables 9 and 21, as well as between variables 21 and 51. These confined circular regions for fault conditions indicate the fluctuations due to the controller actions to bring the reactor temperature, cooling water flow and cooling water outlet temperature under control. The one-variable partial dependence plots show the narrow operating regimes for variables 9, 21 and 51 that are required for normal operating conditions. It is interesting to note how the two-variable partial dependence plot of variables 9 and 51 incorporate the support of these variables. From the scatterplot of variables 9 and 51, no data are available for low values of variable 9 with high values of variable 51, or for high values of variable 9 with low values of variable 51. These regions in the partial dependence plot for variables 9 and 51 have lower average logit values (indicating a lower probability of fault conditions). The lack of support in these regions thus lowers the expected probability for fault conditions.

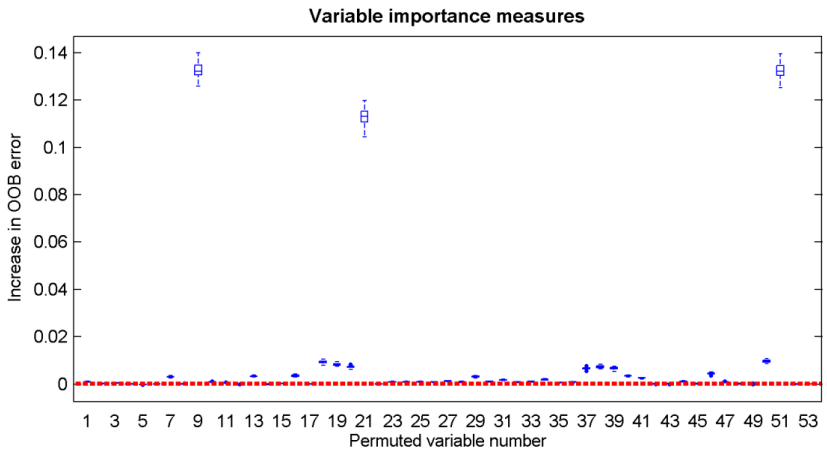

Figure 10.11: Random forest variable importance measures for fault 14 of the Tennessee Eastman process. The dummy variable is shown in red, with the dashed red line indicating the upper 95% confidence limit of the dummy variable importance measure.

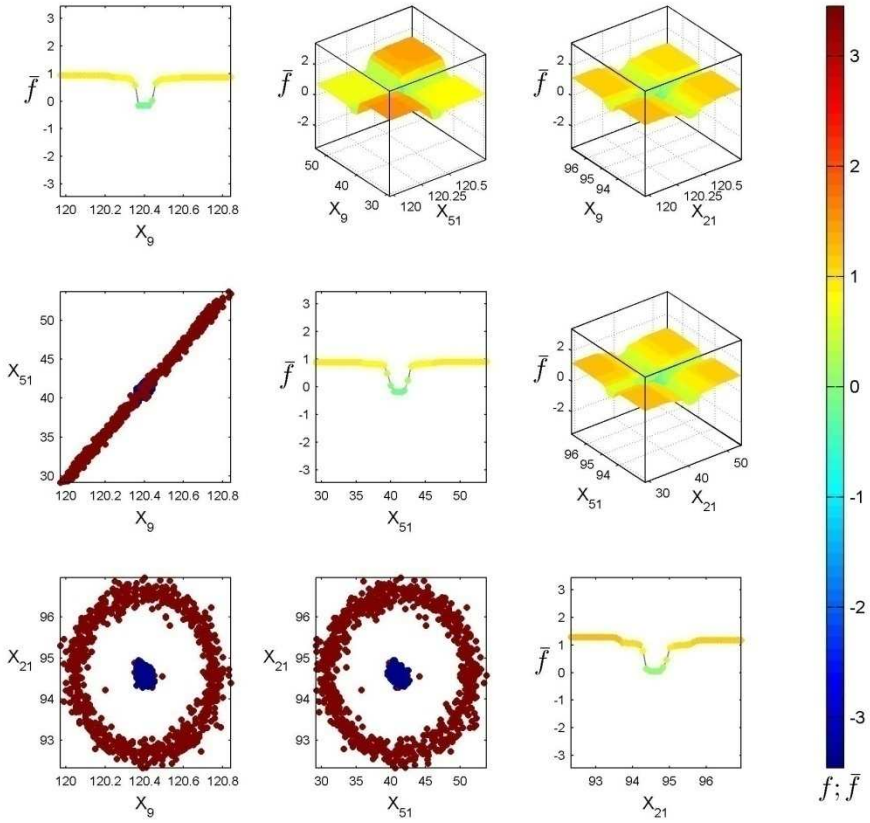

Figure 10.12: Visualization of NOC and fault 14 of the Tennessee Eastman process with partial dependences of the three most important variables. Diagonal plots show one-variable partial dependences, above-diagonal plots show two-variable partial dependences and below-diagonal plots show two-variable scatterplots. The colour range represents the log-odds approximation for the fault class. Negative log-odds suggest classification to the NOC class, while positive log-odds suggest classification to the fault class.

10.5.3 Calcium carbide process fault

It was stated earlier (Chapter 6) that high values for variables 1, 4, 5 and 6 ensure high product grade and tonnages, on which the definition of the fault conditions were based. Variables 2, 3 and 7 are weakly correlated with these product variables (not included in the process data), while variables 8 and 9 are considered to have a negligible influence on the performance of the furnace.

Figure 10.13 gives the random forest variable importance measures for the calcium carbide process fault (low product tonnage), while Figure 10.14 shows the one-variable and two-variable partial dependencies of the three most important variables (as identified by random forest variable importance). From the variable importance plot, variables 1, 4 and 5 were identified as the three most important variables. This agrees with the previous statement. Variable 6, another important variable, exceeds the dummy variable threshold. However variables 2 and 7, which are not considered important, but are correlated with the product tonnage, exceed the dummy variable threshold. It is known (see Chapter 9) that random forest variable importance is sensitive to correlation. The unimportant variables 8 and 9 do not exceed the dummy variable threshold.

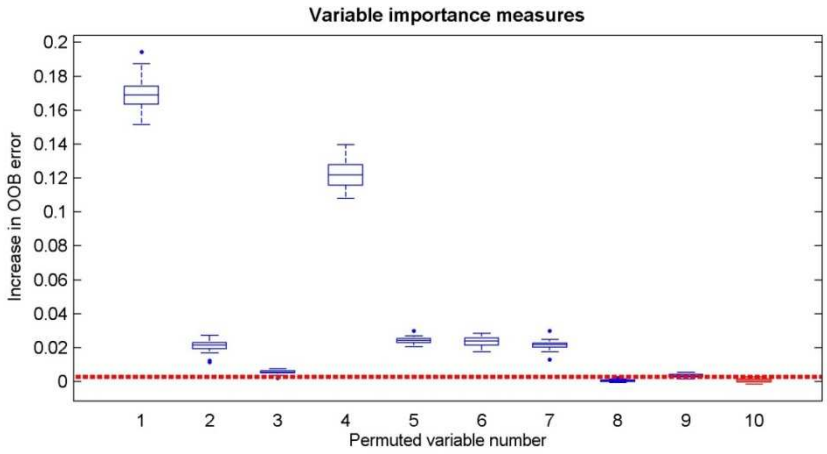

Figure 10.13: Random forest variable importance measures for the fault of the calcium carbide process. The dummy variable is shown in red, with the dashed red line indicating the upper 95% confidence limit of the dummy variable importance measure.

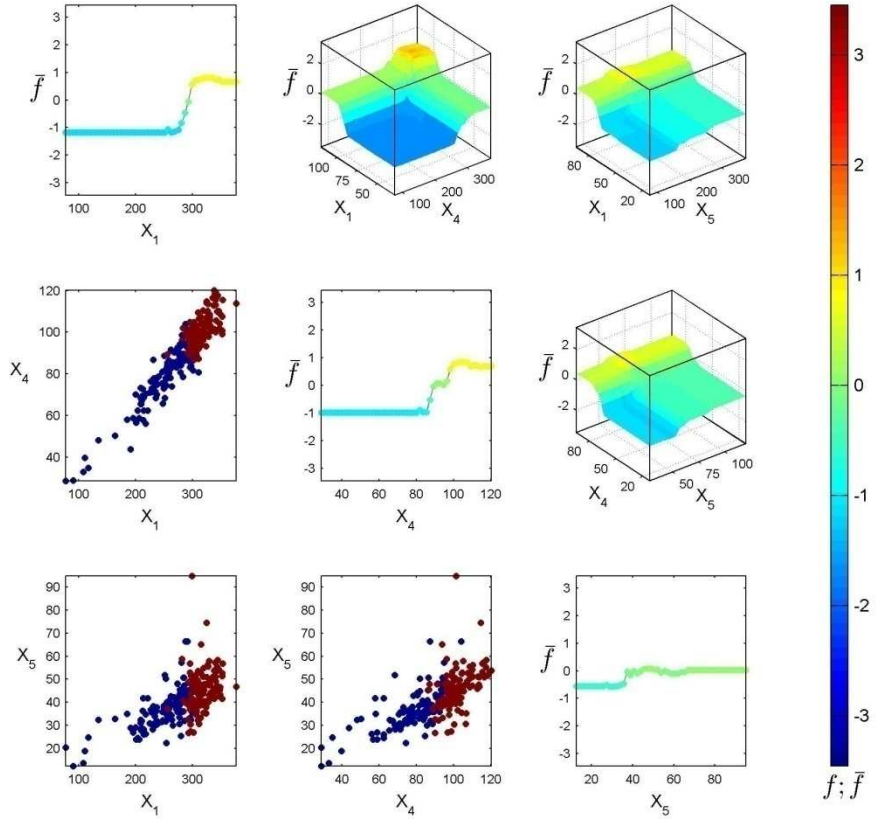

Figure 10.14: Visualization of the calcium carbide process NOC and fault data with partial dependences of the three most important variables. Diagonal plots show one-variable partial dependences, above-diagonal plots show two-variable partial dependences and below-diagonal plots show two-variable scatterplots. The colour range represents the log-odds approximation for the fault class. Negative log-odds suggest classification to the NOC class, while positive log-odds suggest classification to the fault class.

From the one-variable partial dependence plots, the relatively larger influence of variables 1 and 4 can be seen from the range of partial dependences. The threshold values for these variables between NOC and fault conditions are also visible. Even though the influence (or association) of variable 5 is small, interactions between variables 1 and 5, as well as variables 4 and 5, are visible from the two-variable partial dependence

plots. As with fault 14 of the Tennessee Eastman process, areas with low support in the scatterplots correspond to areas in the two-variable partial dependence plots that are shifted towards zero (in this case the shift is upwards, away from high probability of normal operating conditions).

10.6 Conclusions

This study has illustrated the interpretive value of random forest models through the identification of important variables and the visualization of the effect of these important variables on the response. Identifying important variables and illustrating their influence is particularly challenging for nonlinear systems. The nonlinear simulation data case study showed the interpretive ability (up to second-order interaction level) of random forest models, and the failure of linear regression to identify all important variables.

Even where random forests show low overall predictive ability (as measured by pseudo-R^2), the variable importance measures and partial dependences delivered by random forests closely reflected the true underlying structure of the system.

When applying the variable importance and partial dependence methodology to faults from the Tennessee Eastman and calcium carbide processes, the random forest approach correctly identified important variables, and showed nonlinear dependence of the responses on the important input variable.

A novel approach to random forest variable importance thresholding was introduced, which proved successful where unimportant input variables are not correlated (either with other input variables or the response). This threshold technique proved less successful where variables were correlated.

In conclusion, the visualization tool presented in this chapter provides a useful tool for fault identification and interpretation.

Nomenclature

C	variables not included in subset
D	dummy variable indicator
f	predictor model
f_k	modified logit function
$\bar{f}$	partial dependence
K	number of ensemble members
k	class indicator
L	learning set of input variables **X** and output variable **y**
p	proportion of votes
S	variables of interest
$\mathcal{T}$	decision tree function
X	input variables
y	response variable
θ	random vector
ω	variable importance

CHAPTER 11 - CHANGE POINT DETECTION

This chapter presents a framework for detecting changes in time series data with random forest feature extraction. A random forest change point detection algorithm is developed, analogous to the linear approach of singular spectrum analysis.

Random forest change point detection proves to be more successful than singular change point detection in identifying known changes in the investigated case studies. The random forest method is also shown to be more robust to an essential model parameter, the base window extent. Both the SSA and RF diagnostics are not particularly sensitive to noise.

Due to the generation of a proximity matrix, and the training of many regression random forests, random forest change point detection is more computationally expensive than singular spectrum analysis change point detection, to an order of magnitude of three.

As with random forest fault diagnosis, the success of random forest change point detection may relate to the extrapolation inaccuracy of random forest regression forests. Unseen (and presumably change-indicative) variable values have high random forest regression reconstruction errors, and thus high random forest change point detection diagnostics.

11.1 Overview

Fault detection and identification methods, as implemented in the previous studies, do not consider or exploit the dynamic nature of processes. Process data may often contain autocorrelated variables, i.e. variable values are partially dependent on historic variable values. As mentioned in Chapter 2, the assumption of steady state data is not always valid in chemical and other processes, due in part to possible high sampling frequencies and time-varying behaviour (Ku et al., 1995). In terms of exploiting autocorrelation, a number of variants to the original PCA fault diagnosis method have been suggested (Venkatasubramanian et al., 2003b). An extension of PCA incorporating lagged copies of process variables (embedding to capture autocorrelation) was termed dynamic PCA (Ku et al., 1995). To account for changes in the crosscorrelation structure of process variables due to time-varying behaviour such as catalyst deactivation, equipment aging and sensor drift, (Li et al., 2000) suggested recursive PCA. The PCA model is adapted as new process data become available, incorporating new NOC data into the ever-growing training data set. Another approach is moving window PCA (Wang et al., 2005), where a new model is generated on a constant size training set as new data becomes available.

In effect, these modifications aim to capture more (and continually updated) process information in the process data matrix **X** which forms the basis of modeling normal operating conditions with feature extraction techniques. This incorporation of embedding and updating of the original process data in a feature extractive fault diagnosis scheme is shown in Figure 11.1.

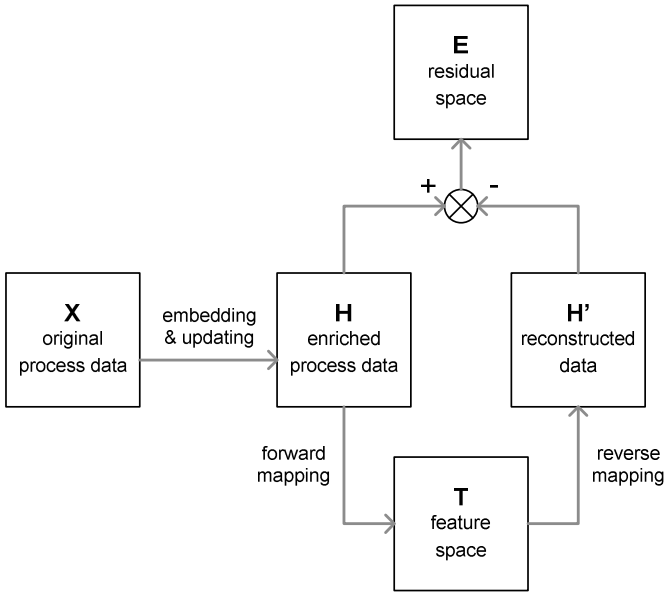

Figure 11.1: Schematic of feature extraction approach to exploit dynamic characteristics of process data in fault detection (H represents the enriched process data matrix)

The framework presented in Figure 11.1 is closely related to change point detection. Change point detection involves determining whether a sudden, possibly significant change has occurred in a system or process. The system is characterized by one or more observed variables which may be correlated. The task of change point detection is then to identify a shift in the process state generating the observed variables.

To identify a change, a description of the preceding state observations is required. This description can be parametric or nonparametric, and is often a challenging task. Parametric methods need extensive a priori knowledge to identify the particular normal state distribution from a family of known distributions. However, once the correct distribution is found, the parametric methods are computationally efficient. Nonparametric methods are not confined to specific distributions, instead deriving normal state definitions from available data. This provides more flexibility in terms of application to wide-ranging data distributions, but often results in more computationally expensive algorithms.

Many monitoring problems can be cast as the detection of a change in the parameters of a static or dynamic system, represented by one or more observed variables. The task of change point detection is then to identify a shift in the process state generating the observed variables, based on a description of the preceding state observations. This can be a challenging task, since the key difficulty is to detect intrinsic changes that are not necessarily directly observed and that are measured together with other types of perturbations.

The extension of the random forest fault diagnosis approach to incorporate embedded and updated data is now cast as a change point detection problem. The aim of the change point detection study is two-fold: to develop a change point strategy employing random forest feature extraction, and to compare the performance of the developed strategy to an existing technique. The random forest approach is constructed analogous to a change point detection methodology previously proposed (Moskvina & Zhigljavsky, 2003): singular spectrum analysis change point detection. The performance of the random forest change point detection scheme will be compared to singular spectrum analysis change point detection.

11.2 Capturing dynamic behaviour by lagging variables

A process variable is often measured in a system of dynamic, interacting variables. As the process variable is one of the outcomes of all the interacting variables, it is reasonable to assume that, given enough samples and

sufficient correlation, this single process variable contains information on the dynamics of all process variables. If one assumes that a dynamic system can be represented by a set of w ordinary differential equations, these equations can be expressed as a single differential equation of the order w. The measured process variable, however, is not a continuous differentiable entity. Rather, by creating w lagged copies of the discrete process variable, this shifting corresponds to first-order differencing of the discrete process variable. These lagged copies span what is known as phase space, and is often a good approximation of the state space of a dynamic system (Elsner & Tsonis, 1996).

The construction of a lagged trajectory matrix **H** (Hankel matrix) for a univariate variable **X** of N elements is shown in the following equation: (with *w* lags)

$$\mathbf{H} = \begin{bmatrix} x_1 & x_2 & \cdots & x_w \\ x_2 & x_3 & \cdots & x_{w+1} \\ \cdots & \cdots & \cdots & \cdots \\ x_{N-w+1} & x_{N-w+2} & \cdots & x_N \end{bmatrix} \qquad \textit{Eqn. 42}$$

11.3 Change point detection techniques

The random forest approach to change point detection in time series data is a nonlinear counterpart of an approach previously described (Moskvina & Zhigljavsky, 2003). Their approach is based on the use of PCA and for comparative purposes; both algorithms are discussed and will be compared on the mentioned data sets.

11.3.1 Random forest change point detection

The random forest change point detection algorithm constructs a subspace $\Re^a$ of a extracted features for a moving window of normal process or reference conditions, and measures the distance of new test data to this reference subspace. This can be done in three steps, viz. construction of an a-dimensional subspace, construction of a test matrix and computation of suitable test statistics or diagnostic sequences.

The embedding of the time series to create the base matrix and subsequent feature extraction and mapping are illustrated in Figure 11.2.[20] The definitions of the base and test matrices are given in Figure 11.3 and Figure 11.4. Given a stationary multivariate time series of observations, a window of width w is slid across the time series data.

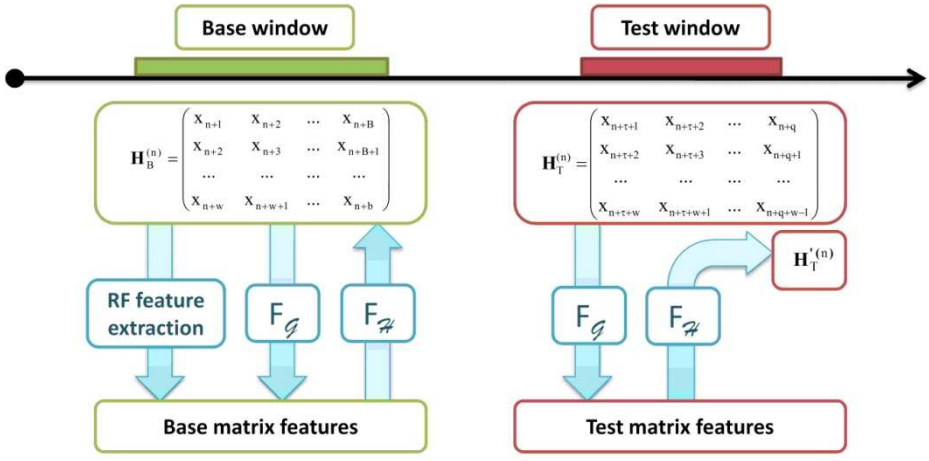

Figure 11.2: Random forest change point detection scheme

[20] *A Hankel matrix can be constructed for a multivariate time series by concatenating, column-wise, the Hankel matrix of each variable of the multivariate time series.*

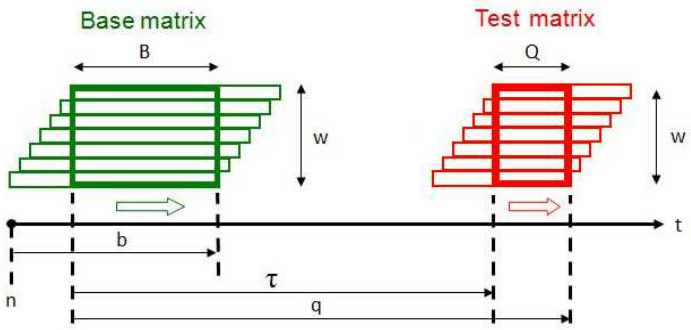

Figure 11.3: Shifting base and test matrices for random forest change point detection.

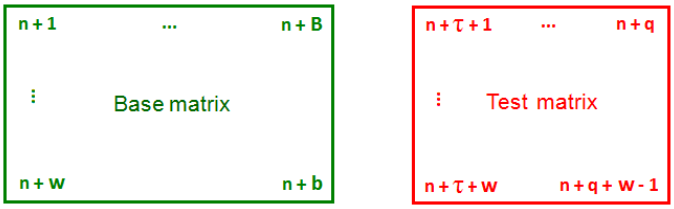

Figure 11.4: Base and test matrix coordinates for random forest change point detection, where w = lag parameter, b = extent of base matrix, τ = time lag between test and base matrices, B = size of base matrix (number of columns) = b – w + 1, n = iteration number, q = test matrix termination index, Q = size of test matrix (number of columns) = q – τ.

For the first step, the aim of construction of a-dimensional space is that the time series data in the normal operating region of the process is represented by a low-dimensional set of features with a set of random forest models ($F_{\mathcal{G}}$) to map new data to this feature space, as well as a different set of random forest models ($F_{\mathcal{A}}$) to reconstruct the time series data from the mapped features. A subsample validation matrix is left out of the training data, in order to account for the generalization error of the random forest models.

In the second step, the test matrix is constructed similar to the base matrix, with only the number of samples and position in the time series that differ.

The third step involves the calculation of test statistics. The sum of squared Euclidian distances of test matrix columns $H_j^{(n)}$ to the a-dimensional space defined by the base matrix serves as a detection statistic from which a diagnostic sequence can be built. This distance metric for the test set is scaled by the distance metric for the validation set, to account for generalization errors. Large values of the diagnostic sequence indicate a change in the structure of the time series.

To construct a threshold, it is assumed that a sample of normal operating condition data of sufficient sample size is available. The diagnostic sequence for these data is calculated and the upper threshold defined as an α % percentile of this sequence.

The test statistic is the ratio of, on the one side, the mean sum of square residuals between the test lag matrix and reconstructed test lag matrix, and on the other, the mean sum of square residuals between the validation base lag matrix and reconstructed validation base lag matrix. Given that the proposed method is an unsupervised approach, certain assumptions are inevitable. For the RF diagnostic, it is assumed that the first 2x*b*+*w* instances of the time series can be considered as representing conditions where no change has occurred. This assumption should be considered during data preprocessing, and the selection of the change detection technique to use.

When the assumption is valid, the next question is the actual value of the threshold. One could determine the distribution to which the diagnostic statistic would belong, estimate the parameters of said distribution, and select a threshold to represent a certain Types I and II error range. As the base window cannot be excessively large (to avoid smoothing over changes), only a restricted sample size is available to estimate the distribution parameters. To circumvent the issues of picking a distribution and collecting enough samples to estimate the distribution parameters accurately, a simple percentile approach is used. The relative distance metric for the first 2x*b*+*w* instances of the time series is calculated and the upper threshold defined as an 1-α % percentile of this sequence.

Significant changes in time series structure will be detected for any reasonable choice of parameters. To detect a small change in noisy data, tuning of parameters may be required. In terms of the choice of lag w, the recommendation (Moskvina & Zhigljavsky, 2003) is to choose w = b/2. Another setting to tune is the length and location of test sample τ, q. It suggested that τ ≥ B. If the difference between τ and q is too large, then the behaviour of e_T becomes too smooth. If it is too small, e_T is sensitive to noise (Moskvina & Zhigljavsky, 2003). Choice of window width b depends on what kind of structural changes is searched for. If b is too large, changes can be missed or smoothed out. If b is too small, an outlier may register as a structural change. To obtain a robust diagnostic sequence, three parameters settings are used, and the average inspected. The three settings as used in a previous study (Moskvina & Zhigljavsky, 2003) are applied, as demonstrated in Figure 11.5.

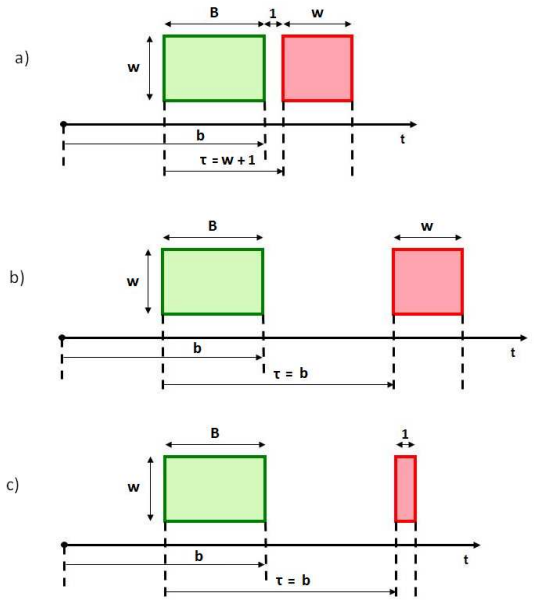

Figure 11.5: Test matrix positions for different parameter settings: a) w = b/2; τ = w+1; q = b+1; b) w = b/2; τ = b; q = b+w; c) w = b/2; τ = b; q = b+1

The random forest change point detection algorithm is summarized here:

Random forest change point detection algorithm

For time step n = 1 to N-w+1:

- Construction of a-dimensional space:
  - A lagged trajectory matrix (Hankel matrix) $\mathbf{H}_B$ for the reference conditions is constructed, with w lags across the time interval [n+1, n+b]
    - $\mathbf{H}_B^{(n)} = \begin{bmatrix} x_{n+1} & x_{n+2} & \cdots & x_{n+B} \\ x_{n+2} & x_{n+3} & \cdots & x_{n+B+1} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n+w} & x_{n+w+1} & \cdots & x_{n+b} \end{bmatrix}$ ***Eqn. 43***
  - The columns of the base matrix are labelled $H_j^{(n)}$
    - $H_j^{(n)} = \left(x_{n+j}, x_{n+j+1}, \ldots, x_{n+j+w-1}\right)^T$ with j = 1, 2, …, B ***Eqn. 44***
  - Random forest feature extraction is employed to calculate a features $\mathbf{T}_B^{(n)}$
  - Mapping and demapping models are trained:
    - A set (F_g) of a random forest regression models to map directly from $\mathbf{H}_B^{(n)}$ to $\mathbf{T}_B^{(n)}$
    - A set ($F_{g'}$) of w random forest regression models to demap directly from $\mathbf{T}_B^{(n)}$ to $\mathbf{H'}_B^{(n)}$, the reconstructed Hankel matrix
- Construction of test matrix:
  - A test matrix of size w x Q is constructed for the testing conditions
    - $\mathbf{H}_T^{(n)} = \begin{bmatrix} x_{n+\tau+1} & x_{n+\tau+2} & \cdots & x_{n+q} \\ x_{n+\tau+2} & x_{n+\tau+3} & \cdots & x_{n+q+1} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n+\tau+w} & x_{n+w+\tau+1} & \cdots & x_{n+q+w-1} \end{bmatrix}$ ***Eqn. 45***
  - The columns of the base matrix are labelled $H_j^{(n)}$
    - $H_j^{(n)} = \left(x_{n+j}, x_{n+j+1}, \ldots, x_{n+j+w-1}\right)^T$ with j = τ+1, τ+2, …, τ+Q ***Eqn. 46***
- Computation of test statistics:
  - Project the test matrix to the nonlinear manifold specified with F_g
  - Calculate the reconstructed test matrix with $F_{g'}$
  - Calculate the distance metric:
    - $e_T = \sum_{j=\tau+1}^{q} \left\| H_j^{(n)} - H'^{(n)}_j \right\|$ ***Eqn. 47***
    - With $H'^{(n)}_j$ the reconstructions of $H_j^{(n)}$
  - Normalize in terms of the test matrix size
    - $\tilde{e}_T = \frac{1}{wQ} e_T$ ***Eqn. 48***
  - Standardize in terms of validation metric
    - $\varepsilon = \frac{\tilde{e}_T}{\tilde{e}_V}$ ***Eqn. 49***
    - The validation metric $\tilde{e}_V$ is calculated on a fraction of base matrix samples that are left out of the training procedure for the mapping and demapping forests F_g and $F_{g'}$
  - Define an upper threshold ε_{crit} from data with no change, using a percentile approach

11.3.2 Singular spectrum analysis change point detection

The singular spectrum analysis change point detection and random forest change point detection algorithms are similar in that two segments of the multivariate time series are likewise embedded into a base and a test matrix. The SSA approach differs from the random forest model, in that principal component analysis or singular value decomposition is used to embed the data (Moskvina & Zhigljavsky, 2003). As a result, it is not necessary to construct forward and reverse models to map new data to the feature space (principal

components) or to reconstruct time series data from the feature space, such as required by the random forest approach ($F_{\mathcal{G}}$ and $F_{\mathcal{H}}$).

Feature extraction by singular value decomposition is not based on a dissimilarity matrix (as for random forests), but the covariance matrix ($\mathbf{\Sigma} \in \Re^{wxw}$) of the embedded data or base trajectory matrix, where $\mathbf{\Sigma} = \mathbf{H}\mathbf{H}^T$ **[*Eqn. 50*]**. Singular value decomposition (SVD) of the lag covariance matrix provides w eigenvalues ($\lambda_1 \geq \ldots \geq \lambda_w \geq 0$) and w eigenvectors ($U_1, \ldots, U_w$). SVD of $\mathbf{H}^T\mathbf{H}$ yields w temporal principal components ($V_1, \ldots, V_w$), which can also be calculated with:

$$V_i = \mathbf{H}^T U_i \qquad \textit{Eqn. 51}$$

The lag matrix **H** can now be expressed as $\mathbf{H} = \mathbf{H}_1 + \mathbf{H}_2 + \ldots + \mathbf{H}_w$ **[*Eqn. 52*]**, where:

$$\mathbf{H}_i = \sqrt{\lambda_i} U_i V_i^T \qquad \textit{Eqn. 53}$$

Group the w indices into two groups and sum the matrices $\mathbf{H}_i$ within each group:

$$I = (i_1, \ldots, i_a) \text{ and } \bar{I} = (i_{1+a}, \ldots, i_w)$$

$$\mathbf{H} = \mathbf{H}_I + \mathbf{H}_{\bar{I}} \quad \text{with } \mathbf{H} = \sum_{i \epsilon I} \mathbf{H}_i \text{ and } \mathbf{H} = \sum_{i \epsilon \bar{I}} \mathbf{H}_i \qquad \textit{Eqn. 54}$$

This grouping implies the selection of an a-dimensional subspace in the w-dimensional space $\Re^w$ of vectors H_j.

The sum of squared Euclidian distances of test matrix columns $H_j^{(n)}$ to the a-dimensional space defined by the base matrix serves as a detection statistic. Since the a eigenvectors spanning the $\Re^a$ space are orthonormal, the summed squared distances metric reduces to:

$$e_T = \sum_{j=\tau+1}^{q} \left[\left(H_j^{(n)}\right)^T H_j^{(n)} - \left(H_j^{(n)}\right)^T \mathbf{U}\mathbf{U}^T H_j^{(n)} \right] \qquad \textit{Eqn. 55}$$

With **U** the w x a matrix of eigenvectors spanning the a-dimensional space previously obtained.

The detection threshold is determined in the same way as for random forests, and the same parameter selection method applied, with the exception that no standardization in terms of validation data is done.

The singular spectrum analysis change point detection algorithm is summarized here:

Singular spectrum analysis change point detection algorithm

For time step n = 1 to N-w+1:

- *Construction of l-dimensional space:*
  - *A lagged trajectory matrix (Hankel matrix) $\boldsymbol{H}_B$ for the reference conditions is constructed, with w lags across the time interval [n+1, n+b]*
    - $\boldsymbol{H}_B^{(n)} = \begin{bmatrix} x_{n+1} & x_{n+2} & \cdots & x_{n+B} \\ x_{n+2} & x_{n+3} & \cdots & x_{n+B+1} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n+w} & x_{n+w+1} & \cdots & x_{n+b} \end{bmatrix}$ ***from Eqn. 43***
  - *The columns of the base matrix are labelled $H_j^{(n)}$*
    - $H_j^{(n)} = \left(x_{n+j}, x_{n+j+1}, \ldots, x_{n+j+w-1}\right)^T$ *with j = 1, 2, …, B;* ***from Eqn. 44***
  - *Construct the covariance matrix $\boldsymbol{\Sigma}_n = \boldsymbol{H}_B^{(n)} (\boldsymbol{H}_B^{(n)})^T$* ***[from Eqn. 35]*** *and apply singular value*

*decomposition to determine **U***

- *Construction of test matrix:*
  - *A test matrix of size w x Q is constructed for the testing conditions*
    - $\boldsymbol{H}_T^{(n)} = \begin{bmatrix} x_{n+\tau+1} & x_{n+\tau+2} & \cdots & x_{n+q} \\ x_{n+\tau+2} & x_{n+\tau+3} & \cdots & x_{n+q+1} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n+\tau+w} & x_{n+w+\tau+1} & \cdots & x_{n+q+w-1} \end{bmatrix}$ ***from Eqn. 45***
  - *The columns of the base matrix are labelled $H_j^{(n)}$*
    - $H_j^{(n)} = \left(x_{n+j}, x_{n+j+1}, \ldots, x_{n+j+w-1}\right)^T$ *with j = τ+1, τ+2, …, τ+Q;* ***from Eqn. 46***
- *Computation of test statistics:*
  - *Calculate the distance metric:*
    - $e_T = \sum_{j=\tau+1}^{q} \left[\left(H_j^{(n)}\right)^T H_j^{(n)} - \left(H_j^{(n)}\right)^T \boldsymbol{U}\boldsymbol{U}^T H_j^{(n)} \right]$ ***from Eqn. 55***
    - *With **U** the w x a matrix of eigenvectors from singular value decomposition performed earlier*
  - *Normalize in terms of the test matrix size (no validation set used in the original SSA approach)*
    - $\varepsilon = \frac{1}{wQ} e_T$ ***from Eqn. 49***
  - *Define an upper threshold ε_{crit} from data with no change, using a percentile approach*

11.4 Change point detection methodology

The RF and SSA change point detection algorithms are to be compared based on their application to four simple simulated data sets with change points, and two more complex simulated chemical systems. The selection of parameters and data sets for application are discussed here.

11.4.1 Parameter selection

For both algorithms, the parameters are selected as follows: The choice of the extent of the base window (b) is based on visual inspection of the time series, to ensure a sufficiently long enough stretch to capture normal variation, but sufficiently short enough to exclude possible changes. The lag parameter (w) is calculated as proposed in a previous study (Moskvina & Zhigljavsky, 2003) as w = b/2. The parameters τ and q are likewise calculated, as recommended in said study (Moskvina & Zhigljavsky, 2003). The sensitivity of the algorithms to changes in b will also be investigated.

Due to the computational expenses of growing multiple sets of $F_{\mathcal{J}}$ and $F_{\mathcal{H}}$ for every sample point, the number of random forest feature extracted is restricted to two. It will be shown that this still provides accurate reconstructions. The number of eigenvectors to apply for reconstruction for the singular spectrum analysis is determined to account for at least 90% cumulative variance on a training set representing normal conditions.

11.4.2 Data sets

Seven data sets are considered in this study: a collection of four simple simulated data sets, simulated Belousov-Zhabotinsky reaction data and simulated autocatalytic reactions data.

- Simple simulated data sets

Four simple systems (T_1, T_2, T_3, T_4) are considered. The first three (T_1-T_3) represent univariate time series observations and the fourth (T_4) multivariate time series. The systems are briefly described below.

T_1: 500 samples, of which the first 250 are identically, independently distributed Gaussian data, with zero mean and unit variance. The last 250 samples have a mean of unity and unit variance to simulate an abrupt mean shift in the data.

T_2: 500 samples, of which the first 250 are identically, independently distributed Gaussian data, with zero mean and unit variance. The last 250 samples have a mean of 0 and a variance of 2 to simulate an abrupt shift in the variance of the data.

T_3: 500 samples of data generated by an autoregressive process of the form x(t+1) = ax(t) + ε(0,0.1), where ε has a zero mean Gaussian distribution with a variance of 0.1. In the first 250 samples, parameter a = 0.9 and in the 2nd 250 samples, a = 0.5 to simulate an abrupt shift in the autocorrelation of the data.

T_4: Two cross-correlated time series of 500 samples each. The first 250 time series samples have a covariance matrix of

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \textit{Eqn. 56}$$

while the last 250 observations have a covariance matrix of

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix} \qquad \textit{Eqn. 57}$$

This simulates an abrupt shift in the correlation between the variables.

- Belousov-Zhabotinsky reaction

The Belousov-Zhabotinsky (BZ) reaction is an unstable chemical reaction that maintains self-oscillations and propagating waves, which may display chaos under certain conditions. A simplified model (Gyorgyi & Field, 1991) is considered here. (See Belousov-Zhabotinsky equation sets given below).

The reaction model is simulated with the ODE45 subroutine in Matlab™ with initial values of x_0 = 0.0099, z_0 = 2.2001 and v_0 = 0.4582. After the first 1000 stationary data points are obtained, a fault condition is introduced in the form of a slow drift in the flow rate k_f, from 4.5x10^{-4} s^{-1} to 5.0x10^{-4} s^{-1} over the next 1000 observations. For the last 1000 observations, the parameter k_f is again kept constant.

In order to visualize the dynamics of the reaction system and the simulated parameter drift, principal component scores can be calculated for the lagged trajectory matrix of the entire time series. The vertical legend on the right hand side of Figure 11.6 indicate the history of the system, with dark blue indicating older systems states, and red indicating more recent system states. The change in the parameters of the system resulted in an outward shift of the trajectory.

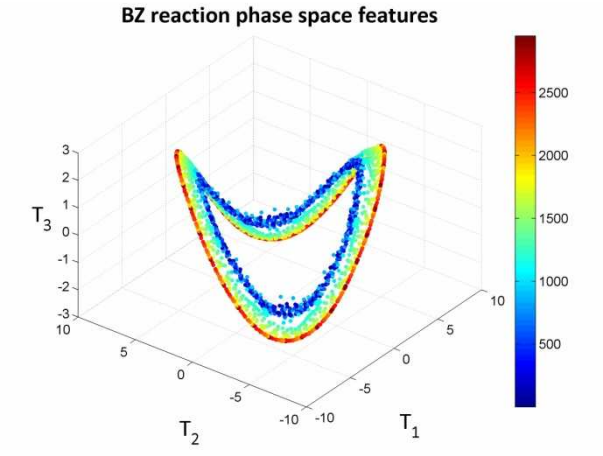

Figure 11.6: Principal component features of lagged trajectory matrix of BZ reaction data

Equation set: Belousov-Zhabotinsky reactions and rate equations

Reactions:

$$X + Y + H \rightarrow 2V \qquad Y + A + 2H \rightarrow V + X \qquad 2X \rightarrow V$$

$$\tfrac{1}{2}X + A + H \rightarrow X + Z \qquad X + Z \rightarrow \tfrac{1}{2}X \qquad V + Z \rightarrow Y$$

$$Z + M \rightarrow products$$

In the previous reactions, A = BrO_3^-, H = H^+ and M = MA are chemical species with constant concentrations, while the concentrations of Y = Br^-, X = $HBrO_2$, Z= Ce(IV) and V = BrMA are the model variables.

Rate equations:

$$r_1 = k_1[H][X][Y], \; k_1 = 4.0\times10^6 \qquad r_2 = k_2[A][H]^2[Y], \; k_2 = 2$$

$$r_3 = k_3[X]^2, \; k_3 = 3000$$

$$r_4 = k_4[A]^{0.5}[H]^{1.5}(C-[Z])[X]^{0.5}, \; k_4 = 55.2$$

$$r_5 = k_5[X][Z], \; k_5 = 7000 \qquad r_6 = \alpha k_6[Z][V], \; k_6 = 0.09$$

$$r_7 = \beta k_7[M][Z], \; k_7 = 0.23$$

Equation set: Belousov-Zhabotinsky differential equations

Differential equations (assuming Y is a fast variable):

$$\frac{dx}{d\tau} = T_0\left[\begin{array}{l} -k_1HY_0x\breve{y} + \dfrac{k_2AH^2Y_0}{X_0}\breve{y} - 2k_3X_0x^2 + \ldots \\ \ldots\dfrac{1}{2}k_4A^{0.5}H^{1.5}X_0^{-0.5}(C - Z_0z)x^{0.5} - \dfrac{1}{2}k_5Z_0xz - k_fx \end{array}\right]$$

$$\frac{dz}{d\tau} = T_0\left[k_4A^{0.5}H^{1.5}X_0^{0.5}\left(\frac{C}{Z_0} - z\right)x^{0.5} - k_5X_0xz - \alpha k_6V_0zv - \beta k_7Mz - k_fx\right]$$

$$\frac{dv}{d\tau} = T_0\left[\frac{2k_1HX_0Y_0}{V_0}x\breve{y} + \frac{k_2AH^2Y_0}{V_0}\breve{y} + \frac{k_3X_0^2}{V_0}x^2 - \alpha k_6Z_0zv - k_fv\right]$$

where τ = time/T_0 and $T_0 = (10k_2AHC)^{-1}$ are the scaled time with scaling factors: x = X/X_0, $X_0 = k_2AH^2/k_5$, y = Y/Y_0, $Y_0 = 4k_2AH^2/k_5$, z = Z/Z_0, $Z_0 = CA/(40M)$, v = V/V_0 and $V_0 = 4AHC/M^2$, which are the scaled concentration variables. The approximation of the fast variable Y is:

$$\breve{y} = \frac{\left[\dfrac{\alpha k_6Z_0V_0zv}{k_1HX_0x + k_2AH^2 + k_f}\right]}{Y_0}$$

*where C is the total cerium ion concentration and α, β and k_f are adjustable parameters. Chaotic behaviour exists for these chemical conditions at A = 0.1*M*, M = 0.25*M*, H = 0.26*M*, C = 0.000833*M*, α = 6000/9, β = 8/23 for different windows of k_f, which is the flow rate.*

- Autocatalytic reaction

The autocatalytic process considered in this simulation consists of two parallel, isothermal autocatalytic reactions taking place in a continuous stirred tank reactor (Lee & Chang, 1996). The system is capable of producing self-sustained oscillations based on cubic autocatalysis with catalyst decay at certain parameters.

The system is simulated with the ODE45 subroutine in MATLAB™. For the first 1000 observations the same parameters as specified by Lee and Chang (1996) are used. A simulated fault condition is then introduced through a slight increase in the feed concentrations of *A* and *D*, reflected in an increase in the parameters γ_1 and γ_2. The parameters γ_1 and γ_2 are allowed to drift slowly from their starting values, γ_1 = 1.5 and γ_2 =4.2 to γ_1 = 1.55 and γ_2 =4.25, after which the parameters are kept constant at their new values, for another 1000 observations.

Equation set: Autocatalytic reactions

Reactions:

$$A + 2B \rightarrow 3B$$
$$B \rightarrow C$$
$$D + 2B \rightarrow 3B$$

where A, B and C are participating chemical species

Rate equations:

$$-r_A = k_1 c_A c_B^2$$
$$r_C = k_2 c_B$$
$$-r_D = k_3 c_D c_B^2$$

where k_1, k_2 and k_3 are the rate constants for the chemical reactions and c_A, c_B and c_D are concentrations of species A, B and D respectively.

Dimensionless numbers:

$$X = \frac{C_A}{C_{A,0}}, \qquad Y = \frac{C_D}{C_{D,0}}, \qquad Z = \frac{C_B}{C_{B,0}}$$
$$a = \frac{k_1 C_{B,0}^2 V}{Q}, \qquad b = \frac{k_3 C_{B,0} V}{Q}, \qquad c = \frac{k_2 V}{Q}$$
$$\gamma_1 = \frac{C_{A,0}}{C_{B,0}}, \qquad \gamma_2 = \frac{C_{D,0}}{C_{B,0}}, \qquad \tau = \frac{t.Q}{V}$$

Differential equations:

$$\frac{dX}{dt} = 1 - X - aXZ^2$$
$$\frac{dY}{dt} = 1 - Y - bYZ^2$$
$$\frac{dZ}{dt} = 1 - (1+c)Z + \gamma_1 aXZ^2 + \gamma_2 bYZ^2$$

Chaotic behaviour is exhibited at a = 18000, b = 400, c = 80; γ_1 = 1.5, γ_2 = 4.2, and with initial conditions X_0 = 0; Y_0 = 0; Z_0 = 0.

Figure 11.7 shows a multivariate embedding of the samples of the *Y* and *Z* variables, where the colour coding gives an indication of the history of the data exemplifying the gradual transition between two equilibrium states (red and blue). As with the BZ reaction, the attractor shifts only slightly.

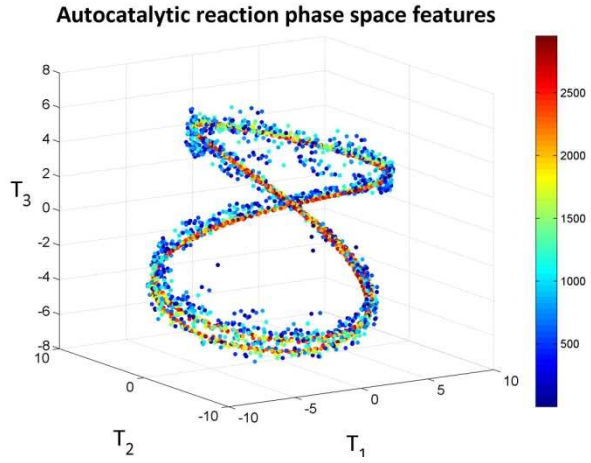

Figure 11.7: Principal component features of lagged trajectory matrix of autocatalytic reaction data

11.5 Results for simple simulated data sets

The change point detection results for the simple data sets (mean shifts, variance shifts, autocorrelation changes and crosscorrelation changes) are now presented. Table 11.1 and Table 11.2 give the parameter sets employed for SSA CPD and RF CPD for the four simple time series. As shown in Table 11.1, the dimensionality of the subspace to which the data were projected in the SSA-based algorithm ranged from 13 to 38, explaining more than 90% of the variance (based on linear reconstruction) in the aggregated trajectory matrix. A significantly lower-dimensional subspace was generated with the random forest model, where two components could explain more than 80% of the variance (based on nonlinear reconstruction) in all four data sets (Table 11.2).

Table 11.1: SSA change point detection parameter values for simple time series (with linear reconstruction correlation values for first iteration Hankel matrices)

Data set	b	w	a	τ_1	τ_2	τ_3	q_1	q_2	q_3	λ_L
T_1	50	25	19	26	50	50	51	75	51	0.908
T_2	50	25	22	26	50	50	51	75	51	0.910
T_3	50	25	13	26	50	50	51	75	51	0.907
T_4	50	25	38	26	50	50	51	75	51	0.900

Table 11.2: RF change point detection parameter values for simple time series (with nonlinear reconstruction correlation values for first iteration Hankel matrices)

Data set	b	w	a	τ_1	τ_2	τ_3	q_1	q_2	q_3	λ_N
T_1	50	25	2	26	50	50	51	75	51	0.834
T_2	50	25	2	26	50	50	51	75	51	0.840
T_3	50	25	2	26	50	50	51	75	51	0.904
T_4	50	25	2	26	50	50	51	75	51	0.849

The performance of the two algorithms on the T_1, T_2, T_3 and T_4 data are given in Figure 11.8, Figure 11.9, Figure 11.10 and Figure 11.11, respectively. In all these figures, the observations themselves are shown on top, while the diagnostic sequences of the SSA and random forest models are shown in the middle and bottom panels respectively. The vertical red lines in the top panels indicate the occurrence of the simulated change. The horizontal broken red lines in the middle and bottom panels represent the detection threshold of each algorithm. In each case, the shown diagnostics are the average values of the diagnostic sequences for the three different parameter settings, as shown in Figure 11.5.

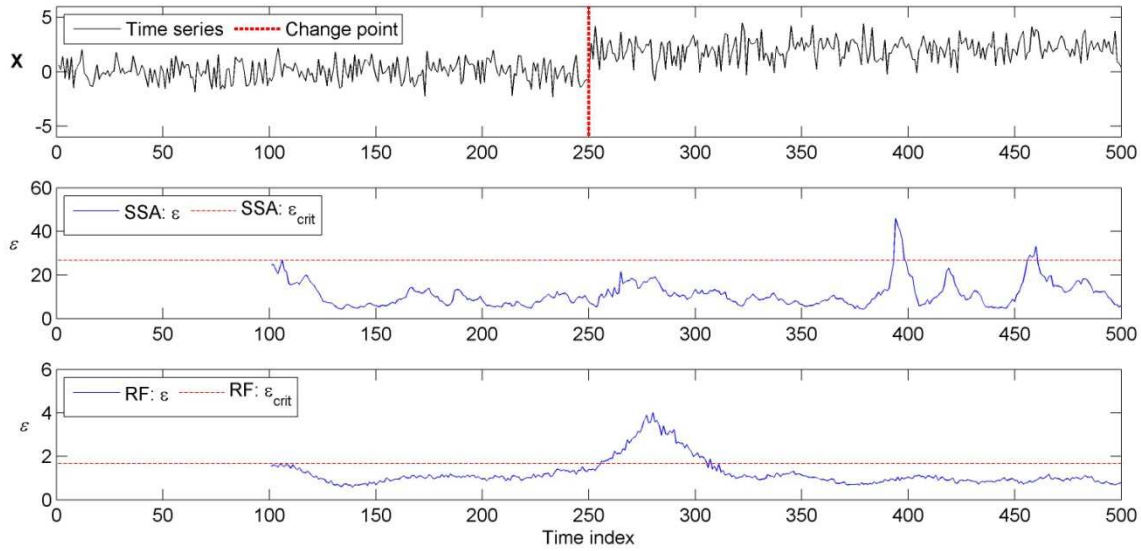

Figure 11.8: Time series and change point detection diagnostic sequences for T_1 (known change shown with horizontal line; average diagnostic sequence in blue; detection threshold in dashed red)

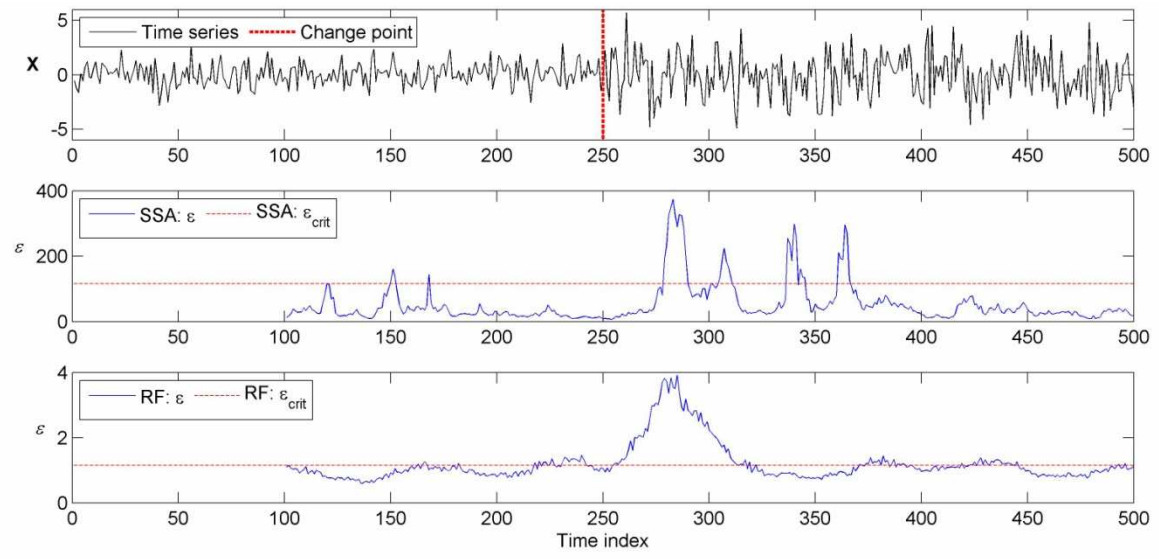

Figure 11.9: Time series and change point detection diagnostic sequences for T_2 (known change shown with horizontal line; average diagnostic sequence in blue; detection threshold in dashed red)

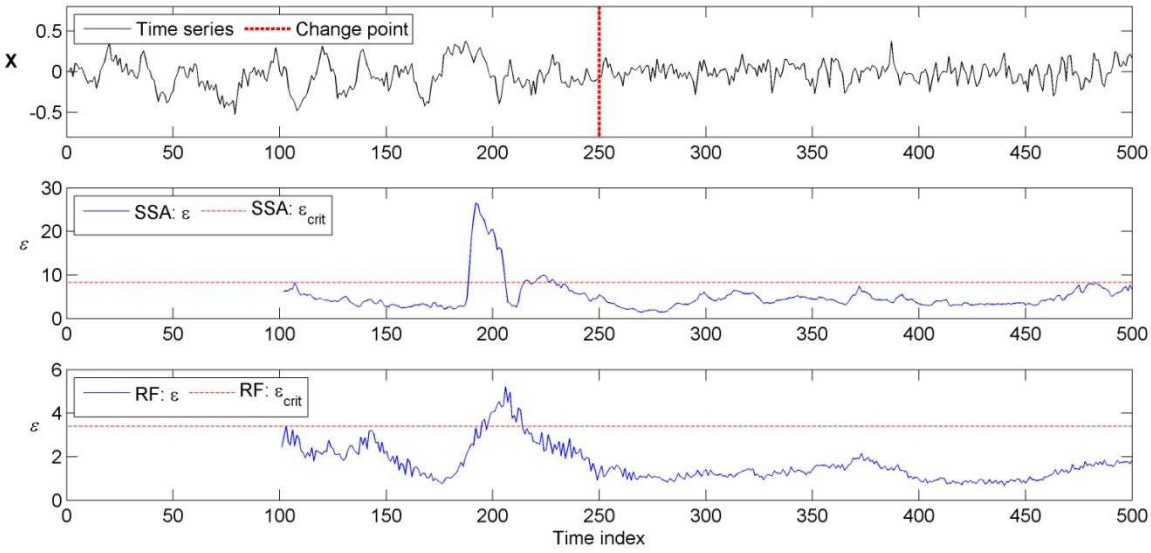

Figure 11.10: Time series and change point detection diagnostic sequences for T_3 (known change shown with horizontal line; average diagnostic sequence in blue; detection threshold in dashed red)

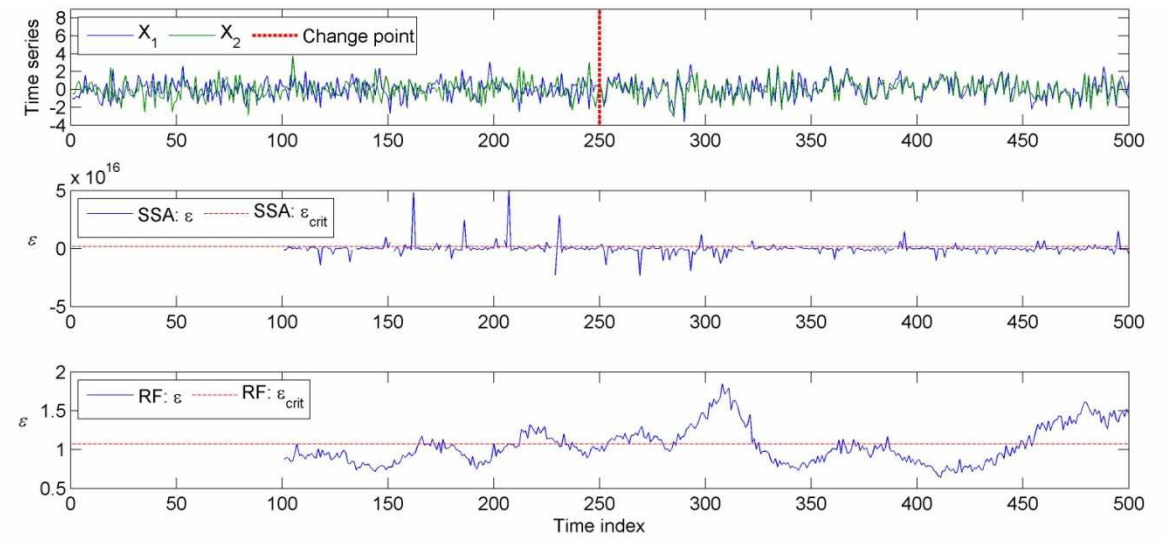

Figure 11.11: Time series and change point detection diagnostic sequences for T_4 (known change shown with horizontal line; average diagnostic sequence in blue; detection threshold in dashed red)

In Figure 11.8, the RF method detects the mean shift soon after its occurrence at 250 time steps, while SSA shows no alarm up until 400 time steps. The RF diagnostic sequence shows a lag of around 60 time steps before returning to levels below the detection threshold.

From Figure 11.9, it is evident that the random forest algorithm detected the variance increase at the time index of 250 earlier than the SSA based algorithm. Both the SSA and random forest diagnostic sequences show a lag before returning to a condition below the detection threshold, once the base window enters the changed section of the time series and the model is updated.

In Figure 11.10, it can be seen that the SSA and RF algorithms both failed to detect the change in the autocorrelation structure of the T_3 data change after the 250th sample. The false alarms at around the 200th sample may be a spurious artifact of the data.

From Figure 11.11, the change in the correlation between the two time series at the 250th sample in data set T_4, was detected by the RF algorithm, but not by the SSA algorithm. The SSA algorithm exhibited erratic behavior, as can be seen from the scale of the vertical axis of the graph in the middle panel.

For these four simple time series, the RF method is able to detect change points for three cases, whereas SSA shows convincing detection for only one time series.

11.6 Results for dynamic reaction simulations

Two complex dynamic systems are considered in this change point detection study: the Belousov-Zhabotinsky reaction and autocatalytic reactions. For each of these systems, changes in the ordinary differential equations describing the data are introduced after 1000 and 2000 time steps of the 3000 time step series. From time step 1000 onward, a linear increase is made to certain model coefficients. From time step 2000 onward, the linear increasing coefficients no longer increase, but remain constant at their values reached at 1999 time steps. The Belousov-Zhabotinsky reaction system is a univariate time series, while the autocatalytic reactions system is a multivariate system with two variables.

Table 11.3 and Table 11.4 give the parameter sets employed for SSA CPD and RF CPD for the two reaction system time series. As in the previous study only two parameters must be defined: the extent of the base window b and the number of features a to extract from the base matrix. The choice of the extent of the base window was based on visual inspection of the time series, and set to 100 time steps. As shown in Table 11.3, the dimensionalities of the subspaces to which the data were projected in the SSA-based algorithm were 3 and 9, for the BZ and autocatalytic systems respectively, explaining 90% of the variance (based on linear reconstruction) in the aggregated trajectory matrix. A significantly lower-dimensional subspace was generated with the random forest model, where two components could explain more than 88% of the variance (based on nonlinear reconstruction) for both data sets (Table 11.4).

Table 11.3: SSA change point detection parameter values for simulated dynamic reaction data sets (with linear reconstruction correlation values for first iteration Hankel matrices)

Data set	b	w	a	τ_1	τ_2	τ_3	q_1	q_2	q_3	λ_L
BZ reaction	100	50	3	51	100	100	101	150	101	0.896
Autocatalytic reactions	100	50	9	51	100	100	101	150	101	0.912

Table 11.4: RF change point detection parameter values for simulated dynamic reaction data sets (with nonlinear reconstruction correlation values for first iteration Hankel matrices)

Data set	b	w	a	τ_1	τ_2	τ_3	q_1	q_2	q_3	λ_N
BZ reaction	100	50	3	51	100	100	101	150	101	0.984
Autocatalytic reactions	100	50	9	51	100	100	101	150	101	0.886

The performance of the two algorithms on the reaction systems data are given in Figure 11.12 and Figure 11.13. In these figures, the observations themselves are shown on top, while the diagnostic sequences of the SSA and random forest models are shown in the middle and bottom panels respectively. The vertical red lines in the top panels indicate the occurrence of the simulated change. The horizontal broken red lines in the middle and bottom panels represent the detection threshold of each algorithm. In each case, the shown

diagnostics are the average values of the diagnostic sequences for the three different parameter settings, as shown in Figure 11.5.

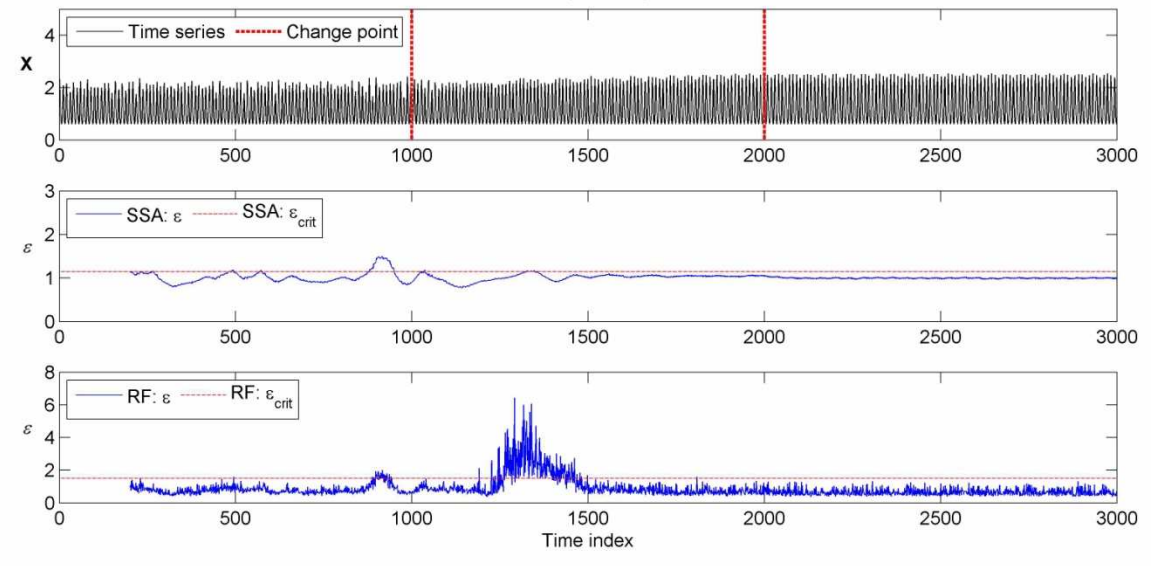

Figure 11.12: Time series and change point detection diagnostic sequences for BZ reaction (known change shown with horizontal line; average diagnostic sequence in blue; detection threshold in dashed red)

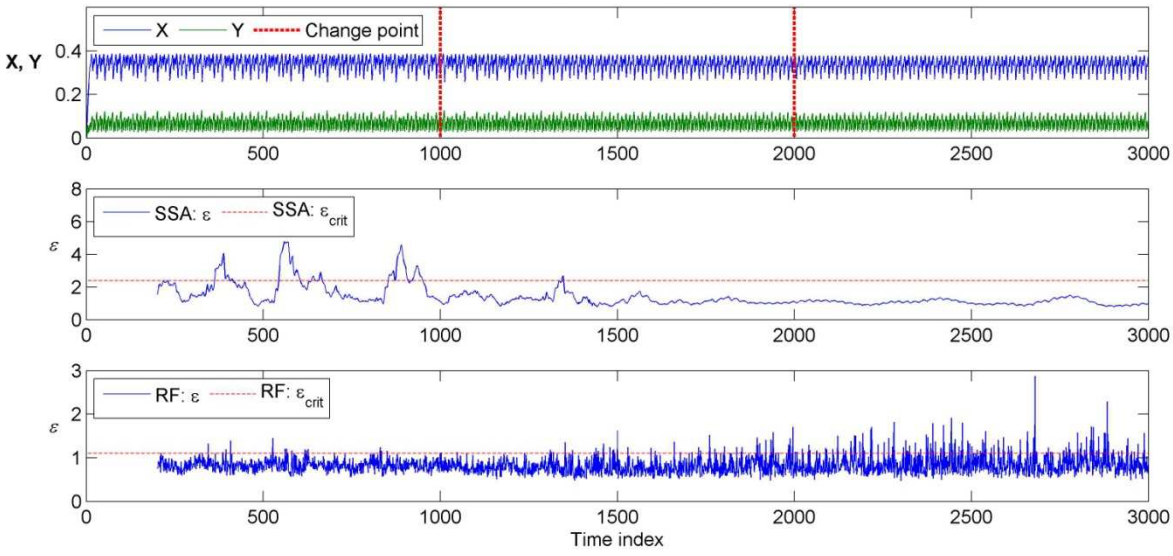

Figure 11.13: Time series and change point detection diagnostic sequences for autocatalytic reaction (known change shown with horizontal line; average diagnostic sequence in blue; detection threshold in dashed red)

Figure 11.12 shows the progression of variable Y in the BZ reaction as a time series (top panel). The middle and bottom panels show the performance of the SSA and RF based algorithms respectively. Only the RF based algorithm could detect the change (at around the 1300th observation). The SSA based algorithm failed to detect any change in the system and instead gave a false alarm at the 900th sample.

Figure 11.13 shows the time series values for the Y and Z variables of the autocatalytic reaction system, with the corresponding diagnostic sequences for SSA and RF approaches. Only the RF-based algorithm could detect

the change (at around the 2000th observation), although the diagnostic is noisy. The SSA-based algorithm failed to detect any change in the system and instead gave several false alarms in the first 1000 samples.

For these two simulated reaction time series, the RF CPD method was successful in detecting (all be it belatedly) changes in the reaction system, whereas the SSA CPD method showed no significant detections.

11.7 Parameter and noise sensitivity

When using the parameter groupings proposed by Moskvina and Zhigljavsky (2003), the only free parameters to estimate are a and b. The number of multiple components required can be accurately determined from reconstruction errors. The choice of b is not as simple. To investigate the effect of b on the SSA and RF diagnostics, different values of b were used for detecting change in time series T_1 from the simple simulated data sets. The resulting diagnostic sequences are given in Figure 11.14 and Figure 11.15.

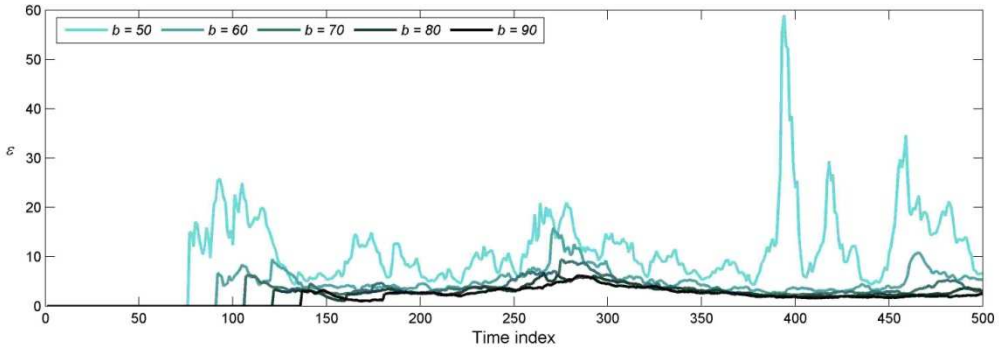

Figure 11.14: Influence of base matrix extent parameter (b) on the average change point detection diagnostic sequence for the SSA approach (based on T_1, using 19 features)

From Figure 11.14, it appears that the SSA diagnostic sequence shows higher variance for smaller base window extents. As the base window extent increases, the diagnostic sequence becomes smoother and more gradual. Here, "gradual" refers to a slow increase in the diagnostic and a corresponding slow decrease. As the base window extent increases, more samples are incorporated in the training of the feature extraction and reconstruction models. This suggests a more stable model, with lower sensitivity to small changes. Lower sensitivity may be a good thing if only large changes need to be detected, but may be a bad thing if small changes are significant.

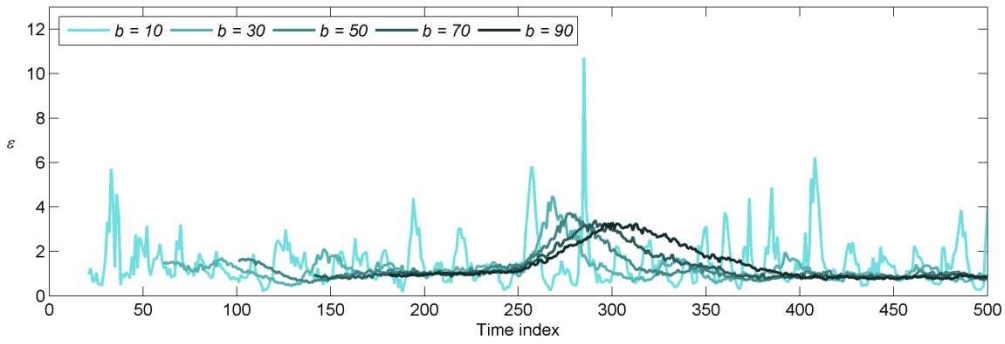

Figure 11.15: Influence of base matrix extent parameter (b) on the average change point detection diagnostic sequence for the RF approach (based on T_1, using 2 features)

From Figure 11.15, it is apparent that a low value for b (10) results in a noisy RF diagnostic sequence. As b increases, the diagnostic sequence becomes less noisy. The probable indicated change points for b-values of

30–90 remain around the actual change point of 250. For this time series, the RF diagnostic sequence is robust to changes in b, bar quite small values of b. Smaller base window extents result in RF diagnostics which increase rapidly and decrease rapidly, while larger base window extents induce slower increases and slower decreases. The point of possible detection, however, stays at 250 time steps. Due to the definitions of the parameter sets, a larger base window implies a larger test window. Larger test windows will include a difference to the base windows for a longer period than smaller test windows. The diagnostic will then remain higher for longer.

Overall, the SSA diagnostic sequence appears more sensitive to changes in b than the RF diagnostic sequence.

To investigate the effect of additive noise on the SSA and RF diagnostics, different levels of Gaussian noise were added to time series T_1 from the simple simulated data sets. Three levels of Gaussian noise, with standard deviations of 10%, 20% and 50% of the difference in means of the signal before and after the change were added. The parameter b was set to 80 for SSA and 50 for RF (to allow for the fact that the SSA diagnostic was more noisy at smaller base window extents, as apparent from Figure 11.14). The results are shown in Figure 11.16 and Figure 11.17.

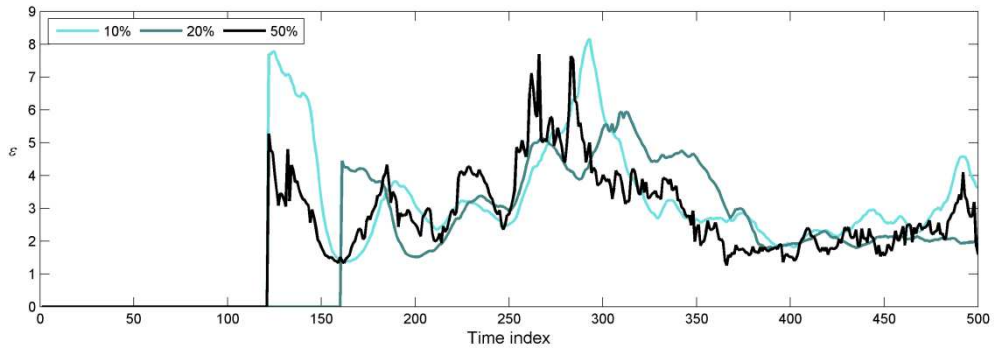

Figure 11.16: Influence of noise on the average change point detection diagnostic sequence for the SSA approach (based on T_1, using 19 features and a base window extent of 80)

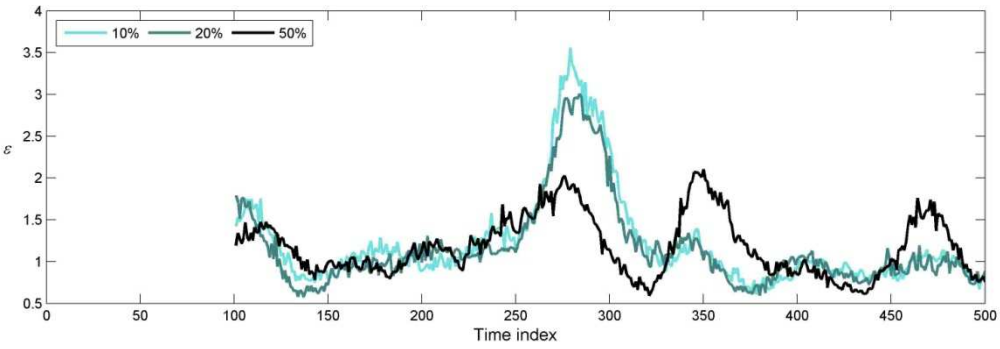

Figure 11.17: Influence of noise on the average change point detection diagnostic sequence for the RF approach (based on T_1, using 2 features and a base window extent of 50)

From Figure 11.16 and Figure 11.17, both the SSA and RF diagnostic sequences appear to maintain their overall structure for the different noise levels. The RF diagnostic seems only to be adversely affected when 50% Gaussian noise is added. This suggests that the RF relative distance diagnostic is, for at least this time series, not sensitive to reasonable noise levels.

11.8 Computational considerations

Figure 11.18 gives the computational times for the SSA and RF change point detection algorithms for a selection of base window extent parameter choices, based on the extraction of two features for the simple time series T_1.

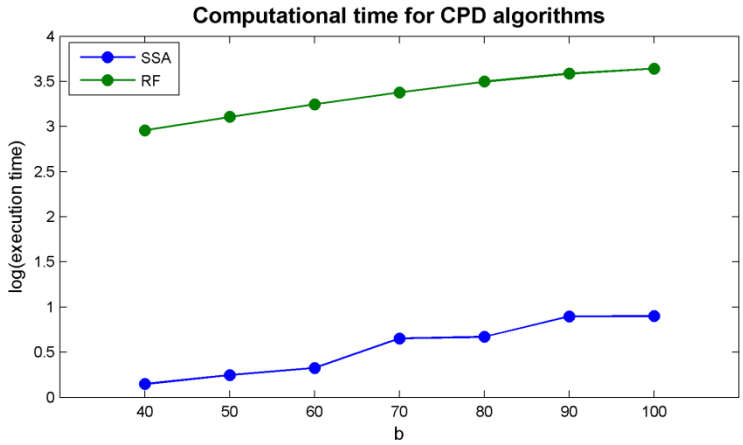

Figure 11.18: Comparison of computational times for SSA and RF change point detection methods for different base matrix extent values (based on T_1 and with each method extracting two features)

The RF approach is at least three orders of magnitude slower than the SSA approach. This is due to the generation and training, at each time step, of a b x b proximity matrix, a mapping regression forests and w demapping regression forests. RF CPD is thus a very computationally expensive method, compared to the SSA approach. For both algorithms, the computational time increases as the base matrix extent increases.

11.9 Discussion

In this chapter, the use of random forest algorithms to detect changes in time series data is compared with the methodology of singular spectrum analysis previously proposed (Moskvina & Zhigljavsky, 2003). The two methodologies are similar in that two windows in series are slid across the time series data being monitored and projections of the data in the leading window are compared with projections of data in the trailing window. However, they differ in that way in which the data are projected and reconstructed for comparative purposes. Among other, singular spectrum analysis is a linear method of projection, based on the covariance matrix of the observed variables and lagged copies of these variables.

In contrast, random forests generate nonlinear projections of the data from a rank based proximity matrix. In principle, this gives an advantage over the linear projections in that these nonlinear projections can approximate more complex data manifolds than linear projections based on linear principal component models.

In the case studies considered in this investigation, the random forest based algorithms performed better than the algorithms based on singular spectrum analysis, especially with regard to the two nonlinear dynamic reaction systems. Unlike the singular spectrum approach, the main drawback of the random forest algorithm is the high computational expense that scales to the power of 2 with an increase in the observation sample size. Despite this, the use of random forests to monitor time series data appears to be promising and justifies further work to be done in this area.

The reason for the success of the random forest change point detection method may be related to the extrapolation characteristics of regression random forests, also applicable to residual distance detection for

random forest fault diagnosis. The RF diagnostic for change point detection is essentially the reconstruction error, or summed residual distances, of the test window data. As shown in Chapter 8, when random forest regression models are presented with input data beyond the range of its training input data, extrapolation performance will be weak. This results in inaccurate reconstructions, and thus larger diagnostics.

Nomenclature

A	BZ and autocatalytic reactant
A	autocatalytic model parameter
$\mathscr{a}$	reduced dimensionality
B	indication of base conditions (subscript)
B	size of base matrix
B	autocatalytic reactant
B	autocatalytic model parameter
b	autocatalytic model parameter
b	extent of base matrix
C	covariance matrix
C	BZ ion concentration and autocatalytic reactant
C	number of classes
c	rate constant
D	autocatalytic model parameter
e	noise variable
$\tilde{e}$	normalized distance metric instance
F	set of random forest regressors
f	simulated time series parameter
$\mathscr{G}$	forward mapping functions (subscript)
g	BZ parameter
H	Hankel matrix
$\mathscr{H}$	reverse mapping functions (subscript)
H	BZ reactant
h	BZ parameter
K	number of ensemble members
m	dimensionality of predictor vector
N	number of observations
n	time interval iteration
Q	autocatalytic system volumetric flow rate
Q	size of test matrix
q	test matrix termination index
r	reaction rate
T	feature matrix
T	indication of test conditions (subscript)
T	simulated time series system
T	BZ variable
t	feature vector
t	time index
U	eigenvector matrix
V	indication of validation conditions (subscript)
V	BZ reactant and autocatalytic volume
v	BZ variable
w	lag parameter
X	predictor matrix
X	BZ reactant
X	autocatalytic dimensionless number
x	predictor vector
x	BZ variable
Y	BZ reactant
Y	autocatalytic dimensionless number
y	response vector
Z	BZ reactant
Z	autocatalytic dimensionless number
z	BZ variable
α	significance level
γ	ratio of species
ε	relative distance metric instance
λ	eigenvalue
μ	BZ and autocatalytic time variable
Σ	size of test matrix
τ	time lag between test and base matrix

CHAPTER 12 - CONCLUSIONS AND RECOMMENDATIONS

12.1 Overview of the contributions of this work

The focus of this work has been the application of unsupervised random forest feature extraction and other random forest interpretation tools to fault diagnosis. To this end, a fault diagnosis framework using random forest features was designed, incorporating many novel aspects. The fault diagnosis framework was further extended and modified to exploit dynamic characteristics of data, and from this a change point detection framework was created. The random forest fault diagnosis and change point detection frameworks developed here have been published in peer-reviewed journals (Auret & Aldrich, 2010a; Auret & Aldrich, 2010b). Figure 12.1 shows an overview of the new techniques, applications, insights and comparisons presented in this dissertation.

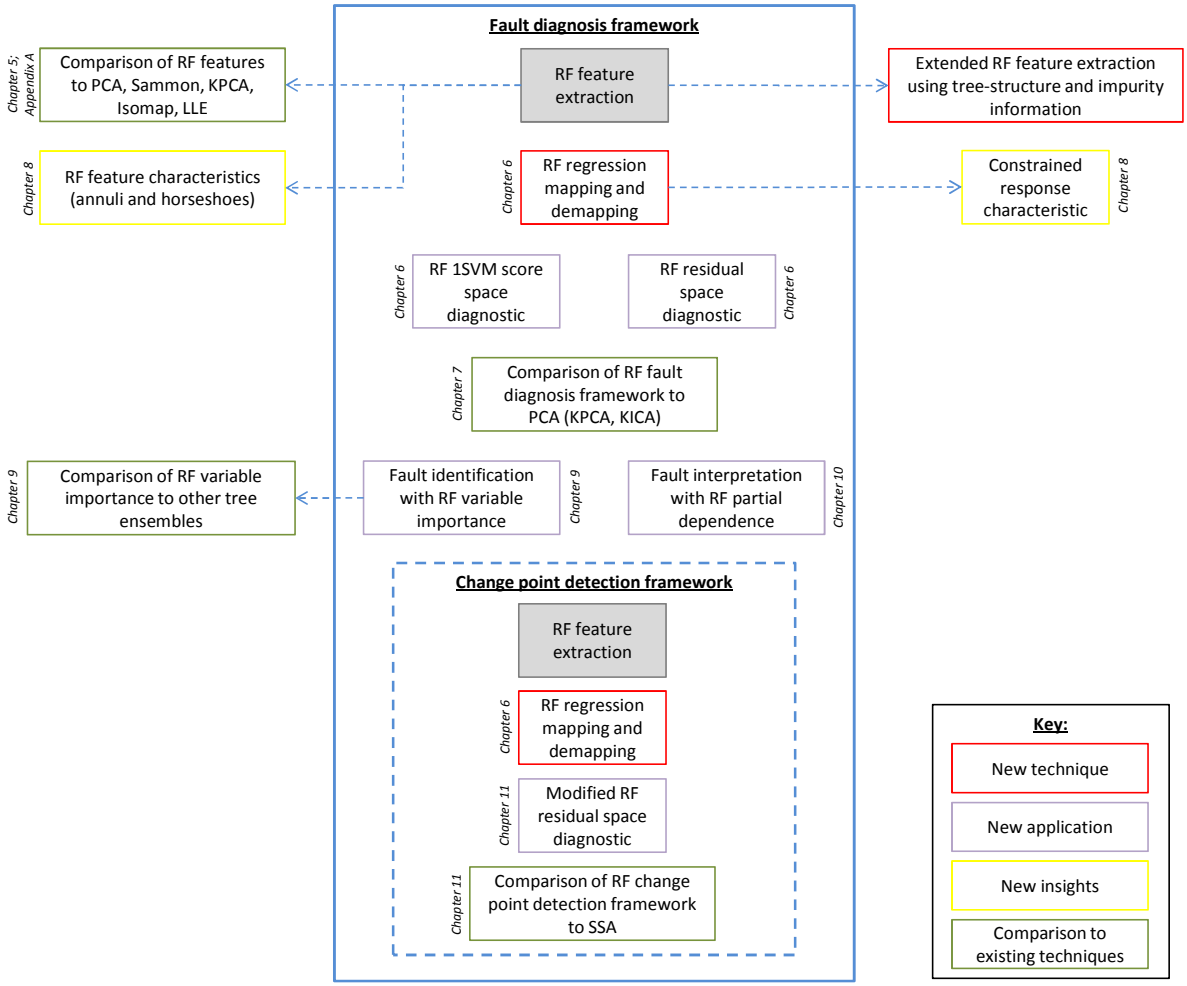

Figure 12.1: Overview of the structure and contributions of this work

12.2 Conclusions on hypothesis and objectives

Conclusions are made in terms of the hypothesis and objectives specified in the introduction to this work (Chapter 1). The hypothesis reads:

Random forests constitute a viable basis for the development of nonlinear process monitoring and fault diagnosis methods.

The hypothesis is supported from experimental results of the various case studies on fault detection and change point detection. Random forest feature extraction showed comparable fault diagnosis performance for the Tennessee Eastman process, better fault identification performance for the simple nonlinear system, and better fault detection performance for the calcium carbide process; as compared to principal component analysis (Chapter 7). Random forest change point detection performed significantly better than singular spectrum analysis change point detection in simple time series and more complex dynamic reaction system time series (Chapter 11). Process interpretation with random forest variable importance and partial dependence gave further insights on process variables closely associated with fault conditions (Chapters 9 and 10).

The objectives of this work are restated here:

- A critical literature survey on feature extraction techniques in fault diagnosis; as well as on the random forest modelling and feature extraction approach
- The assessment of the quality of features extracted with random forests
- The development of an unsupervised fault diagnostic scheme using random forest feature extraction
    - The implementation of a random forest fault diagnostic scheme as robust code
    - The testing of the random forest fault diagnostic schemes on simulated data, a benchmark process engineering problem and a real-world mineral processing data set
- The development of interpretive tools for identifying and interpreting important process variables involved in process changes and faults
    - The implementation of interpretive variable importance and visualization tools based on random forests as robust code
    - The testing of random forest interpretive tools on simulated data and process faults
- The development of a dynamic random forest change point detection scheme
    - The implementation of the dynamic random forest change point detection as robust code
    - The testing of the dynamic random forest change point detection scheme on simulated static and dynamic data sets

The first objective was a critical literature survey on feature extraction techniques in fault diagnosis and the characteristics of random forest feature extraction. Chapters 2 and 3 show the fruit of this survey, with concepts gained from literature incorporated in the development of the random forest algorithms. From a survey of process monitoring methods the benefits of feature extraction, as well as the limitations of nonlinear feature extraction, were discussed. The background on random forests suggested that this versatile nonlinear modelling scheme would be a viable candidate for not only nonlinear feature extraction for fault diagnosis, but also incorporating the various interpretive aspects of the random forest scheme to process data interpretation.

The second objective was the assessment of the quality of features extracted with random forests. Random forest feature extraction was compared to five other techniques on seven data sets, with performance measures incorporating local and global structure preservation. From this comparative study, random forest features were shown to be generally difficult to interpret in terms of geometry present in the original variable space. However, this lack of interpretation does not hinder the accurate reconstruction of the original variable

space from random forest features using random forest regression, as applied to seen training data. A further conclusion was that feature extraction performance measures were not very sensitive to the random forest parameters.

Random forest feature extraction did not outperform all other feature extraction methods on all data sets, and this should not be expected. The old adage of horses for courses should be applied in feature extraction applications: certain techniques are suited better to certain types of data structures. It can then be assumed that data structures exist for which random forest feature extraction will outperform other feature extraction techniques. The investigation of novel feature extraction techniques is then validated by the consideration that data may be structured in an infinite number of configurations.

The third objective considered the development of an unsupervised fault diagnostic scheme using random forest feature extraction. The result of this development is found in Chapter 6. Major design choices in the algorithm development resulted in the use of random forest regression models for mapping and demapping; as well as the application of one-class support vector machines to characterize score distances.

Another aspect of the third task was the testing of the random forest fault diagnostic scheme on simulated data, a benchmark process engineering problem and real-world mineral processing data. A simple nonlinear data set, the Tennessee Eastman process and a calcium carbide process were employed for this purpose, and relevant performance measures evaluated (Chapter 7). Overall, from the fault diagnosis method criteria, the random forest approach developed in Chapter 6 could be considered a suitable option for fault diagnosis. However, the random forest approach is computationally expensive, and this expense increases with the number of features used.

A significant result from the fault diagnosis application study (Chapter 7) was the observation that the residual diagnostic was more prevalent in detecting faults that the score space diagnostic, for the random forest fault diagnosis approach. Chapter 8 further investigated the characteristics of the random forest feature space to determine the reason for the success of the residual diagnostic / failure of the score space diagnostic. The constrained response characteristic, an artefact of using regression forests for explicit mapping and demapping, was found to be responsible for both high score space missing alarm rates and low residual space missing alarm rates. It was thus concluded that the random forest fault diagnosis framework, as it was developed in Chapter 6, has a strong residual space diagnostic, under the assumption that a large and representative sample of NOC data is available for training.

By investigating two-dimensional random forest projections of simulated Gaussian distributions, three common random forest feature structures were found: annulus-like shapes, round data clouds and horseshoes. The emergence of these shapes ties in with multidimensional scaling results found in literature, although further investigation is called for to explain the fundamental mechanisms peculiar to features extracted from random forest proximities. From these results, a simple heuristic rule was suggested for selecting the number of random forest features to extract, given a random forest proximity matrix. If the assumption is further made that NOC data would consist of one continuous group of data, selecting two random forest features may be a good starting place for the random forest fault diagnosis framework. This would replace the initial crossing criteria approach to selecting the number of features to employ for the random forest fault diagnosis framework. Using fewer features not only saves on computational expenses, but may also exclude noisy components.

The fourth objective was the development of interpretive tools for identifying and interpreting important process variables involved in process changes and faults. Chapter 9 investigated the application of random forest variable importance as a fault identification scheme. To this end, supervised random forests can be

trained once normal operating conditions and fault conditions have been identified by a fault detection technique. The level of association of each process variable with the fault condition (its variable importance) can be determined from this supervised random forest by calculating the degradation in performance when each process variable is permuted. The random forest variable importance approach was compared to other tree ensemble methods. It was concluded that random forests with random split selection can provide informative, computationally efficient indicators that can aid fault identification.

The fourth objective was further developed in terms of interpretation of process variables that have been identified as important. The interpretation of identified process variables was investigated Chapter 10, using the approach of random forest partial dependence. The partial dependence methodology was introduced, and together with random forest variable importance, incorporated into a data visualization tool for fault identification and interpretation. For simulated studies, the random forest partial dependence approach was able to capture and present nonlinear structure, even though only low predictive performance was achieved. Partial dependence plots for Tennessee Eastman and calcium carbide process faults showed variable thresholds between normal operating conditions and fault classes. Overall, the partial dependence methodology was shown to be a useful approach to visualizing and interpreting so-called black box models such as random forests, as well as for fault identification and interpretation.

The fifth objective involved the development of a dynamic random forest change point detection scheme. A random forest change point detection algorithm was developed in Chapter 11, analogous to the linear approach of singular spectrum analysis. Random forest change point detection proved to be more successful than singular change point detection in identifying known changes in the investigated case studies. The random forest change point detection scheme was shown to be fairly robust to noise and parameter changes. However, the random forest approach was shown to be much more computationally expensive than the singular spectrum analysis approach, and this computational expense increases with the number of features retained. As with random forest fault diagnosis, the success of random forest change point detection may relate to the constrained response characteristic expounded in Chapter 8.

12.3 General conclusion

With the proliferation of process data, the increasing complexity of process plants and the escalating demands of profitability and safety standards, automated process monitoring is an important tool for garnering valuable information and enabling efficient operation of process plants. Data-driven fault diagnosis schemes aim to exploit the availability of large databases of process data to detect and identify abnormal process conditions. Due to the limitations of linear feature extraction methods, the application of nonlinear feature extraction is a growing topic of interest. In this light, random forests have shown to be a viable contender for fault diagnosis.

The suitability of nonlinear feature extraction methods to detect abnormal data is considered here. Given a distribution of normal operating conditions data, a nonlinear feature extraction method will attempt to find some nonlinear manifold or transformation that captures the structure of the data. If the distribution of new data representing fault conditions conforms to the same manifold or transformation, but is located in a sparse region of the normal operating conditions data, these data points will be flagged as faults in the nonlinear space. However, if the distribution of the new data representing the fault conditions does not conform to the same nonlinear manifold or transformation as the normal operating conditions data, the projection of this data unto the normal operating conditions manifold will give rise to reconstruction errors in the original variable space, and the faulty data are flagged as faults.

This validates the use of nonlinear feature extraction techniques for fault diagnosis in general, and the use of random forest feature extraction specifically. The performance of the fault diagnosis method depends on the suitable selection of model parameters. In the case of random forest feature extraction, the selection of the

number of features has been discussed, and a heuristic proposed. The additional interpretive tools of random forests, in terms of variable importance and partial dependence, are further benefits to the application of these models to process monitoring.

12.4 Recommendations

From the above conclusions, the application of the proposed random forest feature extraction frameworks for fault diagnosis and change point detection is recommended as a tool for process monitoring. The efficacy of these frameworks, as for all data-based process monitoring schemes, depends on a number of factors.

Firstly, for a fault to be detected from process data, the process data must show some distributional change from normal operating conditions to the fault conditions. If the data show no changes, no data-based fault diagnosis scheme will detect a change. The availability of representative process data is then a necessary, but not sufficient, condition for fault detection with data-based fault diagnosis. Data preprocessing then remains a vital component of process monitoring.

Secondly, the estimation of expected false alarm rates for a fault diagnosis scheme is vital. Given the labelling of new data as representing fault conditions, an expected false alarm rate will aid in risk assessment when decisions need to be made in terms of process recovery. This work has not explicitly investigated the calculation of expected false alarm rates. A simplistic approach may be to withhold validation normal operating conditions data (as done in certain case studies in this work) to assess expected false alarm rates. In general, this issue is not resolved for nonlinear feature extraction approaches to fault diagnosis.

An important parameter for the random forest fault diagnosis framework is the number of features to extract to represent normal operating conditions. From this work, it appears that fewer features are necessary than suggested by the crossing criteria initially employed. Exploiting knowledge gained from the nature of random forest proximities, as well as the characteristics of multidimensional scaling, show that cluster counting from the proximity matrix information may provide an initial estimate of optimal feature space dimensionality. This heuristic has been conceptually discussed and empirically demonstrated, but needs additional validation. Further work should be done to illuminate the fundamental mechanisms responsible for annuli, concentric circles and horseshoes in random forest feature spaces.

A disadvantage of the random forest frameworks, compared to linear approaches, has been its high computational expense. The computational expenses of random forest methods can be greatly relieved by parallelizing the training and predictions of tree members. As the tree members of a random forest are independent, they are ideally suited to parallelization.

REFERENCES

Abba, M.C., Sun, H., Hawkins, K.A., Drake, J.A., Hu, Y., Nunez, M.I., Gaddis, S., Shi, T., Horvath, S., Sahin, A. & Aldaz, C.M., 2007. Breast cancer molecular signatures as determined by SAGE: Correlation with lymph node status. *Molecular Cancer Research*, 5(9), 881-890.

Aldrich, C. & Reuter, M.A., 1999. Monitoring of metallurgical reactors by the use of topographic mapping of process data. *Minerals Engineering*, 12(11), 1301-1312.

Amaratunga, D., Cabrera, J. & Lee, Y., 2008a. Enriched random forests. *Bioinformatics*, 24(18), 2010-2014.

Amaratunga, D., Cabrera, J. & Kovtun, V., 2008b. Microarray learning with ABC. *Biostatistics*, 9(1), 128-136.

Antory, D., Irwin, G.W., Kruger, U. & McCullough, G., 2008. Improved process monitoring using nonlinear principal component models. *International Journal of Intelligent Systems*, 23(5), 520-544.

Archer, K.J. & Kimes, R.V., 2008. Empirical characterization of random forest variable importance measures. *Computational Statistics & Data Analysis*, 52(4), 2249-2260.

Auret, L. & Aldrich, C., 2010a. Change point detection in time series data with random forests. *Control Engineering Practice*, 18(8), 990 - 1002.

Auret, L. & Aldrich, C., 2010b. Unsupervised process fault detection with random forests. *Industrial & Engineering Chemistry Research*, doi: 10.1021/ie901975c.

Bakshi, B.R. & Stephanopoulos, G., 1994. Representation of process trends - IV. Induction of real-time patterns from operating data for diagnosis and supervisory control. *Computers & Chemical Engineering*, 18(4), 303-332.

Berk, R., Sherman, L., Barnes, G., Kurtz, E. & Ahlman, L., 2009. Forecasting murder within a population of probationers and parolees: a high stakes application of statistical learning. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 172(1), 191-211.

Berrado, A. & Rassili, A., 2010. Modeling and characterizing of the thixoforming of steel process parameters – the case of forming load. *International Journal of Material Forming*, 3(S1), 735-738.

Bonissone, P. & Iyer, N., 2007. Soft computing applications to Prognostics and Health Management (PHM): Leveraging field data and domain knowledge. In *Computational and Ambient Intelligence*. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 928-939.

Borg, I. & Groenen, P.J.F., 2005. *Modern Multidimensional Scaling: Theory and applications*, USA: Springer.

Breiman, L., 1996. Bagging predictors. *Machine Learning*, 24(2), 123-140.

Breiman, L., 2001. Random forests. *Machine Learning*, 45(1), 5-32.

Breiman, L. & Cutler, A., 2003. Manual on setting up, using, and understanding random forests v4.0. *ftp://ftp.stat.berkeley.edu/pub/users/breiman/Using_random_forests_v4.0.pdf*. Available at: ftp://ftp.stat.berkeley.edu/pub/users/breiman/Using_random_forests_v4.0.pdf [Accessed May 30, 2008].

Breiman, L., Friedman, J.H., Olshen, R.A. & Stone, C.J., 1993. *Classification and regression trees*, Chapman & Hall.

Bureau, A., Dupuis, J., Hayward, B., Falls, K. & Van Eerdewegh, P., 2003. Mapping complex traits using Random Forests. *BMC Genetics*, 4(Suppl 1), S64.

Cai, J., Ma, X. & Li, Q., 2009. On-line monitoring the performance of coal-fired power unit: A method based on support vector machine. *Applied Thermal Engineering*, 29(11-12), 2308-2319.

Carlisle, D.M., Falcone, J. & Meador, M.R., 2008. Predicting the biological condition of streams: use of geospatial indicators of natural and anthropogenic characteristics of watersheds. *Environmental Monitoring and Assessment*, 151(1-4), 143-160.

Carreira-Perpinan, M.A., 1997. *A review of dimension reduction techniques*, Department of Computer Science, University of Sheffield.

Cayton, L., 2005. *Algorithms for manifold learning*, San Diego: University of California. Available at: http://people.kyb.tuebingen.mpg.de/lcayton/resexam.pdf [Accessed October 21, 2009].

Chang, C. & Lin, C., 2001. LIBSVM: A library for support vector machines. Available at: http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Chen, H., Jiang, G. & Yoshihira, K., 2008. Monitoring high-dimensional data for failure detection and localization in large-scale computing systems. *IEEE Transactions on Knowledge and Data Engineering*, 20(1), 13-25.

Chigirev, D. & Bialek, W., 2004. Optimal manifold representation of data: an information theoretic approach. *Advances in Neural Information Processing Systems*, 16, 161-168.

Cho, J., Lee, J., Wook Choi, S., Lee, D. & Lee, I., 2005. Fault identification for process monitoring using kernel principal component analysis. *Chemical Engineering Science*, 60(1), 279-288.

Choi, S.W., Lee, C., Lee, J., Park, J.H. & Lee, I., 2005. Fault detection and identification of nonlinear processes based on kernel PCA. *Chemometrics and Intelligent Laboratory Systems*, 75(1), 55-67.

Choi, S.W. & Lee, I., 2004. Nonlinear dynamic process monitoring based on dynamic kernel PCA. *Chemical Engineering Science*, 59(24), 5897-5908.

Cox, T.F. & Cox, M.A.A., 2001. *Multidimensional scaling*, Chapman & Hall.

Cummings, M. & Segal, M., 2004. Few amino acid positions in rpoB are associated with most of the rifampin resistance in Mycobacterium tuberculosis. *BMC Bioinformatics*, 5(1), 137-143.

Curnow, S.J., Falciani, F., Durrani, O.M., Cheung, C.M.G., Ross, E.J., Wloka, K., Rauz, S., Wallace, G.R., Salmon, M. & Murray, P.I., 2005. Multiplex bead immunoassay analysis of aqueous humor reveals distinct cytokine profiles in uveitis. *Investigative Ophthalmology and Visual Science*, 46(11), 4251-4259.

Cutler, A., 2009. Random forests. In *useR! The R User Conference 2009*. Available at: http://www.agrocampus-ouest.fr/math/useR-2009/.

Cutler, A. & Stevens, J.R., 2006. Random forests for microarrays. In *Methods in Enzymology; DNA Microarrays, Part B: Databases and Statistics*. Academic Press, pp. 422-432.

Cutler, D.R., Edwards, T.C., Beard, K.H., Cutler, A. & Hess, K.T., 2007. Random forests for classification in ecology. *Ecology*, 88(11), 2783-2792.

De Ridder, D. & Duin, R.P.W., 1997. Sammon's mapping using neural networks: A comparison. *Pattern Recognition Letters*, 18(11-13), 1307-1316.

Deloncle, A., Berk, R., D'Andrea, F. & Ghil, M., 2007. Weather regime prediction using statistical learning. *Journal of the Atmospheric Sciences*, 64(5), 1619-1635.

Diaconis, P., Goel, S. & Holmes, S., 2008. Horseshoes in multidimensional scaling and local kernel methods. *Annals of Applied Statistics*, 2(3), 777-807.

Diaz-Uriarte, R. & Alvarez de Andres, S., 2006. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7(1), 3-15.

Dietterich, T.G., 2000. Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*. Lecture Notes in Computer Science. MCS '00. Springer-Verlag, pp. 1-15.

Dong, D. & McAvoy, T.J., 1996. Nonlinear principal component analysis - Based on principal curves and neural networks. *Computers & Chemical Engineering*, 20(1), 65-78.

Downs, J.J. & Vogel, E.F., 1993. A plant-wide industrial process control problem. *Computers & Chemical Engineering*, 17(3), 245-255.

Duin, R.P.W., Juszczak, P., Paclik, E., Pekalska, D. & de Ridder, D.M.J., 2007. *PRTools 4.1, A Matlab toolbox for pattern recognition*, Delft University of Technology.

Dunia, R. & Qin, S.J., 1998. Joint diagnosis of process and sensor faults using principal component analysis. *Control Engineering Practice*, 6(4), 457-469.

Eller, C.D., Regelson, M., Merriman, B., Nelson, S., Horvath, S. & Marahrens, Y., 2007. Repetitive sequence environment distinguishes housekeeping genes. *Gene*, 390(1-2), 153-165.

Elsner, J.B. & Tsonis, A.A., 1996. *Singular spectrum analysis: A new tool in time series analysis*, Plenum Press.

Finehout, E.J., Franck, Z., Choe, L.H., Relkin, N. & Lee, K.H., 2007. Cerebrospinal fluid proteomic biomarkers for Alzheimer's disease. *Annals of Neurology*, 61(2), 120-129.

Fodor, I.K., 2002. *A survey of dimension reduction techniques*, Lawrence Livermore National Laboratory, Center for Applied Scientific Computing.

Frank, A. & Asuncion, A., 2010. UCI Machine Learning Repository. *University of California, Irvine, School of Information and Computer Sciences*. Available at: http://archive.ics.uci.edu/ml.

Friedman, J.H., 2001. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5), 1189-1232.

Furlanello, C., Neteler, M., Merler, S., Menegon, S., Fontanari, S., Donini, A., Rizzoli, A. & Chemini, C., 2003. GIS and the random forest predictor: Integration in R for tick-borne disease risk assessment. In K. Hornik, F. Leisch, & A. Zeileis, eds. *Proceedings of the 3rd International Workshop on Distributed Statistical Computing*. Distributed Statistical Computing 2003. Technische Universität Wien, Vienna, Austria.

Garzón, M.B., Blazek, R., Neteler, M., Dios, R.S.D., Ollero, H.S. & Furlanello, C., 2006. Predicting habitat suitability with machine learning models: The potential area of Pinus sylvestris L. in the Iberian Peninsula. *Ecological Modelling*, 197(3-4), 383-393.

Geng, X., Zhan, D. & Zhou, Z., 2005. Supervised nonlinear dimensionality reduction for visualization and classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 35(6), 1098-1107.

Genuer, R., Poggi, J. & Tuleau-Malot, C., 2010. Variable selection using random forests. *Pattern Recognition Letters*, Article in press, doi: 10.1016/j.patrec.2010.03.014.

Girardello, M., Griggio, M., Whittingham, M.J. & Rushton, S.P., 2010. Models of climate associations and distributions of amphibians in Italy. *Ecological Research*, 25(1), 103-111.

Granitto, P.M., Gasperi, F., Biasioli, F., Trainotti, E. & Furlanello, C., 2007. Modern data mining tools in descriptive sensory analysis: A case study with a random forest approach. *Food Quality and*

Preference, 18(4), 681-689.

Grömping, U., 2009. Variable importance assessment in regression: Linear regression versus random forest. *The American Statistician*, 63(4), 308-319.

Gupta, S., Matthew, S., Abreu, P.M. & Aires-de-Sousa, J., 2006. QSAR analysis of phenolic antioxidants using MOLMAP descriptors of local properties. *Bioorganic & medicinal chemistry*, 14(4), 1199-1206.

Gyorgyi, L. & Field, R.J., 1991. Simple models of deterministic chaos in the Belousov-Zhabotinskii reaction. *Journal of Physical Chemistry*, 95(17), 6594-6602.

Hastie, T., Tibshirani, R. & Friedman, J., 2009. *The elements of statistical learning - Data mining, inference and prediction* Second edition., Springer.

Himmelblau, D.M., 1978. *Fault detection and diagnosis in chemical and petrochemical processes*, Elsevier.

Ho, T.K., 1995. Random decision forests. In *Proceedings of the Third International Conference on Document Analysis and Recognition*. ICDAR1995. Montreal, Canada: IEEE Computer Society, pp. 278-282.

Hothorn, T., Hornik, K. & Zeileis, A., 2006. Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, 15(3), 651-674.

Hsu, C.W., Chang, C.C. & Lin, C.J., 2003. A practical guide to support vector machine classification. Available at: http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf. [Accessed March 30, 2009].

Jemwa, G.T. & Aldrich, C., 2006. Kernel-based fault diagnosis on mineral processing plants. *Minerals Engineering*, 19(11), 1149-1162.

Jemwa, G.T. & Aldrich, C., 2005. Improving process operations using support vector machines and decision trees. *AICHE Journal*, 51(2), 526-543.

Jia, F., Martin, E.B. & Morris, A.J., 1998. Non-linear principal components analysis for process fault detection. *Computers & Chemical Engineering*, 22, S851-S854.

Jorjani, E., Chehrehchelgani, S. & Mesroghli, S., 2007. Prediction of microbial desulfurization of coal using artificial neural networks. *Minerals Engineering*, 20(14), 1285-1292.

Kano, M., Tanaka, S., Hasebe, S. & Hashimoto, I., 2003. Monitoring independent components for fault detection. *AICHE Journal*, 49(4), 969-976.

Kohavi, R. & John, G.H., 1997. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2), 273-324.

Kramer, M.A., 1991. Nonlinear principal component analysis using autoassociative neural networks. *AICHE Journal*, 37(2), 233-243.

Kresta, J.V., MacGregor, J.F. & Marlin, T.E., 1991. Multivariate statistical monitoring of process operating performance. *Canadian Journal of Chemical Engineering*, 69(1), 35-47.

Ku, W., Storer, R.H. & Georgakis, C., 1995. Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 30(1), 179-196.

Lee, J.S. & Chang, K.S., 1996. Applications of chaos and fractals in process systems engineering. *Journal of Process Control*, 6(2), 71-87.

Lee, J.A., Lendasse, A. & Verleysen, M., 2004. Nonlinear projection with curvilinear distances: Isomap versus curvilinear distance analysis. *Neurocomputing*, 57, 49-76.

Lee, J., Qin, S.J. & Lee, I., 2007. Fault detection of non-linear processes using kernel independent component

analysis. *The Canadian Journal of Chemical Engineering*, 85(4), 526-536.

Lee, J., Yoo, C., Choi, S.W., Vanrolleghem, P.A. & Lee, I., 2004a. Nonlinear process monitoring using kernel principal component analysis. *Chemical Engineering Science*, 59(1), 223-234.

Lee, J., Yoo, C. & Lee, I., 2004b. Statistical process monitoring with independent component analysis. *Journal of Process Control*, 14(5), 467-485.

Lennert-Cody, C.E., Roberts, J.J. & Stephenson, R.J., 2008. Effects of gear characteristics on the presence of bigeye tuna (Thunnus obesus) in the catches of the purse-seine fishery of the eastern Pacific Ocean. *ICES Journal of Marine Science*, 65(6), 970-978.

Levina, E. & Bickel, P.J., 2004. Maximum likelihood estimation of intrinsic dimension. *Advances in Neural Information Processing Systems*, 17.

Li, W., Yue, H.H., Valle-Cervantes, S. & Qin, S.J., 2000. Recursive PCA for adaptive process monitoring. *Journal of Process Control*, 10(5), 471-486.

Liaw, A. & Wiener, M., 2002. Classification and regression by randomForest. *R News*, 2(3), 18-22.

Liu, X., Kruger, U., Littler, T., Xie, L. & Wang, S., 2009. Moving window kernel PCA for adaptive monitoring of nonlinear processes. *Chemometrics and Intelligent Laboratory Systems*, 96(2), 132-143.

Ma, C.Y. & Wang, X.Z., 2009. Inductive data mining based on genetic programming: Automatic generation of decision trees from data for process historical data analysis. *Computers & Chemical Engineering*, 33(10), 1602-1616.

MacGregor, J.F. & Kourti, T., 1995. Statistical process control of multivariate processes. *Control Engineering Practice*, 3(3), 403-414.

Maragoudakis, M., Loukis, E. & Pantelides, P., 2008. Random forests identification of gas turbine faults. In *Proceedings of the 2008 19th International Conference on Systems Engineering*. 19th International Conference on Systems Engineering (ICSEng 2008). Washington, DC, USA: IEEE Computer Society Press, pp. 127-132. Available at: http://doi.ieeecomputersociety.org/10.1109/ICSEng.2008.81.

Martin, E.B. & Morris, A.J., 1996. Non-parametric confidence bounds for process performance monitoring charts. *Journal of Process Control*, 6(6), 349-358.

Massinaei, M. & Doostmohammadi, R., 2010. Modeling of bubble surface area flux in an industrial rougher column using artificial neural network and statistical techniques. *Minerals Engineering*, 23(2), 83-90.

Moskvina, V. & Zhigljavsky, A., 2003. An algorithm based on singular spectrum analysis for change point detection. *Communications in Statistics: Simulation and Computation*, 32(2), 319-352.

Nicodemus, K.K., Malley, J.D., Strobl, C. & Ziegler, A., 2010. The behaviour of random forest permutation-based variable importance measures under predictor correlation. *BMC Bioinformatics*, 11(1), 110-122.

Nicodemus, K.K. & Malley, J.D., 2009. Predictor correlation impacts machine learning algorithms: implications for genomic studies. *Bioinformatics*, 25(15), 1884-1890.

Olden, J.D. & Jackson, D.A., 2002. Illuminating the "black box": a randomization approach for understanding variable contributions in artificial neural networks. *Ecological Modelling*, 154(1-2), 135-150.

Oppel, S., Powell, A.N. & Dickson, D.L., 2009. Using an algorithmic model to reveal individually variable movement decisions in a wintering sea duck. *Journal of Animal Ecology*, 78(3), 524-531.

O'Riordan, E., Orlova, T.N., Mei J, J., Butt, K., Chander, P.M., Rahman, S., Mya, M., Hu, R., Momin, J., Eng, E.W., Hampel, D.J., Hartman, B., Kretzler, M., Delaney, V. & Goligorsky, M.S., 2004. Bioinformatic analysis of

the urine proteome of acute allograft rejection. *J Am Soc Nephrol*, 15(12), 3240-3248.

Pang, H., Lin, A., Holford, M., Enerson, B.E., Lu, B., Lawton, M.P., Floyd, E. & Zhao, H., 2006. Pathway analysis using random forests classification and regression. *Bioinformatics*, 22(16), 2028-2036.

Peters, J., Baets, B.D., Verhoest, N.E.C., Samson, R., Degroeve, S., Becker, P.D. & Huybrechts, W., 2007. Random forests as a tool for ecohydrological distribution modelling. *Ecological Modelling*, 207(2-4), 304-318.

Prasad, A.M., Iverson, L.R. & Liaw, A., 2006. Newer classification and regression tree techniques: Bagging and random forests for ecological prediction. *Ecosystems*, 9(2), 181-199.

Qi, Y., Klein-Seetharaman, J. & Bar-Joseph, Z., 2005. Random forest similarity for protein-protein interaction prediction from multiple sources. *Pacific Symposium on Biocomputing*, 10, 531-542.

R Development Core Team, 2010. *R: A Language and Environment for Statistical Computing*, Vienna, Austria: R Foundation for Statistical Computing. Available at: http://www.R-project.org.

Raza, K., Scheel-Toellner, D., Lee, C., Pilling, D., Curnow, S.J., Falciani, F., Trevino, V., Kumar, K., Assi, L., Lord, J., Gordon, C., Buckley, C. & Salmon, M., 2006. Synovial fluid leukocyte apoptosis is inhibited in patients with very early rheumatoid arthritis. *Arthritis Research & Therapy*, 8(4), R120.

Ridgeway, G., 2007a. gbm: Generalized boosted regression models. *http://www.i-pensieri.com/gregr/gbm.shtml*. Available at: http://www.i-pensieri.com/gregr/gbm.shtml.

Ridgeway, W., 2007b. Generalized boosted models: A guide to the gbm package. *http://i-pensieri.com/gregr/papers/gbm-vignette.pdf*. Available at: http://i-pensieri.com/gregr/papers/gbm-vignette.pdf.

Roweis, S.T. & Saul, L.K., 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500), 2323-2326.

Russell, E.L., Chiang, L.H. & Braatz, R.D., 2000. *Data-driven techniques for fault detection and diagnosis in chemical processes*, Springer.

Sakiyama, Y., Yuki, H., Moriya, T., Hattori, K., Suzuki, M., Shimada, K. & Honma, T., 2008. Predicting human liver microsomal stability with machine learning techniques. *Journal of Molecular Graphics and Modelling*, 26(6), 907-915.

Sammon, J.W., 1969. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18(5), 401-409.

Seligson, D.B., Horvath, S., McBrian, M.A., Mah, V., Yu, H., Tze, S., Wang, Q., Chia, D., Goodglick, L. & Kurdistani, S.K., 2009. Global levels of histone modifications predict prognosis in different cancers. *American Journal of Pathology*, 174(5), 1619-1628.

Shao, J. & Rong, G., 2009. Nonlinear process monitoring based on maximum variance unfolding projections. *Expert Systems with Applications*, 36(8), 11332-11340.

Shao, J., Rong, G. & Lee, J.M., 2009. Generalized orthogonal locality preserving projections for nonlinear fault detection and diagnosis. *Chemometrics and Intelligent Laboratory Systems*, 96(1), 75-83.

Shi, T. & Horvath, S., 2006. Unsupervised learning with random forest predictors. *Journal of Computational and Graphical Statistics*, 15(1), 118-138.

Shi, T., Seligson, D., Belldegrun, A.S., Palotie, A. & Horvath, S., 2005. Tumor classification by tissue microarray profiling: Random forest clustering applied to renal cell carcinoma. *Modern Pathology*, 18(4), 547-557.

Sledge, I.J., Havens, T.C., Huband, J.M., Bezdek, J.C. & Keller, J.M., 2009. Finding the number of clusters in

ordered dissimilarities. *Soft Computing*, 13(12), 1125-1142.

Strobl, C., Boulesteix, A.L., Zeileis, A. & Hothorn, T., 2007. Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*, 8, 25-45.

Strobl, C. & Zeileis, A., 2008. *Danger: High power! Exploring the statistical properties of a test for random forest variable importance*, Department of Statistics: University of Munich.

Strobl, C., Boulesteix, A., Kneib, T., Augustin, T. & Zeileis, A., 2008. Conditional variable importance for random forests. *BMC Bioinformatics*, 9(1), 307-317.

Strobl, C., Malley, J. & Tutz, G., 2009. An introduction to recursive partitioning: Rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychological methods*, 14(4), 323-348.

Svetnik, V., Liaw, A., Tong, C., Culberson, J.C., Sheridan, R.P. & Feuston, B.P., 2003. Random forest: A classification and regression tool for compound classification and QSAR modeling. *Journal of Chemical Information and Computer Sciences*, 43(6), 1947-1958.

Svetnik, V., Wang, T., Tong, C., Liaw, A., Sheridan, R.P. & Song, Q., 2005. Boosting: An ensemble learning tool for compound classification and QSAR modeling. *Journal of Chemical Information and Modeling*, 45(3), 786-799.

Tenenbaum, J.B., Silva, V.D. & Langford, J.C., 2000. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500), 2319-2323.

Valentini, G. & Masulli, F., 2002. Ensembles of learning machines. In *Neural Nets*. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 3-20. Available at: http://dx.doi.org/10.1007/3-540-45808-5_1.

Van der Maaten, L. & Hinton, G., 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9, 2579-2625.

Van der Maaten, L.J.P., Postma, E.O. & Van den Herik, H.J., 2009. *Dimensionality reduction: A comparative review*, Tilburg University. Available at: http://homepage.tudelft.nl/19j49/Matlab_Toolbox_for_Dimensionality_Reduction_files/TR_Dimensie reductie.pdf.

Venkatasubramanian, V., Rengaswamy, R. & Kavuri, S.N., 2003a. A review of process fault detection and diagnosis Part II: Quantitative model and search strategies. *Computers & Chemical Engineering*, 27(3), 313-326.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N. & Yin, K., 2003b. A review of process fault detection and diagnosis Part III: Process history based methods. *Computers & Chemical Engineering*, 27(3), 327-346.

Venkatasubramanian, V., Rengaswamy, R., Yin, K. & Kavuri, S.N., 2003c. A review of process fault detection and diagnosis Part I: Quantitative model-based methods. *Computers & Chemical Engineering*, 27(3), 293-311.

Wang, X., Chen, J., Liu, C. & Pan, F., 2010. Hybrid modeling of penicillin fermentation process based on least square support vector machine. *Chemical Engineering Research and Design*, 88(4), 415-420.

Wang, X., Kruger, U. & Irwin, G.W., 2005. Process Monitoring Approach Using Fast Moving Window PCA. *Industrial & Engineering Chemistry Research*, 44(15), 5691-5702.

Wilson, D.J.H., Irwin, G.W. & Lightbody, G., 1999. RBF principal manifolds for process monitoring. *Neural Networks, IEEE Transactions on*, 10; 10(6), 1424-1434.

Wise, B.M. & Gallagher, N.B., 1996. The process chemometrics approach to process monitoring and fault detection. *Journal of Process Control*, 6(6), 329-348.

Yang, B., Di, X. & Han, T., 2008. Random forests classifier for machine fault diagnosis. *Journal of Mechanical Science and Technology*, 22(9), 1716-1725.

Yeh, M., Lee, I., Wu, G., Wu, Y. & Chang, E.Y., 2005. Manifold learning: a promised land or work in progress? In *Proceedings of IEEE Conference on Multimedia and Expo 2005*. ICME 2005. Amsterdam, The Netherlands, pp. 1154 - 1175.

Zhang, J., Martin, E.B. & Morris, A.J., 1997. Process monitoring using non-linear statistical techniques. *Chemical Engineering Journal*, 67(3), 181-189.

Zhang, M.H., Xu, Q.S., Daeyaert, F., Lewi, P.J. & Massart, D.L., 2005. Application of boosting to classification problems in chemometrics. *Analytica Chimica Acta*, 544(1-2), 167-176.

Zhang, Y. & Qin, S.J., 2008. Improved nonlinear fault detection technique and statistical analysis. *AICHE Journal*, 54(12), 3207-3220.

Zhu, Q. & Li, C., 2006. Dimensionality reduction with input training neural network and its application in chemical process modelling. *Chinese Journal of Chemical Engineering*, 14(5), 597-603.

APPENDIX A - ADDITIONAL FEATURE EXTRACTION RESULTS

Appendix A presents additional results from the random forest feature quality validation study (Chapter 5). The results for local and global structure preservation performance measures for six data and seven feature extraction techniques are provided, as well as the sensitivity of these performance measures to the random forest parameters K (number of trees) and M (number of random split variables). The influence of the random forest parameters on the strength / correlation ratio of the forest is also shown.

A.1 Performance measures for feature quality evaluation

The performance measures for ten replicates are shown in the following figures, for all seven data sets, seven feature extraction methods and four performance measure types.

A.1.1 Local structure preservation

From the local structure preservation measures graphs Figure A.1, it is apparent that random forest feature extraction did not perform best for any data sets. However, it appears to only perform outright worst for datacop.

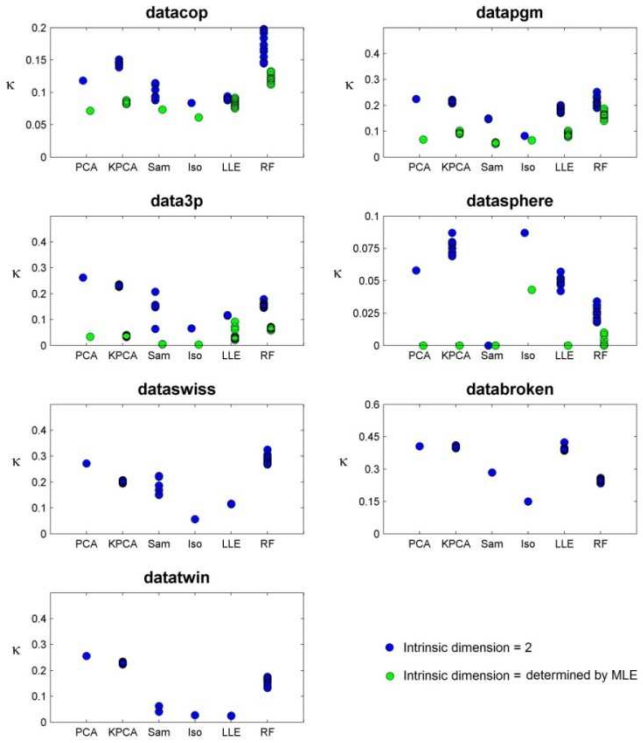

Figure A.1: Local structure preservation performance as expressed by 1-nearest neighbour errors κ (Sam = Sammon mapping, Iso = Isomap)

A.1.2 Linear reconstruction correlation

From Figure A.2, it appears that random forest feature extraction does not perform best on linear reconstruction correlation for the data sets investigated.

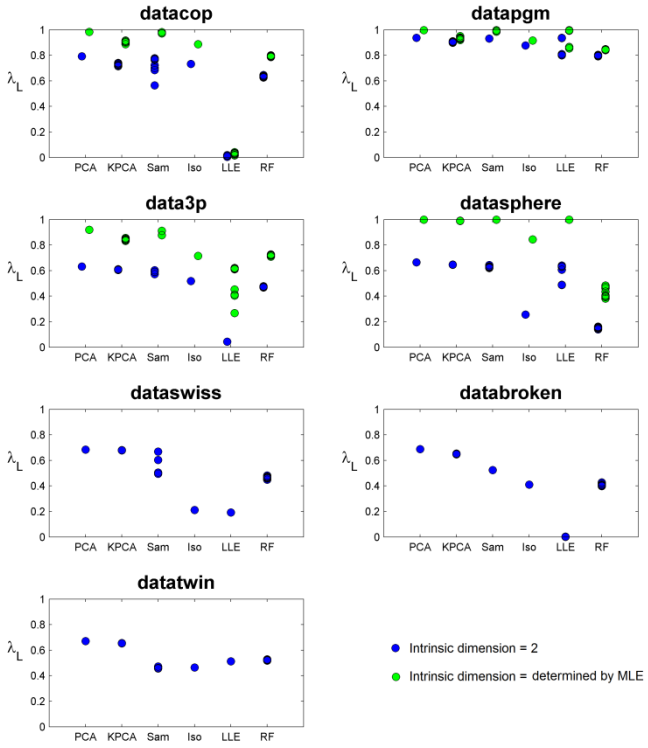

Figure A.2: Global structure preservation performance as expressed by linear reconstruction correlation λ_L (Sam = Sammon mapping, Iso = Isomap)

A.1.3 Nonlinear reconstruction correlation

In terms of nonlinear reconstruction of variables from features, random forest feature extraction appears more successful as compared to linear reconstruction (Figure A.3). Random forest feature extraction performs better on at least one data set compared to all techniques except Isomap, and for the majority of data sets as compared to PCA and LLE. This better performance with nonlinear reconstruction may indicate that linear regression is not able to reconstruct nonlinear features as obtained by random forests.

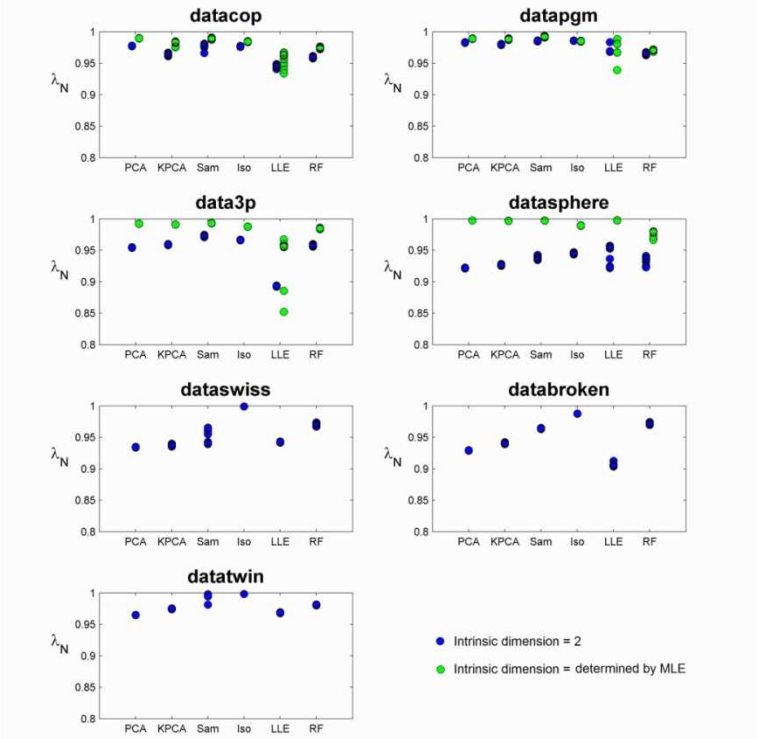

Figure A.3: Global structure preservation performance as expressed by nonlinear reconstruction correlation λ_N (Sam = Sammon mapping, Iso = Isomap)

A.1.4 Pairwise distance correlation

From Figure A.4, it is apparent that random forest feature extraction shows some of the lowest pairwise distance correlations for all data sets. Only LLE appears to do worse.

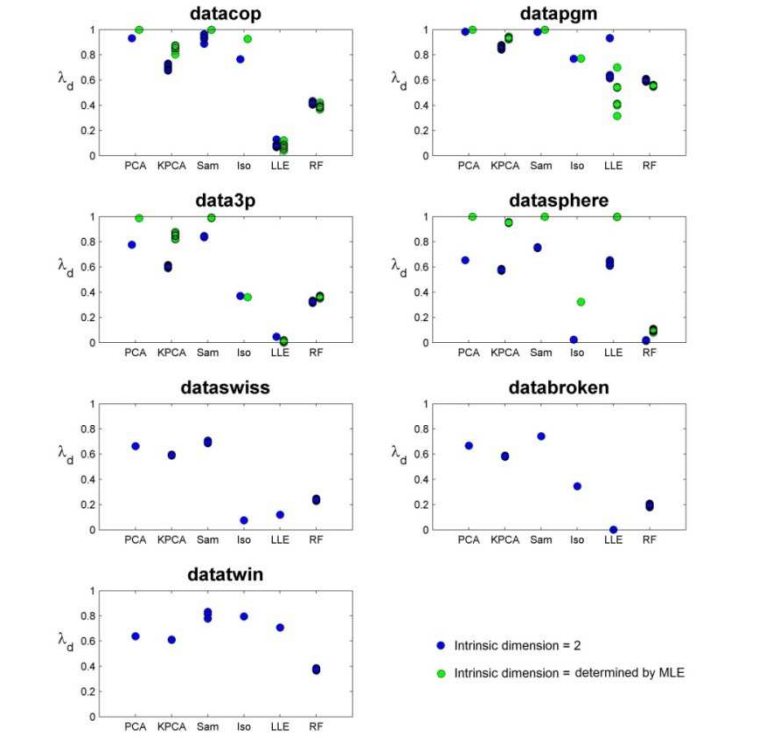

Figure A.4: Global structure preservation performance as expressed by pairwise distance correlation λ_d (Sam = Sammon mapping, Iso = Isomap)

A.2 Feature extraction performance sensitivity

As an extension of the previous study, an investigation was done on the sensitivity of the local structure preservation (κ), global structure preservation (λ_L, λ_N and λ_d) and ensemble performance measures to changes in the random forest parameters, K and M (number of trees and random split variables).

The influence of changes in K on performance measures was investigated for forest sizes from 100 to 1000 trees, on all seven data sets and an intrinsic dimensionality of two. The influence of changes in M was restricted to two data sets (data3p and datacop) as all other data sets are of low dimensionality. The dimensionality of the data sets (ten and twelve, respectively) determine the range in which M was changed. Again, two features were extracted.

A.2.1 Local structure preservation sensitivity

From Figure A.5, three data sets showed a general downward trend in κ as the number of trees in the forest increases, while no clear trend is apparent from the other data sets. No clear trend is evident when changes are made to the number of random split variables (Figure A.6).

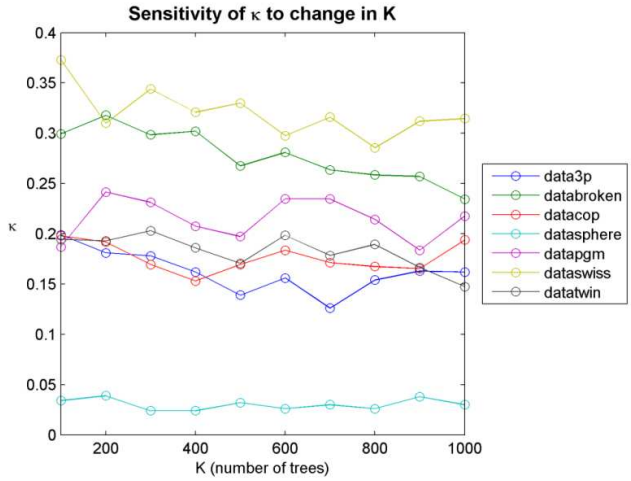

Figure A.5: Sensitivity of local structure preservation measure to change in K

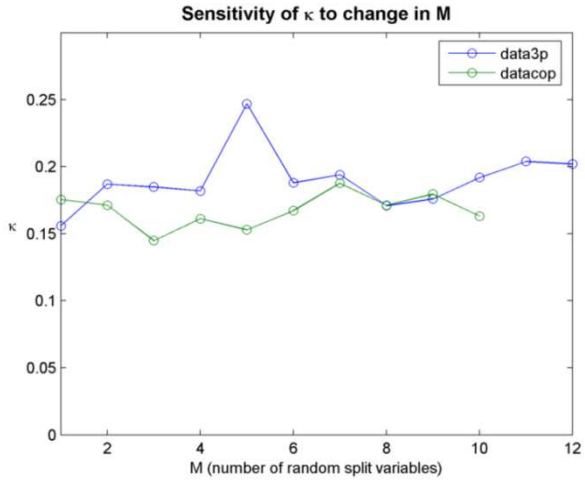

Figure A.6: Sensitivity of local structure preservation measure to change in M

A.2.2 Global structure preservation sensitivity

- Linear reconstruction correlation

Neither the number of trees nor the number of random split variables appears to have a significant influence on the linear reconstruction correlations, from Figure A.7 and Figure A.8.

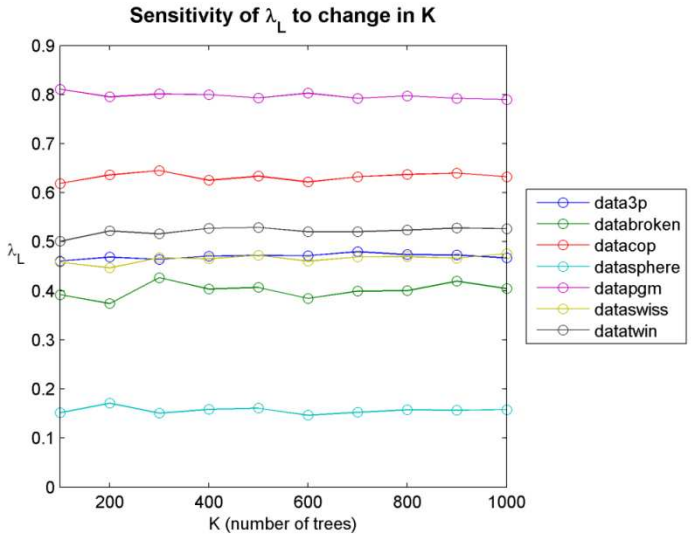

Figure A.7: Sensitivity of linear reconstruction correlation to change in K

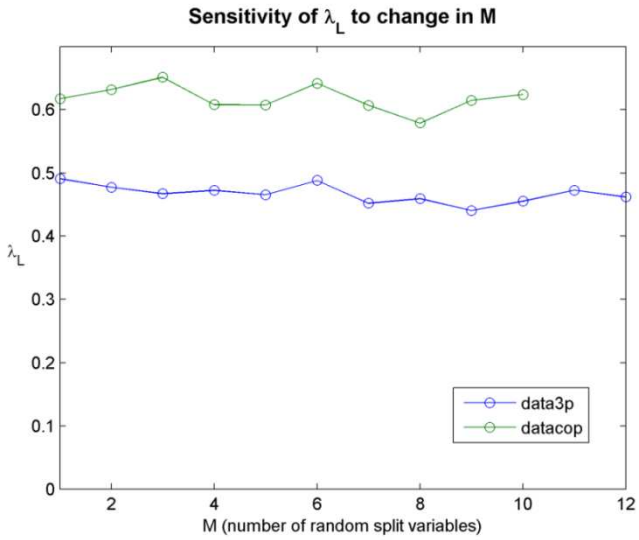

Figure A.8: Sensitivity of linear reconstruction correlation to change in M

- Nonlinear reconstruction correlation

From Figure A.9, three data sets show an upward trend for nonlinear reconstruction correlation as the number of trees increase, although the increase in this performance measure is relatively small. No clear trend is evident when changes are made to the number of random split variables (Figure A.10).

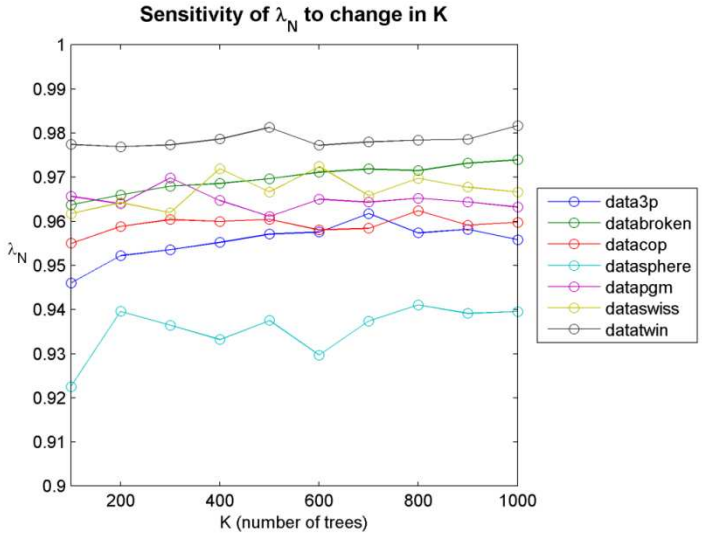

Figure A.9: Sensitivity of nonlinear reconstruction correlation to change in K

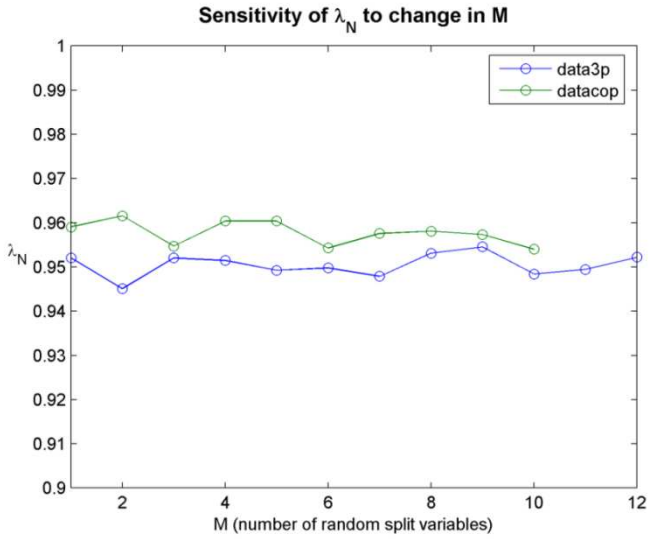

Figure A.10: Sensitivity of nonlinear reconstruction correlation to change in M

- Pairwise distance correlation

No overall trends are apparent in the pairwise correlations as a function of number of trees or number of split variables, although datatwin shows an upward trend as number of trees increases and data3p shows a downward trend as number of split variables increase (Figure A.11 and Figure A.12).

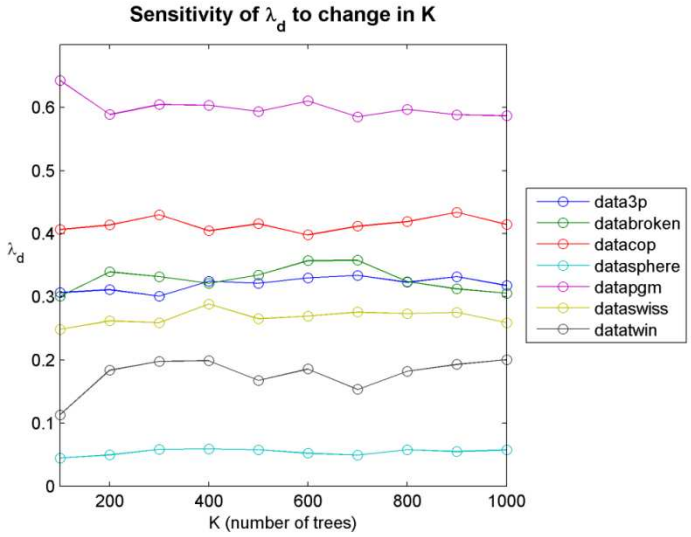

Figure A.11: Sensitivity of pairwise distance correlation to change in K

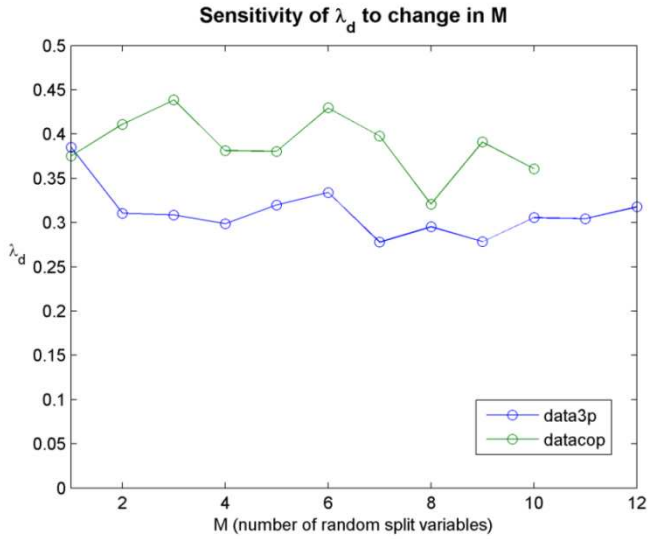

Figure A.12: Sensitivity of pairwise distance correlation to change in M

A.2.3 Strength-correlation ratio sensitivity

From Figure A.13, no trends are apparent for strength-correlation ratio as a function of the number of trees in the random forest for these data sets.

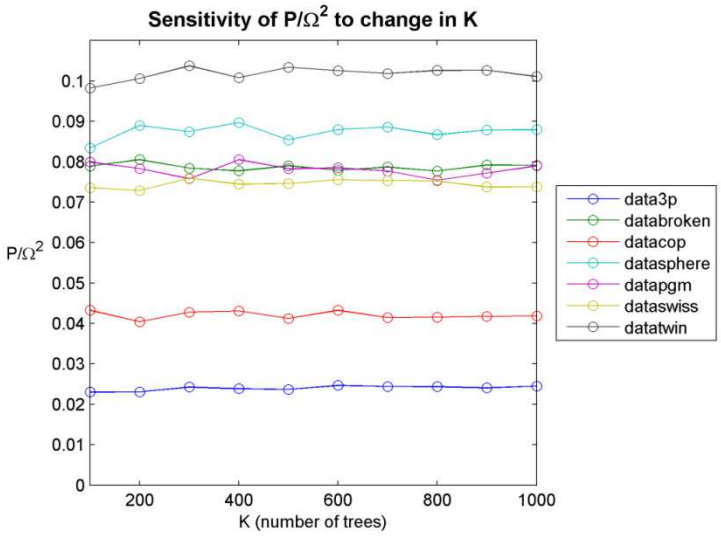

Figure A.13: Sensitivity of strength-correlation ratio to change in K

From Figure A.14, it appears that an increase in the number of split variables causes an increase in the strength-correlation ratio for data3p and datacop. One expects that as the number of random split variables increase, the correlation between trees would increase. This is due to the fact that with more random split variables to select from, there is a higher likelihood of the same split variables being used, and thus a higher likelihood of similar tree structures. For random forest feature extraction applied to data3p and datacop, the increase in correlation due to more random split variables outweighed the possible increase in tree accuracy (and thus overall forest strength).

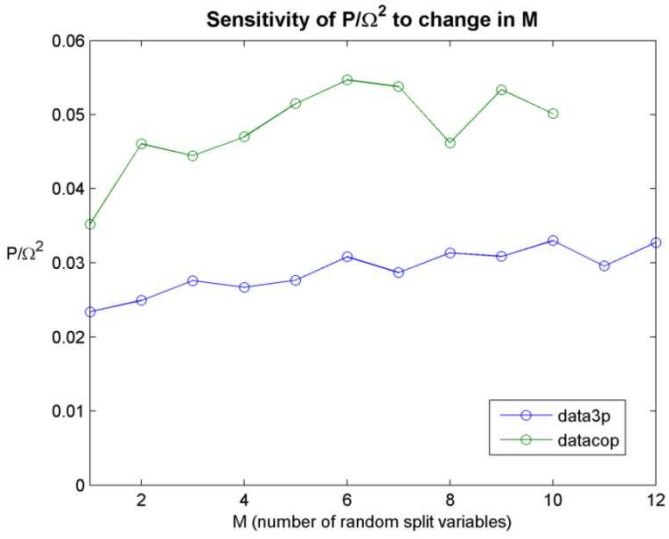

Figure A.14: Sensitivity of strength-correlation ratio to change in M

Nomenclature

K	size of ensemble
M	number of random split variables
P	average correlation of forest
ℓ	left child node
κ	local structure preservation measure: 1-nearest neighbour error
λ_d	global structure preservation measure: distance correlations
λ_L	global structure preservation measure: linear reconstruction correlation
λ_N	global structure preservation measure: nonlinear reconstruction correlation

APPENDIX B - EXTENDED RANDOM FOREST FEATURE EXTRACTION

Appendix B presents work done on expanding the random forest feature extraction algorithm to incorporate additional tree-structure information, in terms of common lineage and impurity landscapes. Two novel heuristic algorithms are proposed, and applied to three data sets.

B.1 Overview

In this study, the use of random forest models to visualize data is considered. As an analytical method, these machine learning methods have a relatively recent origin and have mostly been used for predictive modeling, for which they were primarily designed (Breiman, 2001). However, they can also be used for unsupervised feature extraction and hence visualization of multivariate data sets. Many applications have used unsupervised RF dissimilarities and extracted features, often in combination with clustering algorithms. These include visualization and clustering of microarray data (Abba et al., 2007; Amaratunga et al., 2008b), classification of protein interactions (Qi et al., 2005), identification of factors affecting the recurrence of cancer (Shi et al., 2005; Seligson et al., 2009), clustering for eye disease classification (Curnow et al., 2005) as well as anomaly detection in aircraft flight event logs (Bonissone & Iyer, 2007).

These applications are based on so-called terminal node dissimilarities of the random forest tree members (Breiman & Cutler, 2003). The purpose of this study is to investigate new ways in which dissimilarity information can be obtained from tree members.

B.2 Multidimensional scaling

Given the terminal node dissimilarities obtained from random forests, multidimensional scaling (MDS) can be used to extract features to enable data visualization. The premise of MDS is to find an embedding of the *N* samples in a dimensions whilst preserving the dissimilarity structure of the data. The preservation of dissimilarities is expressed as a stress function $\mathbb{S}$ of the dissimilarities and embedded distances. Two variants of MDS are considered here: classical (ratio) scaling (CMDS) and nonmetric (ordinal) scaling (NMDS). Let **T** be the embedded feature matrix of *N* samples in *d* dimensions, and D_{ij} be a certain type of dissimilarity between samples *i* and *j*, with **D** the full *N* by *N* dissimilarity matrix. The stress functions for CMDS ($\mathbb{S}_C$) and NMDS ($\mathbb{S}_N$) are (Borg & Groenen, 2005):

$$\mathbb{S}_C = \sum_{i>j} \left(D_{ij} - \left\| \mathbf{T}_{i.} - \mathbf{T}_{j.} \right\|_2 \right)^2 \qquad \textit{Eqn. 58}$$

$$\mathbb{S}_N = \sum_{i>j} \left(f(D_{ij}) - \left\| \mathbf{T}_{i.} - \mathbf{T}_{j.} \right\|_2 \right)^2 \qquad \textit{Eqn. 59}$$

Above, $\|.\|_2$ is the L_2 or Euclidian distance, and f(.) is a monotone regression function.

Minimizing $\mathbb{S}_C$ is attractive, as there is an explicit solution in terms of eigenvectors, whereas $\mathbb{S}_N$ is minimized by an iterative approach. The transformation f(.) of NMDS considers only dissimilarity ranking, thus not relying on a rigorous interpretation of the interval values of the dissimilarities. This allows for a more overall representative visualization of the data. However, NMDS can produce degenerate solutions for very small (and similar) within-cluster dissimilarities coupled with very large between-cluster dissimilarities, especially when the original dimensionality of Euclidian dissimilarities is high relative to the number of samples (Borg & Groenen, 2005).

Feature extraction from RF dissimilarities $\mathcal{D}^*$ with CMDS was suggested by Breiman and Cutler (2003), while Shi and Horvath (2006) also investigated feature extraction with NMDS. The premise of this study is that RF feature extraction can be improved (in terms of visualization and clustering criteria) by incorporating more tree structure information in the dissimilarity calculations. The extension of the RF dissimilarity calculation is presented here.

B.3 Node-based dissimilarities

Let tree $\mathcal{T}$ contain N nodes ($n_1, n_2, \ldots, n_N$). A binary node membership matrix **M** for tree $\mathcal{T}$ can be constructed to indicate whether a sample is present in a specific node or not. **M** consists of N rows and N columns.

$$M_{ij} = \iota\left(\mathbf{X}_{i.} \in n_j\right) \qquad \textit{Eqn. 60}$$

A special subset of **M** is the terminal node membership matrix **M***. **M*** consists of N rows and N^* columns, where N^* is the number of terminal nodes ($t_1, t_2, \ldots, t_{N^*}$) for tree $\mathcal{T}$.

$$M^*_{ij} = \iota\left(\mathbf{X}_{i.} \in t_j\right) \qquad \textit{Eqn. 61}$$

A measure of dissimilarity between any two samples i and j can be found by determining whether samples i and j report to the same terminal node t. This dissimilarity D_{ij}^* can be calculated from **M***:

$$D^*_{ij} = \left\|\mathbf{M}^*_{i.} - \mathbf{M}^*_{j.}\right\|_1 \qquad \textit{Eqn. 62}$$

Above, $\|.\|_1$ is the L_1 or Manhattan distance.

An N by N dissimilarity matrix (based on terminal node correspondence) **D*** can be constructed by calculating all pairwise sample dissimilarities, as in Eqn. 62. By accumulating the dissimilarity matrices over all K tree members of a forest, an ensemble dissimilarity $\mathcal{D}^*$ can be obtained:

$$\mathcal{D}^* = \frac{\sum_{k=1}^{K} \mathbf{D}^*(\mathcal{T}_k)}{K} \qquad \textit{Eqn. 63}$$

By only considering out-of-bag samples in the calculation of dissimilarities (and scaling accordingly), Eqn. 63 corresponds to the random forest proximity approach as suggested by Breiman and Cutler (2003).

By only accounting for terminal node membership in the dissimilarity calculation, the information on how many generations of parent nodes two samples have in common is not considered. In effect, this results in a very local measure of dissimilarity per tree. The ensemble averaging of the tree dissimilarities extends the locality of the dissimilarity somewhat, due to the diversity of trees and leaf node subspaces induced by bagging and random split selection.

To achieve a more global measure of dissimilarity, **D**, one can simply apply Eqn. 62 to **M** rather than **M***:

$$D_{ij} = \left\|\mathbf{M}_{i.} - \mathbf{M}_{j.}\right\|_1 \qquad \textit{Eqn. 64}$$

This path dissimilarity D_{ij} between samples i and j represents the number of branches between the terminal nodes to which sample i and j reports. As in Eqn. 63, the tree path dissimilarities can be accumulated to obtain an ensemble path dissimilarity $\mathcal{D}$:

$$\mathcal{D} = \frac{\sum_{k=1}^{K} \mathbf{D}(\mathcal{T}_k)}{K} \qquad \textit{Eqn. 65}$$

B.4 Impurity-based dissimilarities

By calculating the L_1 norm of **M**, the dissimilarities between any node n and parent p combination in a tree are considered equal, i.e. branches have constant lengths. To capture the heterogeneity of branches and node subspaces, the impurity measure I_n and number of samples N_n per node can be considered. Let ρ_n be the scaled path length between a node n and the root node b, calculated as follows:

$$\rho_n = \rho_p + \frac{N_b - N_n}{N_b} \left| I_p - I_n \right| \qquad \textit{Eqn. 66}$$

With $\rho_b = 0$.

The motivation for the use of node impurities rather than the decrease in impurity over a split lies in the fact that the decrease in impurity includes both children nodes. The sample number scaling term aims to increase path length for nodes with smaller sample sizes. A node containing a small number of samples can potentially occupy a small subspace in the input space, compared to a node with a large number of samples. This suggests that a small node has a more unique path than a large node, motivating an increased path length.

Figure B.1 shows an example of a tree with constant path lengths, and the same tree with path lengths scaled to reflect the impurity and sample number change over subsequent splits.

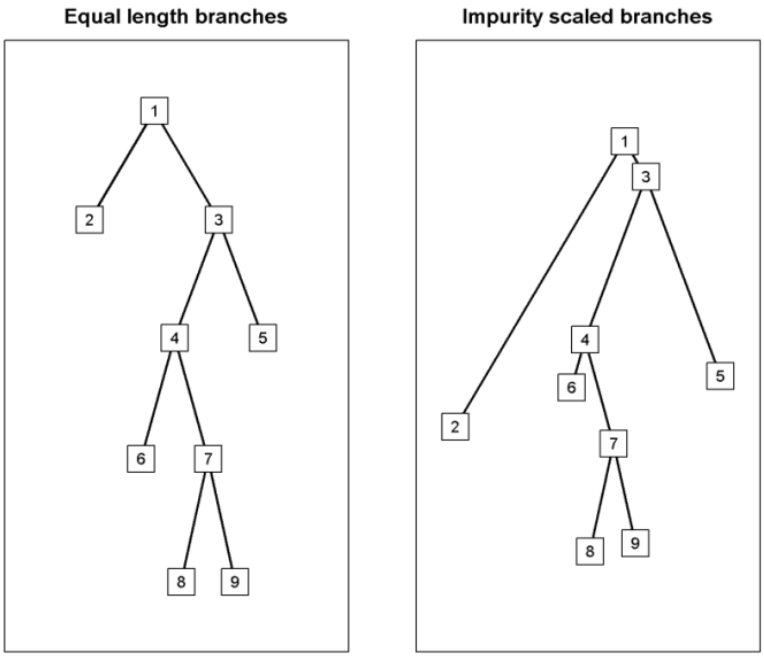

Figure B.1: Example of decision tree with equal and impurity scaled path lengths. Based on actual tree impurities.

A measure of dissimilarity Π_{ij} between two nodes n_i and n_j with a closest common ancestor a can be found from the ratio of the length of divergent paths to the length of total paths:

$$\Pi_{ij} = 1 - \frac{2\rho_a}{\rho_{n_i} + \rho_{n_j}} \qquad \textit{Eqn. 67}$$

An *N* by *N* impurity-based path dissimilarity matrix **D°** can now be constructed for any tree, with the dissimilarity between samples *i* and *j* equal to the dissimilarity of the terminal nodes these samples occupy:

$$D^{\circ}_{ij} = \Pi_{ab}\,, (\mathbf{X}_{i.} \in t_a)\ \&\ \left(\mathbf{X}_{j.} \in t_b\right) \qquad \textit{Eqn. 68}$$

An ensemble impurity-based path dissimilarity $\mathcal{D}^{\circ}$ can be obtained by accumulating **D** over all trees:

$$\mathcal{D}^{\circ} = \frac{\sum_{k=1}^{K} \mathbf{D}^{\circ}(\mathcal{T}_k)}{K} \qquad \textit{Eqn. 69}$$

B.5 Methodology

Three data sets are considered in this study, an artificial data set and two real-world data sets. The data sets are described in Table B.1, with the three-dimensional sphere data set shown in Figure B.2.

Table B.1: Description of data sets used in experiments.

Name	Description	Dimension	Samples
Sphere	Artificial data: Two clusters: a central spherical cluster surrounded by a shell cluster, with constant gap between clusters.	3	1000
Iris	Real data: Edgar Anderson's (1936) measurements of iris flowers for three species, as presented in R data sets (R Development Core Team, 2010).	4	150
Digits	Real data: Image data of handwritten digits from UCI Machine Learning Repository (Frank & Asuncion, 2010). 32x32 bitmaps of handwritten digits are divided into 4x4 blocks, and the number of pixels counted in each block. Ten classes are present, representing digits 0 to 9.	64	300

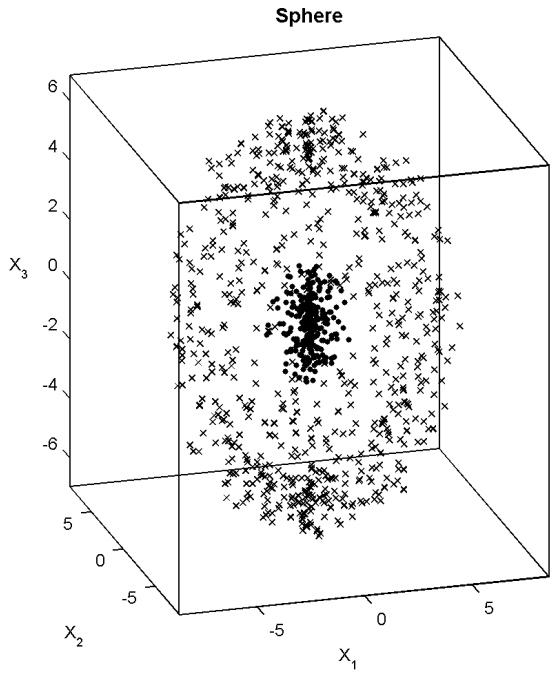

Figure B.2: Sphere data set. Plotting symbols for sphere data: • (inside cluster) and x ***(outside shell).***

The three variants of random forest dissimilarities discussed in the previous section are to be compared to CMDS and NMDS of Euclidian distances of input data. Furthermore, features are extracted from the random forest dissimilarities by both CMDS and NCMDS. These combinations result in eight feature extraction methods, as summarized in Table B.2.

Table B.2: Summary of dissimilarity feature extraction methods used in this study.

	Feature extraction with CMDS	Feature extraction with NMDS
Dissimilarities based on Euclidian distances	**D** with $\mathbb{S}_c$ (CMDS)	**D** with $\mathbb{S}_N$ (NMDS)
Dissimilarities based on terminal node membership (B = Breiman)	$\mathcal{D}^*$ with $\mathbb{S}_c$ (B-CMDS)	$\mathcal{D}^*$ with $\mathbb{S}_N$ (B-NMDS)
Dissimilarities based on tree paths (P = path)	$\mathcal{D}$ with $\mathbb{S}_c$ (P-CMDS)	$\mathcal{D}$ with $\mathbb{S}_N$ (P-NMDS)
Dissimilarities based on impurity scaled tree paths (I = impurity)	$\mathcal{D}^\circ$ with $\mathbb{S}_c$ (I-CMDS)	$\mathcal{D}^\circ$ with $\mathbb{S}_N$ (I-NMDS)

Random forests were constructed using the randomForest package (Liaw & Wiener, 2002) in R (R Development Core Team, 2010) with $\mathcal{D}^*$ calculated using the unsupervised setting of said package. The number of trees *K* were set to 1000, the number of random split variables *m* to the floored square root of *M*, with a default minimum node size of 1. To determine $\mathcal{D}$ and $\mathcal{D}^\circ$, a synthetic class of the product of marginal densities of **X** was created, and the supervised setting of the randomForest package used with class labels indicating original or synthetic data. Construction of membership and impurity matrices were done from the tree structures output. The Matlab functions **`cmdscale`** and **`mdscale`** were used for CMDS and NMDS of **D**, $\mathcal{D}^*$, $\mathcal{D}$ and $\mathcal{D}^\circ$.

The success of the different feature extraction methods can be evaluated in a number of different ways. Visual inspection of overlaid true cluster labels to the features plots can give a qualitative indication of cluster separation and distortion. A distortion metric $\mathbb{D}$ can quantify the average sum of squared discrepancy between pairwise distances in the input space $\mathbf{D}^{\mathbf{X}}$ and pairwise distances in the feature space $\mathbf{D}^{\mathbf{T}}$:

$$\mathbb{D} = \frac{2\sum_{i\neq j}\left(D_{ij}^{\mathbf{X}} - D_{ij}^{\mathbf{T}}\right)^2}{N(N-1)} \qquad \textit{Eqn. 70}$$

Here, $\mathbf{D}^{\mathbf{X}}$ and $\mathbf{D}^{\mathbf{T}}$ are scaled to a range of [0, 1].

Distortion captures global structure, with high distortion values suggesting low correspondence between input distances and output distances. In contrast, the leave-one-out 1-nearest neighbour error $\mathbb{N}$ gives an impression of local structure, as each sample is labelled in accordance to its closest neighbour. Low errors suggest that clusters exhibit a high degree of separation (Van der Maaten et al., 2009). $\mathbb{N}$ was calculated using the PRTools Pattern Recognition Toolbox for Matlab™ (Duin et al., 2007).

B.6 Qualitative inspection of projections

B.6.1 Sphere data set

The visualization of the sphere data set is considered in detail here, as the actual geometric structure is known (see Figure B.3). NMDS based on **D** gives the best depiction of the separate inside cluster and outside shell, with no overlap of these clusters.

The B-CMDS and B-NMDS projections show several disjointed clusters. The B-CMDS projection suggests an exploded view of the inside cluster and outside shell, with some overlap of the two clusters. The several disjoint clusters representing the shell are indicative of the distinct terminal node separations. The B-NMDS further emphasizes the distinct terminal node separations by increased between-cluster distances.

The P-CMDS and P-NMDS projections are more continuous than the B-CMDS and B-NMDS results. Visual inspection of the P-CMDS projection suggests the presence of three clusters: a large cluster on the left, and two smaller (and tighter) clusters on the right. Considering true cluster membership, the left cluster represents the

inner cluster of the sphere data set and the two right clusters the outer shell. The separation of the two right clusters may be due to an initial perpendicular decision tree split plane through the inner cluster. Path dissimilarities of samples separating early in decision trees will be large. The P-NMDS projection shows three clusters, again with a separation between clusters representing the outer shell.

The I-CMDS projection shows two joined clusters: a lengthened horizontal cluster of the sphere inner cluster, and a shorter vertical cluster of the sphere shell. Even though the shell cluster occupies a larger region in the original input space, the shell cluster in the I-CMDS projection is smaller than the inner cluster. The I-NMDS projection gives the best representation of the sphere data of all the tree-based approaches, an outer cluster representing the shell, and an inner cluster representing the inner cluster of the sphere. This projection is similar to the NDMS projection based on **D**.

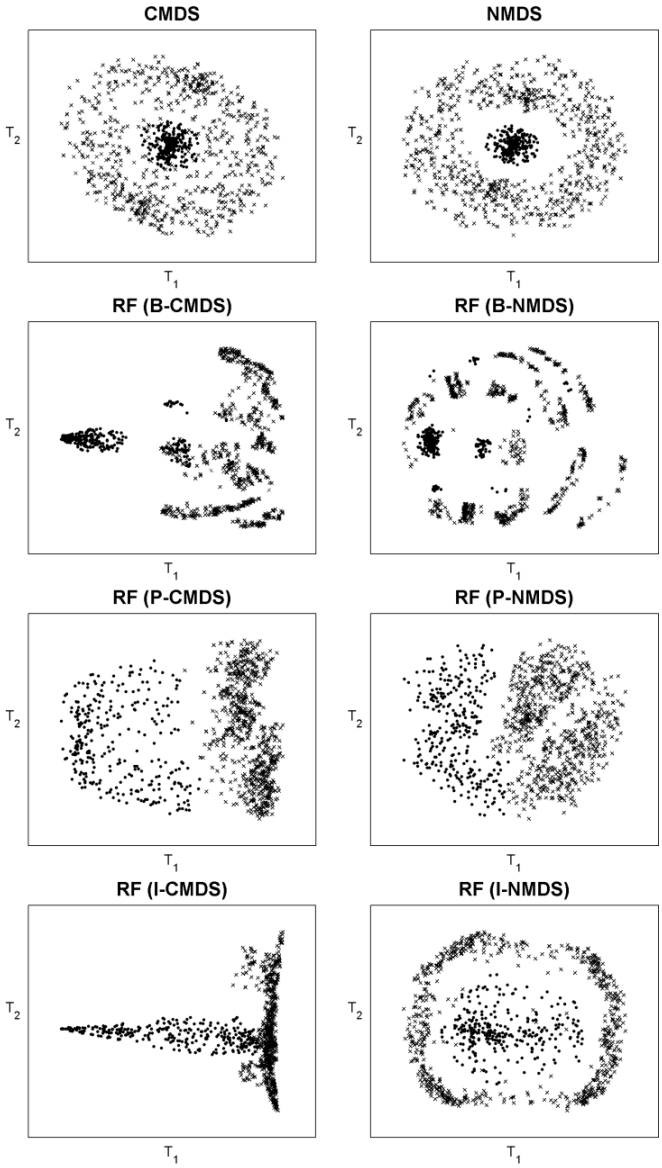

Figure B.3: Features of sphere data set. Plotting symbols for sphere data: • (inside cluster) and x ***(outside shell).***

To further illuminate the nature of the forest dissimilarities, comparisons of the forest feature projections (based on NMDS) to the original data structure in 3D are given in Figures B.4, B.5 and B.6.

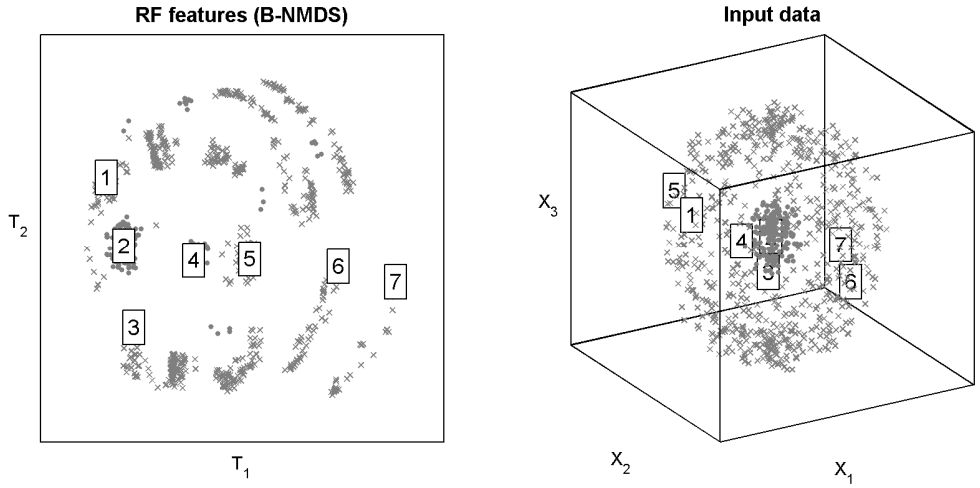

Figure B.4: B-NMDS features of sphere data set, with selected locations in original 3D data. Plotting symbols for sphere data: • (inside cluster) and* x *(outside shell).

From Figure B.4, the disjointed nature of the B-NMDS is apparent. Samples 1 and 5 are proximal in the original 3D data, and are located in the outside shell, while sample 2 is located in the inside cluster. In the B-NMDS projection, samples 1 and 2 are presented as being more proximal than samples 1 and 5, a distortion of the actual distances. From the arrangement in the feature space, it would be expected that samples 4, 5, 6 and 7 would be arranged in that order from the inside cluster (represented by sample 2). However, in the original 3D representation, samples 6 and 7 are on the opposite outside shell side as samples 4 and 5.

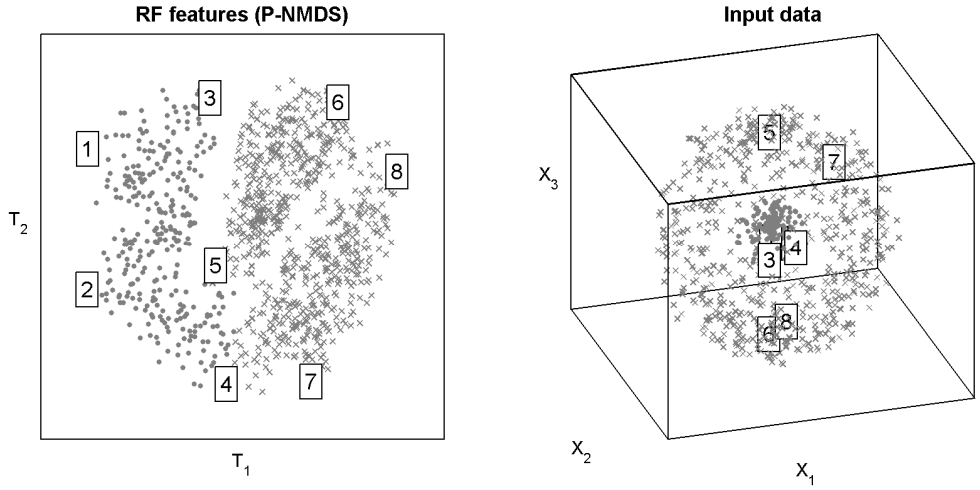

Figure B.5: P-NMDS features of sphere data set, with selected locations in original 3D data. Plotting symbols for sphere data: • (inside cluster) and* x *(outside shell).

From the feature space and original 3D data comparison in Figure B.5, the following is highlighted. Samples 5 and 6 belong to the same cluster in the feature space, but are on opposite sides of the outside shell in the original 3D data. The same is true for samples 7 and 8. Samples 5 and 7, which are proximal in the feature space, are also proximal in the original 3D data. The same is again true for samples 6 and 8. Samples 3 and 4 are proximal in the original 3D data, but distant in the feature space. The presence of the two clusters in feature space representing the outside shell may be due to initial tree-splits as separating planes bisecting samples 5 and 7; 3 and 4 as well as 6 and 8. However, the continuous nature of the feature space samples can be considered an improvement on the B-NMDS feature space.

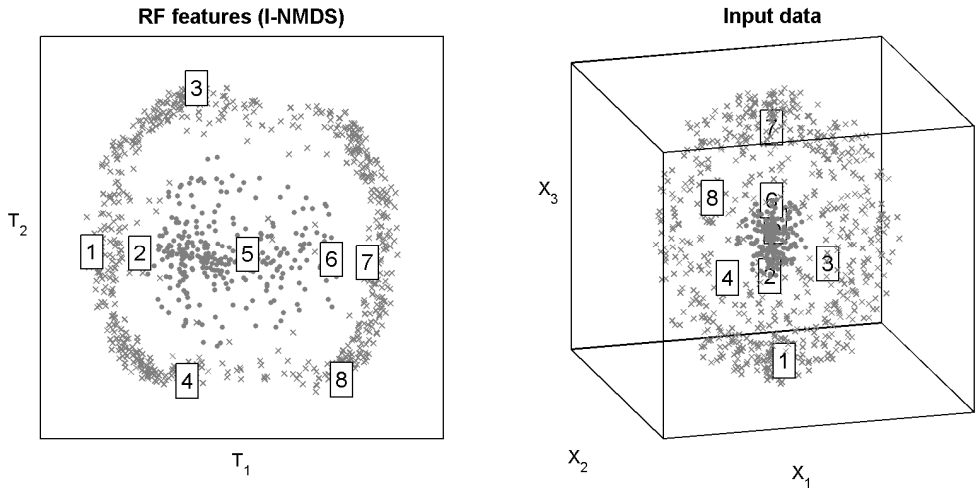

Figure B.6: I-NMDS features of sphere data set, with selected locations in original 3D data. Plotting symbols for sphere data: • (inside cluster) and* x *(outside shell).

A comparison of I-NMDS features and the original 3D is given in Figure B.6. The I-NMDS features show less distortion than the B-NMDS and P-NMDS features. Samples 1 and 7 are on opposite sides of the outside shell in both the feature space and the original 3D data. Sample 4 is located between samples 1 and 8 in both the feature space and the original 3D data. Distortion is present in the feature space in terms of the small gap between the inside cluster and the outside shell. This is an artefact of the isotonic regression of the NMDS feature extraction procedure.

Another comparison of the properties of the forest features and actual data can be made in terms of the distance distributions of the actual data and features. For this purpose, pairwise distances were calculated for the original input data, and in the feature spaces derived by the various forest approaches. The pairwise distances were scaled to the range [0, 1], and kernel density estimation applied (with a Gaussian kernel and bandwidth of 0.1) to all distributions. The results for the sphere data set are presented in Figure B.7.

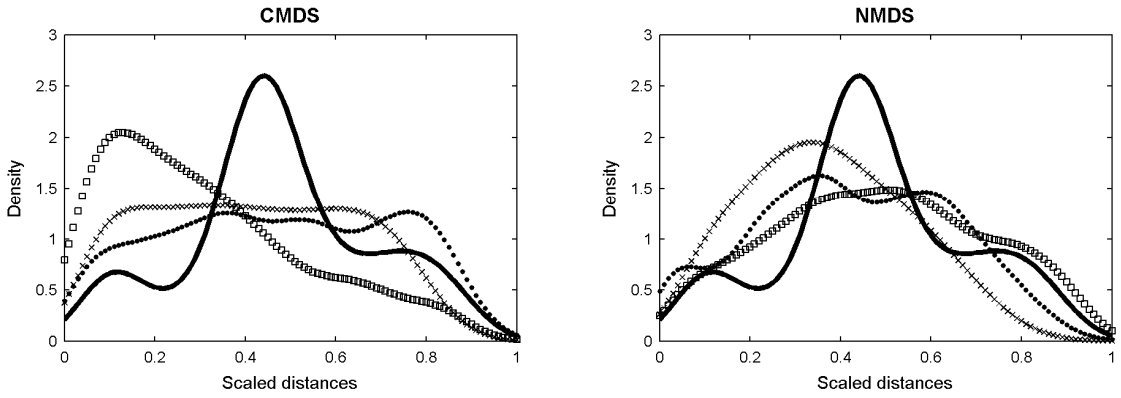

Figure B.7: Scaled actual and forest feature distance distributions of sphere data set. The solid line represents the distribution of the actual distances. Plotting symbols: • represents feature distance distributions derived from $\mathcal{D}^*$ (B-CMDS and B-NMDS);* x *represents feature distance distributions derived from $\mathcal{D}$ (P-CMDS and P-NMDS); □ represents feature distance distributions derived from $\mathcal{D}^\circ$ (I-CMDS and I-NMDS).

From Figure B.7, the actual data distance distribution is observed as trimodal, while multimodality is exhibited by B-CMDS, B-NMDS and I-NMDS. The first mode at small scaled distances represents the intracluster distances of the inside cluster. The second mode at medium scaled distances represents the intercluster distances

between the inside cluster and the outside shell. The third mode at large scaled distances represents the intracluster distances of the outside shell, especially for sample pairs on opposing sides of the shell. The B-NMDS and I-NMDS distance distributions are improvements on the B-CMDS and I-CMDS distance distributions (as evaluated on basis of agreement with the actual distance distribution). The small gap between the inside cluster and outside shell in the I-NMDS projection (as discussed previously, and seen in Figure B.6), is reflected in the lack of clear multimodality of the I-NMDS distribution in Figure B.7.

B.6.2 Iris and digits data sets

The features for the iris and digits data sets are presented in Figure B.8 and Figure B.9, respectively. For the iris data set, none of the visualization techniques are able to show a projection with three distinct clusters, with the two clusters depicted with plotting symbols x and Δ showing some degree of overlap for all projections. A suggestion of the horseshoe effect (see Chapter 8) is seen in the B-CMDS projection for the x and Δ clusters, suggesting that these dissimilarities are ordered and emphasise local dissimilarity. The P-CMDS projection shows less of this horseshoe effect, with a wider cluster for the x and Δ data.

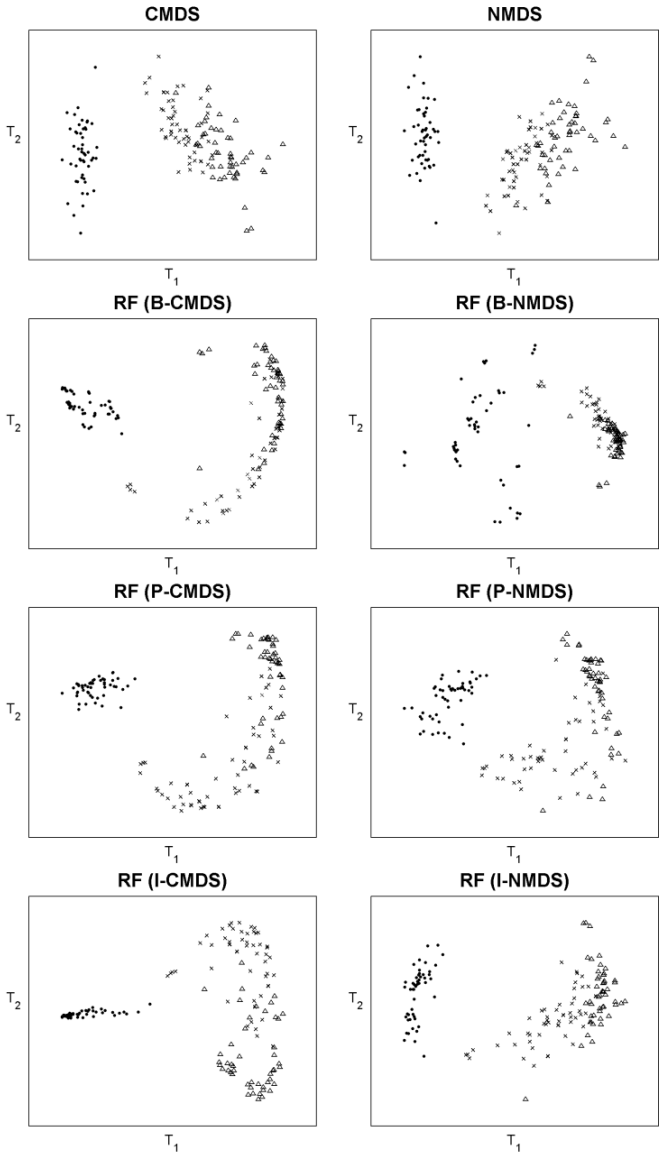

Figure B.8: Features of iris data set. Plotting symbols for iris roll data: •, x and Δ for different species of iris flowers.

As with the B-NMDS projection of the sphere data, the B-NMDS projection of the iris data shows several small clusters for one known cluster. Again, this indicates the disjoint terminal node properties of the decision trees and derived dissimilarities. The B-NMDS, P-NMDS and I-NMDS projections all show bigger cluster sizes for the • cluster, as compared to the B-CMDS, P-CMDS and I-CMDS projections. This suggests that NMDS emphasizes small dissimilarities in afore-mentioned projections.

For the digits data set, an overall conclusion from the projections in Figure B.9 is that no technique is successful in depicting distinct clusters for the different digits. NMDS shows a degenerate solution, possibly due to the large dimensionality and large frequency of "similar" dissimilarities (as discussed before). Visibly distinct groupings with the tree-methods include the digits 0 and 4 for B-CMDS, the digit 1 for P-CMDS and the digits 0 and 4 for I-CMDS.

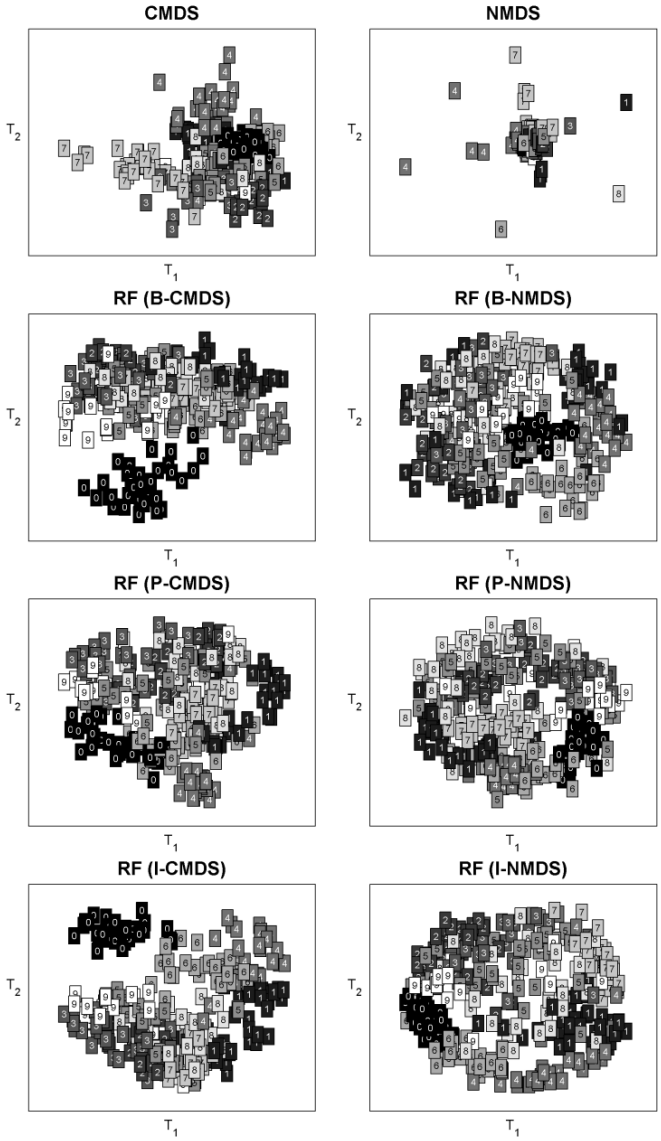

Figure B.9: Features of digits roll data set. Plotting symbols are the numerals 0 to 9, corresponding to the images of handwritten digits 0 to 9.

Comparisons of the distance distributions of the forest features and actual data for the iris and digits data are given in Figures B.10 and B.11, based on kernel density estimation as discussed before.

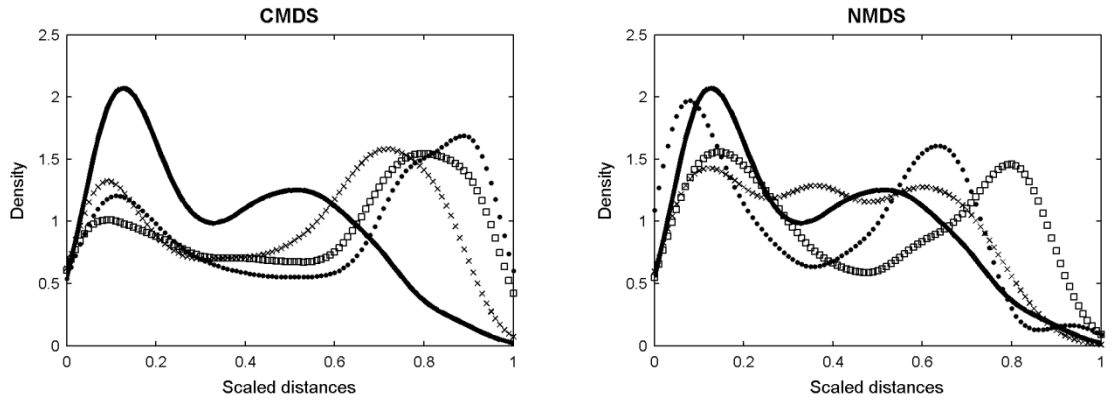

Figure B.10: Scaled actual and forest feature distance distributions of iris data set. The solid line represents the distribution of the actual distances. Plotting symbols: • represents feature distance distributions derived from $\mathcal{D}^*$ (B-CMDS and B-NMDS); x represents feature distance distributions derived from $\mathcal{D}$ (P-CMDS and P-NMDS); □ represents feature distance distributions derived from $\mathcal{D}^\circ$ (I-CMDS and I-NMDS).

From Figure B.10, the actual iris data distance distribution is observed as bimodal. All forest feature distance distributions, save that of P-NMDS, are also bimodal. The bimodal distribution suggests the two clusters (one of a single species, and another of two seemingly inseparable species) present in these forest projections. P-NMDS shows a trimodal distance distribution, which may indicate more successful separation of the true three clusters present in the iris data. As with the sphere data, NMDS shows improved distance distributions (closer agreement with the actual distance distributions) as compared to the CMDS profiles.

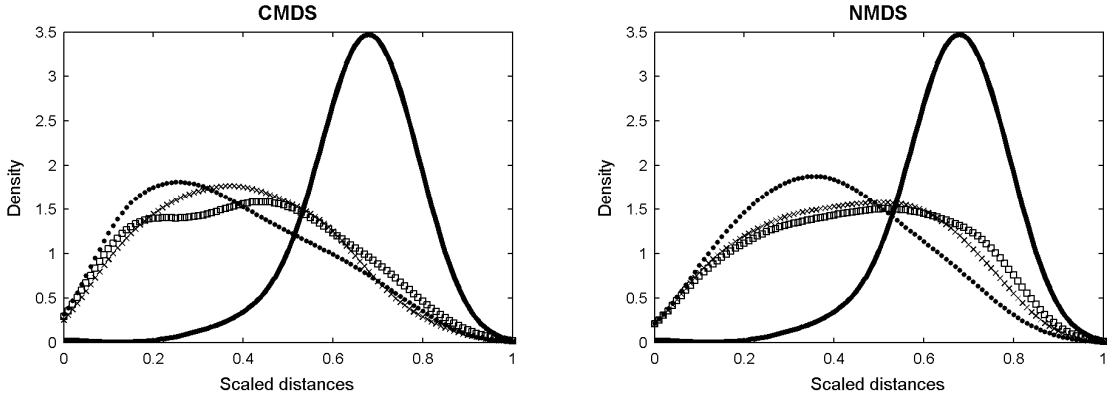

Figure B.11: Scaled actual and forest feature distance distributions of digits data set. The solid line represents the distribution of the actual distances. Plotting symbols: • represents feature distance distributions derived from $\mathcal{D}^*$ (B-CMDS and B-NMDS); x represents feature distance distributions derived from $\mathcal{D}$ (P-CMDS and P-NMDS); □ represents feature distance distributions derived from $\mathcal{D}^\circ$ (I-CMDS and I-NMDS).

From Figure B.11, the actual digits data distance distribution is observed as unimodal, with the highest density of distances moderately large. The unimodal nature of the actual data is due to the scarcity of the 64-dimensional space, as well as the overlapping nature of the digit classes. The "crowding" phenomenon may be at play in the mostly unsuccessful visualization of the digits data. The forest feature distance distributions are mostly (save I-CMDS) unimodal as well, but skewed towards the smaller distance direction. The slight bimodality of the I-CMDS distribution is interpreted as the presence of the clear 0-digit cluster, as seen in Figure B.9.

B.7 Quantitative comparison of projections

The distortion $\mathbb{D}$ and leave-one-out 1-nearest neighbour errors $\mathbb{N}$ for all techniques and all data sets are given in Table B.3.

Table B.3: Visualization criteria results. With bold text indicating the best tree-based performance for each criterion for each data set.

Sphere	$\mathbb{D}$	$\mathbb{N}$	**Iris**	$\mathbb{D}$	$\mathbb{N}$	**Digits**	$\mathbb{D}$	$\mathbb{N}$
CMDS	0.019	0.058	CMDS	0.0001	0.128	CMDS	0.0235	0.560
NMDS	0.011	0.000	NMDS	0.0001	0.121	NMDS	0.0840	0.530
B-CMDS	0.077	0.040	B-CMDS	0.0656	0.243	B-CMDS	0.0435	0.617
B-NMDS	0.034	**0.003**	B-NMDS	0.0257	0.237	B-NMDS	**0.0338**	0.470
P-CMDS	0.078	0.004	P-CMDS	0.0380	0.081	P-CMDS	0.0427	0.523
P-NMDS	0.061	0.004	P-NMDS	**0.0196**	0.108	P-NMDS	0.0573	**0.407**
I-CMDS	0.097	0.018	I-CMDS	0.0646	0.108	I-CMDS	0.0474	0.477
I-NMDS	**0.028**	0.014	I-NMDS	0.0201	**0.047**	I-NMDS	0.0498	0.447

It is observed from Table B.3 that, in general, NMDS decreased both the distortion and 1-nearest neighbour error for all data sets and all types of dissimilarities (**D**, $\boldsymbol{\mathcal{D}}^*$, $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{D}}^\circ$), as compared to CMDS. This suggests that NMDS can better capture global and local structure from **D**, $\boldsymbol{\mathcal{D}}^*$, $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{D}}^\circ$ than CMDS. However, this generalization is not true for the digits data set, where distortion increased when NMDS was applied to **D**, $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{D}}^\circ$. The high dimensionality (64 dimensions) may be to blame, as discussed before.

B.8 Conclusions

The purpose of this study was to investigate new ways in which dissimilarity information can be obtained from tree members. Two new methods for forest dissimilarity measures were developed, namely path dissimilarities and impurity scaled dissimilarities. The developed dissimilarities show further potential for data visualization from single decision trees and for supervised learning approaches, and will be the subject of future work.

Classical and nonmetric multidimensional scaling were employed to obtain features from these dissimilarities. From three case studies, the new methods were shown to improve global and local structure preservation, as compared to the original random forest feature extraction method. Nonmetric multidimensional scaling was further shown to generally result in lower global and local structure distortions than classical multidimensional scaling.

The improvement in data visualization with the new forest dissimilarity techniques can be attributed to the exploitation of more tree structure information, to deliver more globally accurate projections. A trade-off may exist between, on the one hand, capturing global information from tree-structures and, on the other hand, going too far and merely recapturing the original Euclidian distances. This trade-off may reside in the choice of node size.

Nomenclature

$\mathcal{a}$	closest common ancestor node
$\mathcal{b}$	root node
d	feature space dimensionality
$\mathbf{D}$	generic dissimilarity matrix (often Euclidian distances)
$\mathcal{D}$	ensemble path dissimilarity matrix
$\mathcal{D}^*$	ensemble terminal node correspondence dissimilarity matrix
$\mathcal{D}^\circ$	ensemble impurity scaled path dissimilarity matrix
$\mathbb{D}$	distortion metric
$\mathbf{D}$	tree path dissimilarity matrix
$\mathbf{D}^*$	tree terminal node correspondence dissimilarity matrix
$\mathbf{D}^\circ$	tree impurity scaled path dissimilarity matrix
I	impurity measure
J	number of classes
k	ensemble member indicator
K	size of ensemble
ℓ	left child node
$\mathbf{L}$	learning sample
$\mathbf{M}$	node membership matrix
$\mathbf{M}^*$	terminal node membership matrix
n	bootstrap sample size
$\mathcal{n}$	node
$\mathbb{N}$	leave-one-out 1-nearest-neighbour error
N	sample size
N	number of nodes in tree
N^*	number of terminal nodes in tree
p	proportion of samples
$\mathcal{r}$	right child node
$\mathbb{S}$	multidimensional scaling stress function
$\mathcal{t}$	terminal node
$\mathbf{T}$	matrix of features
$\mathcal{T}$	decision tree
$\mathbf{X}$	matrix of input space samples
$\mathbf{y}$	vector of output samples
$\mathbf{y}'$	predicted response variable
Δ	impurity decrease
$\boldsymbol{\theta}$	random vector
ι	indicator function
Π	impurity-path node dissimilarity
ρ	scaled path length

APPENDIX C – PUBLICATIONS AND PRESENTATIONS BASED ON PHD RESEARH

C.1 Papers submitted to international peer-reviewed journals

- Auret, L. and Aldrich, C. 2010. Change point detection in time series data with random forests. *Control Engineering Practice*, 18(8), 990 - 1002. (Published).
- Auret, L. and Aldrich, C., 2010. Unsupervised process fault detection with random forests. *Industrial & Engineering Chemistry Research,* doi: 10.1021/ie901975c. (Article in press).
- Auret, L. and Aldrich, C. 2010. Empirical comparison of tee ensemble variance importance measures. *Chemometrics and Intelligent Laboratory Systems*. (Submitted).
- Auret, L. and Aldrich, C. 2010. Interpretation of nonlinear relationships between process variables by use of random forests. *Minerals Engineering*. (Submitted).

C.2 Full-length peer-reviewed papers in conference proceedings

- Auret, L. and Aldrich, C. 2010. Diagnostic monitoring of concentrator circuits with random forest models. *International Mineral Processing Congress*, Brisbane, Australia, 6-10 September 2010.
- Auret, L. and Aldrich, C. 2010, Fault detection and diagnosis with random forest feature extraction and variable importance methods. *Proceedings of the 13th Symposium on Automation in Mining, Mineral and Metal Processing (MMM 2010)* (http://ifacmmm2010.com)*,* Cape Town, South Africa, 2-4 Aug 2010.
- Auret, L. and Aldrich, C. 2009. Process fault diagnosis using random forests. *South African Chemical Engineering Congress* (www.sacec2009.org), Cape Town, Western Cape, South Africa, 20-22 September, published on CD-ROM, ISBN: 978-1-920355-21-0.

C.3 Non-peer-reviewed presentations at conferences and symposia

- Auret, L. and Aldrich, C., 2010, Process variable direction of influence: tree-ensemble partial dependence plots. *Mineral Processing 2010*, [Western Cape branch of Southern African Institute of Mining and Metallurgy, Cape Town, South Africa, 5-6 Aug 2010]
- Auret, L. and Aldrich, C. 2009. Detecting regime shifts in dynamic process systems with random forest models. *Stellenbosch Statistical Symposium*, [Stellenbosch, South Africa, 24-26 Aug 2009].
- Auret, L. and Aldrich, C., 2009, Change point detection in mineral processes using random forests. *Mineral Processing 2009*, [Western Cape branch of Southern African Institute of Mining and Metallurgy, Cape Town, South Africa, 6-7 Aug 2009]
- Auret, L. and Aldrich, C., 2008, Visualization of data with random forests. *1st Symposium of the South African Chemometrics Society*, [Stellenbosch, South Africa, 1-5 Dec 2008]
- Auret, L. and Aldrich, C., 2008, Visualization of the behaviour of a concentrator circuit on a platinum plant by use of random forests. *Mineral Processing 2008*, [Western Cape branch of Southern African Institute of Mining and Metallurgy, Cape Town, South Africa, 7-8 Aug 2009]
- Auret, L. and Aldrich, C., 2008, Identification of influential variables on a concentrator circuit on a platinum plant by use of random forests. *Mineral Processing 2008*, [Western Cape branch of Southern African Institute of Mining and Metallurgy, Cape Town, South Africa, 7-8 Aug 2009], (Best poster paper award of the conference).

"The ideal engineer is a composite. He is not a scientist, he is not a mathematician, he is not a sociologist or a writer; but he may use the knowledge and techniques of any or all of these disciplines in solving engineering problems."

- N.W. Dougherty