Mixtures of Probabilistic Principal Component Regression: Application in Optimality Assessment

by

Shabnam Sedghi

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

Master of Science

in

Process Control

Department of Chemical and Materials Engineering

University of Alberta

Abstract

Performance of the operating processes may change by time due to uncertainties and process condition changes. Hence, online operating performance assessment has attracted attentions from academia and industry. One of the main ingredients of performance assessment is optimality assessment. On one hand, the optimal condition for an operating process can be estimated by known process optimization methods as an initial design. On the other hand, performance may alter from the optimal design due to disturbances, process condition changes or product driven operating mode changes. As a result, optimality assessment, i.e., monitoring the operating process performance in terms of optimality is of great importance. The main objective of this thesis is to develop a general framework for optimality assessment in multi-mode systems with non-Gaussian behavior by employing probabilistic principal component regression (PPCR) method.

High dimensionality of the process datasets, multiple operating regions caused by uncertainties and simultaneous missing inputs and outputs due to the device failure or delays in measuring certain variables are some of the challenges in optimality assessment. Mixture semi-supervised probabilistic principal component regression (MSPPCR) model is employed that inherently addresses high dimensionality, multimodal behavior and missing outputs. In addition, it is developed under expectation maximization (EM) framework in order to deal with simultaneous missing inputs and outputs. The proposed model is capable of making the most use of all available information for predictive model building.

In many processes, steady state operating modes do not follow Gaussian distribution since they have different operating regions that are caused by uncertainties. Due to the lack of information regarding operating regions, a hierarchical mixture PPCR method is proposed in order to automatically estimate the number of operating regions, and the parameters are estimated through maximum a posteriori (MAP) principle under EM framework that incorporates prior distributions. This method is based on a divisive hierarchical algorithm; however, a merging step is proposed in order to control splitting steps and avoid overestimation of the number of mixture components. Due to its hierarchical nature, a prior knowledge of the possible range of the number of components is not required compared to the traditional methods. Moreover, it is capable of detecting overlapped components because of utilizing minimum message length criterion (MML) as the selection criterion.

A probabilistic framework for optimality assessment and non-optimum cause diagnosis for multi-mode processes with non-Gaussian behavior is proposed. In this framework, operating regions are compared with operating modes that are caused by uncertainties and known governing factors, respectively. Density based clustering (DENCLUE) method is modified and improved for offline operating mode detection. In addition, a predictive operating modes classifier is built based on modified mixture discriminant analysis (MclustDA) method, and it is incorporated with process knowledge in order to improve estimation. For optimality analysis and prediction, MSP-PCR model is employed for steady state modes, and dynamic principal component regression (DPCR) is employed for transitions. A probabilistic framework through sequential forward floating search (SFFS) method is adopted for non-optimum cause diagnosis. The proposed method is capable of optimality assessment for general high dimensional multi-modal systems with non-Gaussian behavior.

Finally, the performance and validity of the proposed methods are verified through various numerical, simulation and industrial examples.

Preface

The materials of chapters 2 and 4 are submitted as the following publications:

1. Submitted as S. Sedghi, A. Sadeghian, B. Huang. “Mixture Semisupervised Probabilistic Principal Component Regression Model with Missing Inputs”. *Computers & Chemical Engineering.*

2. Submitted as S. Sedghi, B. Huang. “Real-time Assessment of Process Operating Performance”. *FOCAPO/CPC 2017.*

I was responsible for the data collection and analysis as well as the manuscript composition of all the publications. Dr. Biao Huang was the supervisory author and was involved with manuscript composition.

Acknowledgements

First and foremost, I would like to thank my supervisor Prof. Biao Huang who kindly gave me the opportunity to investigate my research interest and guided me patiently to the destination through his encouraging comments. I am very thankful to him for his supportive attitude and inspiration during my studies. It was a great pleasure for me to explore my graduate studies under his supervision.

It is my honor to be a member of the Computer Process Control (CPC) group where we broaden our knowledge in various reseach aspects and enjoy participating in discussions. I would like to acknowledge my colleagues Mohammad Rashedi, Anahita Sadeghian, Nima Sammaknejad, Elham Naghoosi, Ruomu Tan, Yanjun Ma, Nabil Magbool Jan, Rishik Ranjan, Shekhar Sharma, Fadi Ibrahim and many others for their help and support.

I would like to acknowledge the Department of Chemical and Materials Engineering, University of Alberta, for giving me the opportunity to pursue my Master's study in a pleasant environment. I would like to thankfully acknowledge the Natural Sciences and Engineering Research Council of Canada for the financial support.

I would like to thank my parents for their unconditional love and support. They always encouraged me to pursue my dreams and have confidence in my abilities. I would also like to thank Shiva, my lovely sister and my best friend, who kindly supports me in every aspect of my life.

Last but not least, I would like to thank my dear husband Vahid Vajihinejad for giving me faith and support to pass difficult days and making every moment enjoyable to me. Thank you for your unlimited support, love and encouragement in following my dreams.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Industrial processes are affected by disturbances or process condition changes that result in the deviation of their performance. As a result, developing methods of online operating performance assessment is of great importance in both academia and industry. Many methods have been investigated for process optimization in order to find the optimal condition for process operation. However, the process performance may deviate from the initial design due to uncertainties and process condition changes. Monitoring the process performance on optimality is called optimality assessment. Moreover, safety assessment is another important ingredient of the performance assessment. Safety analysis based on qualitative and quantitative methods as well as process monitoring have been investigated in many studies; hoewever there are few studies on optimality assessment.

Some challenges are associated with optimality assessment of industrial processes. First, high dimensionality of the process datasets is common in data driven analysis. Second, in many practical problems, the optimality index related variables are often measured slowly compared to other process variables. Moreover, some input variables may have missing values due to the device failure and so on. Third, in many processes, main operating modes do not follow uni-modal Gaussian distributions due to non-Gaussian disturbances. In other words, in each operating mode, there are several operating regions that are caused by uncertainties. Fourth, the change of operating mode happens in some processes because of the operating condition changes, different product demands, etc that causes different steady state modes and transitions. The main objective of this thesis is to address the above mentioned issues associated with optimality assessment based on statistical data driven modeling methods and develop a systematic algorithm for it.

1.2 Thesis outline

The rest of this thesis is organized as follows:

In chapter 2, the mixture semi-supervised probabilistic principal component regression (MSPPCR) method is developed in order to tackle high dimensionality, multimodal behavior and simultaneous missing inputs and outputs of the process datasets in optimality assessment. This method is developed based on the expectation maximization (EM) algorithm in the case of multi-mode operation, and simultaneous missing data in both inputs and outputs of the historical dataset. The developed method is applicable when some input variables have missing data completely at random in addition to multi-modal and semi-supervised cases that have been investigated in literature so far. The importance of this chapter is to employ the most information of all available measurements for model building. The EM algorithm is employed since it can provide a maximum likelihood solution by iteratively updating the estimated values for the missing data based on the updated parameters of the model. Finally, applications on simulation and industrial examples are provided that have confirmed the validity of the proposed method.

The objective of chapter 3 is to estimate the number of operating regions in non-Gaussian processes in optimality assessment. This chapter provides a hierarchical mixture probabilistic principal component regression (MPPCR) method in order to estimate the number of mixture components in multimode systems. In this chapter a hierarchical MPPCR method is proposed for automatic estimation of the number of mixture components. This method is based on a divisive hierarchical algorithm, and in the initial stage, it starts with the minimum possible number of mixture components. The Minimum Message Length (MML) criterion is employed to make the decision of splitting components. Moreover, a merging step is introduced in order to control the splitting steps that prevents from overestimating the number of mixture components. The EM algorithm is employed to estimate the parameters of the developed model based on maximum a posteriori (MAP) principle. A numerial example and an experimental example are provided that have demonstrated the performance.

In chapter 4, a novel method for operating optimality assessment and non-optimum cause diagnosis is proposed. Kernel density based method for mode detection is adopted and improved in order to differentiate noise from transition, find true labels, and also increase the accuracy of estimation of the exact mode change instant. Then mixture discriminant analysis (MclustDA) method is employed to construct the predictive opearting modes classifier, and the process knowledge is incorporated in order to increase prediction accuracy. In each steady state operating mode, the proposed hierarchical MPPCR is utilized for estimating the number of operating regions that are caused by uncertainties, and MSPPCR model is employed for optimality analysis and predictive model building. Dynamic principal component regression (DPCR) model is employed for grade analysis and predictive model building in transitions. For non-optimum performance, a probabilistic framework through sequential forward floating search (SFFS) method is proposed for causal variables diagnosis. Finally, the performance is evaluated through a simulation example.

Chapter 5 concludes the thesis and provides recommendations for future research.

1.3 Submitted Publications

Materials of this thesis have been addressed in the following publications:

1. Submitted as S. Sedghi, A. Sadeghian, B. Huang. “Mixture Semisupervised Probabilistic Principal Component Regression Model with Missing Inputs”. *Computers & Chemical Engineering.* (Chapter 2 - Complete Version Except Section 2.4.3)

2. Submitted as S. Sedghi, B. Huang. “Real-time Assessment of Process Operating Performance”. *FOCAPO/CPC 2017.* (Chapter 4 - Short Version)

1.4 Main Contributions

The main contributions of this thesis can be summarized as:

1. Development of MPPCR in order to deal with simultaneous missing inputs and outputs of the process datasets.

2. Development of hierarchical MPPCR through introducing merging and splitting steps for automatic estimation of the number of mixture components in multimode systems.

3. Proposing a novel optimality assessment and non-optimum cause diagnosis framework for non-Gaussian multimode processes.

Chapter 2

Mixture Semisupervised Probabilistic Principal Component Regression Model with Missing Inputs

Principal component regression (PCR) has been widely used as a multivariate method for data-based soft sensor design. In order to take advantage of probabilistic features, it has been extended to probabilistic PCR (PPCR). Commonly, industrial processes operate in multiple operating modes. Moreover, in most cases, outputs are measured at a slower rate than inputs, and for each sample of input variable, its corresponding output may not always exist. These two issues have been solved by developing the mixture semi-supervised PPCR (MSPPCR) method. In this chapter, we extend this developed model to the case of simultaneous missing data in both input and output. Missing data in multidimensional input space constitutes a significantly more challenging problem. Missing input data occurs frequently in industrial plants because of sensor failure and other problems. We develop and solve the MSPPCR model by using the expectation-maximization (EM) algorithm to deal with missing inputs, in addition to missing outputs and multi-mode conditions. Finally, we present three case studies to demonstrate its performance.

2.1 Introduction

Improving efficiency of industrial processes while respecting the safety standards is a growing interest. To make a balance between the above mentioned factors, proper control and monitoring strategies should be designed. Applying advanced process control methods requires suitable measuring devices.[1] On-line measurement devices are often unreliable and expensive to maintain. Moreover, the key process variables are usually measured by on-line analyzers or offline sample analysis in laboratory, and slowly processed measurements of online-analyzer and large delays of laboratory analysis have negative effects on the outcome of applied control techniques.[2]
Over the last two decades, soft sensors have been studied and applied for obtaining key process variables based on developed predictive models.[3] Predictive models of the soft sensors can be based on first principles or based on data. If a model based on first principles can accurately predict a process, a first principles model-based soft sensor can be designed; however a detailed first principles based soft sensor model is generally computationally heavy for real-time analysis.[4] Most of the soft sensors developed so far employ data driven strategies and are designed by extracting information hidden in the historical datasets.[5] Among the data-driven modeling methods for soft sensor design, Principal Component Analysis (PCA) or Principal Component Regression (PCR)[6 7 8] , Partial Least Squares (PLS)[9 10] , Artificial Neural Networks (ANN)[11 12] and Support Vector Machine (SVM)[13] are the most popular ones.

PCA, which is based on dimensionality reduction by introducing latent variables, has been known as one of the most popular methods in soft sensor design. Since PCA is a deterministic method, it has been extended to Probabilistic PCA (PPCA) so that probabilistic inference can be conducted.[14] PPCA model has many important benefits including ease in statistical testing, extending to a mixture of sub-models and dealing with missing data points and so on.[15]

Feasibility to develop a mixture PPCA model is an important property since the single PPCA model only performs well on linear unimodal processes whereas in real applications, the operating systems are generaly nonlinear. In this case, mixture

PPCA model can be applied to estimate the nonlinearity by considering the combination of several linear sub-models. Furthermore, many industrial plants have more than one operating mode, hence, a single PPCA model is not capable of giving an accurate estimate of the process. To address these drawbacks, mixture PPCA model has been introduced in a number of studies[15 16 17] .

The other issue in the design of soft sensors is missing data points. Commonly, both input and output data points are required to design a soft sensor. However, in many cases, due to the sensor failure or delays in measuring some variables, not all input and output data points are available. In[17] labeled and unlabeled datasets are introduced. The part of data which containts output measurements is named as the labeled dataset, and the rest of the missing outputs are named as the unlabeled dataset. The authors in[17] have worked extensively on the mixture semi-supervised Probabilistic Principal Component Regression (MSPPCR) method for soft sensor design. In their method, the information of both labeled and unlabeled datasets are incorporated into the model design. However, their work still does not consider missing data points in input variables. The extension of the methodology from missing output only to simultaneously considering missing input and missing output is non-trivial as evident in the following derivations. Part of the reason for the complexity is: when only the missing output is considered, the data can be simply classified into supervised (available output) and unsupervised (unavailable output). However, when there are missing data in the input and since input is multidimensional, some dimensions have missing data while other dimensions do not have missing data at any sample. So the data cannot be simply classified as missing or not. The main contribution of this work is to derive a MSPPCR model that can deal with both missing input and missing output data.

In data analysis involving missing data, there exists two general approaches to proceed. One way is to discard the missing data points but doing so will result in loss of information. The second way is to predict the missing values. The second way, which is also called imputation, includes case-wise deletion, mean substitution, the last observation carried forward (LOCF) method, regression imputation, Expectation

Maximization (EM)-based algorithm and so on[18 19] .

In this chapter, we will develop the MSPPCR method based on the EM algorithm in the case of multi-mode operation, and missing values in both inputs and outputs of the historical dataset. In addition to multi-mode and semi-supervised cases that have been studied in literature so far, the developed method is also applicable when some input variables have missing data completely at random. The significance of this chapter is to make the most use of all available measurements for building a model. EM algorithm is selected since it can provide a maximum likelihood solution and allows us to iteratively update the estimated values for the missing data based on updated parameters of the model.[20]

The rest of this chapter is organized as follows. Section 1 provides an overview of the fundamentals of PCR, Probabilistic PCR (PPCR), and MSPPCR methods. In section 2, the developed MSPPCR is presented. A numerical simulation example, a classic multimode problem, the Tennessee Eastman (TE) process, and an industrial application are studied in section 3. Conclusions are presented in section 4.

2.2 Preliminaries

2.2.1 PCR

PCR involves two stages. In the first stage, principal components (PCs) of the input (X), called latent variables (T), are extracted using the PCA method. Then these latent variables are utilized in the regression equation.

Let $X \in R^{n\times m}$ and $Y \in R^{n\times r}$ be the input and output datasets, respectively, where n is the number of samples, m is the number of input variables, and r is the number of output variables.

The PCR equations in the multivariate regression problem are:[21]

$$X = TP^T + E \tag{2.1}$$

$$Y = TC^T + F \tag{2.2}$$

where $T \in R^{n\times q}$is the matrix of principal components, q is the number of selected principal components, $P \in R^{m\times q}$ is the loading matrix, $C \in R^{r\times q}$ is the regression matrix between Y and T, and $E \in R^{n\times m}$ and $F \in R^{n\times r}$ are residuals for the PCA step and the regression step, respectively.

2.2.2 PPCR

Let X and Y be the input and output datasets with the same properties as mentioned above for the PCR model. We can present the PPCR model based on the following generative model:

$$x = Pt + e \tag{2.3}$$

$$y = Ct + f \tag{2.4}$$

where $x \in R^{m\times 1}$and $y \in R^{r\times 1}$are one data sample of X and Y respectively, $P \in R^{m\times q}$and $C \in R^{r\times q}$ are weighting matrices, $t \in R^{q\times 1}$is a vector of hidden variables, and $e \in R^{m\times 1}$and $f \in R^{r\times 1}$ are measurement noises of input and output variables.
In PPCR, it is assumed that the latent variables are independent and identically distributed (iid) with Gaussian distribution of $t \sim \mathcal{N}(0, I)$, where I is the identity matrix. Moreover, Gaussian distributions of $e \sim \mathcal{N}(0, \sigma_x^2 I)$ and $f \sim \mathcal{N}(0, \sigma_y^2 I)$ are considered for the measurement noises of input and output variables with the noise variance as σ_x^2 and σ_y^2 , respectively. One can estimate the best model parameters $\{P, C, \sigma_x^2, \sigma_y^2\}$ by maximizing the following likelihood function:

$$L(X, Y) = \ln p(X, Y|P, C, \sigma_x^2, \sigma_y^2) = \ln \prod_{i=1}^{n} p(x_i, y_i|P, C, \sigma_x^2, \sigma_y^2) \tag{2.5}$$

To formulate the marginal probability $p(x, y)$, one should integrate out the latent variable (t) as follows:

$$p(x_i, y_i|P, C, \sigma_x^2, \sigma_y^2) = \int p(x_i|t_i, P, \sigma_x^2)p(y_i|t_i, C, \sigma_y^2)p(t_i)\ dt_i \tag{2.6}$$

To find the optimal solution for the likelihood function, one can use the EM algorithm. The results have been presented in.[22]

2.2.3 MSPPCR

In a MSPPCR, a total of K individual semi-supervised PCR models are incorporated. In each sub-model represented by k, the number of selected latent variables is q. Supposing that the sizes of the labeled and unlabeled datasets are n_1 and n_2 respectively, the MSPPCR model can be expressed as:

$$x_{i,k} = P_k t_{i,k} + e_{i,k} + \mu_{x,k}, k = 1, 2, ..., K \tag{2.7}$$

$$y_{j,k} = C_k t_{j,k} + f_{j,k} + \mu_{y,k}, k = 1, 2, ..., K \tag{2.8}$$

$$x_i = \begin{cases} \sum_{k=1}^{K} p_1(k) x_{i,k}, & \text{if } 1 \leq i \leq n_1 \\ \sum_{k=1}^{K} p_2(k) x_{i,k}, & \text{if } n_1 + 1 \leq i \leq n \end{cases}$$

$$y_j = \sum_{k=1}^{K} p_1(k) y_{j,k}$$

where $i = 1, 2, \cdots, n, j = 1, 2, \cdots, n_1$. $\mu_{x,k}$ and $\mu_{y,k}$ are the mean values of the input and output datasets in k^{th} sub-model. $p_1(k)$ and $p_2(k)$ are the mixing proportions of k^{th} sub-model for the labeled and unlabeled datasets, respectively with the constraint of $\sum_{k=1}^{K} p_1(k) = 1$ and $\sum_{k=1}^{K} p_2(k) = 1$. $P_k \in R^{m \times q}$ and $C_k \in R^{r \times q}$ are weighting matrices of the k^{th} sub-model, $t_k \in R^{q \times 1}$is a vector of latent variables, and $e_k \in R^{m \times 1}$ and $f_k \in R^{r \times 1}$are measurement noises of the input and output variables in the k^{th} sub-model. Similar to the single PPCR model, Gaussian distributions are assumed for the hidden variables and measurement noises in each sub-model. Therefore, $t_k \sim \mathcal{N}(0, I)$, $e_k \sim \mathcal{N}(0, \sigma_{x,k}^2 I)$ and $f_k \sim \mathcal{N}(0, \sigma_{y,k}^2 I)$.
One can find the optimal values of the model parameters $\Theta = \{\theta\}_k = \{P_k, C_k, \sigma_{x,k}^2, \sigma_{y,k}^2$ $, \mu_{x,k}, \mu_{y,k}\}$by maximizing the following likelihood function.

$$\begin{aligned} L(X, Y|\Theta) &= \ln p(X, Y|\Theta) = \ln[p(X_1, Y|\Theta) p(X_2|\Theta)] \\ &= \ln p(X_1, Y|\Theta) + \ln p(X_2|\Theta) \end{aligned} \tag{2.9}$$

where $X_1 \in R^{n_1 \times m}$is the labeled dataset with the corresponding output of $Y \in R^{n_1 \times r}$, and $X_2 \in R^{n_2 \times m}$is the unlabeled dataset where its output is not available. The complete explanation of the algorithm and the results of MSPPCR are available in.[17]

2.3 Development of MSPPCR with missing input data

2.3.1 Model formulation

For the MSPPCR to be developed, we assume that a total of K modes exist, and in each mode, q latent variables are considered. Similar to MSPPCR,[17] n_1 out of the n samples in the dataset are labeled and the remaining n_2 of them are unlabeled. The generative model of the developed MSPPCR is the same as the one stated in Equations 2.7 and 2.8. Since in this model we are considering the case that some input variables have their values missing completely at random (MCAR),[23] x can be expressed as $x^T = [x_o^T, x_m^T]$, where x_o and x_m are the sub-vectors of variables with observed and missing data, respectively.[24]

To illustrate, let us consider an example of a dataset consisting of 4 data points including 1 output variable and 3 input variables, respectively. Some input and output variables have missing values, and the data pattern may be shown as:

$$X = \begin{pmatrix} x_{11} & x_{12} & - \\ - & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \\ x_{41} & - & - \end{pmatrix}, Y = \begin{pmatrix} y_1 \\ - \\ y_3 \\ - \end{pmatrix}$$

where "$-$" indicates missing values. Since the rearranged form of input is $x^T = [x_o^T, x_m^T]$, each input data point x_i, $i = 1, ..., 4$ should be arranged as:

$$x_1 = \begin{pmatrix} x_{11} \\ x_{12} \\ - \end{pmatrix}, x_2 = \begin{pmatrix} x_{22} \\ x_{23} \\ - \end{pmatrix}, x_3 = \begin{pmatrix} x_{31} \\ x_{32} \\ x_{33} \end{pmatrix}, x_4 = \begin{pmatrix} x_{41} \\ - \\ - \end{pmatrix}$$

Thus,

$$x_{1,o} = \begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix}, x_{2,o} = \begin{pmatrix} x_{22} \\ x_{23} \end{pmatrix}, x_{3,o} = \begin{pmatrix} x_{31} \\ x_{32} \\ x_{33} \end{pmatrix}, x_{4,o} = \begin{pmatrix} x_{41} \end{pmatrix}.$$

Now return to our derivations. Since it is assumed that the two noise models are independent of each other, we can have the feature of conditional independence, i.e. given the latent variables, all inputs and outputs are conditionally independent of each other.[25] The likelihood of observations (x_o, y) for labeled datasets and (x_o) for unlabeled datasets for each individual mode is obtained as follows:

$$p(x_o, y|P_{k,o}, C_k, \sigma^2_{x,k,o}, \sigma^2_{y,k}) = \int p(x_o|t_k, P_{k,o}, \sigma^2_{x,k,o})p(y|t_k, C_k, \sigma^2_{y,k})p(t_k)dt_k \tag{2.10}$$

$$p(x_o|P_{k,o}, \sigma^2_{x,k,o}) = \int p(x_o|t_k, P_{k,o}, \sigma^2_{x,k,o})p(t_k)dt_k \tag{2.11}$$

As defined earlier, indices o and m stand for observed and missing parts of the variables, respectively. For example $P_{k,o}$,$\sigma^2_{x,k,o}$ are the weighting matrix and noise variance in the k^{th} sub-model corresponding to the observed inputs, respectively.

Since we have assumed that some input variables are missing completely at random, we have not considered the missing variables in the likelihood function. In other words, those missing variables could have been simply removed. Although this is a simple approach and can be solved by MSPPCR, it does not make the best use of available data. As a result, we resort to the EM algorithm for maximum likelihood estimation since it can handle the missing data points.[20] In the EM algorithm, we maximize the expectation of the complete data log-likelihood instead of only maximizing the likelihood of observed data. This algorithm consists of two steps, expectation and maximization, which are iterated until convergence. In the expectation step, we find the expected value of the complete data log-likelihood given the old parameters. In the maximization step, we maximize the derived expected value with respect to the model parameters $\Theta = \{\theta\}_k = \{P_k, C_k, \sigma^2_{x,k}, \sigma^2_{y,k}, \mu_{x,k}, \mu_{y,k}\}$ and update their values from the previous iteration.

Let $X_1 = [x_1, x_2, ..., x_{n_1}]^T \in R^{n_1 \times m}$, $Y = [y_1, y_2, ..., y_{n_1}]^T \in R^{n_1 \times r}$be the labeled dataset, $X_2 = [x_{n_1+1}, x_{n_1+2}, ..., x_n]^T \in R^{n_2 \times m}$ be the unlabeled dataset, $X = [x_1, x_2, ..., x_n]^T \in R^{n \times q}$ be complete input data and $T = [t_1, t_2, ..., t_n]^T \in R^{n \times q}$ be latent variables values. Note that the complete input data means assembly of all input variables, i.e. they contain input variables with and without missing data. The complete log data likelihood function can be expressed as:

$$\begin{aligned} L(X, Y, T, k|\Theta) =& L(X_1, Y, T, k|\Theta) + L(X_2, T, k|\Theta) \\ =& \ln p(X_1, Y, T, k|\Theta) + \ln p(X_2, T, k|\Theta) \end{aligned} \tag{2.12}$$

To employ the EM algorithm, we consider T, k, and X as hidden variables. Although X consists of observed and missing inputs, we treat entire X as hidden. The treatment of X derived below will automatically assign observed X as deterministic one with zero variance so that they are equal to the measured values as will be seen shortly. As a result, the Q function of the EM agorithm, which is the expected value of the complete data log-likelihood with respect to the joint distribution of hidden/missing variables k, T and X given Y, X_o, and Θ_{old}, can be expressed as:

$$\begin{aligned}
Q =& E_{X,T,k|Y,X_o,\Theta_{old}}[\ln[p(X,Y,T,k|\Theta)]] \\
=& E_{X,T,k|Y,X_o,\Theta_{old}}[\ln[\prod_{i=1}^{n_1} p(x_i,y_i,t_i,k|\Theta)]] + E_{X,T,k|X_o,\Theta_{old}}[\ln[\prod_{i=n_1+1}^{n} p(x_i,t_i,k|\Theta)]] \\
=& \sum_{i=1}^{n_1}\sum_{k=1}^{K} p(k|x_{i,o},y_i,\Theta_{old}) \int\int \ln[p(x_{i,k},y_i,t_{i,k}|k,\Theta)p_1(k)]p(x_{i,k},t_{i,k}|y_i,x_{i,o},k,\Theta_{old})dx_{i,k}dt_{i,k} \\
+& \sum_{i=n_1+1}^{n}\sum_{k=1}^{K} p(k|x_{i,o},\Theta_{old}) \int\int \ln[p(x_{i,k},t_{i,k}|k,\Theta)p_2(k)]p(x_{i,k},t_{i,k}|x_{i,o},k,\Theta_{old})dx_{i,k}dt_{i,k} \\
=& \sum_{i=1}^{n_1}\sum_{k=1}^{K} p(k|x_{i,o},y_i,\Theta_{old})\Big\{\ln p_1(k) + E_{t_{i,k}|y_i,x_{i,o},k,\Theta_{old}} E_{x_{i,k}|y_i,x_{i,o},t_{i,k},k,\Theta_{old}}[\ln p(x_{i,k},y_i,t_{i,k}|k,\Theta)]\Big\} \\
+& \sum_{i=n_1+1}^{n}\sum_{k=1}^{K} p(k|x_{i,o},\Theta_{old})\Big\{\ln p_2(k) + E_{t_{i,k}|x_{i,o},k,\Theta_{old}} E_{x_{i,k}|x_{i,o},t_{i,k},k,\Theta_{old}}[\ln p(x_{i,k},t_{i,k}|k,\Theta)]\Big\}
\end{aligned} \tag{2.13}$$

Note that the indices i, k, m, o stand for the i^{th} data point, k^{th} sub-model, missing part and observed part, respectively. For example, in the following formulae the variable with the index of *i,k,m* represents input variables that have missing data of i^{th} data point from k^{th} sub-model.

In the expectation step, based on the parameters estimated in the previous maximization step, we need to determine the posterior probabilities of $p(k|x_{i,o},y_i,\Theta_{old})$, $p(t_i|y_i,x_{i,o},k,\Theta_{old})$ and $p(x_{i,k}|y_i,x_{i,o},t_{i,k},k,\Theta_{old})$ for the labeled dataset and $p(k|x_{i,o},\Theta_{old})$, $p(t_i|x_{i,o},k,\Theta_{old})$ and $p(x_{i,k}|x_{i,o},t_{i,k},k,\Theta_{old})$ for the unlabeled dataset. They can be derived based on the Bayes' rule as follows:

For labeled dataset:

$$p(k|x_{i,o}, y_i, \Theta_{old}) = \frac{p(x_{i,o}, y_i|k, \Theta_{old}) p_1(k|\Theta_{old})}{p(x_{i,o}, y_i|\Theta_{old})} \tag{2.14}$$

For unlabeled dataset:

$$p(k|x_{i,o}, \Theta_{old}) = \frac{p(x_{i,o}|k, \Theta_{old}) p_2(k|\Theta_{old})}{p(x_{i,o}|\Theta_{old})} \tag{2.15}$$

where $x_{i,o}|k, \Theta_{old} \sim \mathcal{N}(\mu_{x,k,o}, P_{k,o}P_{k,o}^T + \sigma_{x,k,o}^2 I)$ and $y_i|k, \Theta_{old} \sim \mathcal{N}(\mu_{y,k}, C_k C_k^T + \sigma_{y,k}^2 I)$, so the distribution of $x_{i,o}, y_i|k, \Theta_{old}$ can be derived as:

$$x_{i,o}, y_i|k, \Theta_{old} \sim \mathcal{N}\left(\begin{pmatrix} \mu_{x,k,o} \\ \mu_{y,k} \end{pmatrix}, \begin{pmatrix} P_{k,o} \\ C_k \end{pmatrix} \begin{pmatrix} P_{k,o} \\ C_k \end{pmatrix}^T + \begin{pmatrix} \sigma_{x,k,o}^2 I_o & 0_{o,m} \\ 0_{m,o} & \sigma_{y,k}^2 I_m \end{pmatrix} \right)$$

$p_1(k|\Theta_{old})$ and $p_2(k|\Theta_{old})$ are the mixing proportions of each sub-model for the labeled and unlabeled datasets, respectively. The constraints of $\sum_{k=1}^{K} p_1(k|\Theta_{old}) = 1$ and $\sum_{k=1}^{K} p_2(k|\Theta_{old}) = 1$ should be respected and the parameters are computed in the maximization step. In addition, the denominators need not to be calculated since they are normalizing constants.

For labeled dataset:

$$p(t_i|x_{i,o}, y_i, k, \Theta_{old}) = \frac{p(x_{i,o}|t_i, k, \Theta_{old}) p(y_i|t_i, k, \Theta_{old}) p(t_i|k, \Theta_{old})}{p(x_{i,o}, y_i|k, \Theta_{old})} \tag{2.16}$$

For unlabeled dataset:

$$p(t_i|x_{i,o}, k, \Theta_{old}) = \frac{p(x_{i,o}|t_i, k, \Theta_{old}) p(t_i|k, \Theta_{old})}{p(x_{i,o}|k, \Theta_{old})} \tag{2.17}$$

where, $x_{i,o}|t_i, k, \Theta_{old} \sim \mathcal{N}(P_{k,o}t_{i,k} + \mu_{x,k,o}, \sigma_{x,k,o}^2 I)$, $y_i|t_i, k, \Theta_{old} \sim \mathcal{N}(C_k t_{i,k} + \mu_{y,k}, \sigma_{y,k}^2 I)$ and $t_i|k, \Theta_{old} \sim \mathcal{N}(0, I)$; Therefore $p(t_i|x_{i,o}, y_i, k, \Theta_{old})$ and $p(t_i|x_{i,o}, k, \Theta_{old})$ are distributed as Gaussian with means and variances as follows:

For labeled dataset:

$$\begin{aligned} E(t_{i,k}|x_{i,o}, y_i, k, \Theta_{old}) = {} & (\sigma_{x,k,o}^{-2} P_{k,o}^T P_{k,o} + \sigma_{y,k}^{-2} C_k^T C_{k,} + I)^{-1} \\ & \{\sigma_{x,k,o}^{-2} P_{k,o}^T (x_{i,o} - \mu_{x,k,o}) + \sigma_{y,k}^{-2} C_k (y_i - \mu_{y,k})\} \end{aligned} \tag{2.18}$$

$$\begin{aligned} E(t_{i,k}t_{i,k}^T|x_{i,o}, y_i, k, \Theta_{old}) = (\sigma_{x,k,o}^{-2}P_{k,o}^T P_{k,o} + \sigma_{y,k}^{-2}C_k^T C_{k,} + I)^{-1} \\ + E(t_{i,k}|x_{i,o}, y_i, k, \Theta_{old})E^T(t_{i,k}|x_{i,o}, y_i, k, \Theta_{old}) \end{aligned} \tag{2.19}$$

where $i = 1, 2 \cdots, n_1$, and for unlabeled dataset:

$$E(t_{i,k}|x_{i,o}, k, \Theta_{old}) = (P_{k,o}^T P_{k,o} + \sigma_{x,k,o}^2 I)^{-1} P_{k,o}^T (x_{i,o} - \mu_{x,k,o}) \tag{2.20}$$

$$\begin{aligned} E(t_{i,k}t_{i,k}^T|x_{i,o}, k, \Theta_{old}) = \sigma_{x,k,o}^2 (P_{k,o}^T P_{k,o} + \sigma_{x,k,o}^2 I)^{-1} \\ +E(t_{i,k}|x_{i,o}, k, \Theta_{old})E^T(t_{i,k}|x_{i,o}, k, \Theta_{old}) \end{aligned} \tag{2.21}$$

where $i = 1, 2, \cdots, n_2$.

For labeled dataset:

$$p(x_i|y_i, x_{i,o}, t_i, k, \Theta_{old}) = \frac{p(t_i|x_i, y_i, x_{i,o}, k, \Theta_{old})p(x_i|y_i, x_{i,o}, k, \Theta_{old})}{p(t_i|y_i, x_{i,o}, k, \Theta_{old})} \tag{2.22}$$

where $i = 1, 2, \cdots, n_1$, and for unlabeled dataset:

$$p(x_i|x_{i,o}, t_i, k, \Theta_{old}) = \frac{p(t_i|x_i, x_{i,o}, k, \Theta_{old})p(x_i|x_{i,o}, k, \Theta_{old})}{p(t_i|x_{i,o}, k, \Theta_{old})} \tag{2.23}$$

where $i = 1, 2, \cdots, n_2$. Since $x_i^T = [x_{i,o}^T, x_{i,m}^T]$:

$$\begin{aligned} p(t_i|x_i, y_i, x_{i,o}, k, \Theta_{old}) &= p(t_i|x_{i,o}, x_{i,m}, y_i, x_{i,o}, k, \Theta_{old}) \\ &= p(t_i|x_{i,o}, x_{i,m}, y_i, k, \Theta_{old}) \end{aligned} \tag{2.24}$$

$$\begin{aligned} p(t_i|x_i, x_{i,o}, k, \Theta_{old}) &= p(t_i|x_{i,o}, x_{i,o}, x_{i,m}, k, \Theta_{old}) \\ &= p(t_i|x_{i,o}, x_{i,m}, k, \Theta_{old}) \end{aligned} \tag{2.25}$$

Since $x_{i,m}$ is not available, to make the problem tractable the following approximation may be applied:

$$p(t_i|x_{i,o}, x_{i,m}, y_i, k, \Theta_{old}) \approx p(t_i|x_{i,o}, y_i, k, \Theta_{old}) \tag{2.26}$$

$$p(t_i|x_{i,o}, x_{i,m}, k, \Theta_{old}) \approx p(t_i|x_{i,o}, k, \Theta_{old}) \tag{2.27}$$

By substituting Equations 2.26 and 2.27 in Equations 2.22 and 2.23 respectively, we obtain the following approximations:

$$p(x_i|y_i, x_{i,o}, t_i, k, \Theta_{old}) \approx p(x_i|y_i, x_{i,o}, k, \Theta_{old}) \tag{2.28}$$

$$p(x_i|x_{i,o},t_i,k,\Theta_{old}) \approx p(x_i|x_{i,o},k,\Theta_{old}) \tag{2.29}$$

To find the distributions of $p(x_i|x_{i,o},y_i,k,\Theta_{old})$ and $p(x_i|x_{i,o},k,\Theta_{old})$, first we need to find $p(x_{i,m}|y_i,x_{i,o},k,\Theta_{old})$ and $p(x_{i,m}|x_{i,o},k,\Theta_{old})$, i.e. the posterior distribution of the missing input variables. The means and variances of these probability distributions can be formulated by using the generative model of MSPPCR which was stated earlier. Equation 2.7 is partitioned into missing and observed parts, hence it can be expressed as:

$$\begin{pmatrix} x_{i,k,o} \\ x_{i,k,m} \end{pmatrix} = \begin{pmatrix} P_{k,o} \\ P_{k,m} \end{pmatrix} t_{i,k} + e_{i,k} + \begin{pmatrix} \mu_{x,k,o} \\ \mu_{x,k,m} \end{pmatrix}, k = 1,2,...,K \tag{2.30}$$

The equation for the missing data can be written as:

$$x_{i,k,m} = P_{k,m}t_{i,k} + e_{i,k} + \mu_{x,k,m}, k = 1,2,...,K \tag{2.31}$$

Therefore, the sufficient statistics of $p(x_{i,m}|y_i,x_{i,o},k,\Theta_{old})$ and $p(x_{i,m}|x_{i,o},k,\Theta_{old})$ distributions are found by obtaining the expected value and covariance of both sides of Equation 2.31 as follows:

For labeled datasets:

$$E(x_{i,k,m}|x_{i,o},y_i,k,\Theta_{old}) = P_{k,m}E(t_{i,k}|x_{i,o},y_i,k,\Theta_{old}) + \mu_{x,k,m} \tag{2.32}$$

By substituting the value of $E(t_{i,k}|x_{i,o},y_i,k,\Theta_{old})$ using Equation 2.18, we have:

$$\begin{aligned} E(x_{i,k,m}|x_{i,o},y_i,k,\Theta_{old}) = P_{k,m}[(&\sigma_{x,k,o}^{-2}P_{k,o}^TP_{k,o} + \sigma_{y,k}^{-2}C_k^TC_{k,} + I)^{-1} \\ &\{\sigma_{x,k,o}^{-2}P_{k,o}^T(x_{i,o} - \mu_{x,k,o}) + \sigma_{y,k}^{-2}C_k(y_i - \mu_{y,k})\}] + \mu_{x,k,m} \end{aligned} \tag{2.33}$$

Similarly,

$$\begin{aligned} &E(x_{i,k,m}x_{i,k,m}^T|x_{i,o},y_i,k,\Theta_{old}) = P_{k,m}[E(t_{i,k}t_{i,k}^T|x_{i,o},y_i,k,\Theta_{old}) \\ &- E(t_{i,k}|x_{i,o},y_i,k,\Theta_{old})E^T(t_{i,k}|x_{i,o},y_i,k,\Theta_{old})]P_{k,m}^T + \sigma_{x,k,m}^2 \\ &+ E(x_{i,k,m}|x_{i,o},y_i,k,\Theta_{old})E^T(x_{i,k,m}|x_{i,o},y_i,k,\Theta_{old}) \end{aligned} \tag{2.34}$$

$E(t_{i,k}|x_{i,o},y_i,k,\Theta_{old})$ and $E(t_{i,k}t_{i,k}^T|x_{i,o},y_i,k,\Theta_{old})$ are replaced by Equations 2.18 and 2.20, respectively. This results in:

$$\begin{aligned} E(x_{i,k,m}x_{i,k,m}^T|x_{i,o},y_i,k,\Theta_{old}) =& P_{k,m}(\sigma_{x,k,o}^{-2}P_{k,o}^TP_{k,o} + \sigma_{y,k}^{-2}C_k^TC_{k,} + I)^{-1}P_{k,m}^T + \sigma_{x,k,m}^2 \\ &+ E(x_{i,k,m}|x_{i,o},y_i,k,\Theta_{old})E^T(x_{i,k,m}|x_{i,o},y_i,k,\Theta_{old}) \end{aligned} \tag{2.35}$$

where $i = 1, 2, \cdots, n_1$, and for unlabeled dataset:

$$E(x_{i,k,m}|x_{i,o}, k, \Theta_{old}) = P_{k,m}E(t_{i,k}|x_{i,o}, k, \Theta_{old}) + \mu_{x,k,m} \tag{2.36}$$

$E(t_{i,k}|x_{i,o}, k, \Theta_{old})$ is substituted using Equation 2.20, and the result is:

$$E(x_{i,k,m}|x_{i,o}, k, \Theta_{old}) = P_{k,m}[(P_{k,o}^T P_{k,o} + \sigma_{x,k,o}^2 I)^{-1} P_{k,o}^T (x_{i,o} - \mu_{x,k,o})] + \mu_{x,k,m} \tag{2.37}$$

Similarly,

$$\begin{aligned} E(x_{i,k,m}x_{i,k,m}^T|x_{i,o}, k, \Theta_{old}) =& P_{k,m}[E(t_{i,k}t_{i,k}^T|x_{i,o}, k, \Theta_{old}) - E(t_{i,k}|x_{i,o}, k, \Theta_{old}) \\ & E^T(t_{i,k}|x_{i,o}, k, \Theta_{old})^T]P_{k,m} + \sigma_{x,k,m}^2 \\ & + E(x_{i,k,m}|x_{i,o}, y_i, k, \Theta_{old})E^T(x_{i,k,m}|x_{i,o}, y_i, k, \Theta_{old}) \end{aligned} \tag{2.38}$$

The values of $E(t_{i,k}|x_{i,o}, k, \Theta_{old})$ and $E(t_{i,k}t_{i,k}^T|x_{i,o}, k, \Theta_{old})$ are found using Equations 2.20 and 2.22, respectively, and are then substituted in Equation 2.38. As a result, we have:

$$\begin{aligned} E(x_{i,k,m}x_{i,k,m}^T|x_{i,o}, k, \Theta_{old}) =& P_{k,m}[\sigma_{x,k,o}^2(P_{k,o}^T P_{k,o} + \sigma_{x,k,o}^2 I)^{-1}]P_{k,m}^T + \sigma_{x,k,m}^2 \\ & + E(x_{i,k,m}|x_{i,o}, k, \Theta_{old})E^T(x_{i,k,m}|x_{i,o}, k, \Theta_{old}) \end{aligned} \tag{2.39}$$

where $i = 1, 2, \cdots, n_2$. Therefore, the sufficient statistics of the distribution of $p(x_i|y_i, x_{i,o}, k, \Theta_{old})$ and $p(x_i|x_{i,o}, k, \Theta_{old})$ when combining distributions of both missing and observed variables are derived by following a similar rationale as in:[24]

For labeled dataset:

$$E(x_{i,k}|x_{i,o}, y_i, k, \Theta_{old}) = \begin{pmatrix} x_{i,o} \\ E(x_{i,k,m}|x_{i,o}, y_i, k, \Theta_{old}) \end{pmatrix} \tag{2.40}$$

$$cov(x_{i,k}, x_{i,k}|x_{i,o}, y_i, k, \Theta_{old}) = \begin{pmatrix} 0 & 0 \\ 0 & cov(x_{i,k,m}, x_{i,k,m}|x_{i,o}, y_i, k, \Theta_{old}) \end{pmatrix} \tag{2.41}$$

where $i = 1, 2, \cdots, n_1$, and for unlabeled dataset:

$$E(x_{i,k}|x_{i,o}, k, \Theta_{old}) = \begin{pmatrix} x_{i,o} \\ E(x_{i,k,m}|x_{i,o}, k, \Theta_{old}) \end{pmatrix} \tag{2.42}$$

$$cov(x_{i,k}, x_{i,k}|x_{i,o}, k, \Theta_{old}) = \begin{pmatrix} 0 & 0 \\ 0 & cov(x_{i,k,m}, x_{i,k,m}|x_{i,o}, k, \Theta_{old}) \end{pmatrix} \tag{2.43}$$

where $i = 1, 2, \cdots, n_2$.

In the maximization step, we maximize the expected value of the complete data log-likelihood with respect to the model parameters. Followings are the maximization results:

$$p_1(k) = \frac{1}{n_1}\sum_{i=1}^{n_1} p(k|x_{i,o}, y_i, \Theta_{old}) \tag{2.44}$$

$$p_2(k) = \frac{1}{n_2}\sum_{i=n_1+1}^{n} p(k|x_{i,o}, \Theta_{old}) \tag{2.45}$$

$$p(k) = \frac{1}{n}\{\sum_{i=1}^{n_1} p(k|x_{i,o}, y_i, \Theta_{old}) + \sum_{i=n_1+1}^{n} p(k|x_{i,o}, \Theta_{old})\} \tag{2.46}$$

The procedure of deriving Equations 2.44 to 2.46 is similar to that in[17] and hence the details are omitted. Only the final results are given below:

$$\begin{aligned} &\frac{\partial E[L(X,Y,T,k|\Theta)]}{\partial P_k} = 0 \Longrightarrow \\ P_k^{new} =&\Big[\sum_{i=1}^{n_1}[p(k|x_{i,o}, y_i, \Theta_{old})(E(x_{i,k}|x_{i,o}, y_i, k, \Theta_{old}) - \mu_{x,k})E^T(t_{i,k}|x_{i,o}, y_i, k, \Theta_{old})] \\ &+ \sum_{i=n_1+1}^{n}[p(k|x_{i,o}, \Theta_{old})(E(x_{i,k}|x_{i,o}, k, \Theta_{old}) - \mu_{x,k})E^T(t_{i,k}|x_{i,o}, k, \Theta_{old})]\Big] \\ &\times\Big[\sum_{i=1}^{n_1}[p(k|x_{i,o}, y_i, \Theta_{old})E(t_{i,k}t_{i,k}^T|x_{i,o}, y_i, k, \Theta_{old})] \\ &+ \sum_{i=n_1}^{n}[p(k|x_{i,o}, \Theta_{old})E(t_{i,k}t_{i,k}^T|x_{i,o}, k, \Theta_{old})]\Big]^{-1} \end{aligned} \tag{2.47}$$

$$\begin{aligned} &\frac{\partial E[L(X,Y,T,k|\Theta)]}{\partial C_k} = 0 \Longrightarrow \\ C_k^{new} =&\Big[\sum_{i=1}^{n_1}[p(k|x_{i,o}, y_i, \Theta_{old})(y_i - \mu_{y,k})E^T(t_{i,k}|x_{i,o}, y_i, k, \Theta_{old})\Big] \\ &\times\Big[\sum_{i=1}^{n_1}[p(k|x_{i,o}, y_i, \Theta_{old})E(t_{i,k}t_{i,k}^T|x_{i,o}, y_i, k, \Theta_{old})]\Big]^{-1} \end{aligned} \tag{2.48}$$

$$
\begin{aligned}
&\frac{\partial E[L(X,Y,T,k|\Theta)]}{\partial \mu_{x,k}} = 0 \Longrightarrow \\
\mu_{x,k}^{new} =& \Big[\sum_{i=1}^{n_1} p(k|x_{i,o},y_i,\Theta_{old})[E(x_{i,k}|x_{i,o},y_i,k,\Theta_{old}) - P_k E(t_{i,k}|x_{i,o},y_i,k,\Theta_{old})] \\
&+ \sum_{i=n_1+1}^{n} p(k|x_{i,o},\Theta_{old})[E(x_{i,k}|x_{i,o},k,\Theta_{old}) - P_k E^T(t_{i,k}|x_{i,o},k,\Theta_{old})]\Big] \\
&/\Big[\sum_{i=1}^{n_1} p(k|x_{i,o},y_i,\Theta_{old}) + \sum_{i=n_1+1}^{n} p(k|x_{i,o},\Theta_{old})\Big]
\end{aligned}
\tag{2.49}
$$

$$
\begin{aligned}
&\frac{\partial E[L(X,Y,T,k|\Theta)]}{\partial \mu_{y,k}} = 0 \Longrightarrow \\
\mu_{y,k}^{new} =& \Big[\sum_{i=1}^{n_1} p(k|x_{i,o},y_i,\Theta_{old})[y_i - C_k E(t_{i,k}|x_{i,o},y_i,k,\Theta_{old})]\Big] / \Big[\sum_{i=1}^{n_1} p(k|x_{i,o},y_i,\Theta_{old})\Big]
\end{aligned}
\tag{2.50}
$$

$$
\begin{aligned}
&\frac{\partial E[L(X,Y,T,k|\Theta)]}{\partial \sigma_{x,k}^{2new}} = 0 \Longrightarrow \\
\sigma_{x,k}^{2new} =& \Big\{\sum_{i=1}^{n_1} p(k|x_{i,o},y_i,\Theta_{old})\Big[(E(x_{i,k}|x_{i,o},y_i,k,\Theta_{old}) - \mu_{x,k})^T \\
&(E(x_{i,k}|x_{i,o},y_i,k,\Theta_{old}) - \mu_{x,k}) - 2E^T(t_{i,k}|x_{i,o},y_i,k,\Theta_{old}) \\
&P_k^{newT}(E(x_{i,k}|x_{i,o},y_i,k,\Theta_{old}) - \mu_{x,k}) \\
&+trace[P_k^{newT}P_k^{new}E(t_{i,k}t_{i,k}^T|x_{i,o},y_i,k,\Theta_{old})]+ \\
&+trace[cov(x_{i,k},x_{i,k}|x_{i,o},y_i,k,\Theta_{old})]\Big] \\
&+\sum_{i=n_1+1}^{n} p(k|x_{i,o},\Theta_{old})\Big[(E(x_{i,k}|x_{i,o},k,\Theta_{old}) - \mu_{x,k})^T \\
&(E(x_{i,k}|x_{i,o},k,\Theta_{old}) - \mu_{x,k}) - 2E^T(t_{i,k}|x_{i,o},k,\Theta_{old}) \\
&P_k^{newT}(E(x_{i,k}|x_{i,o},k,\Theta_{old}) - \mu_{x,k}) \\
&+trace[P_k^{newT}P_k^{new}E(t_{i,k}t_{i,k}^T|x_{i,o},k,\Theta_{old})] \\
&+trace[cov(x_{i,k},x_{i,k}|x_{i,o},k,\Theta_{old})]\Big]\Big\}/ \\
&\Big\{m[\sum_{i=1}^{n_1} p(k|x_{i,o},y_i,\Theta_{old}) + \sum_{i=n_1+1}^{n} p(k|x_{i,o},\Theta_{old})]\Big\}
\end{aligned}
\tag{2.51}
$$

$$
\begin{aligned}
&\frac{\partial E[L(X,Y,T,k|\Theta)]}{\partial \sigma_{y,k}^{2new}} = 0 \Longrightarrow \\
\sigma_{y,k}^{2new} =& \Big\{ \sum_{i=1}^{n_1} p(k|x_{i,o}, y_i, \Theta_{old}) \Big[(y_i - \mu_{y,k})^T (y_i - \mu_{y,k}) - 2E^T(t_{i,k}|x_{i,o}, y_i, k, \Theta_{old}) \\
& C_k^{newT}(y_i - \mu_{y,k}) + trace[C_k^{newT} C_k^{new} E(t_{i,k}t_{i,k}^T|x_{i,o}, y_i, k, \Theta_{old})] \Big] \Big\} / \\
& \Big\{ r(\sum_{i=1}^{n_1} p(k|x_{i,o}, y_i, \Theta_{old})) \Big\}
\end{aligned}
\tag{2.52}
$$

We iterate the equations over the expectation and maximization steps until the parameters converge to their optimal values.

2.3.2 On-line predictions

Soft sensor can be constructed based on the developed MSPPCR model. Let us assume that x_{new} is the new data point in online application. At first, we find the posterior probability of each operating mode as follows:

$$
p(k|x_{new}, \Theta) = \frac{p(x_{new}|k, \Theta) p(k|\Theta)}{p(x_{new}|\Theta)} \tag{2.53}
$$

Then we compute the estimated latent variable $\hat{\mathrm{t}}_{k,new}$ corresponding to each operating mode as:

$$
\hat{t}_{k,new} = (P_k^T P_k + \sigma_{x,k}^2 I)^{-1} P_k^T (x_{new} - \mu_{x,k}) \tag{2.54}
$$

The predicted output corresponding to each mode is:

$$
\hat{y}_{k,new} = C_k \hat{t}_{k,new} = C_k (P_k^T P_k + \sigma_{x,k}^2 I)^{-1} P_k^T (x_{new} - \mu_{x,k}) + \mu_{y,k} \tag{2.55}
$$

The final predicted value of the output computed by weighting over all K modes is:

$$
\hat{y}_{new} = \sum_{k=1}^{K} p(k|x_{new}, \Theta) \hat{y}_{k,new} \tag{2.56}
$$

Finally, by comparing with the real outputs y, one can evaluate the performance of the soft sensor using root mean squared error (RMSE), R squared test, etc.

2.4 Case studies

In this section, we will demonstrate the validity of our developed model. In the first part, the developed algorithm is illustrated by a numerical example. In the second part, the method is evaluated through the TE simulation process.

2.4.1 Numerical example

A numerical dataset is simulated using the following model:

$$\begin{aligned} x_k &= P_k t_k + e_k \\ y_k &= C_k t_k + f_k \end{aligned} \tag{2.57}$$

Considering a three operating mode problem, the values of k are either 1,2 or 3. There are six input and one output variables in each operating mode. The simulated number of latent variables is three. P_k and C_k are weighting matrices of dimensions 6×3 and 1×3, respectively that are selected randomly. t_k is the latent variable vector in each operating mode and follows a Gaussian distribution of $\mathcal{N}(0, I)$. e_k and f_k are input and output measurement noises in each mode, and also follow Gaussian distributions with zero mean and variance of 0.01^2 , i.e. $\sigma^2_{x,k} = 0.01^2$ and $\sigma^2_{y,k} = 0.01^2$

In each operating mode, 1500 data samples are generated, where 1000 data points are from the training set and the remaining 500 are from the validation set. To simulate a multi-rate problem, 90 % of the output data points are removed. Hence, 90 percent of the dataset is unlabeled and 10 percent is labeled. Moreover, to evaluate the performance of the developed MSPPCR, 10 percent of the input variables are also removed randomly (MCAR). Note that missing points are randomly selected and can be from any of the six variables.

After applying the proposed MSPPCR, the estimated priors of each operating mode, 0.3291, 0.3334, and 0.3374, are very close to the real value of 1/3 for each. To evaluate the proposed model, its performance is compared with another method in which MSPPCR is applied after the missing variables have been simply replaced by their mean values. Hereafter, we call it the mean replacement method for simplicity. By applying the mean replacement method, priors of each operating mode are estimated as 0.6202, 0.1853 and 0.1945 which are significantly different from the real values. Therefore, the proposed method can identify the corresponding modes much more accurately than the mean replacement method.

To evaluate the model performance, its prediction accuracy is tested on the validation dataset. Using the proposed method the R^2 of the prediction is 0.9954 and its RMSE is 0.0937. On the other hand, using the mean replacement method, R^2 is 0.8301 and RMSE is 0.5699. The trends of predicted and real values for the two methods are shown in Figures 2.1-2.4 for comparison.

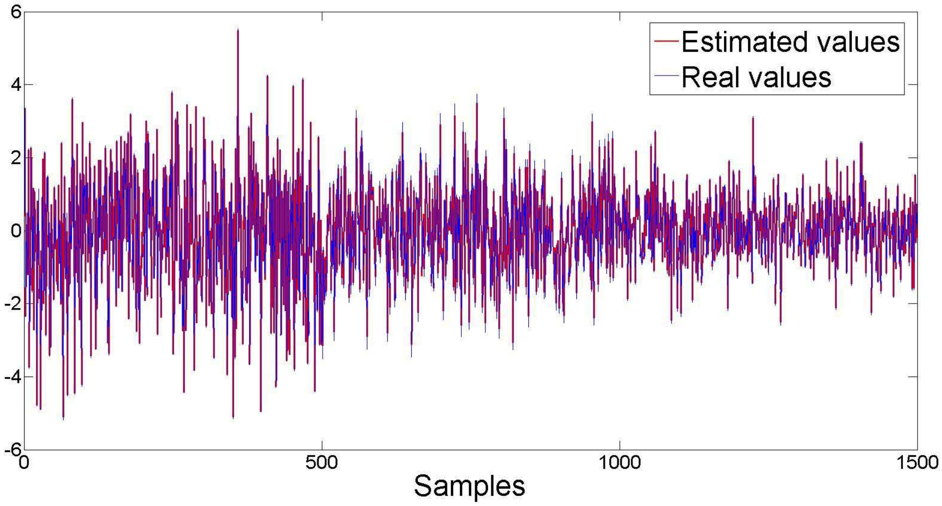

Figure 2.1: *Comparison of predicted and real values using the proposed method*

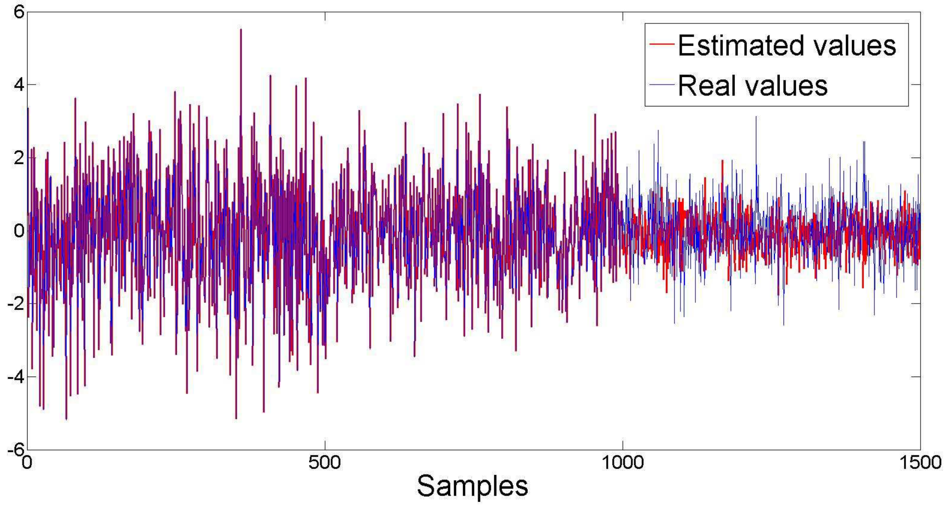

Figure 2.2: *Comparison of predicted and real values using the mean replacement method*

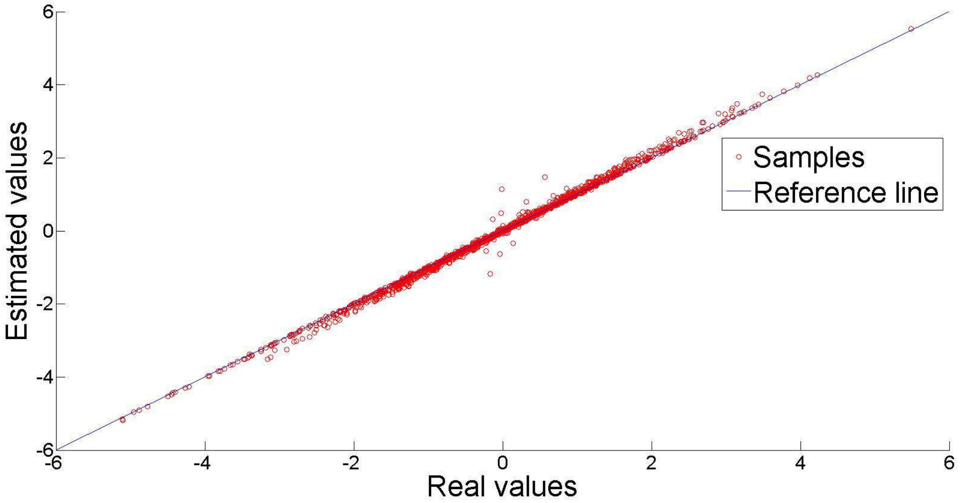

Figure 2.3: *Comparison of predicted and real values using the proposed method*

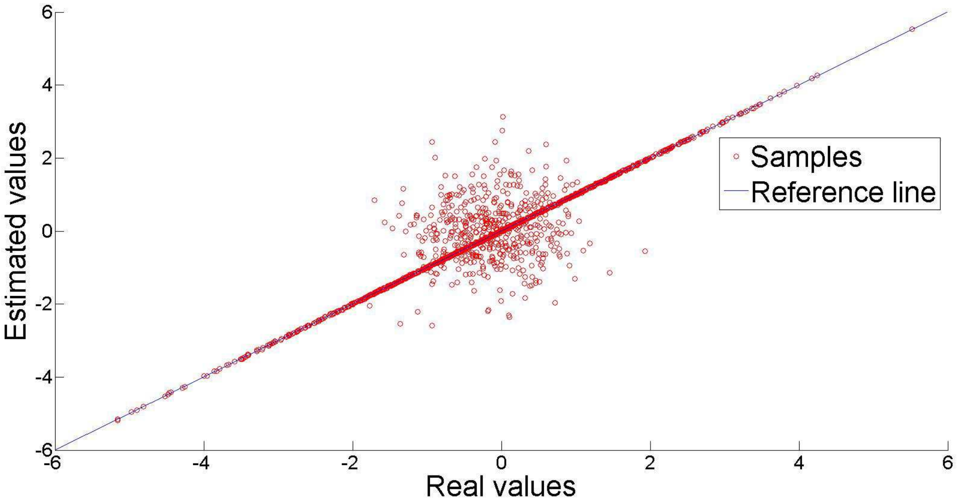

Figure 2.4: *Comparison of predicted and real values using the mean replacement method*

The performance of the two mentioned methods is summarized in Table 2.1. It can be seen that our proposed method has a better performance compared to the mean replacement method in all aspects. This includes better detection of modes, higher prediction performance and lower prediction error.

Table 2.1: *Comparison of the performance of two methods*

Method	Estimated priors	R-squared	RMSE
Developed MSPCR	[0.3291,0.3334,0.3374]	0.9954	0.0937
Mean Replacement	[0.6202,0.1853,0.1945]	0.8301	0.5699

2.4.2 Tennessee Eastman benchmark process

TE process benchmark has been widely used for the testing of various methods in process control, monitoring, optimization, etc. The model was first developed by[26] based on the industrial process of TE chemical company. The process includes five main units: a reactor, a product condenser, a vapor-liquid separator, a recycle compressor and a product stripper. The purpose of this process is to produce two main products, G and H, from four reactants A, C, D and E. Besides, F, a by-product, might be produced under non-ideal situations. Note that reactants and products are in gaseous and liquid phases, respectively, and all reactions are exothermic and irreversible. The process has 12 manipulated variables and 41 measured variables out of which 22 process variables are measured continuously, and 19 components variables have slow rate measurement. There are six operation modes based on the three different G/H mass ratios.[26] The schematic of the process is shown in Figure 2.5.

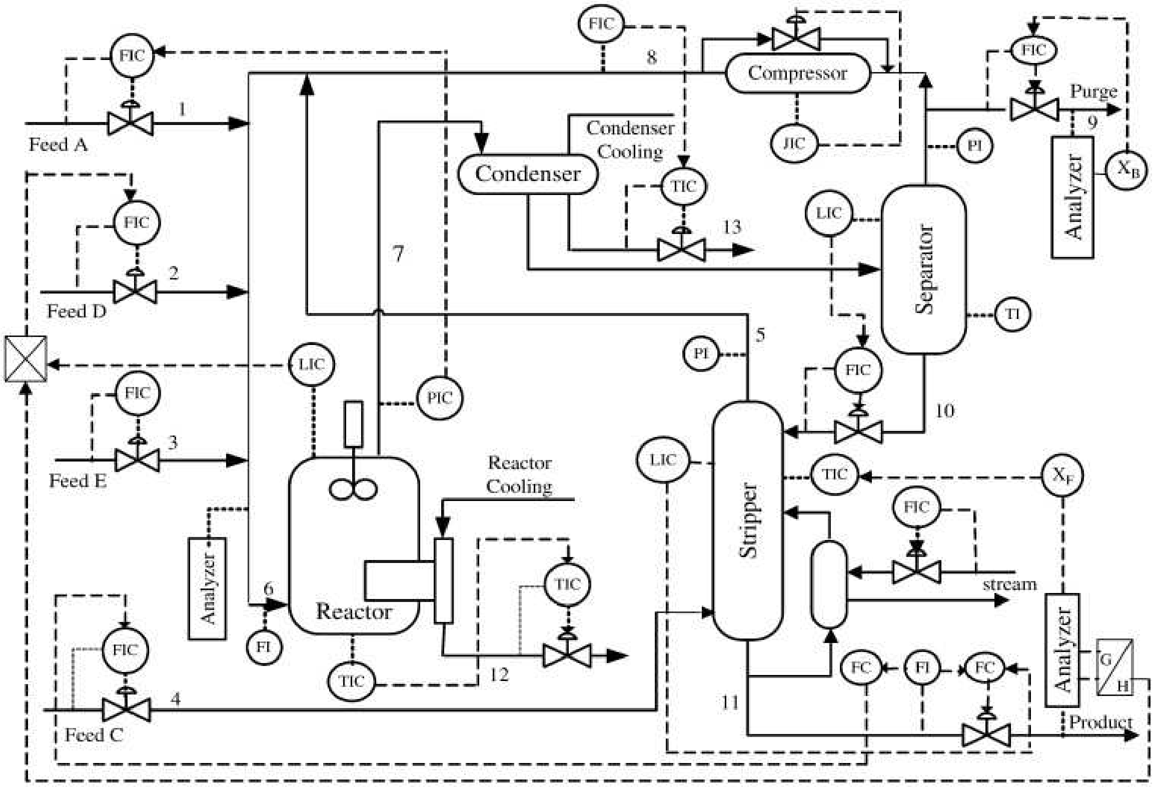

Figure 2.5: *The schematic diagram of Tennessee Eastman process*[22]

Since the open loop process is unstable, we have applied the decentralized control strategy developed by.[27]

To evaluate the performance of the developed model as a soft sensor, we have simulated three different operating points in mode 1 of the TE process in which G/H mass ratio is 50/50.[26] This leads to three different modes with the same production demand caused by changing the set points of some process variables. The selected set points are shown in Table 2.2.

The aim of our soft sensor design is to predict the percentage of product components F, G and H in purge gas based on the other 22 commonly measured variables. We have simulated each operating mode for 37.5 hours, and the sampling period is 0.05 hours. Therefore, we have collected 750 data points in each operating mode and selected 500 for training and the remaining 250 data points for validation. Finally, we have 1500 data points of all operating modes for training and 750 for validation. Moreover, since the output variables are assumed to be slow rate measurements, 90 percent of them are removed. To evaluate the performance of the proposed method in dealing with missing data points, 10 percent of the input variables are also removed with the MCAR mechanism.

Table 2.2: *Properties of stable modes*

Stable mode	1	2	3 (°C)
Reactor pressure (kpa)	2800	2700	2750
Seperator level set point (%)	50	70	60
Reactor temperature (°C)	122.9	130	135

Based on the above mentioned data-set, the developed MSPPCR model is built. The estimated prior probabilities of operating modes, 0.3333, 0.3339, and 0.3327, are very close to the real prior probabilities of 1/3 for each mode. The posterior probabilities of each mode for data points are given in Figure 2.6. The results show that the developed model detects the operating modes accurately despite having missing input and output data.

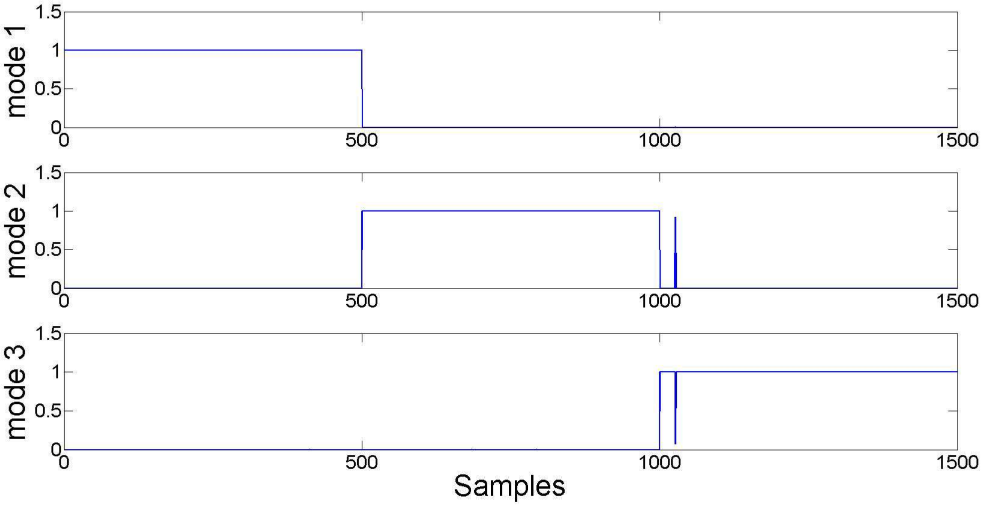

Figure 2.6: *Posterior probabilities of each mode estimated by the developed MSPCR*

The validation dataset is used to test the model prediction accuracy. The comparison plots of real and predicted values are given in Figure 2.7 . The results of the developed soft sensor for this case study are summarized in Table 2.3. Based on the simulation results, one can see that the proposed model has acieved high accuracy in predicting quality variables, i.e. for all three outputs it shows high values for R-squared tests and low values for RMSE in the validation dataset. In addition, it has correctly detected the operating modes to which each data point belongs. Therefore, in addition to dealing with missing outputs, this method can also handle data with missing input variables.

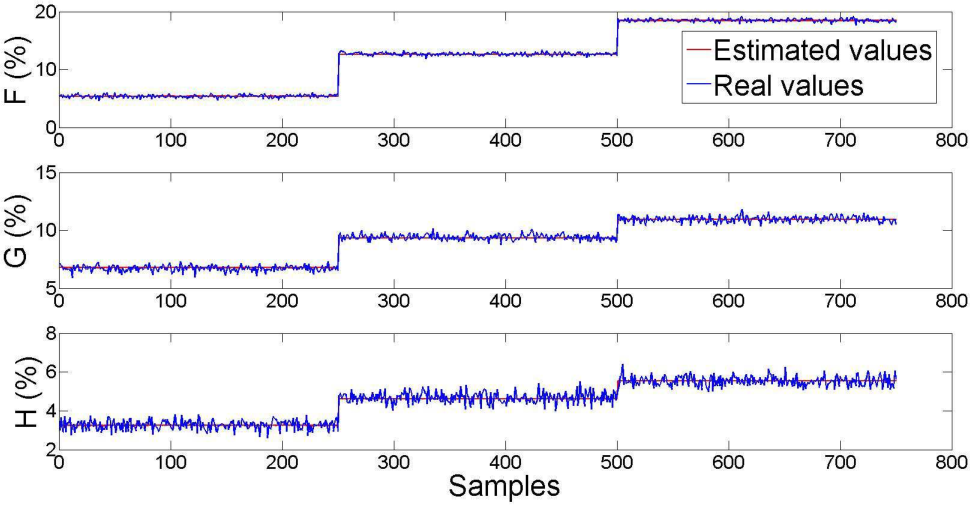

Figure 2.7: *The comparison plots of real and estimated values*

Table 2.3: *Performance of the designed soft sensor*

Purge component	R-squared	RMSE
F	0.9979	0.2489
G	0.9793	0.2552
H	0.9272	0.2621

2.4.3 Industrial Application

Oil is known as an essential raw material for the organic chemistry and its mixture composition varies depending on the location it is produced. Its main components are Hydrocarbons, Sulphur compounds, Nitrogen compunds and Oxygenates. There are many methods such as hydrotreating to remove acidic compounds and impurities in order to improve properties of the streams. Hydrotreaters are the well known units in petroleum refineries where reactions convert organic nitrogen and sulfur into NH_3 and H_2S as well as producing light hydrocarbons.

One of the most common applications of hydrotreating is Naphtha Hydrotreaters. In Naphtha Hydrotreaters Olefinic and Diolefinic feedstock compounds are saturated and the Sulfur and Nitrogen content is reduced in order to improve the quality of Naphtha product.[28] The schematic of the process is shown in Figure 2.8.

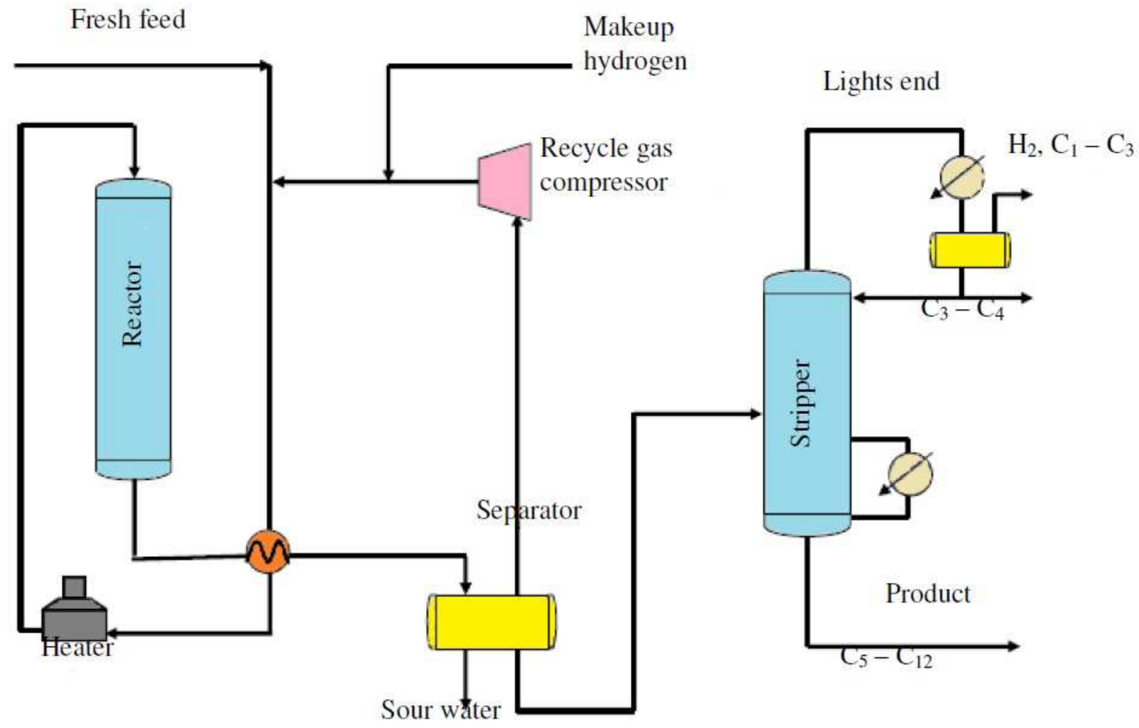

Figure 2.8: *The schematic diagram of Naphtha Hydrotreater*[28]

The objective is to build a model to continuously predict the sulfur content of the product that is not frequently available and the available data is not reliable due to uncertainties. The data set consists of a total of 10197 data samples including 222 input and 1 output variables with the average sampling period of 30 minutes. The dataset is partitioned into two parts of training (7397 samples) and validation (2800 samples). 14 of the variables that are highly correlated with the output are selected as regressors. To construct the model, number of principal components and modes are selected as 11 and 2, respectively. In offline training, corresponding outputs of 1432 samples ($\approx$ 19%) are not available. To evaluate the performance of the proposed method in dealing with missing data points, the predictive model is built in the following conditions:

1. Considering the original data set
2. Removing 10 percent of input variables (MCAR) in offline training
3. Removing 10 percent of input variables (MCAR) in online validation

For the above mentioned conditions, MSPCR model is constructed, and in the case of having missing inputs, both the developed method and the mean replacement method are employed. The comparison plots of real and predicted values for each condition are given in Figures 2.9-2.13. Note that output values are normalized. The results of the developed soft sensors are summarized in Table 2.4.

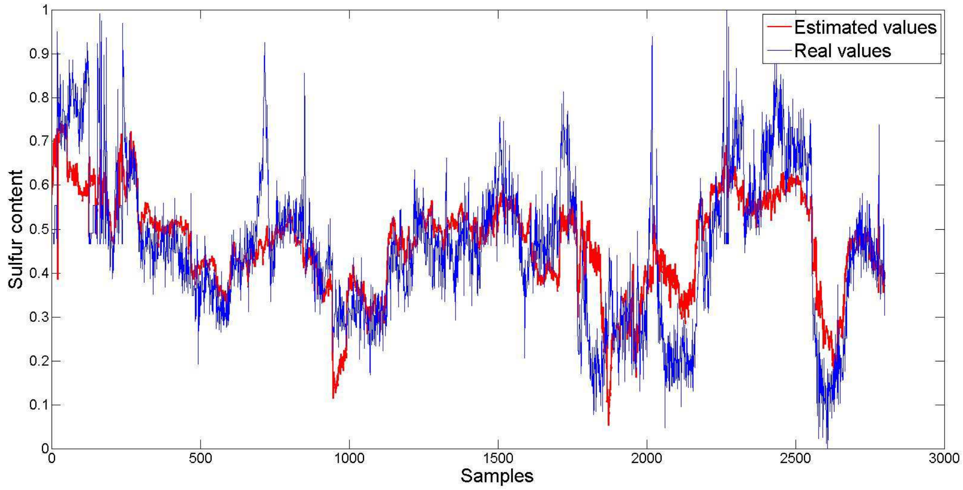

Figure 2.9: *Condition 1: the comparison plots of real and estimated values*

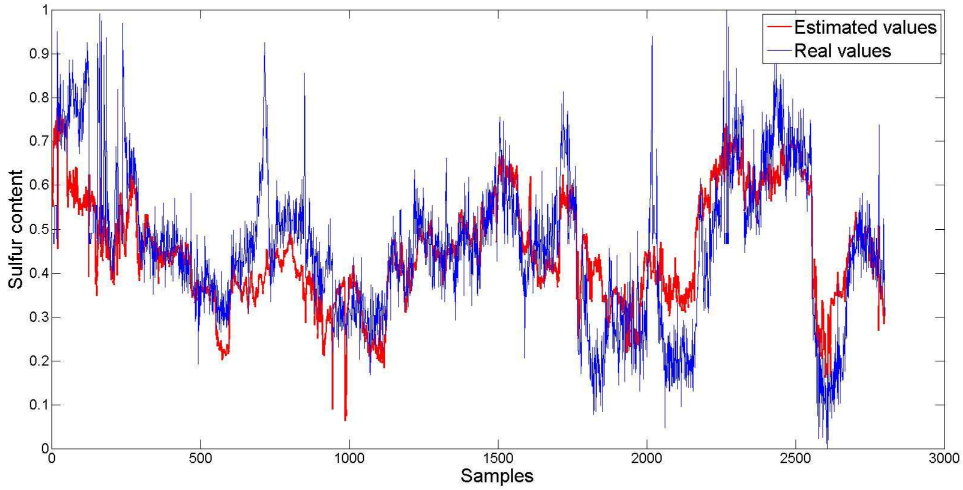

Figure 2.10: *Condition 2: the comparison plots of real and estimated values (the developed MSPCR method)*

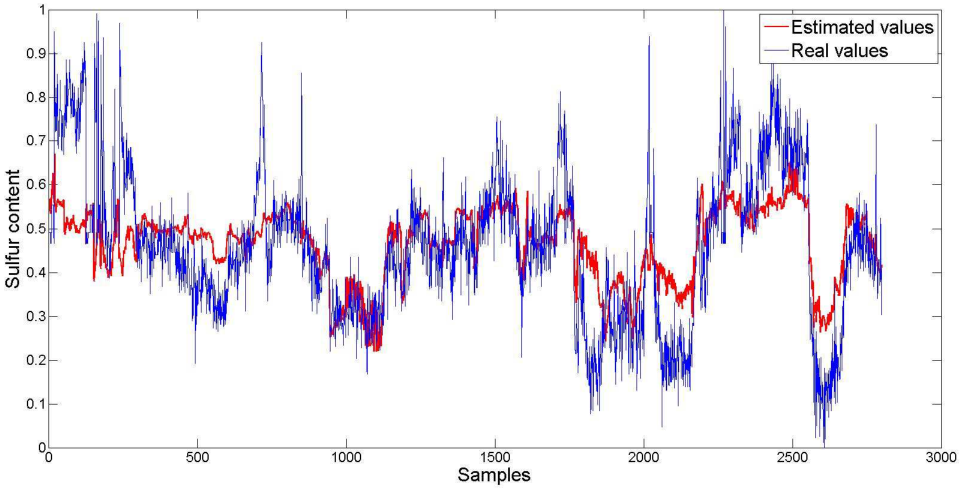

Figure 2.11: *Condition 2: the comparison plots of real and estimated values (the mean replacement method)*

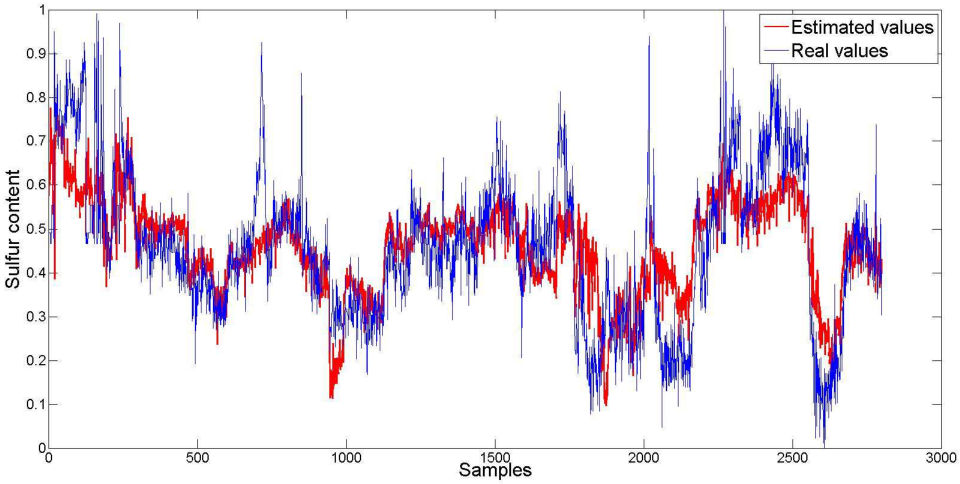

Figure 2.12: *Condition 3: the comparison plots of real and estimated values (the developed MSPCR method)*

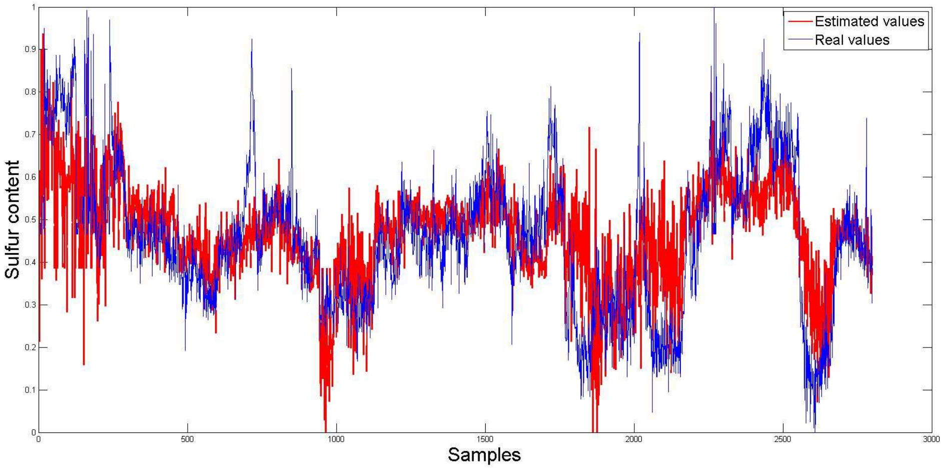

Figure 2.13: *Condition 3: the comparison plots of real and estimated values (the mean replacement method)*

Table 2.4: *Performance of the designed soft sensor*

Condition	method	R-squared	RMSE
1	MSPCR	0.5411	0.2989
2	Developed MSPCR	0.5009	0.3117
2	Mean Replacement	0.4452	0.3286
3	Developed MSPCR	0.5204	0.3055
3	Mean Replacement	0.3974	0.3425

Based on the results, the developed method has achieved higher accuracy, i.e. higher values for R-squared test and lower values for RMSE in estimating sulfur content in comparison with the mean replacement method for both conditions 2 and 3. In addition, the performance of the soft sensor employing the proposed method has slightly changed compared to the condition 1 that there are not any missing inputs. As a result, the proposed method can deal with missing inputs in addition to missing outputs.

2.5 Conclusion

In this chapter, we have extended the existing MSPPCR model to the most general solution including multi-mode, missing output and missing input problems. Since in reality, we regularly have missing inputs in addition to missing outputs, by using the EM algorithm, we have extended the MSPPCR model to deal with missing data in both inputs and outputs. Therefore, this work makes the general PPCA methodology more applicable for solving real industrial problems. Compared to the traditional methods of missing data treatments such as using the mean value of variables, this method has a better performance in mode detection and quality variable prediction. We have presented three case studies, and all have confirmed the improved accuracy of our developed model.

Chapter 3

Unsupervised Hierarchical Mixtures of Probabilistic Principal Component Regression

PPCR is a probabilistic counterpart of PCR which is based on dimensionality reduction. In order to deal with nonlinearities as well as multi-mode behavior, it has been extended to mixture PPCR (MPPCR). To build a model for a multi-mode system, the associated problem with MPPCR is to estimate the number of mixture components. In this chapter, we propose a hierarchical MPPCR approach for automatically estimating the number of components. This method is based on a divisive hierarchical algorithm, and initially starts with the minimum possible number of components. At each stage, the decision for splitting the components is made based on the Minimum Message Length (MML) criterion. In addition, a merging step is proposed for detected highly overlapped components that controls the splitting step performance. Furthermore, the developed hierarchical MPPCR model is solved through maximum a posteriori (MAP) principle under the EM algorithm in order to utilize prior distributions. Finally, two case studies are presented to demonstrate the model performance.

3.1 Introduction

Principal Component Analysis (PCA) is one of the most popular dimensionality reduction methods which employs feature transformation techniques. It has been extended to probabilistic PCA (PPCA) in order to benefit from probabilistic features.[14] PPCA has many important advantages including feasibility to construct a mixture of sub PPCA models.[15] Extending to a mixture of sub models enables this method to be applied on nonlinear systems (considering several linear sub-models) as well as multi-mode systems[16] as opposed to the single PPCA model, which has a good performance only on single unimodal processes.

To estimate the parameters of the mixture PPCA (MPPCA) model, the EM algorithm is usually employed to maximize the likelihood function. However, this algorithm suffers from an initialization problem. In other words, its performance is highly sensitive to initial parameters, and it is probable to converge to a local maximum. Initialization techniques have been investigated in a number of studies[29 30] .In general, initialization methods can be classifed as deterministic or stochastic.[31] In deterministic methods, the initial values are specified by employing primary clustering algorithms such as hierarchical clustering[32 33] . On the other hand, in stochastic methods, different starting values are tried, and the solution with the highest likelihood value is selected.[34] Since there is no single method with the best performance for all applications, a proper initialization technique is selected based on the problem considered. For example, emEM and RndEM, as stochastic methods, have better performance for overlapped mixture components, while hierarchical and K-means clustering methods are preferred for well-separated components[31].

The other issue associated with MPPCA is selecting the appropriate number of mixture components. The problem is similar to the usual trade-off for model order selection. In other words, selecting too many components may overfit the training dataset and cause poor performance in prediction, while selecting too few components may not provide a good estimation of the true model.[35] A common procedure to address this problem is to estimate the parameters for a set of model candidates, and then choose the best model based on the model selection criterion.[36]

Figueiredo and Jain[35] have worked extensively on unsupervised learning of finite mixture models. They proposed a one stage algorithm to select the number of components. This algorithm integrates the parameter estimation of the model candidates and model selection in one step. It starts with the largest possible number of components, and then removes the components with the least estimated priors until the number of components reach the minimum possible number of components. For a comparison of different model candidates, the minimum message length (MML) criterion is employed and has shown satisfactory results. Recently, a novel model selection method for MPPCA was proposed by Zhao.[37] In this work, a new hierarchical Bayesian information criterion (BIC) is proposed, and both one stage and two stage algorithms are studied. Although the above mentioned methods have shown good performance in estimating the number of components, a priori knowledge about the possible range of components is required. In other words, by selecting a wide range for the possible number of components, these methods will be extensively time consuming.

A hierarchical approach for building MPPCA was first proposed by Bishop and Tipping[38] to aid in visualization of high-dimensional datasets through latent variables. The algorithm starts by building a single PPCA model and then increasing the components in subsequent levels to detect and visualize clusters and subclusters of the system. In this method, the decision on splitting the components and selecting the number of offspring of each cluster is made by the user. As a result, although it is a proper method for visualization, due to its human-driven nature, it may not be suitable for detecting the number of components. In other words, it is time consuming and gives varying results depending on the human decisions.[39] Consequently, the hierarchical MPPCA method was modified by Su and Dy[39] to be utilized in selecting the number of components in MPPCA. The authors proposed an automated hierarchical MPPCA that employs the integrated classification likelihood (ICL) criterion to decide when to split the components, and the procedure is repeated until no components can be split. This method has shown satisfactory results in experiments.

However, there are two issues associated with it. First, due to the clustering application of the ICL criterion, the method has shown a poor performance in detecting overlapped components.[35] Second, due to its hierarchical nature, when a component is split, the next level will be performed on new components, and what is done at the previous level cannot be undone.[40]

In this chapter, we will develop a hierarchical mixture of PPCR (MPPCR) to be employed for constructing predictive models. In this developed model, the above mentioned drawbacks are addressed. First, the minimum message length criterion proposed by Figueiredo and Jain[35] is employed instead of the ICL in order to improve performance in detecting overlapped components. Second, after each splitting step, highly overlapped components are detected, and a merging step is introduced.

The rest of this chapter is organized as follows. Section 3.2 describes the problem. In section 3.3, an overview of the fundamentals of hierarchical MPPCA is provided. In section 3.4, the MML criterion is discussed. Section 3.5 presents the developed hierarchical MPPCR model. A numerical example and an experimental example are provided in section 3.6. Conclusions are presented in section 3.7.

3.2 Problem Statement

Mixture models have shown a great performance in describing undefined distributions, multi-mode systems as well as nonlinear systems by considering several linear sub-models.[29] However, the challenging question is how to find the proper number of mixtures in order to avoid overfitting caused by selecting too many components, as well as weak models caused by selecting too few components. A number of methods have been proposed for selecting the number of components, and these can be classified into deterministic[41 42 43] (such as BIC) and stochastic[44 45] (such as Markov chain Monte Carlo (MCMC)) categories from a computational point of view.[35] In this chapter, we will focus on deterministic methods to find the proper number of components.

The likelihood function is nondecreasing and is a function of the number of components, which suggests that it cannot be the only criterion for estimating the number of components. In other words, increasing the number of components increases the complexity of the model. This results in an increased goodness of fit between the predicted and observed value of the training dataset, thereby causing over-fitting on the training dataset. To overcome this problem, the model selection criterion is defined by the sum of the likelihood term (which defines the fit between the model and training data set) and the penalizing term (which controls model complexity).[46] Consider $C(\hat{\theta}(k), k)$ as a model selection criterion. The proper number of components is estimated by the following optimization problem:[35]

$$\hat{k} = arg \min_{k} C(\hat{\theta}(k), k), k = k_{min}, ..., k_{max} \tag{3.1}$$

where $\hat{\theta}$ represents the estimated parameters of a model with k components. The generic form of the selection criterion is as follows:

$$C(\hat{\theta}(k), k) = -\log p(x|\hat{\theta}(k)) + N(k) \tag{3.2}$$

where x denotes the training dataset, and $N(k)$ is the penalizing term of the k component model that is an increasing function of k. According to Equation 3.2, the likelihood term of the criterion function is a nonincreasing function of k (due to its negative sign), while the penalizing term is increasing by an increment of k. As a result, the minimization of the criterion gives a conservative answer based on the fit and complexity.

3.3 Unsupervised Hierarchical MPPCA

The hierarchical representation of MPPCA can be built based on agglomorative, and also divisive algorithms. In the agglomorative algorithms, the procedure starts with the maximum possible number of components (k_{max}), and at each further step, similar groups are merged. On the other hand, the divisive algorithms start with one component model, and the number of components increases until a stopping criterion is satisfied.[39] Since determining an optimal value for k_{max} is not straight forward, and starting with a large k_{max} is time consuming, a divisive algorithm is investigated in

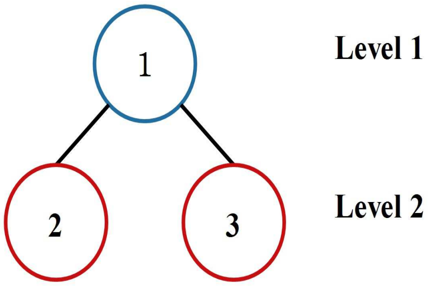

Figure 3.1: *Building hierarchical MPPCA: step 1*

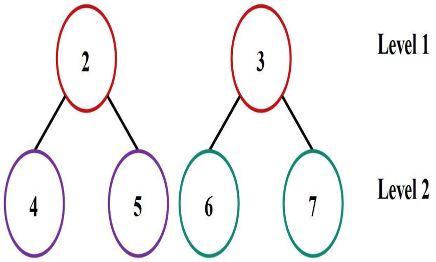

Figure 3.2: *Building hierarchical MPPCA: step 2*

this thesis.

The divisive hierarchical MPPCA starts with a model of single PPCA, and at further levels, splits into its offspring. After each component is split, the selection criterion of each parent is compared with its offspring. The parent is split into its offspring if the selection criterion of the offspring is more optimal than the parent. Otherwise, the parent is retained and is not split in subsequent steps. Consider an illustrative example to clarify the procedure. Generally, each parent can have any number of offspring, but in the following example we have assumed the number of offspring is two:

In the first step, the model is considered as unimodal, and the PPCA parameters are estimated. This model is shown as 1 in Figure 3.1. In the next level, the model is split into two offspring, and the parameters are estimated using the hierarchical MPPCA method that will be discussed in the next sections. Finally, the selection criterion of the parent (1) is compared to its offspring (2,3) in level 2. Here, we assume that the criterion of (2,3) is more optimal than (1), so the parent (1) is split.

As shown in Figure 3.2, since in step 1 the selection criterion of the offspring (2,3) is more optimal than their parent (1), the offspring of the component (1), i.e., (2) and (3) are retained and considered in this step as parents. In this step, (2) and (3) are split into their offspring (4,5) and (6,7), respectively, and the parameters of the offspring components (4,5,6,7) are estimated. Then, the criterion of each parent is compared with its respective offspring, i.e., the criterion of (2) is compared to (4,5), and the criterion of (3) is compared to (6,7). Assume the criterion of (2) is more optimal than (4,5) while criterion of (3) is less optimal than (6,7). This would result in component (2) not getting split while component (3) gets split into (6) and (7).

In step 3, based on the previous levels, components (2), (6) and (7) are preserved as parents. Since component (2) went through the splitting process in step 2, it will not be further split in the next levels. However, (6) and (7) are split into their offspring. Assume that the criterion of both components (6) and (7) is more optimal than (8,9) and (10,11), respectively. This would result in (6) and (7) being the final components. Therefore, three components are estimated, i.e., (2), (6) and (7). The schematic of this step is shown in Figure 3.3.

Figure 3.4 provides an overview of hierarchical MPPCA for the above mentioned example. The algorithm started with one component, and finally three components namely, (2), (6) and (7), are selected.

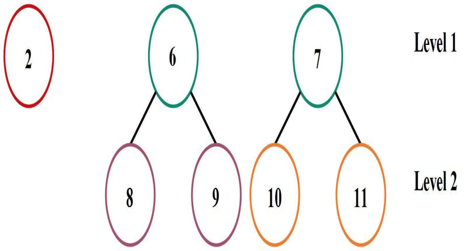

Figure 3.3: *Building hierarchical MPPCA: step 3*

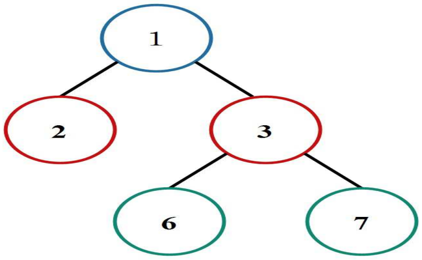

Figure 3.4: *Building hierarchical MPPCA: an overview*

3.3.1 Parameter Estimation: MPPCA

In MPPCA, a total of K PPCA models are incorporated. Consider the number of selected latent variables in each sub-model (represented by k) is q. The MPPCA model can be formulated as:

$$x_{i,k} = P_k t_{i,k} + e_{i,k} + \mu_{x,k} \quad , \; k = 1, 2, ..., K \tag{3.3}$$

where $x_{i,k} \in R^{m\times 1}$ is one data sample of the input dataset, $\mu_{x,k}$ is the mean value of input variables, $P_k \in R^{m\times q}$ is the weighting matrix, $t_{i,k} \in R^{q\times 1}$ is a vector of hidden variables, and $e_{i,k} \in R^{m\times 1}$ is the measurement noise of input variables in the k^{th} sub-model. Gaussian distributions of $e_k \sim \mathcal{N}(0, \sigma^2_{x,k} I)$ and $t_k \sim \mathcal{N}(0, I)$ are considered for the measurement noise of the input and the latent variables, respectively.

One can estimate the best model parameters $\Theta = \{\theta\}_k = \{P_k, \sigma^2_{x,k}, \mu_{x,k}\}$ by maximizing the likelihood function $L(X|\Theta)$. Since direct maximization of the observed likelihood function is difficult, the EM algorithm is employed for maximum likelihood estimation. Tipping and Bishop[15] have described a detailed explanation in their work.

3.3.2 Parameter Estimation: Unsupervised Hierarchical MP-PCA

One can incorporate the MPPCA model into a hierarchical framework.[38] Assume in the previous level K components are detected, so there are K components as parents, and each one (k) is going to be split into g_k offspring. The formulation of hierarchical MPPCA is as follows:

$$x_{i,(k,c)} = P_{(k,c)} t_{i,(k,c)} + e_{i,(k,c)} + \mu_{x,(k,c)} \quad , \; k = 1, 2, ..., K \quad , \; c = 1, 2, ..., g_k \tag{3.4}$$

where (k, c) represents the offspring (c) of parent (k). The probability density function can be formulated as:

$$p(X|\Theta) = \sum_{k=1}^{K} \sum_{c=1}^{g_k} p(k, c) p(X|\theta_{k,c}) \tag{3.5}$$

Since the prior probability of the parents $p(k)$ is fixed from the previous level, the joint probability of k and c, i.e. $p(k, c)$, is separated, and the probability density function can be expressed as:

$$p(X|\Theta) = \sum_{k=1}^{K} p(k) \sum_{c=1}^{g_k} p(c|k) p(X|\theta_{k,c}) \tag{3.6}$$

where $p(c|k)$ represents the mixing coefficients for offspring c of parent k and satisfies the constraint of $\sum_{c=1}^{g_k} p(c|k) = 1$. One should formulate the likelihood function to estimate the parameters in this level. Since the data points, X, and the indicator variables of parents, Z, are observed in this level, the likelihood function can be expressed as:

$$\begin{aligned} L = \ln(p(X, Z|\Theta)) &= \ln\Big(\prod_{i=1}^{n}\prod_{k=1}^{K}(p(x_i|\theta_k)p(k))^{z_{ik}}\Big) \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik}\ln\Big(p(k)\sum_{c=1}^{g_k} p(x_i|\theta_{k,c})p(c|k)\Big) \end{aligned} \tag{3.7}$$

Since the expectation of the indicator variables are estimated in the previous level, the likelihood function can be expressed as:

$$L = \sum_{i=1}^{n}\sum_{k=1}^{K} p(k|x_i,\theta_k)\ln\Big(p(k)\sum_{c=1}^{g_k} p(x_i|\theta_{k,c})p(c|k)\Big) \tag{3.8}$$

where $p(k|x_i, \theta_k)$ is constant. To estimate the maximum likelihood, one can use the EM algorithm. The expected complete-data likelihood can be formulated as:

$$E(L_C) = \sum_{i=1}^{n}\sum_{k=1}^{K} p(k|x_i,\theta_k)\Big\{\ln p(k) + \sum_{c=1}^{g_k} p(c|x_i,\theta_{k,c},k)\ln\{p(c|k)p(x_i|\theta_{k,c})\}\Big\} \tag{3.9}$$

The optimal parameters can be estimated by maximizing Equation 3.9. More details are provided by Bishop and Tipping.[38]

3.4 Minimum Message Length Criterion

Statistical estimation can be considered as a coding process.[47] Researchers have explained the philosophy of minimum encoding methods as follows[47][29] :

“We may first estimate the parameters and then encode the data under the assumption that these are the true values. The encoded string must now, however, contain a specification of the estimated values. Any model is, therefore, only worth considering if the shortening of the encoded data string achieved by adopting it more than compensates for the lengthening caused by having to quote estimated parameter values. We thus naturally arrive at a very simple trade-off between the complexity of a model and its goodness of fit. A more complicated model will usually fit a data-set better than a simpler one, enabling a briefer encoding of the data, but this must be paid for by the cost of the greater number of parameter estimates."

In this approach, the message length has two parts. The first part contains the prior knowledge of the parameters θ, and the second part consists of the observed data X under the assumption that the estimated parameters are the true values. As a result, the message length can be expressed as:[35]

$$Length(\theta, X) = Length(\theta) + Length(X|\theta) \tag{3.10}$$

The expected message length is expressed in[48 35] as follows:

$$E(Length(\theta, X)) = -\ln p(\theta) - \ln p(X|\theta) + \frac{1}{2}\ln|I(\theta)| + \frac{c}{2}(1 + \ln k_c) \tag{3.11}$$

where $p(\theta)$ presents the prior probability of the parameters, $|I(\theta)|$ presents the expected Fisher matrix information, c is the number of unknown parameters, and k_c is the optimal quantizing lattice constant for R^c.

By approximating c as 1, k_c as 1/12, replacing $|I(\theta)|$ by the complete Fisher information matrix, and considering the standard Jeffrey's prior for parameters, Equation 3.11 is reformulated as follows:[35]

$$L(\theta, X) = \frac{N}{2} \sum_{m:\pi_m > 0} \ln(\frac{n\pi_m}{12}) + \frac{k_{nz}}{2} \ln \frac{n}{12} + \frac{k_{nz}(N+1)}{2} - \ln p(X|\theta) \tag{3.12}$$

where π_m denotes the prior probability of each component, N is the number of parameters in each component, n is the total number of data points , and k_{nz} is the number of components with non-zero priors. The MML method has shown good performance as a selection criterion for detecting the number of components in both overlapped and separated mixtures, and it outperforms other selection criteria.[35]

3.5 Unsupervised Hierarchical MPPCR

3.5.1 Derivations of the algorithm

In hierarchcial MPPCR the first step starts with a single PPCR model. The details of the PPCR model are presented in section 2.2.2. In the next steps, each component is split into its offspring. In general, the number of offspring can be arbitrarily selected, however, selecting more than two offspring may result in estimating too many components. As a result, it is assumed that the number of offspring is two.[39]

Let $X = [x_1, x_2, ..., x_n]^T \in R^{n \times m}$ and $Y = [y_1, y_2, ..., y_n] \in R^{n \times r}$ be the input and output datasets, respectively, and $T = [t_1, t_2, ..., t_n]^T \in R^{n \times q}$ be the latent variable values, where n is the number of samples, m is the number of input variables, r is the number of output variables, and q is the number of selected latent variables. Assume K components are detected in the previous level, and in this level each component (k) is going to be split into g_k offspring. Since it is assumed the maximum number of offspring for each parent is two, the maximum of g_k for different k equals two. The hierarchical MPPCR model can be formulated as:

$$x_{i,(k,c)} = P_{(k,c)} t_{i,(k,c)} + e_{i,(k,c)} + \mu_{x,(k,c)} \quad , \ k = 1, 2, ..., K \quad , \ c = 1, 2, ..., g_k \tag{3.13}$$

$$y_{i,(k,c)} = C_{(k,c)} t_{i,(k,c)} + f_{i,(k,c)} + \mu_{y,(k,c)} \quad , \ k = 1, 2, ..., K \quad , \ c = 1, 2, ..., g_k \tag{3.14}$$

where $i = 1, 2,..., n$, (k, c) represents offspring (c) of parent (k), $\mu_{x,(k,c)}$ and $\mu_{y,(k,c)}$ are the mean values of the input and output datasets in each sub model (k, c), respectively. $p(k, c)$ is the mixing proportions of each sub-model offspring (c) and parent (k) with the constraint of $\sum_{k=1}^{K} \sum_{c=1}^{g_k} p(k, c) = 1$, and $\sum_{c=1}^{g_k} p(c|k) = 1$ for each parent $k = 1, 2, ..., K$. $P_{(k,c)} \in R^{m \times q}$ and $C_{(k,c)} \in R^{r \times q}$ are weighting matrices of the (k, c) sub-model, $t_{(k,c)} \in R^{q \times 1}$is a vector of latent variables, and $e_{(k,c)} \in R^{m \times 1}$ and $f_{(k,c)} \in R^{r \times 1}$are measurement noises of the input and output variables in the (k, c) sub-model, respectively. It is assumed that the hidden variables and measurement noises follow the Gaussian distribution of $t_{(k,c)} \sim \mathcal{N}(0, I)$, $e_{(k,c)} \sim \mathcal{N}(0, \sigma^2_{x,(k,c)} I)$ and $f_{(k,c)} \sim \mathcal{N}(0, \sigma^2_{y,(k,c)} I)$ in each submodel (k,c).

Equation 3.12 is equivalent to a posteriori density by considering a Drichlet-type prior:[35]

$$p(\pi_1, ..., \pi_k) \propto exp\{-\frac{N}{2} \sum_{m=1}^{k} log(\pi_m)\} \tag{3.15}$$

where π_m is $p(m)$. To estimate the parameters $\Theta = \{\theta\}_{(k,c)} = \{P_{(k,c)}, C_{(k,c)}, \sigma^2_{x,(k,c)}, \sigma^2_{y,(k,c)}, \mu_{x,(k,c)}, \mu_{y,(k,c)}, \pi_{(k,c)}\}$, the Drichlet prior described in Equation 3.15 for component priors $\pi_{k,c}$, and a flat prior for other parameters are assumed, and the maximum a posteriori (MAP) function is formulated and then maximized by employing the EM algorithm. Note that in the hierarchical formulation $\Theta = \{\Theta_{offspring}, \Theta_{parent}\}$, where Θ_{parent} is fixed from the previous level. Since the observed variables are $\{X, Y, k\}$, the MAP criterion is as follows:

$$\hat{\Theta}_{MAP} = arg\max_{\Theta}\{\ln p(X, Y, k, \Theta)\} = arg\max_{\Theta}\{\ln p(X, Y, k|\Theta) + \ln p(\Theta)\} \quad (3.16)$$

One can maximize the MAP function using the EM algorithm by considering complete data that includes observed and hidden variables. In this problem, the observed variables are $\{X, Y, k\}$, and hidden variables are $\{T, c\}$. Note that the indicators of parent components that equal the posterior distribution of each parent component, i.e., $p(k|x_i, y_i, \Theta_{parent})$ are fixed from the previous level, where Θ_{parent} denotes parameters of the previous level. The three main steps of the EM algorithm, which are initialization, expectation, and maximization, are described in the following subsections:

3.5.1.1 Initialization

Initialization has an important impact on the performance of the EM algorithm, and improper initial values may cause convergence to a suboptimal response. Initializiation techniques have been investigated in a number of studies and are mainly classified into deterministic and stochastic methods[29 30 31] . In deterministic methods, the initial values are mainly selected based on clustering methods. In these methods, the initial values are fixed, and new candidates for initialization are not proposed. In addition, due to their clustering nature, these methods have shown better performance in separated components. On the other hand, in stochastic methods, different starting values are examined, and the candidate with the highest likelihood value is selected.[31] Based on the problem considered, a proper initialization method can be selected.

In the problem considered here, stochastic methods are selected because of their good performance in detecting overlapped components.[31] Among available stochastic initialization techniques, "xem-EM" is selected since it has shown good performance in most cases in a comprehensive study of stochastic methods.[34] In "1 em-EM" method, several short runs of the EM algorithm from the random initial values are conducted. Then the solution of the short run of the EM which maximizes the likelihood function is selected as the initial values for a long run of the EM algorithm. A short run of the EM means that the threshold of the convergence is larger than a long run of the EM, so it stops after a fewer iterations. In "x em-EM" method, "1 em-EM" algorithm is repeated x times, and the solution with the highest likelihood value is selected as the final solution.[34]

3.5.1.2 Expectation

The Q function of the EM algorithm, that is the expected value of the complete data log-aposteriori with respect to the joint distribution of the hidden variables given the observed variables and Θ_{old}, can be written as:

$$
\begin{aligned}
&E_{T,c|X,Y,k,\Theta_{old}}\Big[\ln[p(X,Y,T,k,c,\Theta)]\Big] = E_{T,c|X,Y,k,\Theta_{old}}\Big[\ln[p(X,Y,T,k,c|\Theta)] + \ln[p(\Theta)]\Big] \\
=&E_{T,c|X,Y,k,\Theta_{old}}\Big[\ln[\prod_{i=1}^{n}\prod_{k=1}^{K} p(x_{i,k},y_{i,k},t_{i,k},k,c|\Theta)^{p(k|x_i,y_i,\Theta_{parent})}] + \ln[\prod_{k=1}^{K} p(\theta_k)]\Big] \\
=&\sum_{i=1}^{n}\sum_{k=1}^{K} p(k|x_i,y_i,\Theta_{parent}) E_{t_{i,k},c|x_i,y_i,k,\Theta_{old}}\Big[\ln[p(x_{i,k},y_{i,k},t_{i,k},k,c|\Theta)]\Big] + \sum_{k=1}^{K}\ln[p(\theta_k)] \\
=&\sum_{i=1}^{n}\sum_{k=1}^{K} p(k|x_i,y_i,\Theta_{parent}) \sum_{c=1}^{g_k} p(c|x_i,y_i,k,\Theta_{old}) \int \ln\Big[p(x_{i,k,c},y_{i,k,c},t_{i,k,c}|k,c,\Theta)p(c|k)p(k)\Big] \\
&p(t_{i,k,c}|x_i,y_i,k,c,\Theta_{old})dt_{i,k,c} + \sum_{k=1}^{K}\sum_{c=1}^{g_k}\ln[p(\theta_{(k,c)})] \\
=&\sum_{i=1}^{n}\sum_{k=1}^{K} p(k|x_i,y_i,\Theta_{parent}) \sum_{c=1}^{g_k} p(c|x_i,y_i,k,\Theta_{old}) \Big\{ \ln p(c|k) + \ln p(k) + \\
&\int \ln[p(x_{i,k,c},y_{i,k,c},t_{i,k,c}|k,c,\Theta)] p(t_{i,k,c}|x_i,y_i,k,c,\Theta_{old})dt_{i,k,c}\Big\} + \sum_{k=1}^{K}\sum_{c=1}^{g_k}\ln[p(\theta_{(k,c)})]
\end{aligned}
\tag{3.17}
$$

Note that the indices i, k, c denote the i^{th} data point, k^{th} parent component, and c^{th} offspring component, respectively.

In the expectation step, the posterior probabilities of $p(c|x_i, y_i, k, \Theta_{old})$ and $p(t_i|x_i, y_i, k, c, \Theta_{old})$ are determined based on the parameters estimated in the previous M-step, and can be derived based on the Bayes' rule as follows:

$$p(c|x_i, y_i, k, \Theta_{old}) = \frac{p(x_i, y_i|k, c, \Theta_{old})p(c|k, \Theta_{old})}{p(x_i, y_i|k, \Theta_{old})} \tag{3.18}$$

where $p(c|k, \Theta_{old})$ is the mixing proportions of each offspring component given its parent, and the constraints of $\sum_{c=1}^{g_k} p(c|k, \Theta_{old}) = 1$ have been imposed. The posterior probability of parent (k) and offspring (c) is as follows:

$$p(c, k|x_i, y_i, \Theta_{old}) = p(c|x_i, y_i, k, \Theta_{old})p(k|x_i, y_i, \Theta_{parent}) \tag{3.19}$$

Note that $p(k|x_i, y_i, \Theta_{parent})$ is fixed from the previous level.

$$p(t_i|x_i, y_i, k, c, \Theta_{old}) = \frac{p(x_i|t_i, k, c, \Theta_{old})p(y_i|t_i, k, c, \Theta_{old})p(t_i|k, c, \Theta_{old})}{p(x_i, y_i|k, c, \Theta_{old})} \tag{3.20}$$

where

$$\begin{aligned} x_i|t_i, k, c, \Theta_{old} &\sim \mathcal{N}(P_{k,c}t_{i,k,c} + \mu_{x,k,c}, \sigma^2_{x,k,c}I) \\ y_i|t_i, k, c, \Theta_{old} &\sim \mathcal{N}(C_{k,c}t_{i,k,c} + \mu_{y,k,c}, \sigma^2_{y,k,c}I) \\ t_i|k, c, \Theta_{old} &\sim \mathcal{N}(0, I) \end{aligned}$$

Therefore $t_i|x_i, y_i, k, c, \Theta_{old}$ has a Gaussian distribution with mean and variance as follows:
For labeled datasets:

$$\begin{aligned} E(t_{i,k,c}|x_i, y_i, k, c, \Theta_{old}) = &(\sigma^{-2}_{x,k,c}P^T_{k,c}P_{k,c} + \sigma^{-2}_{y,k,c}C^T_{k,c}C_{k,c} + I)^{-1} \\ &\{\sigma^{-2}_{x,k,c}P^T_{k,c}(x_i - \mu_{x,k,c}) + \sigma^{-2}_{y,k,c}C_{k,c}(y_i - \mu_{y,k,c})\} \end{aligned} \tag{3.21}$$

$$\begin{aligned} E(t_{i,k,c}t^T_{i,k,c}|x_i, y_i, k, c, \Theta_{old}) = &(\sigma^{-2}_{x,k,c}P^T_{k,c}P_{k,c} + \sigma^{-2}_{y,k,c}C^T_{k,c}C_{k,c} + I)^{-1} \\ &+E(t_{i,k,c}|x_i, y_i, k, c, \Theta_{old})E^T(t_{i,k,c}|x_i, y_i, k, c, \Theta_{old}) \end{aligned} \tag{3.22}$$

3.5.1.3 Maximization

In the maximization step, one can maximize the expected value of the complete data log-aposteriori with respect to the model parameters. To update the priors of each component, consider the expected value of the log-aposteriori in Equation 3.17 and separate the terms with the proportional value $p(c|k)$ as follows:

$$\begin{aligned} l_1(c) &= \sum_{i=1}^{n}\sum_{k=1}^{K} p(k|x_i, y_i, \Theta_{parent}) \sum_{c=1}^{g_k} p(c|x_i, y_i, k, \Theta_{old}) \ln p(c|k) + \sum_{k=1}^{K}\sum_{c=1}^{g_k} \ln[p(\theta_{(k,c)})] \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K} p(k|x_i, y_i, \Theta_{parent}) \sum_{c=1}^{g_k} p(c|x_i, y_i, k, \Theta_{old}) \ln p(c|k) + \sum_{k=1}^{K}\sum_{c=1}^{g_k} \ln[\beta(p(c|k)p(k))^{\frac{-N}{2}}] \end{aligned} \tag{3.23}$$

Considering the constraint of $\sum_{c=1}^{g_k} p(c|k) = 1$, a Lagrange multiplier λ is introduced, and the updated value of $p(c|k)$ is determined by maximizing the following equation:

$$l_2(c) = l_1(c) + \lambda(\sum_{c=1}^{g_k} p(c|k) - 1) \tag{3.24}$$

Equation 3.25 is derived by setting the derivatives of $l_2(c)$ with respect to $p(c|k)$ to zero:

$$\begin{aligned} &\sum_{i=1}^{n}[p(k|x_i, y_i, \Theta_{parent})p(c|x_i, y_i, k, \Theta_{old})] - \frac{N}{2} + \lambda p(c|k) = 0 \\ \Longrightarrow p(c|k) = &-\frac{\sum_{i=1}^{n}[p(k|x_i, y_i, \Theta_{parent})p(c|x_i, y_i, k, \Theta_{old})] - \dfrac{N}{2}}{\lambda} \end{aligned} \tag{3.25}$$

Since priors of the components $p(c|k)$ cannot be negative, Equation 3.25 can be expressed as:

$$p(c|k) = -\frac{\max\left\{0, \sum_{i=1}^{n}[p(k|x_i, y_i, \Theta_{parent})p(c|x_i, y_i, k, \Theta_{old})] - \dfrac{N}{2}\right\}}{\lambda} \tag{3.26}$$

Considering $\sum_{c=1}^{g_k} p(c|k) = 1$, λ can be computed as:

$$\lambda = -\sum_{c=1}^{g_k} \max\left\{0, \sum_{i=1}^{n}[p(k|x_i, y_i, \Theta_{parent})p(c|x_i, y_i, k, \Theta_{old})] - \frac{N}{2}\right\} \tag{3.27}$$

As a result, updated value of the proportions is given by:

$$p(c|k) = \frac{\max\left\{0, \sum_{i=1}^{n}[p(k|x_i, y_i, \Theta_{parent})p(c|x_i, y_i, k, \Theta_{old})] - \dfrac{N}{2}\right\}}{\sum_{c=1}^{g_k} \max\left\{0, \sum_{i=1}^{n}[p(k|x_i, y_i, \Theta_{parent})p(c|x_i, y_i, k, \Theta_{old})] - \dfrac{N}{2}\right\}} \tag{3.28}$$

The prior probability of component (c, k) is computed using the following equation:

$$p(c,k) = p(c|k)p(k) \tag{3.29}$$

where $p(k)$ has a constant value from the previous level. The maximization results of the other parameters are as follows:

$$\begin{aligned}
&\frac{\partial E\Big[\ln[p(X,Y,T,k,c,\Theta)]\Big]}{\partial P_{k,c}} = 0 \Longrightarrow \\
P_{k,c}^{new} =& \Big[\sum_{i=1}^{n}[p(k,c|x_i,y_i,\Theta_{old})(x_i - \mu_{x,k,c})E^T(t_{i,k,c}|x_i,y_i,k,c,\Theta_{old})]\Big] \\
&\times \Big[\sum_{i=1}^{n_1}[p(k,c|x_i,y_i,\Theta_{old})E(t_{i,k,c}t_{i,k,c}^T|x_i,y_i,k,c,\Theta_{old})]\Big]^{-1}
\end{aligned} \tag{3.30}$$

$$\begin{aligned}
&\frac{\partial E\Big[\ln[p(X,Y,T,k,c,\Theta)]\Big]}{\partial C_{k,c}} = 0 \Longrightarrow \\
C_{k,c}^{new} =& \Big[\sum_{i=1}^{n}[p(k,c|x_i,y_i,\Theta_{old})(y_i - \mu_{y,k,c})E^T(t_{i,k,c}|x_i,y_i,k,c,\Theta_{old})]\Big] \\
&\times \Big[\sum_{i=1}^{n}[p(k,c|x_i,y_i,\Theta_{old})E(t_{i,k,c}t_{i,k,c}^T|x_i,y_i,k,c,\Theta_{old})]\Big]^{-1}
\end{aligned} \tag{3.31}$$

$$\begin{aligned}
&\frac{\partial E\Big[\ln[p(X,Y,T,k,c,\Theta)]\Big]}{\partial \mu_{x,k,c}} = 0 \Longrightarrow \\
\mu_{x,k,c}^{new} =& \frac{\Big\{\sum_{i=1}^{n} p(k,c|x_i,y_i,\Theta_{old})[x_i - P_{k,c}E(t_{i,k,c}|x_i,y_i,k,c,\Theta_{old})]\Big\}}{\Big\{\sum_{i=1}^{n} p(k,c|x_i,y_i,\Theta_{old})\Big\}}
\end{aligned} \tag{3.32}$$

$$\begin{aligned}
&\frac{\partial E\Big[\ln[p(X,Y,T,k,c,\Theta)]\Big]}{\partial \mu_{y,k,c}} = 0 \Longrightarrow \\
\mu_{y,k,c}^{new} =& \frac{\Big\{\sum_{i=1}^{n} p(k,c|x_i,y_i,\Theta_{old})[y_i - C_{k,c}E(t_{i,k,c}|x_i,y_i,k,c,\Theta_{old})]\Big\}}{\Big\{\sum_{i=1}^{n} p(k,c|x_i,y_i,\Theta_{old})\Big\}}
\end{aligned} \tag{3.33}$$

$$\frac{\partial E\Big[\ln[p(X,Y,T,k,c,\Theta)]\Big]}{\partial \sigma^{2new}_{x,k,c}} = 0 \Longrightarrow$$

$$\begin{aligned}\sigma^{2new}_{x,k,c} =&\Big\{\sum_{i=1}^{n} p(k,c|x_i,y_i,\Theta_{old})[(x_i-\mu_{x,k,c})^T \\ &(x_i-\mu_{x,k,c}) - 2E^T(t_{i,k,c}|x_i,y_i,k,c,\Theta_{old})P^{newT}_{k,c}(x_i-\mu_{x,k}) + \\ &trace(P^{newT}_{k,c}P^{new}_{k,c}E(t_{i,k,c}t^T_{i,k,c}|x_i,y_i,k,c,\Theta_{old}))\Big\}/ \\ &\Big\{m(\sum_{i=1}^{n} p(k,c|x_i,y_i,\Theta_{old}))\Big\}\end{aligned} \tag{3.34}$$

$$\frac{\partial E\Big[\ln[p(X,Y,T,k,c,\Theta)]\Big]}{\partial \sigma^{2new}_{y,k,c}} = 0 \Longrightarrow$$

$$\begin{aligned}\sigma^{2new}_{y,k,c} =&\Big\{\sum_{i=1}^{n} p(k,c|x_i,y_i,\Theta_{old})[(y_i-\mu_{y,k,c})^T(y_i-\mu_{y,k,c}) \\ &- 2E^T(t_{i,k,c}|x_i,y_i,k,c,\Theta_{old})C^{newT}_{k,c}(y_i-\mu_{y,k}) + \\ &trace(C^{newT}_{k}C^{new}_{k,c}E(t_{i,k,c}t^T_{i,k,c}|x_{i,o},y_i,k,c,\Theta_{old}))]\Big\}/ \\ &\Big\{r(\sum_{i=1}^{n} p(k|x_i,y_i,\Theta_{old}))\Big\}\end{aligned} \tag{3.35}$$

The expectation and maximization steps are iterated over the equations until the parameters converge to their optimal values.

3.5.2 Splitting Components

At each level, after building the hierarchical model described in section 3.5.1, the performance of the parent components are compared with its two offspring . If the two offspring model outperforms their parent model, the parent is replaced by the offspring. Otherwise, the parent is not split and its model is retained as the best one.[39]

In order to compare the performance of parents and their offspring, a proper model selection criterion should be employed. In this thesis, we selected the MML criterion that is described in section 3.4. To utilize the MML criterion for each component in Equation 3.12, the value of $\log p(X|\Theta)$ for each component should be determined using the following equation:

$$p(X, Y, z|\Theta) = p(z|X, Y, \Theta)p(X, Y|\Theta) \tag{3.36}$$

where z indicates the origin component of the samples.

$$\begin{aligned}\ln p(X, Y|\Theta) = &-\sum_{i=1}^{n}\sum_{k=1}^{K} p(k|x_i, y_i, \theta_k)\ln p(k|x_i, y_i, \theta_k) \\ &+\sum_{i=1}^{n}\sum_{k=1}^{K} p(k|x_i, y_i, \theta_k)\ln[p(y_i, x_i|\theta_k)p(k)]\end{aligned} \tag{3.37}$$

As a result, the likelihood value corresponding to each component (k) is as follows:

$$\begin{aligned}\ln p(X, Y|\Theta)_k = &-\sum_{i=1}^{n} p(k|x_i, y_i, \theta_k)\ln p(k|x_i, y_i, \theta_k) \\ &+\sum_{i=1}^{n} p(k|x_i, y_i, \theta_k)\ln[p(y_i, x_i|\theta_k)p(k)]\end{aligned} \tag{3.38}$$

In summary, for component (k), the selection criterion is as follows:

$$\begin{aligned}L(\theta, X, Y)_k = &\frac{N}{2}\ln(\frac{n\pi_k}{12}) + \frac{1}{2}\ln\frac{n}{12} + \frac{N+1}{2} + \sum_{i=1}^{n} p(k|x_i, y_i, \theta_k)\ln p(k|x_i, y_i, \theta_k) \\ &-\sum_{i=1}^{n} p(k|x_i, y_i, \theta_k)\ln[p(y_i, x_i|\theta_k)p(k)]\end{aligned} \tag{3.39}$$

And for the offspring of component k, the selection criterion is:

$$\begin{aligned}L(\theta, X, Y)_{C(k)} = &\frac{N}{2}\sum_{(c,k):\pi_{c,k}>0}\ln(\frac{n\pi_{c,k}}{12}) + \frac{k_{nz}}{2}\ln\frac{n}{12} + \frac{k_{nz}(N+1)}{2} \\ &+\sum_{i=1}^{n}\sum_{c=1}^{k_{nz}} p(c, k|x_i, y_i, \theta_{c,k})\ln p(c, k|x_i, y_i, \theta_{c,k}) \\ &-\sum_{i=1}^{n}\sum_{k=1}^{k_{nz}} p(c, k|x_i, y_i, \theta_{c,k})\ln[p(y_i, x_i|\theta_{c,k})p(c, k)]\end{aligned} \tag{3.40}$$

where k_{nz} denotes the number of components whose prior probabilities, i.e., $\pi_{c,k}$ are larger than zero.

3.5.3 Merging Components

Despite a good performance of hierarchical MPPCR, due to its hierarchical nature, it may have some problems in estimating the correct number of clusters. In other words, when a component is split, the next step is performed on the new estimated components, and it is not possibe to undo what is done at the previous level. To overcome this problem, we have added a step of merging highy overlapped components to this method. In other words, after each splitting step, the highly overlapped components (if they exist) are detected and merged. We provide a detailed explanation of the problem through an illustrative example:

Consider a two dimensional dataset containing three mixture components. Applying hierarchical MPPCR to this problem may lead to estimating an incorrect number of mixture components because one of the components may be split into two components in the first level. The scatter plot of this example is given in Figure 3.5. According to this plot, component 2 is split into two components, and since there is no merging step, in the further level each new component has been split into two components, and the final number of components is incorrectly estimated as 4.

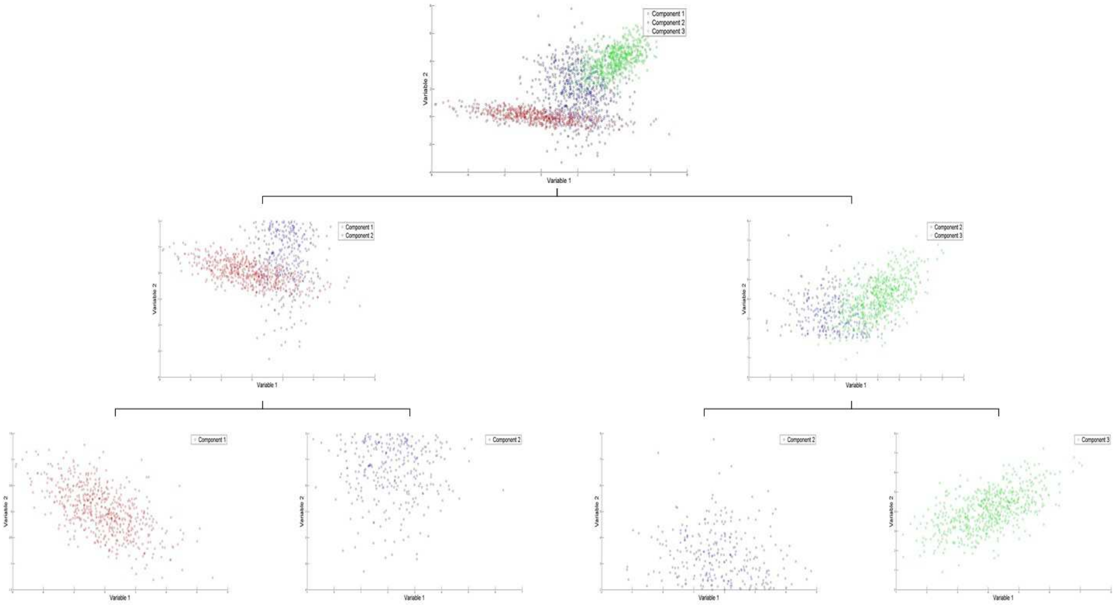

Figure 3.5: *Plot of hierarchical MPPCR example*

In order to overcome this issue, we propose a merging step after each splitting step. To illustrate, in this example, we apply the merging test at the third level (when there are four components from two parents that are each other's sibling). Component 2 that is split will show a high overlap criterion, so the split components are merged with each other. The scatter plot in this case is given in Figure 3.6.

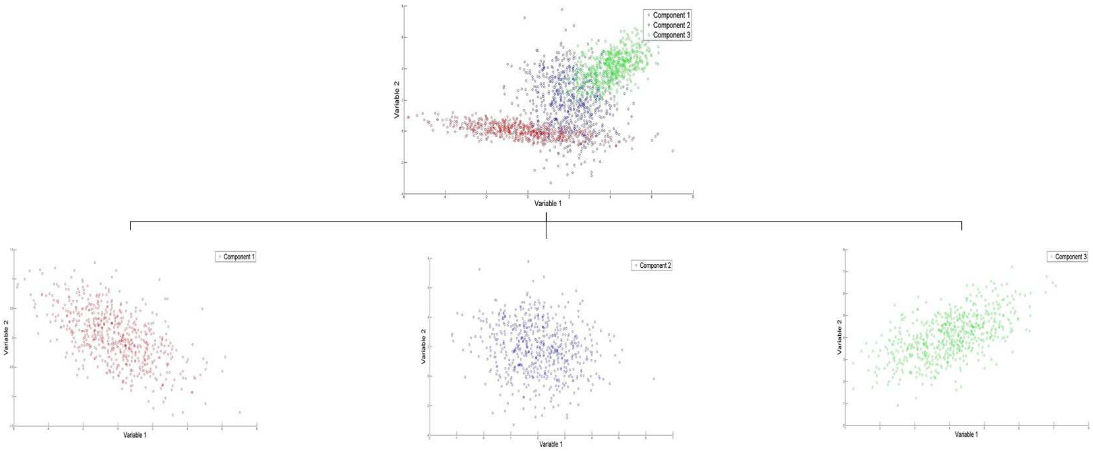

Figure 3.6: *Hierarchical plot of the developed hierarchical MPPCR example*

3.5.3.1 Detecting Highly Overlapped Components

Many methods have been proposed in literature for detecting overlapped clusters, such as the ridge line method,[49] entropy based criterion,[50] and misclassification probability methods.[51] Henning[51] provides a detailed review of the available methods. Recently "Directly Estimated Misclassification Probability" method ($DEMP_+$) was poposed by Melnykov[52] and has shown desirable results in detecting overlapped clusters. In this thesis, we utilize this method for detecting highly overlapped components.

After convergence of the EM algorithm and estimation of the optimal parameters, the posterior probability of each component is calculated. Based on the Baye's rule, each observation is assigned to the component that has the highest posterior probability. Since the ($DEMP_+$) method is based on misclassification probabilities, this rule can be utilized to compute pairwise misclassification probabilities. Assume a sampling point x_i is originated from component k distribution. The probability that

x_i is misclassified to component j is given as:[52]

$$w_{j|k} = p\Big(t_{ik} < t_{ij} | x_i \sim p(x_i; \theta_k)\Big) \tag{3.41}$$

where t_{ik} and t_{ij} denote posterior probabilities computed for sampling point x_i based on components k and j, respectively.

$$t_{ik} = p(k|x_i, \Theta) = \frac{p(x_i|k, \Theta)\pi_k}{p(x_i|\Theta)} \tag{3.42}$$

$$t_{ij} = p(j|x_i, \Theta) = \frac{p(x_i|j, \Theta)\pi_j}{p(x_i|\Theta)} \tag{3.43}$$

By substituting Equations 3.42 and 3.43 into Equation 3.41, we get:

$$w_{j|k} = p\Big(p(x_i|j, \Theta)\pi_j < p(x_i|k, \Theta)\pi_k | x_i \sim p(x_i; \theta_k)\Big) \tag{3.44}$$

For each two component k and j, their misclassification probability can be computed as:

$$w_{j,k} = w_{j|k} + w_{k|j} \tag{3.45}$$

The value of $w_{j,k}$ can be estimated using Monte Carlo simulations. However, another approach is proposed by Maitra and Melnykov.[53] Consider computing $w_{j|k}$. In order to utilize Monte Carlo simulation, a large sample $y_1, y_2, ..., y_N$ is separated from the Gaussian component k, $p(x_i; \theta_k)$, and the posterior probability of component k and j for each sampling point is computed. As a result, $w_{j|k}$ can be estimated as:

$$\hat{w}_{j|k} = \frac{1}{N} \sum_{i=1}^{N} I\Big(p(y_i|j, \Theta)\pi_j < p(y_i|k, \Theta)\pi_k\Big) \tag{3.46}$$

where I is an indicator function that can have a value of 0 or 1.

The value of the misclassification probability determines the level of overlap between two mixture components. The relationship between Rand index (cluster similarity index) and the level of overlap for different dimensions is provided by Melnykov.[52] This value is selected based on the problem. Note that in this problem, since we wish to detect the component that is incorrectly split, high values for w should be selected.

3.5.3.2 Parameter Estimation of Merged Component

The estimation of the parameters of merged components is similar to MPPCR[22] with some fixed components and known priors for all components. Consider the example in section 3.5.3. After estimating the level of overlap, a high overlap value for component 2 on the right and left sections in Figure 3.5 are detected. As a result, the two blue components (component 2) should be merged. To estimate the parameters of the merged components, parameters of the component 1 and 3 are fixed. The prior and posterior probability of the merged component also equals the summation of the priors of two origin components. Other parameters can be estimated using the EM algorithm for the MPPCR problem.

A summary of the proposed framework is given in Figure 3.7.

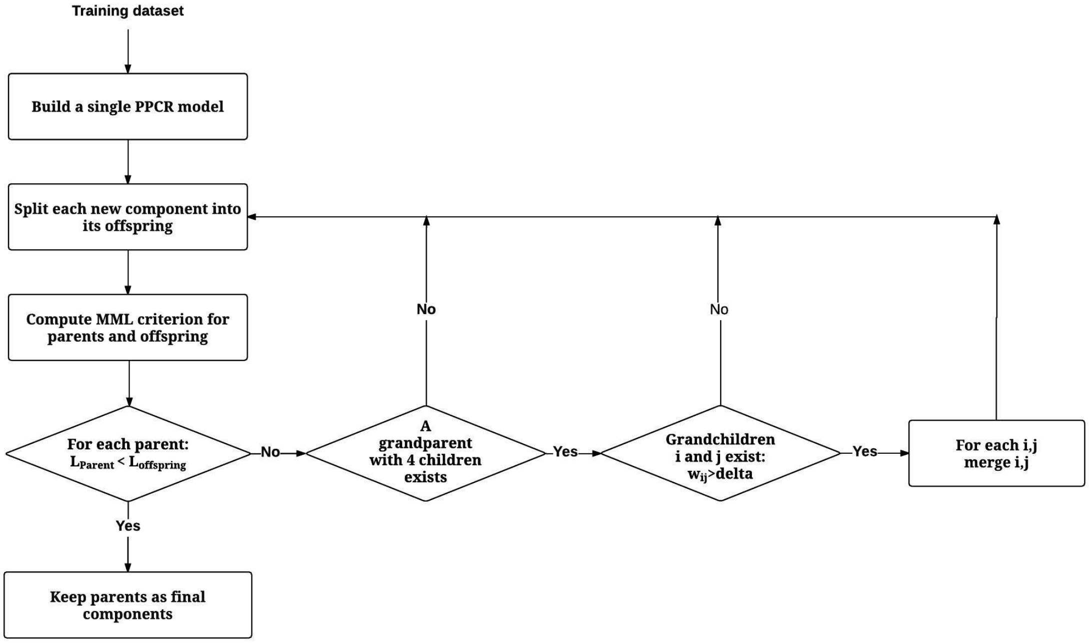
Training dataset
Build a single PPCR model
Split each new component into its offspring
Compute MML criterion for parents and offspring
For each parent: $L_{Parent} < L_{offspring}$
No
A grandparent with 4 children exists
Yes
Grandchildren i and j exist: w_{ij}>delta
Yes
For each i,j merge i,j
No
No
Yes
Keep parents as final components

Figure 3.7: *An overview of the proposed hierarchical MPPCR*

3.6 Case studies

Two case studies are considered in this section to demonstrate the performance of the developed method. In the first part, a numerical example is provided, and in the second part, the developed method is illustrated through an experimental example.

3.6.1 Numerical example

The purpose of this numerical example is to demonstrate the capability of the proposed method in estimating the number of components. A simulated dataset is generated using the following model:

$$\begin{aligned} x_k &= P_k t_k + e_k + \mu_{x,k} \\ y_k &= C_k t_k + f_k + \mu_{y,k} \end{aligned} \tag{3.47}$$

It is a three mixture component problem, and k has the values of 1, 2 or 3. There are two input and two output variables in each component. The simulated number of latent variables is two. P_k and C_k are 2×2 weighting matrices that are selected randomly. t_k is the latent variable vector in each component and follows a Gaussian distribution of $\mathcal{N}(0, I)$. e_k and f_k are input and output measurement noises in each component, respectively, and also follow Gaussian distributions with zero mean and a variance of 0.01^2 , i.e., $\sigma^2_{x,k} = 0.01^2$ and $\sigma^2_{y,k} = 0.01^2$. $\mu_{x,k}$ and $\mu_{y,k}$ are mean values of the input and output in each component, respectively. In each component, 600 data samples are generated and are used for estimating the number of components. The parameters of the model are given in Table 3.1.

Table 3.1: *Model parameters to generate the simulation data*

$$\pi = \begin{pmatrix} 1/3 & 1/3 & 1/3 \end{pmatrix}$$

$$P_1 = C_1 = \begin{pmatrix} 1.77 & 0.7316 \\ -0.3374 & 0.4883 \end{pmatrix}, \; P_2 = C_2 = \begin{pmatrix} 0.6027 & -0.1601 \\ -0.7104 & 1.2871 \end{pmatrix}$$

$$P_3 = C_3 = \begin{pmatrix} -0.7157 & -0.9474 \\ 0.465 & -0.395 \end{pmatrix}$$

$$\mu_{x,1} = \mu_{y,1} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \; \mu_{x,2} = \mu_{y,2} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \; \mu_{x,3} = \mu_{y,3} = \begin{pmatrix} 4 \\ 4 \end{pmatrix}$$

In order to evaluate the performance, two methods are compared. The first method is the developed hierarchical MPPCR method, and the second method is the hierarchical MPPCR method without the merging step. To compare the results of the two methods, three criteria are evaluated. The first criterion is the number of components that are estimated. The second criterion is the Adjusted Rand Index (ARI), and the third criterion is the Fowlkes–Mallows Index (FM index). The latter criteria can be described as follows:

3.6.1.1 Adjusted Rand Index (ARI)

The ARI demonstrates the level of agreement between two partitions. It is usually considered as a clustering validation criterion between the true and estimated clusters. Assume i and j represent the partitions of each partitioning P_1 and P_2. The formulation of ARI can be expressed as[54]:[55]

$$ARI = \frac{\sum_{i,j}\binom{n_{i,j}}{2} - \left[\sum_i \binom{n_{i.}}{2} \sum_j \binom{n_{.j}}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{n_{i.}}{2} + \sum_j \binom{n_{.j}}{2}\right] - \left[\sum_i \binom{n_{i.}}{2} \sum_j \binom{n_{.j}}{2}\right] / \binom{n}{2}} \tag{3.48}$$

where n is the total number of objects, $n_{i,j}$ is the number of objects in both partitions $P_{1,i}$ and $P_{2,j}$, and $n_{i.}$ and $n_{.j}$ are the number of objects in partitions $P_{1,i}$ and $P_{2,j}$, respectively.

3.6.1.2 Fowlkes–Mallows Index (FM index)

FM index is the geometric mean of precision, i.e. the probability that two objects are in the same estimated cluster given the same true lables, and the probability that two objects have the same true labels given the same estimated clusters. The formulation of FM index is as follows[56]:[39]

$$FM = \frac{\sum_{i=1}^{G} \sum_{j=1}^{K} \binom{n_{i,j}}{2}}{\left(\sum_{i=1}^{G} \binom{n_{i.}}{2} \sum_{j=1}^{K} \binom{n_{.j}}{2}\right)^{0.5}} \tag{3.49}$$

The developed hierarchical MPPCR, and the hierarchical MPPCR method without the merging step are built based on the simulated dataset. The estimated number of components using the developed method is three, which is the same as the true number of components. On the other hand, the regular hierarchical MPPCR model estimates the number of components to be four. The posterior probabilities of each component for both models are given in Figures 3.8 and 3.9.

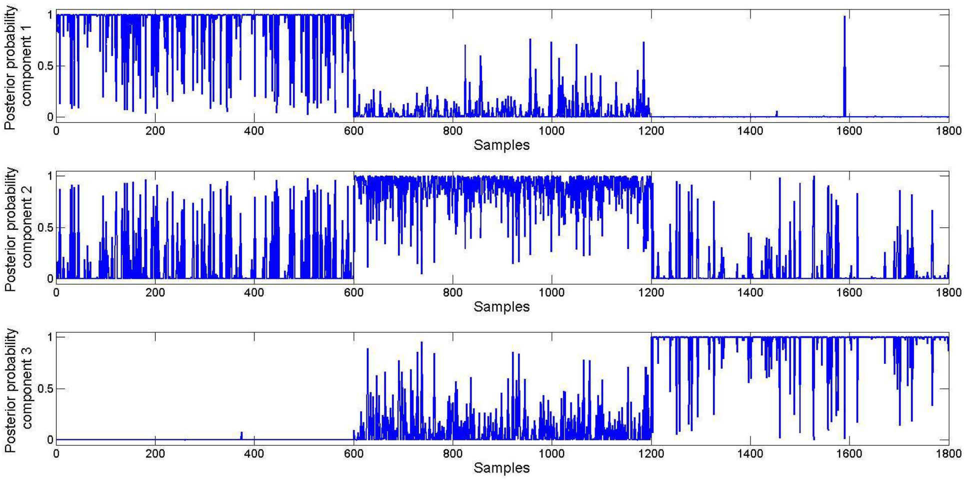

Figure 3.8: *Estimated posterior probabilities using the developed hierarchical MPPCR model*

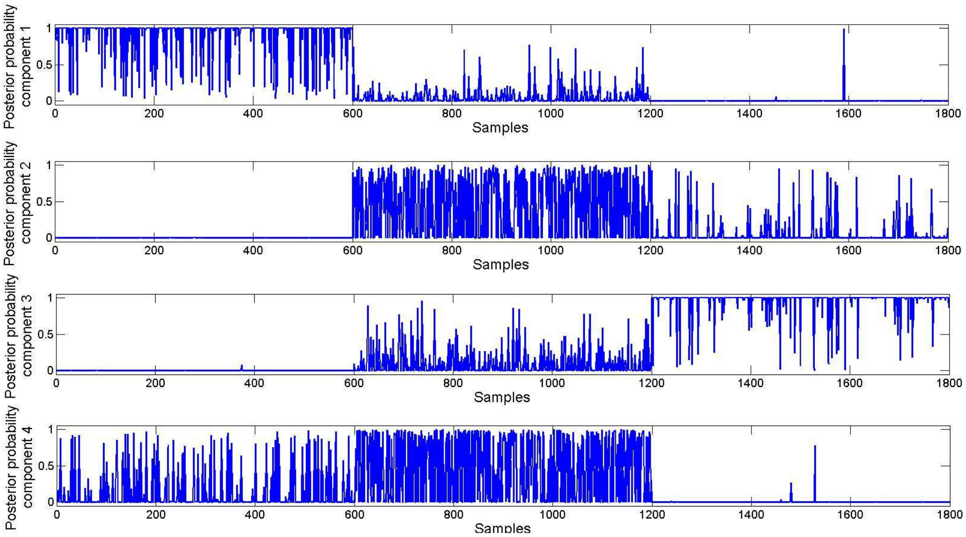

Figure 3.9: *Estimated posterior probabilities using the regular hierarchical MPPCR model*

According to Figures 3.8 and 3.9, the developed model has estimated three components, and the partitions are approximately detected. However, using the regular hierarchical MPPCR, four components are detected. In other words, using regular hierarchical MPPCR, component 2 is split into components 2 and 4 while using the developed model, components 2 and 4 are merged and are formed as component 2. In addition, for evaluating the performance of both models, the ARI and FM index are calculated for each approach based on the true partitioning labels. For the developed model the ARI and FM index are 0.8006 and 0.8672, respectively. On the other hand, for the regular hierarchical MPPCR model, the ARI and FM index are 0.7145 and 0.8040, respectively. Since for both indices, a value closer to 1 shows a better agreement between the estimated and true partitions, the developed method shows a better performance in detecting components. The results are summarized in Table 3.2.

Table 3.2: *The results for the developed and regular hierarchical MPPCR*

Model	Estimated K	ARI	FM
Developed hierarchical MPPCR	3	0.8006	0.8672
Regular hierarchical MPPCR	4	0.7145	0.8040

3.6.2 Experimental example: Hybrid Tank System

An experiment is conducted on a hybrid tank system in the process control laboratory at University of Alberta to demonstrate the performance of the proposed method in detecting the number of operating regions, i.e., the number of mixture components. The schematic of the system is presented in Figure 3.10. The system contains three tanks with the same dimensions. Tank 2 is connected to tanks 1 and 3 through six valves V_1, V_2, V_3, V_4, V_6 and V_8. All the tanks have individual discharge pipes, i.e., V_5, V_7, V_9 at the bottom. Tanks 1 and 3 have inlet pumps to let the water flow into the system, and all the valves can be open or closed to design the operation pattern. The water level of tanks 1 and 2 are controlled through a cascade control that allows to manipulate flow rates and the speed of pumps.

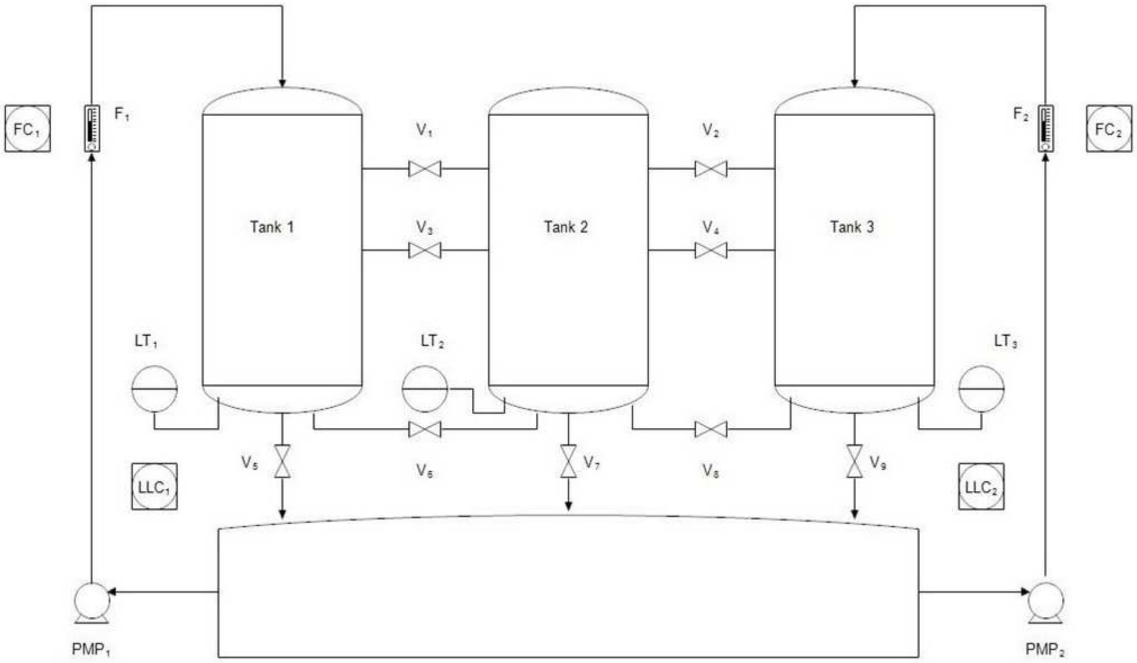

Figure 3.10: *The schematic diagram of the Hybrid Tank System*

In this experiment, valves V_1, V_2, V_3, V_4 are open to make the water flow possible at high water levels. In addition, valve V_7 is also open to make the water level of tank 2 stable. The experiment is conducted, and 500 sampling data are collected in three operating regions, around set points given in Table 3.3. Collected sampling data points are presented in Figure 3.11. Note that the values are normalized. In operating region 1, the flow is limited to tank 3. In operating region 2, tank 3 and 2 have flow, and in the operating region 3 the flow enters all the three tanks. In order to determine the number of operating regions, steady state parts of the dataset are separated. The water level of tank 3 is selected as an output, and the flow rate of tanks 1 and 3, and the water level of tank 1 are selected as inputs. The developed hierarchical MPPCR is applied on the dataset. As a result, the number of operating regions, i.e., mixture components, is estimated as 3, and ARI and FM index values are 0.9849 and 0.99, respectively. This indicates a high agreement between the estimated and true mixture components.

Table 3.3: *The hybrid tank system operating regions*

Region	Tank 3 water level	Tank 1 water level
1	30	0
2	50	0
3	65	65

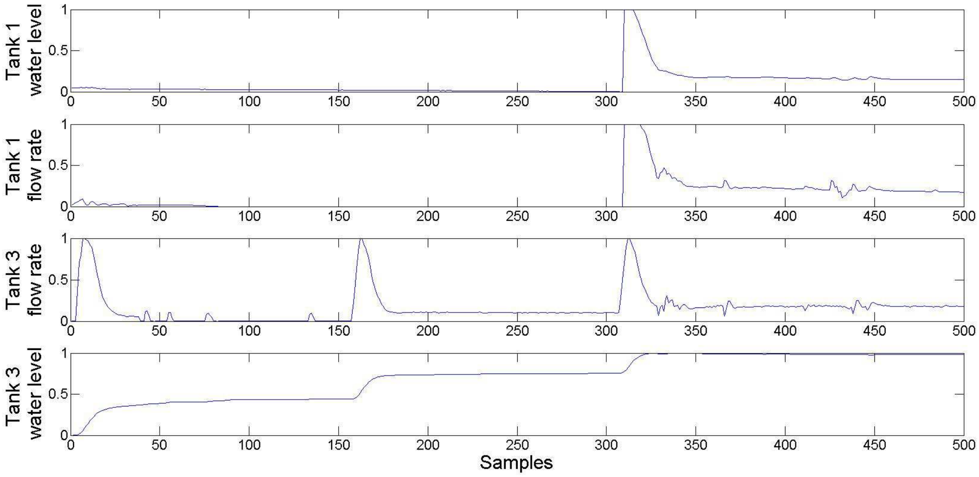

Figure 3.11: *Collected dataset*

3.7 Conclusion

In this chapter, we proposed a hierarchical MPPCR model which automatically determines the number of components in a multi-mode system. Since, in most available methods a priori information about the possible range of the number of components is required, if this information is not available, these methods may be too time consuming and inaccurate. This chapter is based on a hierarchical framework that starts with the minimum possible number of components that may be split into more components based on the MML criterion. In addition, a merging step added to the hierarchical MPPCR has improved the performance. We presented two case studies which demonstrated the improved performance of the developed model.

Chapter 4

Optimality Assessment

Performance of the processes may deviate from the initial design over time due to disturbances and uncertainties. Because of the great importance of optimality, it is necessary to develop systematic methods for online optimality assessment based on the operating process data. Some processes may have multiple operating modes caused by the set point change of the critical process variables in order to achieve different product demands. On the other hand, operating region in each operating mode can alter because of uncertainties. In this chapter, we will propose an optimality assessment approach for multi-mode processes with multiple operating regions. Kernel density based method for offline mode detection is adopted and improved in order to recognize noise from transition, identify true labels, and also improve the accuracy of estimation of the mode change instant. Then a modified mixture discriminant analysis (MclustDA) method based on the detected labels is employed to build the predictive classifier and incorporated with process knowledge to increase precision of the online mode estimation. In each steady state operating mode, developed hierarchical MPPCR in chapter 3 is employed for estimating the number of operating regions. The developed mixture semi-supervised probabilistic principle component regression with missing inputs in chapter 2 is trained to detect operating regions and local optimality value of each region, and build predictive model for online assessment in the case of missing inputs as well as outputs. In addition, for transitions, dynamic PCA is utilized for transition grades analysis as well as prediction of the optimality value. In online assessment, the operating mode, operating region and optimality value are predicted, and in the case of non-optimum performance, the causal variables are detected using adopted probabilistic framework through sequential forward floating search (SFFS) method. Finally, the proposed method is applied on Tennessee Eastman benchmark process to evaluate the performance.

4.1 Introduction

Process operating performance assessment including both optimality and safety is known as an important issue in process industry and has attracted attentions in both academia and industry. Since performance of processes may deteriorate over time, due to disturbances or process condition changes, assessment of the performance in online operating processes is necessary.[57]

Safety assessment is one of the important ingredients of process performance assessment. Many studies have considered safety analysis based on qualitative and quantitative methods[58 59 60 61 62] . As a parallel study, Process monitoring including fault detection and diagnosis is well known and has been studied extensively[63 64 65 66 67] . Moreover, many methods have been developed for process optimization[68 69 70 71 72] . In such methods, the ultimate goal is to find the optimal conditions to run the process. On the other hand, due to the process disturbances and other uncertainties, the process performance will depart from the optimum. Therefore, it is necessary to continuously monitor process performance. This type of analysis has been named optimality assessment. Although various aspects of safety analysis have been studied extensively, optimality assessment has not been well studied.

Recently, some studies[57 73 74 75] have been conducted on opitmality assessment. For instance, Ye et al. studied online probabilistic safety and optimality assessment for multimode processes.[57] They used the Gaussian mixture model (GMM) to find the characteristics of steady state modes. Safety and optimality indices were defined and calculated based on the process knowledge and data distributions. To classify the obtained optimality (OI) and safety indices (SI) into different performance levels, a hierarchical-level classification was proposed, and for each class, margin analysis was proposed. Finally, the performance was predicted during the online assessment. However, there are some restrictions associated with this method. For example, in many cases safety and optimality indices are not available frequently, and it is necessary to develop a predictive model to estimate their value continuously. In addition, each operation mode is assumed to follow a Gaussian distribution that may not happen in practice. Also, the proposed method does not consider the cause diagnosis of non-ideal performance in the system.

In another work by Liu and his co-workers, optimality assessment was studied by introducing the comprehensive economic index (CEI) as an optimality index.[73] They partitioned the dataset into different grades based on the value of their CEI. They developed a model for each performance grade using the total projection to latent structure (T-PLS) method. They also developed a non-optimum cause identification method based on variable contributions. In further work, the same researchers extended the optimality assessment for multimode processes and worked on transitions as well as steady state modes.[74] Very recently, another work extended finite gaussian mixture model based gaussian process regression (FGMM-GPR) method to non-Gaussian multimode processes.[75] In this work, in addition to steady state modes, transitions and non-optimum cause identification were studied. The mentioned studies have covered optimality assessment, however some limitations are associated with these methods. First, the number of mixture components in each operating mode is considered to be known whereas usually it is not available in practice. Second, in many cases, not all input and output data points are available to assess the performance due to the sensor failure or delays in measuring some variables. Therefore, any proposed method should be able to deal with the missing outputs and inputs due to large measurement delays and mesurement failure. Third, it is assumed that the operating modes are labeled. However, in practice the labels are not always available, and unsupervised techniques should be employed to estimate them.

There are some important issues associated with optimality assessment of industrial processes. First, some of the processes have multiple operating modes due to the operating condition changes, different product demands, etc. This issue leads to different steady state modes and the transitions between steady state modes. The steady state modes are the major parts of the operating process, and the main products with the desired charecteristics are produced in these stages. On the other hand, transitions are short periods between steady state modes that have dynamic features, and main products cannot be obtained from the transitions. Second, in most processes, each main operating mode, i.e. steady state mode, does not follow a unimodal Gaussian distribution due to the uncertainties in industrial processes. In other words, change of operating mode is usually based on the production demand with different components, component ratio, rate of production and so on that is specified by changing the operating points of main operational variables.[74] On the other hand, in each operating mode, there are several operating regions that are caused by the uncertainties. Third, in many practical processes, the optimality related variables have slower rates of measurement compared to the process variables, and they are always not available. In addition, some input variables may not be always available due to the measurement device failure.

In this chapter, a novel method for optimality assessment based on Probabilistic Principal Component Regression (PPCR) is proposed. It is first described for the unimodal processes that are more common in practice and then is extended to multiple operating mode processes. For unimodal processes, the developed method consists of two stages: offline training and online assessment. In offline training, the steady state data including process variables as well as optimality index are collected. Depending on the process, the definition of OI can vary. For example, depending on the process OI can be operation costs, profit, product quality and so on. To have online estimation of OI, the aim is to build a predictive model of OI based on the process variables. Since each operating mode usually has multiple operating regions, and the datasets contain simultaneous missing inputs as well as outputs, the developed Mixture Semisupervised PPCR (MSPPCR) with missing inputs is employed for modeling. However, since the number of operating regions is usually unknown, the developed hierarchical Mixtures of PPCR is employed to estimate the number of operating regions. MSPPCR model describes the Gaussian distribution of OI in each operating region, based on which the representative value of OI in each operating region can be obtained. By comparison of the local indices of each operating region, their optimality condition is analyzed. In online assessment, the operating region of new data point is estimated based on its posterior probability. Based on the constructed model, OI is predicted using Bayesian inference to evaluate the process performance. When the process performance is non-optimum, diagnosing the cause of the problem helps steering the process to a better operation performance. The probabilistic contribution analysis based on missing variable[76] approach is adopted to address this issue. We utilize sequential floating forward search (SFFS) method instead of a branch and bound method to ease the solution.

For multiple operating mode processes, it is assumed that the data points are unlabeled with respect to the operating modes. In other words, the number of operating modes and the operating mode of each data point are unknown. To estimate the labels of the dataset, critical process variables governing the change of operating modes are selected and named as scheduling variables. Based on the selected scheduling variables, a local kernel density based approach for offline mode detection[77] is adopted and improved to differentiate noise from transition, detect true labels, and also enhance the precision of the estimation of the mode change instant. In order to estimate the operating modes in online assessment, Mixture Discriminant Analysis (MDA) is built based on the labeled data set. In addition, to improve the accuracy of online mode identification, the process knowledge is incorporrated with the MDA results. When the steady state modes and transitions are detected, optimality assessmnet of steady state modes follows the same procedure as uni-modal processes while for transitions Dynamic PCR (DPCR) model is built, and the performance grades are compared based on the DPCR loading matrices.[78]

The rest of this chapter is arranged as follows: In section 4.2, the proposed optimality assessment strategy for steady state modes is described. In section 4.3, the assessment method for the transitions are studied. In section 4.4, the mode identification method for multiple operating mode processes is described. In section 4.5, our proposed approach is tested on Tennessee Eastman (TE) process. Conclusions are presented in section 4.6.

4.2 Steady State Modes

4.2.1 Steady State Definition

Steady state modes are the main parts of the operating process where no essential change in the critical process variables, flowsheet configuration, product demand and so on happens. The main products with the desired characteristics are produced in these operating modes. The process unit is quasi steady state, and as a result its main components are at steady state. Note that the steady state from the process perspctive means the process variables change within a small range, and the slope of their change is small[78].[79]

4.2.2 Offline Training

In offline training, the number of operating regions as well as the model of the training dataset using developed MSPCR model is characterized. In the next step, based on the obtained Gaussian distributions for the OI, the local OI values of each operating region is obtained. For example, a scatter plot of an operating mode with three operating regions projected into two variables is presented in Figure 4.1. In this system, each operating region is characterized utilizing a Gaussian distribution. Since each region follows a Gaussian distribution, the local OI of each region equals the mean value of its distribution.

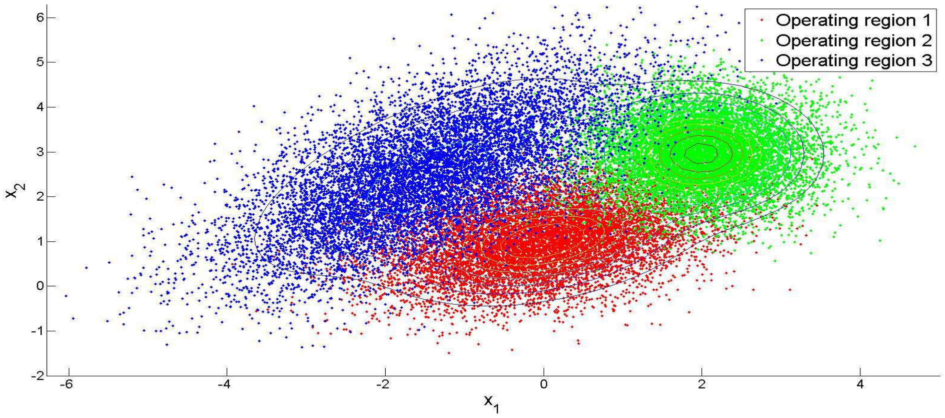

Figure 4.1: *Scatter plot of variables (Illustrative example for operating regions)*

Furthermore, based on the process knowledge, some classes for optimality values are defined, and the obtained operating regions are assigned to their corresponding classes. Note that the OI definitions depend on the considered process. Commonly, the OI can be operation cost, product quality, profits, etc.

4.2.2.1 Data Modeling

Let us assume $X = [x(1)\ \ x(2)\ \ ...\ \ x(n)] \in R^{p\times n}$ is the available dataset of the process variables with fast rate of measurement, where n is the number of data points and p is the number of process variables. On the other hand, the critical variables in optimality assessment is the OI that commonly has slower rate of measurement compared with the available process variables. To address this issue, a predictive model should be built for the OI. Assume $Y = [y(1)\ \ y(2)\ \ ...\ \ y(n_1)] \in R^{n_1\times 1}$ is the availale OI values, where n_1 is the number of labeled data points. Since the number of operating regions for modeling is unknown, it is estimated based on the complete dataset using the developed hierarchical MPPCR. In case of existence of missing values, the developed MSPPCR is employed to build predictive model where the input is X and the output is Y as mentioned above.

4.2.2.2 Analysis of OI

The definition of the OI depends on the considered operating process, and may vary in different processes. It can be product quality, operation costs, profit, etc. In the modeling part, the probabilistic predictive model of the OI based on the process variables is built. As a result, the Gaussian distribution for the OI in each operating region k is as follows:

$$f_k(y) \sim N(\mu_{y,k}, C_k C_k^T + \sigma_{y,k}^2 I) \tag{4.1}$$

In order to find the representative OI (called as the local OI) in each operating region, its expected value should be found as follows:[57]

$$OI_k = E(y) = \int f_k(y) y dy \tag{4.2}$$

Since we have the Gaussian distribution of y, the above expected value is equal to the mean value of the obtained Gaussian distribution for y, so:

$$OI_k = \mu_{y,k} \tag{4.3}$$

There are not specific rules to define classes for optimality criteria. It is completely dependent on the process that we are studying.[57] It mainly depends on two criteria: first, definition and nature of the OI, second; the possible range for the OI. Based on these criteria, the optimality classes can be defined.

4.2.3 Online Assessment

In online assessment, the operating region of the new data point is analyzed based on its posterior probability in each operating region. In the next step, the OI value is predicted based on the developed MSPCR model and Baye's rule, and its performance class is determined. Finally, if the performance is far from the optimum, the cause will be detected.

4.2.3.1 Operating Region and OI Estimation

The posterior probability of each operating region can be estimated as:

$$p(k|x_{new}, \Theta) = \frac{p(x_{new}|k, \Theta)p(k|\Theta)}{p(x_{new}|\Theta)} \tag{4.4}$$

Then the estimated latent variable $\hat{\mathrm{t}}_{k,new}$ in each operating region is as follows:

$$\hat{t}_{k,new} = (\sigma_{x,k}^2 I + P_k^T P_k)^{-1} P_k^T (x_{new} - \mu_{x,k}) \tag{4.5}$$

The predicted OI in each operating region is as follows:

$$\begin{aligned} \hat{y}_{k,new} &= C_k \hat{t}_{k,new} + \mu_{y,k} \\ &= C_k (\sigma_{x,k}^2 I + P_k^T P_k)^{-1} P_k^T (x_{new} - \mu_{x,k}) + \mu_{y,k} \end{aligned} \tag{4.6}$$

The predicted value of OI over all K operating regions is:

$$\hat{OI}(x_{new}) = \hat{y}_{new} = \sum_{k=1}^{K} p(k|x_{new}, \Theta) \hat{y}_{k,new} \tag{4.7}$$

Finally, one can find the optimality class of the new data points by comparing the obtained values for OI of the new data point with the defined classes in offline training.

4.2.3.2 Non-optimum cause diagnosis

When the process performance is non-optimum, it is beneficial to find the causal variables. One can use probabilistic contribution analysis technigue based on the missing variables. So far, this method has been applied for fault detection[76],[24] outlier detection,[80] etc. In this chapter, we will adopt this method for cause diagnosis in optimality assessment.

The adopted cause diagnosis method in optimality assessment is based on the comparison between the detected optimum operating region in offline training and the new data point with non-optimum performance. In offline training, performance of each operating region is determined based on its OI value, and its corresponding performance class is found. The most optimal region with respect to OI is called the reference region for optimality. When the predicted new data point performance is not optimal, it is treated as an abnormal data point from the reference optimal region detected in offline training, and the probabilistic cause detection method is applied on it.

The probabilistic framework for cause detection based on the missing variable approach has been applied on PPCA in Ref.[24] The authors proposed a single criterion for abnormal behavior detection instead of using T^2 and SPE criteria.[63] Following, the proposed method by Chen et al[24] is described and will be adopted to our problem. Let us assume that in online assessment a new data point with non-optimum performance is detected. Therefore, the Mahalanobis distance of the new data point from the reference region will be larger than $\chi_r^2(\beta)$:

$$M^2 = (x_{new} - \mu_{x,ref})^T C_{ref}^{-1}(x_{new} - \mu_{x,ref}) > \chi_r^2(\beta) \tag{4.8}$$

where $\chi_r^2(\beta)$ is the β-fractile of the chi-square distribution with r degree of freedom, x_{new} is the new data point in online assessment, $\mu_{x,ref}$ is the mean of Gaussian distribution of the reference region, and C_{ref} is the covariance matrix of the reference region that equals to $P_{ref}P_{ref}^T + \sigma_{x,ref}^2 I$. Missing variable approach in PPCA was introduced in Ref.[24] According to this method, new non-optimum data point, and the mean and covariance of the reference region are partitioned into observed and missing parts as follows:

$$\hat{x}_{new}^T = \begin{pmatrix} \hat{x}_o^T & \hat{x}_m^T \end{pmatrix} \tag{4.9}$$

$$\hat{\mu}_{x,ref} = \begin{pmatrix} \mu_o \\ \mu_m \end{pmatrix} \tag{4.10}$$

$$\hat{C}_{x,ref} = \begin{pmatrix} C_{oo} & C_{om} \\ C_{mo} & C_{mm} \end{pmatrix} \tag{4.11}$$

where indices of o and m stand for the observed and missing parts, respectively. The conditional probability of x_m given x_o follows a Gaussian distribution of $x_m|x_o \sim N(z_m, Q_m)$, where z_m and Q_m are as follows:

$$z_m = \mu_m + C_{mo}C_{oo}^{-1}(x_o - \mu_o) \tag{4.12}$$

$$Q_m = C_{mm} - C_{mo}C_{oo}^{-1}C_{om} \tag{4.13}$$

The conditional distribution of the complete vector $\hat{x}_{new}$ is Gaussian $\hat{x}_{new}|\hat{x}_o \sim N(z, Q)$, where

$$z = \begin{pmatrix} \hat{x}_o \\ z_m \end{pmatrix} \tag{4.14}$$

$$Q = \begin{pmatrix} 0 & 0 \\ 0 & Q_m \end{pmatrix} \tag{4.15}$$

The expected value of M^2 with respect to the conditional distribution of the complete vector given observed variables $\hat{x}_{new}|\hat{x}_o \sim N(z, Q)$ is as follows:

$$\begin{aligned} E(M^2) &= E((\hat{x}_{new} - \hat{\mu}_{x,ref})^T \hat{C}_{x,ref}^{-1} (\hat{x}_{new} - \hat{\mu}_{x,ref})) \\ &= Tr(\hat{C}_{x,ref}^{-1}[(z - \hat{\mu}_{x,ref})(z - \hat{\mu}_{x,ref})^T + Q]) \end{aligned} \tag{4.16}$$

In this approach, each variable of x_{new} is assumed to be missing, and the expected value of the Mahalanobis distance from the reference mode $E(M^2)$ is recalculated. If the missing variable contributes significantly to the abnormal event, $E(M^2)$ will have a considerable decrease compared to M^2. Moreover, if the value of $E(M^2)$ is less than the defined threshold $\chi_r^2(\beta)$, it is concluded that the missing variable is the cause of problem. Since this approach evaluates each single variable, it does not find the joint contribution of the multiple causal variables.

Recently Kariwala et al. has developed the above mentioned method for the case that the cause is a group of measured variables.[76] This approach starts with considering single variables as missing and calculating $E(M^2)$. In the next steps, the number of selected missing variables is increased until the re-calculated value of $E(M^2)$ becomes less than confidence bound $\chi_r^2(\beta)$. Therefore, the aim is to find the minimum number of missing variables in which $E(M^2)$ is less than the threshold.

One of the obstacles of this method is finding a set of missing variables from all variables, that is very time-consuming using the exhaustive search method. Kariwala et al. proposed upward branch and bound method to solve this problem.[76] Although this method gives the optimal solution, it becomes time-consuming when dealing with large systems.

In addition to the exhaustive search and branch and bound method, there are several subset selection methods such as sequential forward selection (SFS), sequential backward selection (SBS), plus l- take away r selection, sequential forward floating search (SFFS), sequential backward floating search (SBFS) , etc.[81] Several subset selection methods are compared in Ref[82].[83] The authors concluded that the sequential forward floating search (SFFS) method has almost the same performance as branch and bound algorithm and requires lower computational time. In this chapter, we use SFFS algorithm to find causal variables.

SFFS method

SFFS algorithm detects the optimum subset of features by adding a new feature in each step to the selected subset in the last step and removing some of the features added in the last steps to the subset in order to avoid local optimum.[84] SFFS method has been extensively studied in Ref,[85] we will briefly review the algorithm in the following sections.

Preliminaries

Let us assume $Y = \{y_i : 1 \leq i \leq D\}$ is the set of D available features, and the aim is to choose the subset of r features such as $X_r = \{x_i : 1 \leq i \leq r, x_i \in Y\}$. Feature selection criterion function is J, and the goal is to find a subset of k features that maximizes J. Individual significance $S_0(y_i)$ is defined as the $J(y_i)$ when only i^{th} feature y_i is used, $i = 1, 2, ..., D$.

Let us assume we have already selected the subset of k features X_k. The significance $S_{k-1}(x_j)$ for x_j, where $j = 1, 2, ..., k$, in the selected subset X_k is defined as:

$$S_{k-1}(x_j) = J(X_k) - J(X_k - x_j) \tag{4.17}$$

The significance $S_{k+1}(f_j)$ for f_j in the unselected subset $Y - X_k$, where $Y - X_k = \{f_i : i = 1, 2, ..., D - k, f_i \in Y, f_i \neq x_l \text{ for all } x_l \in X_k\}$ with respect to the selected subset X_k is defined as:

$$S_{k+1}(f_j) = J(X_k + f_j) - J(X_k) \tag{4.18}$$

The feature x_j from the selected subset X_k is called the most significant feature in the set X_k if:

$$\begin{aligned} S_{k-1}(x_j) &= \max_{1 \leq i \leq k} S_{k-1}(x_i) \\ \Rightarrow J(X_k - x_j) &= \min_{1 \leq i \leq k} J(X_k - x_i) \end{aligned} \tag{4.19}$$

The feature x_j from X_k is called the least significant feature in the set X_k if:

$$\begin{aligned} S_{k-1}(x_j) &= \min_{1 \leq i \leq k} S_{k-1}(x_i) \\ \Rightarrow J(X_k - x_j) &= \max_{1 \leq i \leq k} J(X_k - x_i) \end{aligned} \tag{4.20}$$

The feature f_j from the unselected subset $Y - X_k$ is called the most significant feature with respect to the set X_k if:

$$\begin{aligned} S_{k+1}(f_j) &= \max_{1 \leq i \leq D-k} S_{k+1}(f_i) \\ \Rightarrow J(X_k + f_j) &= \max_{1 \leq i \leq D-k} J(X_k + f_i) \end{aligned} \tag{4.21}$$

The feature f_j from the unselected subset $Y - X_k$ is called the least significant feature with respect to the set X_k if:

$$\begin{aligned} S_{k+1}(f_j) &= \min_{1 \leq i \leq D-k} S_{k+1}(f_i) \\ \Rightarrow J(X_k + f_j) &= \min_{1 \leq i \leq D-k} J(X_k + f_i) \end{aligned} \tag{4.22}$$

SFFS Algorithm

Suppose we have already selected the subset of k features X_k from the complete set of $Y = \{y_j | j = 1, 2, ..., D\}$. The goal is to find r features that maximize the criterion function of J. Note that the algorithm starts by setting $k = 0$ and $X_0 = \varnothing$

Step1 (Inclusion): In this step, the most significant feature x_{k+1} with respect to X_k from $Y - X_k$ is added to the set X_k to form X_{k+1}, i.e. $X_{k+1} = X_k + x_{k+1}$

Step2 (Conditional exclusion): Detect the least significant feature x_l in X_{k+1}. If x_l is x_{k+1} that was added in step 1, then set $k = k + 1$ and go back to step 1. If x_l is not x_{k+1}, remove x_r from X_{k+1}, and form a new subset $X_k^{'}$, i.e. $X_k^{'} = X_{k+1} - x_r$, such that $J(X_k^{'}) > J(X_k)$. If $k = 2$, set $X_k = X_k^{'}$, and return to step 1, esle go to step 3.

Step2 (Continuation of conditional exclusion): Find the least significant feature x_s in the set $X_k^{'}$. If $J(X_k^{'} - x_s) \leq J(X_{k-1})$, set $X_k = X_k^{'}$, and $j(X_k) = J(X_k^{'})$ and return to step 1. Else if $J(X_k^{'} - x_s) > J(X_{k-1})$ then remove x_s from $X_k^{'}$ and reduce the selected subset to $X_{k-1}^{'}$, i.e. $X_{k-1}^{'} = X_k^{'} - x_s$, and set $k = k - 1$. If now $k = 2$, then set $X_k = X_k^{'}$ and $j(X_k) = J(X_k^{'})$ and return to step 1, else repeat step 3.

SFFS Algorithm for Cause Diagnosis

The complete set is $x_{new} = Y = \{y_j | j = 1, 2, ..., p\}$ that includes all measured varaiabels of x_{new}. The aim is to find the minimum number of missing variables such that the re-calculated value of $E(M^2)$ is less than confidence bound $\chi_r^2(\beta)$. Suppose we have selected a subset of k missing variables X_k, the criterion function in this problem is as follows:

$$J(X_k) = M^2 - E_{\hat{Y}|X_k}(M^2) \tag{4.23}$$

where $E_{\hat{Y}|X_k}(M^2)$ is the expected value of M^2 conditioning on the selected subset be missing, i.e., X_k. The algorithm starts with $k = 1$ and is as follows:

1. Find k features from Y that is called X_k using SFFS algorithm that maximizes $J(X_k)$.

2. If $E_{\hat{Y}|X_k}(M^2)$ is less than confidence bound, $\chi_r^2(\beta)$, X_k is the final set of causes, else $k = k + 1$ and go to step 1.

4.3 Transition Modeling

4.3.1 Transition Definition

Transitions mainly happen in processes with multiple opeationg modes between two steady state modes. In this thesis, it is assumed that the changes of operating modes are supervised. In other words, the modes are altered because of change in production demand, deliberate process condition changes and so on. Because of the supervised nature of operating mode changes, critical process variables governing the change of operating modes are known in each process. These variables have a key role in detecting operating modes in performance assessment and are named as scheduling variables.

In an operating multi-mode process, main products with desired charecteristics are produced in steady state modes, and the transitions mostly produce subproducts with varying features. Several methods are proposed in literature to assess the dynamic behavior of transitions. For instance, Yu approximated the behavior of the transitions using finite mixture models (FMM).[86] In another work, transitions are modeled based on the weighted sum of sub-PCA models.[87] In another study, transition behaviors are approximated using weighted sum of steady state PCA models before and after the transition.[88] Srinivasan and his co-workers have proposed dynamic PCA (DPCA) model for analyzing the transitions behavior.[78] They have utilized similarity index between DPCA loading matrices to find similar transitions. For the purpose of optimality assessment, this method has some advantages. First, this method provides a dynamic model for transitions that is independent of steady state models. Second, transitions have their own loading matrices that can be utilized for comparison purposes. As a result, DPCA is employed in this work for transitions analysis.

4.3.2 Transition Grades Analysis

Assume there are $T_{i,j}$ transitions from steady state operating mode i to j. For each transition t between i,j, $X^t_{i,j} = [x^t_{i,j}(1)\ x^t_{i,j}(2)\ ...\ x^t_{i,j}(n^t_{i,j})]^T \in R^{n^t_{i,j}\times p}$ and $Y^t_{i,j} = [y^t_{i,j}(1)\ y^t_{i,j}(2)\ ...\ y^t_{i,j}(n^t_{i,j})]^T \in R^{n^t_{i,j}\times 1}$ are the available datasets of process variables, and optimality index, respectively, where $n^t_{i,j}$ presents the number of samples in transition t between i,j.

Since the transitions are defined based on the scheduling variables, these variables are seperated for analysis of the dynamic loading matrix of the transitions. As a result, $X^{t(s)}_{i,j} = [x^{t(s)}_{i,j}(1)\ x^{t(s)}_{i,j}(2)\ ...\ x^{t(s)}_{i,j}(n^t_{i,j})]^T \in R^{n^t_{i,j}\times N(s)}$ denotes the dataset based on the scheduling variables, where s stands for the scheduling variables, and $N(s)$ denotes the number of the scheduling variables.

DPCA considers the autocorrelation in the variables as well as their time varying features by incorporating time lagged information in the data matrix.[78] As a result the extended data matrix of X at time r is named as X_{delay} and formulated as:[89]

$$X_{delay}(l) = \begin{bmatrix} X(r) & X(r-1) & ... & X(r-l) \end{bmatrix} \tag{4.24}$$

where l denotes the order of time dependency. In order to find the DPCA projection to the latent variables, original PCA is applied on the extended data matrix $X_{delay}(l) \in R^{n\times p(l+1)}$ as follows:

$$X_{delay}(l) = T_D P_D^T + E_D \tag{4.25}$$

where $T_D \in R^{n\times k}$ denotes the latent variables, $P_D \in R^{p\times k}$ loading matrix and $E_D \in R^{n\times p(l+1)}$ residuals of the DPCA. Note that the l order and number of principal components k can be estimated based on the method developed in.[89]

In this work, for each transition t from operating mode i to j, the extended data matrix $X_{i,j}^{t(s)}(l) = \left[X_{i,j}^{t(s)}(r) \quad X_{i,j}^{t(s)}(r-1) \quad ... \quad X_{i,j}^{t(s)}(r-l) \right]$ is constructed, and its corresponding loading matrix based on Equation 4.25 is estimated and shown as $P_{i,j}^{t(s)}$. For each two transitions S and T between operating modes i and j, the similarity indices are computed as follows:[78]

$$S_{DPCA}^{\lambda}(S,T) = \frac{\sum_{n=1}^{k} \sum_{m=1}^{k} \lambda_n^S \lambda_m^T \cos^2 \theta_{nm}}{\sum_{n=1}^{k} \lambda_n^S \lambda_n^T} \tag{4.26}$$

where k is the number of principal components, λ_n^S and λ_m^T denote the n^{th} and m^{th} eigenvalues in transitions S and T covariance matrix, and $\cos^2 \theta_{nm}$ presents the cosine of the angles between n^{th} and m^{th} principal components of S and T.

When the similarity indices are computed, each two transitions S and T with the same initial and final operating modes are put in the same transition grade if their similarity index is larger than a threshold:

$$S_{DPCA}^{\lambda}(S,T) > \theta_T \tag{4.27}$$

where θ_T denotes trend-deviation.[78]

4.3.3 Transition Predictive Modeling

The predictive model is built based on the complete training dataset. Let us again assume for each transition t between i,j, $X_{i,j}^t = [x_{i,j}^t(1) \;\; x_{i,j}^t(2) \;\; ... \;\; x_{i,j}^t(n_{i,j}^t)]^T \in R^{n_{i,j}^t \times p}$ and $Y_{i,j}^t = [y_{i,j}^t(1) \;\; y_{i,j}^t(2) \;\; ... \;\; y_{i,j}^t(n_{i,j}^t)]^T \in R^{n_{i,j}^t \times 1}$ are the available datasets of process variables, and optimality index, respectively. Similar to transition grade analysis, DPCA model is built. However, the extended data matrix, i.e. $X_{i,j,delay}^t(l) \in R^{n \times p(l+1)}$ is based on all process variables. As a result, for each transition t between operating modes i , j, the DPCA is built as follows:

$$X_{i,j,delay}^t(l) = T_{i,j}^t {P_{i,j}^t}^T + E_{i,j}^t \tag{4.28}$$

When DPCA is built, the regression step is applied on detected latent variables as follows:

$$Y_{i,j,delay}^t(l) = T_{i,j}^t {C_{i,j}^t}^T + F_{i,j}^t \tag{4.29}$$

where $C^t_{i,j} \in R^{1\times k}$ and $F^t_{i,j} \in R^{n\times 1}$ denote regression matrix between $Y^t_{i,j}(l) \in R^{n\times 1}$ and $T^t_{i,j} \in R^{n\times k}$ and residuals for the regression step, respectively.

4.3.4 Online Assessment

In online assessment, the operating mode of each data point including steady state and transitions is estimated based on the mode identification method that is discussed in the next section. If a data point belongs to transition, its corresponding grade is also identified. Let us assume the new data point belongs to the grade p of the transition between mode i and j. Its corresponding optimality index is estimated as:

$$\hat{OI}(x_{new}) = \frac{\sum_{k=1}^{K^p_{ij}} \hat{OI}^p_k(x_{new})}{K^p_{ij}} \tag{4.30}$$

where K^p_{ij} denotes the number of historical transitions in grade p between modes i and j, $\hat{OI}^p_k(x_{new})$ presents the estimated value of each model in grade p between modes i and j, and $\hat{OI}(x_{new})$ is the final estimated value. A similar approach to Equation 4.30 for estimating transitions is proposed in.[75] However, they have utilized the historical value of transitions instead of $\hat{OI}^p_k(x_{new})$ that is based on the DPCA based regression model.

4.4 Mode Identification

In application, some of the processes have multiple operating modes due to the change in production demand, characteristics, and so on. Therefore, applying the method developed in section 4.2 may result in unsatisfactory performance in multi-mode processes. In order to extend the proposed algorithm to multi-mode systems, a mode identification step should be considered to detect the steady state modes as well as transitions, and then employ the methods discussed in sections 4.2 and 4.3. The mode identification consisits of labeling the operating modes and building the predictive classifier based on the estimated labels.

4.4.1 Operating Modes Labeling

Srinivasan et al. have extensively studied multi-mode process behaviors.[78] They have proposed two-step clustering method based on PCA to label the steady state modes and transitions. Although this method introduces a general approach for mode identification, determining several tuning parameters is required, and it has high complexity to be applied in real applications.[77]

Multiple PCA based approach is proposed by Zhao et al.[87] and Yao and his coworker[88] for labeling the modes. In these methods, several sub-PCA models are built, and the membership indices are defined with respect to each sub model that leads to identification of operating modes. The transitions are formulated based on weighted average of steady state modes that may not be able to describe the transitions behavior presicely. Another PCA based method is proposed by Tan et al.[90] that starts with several sub-PCA models, and determines the operating modes based on the similarity index between different sub-PCA loading matrices. Transitions are seperated into several segments that are described by a sub-PCA model. Since transitions usually have dynamic behavior, a sub PCA model may not result in an acceptable estimation of their behaviour.

Recently Quinones-Grueiro et al.[77] have proposed an offline mode identification method based on a local kernel density estimation for monitoring application. This method is based on density based clustering (DENCLUE) method and is modified to integrate the process sequence information to improve the accuracy. Readers can refer to[91][92] for a detailed description of DENCLUE approach. In this thesis, the proposed offline mode identification by Quinones-Grueiro et al.[77] is adopted and improved for optimality assessment. This algorithm provides an efficient procedure for operating modes labeling without the requirement of knowing the number of modes as a prior knowledge.

4.4.1.1 Scheduling Variables

In this chapter, it is assumed that the operating mode changes are due to supervised production demands or process condition changes. In other words, some specific variables are responsible for the operating mode changes that are known in each considered process. These variables have a critical role in determining operating modes and are called scheduling variables. For the mode identification purposes, these variables are considered for the analysis.

4.4.1.2 Offline Mode Identification

Suppose the dataset for the mode identification that includes scheduling variables only is $X_s \in R^{n \times s}$ where n is the number of samples and s is the number of scheduling variables. In the first step, dataset is segmented into different windows with the length of T. The segmented window length can be selected[77] based on the process dynamics. However, in this chapter its value is suggested as TS_{min} that is the minimum period of a steady state mode. In other words, a steady state section of the process can be assumed as an operating mode if it lasts at least for a minimum period. This minimum period is called dwell time and shown as TS_{min},[78] and its specific value for each process can be provided by the process engineers.

After dataset is segmented, for the data points in each window the density function is evaluated as follows:

$$\hat{f}(x) = \frac{1}{nh_x} \sum_{i=1}^{n} K(\frac{x - x_i}{h_x}) \tag{4.31}$$

where K represents the kernel function that should be non-negative, symmetric, with the constraint of $\int K(x)dx = 1$.[93] Gaussian kernel is utilized since it can provide a smooth transition. Assume z is a s dimensional data point:

$$K(z) = \frac{1}{(2\pi)^{s/2}} \exp\Big(-\frac{z^T z}{2}\Big) \tag{4.32}$$

Considering $z = \dfrac{x - x_i}{h_x}$, one can obtain:

$$K(\frac{x - x_i}{h_x}) = \frac{1}{(2\pi)^{s/2}} \exp\Big(-\frac{(x - x_i)^T(x - x_i)}{2h_x^2}\Big) \tag{4.33}$$

where h_x represents the bandwidth parameter which is defined based on the nearest neighbor density approach.[93] In other words, h presents the smoothness of the density estimation. In the original formulation of the kernel density estimation, h value is fixed for all data points while the nearest neighbor density approach determines a specific h value for each x based on its distance to the k^{th} nearest neighbor:[77]

$$h_x = dist(x, x_{k^{th}}) \tag{4.34}$$

where $x_{k^{th}}$ is the k^{th} nearest neighbor of x. In this approach k has a great influence on the estimated distributions. In other words, large values for k may lead to detecting fewer operating modes while low values may result in detecting more operating modes. As a result, a range of minimum and maximum values (maximum k is suggested as $TS_{min}/2$[77]) for k is considered, and the procedure is repeated until the number of estimated modes become constant, and then the smallest k with the constant number of modes is selected as the final k value. If it does not happen, the best value for k is selected based on the cluster validation indices[77][94] .

Based on the defined density function, the corresponding density attractors of each data point x^* that are the local maxima of the density function for each data point are computed[91][92] . The density attractors are computed using an iterative procedure that is called *Fast Hill Climbing* and is derived by setting the first derivatives of $\hat{f}(x)$ to zero[95] :

$$x^{(l+1)} = \frac{\sum_{t=1}^{n} K(\frac{x^{(l)} - x_t}{h})x_t}{\sum_{t=1}^{n} K(\frac{x^{(l)} - x_t}{h})} \tag{4.35}$$

where $x^{(l)}$ is the current iteration and $x^{(l+1)}$ is the updated value. For each data point, the procedure starts with $x^{(l)} = x$ and continues until $[\hat{f}(x^{(l)}) - \hat{f}(x^{(l-1)})]/\hat{f}(x^{(l)}) < \epsilon$.[95]

If all the data points in a window converge to the same density attractor, the window is considered as steady state. Otherwise, it is assumed as a transition part. Note that the length of each window is selected as TS_{min}, therefore each window can represent a steady state mode.

In order to detect the same steady state modes, it is proposed by Quinones-Grueiro et al.[77] to compare the summation of distance between two density attractors of the neighboring windows and the distance of the two density attractors with each other. Assume the distance of each density attractor x_{at} with its k^{th} neighbor $x_{k^{th}}$ is $dist(x_{at}, x_{k^{th}})$, the criterion to integrate two windows i and j is as follows:[77]

$$dist(x_{at_i}, x_{at_j}) \leq dist(x_{at_i}, x_{k_i^{th}}) + dist(x_{at_j}, x_{k_j^{th}}) \tag{4.36}$$

where k^{th} is considered to have the same value as k in Equation 4.34.[77] In this application, since the aim is to find the labels of each data point, in addition to the neighbors, all the initial and final windows of steady state modes are compared to each other based on the criterion in Equation 4.36 to detect final steady state modes, transitions, and noise. To illsutrate, consider the example provided in Figure 4.2. It shows a time series data that is segmented into 10 windows with the length of TS_{min}. In each window, the density attractors are computed, and the steady state and non-steady state parts are detected based on the density attractors variation. When each window is studied separately, their relationship is investigated based on Equation 4.36. In this example, windows (1,2), (4,5,6) and (8,9,10) are compared, and since they obey the Equation 4.36, they are considered as the same steady state modes. In the subsequent step, windows 2 and 4 are compared. It is assumed that windows 2 and 4 obey the Equation 4.36, and they belong to the same steady state mode. Therefore, it results in considering window 3 as noise instead of transition between two different modes. In the final step, windows 6 and 8 are compared. Since they do not belong to the same steady state mode, window 7 is a transition between two different modes. As a result, this time series consists of 2 steady state modes (red and green boxes) and one transition.

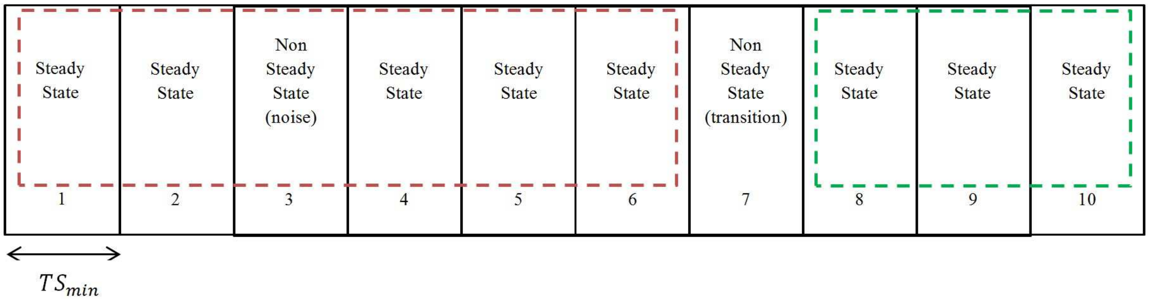

Figure 4.2: *Offline mode identification example*

In order to find the exact start and end time of the transitions, each transition part is segmented into shorter length windows to investigate the dynamics more clearly. The length T_{tr} is assumed to be 2-3 times of the number of process variables, and is known as the minimum transition mode length.[90] To illustrate, window 7 in Figure 4.2 is segmented into 7 sub windows that is shown in Figure 4.3. Similar to the above mentioned mode identification process, the density attractors with the new window size is estimated, and the transitions and steady states are detected. In this example, window 7_1 belongs to the steady state mode (red box), and windows 7_6 and 7_7 belong to the other steady state mode (green box). Final transition consists of windows 7_2, 7_3, 7_4 and 7_5.

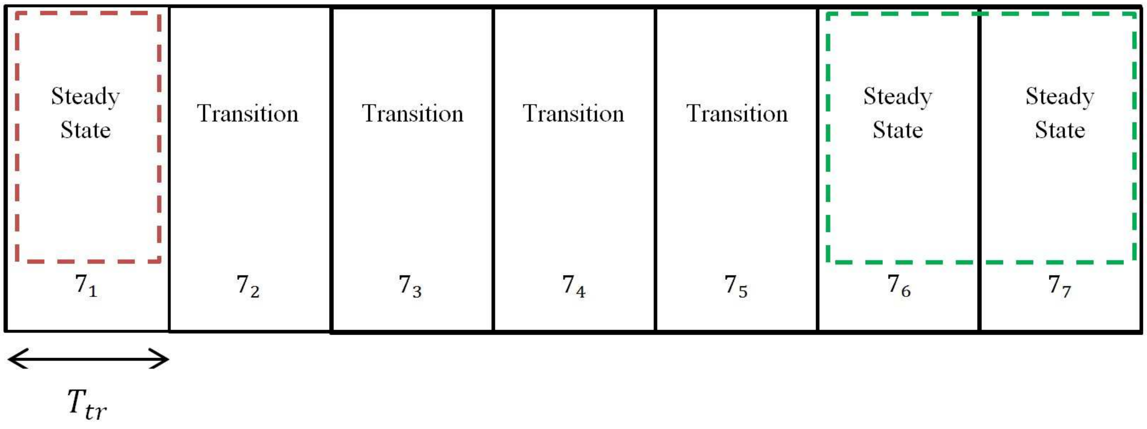

Figure 4.3: *Transition mode identification example: window 7*

The overal offline mode identification algorithm is described in Figure 4.4.

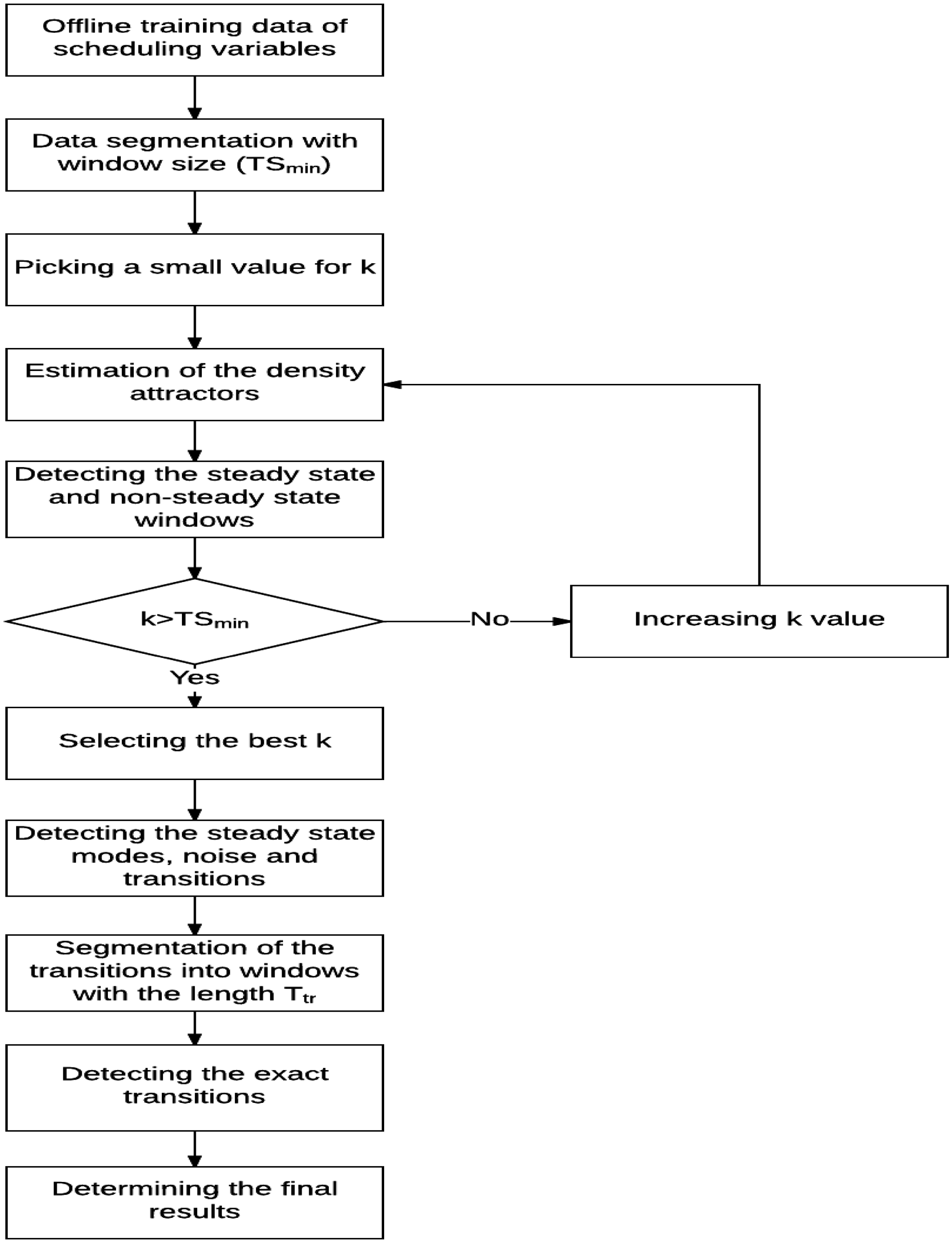
Offline training data of scheduling variables
Data segmentation with window size (TSmin)
Picking a small value for k
Estimation of the density attractors
Detecting the steady state and non-steady state windows
k>TSmin
No
Increasing k value
Yes
Selecting the best k
Detecting the steady state modes, noise and transitions
Segmentation of the transitions into windows with the length Ttr
Detecting the exact transitions
Determining the final results

Figure 4.4: *An overview of the offline mode identification*

4.4.2 Online Mode Detection

4.4.2.1 Building Predictive Classifier

In online assessment, the corresponding operating mode of each new data point is estimated to select the proper modeling and analysis approach. As a result, a predictive classifier is built based on the estimated lables in the offline mode identification step. There are a number of classification methods such as decision tree classifiers,[96] Bayesian classifiers,[97] support vector machines,[98] discriminant analysis[99][100] and so on that can be selected based on the application. In this thesis, mixture discriminant analysis (MDA) method that is the extension of discriminant analysis (DA) method is investigated. In DA method, each class is considered as a Gaussian distribution, and the covariance is considered as the same for different classes. This results in representing each class by its centre and applying the classification based on the closest centre.[101] However, in many problems, one Gaussian distribution cannot represent a class perfectly. As a result, it has been extended to MDA that considers a mixture Gaussian distribition for each class. This method has some benefits including modeling non-gaussian classes, building non-linear boundaries between classes and so on.

Let us assume $X_s \in R^{N \times s}$ is the data points based on scheduling variables where N is the number of data points, and s is the number of scheduling variables. Consider $X_{s,new}$ contains K classes, and each class k follows a mixture Gaussian distribution with R_k components. As a result, the probability of X_s given each operating mode k is as follows:[101]

$$p(X_s \mid G = k) = \sum_{r=1}^{R_k} \pi_{kr} p(X_s \mid \mu_{kr}, \Sigma) \tag{4.37}$$

where π_{kr} is the mixing proportions of each component with the constraint of $\sum_{r=1}^{R_k} \pi_{kr} = 1$. Similar to DA, in MDA it is also assumed that for all mixture compoenents the covariance matrices are the same as Σ. Therefore, the posterior probability of each class k is as follows:

$$p(G = k \mid X) = \frac{\sum_{r=1}^{R_k} \pi_{kr} p(X \mid \mu_{kr}, \Sigma)\Pi_k}{\sum_{l=1}^{K} \sum_{r=1}^{R_l} \pi_{lr} p(X \mid \mu_{lr}, \Sigma)\Pi_l} \tag{4.38}$$

Finally the parameters are estimated using the EM algorithm.[101]

The above mentioned MDA method is associated with two limitations. First, the number of Gaussian components is assumed to be known as a prior knowledge. Second, the covariance of all components is assumed to be the same that may decrease the flexibility of model building. Fraley and Raftery[32] have integrated MDA with model based clustering (Mclust) that is called MclustDA. In this approach, the above mentioned limitations are addressed. To address the first limitation, the covariance matrix of each component Σ_{kr} is parametrized through eigenvalue decomposition as follows:[102]

$$\Sigma_{kr} = \lambda_{kr} D_{kr} A_{kr} D_{kr}^T \tag{4.39}$$

where D_{kr} presents the matrix of eigenvectors, A_{kr} is a diagonal matrix of the values proportional to the eigenvalues and λ_{kr} is the corresponding constant.

The second limitation is tackled by considering a range of possible number of mixtures of components, and selecting the best parsimonious model based on BIC criterion. The overal procedure is as follows:[32]

1. A maximum number of components in each class, and a set of mixture model candidates (based on the covariance matrix form) are selected.

2. For each class hierarchical agglomeration is applied to find the approximate classification up to the maximum number of components.

3. The EM algorithm is applied on the MclustDA problem for each model candidate and each number of clusters. The initial value for the EM algorithm is the results of the step 2.

4. The best model structure and number of components is selected based on the BIC criterion. The MCLUST package exists in R programming language, and it is used in this thesis for simulation studies.[103]

To summarize, after offline mode identification, the classification model of each operating mode including steady state modes and transition grades is built based on the MclustDA method. Note that each transition has different grades that is considered as a class and is modeled by a mixture Gaussian distribution.

4.4.2.2 Online Mode Estimation

For online prediction of the operating modes, the process knowledge is incorporrated to increase the accuracy of the prediction.[75] In other words, instead of computing posterior distributions of all operating modes for each data point, the posterior of related operating modes are computed as follows:
1. If the current operating mode of the data point is steady state mode i, for the next point, the posterior probability of mode i, and all the transitions from mode i are computed.
2. If the current data point is in the grade p of transition ij, i.e., $\{ij\}_p$, the posterior probability of $\{ij\}_p$ and steady state mode j are computed.

Finaly the data point is classified to the operating mode with the highest posterior probability. Note that considering each single data point may lead to incorrect solution in noisy environments. In this case, it is suggested to evaluate a window of the data points that provides more robust estimation of the operating mode change.

The schematic diagram of the proposed method for operating optimality assessment is given in Figure 4.5.

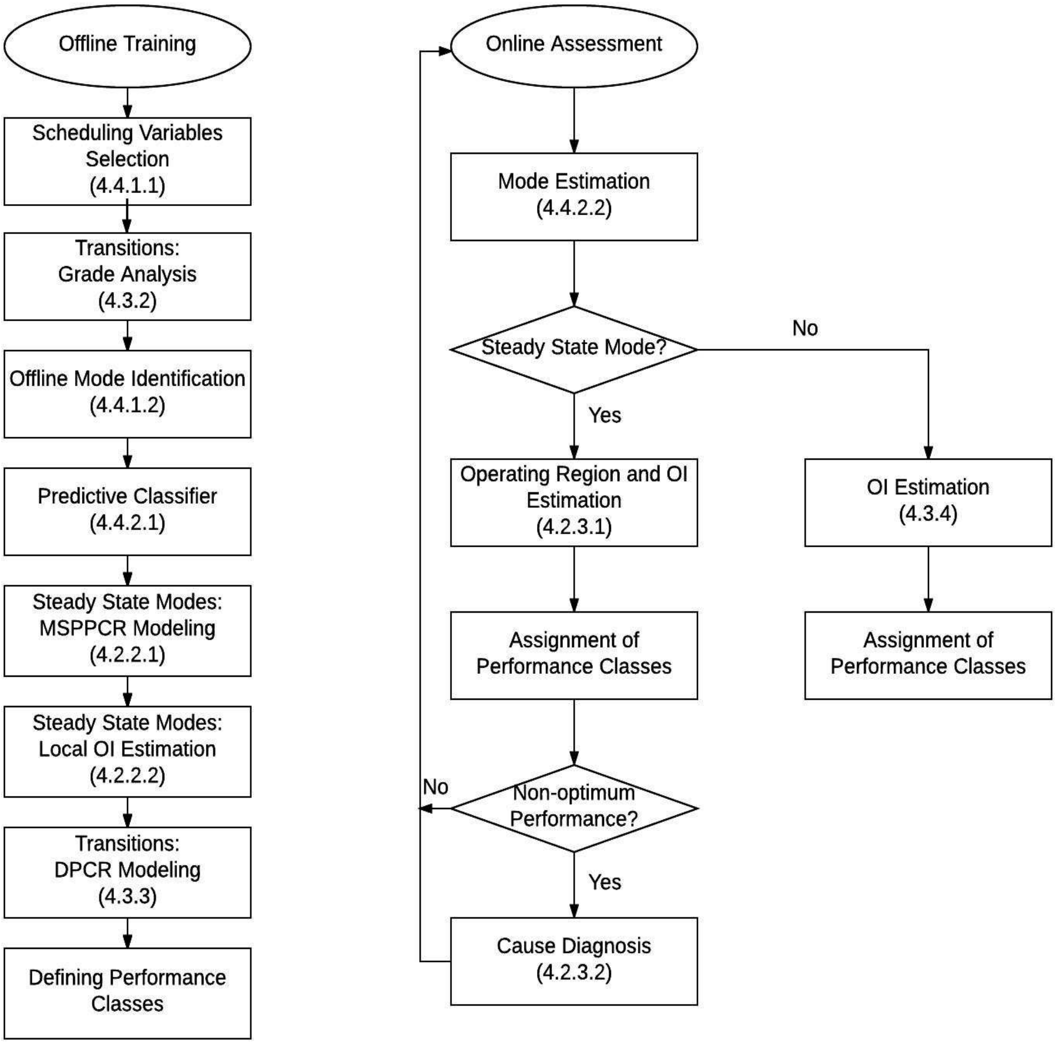
Offline Training
Scheduling Variables Selection (4.4.1.1)
Transitions: Grade Analysis (4.3.2)
Offline Mode Identification (4.4.1.2)
Predictive Classifier (4.4.2.1)
Steady State Modes: MSPPCR Modeling (4.2.2.1)
Steady State Modes: Local OI Estimation (4.2.2.2)
Transitions: DPCR Modeling (4.3.3)
Defining Performance Classes
Online Assessment
Mode Estimation (4.4.2.2)
Steady State Mode?
No
Yes
Operating Region and OI Estimation (4.2.3.1)
OI Estimation (4.3.4)
Assignment of Performance Classes
Assignment of Performance Classes
Non-optimum Performance?
No
Yes
Cause Diagnosis (4.2.3.2)

Figure 4.5: *Schematic diagram of the proposed method for operating optimality assessment*

4.5 Tennessee Eastman benchmark process

TE benchmark process has been broadly used for evaluation of many methods in process control, soft sensor design, monitoring, etc. The model was first developed based on the industrial process of TE chemical company by Downs and Vogel.[26] The process contains five major units: a reactor, a product condenser, a vapor-liquid separator, a recycle compressor and a product stripper. The function of TE process is to manufacture two main products that are G and H, from four reactants A, C, D and E. However, F might be produced under non-ideal situations as a by-product. All the reactions are exotehrmic and irreversible. The products including main products and by-products are in liquid phase, and the reactants are in gaseous phase. The process has 12 manipulated variables, 41 measured variables in which 22 process variables are measured continuously, and 19 components variables are measured at a slow rate. There are six operating modes based on the three different G/H mass ratios and production rates.[26] Figure 4.6 presents the schematic of the process. In order to have stable process, the decentralized control strategy is applied on the open loop process that was developed by Ricker.[27]

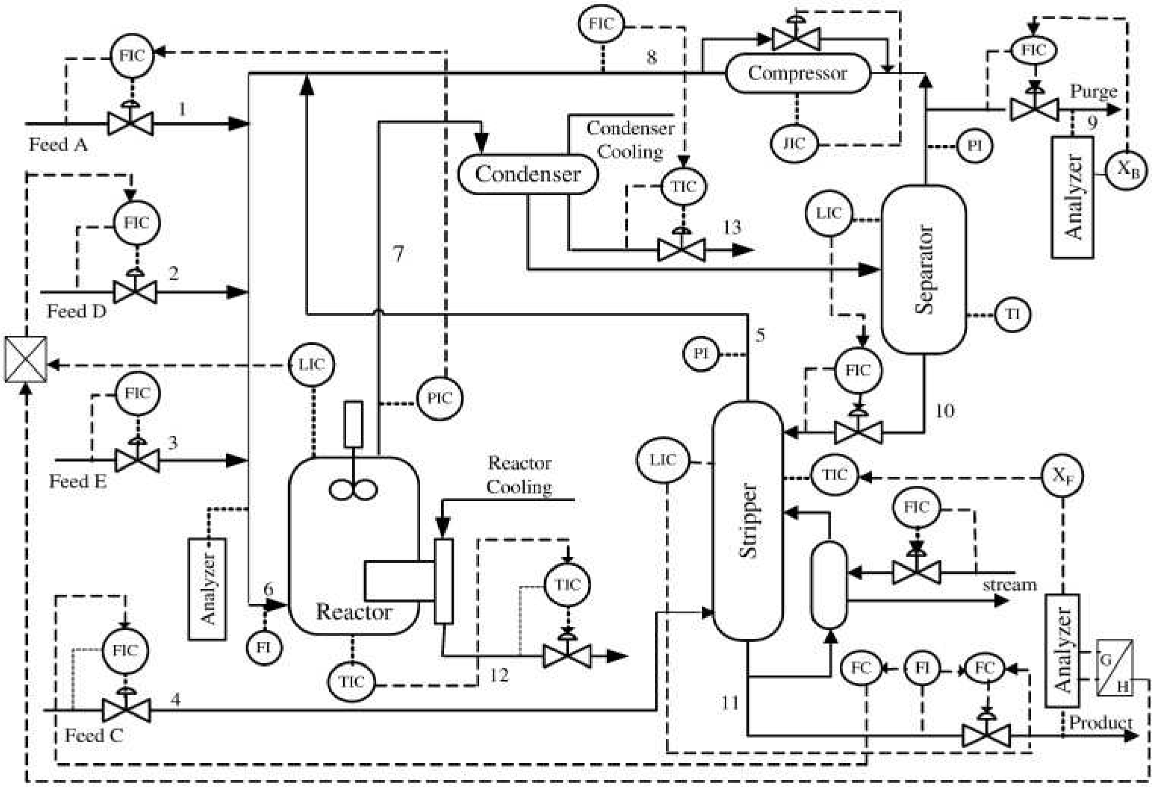

Figure 4.6: *The schematic diagram of Tennessee Eastman process*[22]

To evaluate the performance of the proposed method for optimality assessment of multi-mode processes, three different operating modes with G/H mass ratio of the 50/50, 10/90 and 40/60 are simulated. Corresponding to the change of operating mode, product component ratio (G/H), set point of the level and temperature of the reactor are changed. The selected set points are shown in Table 4.1. In addition, in each operating mode two uncertainities are added that result in having several operating regions in the vicinity of the base operating mode. The uncertainties are stated in Table 4.2.

The operation cost is selected as OI that is defined according to Ref[26] as follows: Total operation cost= (purge cost)(purge rate)+(product steam costs)(product rate) +(compressor costs)(compressor work)+(steam costs)(steam rate)

Table 4.1: *Properties of stable operating modes*

Stable mode	G/H mass ratio	Reactor level (%)	Reactor temperature (°C)
1	50/50	65	122.9
2	10/90	50	130
3	40/60	55	135

Table 4.2: *process uncertainties*

	Process variable	Type
1	B composition (stream 4)	Step
2	Reactor pressure	Step

Offline Training

To apply the proposed optimality assessmnet approach, OI, and 22 commonly measured variables are collected. The process is simulated for 2260 and 970 hours with the sampling period of 0.1 hour for offline training, and online assessment, respectively. The process contains three main steady state operating modes by adjusting the main operating variables stated in Table 4.1, and in each operating mode there are three operating regions caused by uncertainties stated in Table 4.2. The dataset contains all steady state modes and transitions. In order to evaluate transition grades, two trajectories are simulated for transitions between mode 1 and 3. The first one is changing set points directly, and the second one is changing set points twice. In other words, set points change from initial value to the middle value, and then change to the final value. In addition, since in real application, input and output variables may contain missing values, it is assumed that 10 percent of the offline training data set, including inputs and output, is missing completely at random.

The scheduling variables are selected as reactor level, and reactor temperature since they are measured continuously and are critical variables in operating mode change. After that, offline mode identification is applied to find the labels of the dataset. The window size that equals the minimum period of a steady state mode TS_{min} is selected as 100. In addition, clustering is repeated for k values between 5 to 50, and finally the results for k from 20 to 50 became consistent. As a result, 20 is selected for k value. The detected labels and the true labels are compared, and the ARI and FM index values are computed as 0.9922 and 0.9949, respectively. This result indicates a high agreement between the detected and true labels. In the subsequent step, transition grade analysis is applied on the detected transitions to detect transition grades. Trend deviation θ_T is selected as 0.95. The analysis results in two transition grades from operating mode 3 to 1 and one grade from operating 1 to 3. Note that for other transitions such as from operating mode 2 to 3, one trajectory is simulated, so transition analysis is not required. The offline training data projected into two variables of the A and C feed (stream 4) and recycle flow (stream 8) are shown in Figure 4.7. To clarify, the approximated boundary of each operating mode is shown in the figure. After that, based on the detected labels, predictive classifier is built based on the scheduling variables.

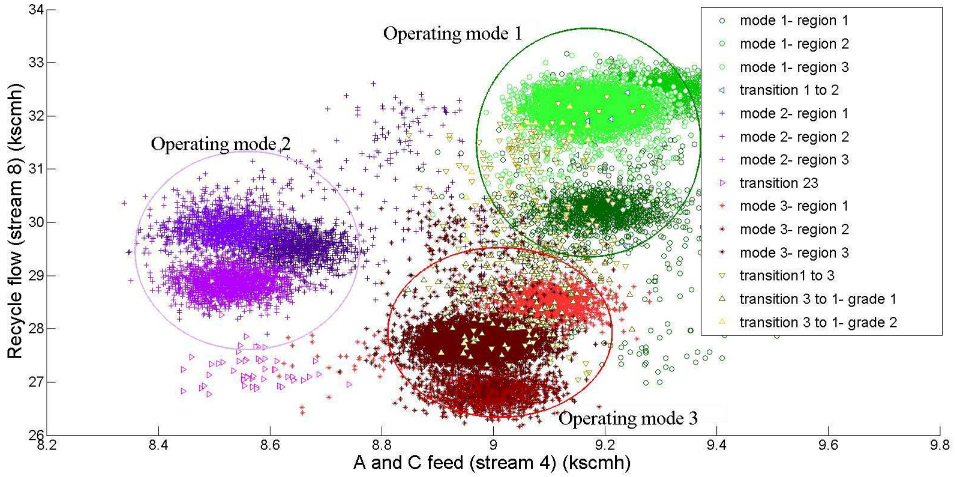

Figure 4.7: *Two dimensional plot of the offline training data*

In the next step, the number of operating regions in each steady state operating mode is estimated based on the developed hierarchical MPPCR method. As a result, three operating regions are estimated in each operating mode that is the same as the true values. After that, the developed MSPPCR model with the missing inputs is applied to build the predictive model of the steady state operating modes. When the model is constructed, the local OI value of each operating region is computed. The defined levels for the OI values are stated in Table 4.3. Note that optimality levels are defined as worse with a larger number of level. Based on the defined level and the local OI values for each operating mode, corresponding performance levels for each mode are found. The local OI values and levels for each operating mode are given in Table 4.4. Based on the obtained levels, the mode 1 has the most optimal performance in comparison with the operating modes 2, and 3. In addition, the most optimal region is the operating region 3 of the mode 1, and the least optimal region is the operating region 1 of the mode 3. In addition, transition models are built based on DPCR method.

Table 4.3: *Defined OI levels*

OI range ($\$hr^{-1}$)	Optimality level
100-140	1
141-180	2
181-220	3
Above 221	4

Table 4.4: *Local OI levels*

	mode 1		mode 2		mode 3	
Operating region	Local OI ($\$hr^{-1}$)	Optimality level	Local OI ($\$hr^{-1}$)	Optimality level	Local OI ($\$hr^{-1}$)	Optimality level
1	142.96	2	256.67	4	309.71	4
2	179.57	2	186.22	3	275.07	4
3	119.96	1	205.61	3	256.44	4

Online Assessment

In online assessment, the operating modes are predicted based on the predictive classifier built in offline training. The computed clasification error for online mode detection that is the number of incorrect estimated devided by the total number of assessment data points is 0.0079, which indicates a high accuracy of mode detection. Based on the estimated modes, the OI values are predicted using the constructed models including DPCR and MSPPCR in offline training. The comparison plot of predicted and real values for the OI is given in Figure 4.8. Since the emplyed models vary along the process, the corresponding models are stated in the figure. In addition, RMSE and R^2 values are computed as 0.3678 and 0.8588, respectively. Based on the results, the constructed model has a high accuracy in predictiong the OI values, i.e. it has high values for R^2 test and low values for RMSE. The offline mode identification, online mode detection and prediction results are summarized in Table 4.5.

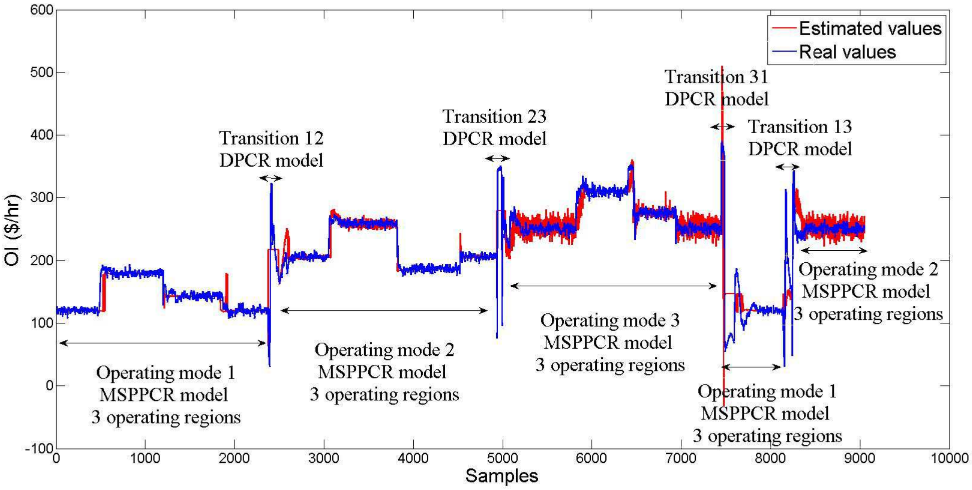

Figure 4.8: *Comparison of predicted and real values of OI*

Table 4.5: *Summary of the results*

	Results
Offline mode identification	ARI=0.9922, FM index=0.9949
Online mode detection	Classification error=0.0079
Prediction	RMSE=0.3670, R^2=0.8588

The optimality levels can be detected based on the criteria stated in Table 4.3. The estimated optimality levels are shown in Figure 4.9. According to Figure 4.8, the process starts with optimum opetration and then jumps to the level 2 of optimality. The 1383th sampling data point is selected to find the cause of non-optimum performance. Based on the previous estimations, this data point belongs to operating region 3 of operating mode 1. Based on Table 4.4, operating region 1 has the lowest OI level in mode 1, therefore it is selected as the reference mode for non-optimum cause detection. The distance of this data point from the reference mode is 218.29 that is greater than $\chi^2_{22}(0.95)$ ($\chi^2_{22}(0.95)$= 33.924). Based on the method described in section 4.2.3.2, the causes of non-optimum performance are detected. After applying this method, 10 causal variables are found that are stated in Table 6. They are ordered from the most to least effective causes by using the numbers of 1 to 10, respectively. Their contribution percentage is also shown in Figure 4.10. When these 10 variables are assumed to be missing, the distance from the reference mode becomes 28.68 that is less than $\chi^2_{22}(0.95)$ that indicates the process is steered to the optimum performance.

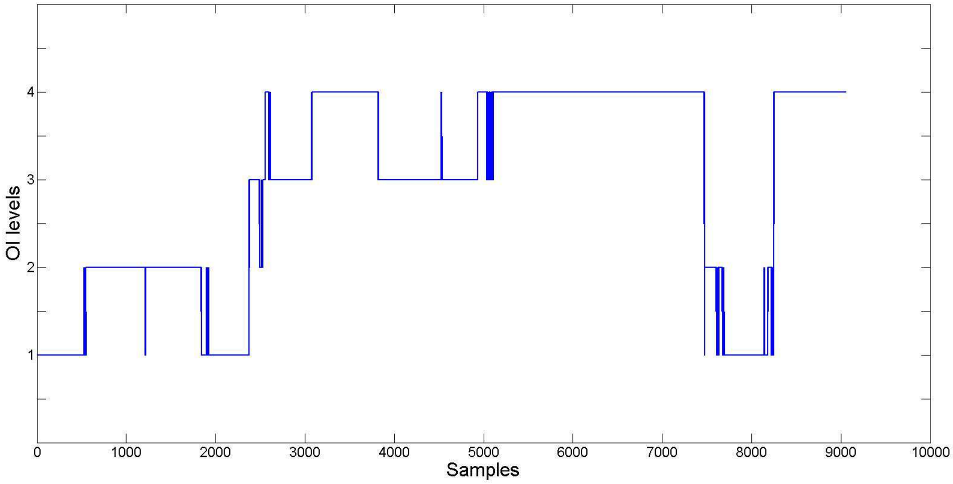

Figure 4.9: *Estimated OI levels*

The new data point is originated from the third operating region of the operating mode 1, and in this operating region reactor pressure is changed as an uncertainty. Non-optimum cause identification results has also expressed the variables that have strong relationships with the reactor pressure. The rate constant of the reaction and as a result reaction rate is dependent on the temperature that is coupled with pressure based on Arrhenius' equation.[74] Therefore, when the reactor pressure deviates from the optimal set point, the ratio of the reactants and products flowing to other subprocesses deviates from the reference operating region. Subsequently, there will be some adjustments in the next subprocesses due to the happened changes that are indicated in stripper pressure, product seperator pressure, product seperator temperature, and seperator cooling water outlet temperature. Consequently, due to the change of the amount of product and reactants flowing in the process, reactor feed rate, recycle flow, compressor work, D feed and stripper level will be affected. This procedure of causality detection gives insight of what variables are coupled with each other in the process and how to make changes to return the process to the optimum region.

Table 4.6: *Non-optimality cause variables at sample 1383*

Order	Variable number	Variable name
1	7	Reactor pressure
2	16	Stripper pressure
3	13	Product seperator pressure
4	6	Reactor feed rate
5	5	Recycle flow
6	20	Compressor work
7	11	Product seperator temperature
8	22	Seperator cooling water outlet temperature
9	2	D feed
10	15	Stripper level

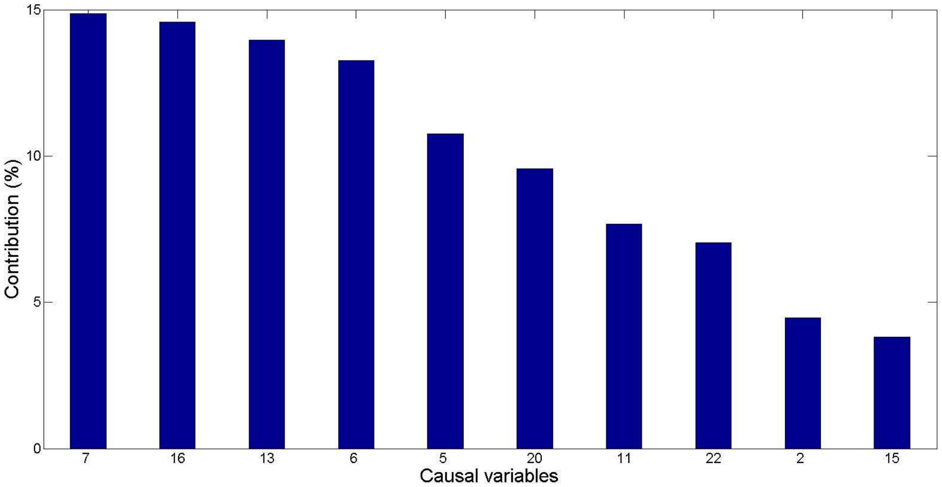

Figure 4.10: *Contribution percentage of the causal variables at sample 1383*

4.6 Conclusion

In this chapter, a novel framework for operating optimality assessment and non-optimum cause diagnosis is proposed. The operating modes are detected using improved DENCLUE based method, and the MclustDA method is employed for predictive mode classifier building. In each steady state mode, MSPPCR model is used for OI analysis in each operating region as well as predictive model building. In transitions, DPCR model is employed for transition grade analysis and buildig predictive model. In online assessment, operating mode of the new data point is estimated, and the optimality is assessed. Adopted probabilistic framework through SFFS is employed for diagnosing causal variables of the non-optimum performance. TE benchmark process is presented that has confirmed the applicability of the proposed method.

Chapter 5

Conclusions

In this chapter, the conclusions inferred from the chapters of the thesis are discussed, and some recommendations for future research in this area are given.

5.1 Summary

The main focus of this thesis is introducing a general framework for optimality assessment in multi-mode systems with non-Gaussian behavior utilizing probabilistc principal component regression (PPCR) model. The main contributions of this thesis can be summarized as:

Chapter 1 provides the motivation and challenges in optimality assessment of industrial processes. It also provides a review of the outline and contributions of each chapter of the thesis.

In chapter 2 high dimensionality of the process datasets as well as their multimodal feature is addressed by introducing mixture PPCR (MPPCR) framework. Commonly, in industrial processes, outputs have slower rate of measurement compared to inputs, which results in missing output values for the corresponding inputs. In addition, due to sensor failure or delays in analysis of some variables, inputs also contain missing values. In order to overcome the mentioned problems, the mixture semi-supervised PPCR (MSPPCR) model is extended under the expectation maximization (EM) framework in order to improve the efficiency of predictive model building in the case of simultaneous missing inputs and outputs completely at random in datasets. The main significance of this chapter is the utilization of all the available information of the dataset to construct the predictive model. The developed model is applied on numerical and simulation examples and also industrial sulfur content in naphta hydrotreater dataset, and all have confirmed the improved performance of the proposed model.

Since different operating regions in an operating mode are caused by uncertainties, a prior knowledge for the number of operating regions does not exist. In chapter 3 MPPCR model is extended under divisive hierarchical framework in order to estimate the number of operating regions in non-Gaussian stable operating modes. Maximum a posteriori (MAP) estimation under EM framework is utilized for parameter estimation in order to utilize prior knowledge. Compared to the traditional methods, since in the proposed method the number of mixture components is estimated by splitting and merging procedure, a prior knowledge of the possible number of components is not required. In addition, performance of the proposed method is improved because of utilizing minimum message length (MML) criterion for selection that is capable of detecting overlapped components as well as proposing the merging step for highly overlapped components in order to control splitting steps. Finally, the improved performance of the proposed method is demonstrated under a numerical example and also experimental hybrid tank system.

In chapter 4, a probabilsitic framework for operating optimality assessment and non-optimum cause diagnosis for multi-mode processes with non-Gaussian behavior is established. Density based clustering (DENCLUE) method is adopted and improved for detecting and labeling operating modes, and a modified mixture discriminant analysis (MclustDA) is utilized for building the predictive classifier. The optimality analysis and modeling in each steady state operating mode is based on MSPPCR model. In addition, due to the dynamic characteristic of transitions, dynamic principal component regression (DPCR) is selecetd for their grade analysis and predictive modeling. For non-optimum cause diagnosis a probabilistic framework through sequential forward floating search (SFFS) method is adopted. The applicability of the proposed framework is demonstrated through a simulation example that has validated the performance.

5.2 Recommendations for future work

The first chapter focus on dealing with simultaneous missing values in both inputs and outputs in MPPCR model. The missing values are considered to follows missing completely at random (MCAR) procedure, and it is solved through the EM framework. However, missing data can also be considered as missing not at random (MNAR) or missing at random (MAR) in practical problems.[104] In other words, for future work three types of missing data can be incorporated based on a prior knwoledge.

Divisive hierarchical approach with merging step is proposed in chapter 3 in order to estimate the number of mixture components. In this approach the number of offspring is assumed to have a user defined value, and in this thesis it is selected to have a fixed value of 2. Christopher M. Bishop and Michael E. Tipping[38] introduced a visulaization approach to determine the proper number of components in each step. However, it has some limitations due to its human driven nature and restricting the number of latent variables to 2. For future research, intoducing an interactive method for selecting the number of offspring at each step can help to enhance the accuracy and reduce the computational time.

In chapter 3, a mode identification step based on scheduling variables is considered in order to seperate operating modes and perform analysis and predictive model building on each of them separately. Extending MPPCR model in order to have two type of mixture components based on the scheduling variables for operating mode detection as well as the whole dataset for operating region detection that are caused by uncertainties as a future research can significantly reduce the complexity and computational time.

Bibliography

[1] Fortuna, L.; Graziani, S.; Rizzo, A.; Xibilia, M. G. *Soft sensors for monitoring and control of industrial processes*; Springer Science & Business Media, 2007.

[2] Khatibisepehr, S.; Huang, B.; Khare, S. *Journal of Process Control* **2013**, *23*, 1575–1596.

[3] Kadlec, P.; Gabrys, B.; Strandt, S. *Computers & Chemical Engineering* **2009**, *33*, 795–814.

[4] Lin, B.; Recke, B.; Knudsen, J. K.; Jørgensen, S. B. *Computers & chemical engineering* **2007**, *31*, 419–425.

[5] Yuan, X.; Ge, Z.; Song, Z. *Chemometrics and Intelligent Laboratory Systems* **2014**, *138*, 97–109.

[6] Jolliffe, I. T. *Introduction*; Springer, 2002.

[7] Huang, S.-M.; Yang, J.-F. *Signal Processing Letters, IEEE* **2012**, *19*, 179–182.

[8] Wold, S.; Esbensen, K.; Geladi, P. *Chemometrics and intelligent laboratory systems* **1987**, *2*, 37–52.

[9] Geladi, P.; Kowalski, B. R. *Analytica chimica acta* **1986**, *185*, 1–17.

[10] Galicia, H. J.; He, Q. P.; Wang, J. *Control Engineering Practice* **2012**, *20*, 747–760.

[11] Gonzaga, J.; Meleiro, L.; Kiang, C.; Maciel Filho, R. *Computers & Chemical Engineering* **2009**, *33*, 43–49.

[12] Willis, M. J.; Montague, G. A.; Di Massimo, C.; Tham, M. T.; Morris, A. *Automatica* **1992**, *28*, 1181–1187.

[13] Yan, W.; Shao, H.; Wang, X. *Computers & Chemical Engineering* **2004**, *28*, 1489–1498.

[14] Tipping, M. E.; Bishop, C. M. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **1999**, *61*, 611–622.

[15] Tipping, M. E.; Bishop, C. M. *Neural computation* **1999**, *11*, 443–482.

[16] Ge, Z.; Song, Z. *AIChE Journal* **2010**, *56*, 2838–2849.

[17] Ge, Z.; Huang, B.; Song, Z. *AIChE Journal* **2014**, *60*, 533–545.

[18] Khatibisepehr, S.; Huang, B. *Industrial & Engineering Chemistry Research* **2008**, *47*, 8713–8723.

[19] Magnani, M. *Department of Computer Science, University of Bologna, Italy* **2004**, 1–10.

[20] McLachlan, G.; Krishnan, T. *The EM algorithm and extensions*; John Wiley & Sons, 2007; Vol. 382.

[21] Geladi, P.; Esbensen, K. *Journal of Chemometrics* **1991**, *5*, 97–111.

[22] Ge, Z.; Gao, F.; Song, Z. *Chemometrics and Intelligent Laboratory Systems* **2011**, *105*, 91–105.

[23] Walczak, B.; Massart, D. L. *Chemometrics and Intelligent Laboratory Systems* **2001**, *58*, 29–42.

[24] Chen, T.; Sun, Y. *Control Engineering Practice* **2009**, *17*, 469–477.

[25] Yu, S.; Yu, K.; Tresp, V.; Kriegel, H.-P.; Wu, M. Supervised probabilistic principal component analysis. Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. 2006; pp 464–473.

[26] Downs, J. J.; Vogel, E. F. *Computers & chemical engineering* **1993**, *17*, 245–255.

[27] Ricker, N. L. *Journal of Process Control* **1996**, *6*, 205–221.

[28] Kolmetz, K.; Jaya, A. *www.klmtechgroup.com* **2013**, *accessed: June 2016*.

[29] McLachlan, G.; Peel, D. *Finite mixture models*; John Wiley & Sons, 2004.

[30] Maitra, R. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on* **2009**, *6*, 144–157.

[31] Melnykov, V.; Melnykov, I. *Computational Statistics & Data Analysis* **2012**, *56*, 1381–1395.

[32] Fraley, C.; Raftery, A. E. *Journal of the American statistical Association* **2002**, *97*, 611–631.

[33] He, J.; Lan, M.; Tan, C.-L.; Sung, S.-Y.; Low, H.-B. Initialization of cluster refinement algorithms: A review and comparative study. Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on. 2004.

[34] Biernacki, C.; Celeux, G.; Govaert, G. *Computational Statistics & Data Analysis* **2003**, *41*, 561–575.

[35] Figueiredo, M. A.; Jain, A. K. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **2002**, *24*, 381–396.

[36] Biernacki, C.; Govaert, G. *Computing Science and Statistics* **1997**, 451–457.

[37] Zhao, J. *Cybernetics, IEEE Transactions on* **2014**, *44*, 1871–1883.

[38] Bishop, C. M.; Tipping, M. E. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **1998**, *20*, 281–293.

[39] Su, T.; Dy, J. G. Automated hierarchical mixtures of probabilistic principal component analyzers. Proceedings of the twenty-first international conference on Machine learning. 2004; p 98.

[40] Han, J.; Kamber, M.; Pei, J. *Data mining: concepts and techniques*; Elsevier, 2011.

[41] Roberts, S. J.; Husmeier, D.; Rezek, I.; Penny, W. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **1998**, *20*, 1133–1142.

[42] Windham, M. P.; Cutler, A. *Journal of the American Statistical Association* **1992**, *87*, 1188–1192.

[43] others,, et al. Variational Inference for Bayesian Mixtures of Factor Analysers. NIPS. 1999; pp 449–455.

[44] Bensmail, H.; Celeux, G.; Raftery, A. E.; Robert, C. P. *Statistics and Computing* **1997**, *7*, 1–10.

[45] Richardson, S.; Green, P. J. *Journal of the Royal Statistical Society: series B (statistical methodology)* **1997**, *59*, 731–792.

[46] Ye, M.; Meyer, P. D.; Neuman, S. P. *Water Resources Research* **2008**, *44*.

[47] Wallace, C. S.; Freeman, P. R. *Journal of the Royal Statistical Society. Series B (Methodological)* **1987**, 240–265.

[48] Baxter, R. A.; Oliver, J. J. *Statistics and Computing* **2000**, *10*, 5–16.

[49] Ray, S.; Lindsay, B. G. *Annals of Statistics* **2005**, 2042–2065.

[50] Baudry, J.-P.; Raftery, A. E.; Celeux, G.; Lo, K.; Gottardo, R. *Journal of Computational and Graphical Statistics* **2012**,

[51] Hennig, C. *Advances in data analysis and classification* **2010**, *4*, 3–34.

[52] Melnykov, V. *Journal of Computational and Graphical Statistics* **2016**, *25*, 66–90.

[53] Maitra, R.; Melnykov, V. *Journal of Computational and Graphical Statistics* **2010**, *19*, 354–376.

[54] Hubert, L.; Arabie, P. *Journal of classification* **1985**, *2*, 193–218.

[55] Yeung, K. Y.; Ruzzo, W. L. *Bioinformatics* **2001**, *17*, 763–774.

[56] Wagner, S.; Wagner, D. *Comparing clusterings: an overview*; Universität Karlsruhe, Fakultät für Informatik Karlsruhe, 2007.

[57] Ye, L.; Liu, Y.; Fei, Z.; Liang, J. *Industrial & engineering chemistry research* **2009**, *48*, 10912–10923.

[58] Crowl, D. A.; Louvar, J. F. *Chemical process safety: fundamentals with applications*; Pearson Education, 2001.

[59] Rouvroye, J. L.; van den Bliek, E. G. *Reliability Engineering & System Safety* **2002**, *75*, 289–294.

[60] Khan, F. I.; Husain, T.; Abbasi, S. A. *Process Safety and Environmental Protection* **2001**, *79*, 65–80.

[61] Montella, A. *Transportation Research Record: Journal of the Transportation Research Board* **2005**, 62–72.

[62] Apostolakis, G. E. *Risk analysis* **2004**, *24*, 515–520.

[63] MacGregor, J. F.; Kourti, T. *Control Engineering Practice* **1995**, *3*, 403–414.

[64] Choi, S. W.; Park, J. H.; Lee, I.-B. *Computers & chemical engineering* **2004**, *28*, 1377–1387.

[65] Kim, D.; Lee, I.-B. *Chemometrics and intelligent laboratory systems* **2003**, *67*, 109–123.

[66] Isermann, R. *Automatica* **1984**, *20*, 387–404.

[67] Frank, P. M. *automatica* **1990**, *26*, 459–474.

[68] Biegler, L. T. *Solution of Dynamic Optimization Problems by Successive Quadratic Programming and Orthogonal Collocation.*; 1983.

[69] Goffe, W. L.; Ferrier, G. D.; Rogers, J. *Journal of econometrics* **1994**, *60*, 65–99.

[70] McCall, J. *Journal of Computational and Applied Mathematics* **2005**, *184*, 205–222.

[71] Fu, M. C. *Annals of Operations Research* **1994**, *53*, 199–247.

[72] Marler, R. T.; Arora, J. S. *Structural and multidisciplinary optimization* **2004**, *26*, 369–395.

[73] Liu, Y.; Chang, Y.; Wang, F. *Journal of Process Control* **2014**, *24*, 1548–1555.

[74] Liu, Y.; Wang, F.; Chang, Y.; Ma, R. *Chemical Engineering Research and Design* **2015**, *97*, 77–90.

[75] Liu, Y.; Wang, F.; Chang, Y.; Ma, R. *Chemical Engineering Science* **2015**, *137*, 106–118.

[76] Kariwala, V.; Odiowei, P.-E.; Cao, Y.; Chen, T. *Journal of Process Control* **2010**, *20*, 1198–1206.

[77] Quiñones-Grueiro, M.; Prieto-Moreno, A.; Llanes-Santiago, O. *Industrial & Engineering Chemistry Research* **2016**, *55*, 692–702.

[78] Srinivasan, R.; Wang, C.; Ho, W.; Lim, K. *Industrial & engineering chemistry research* **2004**, *43*, 2123–2139.

[79] Jiang, T.; Chen, B.; He, X.; Stuart, P. *Computers & chemical engineering* **2003**, *27*, 569–578.

[80] Chen, T.; Martin, E.; Montague, G. *Computational Statistics & Data Analysis* **2009**, *53*, 3706–3716.

[81] Jain, A. K.; Duin, R. P. W.; Mao, J. *IEEE Transactions on pattern analysis and machine intelligence* **2000**, *22*, 4–37.

[82] Ferri, F.; Pudil, P.; Hatef, M.; Kittler, J. *Pattern Recognition in Practice IV* **1994**, 403–413.

[83] Jain, A.; Zongker, D. *IEEE transactions on pattern analysis and machine intelligence* **1997**, *19*, 153–158.

[84] Ververidis, D.; Kotropoulos, C. *Signal Processing* **2008**, *88*, 2956–2970.

[85] Pudil, P.; Novovičová, J.; Kittler, J. *Pattern recognition letters* **1994**, *15*, 1119–1125.

[86] Yu, J. *Chemical engineering science* **2012**, *82*, 22–30.

[87] Zhao, C.; Wang, F.; Lu, N.; Jia, M. *Journal of Process Control* **2007**, *17*, 728–741.

[88] Yao, Y.; Gao, F. *Journal of Process Control* **2009**, *19*, 816–826.

[89] Ku, W.; Storer, R. H.; Georgakis, C. *Chemometrics and intelligent laboratory systems* **1995**, *30*, 179–196.

[90] Tan, S.; Wang, F.; Peng, J.; Chang, Y.; Wang, S. *Industrial & Engineering Chemistry Research* **2011**, *51*, 374–388.

[91] Hinneburg, A.; Keim, D. A. An efficient approach to clustering in large multimedia databases with noise. KDD. 1998; pp 58–65.

[92] Hinneburg, A.; Keim, D. A. *Knowledge and Information Systems* **2003**, *5*, 387–415.

[93] Zaki, M. J.; Meira Jr, W. *Data mining and analysis: fundamental concepts and algorithms*; Cambridge University Press, 2014.

[94] Halkidi, M.; Batistakis, Y.; Vazirgiannis, M. *Journal of intelligent information systems* **2001**, *17*, 107–145.

[95] Hinneburg, A.; Gabriel, H.-H. Denclue 2.0: Fast clustering based on kernel density estimation. International symposium on intelligent data analysis. 2007; pp 70–80.

[96] Safavian, S. R.; Landgrebe, D. **1990**,

[97] Cheng, J.; Greiner, R. Comparing Bayesian network classifiers. Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence. 1999; pp 101–108.

[98] others,, et al. *ISIS technical report* **1998**, *14*.

[99] Friedman, J. H. *Journal of the American statistical association* **1989**, *84*, 165–175.

[100] Hastie, T.; Buja, A.; Tibshirani, R. *The Annals of Statistics* **1995**, 73–102.

[101] Friedman, J.; Hastie, T.; Tibshirani, R. *The elements of statistical learning*; Springer series in statistics Springer, Berlin, 2001; Vol. 1.

[102] Banfield, J. D.; Raftery, A. E. *Biometrics* **1993**, 803–821.

[103] Fraley, C.; Raftery, A. E.; Scrucca, L.; Murphy, T. B.; Fop, M.; Scrucca, M. L. **2016**,

[104] Donders, A. R. T.; van der Heijden, G. J.; Stijnen, T.; Moons, K. G. *Journal of clinical epidemiology* **2006**, *59*, 1087–1091.