

**ADVANCED DATA ANALYTICS FOR DATA-RICH MULTISTAGE
MANUFACTURING PROCESSES**

A Dissertation
Presented to
The Academic Faculty

by

Andi Wang

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Industrial and Systems Engineering

Georgia Institute of Technology
May 2021

COPYRIGHT © 2021 BY ANDI WANG

ADVANCED DATA ANALYTICS FOR DATA-RICH MULTISTAGE MANUFACTURING PROCESSES

Approved by:

Dr. Jianjun Shi, Advisor
H. Milton Stewart School of Industrial
and Systems Engineering
Georgia Institute of Technology

Dr. Jing Li
H. Milton Stewart School of Industrial
and Systems Engineering
Georgia Institute of Technology

Dr. Kamran Paynabar
H. Milton Stewart School of Industrial
and Systems Engineering
Georgia Institute of Technology

Dr. Roshan Vengazhiyil Joseph
H. Milton Stewart School of
Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Yajun Mei
H. Milton Stewart School of
Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Hao Yan
School of Computing, Informatics,
and Decision Systems Engineering
Arizona State University

Date Approved:

To my family and my parents.

ACKNOWLEDGEMENTS

I would like to first express my gratitude to my advisor Dr. Jianjun Shi for his guidance. He always listens to me patiently, gives constructive comments, and continuously encourages me on my research. From him, I gained tremendous experience in developing research topics and tackling real engineering problems. More importantly, he exemplifies a mentor who cares for the students and a researcher who works with passion, dedication, calmness, and a great sense of humor. These invaluable characteristics will greatly inspire me for my future professional and personal life and make me become a better person. I am so thankful for studying under his supervision, and it is an invaluable fortune in my life.

I would also like to thank Dr. Xi Zhang, who patiently guided me and devoted great efforts to my thesis research. I would thank the committee members, Dr. Rosha Joseph, Dr. Jing Li, Dr. Yajun Mei, Dr. Kamran Paynabar, and Dr. Hao Yan, for reading this thesis and providing valuable suggestions.

I would thank the current and past members of Dr. Shi's group: Xiaowei Yue, Yuchen Wen, Xinran Shi, Zhen Zhong, Dhari Alenezi, Michael Biehler, Juan Du, Feng Wang, and Huihui Miao. I am grateful to have been with them, as we learn from each other, help each other, encourage each other, and build each other up in a collaborative environment.

Finally, I also wish to thank my parents for their continuous support and encouragement to achieve my goal. Last but not least, I thank my wife for her persistent

love and care that I am deeply indebted to and thank my daughter, who brought incredible joy to our family. This thesis is dedicated to them.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iv
LIST OF TABLES	vii
LIST OF FIGURES	ix
SUMMARY	xi
CHAPTER 1. Introduction	1
CHAPTER 2. Ranking Features to Promote Diversity: An Approach Based on Sparse Distance Correlation	5
2.1 Introduction	5
2.2 Literature review	8
2.3 Sparse distance correlation (SpaDC) ranking procedure	11
2.3.1 Distance correlation	11
2.3.2 Distance covariance based on the weighted l_1 -distance metric	13
2.3.3 Formulating the optimization problem	14
2.3.4 Feature ranking with distance correlation criteria	15
2.4 Theoretical properties and discussions	18
2.4.1 Theoretical properties	19
2.4.2 Intuitive illustrations	21
2.4.3 Discussion	24
2.5 Simulation studies	25
2.5.1 Existing benchmarks and general settings	25
2.5.2 Simulation with independent features	26
2.5.3 Simulation with dependent features	30
2.6 Case Studies	34
2.6.1 Epitaxy process in solar cell manufacturing	34
2.6.2 Lithography process in semiconductor manufacturing	37
2.7 Summary	40
CHAPTER 3. Holistic Modeling and Analysis of Multistage Manufacturing Processes with Sparse Effective Inputs and Mixed Profile Outputs	41
3.1 Introduction	41
3.2 Literature review	45
3.3 Holistic Analysis Framework for MMP generating profiles and images	47
3.3.1 Data scenario and problem description	48
3.3.2 Problem formulation	53
3.3.3 Problem solution	57
3.3.4 Convergent rate and complexity analysis	66
3.3.5 Discussion	67

3.4	Simulation studies for performance evaluation	70
3.4.1	Engineering background	70
3.4.2	Specifications of simulation settings	72
3.4.3	Optimization procedure and results of simulation studies	74
3.5	Summary	83
CHAPTER 4. Multiple Event Identification and Characterization by Retrospective Analysis of Structured Data Streams		85
4.1	Introduction	85
4.2	Literature Review	88
4.3	The Method of Multiple Event Identification and Characterization	91
4.3.1	Problem formulation	93
4.3.2	Solution algorithms	96
4.4	Simulation Studies	101
4.4.1	Simulation setup	101
4.4.2	Estimation results and evaluations	105
4.5	Case Study	110
4.6	Summary	113
CHAPTER 5. Conclusion		115
5.1	Summary of the contributions	115
5.2	Related Works and Future Works	116
Appendix A. Supplementary Materials for Chapter 2		118
A.1	The derivation of $V_{n,\beta}(X, Y)$ and $R_n^2(X, Y)$ in Section 2.3	118
A.2	Proof of the claim on zero weighed population distance correlation	119
A.3	Proof of Proposition 1	120
A.4	Problem (4) is conic quadratic programming problem	121
A.5	Proof of Proposition 2	122
A.6	Proof of Proposition 3	125
A.7	Proof of Proposition 4	127
Appendix B. Supplementary Materials for Chapter 3		130
B.1	Proofs for Proposition 6 and Proposition 7	130
B.2	The specifications of simulating images and curves in Section 3.4.2	131
B.3	Illustrations of the estimated parameters in Section 3.4.3	132
Appendix C. Supplementary Materials for Chapter 4		138
C.1	Derivation of Proposition 10	138
C.2	Proof of Proposition 11	139
REFERENCES		140

LIST OF TABLES

Table 1	Effect matrices estimated in formulation (6)	65
Table 2	The type and dimension of data from each stage	72
Table 3	Summary of the simulation settings	76
Table 4	The estimation error $d_{i,k}^2 (1 \leq i \leq k \leq 4)$ and the associated σ_{rep} and σ_{par} in brackets	79
Table 5	The estimation error of $d_{k0}^2 (1 \leq k \leq 4)$ and the associated σ_{rep} and σ_{par} in brackets	80
Table 6	Number of variation patterns and the associated σ_{rep} and σ_{par} in brackets	83
Table 7	Simulation setups	103
Table 8	The error rate of event sequence identification	108
Table 9	The values of V_{ξ} for nine setups	109

LIST OF FIGURES

Figure 1	Schematic illustration of an MMP	1
Figure 2	Illustration of the optimization problem (4).	24
Figure 3	The comparison results in setting 1.	28
Figure 4	The comparison results in setting 2.	29
Figure 5	The comparison results in setting 3.	29
Figure 6	The comparison results in (a) setting 4 and (b) setting 5.	33
Figure 7	(a) The control chart for X_1 and (b) the control chart for SCE.	35
Figure 8	(a) Distance correlations between the two features and (b) the distance correlation between each feature and its quality variable.	36
Figure 9	(a) The overlay error on one sample wafer and (b) the locations of the defects.	38
Figure 10	The ranking result of six methods.	39
Figure 11	The illustration of the model	51
Figure 12	The outputs of curves and images from four stages.	73
Figure 13	The outputs of curves and images from four stages.	75
Figure 14	An illustration of an estimation of \widehat{B}_{10} , \widehat{B}_{20} , \widehat{B}_{10} and \widehat{B}_{40} from one dataset.	77
Figure 15	The collected data $x_{t,i}(s)$'s and the strengths of the unknown underlying events $y_{k,t}$'s.	92
Figure 16	The sample paths of y^0 in initialization, corresponding to $\alpha = 1/5$, $1/10$, and $1/15$. The circles, crosses, and dots indicate three events.	101
Figure 17	The scatter plot of r_f and r_o and selected event sequences Y 's.	105

Figure 18	(a) The convergence of the BCD algorithm. (b) The convergence of each B-update and Y-update.	106
Figure 19	The true event sequence and the estimated event sequence according to Setup 1.	107
Figure 20	The estimated event signatures on ten sensors of the three events, according to Setup 1. The horizontal axis in each figure represents the measurement points in each signal.	108
Figure 21	The illustration of a rolling process, where the blue square represents the measurement plane of the laser gauge. The shape at the right side illustrates the cross-sectional shape of a rolling bar and its diameter measurements along six axes.	110
Figure 22	The illustration of the raw data for the case study.	111
Figure 23	The event signatures on six signals for event 1 (first row) and event 2 (second row).	112
Figure 24	The sequence of event 1 (solid line) and event 2 (dashed line).	112
Figure 25	The sample signals that are associated with (a) event 1 and (b) event 2.	112
Figure B.3.1	The estimated matrices $\hat{\mathbf{B}}_{i,j,k}$, for $i=1, 2, 3$ and 4 , respectively.	137

SUMMARY

Nowadays, multistage manufacturing processes (MMPs) are usually equipped with complex sensing systems. They generate data with several unique characteristics: the output quality measurements from each stage are of different types, the comprehensive set of inputs (or process variables) have distinct degrees of influence over the process, and the relationship between the inputs and outputs is sometimes ambiguous, and multiple types of faults repetitively occur to the process during its operation. These characteristics of the data lead to new challenges in the data analytics of MMPs.

In this thesis, we conduct three studies to tackle those new challenges from MMPs. In the first study, we propose a feature ranking scheme that ranks the process features based on their relationship with the final product quality. Our ranking scheme is called sparse distance correlation (SpaDC), and it satisfies the important diversity criteria from the engineering perspective and encourages the features that uniquely characterize the manufacturing process to be prioritized. The theoretical properties of SpaDC are studied. Simulations, as well as two real-case studies, are conducted to validate the method.

In the second study, we propose a holistic modeling approach for the MMPs, aiming at understanding how intermediate quality measurements of mixed profile outputs relate to sparse effective inputs. This model can identify the effective inputs, output variation patterns, and establish connections between them. Specifically, the aforementioned objective is achieved by formulating and solving an optimization problem that involves the effects of process inputs on the outputs across the entire MMP. This ADMM algorithm that

solves this problem is highly parallelizable and thus can handle a large amount of data of mixed types obtained from MMPs.

In the third study, a retrospective analysis method is proposed for multiple functional signals. This method simultaneously identifies when multiple events occur to the system and characterizes how they affect the multiple sensing signals. A problem is formulated using the dictionary learning method, and the solution is obtained by iteratively updating the event signatures and sequences using ADMM algorithms.

In the end, the potential extensions to the general interconnect systems are discussed.

CHAPTER 1. INTRODUCTION

Multistage manufacturing processes (MMPs) are complex manufacturing systems that involve multiple operations or stations to fabricate a product [1]. Typically, numerous sensing systems are installed in the MMPs to collect the data that relate to the manufacturing process, including the input variables from each stage, intermediate product quality measurements from each stage, and final quality of the product. An illustration of an MMP is shown in Figure 1.

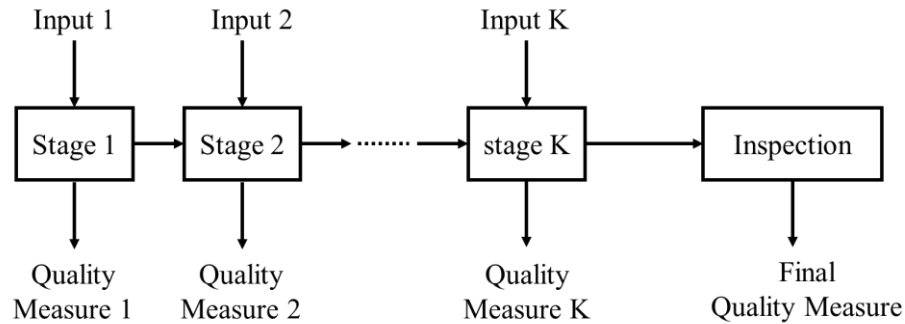


Figure 1 Schematic illustration of an MMP

In literature, the data analytics of MMP has been conducted for decades [1-3], where state-space models were proposed to describe the stream of variation (SOV) in MMPs for assembly and machining processes. However, the massive data generated from the contemporary advanced MMPs have some unique characteristics. During the authors' study, the author participated in multiple engineering projects in semiconductor manufacturing industry and metallurgical industry. Both projects involve MMPs where each manufacturing stage generates rich data of different types. Through the analysis of these data, we aim to discover the association between the process variables and quality

variables from different stages and to gain more understanding of the manufacturing system. Compared with the MMPs analyzed in existing literature, however, the MMPs in the semiconductor manufacturing processes and the steel rolling processes investigated in this thesis have several unique characteristics:

1. These MMPs generate multiple types of data, as the measurements are typically collected by distinct types of sensors. In the semiconductor manufacturing processes, the intermediate quality measurements contain different types of data. From the CVD stages, the film thicknesses are measured at multiple locations on the wafer, generating a thin-film thickness map. From the lithography stages, the overlay error between two layers is measured at many different locations, generating 2D vector fields. In the steel manufacturing processes, the sensing signals of rolling speed, the temperature of the rolling bars, and the dimensional measurements along the rolling bars have distinct properties in smoothness and variation patterns.
2. Many potential root causes may affect each stage of the manufacturing process, whereas a product quality defect (or fault) is only caused by very few root causes at a given time period of operations. For example, the position of the exposures in the lithography stage of the semiconductor manufacturing processes is influenced by tens of potential root causes associated with lens, reticle, and the wafer's locations. Sometimes, these potential root causes are measured during the process, making it possible for us to identify the variables that are related to the product quality measures and establish their relationship. Other times, the faults on the process are not observable, while we need to estimate the duration and the strengths

of the events and identify the underlying root causes based on the latent recurring signatures from the multiple process signals.

3. Unlike the multistage assembly and machining process, the semiconductor manufacturing processes typically do not have a well-defined low-dimensional state vector that determines the state of the manufacturing stages. The complex physical laws that govern the semiconductor manufacturing processes do not allow one stage of manufacturing described by a few numerical values. Meanwhile, the Markovian property in the state-space model does not hold: the quality output from a critical downstream manufacturing stage is possibly influenced by an early upstream stage that is related to it.
4. The relationship among the measurements from the MMP is sometimes ambiguous. If we do not have prior knowledge about the model, directly modeling their dependency can be intractable or even infeasible, given the large amount of data we collect from the process.

Studying the rich data collected from these MMPs provide unprecedented opportunities to understand the manufacturing process, which leads to quality improvement. In this thesis, three studies are conducted to tackle the analytical problems in MMPs.

First, we propose an automatic tool for process diagnostics: an algorithm that ranks the process features from the intermediate quality measures according to the extent of their dependent relationship with the final product quality. This problem is rooted in the requirements of practitioners from the semiconductor manufacturing industry. The developed feature ranking scheme is based on sparse distance correlation (SpaDC). It

considers the arbitrary dependent relationship between the process features and the final quality of the product, and it satisfies the important diversity criteria from the engineering perspective and encourages the features that uniquely characterize the manufacturing process to be prioritized.

The second study focuses on diagnosing and explores the linear relationship between multiple potential root causes from an MMP and mixed profile outputs. Specifically, a modeling framework is proposed to answer three interrelated questions: (i) which potential root causes from the stages are related to the variations of the outputs? (ii) what are the variation patterns of the outputs caused by these inputs? (iii) how each individual process input affects the manufacturing process? In the second study, a holistic modeling and analysis method is developed to address the above three questions simultaneously.

In the third study, a retrospective analysis method is proposed for a historical multi-functional data set, which simultaneously identifies when multiple events occur to these multi-functional data and characterizes how they affect the sensing signals. The problem formulation is motivated by the dictionary learning method, and the solution is obtained by iteratively updating the event signatures and sequences using ADMM algorithms. A simulation study and a case study of the steel rolling process validate our approach.

Those three studies summarized above will be introduced in Chapter 2, Chapter 3, and Chapter 4, respectively. These studies provide some useful modeling approaches for a class of MMPs that appear in advanced manufacturing applications. Chapter 5 summarizes the thesis, and several future research areas are pointed out.

CHAPTER 2. RANKING FEATURES TO PROMOTE DIVERSITY: AN APPROACH BASED ON SPARSE DISTANCE CORRELATION

2.1 Introduction

A key task of quality engineering is to identify the root causes that drive the variation of the product quality. In traditional statistical quality control, the identification of the major and minor factors in personnel, machines, materials, methods, and environments is mainly based on experiential knowledge. Fishbone diagrams and Pareto charts [4, 5] have been widely used as standard methods to illustrate the leading root causes and thus focus limited quality improvement budgets on a few quality problems.

Nowadays, advanced sensing technologies are widely utilized in manufacturing processes and generate a large amount of process data from system components. By retrospective analysis of the dependent relationship between the product quality variable and the process features obtained from the manufacturing processes, we are interested in *automatically* ranking and identifying the potential factors that affect the product quality.

The automatic root cause analysis approach shall be stipulated by understanding inherent characteristics of the process features. First, as many sensors are installed in the entire production line, the number of total process features is usually large. However, there are typically limited root causes among all the potential root causes that lead to the process faults and disturbance in a period of time, and each affects multiple sensors measuring different physical variables at the same time, resulting in dependency and redundancy

among the process features [5]. Furthermore, a specific product quality issue only involves a few disturbances, and thus many process features may be weakly, or even not related to the quality variable. Finally, the dependency relationship between the quality variable and the process features, and the dependency relationship among the process features themselves, are complex and ambiguous: they may be nonlinearly related, and certain features may relate to the variance of the quality variable instead of its mean. In summary, *many* process features can be collected from a production process, but some of them are strongly dependent as they are driven by *fewer* root causes, and only *a few* features relate to the quality variable through *complex* relationships.

To achieve root cause analysis, the above characteristics of the process data motivate the practitioners to rank the features based on the dependency relationship with the quality variable. Examples can be found in literature [6, 7]. Compared with the root cause analysis procedure based on building and analyzing predictive models [8], feature ranking can be used on processes with a larger size of features and complex dependency relationship between process features and quality variables. Although more advanced predictive methods, such as ensemble methods, can be applied, they are all based on predetermined algorithms, and thus requires reconfiguration once the process changes. Furthermore, predictive models cannot not capture certain relationship between process features and quality variables, for example, when the variance of the quality variable is dependent with the process features.

The above characteristics of the data further stipulate two specific requirements for the feature ranking procedure. First, the ranking should be based on *general* dependency, given the complex and ambiguous relationship between process features and quality

variable. This general dependency measure shall take all potential dependency relationships between process features and quality variables into accounts – including the nonlinear relationships between process features and the quality variables, as well as the association between the process features and the variance of the quality variable. Second, since many process features are associated with few root causes, the ranking procedure shall satisfy the *diversity rule* – a process feature shall be prioritized if it is not correlated to other features already deemed to be strongly related to the quality variable. With the diversity rule, only one feature within a bunch of dependent features according to each root cause is selected, and thereby encourages a small number of leading features to cover all potential root causes that relate to the quality variable. If the diversity rule is not satisfied, the highly ranked features will all relate to the prime root cause; whereas other process features showing less dependency to the quality variable and related to other minor root causes are neglected. In this way, the highly ranked features cannot represent all the necessary information for root cause diagnosis. Thus, a ranking scheme without the consideration of the diversity rule may lead to misleading results of root cause diagnosis. As we will see from the literature review, few existing feature ranking methods consider diversity or discrepancy of features. However, this goal is usually achieved by traditional quality tools like fishbone charts, as they intrinsically consider the difference of items therein.

In this chapter, we develop a feature ranking scheme that satisfies both requirements discussed above: it is based on a general dependency measure and satisfies the diversity rule. The ranking method is originated from the *distance correlation*, where we incorporated a new distance metric with the weights on features. To rank the features,

we formulate an optimization problem by maximizing the distance correlation while maintaining a certain degree of the sparsity of the weights. This optimization problem is essentially a conic quadratic programming problem [9], and thus can be solved effectively. This method is named as *Sparse Distance Correlation* method. As discussed above, it is suitable for retrospective analysis of the process data generated from manufacturing systems for identifying the leading features related to the variation of the quality variable.

The remainder of this chapter is organized as follows. Section 2.2 reviews the related literature on feature ranking and general dependency measures. Section 2.3 introduces the proposed SpaDC method. Section 2.4 investigates the theoretical properties of SpaDC, and provides an intuitive explanation of how it works and discusses certain characteristics of the method. Section 2.5 validates the method using simulation studies, which illustrates how the SpaDC method prioritizes the features that are dependent with the quality variable, and simultaneously satisfy the diversity criterion. Section 2.6 presents two applications of SpaDC: one involves ranking twenty-four process features in the epitaxy process of a solar cell manufacturing process, and the other involves ranking over one thousand overlay measurements in a lithography process, to further test the performance of SpaDC in high-dimensional settings. Section 2.7 concludes this chapter. Proofs are provided in the Appendix A.

2.2 Literature review

The problem of feature ranking and selection has been studied in the literature for a long time. Most feature selection methods are developed with a statistical model that associates the features and the responses. For linear models, methods such as stepwise regression [10]

and Lasso [11] can be used for feature ranking. Grömping [12] introduced the R package ‘relaimpo’, which provides six different assessments for the relative importance of regressors in the linear model, either based on the regression coefficients and their standard error, or the decomposition of R^2 statistics. Choi, et al. [13] discussed how ridge regression can also help to infer the importance of variables, and the ranking result is evaluated by concordance score, with the comparison with LASSO and the elastic net regression. They discovered that when the pairwise correlations among the features are heterogeneous, the ridge regression has improved ranking performance. However, these model-based ranking procedures are based on linear models between inputs and outputs and thus only aim for designated situations.

Except for the model-based feature ranking procedure, there are also ranking methods based on general dependency indices. General dependency indices are the extensions of Pearson correlation coefficients that not only measure the correlation between variables, but also take the general dependency of random variables into account. Examples of general dependency indices include mutual information [14], distance correlation [15, 16], and Hilbert-Schmidt independence criterion (HSIC) [17, 18]. Among them, the mutual-information-based method requires the estimation of the marginal and joint densities of each variable, and thus is difficult to be calculated efficiently. Distance correlation received much attention in recent years. The distance correlation originates from energy distance [19], a technique that characterizes the difference between distributions using pairs of observations. It was used by [20] to balance the distributions of covariates for estimating causal effects based on observational data. The distance correlation and the HSIC were shown to be equivalent [21].

General dependency indices can be used for feature ranking. In literature, Song, et al. [22] established an HSIC-based stepwise feature selection method, which can also be used for feature ranking. Li, et al. [23] and Kong, et al. [24] developed a simple feature screening method by selecting a bound to remove the features with a small distance correlation of the response variable. Yenigun and Rizzo [25] propose a stepwise variable selection method using distance correlation for regression modeling. However, these ranking procedures do not take diversity rule into consideration.

Recently, the concept of *diversity* of the features has been proposed in [26]. They propose to aggregate multiple the estimators in linear regression to form an overall fit. To achieve high accuracy of the prediction, they suggest diversity among the groups of features used by these estimators. In essence, the diversity between groups of features are encouraged as they provide unique information for the predictor of interest, which coincides our proposed diversity rule of feature ranking. The diversity rule of feature ranking is also similar to the minimal-redundancy-maximal-relevance (mRMR) criterion [27], which adopts a step-wise procedure and selects the m -th feature as the one most relevant to the output and most irrelevant with the previous $m - 1$ features. However, it is based on mutual information criterion, which relies on the density estimation for every pair of features and thus involves high computational complexity. Instead, the SpaDC method is based on the distance correlation from each pair of features, which can be calculated efficiently using the method proposed in Huo and Székely [28].

2.3 Sparse distance correlation (SpaDC) ranking procedure

Let $\mathbf{X} = (X_1, \dots, X_p)$ be the p -dimensional process features, and let Y be the associated quality variable. When n products are fabricated from the manufacturing system, the features are formatted into a data matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$. Let $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_p] = \begin{bmatrix} \mathbf{x}^{(1)\top} \\ \vdots \\ \mathbf{x}^{(n)\top} \end{bmatrix}$, where \mathbf{x}_i represents the i -th process feature of all products and $\mathbf{x}^{(j)\top}$ represents all process features obtained from sample j . The quality indices of these n products are denoted as $\mathbf{y} = (y^{(1)}, \dots, y^{(n)}) \in \mathbb{R}^{n \times 1}$. From the data \mathbf{X} and \mathbf{y} , we aim to obtain the ranks of the features that satisfy the diversity requirements.

2.3.1 Distance correlation

Our feature ranking procedure is based on distance correlation [29]. It is an energy statistic [30], the function of distances between all pairs of samples. As introduced in the Literature Review, it is a general dependency measure and can identify general dependency relationships.

Let random vector (\mathbf{X}, Y) follow an arbitrary joint distribution $F_{\mathbf{X}, Y}$. The distance covariance and distance correlation between \mathbf{X} and Y are defined based on two prescribed distance metrics $d_{\mathbf{X}}(\cdot, \cdot)$ and $d_Y(\cdot, \cdot)$ of space \mathbb{R}^p and \mathbb{R} respectively [15]. With these distance metrics, the *population distance covariance* for (\mathbf{X}, Y) is defined as the square root of

$$V^2(\mathbf{X}, Y) = \mathbb{E}[(d_{\mathbf{X}}(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}) - \bar{d}_{\mathbf{X}}(\mathbf{X}^{(1)}) - \bar{d}_{\mathbf{X}}(\mathbf{X}^{(2)}) + \bar{\bar{d}}_{\mathbf{X}}) \\ \times (d_Y(Y^{(1)}, Y^{(2)}) - \bar{d}_Y(Y^{(1)}) - \bar{d}_Y(Y^{(2)}) + \bar{\bar{d}}_Y)],$$

where $\bar{d}_X(\cdot) = \mathbb{E}_{X_1}[d_X(\cdot, \mathbf{X}^{(1)})]$, $\bar{\bar{d}}_X = \mathbb{E}_{X_1, X_2}[d_X(\mathbf{X}^{(1)}, \mathbf{X}^{(2)})]$, and $(\mathbf{X}^{(1)}, Y^{(1)})$ and $(\mathbf{X}^{(2)}, Y^{(2)})$ are two independent samples from the distribution $F_{X,Y}$. The function $\bar{d}_Y(\cdot)$ and the quantity $\bar{\bar{d}}_Y$ are defined similarly.

Based on $V^2(\mathbf{X}, Y)$, the *squared-distance correlation* between random vector \mathbf{X} and Y is defined as

$$R^2(\mathbf{X}, Y) = \frac{V^2(\mathbf{X}, Y)}{\sqrt{V^2(\mathbf{X}, \mathbf{X})V^2(Y, Y)}} \text{ if } V(\mathbf{X}, \mathbf{X})V(Y, Y) > 0.$$

Under certain conditions of $d_X(\cdot, \cdot)$ and $d_Y(\cdot, \cdot)$ [15], the value of $R^2(\mathbf{X}, Y)$ can be regarded as a dependency measure between \mathbf{X} and Y , as $0 \leq R^2(\mathbf{X}, Y) \leq 1$ and $R^2(\mathbf{X}, Y) = 0$ if and only if \mathbf{X} and Y are independent.

From the observed samples $\mathbf{X} \in \mathbb{R}^{n \times p}$ and $\mathbf{y} \in \mathbb{R}^{n \times 1}$, $V(\mathbf{X}, Y)$ and $R^2(\mathbf{X}, Y)$ can be estimated with the following procedure. First, calculate the pairwise distance $a_{kl} = d_X(\mathbf{x}^{(k)}, \mathbf{x}^{(l)})$, and then obtain $A_{kl} = a_{kl} - \bar{a}_{k\cdot} - \bar{a}_{\cdot l} + \bar{a}_{\cdot\cdot}$ where $\bar{a}_{k\cdot} = \frac{1}{n} \sum_{l=1}^n a_{kl}$, $\bar{a}_{\cdot l} = \frac{1}{n} \sum_{k=1}^n a_{kl}$, and $\bar{a}_{\cdot\cdot} = \frac{1}{n^2} \sum_{k,l=1}^n a_{kl}$. Similarly, calculate B_{kl} based on $b_{kl} = d_Y(y^{(k)}, y^{(l)})$.

The *sample distance covariance* is defined as

$$V_n^2(\mathbf{X}, \mathbf{y}) = \frac{1}{n^2} \sum_{k,l=1}^n A_{kl} B_{kl}. \tag{1}$$

The squared sample distance correlation $\hat{R}_n^2(\mathbf{X}, \mathbf{y})$ is defined analogously as

$$R_n^2(\mathbf{X}, \mathbf{y}) = \frac{V_n^2(\mathbf{X}, \mathbf{y})}{\sqrt{V_n^2(\mathbf{X}, \mathbf{X})V_n^2(\mathbf{y}, \mathbf{y})}}$$

when $V_n(\mathbf{X}, \mathbf{X}), V_n(\mathbf{y}, \mathbf{y}) > 0$.

Evidently, $R_n^2(\mathbf{X}, \mathbf{y})$ and $V_n(\mathbf{X}, \mathbf{y})$ are consistent estimators to their population counterparts. Through their sampling distributions, these statistics can be used to test the general independence between \mathbf{X} and Y . The effectiveness of the distance-based method in detecting general relationships has been validated in the literature [31].

2.3.2 Distance covariance based on the weighted ℓ_1 -distance metric

SpaDC assigns a weight $\beta_i \geq 0$ to each process feature X_i , and performs the ranking based on regularization path of $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)^\top$ when maximizing the sample distance correlation between \mathbf{X} and \mathbf{y} . To calculate the sample distance correlation from the dataset, we define the following the $\boldsymbol{\beta}$ -weighted ℓ_1 -distance between features

$$d_{\boldsymbol{\beta}}(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^p \beta_i |x_i - x'_i|. \quad (2)$$

The ℓ_1 -distance is used here because it leads to a convex formulation of an optimization problem, as we shall see later. It should be pointed that this distance metric cannot directly the dependence between Y and the interaction effects of X s. Here, we apply the Euclidean distance metric on the domain of \mathbf{y} , and the weighted sample distance covariance and the weighted sample distance correlation can be directly derived from Equation (1). The detailed derivation is given in Appendix A.1.

$$V_{n,\boldsymbol{\beta}}^2(\mathbf{X}, \mathbf{y}) = \mathbf{d}_n^\top \boldsymbol{\beta}; \quad R_{n,\boldsymbol{\beta}}^2(\mathbf{X}, \mathbf{y}) \propto \frac{V_{n,\boldsymbol{\beta}}^2(\mathbf{X}, \mathbf{y})}{\sqrt{V_{n,\boldsymbol{\beta}}^2(\mathbf{X}, \mathbf{X})}} = \frac{\mathbf{d}_n^\top \boldsymbol{\beta}}{\sqrt{\boldsymbol{\beta}^\top \mathbf{F}_n \boldsymbol{\beta}}}.$$

Here, the j^{th} element of vector \mathbf{d}_n is $d_{n,j} = V_n(\mathbf{x}_j, \mathbf{y})$, and the (i, j) -element of \mathbf{F}_n is $[\mathbf{F}_n]_{ij} = V_n(\mathbf{x}_i, \mathbf{x}_j)$. Notably, \mathbf{d}_n and \mathbf{F}_n are calculated from the sample distance covariance between the feature and the quality variable, and each pair of features, respectively, thereby the fast calculation procedure [28] can be employed. We note that $V_{\beta}(\mathbf{X}, Y) = 0$ if and only if each feature X_i is independent of Y for every i corresponding to $\beta_i > 0$, as shown in Appendix A.2.

2.3.3 Formulating the optimization problem

We assume that the feature $\mathbf{x}_1, \dots, \mathbf{x}_p$ are scaled to have $V_n(\mathbf{x}_i, \mathbf{x}_i) = 1$ for $i = 1, \dots, p$. We formulate the following optimization problem to achieve feature ranking:

$$\max_{\beta} \mathbf{d}_n^{\top} \beta \tag{3}$$

subject to $\beta^{\top} \mathbf{F}_n \beta = 1, \sum_{i=1}^p \beta_i \leq c; \beta_i \geq 0$ for all $i = 1, \dots, p$.

In this formulation, our aim is to find a sparse weight vector β that leads to the maximum weighted sample distance correlation $R_{n,\beta}^2(\mathbf{X}, \mathbf{y}) \propto \frac{\mathbf{d}_n^{\top} \beta}{\sqrt{\beta^{\top} \mathbf{F}_n \beta}}$. The denominator of $R_{n,\beta}^2(\mathbf{X}, \mathbf{y})$ is restricted to 1, and the constraint $\sum_{i=1}^p \beta_i \leq c$ is applied to encourage sparsity of β for key feature ranking. The parameter c controls the level of regularization, and the positive elements of the solution specify a subset of features that relate to Y . Considering that formulation (3) is not a convex optimization problem due to the constraint, it is further relaxed to the following convex optimization problem:

$$\min_{\boldsymbol{\beta}} -\boldsymbol{\beta}^\top \mathbf{d}_n \tag{4}$$

$$s. t. \boldsymbol{\beta}^\top \mathbf{F}_n \boldsymbol{\beta} \leq 1; \sum_{i=1}^p \beta_i \leq c; \beta_i \geq 0 \text{ for all } i = 1, \dots, p.$$

Proposition 1 gives the result on the validity of the relaxation and the uniqueness of the solution.

Proposition 1 With probability 1, all elements in vector \mathbf{d}_n have different values and \mathbf{F}_n is positive definite. As a result,

- (1) if formulation (3) is feasible, formulation (4) has a unique optimal solution; and
- (2) if formulation (3) is not feasible, then at most one element of $\boldsymbol{\beta}(c)$ is nonzero, and the optimal solution of (4) is also unique. ■

The proof is given in Appendix A.3.

Problem (4) can be transformed to a standard form of a conic quadratic programming problem [9], as detailed in Appendix A.4. Therefore, it can be solved efficiently with the existing interior-point convex optimization solver. In Section 2.4, we shall see that this problem leads to diversity, the intriguing property which is critical for feature ranking.

2.3.4 Feature ranking with distance correlation criteria

The SpaDC method ranks the features by solving Problem (4) with different values of regularization parameter c . According to Proposition 1, the solution to Problem (4) is unique, and we denoted it by $\boldsymbol{\beta}(c)$. Let $\mathcal{J}(c) = \{i: [\boldsymbol{\beta}(c)]_i > 0\}$ be the set of nonzero

elements of $\boldsymbol{\beta}(c)$. As c increases from 0 to a larger number, some elements among $\boldsymbol{\beta}(c)$ enters $\mathcal{J}(c)$ and the features are ranked based on the sequence of their first appearance in it. Specifically, each feature X_i is associated with a threshold

$$T_i = \inf\{c: i \in \mathcal{J}(c)\}. \quad (5)$$

The features X_1, \dots, X_p are then ranked by sorting T_1, \dots, T_p .

To implement the above idea, we first need to calculate all possible sets $\mathcal{J}(c)$ for a series of values $c \geq 0$. A direct approach to achieve this goal is to construct a regularization path $\{\boldsymbol{\beta}(c): c \geq 0\}$. However, there is no existing method to achieve this for formulation (4). The presence of quadratic constraints of $\boldsymbol{\beta}$ makes this problem essentially different from the problems whose regularization paths are well-studied [32-35]. For this reason, we need to evaluate $\boldsymbol{\beta}(c_j)$ for a series of values $c_j, j = 1, \dots, J$ and construct a dictionary $\mathcal{D} = \{(c_j, \boldsymbol{\beta}(c_j)): j = 1, \dots, J\}$. With such a dictionary \mathcal{D} , we can obtain $\tilde{T}_i = \min\{c_j: i \in \mathcal{J}(c_j), j = 1, \dots, J\}$, by which we rank feature X_i 's.

There are two specific implementations to obtain \mathcal{D} . One implementation is to adopt a bisection search algorithm. Using Proposition 2 below, we can effectively limit the values of c 's for which the problems need to be solved in the bisection search algorithm.

Proposition 2: Let \mathbf{F}_n be positive definite and all elements of \mathbf{d}_n are different.

- (1) Formulation (3) is not feasible if $c < 1$, and it is feasible when $c \geq 1$.
- (2) $\mathcal{J}(c) = \mathcal{J}(\sqrt{p})$ for $c > \sqrt{p}$.

(3) If $1 \leq c_1 < \tilde{c} < c_2 \leq \sqrt{p}$ and $\mathcal{J}(c_1) = \mathcal{J}(c_2)$, $\mathcal{J}(\tilde{c}) = \mathcal{J}(c_1) = \mathcal{J}(c_2)$. ■

The proof of Proposition 2 is given in Appendix A.5. Statements (1) and (2) of Proposition 2 specify that Problem (4) only needs to be solved for $c \in [1, \sqrt{p}]$, and statement (3) indicates that if the solution of formulation (4) at c_1, c_2 shows that $\mathcal{J}(c_1) = \mathcal{J}(c_2)$, then solving (4) again for $c \in (c_1, c_2)$ is unnecessary. With Proposition 2, we implemented a bisection search algorithm (Algorithm 1) to determine the ranks of all features. According to Proposition 2, the exploration starts with $c_{\min} = 1$ and $c_{\max} = \sqrt{p}$ in Step 1. In Step 2, the subroutine `Search_Interval` to find all the possible $\mathcal{J}(c)$'s according to $c \in (c_1, c_2)$, by evaluate if the middle point \tilde{c} satisfy $\mathcal{J}(\tilde{c}) = \mathcal{J}(c_1)$ or $\mathcal{J}(\tilde{c}) = \mathcal{J}(c_2)$, and explore the subintervals (c_1, \tilde{c}) if $\mathcal{J}(\tilde{c}) \neq \mathcal{J}(c_1)$ or the subinterval (\tilde{c}, c_2) if $\mathcal{J}(\tilde{c}) \neq \mathcal{J}(c_2)$ recursively.

Algorithm 1: Bisection search for ranking the features

1. Initiate $c_{\min} = 1$, $c_{\max} = p$, and calculate $\mathcal{J}(c_{\min})$ and $\mathcal{J}(c_{\max})$. Initiate the dictionary $\mathcal{D} = \{(c_{\min}, \mathcal{J}(c_{\min})), (c_{\max}, \mathcal{J}(c_{\max}))\}$; Set K_{\max} , the maximum levels of recursion.
 2. Call `Search_Interval`($c_{\min}, c_{\max}, \mathcal{J}(c_{\min}), \mathcal{J}(c_{\max}), 0$).
 3. Calculate $k_i = \operatorname{argmin}_c \{c; \mathcal{J}(c)\} \in \mathcal{D}; i \in \mathcal{J}(c)$. Then the rank of the features is determined by the ascending order of $k_i, i = 1, \dots, p$.
-

subroutine `Search_Interval` ($c_1, c_2, \mathcal{J}(c_1), \mathcal{J}(c_2), K$)

1. Let $c = (c_1 + c_2) / 2$, and calculate $\mathcal{J}(c)$. If $\mathcal{J}(c) \neq \mathcal{J}(c_1)$ and $\mathcal{J}(c) \neq \mathcal{J}(c_2)$, write $\{c; \mathcal{J}(c)\}$ to the dictionary \mathcal{D} ;
 2. If $K \geq K_{\max}$ **return**;
 3. If $\mathcal{J}(c) \neq \mathcal{J}(c_1)$, call `Search_Interval`($c_1, c, \mathcal{J}(c_1), \mathcal{J}(c), K + 1$);
 4. If $\mathcal{J}(c) \neq \mathcal{J}(c_2)$, call `Search_Interval`($c, c_2, \mathcal{J}(c), \mathcal{J}(c_2), K + 1$);
-

Besides the bisection method, the warm-start strategy, motivated by Friedman, et al. [36], is another implementation that is especially suitable when there are many process features while we are only interested in obtaining the ranks of the leading r features. This algorithm is summarized in Algorithm **22222222**. In this procedure, we start with $c = 1$.

In every step, we solve the optimization problem (4) at $c = k\delta$ using the interior point method, by setting $(k - 1)\delta$ as the initial point. If the nonzero elements of $\beta(k\delta)$ and $\beta((k - 1)\delta)$ are different, we add the solution $(k\delta, \mathcal{J}(k\delta))$ to the dictionary.

Algorithm 2: Warm-start procedure for ranking the features

Initiate $k = 1$. and solve $\beta(1)$. Initiate the output set $\mathcal{D} = \{(1, \beta(1))\}$.

Loop:

Solve $\beta(k\delta)$ using an iterative algorithm, starting from $\beta((k - 1)\delta)$ if possible

Add features $(k\delta, \mathcal{J}(k\delta))$ to \mathcal{D} if $\mathcal{J}(k\delta) \neq \mathcal{J}((k - 1)\delta)$

$k = k + 1$

Until $|\mathcal{D}| \geq r$.

In practice, features i and i' may share the same rank when $k_i = k_{i'}$. We regard such tied features with the same priority. For some features, no matter how we increase c the solved weight will always be 0. These features are regarded as having the least importance with respect to Y . The ties may be caused by small search depth, or some inherent reasons related to the ranking procedure which will be elaborated in Section 4.3.

2.4 Theoretical properties and discussions

In this section, we first investigate the theoretical properties of the SpaDC method. We show that under certain conditions, the features dependent with Y are ranked over the independent ones so that the diversity requirements can be achieved. The intuitive explanation how SpaDC satisfies the diversity requirement and how the ties are generated are illustrated using a three-feature demonstration. Finally, we discuss the applicable conditions of the algorithms.

2.4.1 Theoretical properties

Let us assume that Y is dependent with some of the features X_1, \dots, X_m and independent with the other features X_{m+1}, \dots, X_p . Proposition 3 below states that the probability that $\mathcal{J}(c) = \{1, \dots, m\}$ for some $c > 0$ will converge to 1 as the sample size $n \rightarrow \infty$, under the condition (A) below.

Proposition 3 Let $\mathbf{X} = (\mathbf{X}_1^\top, \mathbf{X}_2^\top)^\top$, where $\mathbf{X}_1 = (X_1, \dots, X_m)^\top \in \mathbb{R}^m$ and $\mathbf{X}_2 = (X_{m+1}, \dots, X_p)^\top \in \mathbb{R}^{p-m}$. X_i is independent with Y if and only if $i > m$. Assume that $E|X_i|^{2v} < \infty$ for all $i = 1, \dots, p$ and $E|Y|^{2v} < \infty$ for some even number $v \geq 2$. Let $A(\mathbf{X}_n, \mathbf{Y}_n)$ indicate the event that some c exists such that $\mathcal{J}(c) = \{1, \dots, m\}$. Let $[V(X_i, X_j)]_{p \times p} := \mathbf{F} = \begin{bmatrix} \mathbf{F}_{11} & \mathbf{F}_{12} \\ \mathbf{F}_{21} & \mathbf{F}_{22} \end{bmatrix}$, the population counterpart of \mathbf{F}_n , and let $\mathbf{d} = \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{bmatrix}$ be the population counterpart of \mathbf{d}_n . If the vector \mathbf{d}_1 belongs to the interior of the cone spanned by vectors $\mathbf{F}_{11}^{(1)}, \mathbf{F}_{11}^{(2)}, \dots, \mathbf{F}_{11}^{(m)}, \mathbf{1}_m$, where $\mathbf{F}_{11}^{(1)}, \dots, \mathbf{F}_{11}^{(m)}$ are the columns of \mathbf{F}_{11} , $\mathbf{1}_m = (1, \dots, 1)^\top \in \mathbb{R}^m$, we have $P(A(\mathbf{X}_n, \mathbf{Y}_n)) = 1 - O(n^{1-v})$.

The proof of Proposition 3 is given in Appendix A.6. Proposition 3 points out that the probability that there exists a c such that $\mathcal{J}(c)$ contains exactly the dependent features goes to 1 when $n \rightarrow \infty$.

The statement in Proposition 33333 relies on the condition that vector \mathbf{d}_1 belongs to the interior of the cone spanned by vectors $\mathbf{F}_{11}^{(1)}, \mathbf{F}_{11}^{(2)}, \dots, \mathbf{F}_{11}^{(m)}, \mathbf{1}_m$. In general, it holds when the dependency among \mathbf{X}_1 is weak, because the cone spanned by $[\mathbf{F}_{11}^{(1)}, \dots, \mathbf{F}_{11}^{(m)}, \mathbf{1}]$

has a large range in this scenario. Especially, if all features \mathbf{X}_1 are independent, the cone is simply \mathbb{R}_+^m and any $\mathbf{d}_1 > 0$ must lay in this cone and the statement of Proposition 3 holds.

Despite the implication of Proposition 3, the SpaDC method does not simply select the features with the largest sample distance correlation with Y like Li, et al. [23]. The following proposition illustrates how SpaDC method achieves the diversity requirement, and it will be illustrated intuitively in Section 4.2.

Proposition 4 Let $\mathbf{F} = [V(X_i, X_j)]_{p \times p}$ and $\mathbf{d} = [V(X_i, Y)]_{p \times 1}$, where $V(\cdot, \cdot)$ is the distance covariance based on univariate Euclidean metrics. Write \mathbf{F} and \mathbf{d} in the following block-wise form:

$$\mathbf{F} = \begin{pmatrix} \mathbf{F}_{11} & \mathbf{f}_1 & \mathbf{F}_{12} \\ \mathbf{f}_1^\top & f_{11} & \mathbf{f}_2^\top \\ \mathbf{F}_{12}^\top & \mathbf{f}_2 & \mathbf{F}_{22} \end{pmatrix}; \mathbf{d} = \begin{pmatrix} \mathbf{d}_1 \\ d_* \\ \mathbf{d}_2 \end{pmatrix},$$

where

$$\mathbf{F}_{11} \in \mathbb{R}^{m \times m}, \mathbf{f}_1 \in \mathbb{R}^{m \times 1}, \mathbf{F}_{12} \in \mathbb{R}^{m \times (p-m-1)},$$

$$\mathbf{f}_2 \in \mathbb{R}^{(p-m-1) \times 1}, \text{ and } \mathbf{F}_{22} \in \mathbb{R}^{(p-m-1) \times (p-m-1)}.$$

(1) If the probability that “there exists value c with $c > 1$, formulation (4) has a solution $(\boldsymbol{\beta}_{1,n}^\top \ 0)^\top$ with $\boldsymbol{\beta}_{1,n} > 0$ ” goes to 1 when $n \rightarrow \infty$, $\mathbf{d}_1 = \mathbf{F}_{11}\boldsymbol{\gamma}_1 + \mu\mathbf{1}$ for some $\boldsymbol{\gamma}_1 \geq 0$ and $\mu \geq 0$.

(2) Assume the condition in (1) holds. Under additional assumptions $\mathbf{f}_1 = \mathbf{0}$, $\mathbf{F}_{12}^\top\boldsymbol{\gamma}_1 + d_*\mathbf{f}_2 > \mathbf{d}_2$, and $d_* > \mu$, the probability that some c' exists such that $m + 1 \in \mathcal{J}(c')$ and

$m + 2, \dots, p \notin \mathcal{J}(c')$ goes to 1 when $n \rightarrow \infty$. It indicates that the probability that $\mathcal{J}(c_i)$ is an increasing set sequence as c_i increases, whereas X_{m+1} is not ranked before the features X_{m+2}, \dots, X_p that goes to zero. ■

The proof of this proposition is given in Appendix A.7. In Proposition 4, part (1) guarantees that the probability that features X_1, \dots, X_m are selected goes to 1 when $n \rightarrow \infty$. Part (2) gives the critical condition that with a high probability, $\mathcal{J}(c') = \{1, \dots, m + 1\}$ for some c' . Except that the feature $m + 1$ is independent with the features $m + 2, \dots, p$, the following situations help to satisfy the assumptions in Part (2):

- X_{m+1} is strongly dependent with Y (i.e., large d_*) as well as the rest of the features X_{m+2}, \dots, X_p (i.e., the large elements in \mathbf{f}_2).
- The dependency between each of X_{m+2}, \dots, X_p and Y is small (i.e., the small elements in \mathbf{d}_2).
- The dependency between X_{m+2}, \dots, X_p and certain members in X_1, \dots, X_m is strong (i.e., $\mathbf{F}_{12}^\top \mathbf{Y}_1$ is large).

The last situation indicates the diversity requirement.

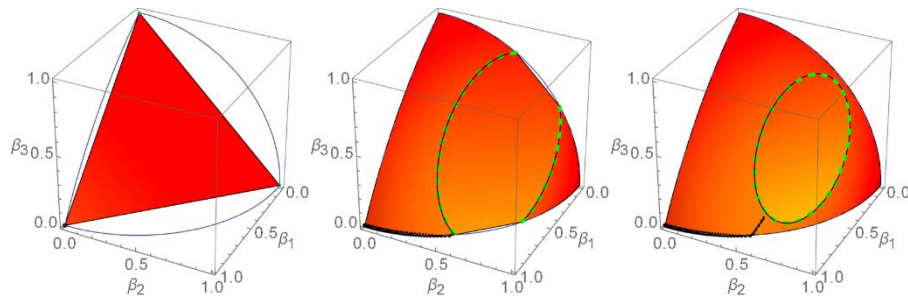
2.4.2 A three-feature illustration

To acquire an in-depth understanding of the SpaDC method, we consider two simple situations of ranking three features X_1, \dots, X_3 and attempt to illustrate the diversity requirement and the ties in the features, respectively. Specifically, each subplot of Figure 2 illustrates the feasible set of the optimization problem (4) and the objective values. Three

axes denote the decision variables β_1, β_2 and β_3 . The 3D shape with colors represents the feasible region of the problem (4) when the sample size is large: the intersection of the ellipsoid $\boldsymbol{\beta}^\top \mathbf{F} \boldsymbol{\beta} \leq 1$, the half-space $\boldsymbol{\beta}^\top \mathbf{1} \leq c$, and the octant $\{\boldsymbol{\beta}: \beta_1, \beta_2, \beta_3 \geq 0\}$. Note that the dashed green curve represents the intersection between the ellipsoid $\boldsymbol{\beta}^\top \mathbf{F} \boldsymbol{\beta} \leq 1$ and the plane $\boldsymbol{\beta}^\top \mathbf{1} = c$. The color of this 3D shapes' surface illustrates the negative objective value, $\mathbf{d}^\top \boldsymbol{\beta}$, where a large value is indicated by yellow color, and a small value is indicated by red color. Since the consistency result that the sample distance covariance \mathbf{d}_n and \mathbf{F}_n converges to \mathbf{d} and \mathbf{F} respectively, the solid region and the color of the 3D shape in each diagram illustrate the limit of the feasible region and the objective function when the sample size $n \rightarrow \infty$. In each diagram, the thick black line illustrate the path of $\boldsymbol{\beta}(c)$ when c changes from $c = 1$ to the current value.

The first row of figures illustrates how the diversity requirement is satisfied. Here X_1 and X_3 are strongly dependent with $F_{13} = 0.5$, $F_{12} = 0$ and $F_{23} = 0$. The feature X_3 is strongly related to Y with $d_1 = 0.6$, whereas $d_2 = d_3 = 0.4$. We can see how the diversity requirement works by observing that when $c = 1$, only $\beta_1 > 0$, and $\boldsymbol{\beta} = (\beta_1, 0, 0)$, as illustrated in the left figure. When c increases, β_1 and β_2 become non-zero, and $\boldsymbol{\beta} = (\beta_1, \beta_2, 0)$, as illustrated in the figure in the middle. When c becomes even larger, all β_1, \dots, β_3 are positive, and $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3)$. Therefore, X_1 ranks first, X_2 ranks second, and X_3 ranks third although $d_2 = d_3$, due to the curvature of the ellipsoid surface driven by \mathbf{F} , the dependency relationship among features. From this illustration, we can see that SpaDC achieves the diversity requirement through the interaction between the surface constraint $\boldsymbol{\beta}^\top \mathbf{F}_n \boldsymbol{\beta} = 1$ and the objective function $\boldsymbol{\beta}^\top \mathbf{d}_n$ in the optimization problem.

The three figures in the second row illustrate the tie of the features. The features X_1 and X_2 here are weakly dependent on Y , whereas an independent feature X_3 is strongly dependent with Y . In these figures, the description of the colored 3D shape, the dashed green curve, and the thick black curve have the same meaning as the figures in the first row. We can see from the left figure that when $c = 1$, $\boldsymbol{\beta}(1) = (0,0,1)^\top$. However, as shown in the figure in the middle, all elements of $\boldsymbol{\beta}(c)$ become positive simultaneously when c increases, as all points except for $(0,0,1)$ on the thick black line have three positive coordinates. Therefore, X_3 ranks first, and X_1, X_2 tie at the second place. From this example, we can see that the ties are inherent to the optimization problem, and it typically happens to highly dependent features that are less related to the quality variable. As a result, the SpaDC procedure may cluster the features into ordered groups with tying features. However, the authors do not regard it as a disadvantage for the proposed method in engineering practice, as the groups indicate different degrees of importance of features. As the features in the early groups tend to be more related to the quality variable and not dependent on each other, these tying features also provide useful information for root cause diagnosis and process monitoring.



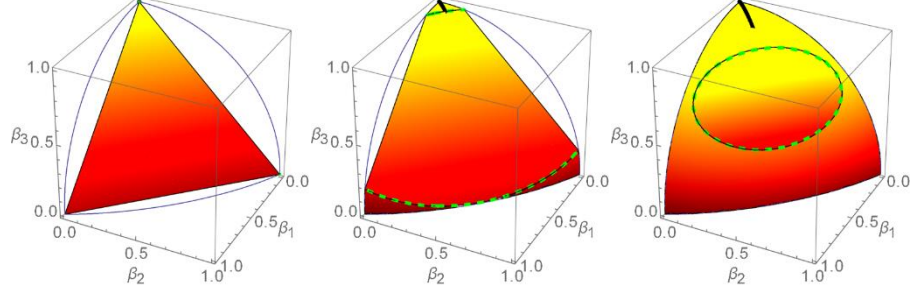


Figure 2 Illustration of the optimization problem (4).

2.4.3 Discussion

In this section, we discuss the computational complexity of the SpaDC procedure and discuss one limitation of the SpaDC algorithm on identifying the interaction effect of the features.

Computational Complexity. The overall computational time for the SpaDC involves two parts: (1) the calculation time of \mathbf{d}_n and \mathbf{F}_n and (2) the computational time for solving Problem (4) with a series of c 's. The vector \mathbf{d}_n and matrix \mathbf{F}_n involve $p(p+1)/2$ values of sample distance covariances. Using the method of Huo and Székely [28], each of these elements can be calculated with $O(n \log n)$ floating point operations, and the computation of different elements can be performed in parallel. For the second-order cone programming, the computation time for each $\boldsymbol{\beta}(c)$ is $O(p^3 \log(1/\epsilon))$ for calculating a solution $\boldsymbol{\beta}(c)$ with ϵ -accuracy [9].

Interaction Effects: The SpaDC method is essentially based on the statistics of \mathbf{d}_n and \mathbf{F}_n , the pairwise sample distance covariance between features and the sample distance covariance between \mathbf{x}_i and \mathbf{y} . For this reason, it cannot identify the dependency between Y and the *interaction effects* of two or more feature X_i 's. A deeper reason is that the weighted

ℓ_1 -distance is not a strong negative type [15] and cannot account for all dependency relationships between \mathbf{X} and Y , though it facilitates a convex formulation of the optimization problem. As a remedy, we can add low-order interactions of multiple process features like $X_i X_j$ to the inputs. However, in the chapter we focus on root-cause trace and diagnosis, where the main effects of process features are more important.

2.5 Simulation studies

In this section, we compare the SpaDC method with other existing feature selection and ranking methods in the literature. We aim to validate that our scheme ranks the dependent features prior to the independent ones, and meanwhile, it satisfies the diversity requirement.

2.5.1 Existing benchmarks and general settings

Six existing feature selection and ranking methods are used in the simulation study for benchmarking purposes. Yenigun and Rizzo [25] propose a stepwise variable selection method for a regression model based on the distance correlation of the residuals. This method, which is called YR method in short, derives a variable ranking method directly because a forward-selection procedure naturally gives an order of the variables. Li, et al. [23] propose a feature screening method through ranking the features X_1, \dots, X_p according to the individual relationship with Y , and the ranking scheme is called as LZZ in our simulation study. We also included the LMG method [37] implemented with the R package ‘relaimpo’ [12] in our comparison study, which ranks features based on the R^2 statistics of linear models. Three feature ranking methods in our comparison are based on predictive models. Two of them are based on linear models, i.e., the Lasso and adaptive Lasso methods [38, 39]. We used the MATLAB package `penalized` for computing the

regularization paths [40] for them, from which we rank the process features. The last method is based on feature importance indices of the random forest model [41], and we abbreviate it as RF method.

In the next two subsections, we will consider five settings where the features are independent and dependent. Under each setting, we generally follow Yenigun and Rizzo [25] and generate the datasets (\mathbf{X}, \mathbf{y}) for 1000 times. Six competing methods are applied to these 1000 datasets, generating 1000 sequences of the corresponding features. For $i = 1, \dots, p$, we count the number of times that each feature is ranked as the i^{th} one. When a tie of r features appears in a ranked feature sequence, each feature in this tie is then counted as $1/r$ replication on every tied rank. For example, assume that feature X_1 is ranked as the first feature; X_2 and X_3 are tied at the second feature in one replication. For X_1 , this replication is counted as one replication ranked as the first feature. For X_2 and X_3 , half replication is counted as the second feature, and half is counted as the third feature. Finally, the ranking distribution for each feature is calculated.

2.5.2 *Simulation with independent features*

In the first three settings, the number of features to be ranked is $p = 8$, and they are independent of each other.

- Setting 1: Let $X_1, \dots, X_8 \sim N(0,1)$, and $Y = |X_1| + X_2^2 + X_3 + \varepsilon$, where $\varepsilon \sim N(0,1)$. A total of 100 samples are generated from (\mathbf{X}, Y) .
- Setting 2: Let $X_1, \dots, X_8 \sim N(0,1)$ and $Y = \log(4 + \sin(2X_1)) + \sin(X_2) + X_3^2 + X_4 + 0.1) + \varepsilon$, where $\varepsilon \sim N(0,0.1^2)$. The sample size is 500.

- Setting 3: Let Y be dependent on three variables X_1, \dots, X_3 with $Y = Z(4 - X_1^2 - X_2^2 - X_3^2) + \varepsilon$, where $X_1, X_2, X_3 \sim \text{Unif}(-1, 1)$, $\varepsilon \sim N(0, 0.1^2)$ and $Z = \pm 1$ with equal probability. Here Z is independent with X_1, \dots, X_3 . Here Y has the equal probability of being positive or negative. A total of 500 samples are generated from (\mathbf{X}, Y) .

The ranking result of setting 1 is illustrated in Figure 3. In each diagram corresponding to X_j , each line illustrates the frequency that X_j is ranked from the 1st to the 8th place using a specific method, where the horizontal axis indicates the rank of the corresponding variable, and the vertical axis is the frequency value. We find that all methods rank X_3 at the first place most of the time. SpaDC (blue), LZZ (green) and YR (cyan) methods usually rank X_1 and X_2 at the second place and the third place. However, Lasso (orange), Adaptive Lasso (yellow) and LMG method (purple) tend to rank X_1 and X_2 to the first three places less often, and RF (red) ranks X_2 even fewer to the top-three. This is because Lasso and AdpLasso only capture the linear relationship, and random forest is not as sensitive to nonlinear dependency relationships as distance correlation-based methods.

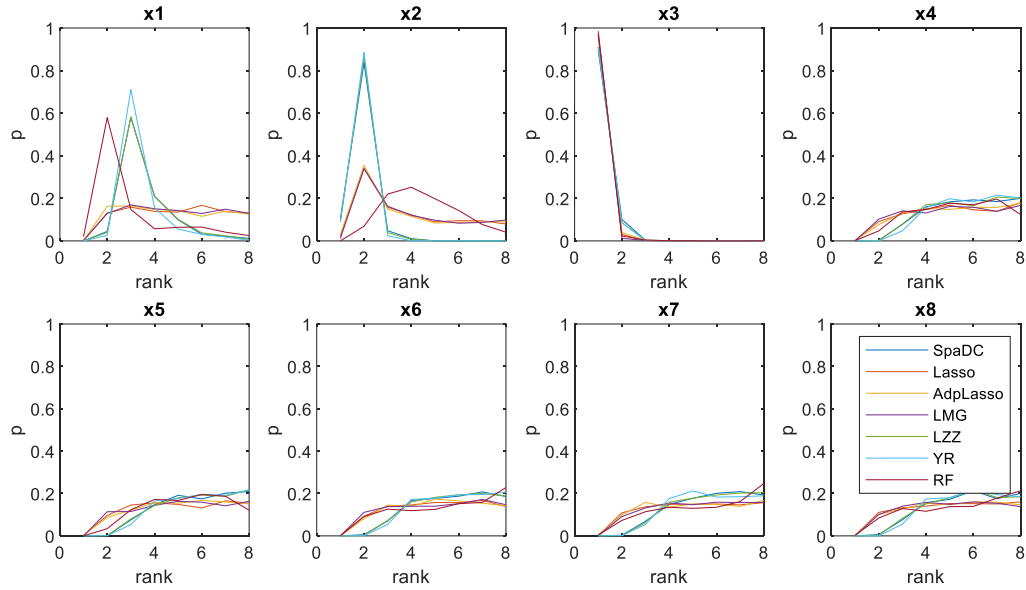


Figure 3 The comparison results in setting 1.

For the nonlinear relationship specified in setting 2, SpaDC, LZZ, and YR rank X_1, \dots, X_4 ahead of X_5, \dots, X_8 in most replications, as shown in Figure 4, whose interpretation is the same as that in Figure 1. However, the Lasso, AdpLasso, LMG, and RF methods rank features in X_3 to the 5th to 8th places most of the time. Hence, the schemes of the SpaDC, LZZ, and YR methods are more likely to rank the dependent features before the irrelevant ones when nonlinear dependency exists.

The results of setting 3 are illustrated in Figure 5, which shows that the methods based on general dependency measures tend to rank dependent features before the independent ones. However, the ranking methods based on predictive models (Lasso, adaptive Lasso, LMG, and RF) cannot deliver such performance because the features X_1, \dots, X_3 influence the variance of Y instead of its mean.

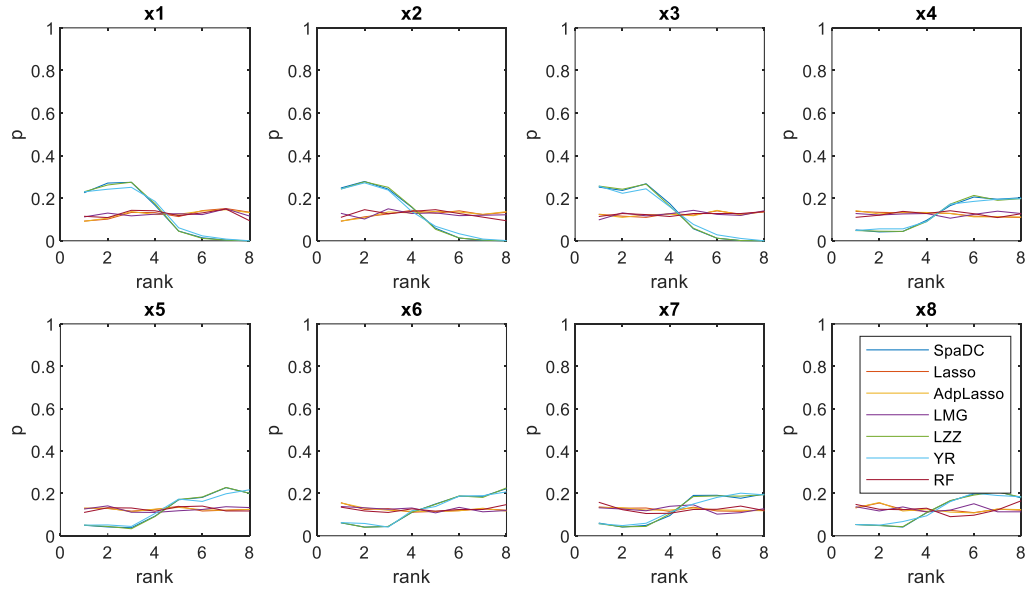


Figure 4 The comparison results in setting 2.

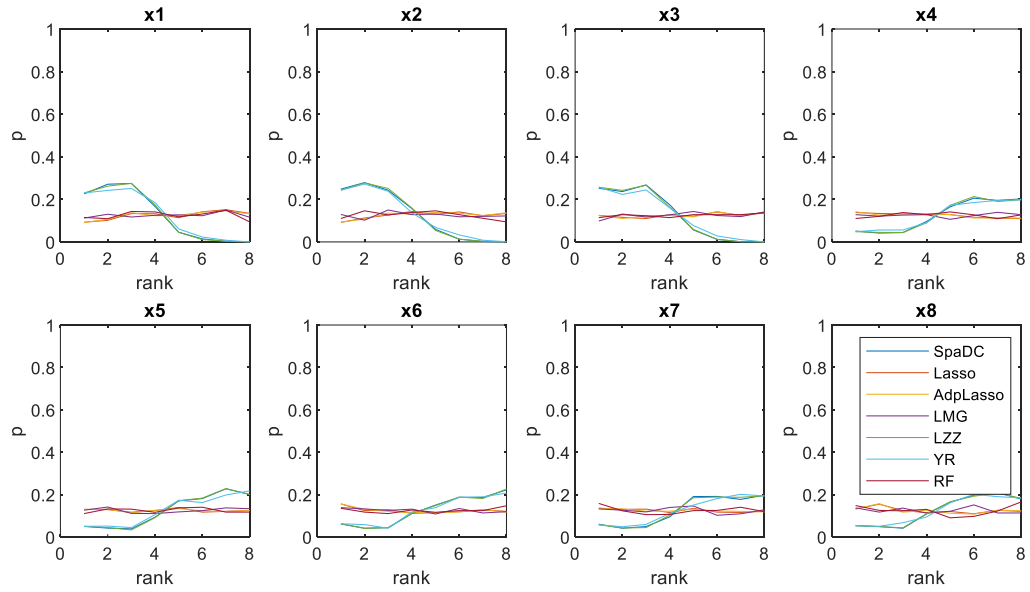


Figure 5 The comparison results in setting 3.

2.5.3 Simulation with dependent features

In the next two settings, we investigate the situation where the features are dependent. We focus on testing if the diversity requirement is satisfied.

- Setting 4: Three features, $\mathbf{X} = (X_1, X_2, X_3)^\top$ are generated, and Y represents the quality variable. $(X_1, X_2, X_3, Y)^\top$ jointly follows a multivariate normal distribution with zero mean and the following covariance structure:

$$\Sigma_{\mathbf{X},Y} = \begin{pmatrix} 1 & 0 & \rho_{13} & M \\ 0 & 1 & 0 & c \\ \rho_{13} & 0 & 1 & c \\ M & c & c & 1 \end{pmatrix}.$$

The feature X_1 and Y are strongly correlated ($\text{corr}(X_1, Y) = M = 0.6$), while (X_2, X_3) and Y are weakly correlated ($c = \text{corr}(X_2, Y) = \text{corr}(X_3, Y) = 0.2 < M$). Among the three features, X_2 is independent with X_1 , while X_3 is correlated with X_1 with a correlation coefficient of $\rho = 0.1, 0.2, 0.3, 0.4, 0.5$ and 0.6 . A total of 1000 samples are generated in this setting.

- Setting 5: A total of six features X_1, \dots, X_6 are generated, and Y represents the quality variable. $(X_1, \dots, X_6, Y)^\top$ jointly follows a multivariate normal distribution with zero mean and the following covariance structure

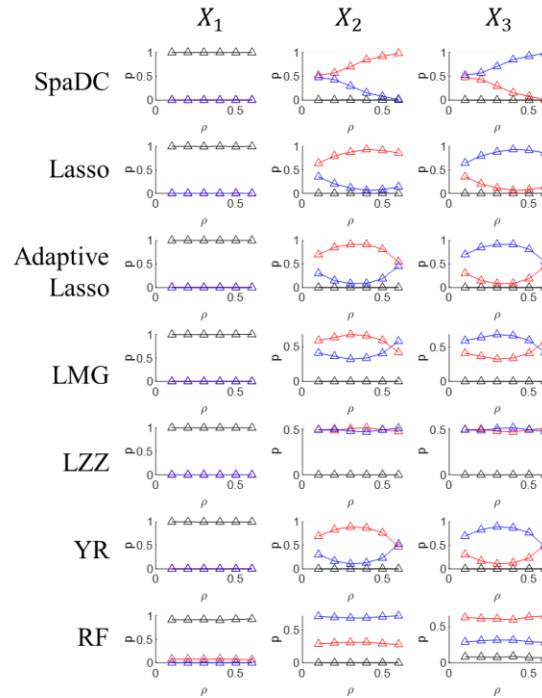
$$\Sigma_{\mathbf{X},Y} = \begin{pmatrix} 1 & \rho_{12} & 0 & 0 & 0 & 0 & M \\ \rho_{12} & 1 & 0 & 0 & 0 & 0 & c \\ 0 & 0 & 1 & \rho_{34} & 0 & 0 & M \\ 0 & 0 & \rho_{34} & 1 & 0 & 0 & c \\ 0 & 0 & 0 & 0 & 1 & \rho_{56} & c \\ 0 & 0 & 0 & 0 & \rho_{56} & 1 & c \\ M & c & M & c & c & c & 1 \end{pmatrix}.$$

With this structure, the features X_1, \dots, X_6 can be divided into three correlated groups: $[X_1, X_2], [X_3, X_4]$ and $[X_5, X_6]$. Y is strongly correlated with X_1 and X_3 , with $M = 0.6$. Meanwhile, Y is weakly correlated with the rest, that is, $c = 0.2$. The parameters ρ_{12}, ρ_{34} and ρ_{56} are equal, and they are selected from four values, i.e., 0.4, 0.5, 0.6 and 0.7. A total of 1000 samples are generated from (\mathbf{X}, Y) .

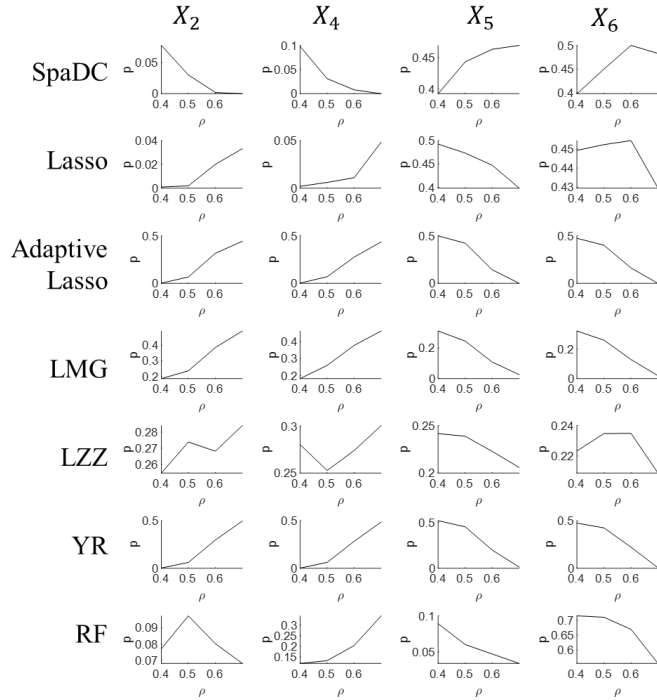
Figure 6 (a) illustrates the distribution of the ranks for X_1, X_2 and X_3 in setting 4 through line charts. Here, the row indicates the method, and the column indicates the variable X_1, \dots, X_3 . Each diagram describes the proportion of runs (y-axis) that this feature is ranked as the first (black line), the second (red line) and the third (blue line) place using this method, when ρ (x-axis) varies from 0.1 to 0.6. According to the results of setting 4, feature X_1 is always ranked as the first one. When ρ is 0.1, the frequencies that the ranks of X_2 and X_3 in SpaDC are distributed at the second and the third places are very close, as can be observed from the panel corresponding to X_2 and X_3 for the SpaDC method. However, when ρ increases from 0.1 to 0.6, X_2 and X_3 are more inclined to be ranked in the second and third places by the SpaDC method, respectively. This situation is not observed in the other four methods. Recall that X_2 is independent with X_1 , and thus prioritized to the second place. Therefore, SpaDC tends to prioritize the features that are independent of others with lower ranks to meet the diversity requirement, whereas other methods do not.

Recall that in setting 5, Y is strongly correlated with X_1 and X_3 , with $M = 0.6$. As expected, the results show that X_1 and X_3 are ranked in the first two places in most of the 1000 replications for methods in comparison. Figure 6 (b) shows how each method ranks

other features to the *third* place among these replications: the y-axis of each diagram represents the proportion one method rank X_j to the third place, $j = 2, 4, 5, 6$, and the x-axis illustrates $\rho = \rho_{12} = \rho_{34} = \rho_{56}$. For $\rho = 0.4, \dots, 0.7$, the SpaDC method ranks X_5 or X_6 in the third place with more replications than X_2 or X_4 . As ρ increases, the gap becomes much larger, and when $\rho = 0.7$, X_5 or X_6 is always ranked to the third place following X_1 and X_3 . This trend is not observed in the LZZ or YR method. Recall that X_5 and X_6 are the features that are not dependent with X_1 and X_3 , the result of this example demonstrates that SpaDC meets the diversity requirement when the relationship between \mathbf{X} and Y becomes more complex. However, the other four methods do not have such properties.



(a)



(b)

Figure 6 The comparison results in (a) setting 4 and (b) setting 5.

In conclusion, according to the results of the first three simulation settings, the SpaDC method is similar to the YR or LZZ method when the process features are independent of each other. Compared with the schemes based on linear models (i.e., Lasso and Adaptive Lasso) and random forest, the ranking schemes based on general dependency can capture the nonlinear dependency between the features \mathbf{X} and the quality variable Y as well as the case where features \mathbf{X} affect the variance of Y . The simulations under settings 4 and 5 further illustrate that the SpaDC method is superior to the YR and LZZ methods in satisfying the diversity requirement.

2.6 Case Studies

In this Section, we validate the SpaDC method using two real examples. One is the data analysis of a solar cell manufacturing process. Another one is the analysis of overlay data from a lithography process.

2.6.1 Epitaxy process in solar cell manufacturing

A solar cell manufacturing process has multiple stages, including epitaxy and evaporation. McEvoy, et al. [42] provided a detailed introduction of the fabrication process. In the epitaxy stage, semiconductor materials are deposited layer by layer on top of a substrate through a chemical vapor decomposition process. During this process, critical *in situ* process variables are measured, including temperature and reflectance. They are transformed to process features. The most important product quality variable, solar conversion efficiency (SCE), is generally tested offline after the manufacturing stages are completed. Practitioners are interested in ranking the process features based on their relationship with the SCE, so that they can monitor a small number of leading features observed during the manufacturing process and react as soon as the process changes without waiting for the SCE measurements from the final product.

The solar cell manufacturing process being investigated generates multiple functional signals that represent the reflectance of the wafer layer growth and the temperature within the chamber during the epitaxy process that generates three layers of thin films. Twenty-four features $\mathbf{X} = (X_1, \dots, X_{24})$ are extracted [43]. Among the 24 features, eighteen of them are obtained from the in-situ reflectance signals during this epitaxy process, and six of them are obtained from the feature extraction of the in-situ

temperature signals. The corresponding SCE, denoted as Y , is measured for the finished products. 50 samples of (\mathbf{X}, Y) are collected and the features are ranked using SpaDC.

The ranking results show that 24 features are ranked as $X_1, X_8, X_{23}, X_{22}, X_{24}, X_{20}, X_{11}, X_{10}$, followed by all the rest of the features tied together. The values of c corresponding to the leading eight features are 1.0, 1.6177, 1.7188, 2.0781, 2.2466, 2.2578, and 2.3926.

The strongest dependency between X_1 and Y is validated by the seven known follow-up samples acquired after these 50 samples. Figure 7 (a) shows that the final quality of the first two follow-up samples is in control and the last five follow-up samples are with a shifted mean. We check the individual control charts that monitor each feature and find that only X_1 exhibits an abrupt change during the last five samples, as shown in Figure 7 (b). This result shows the SpaDC method ranks X_1 correctly as the first feature.

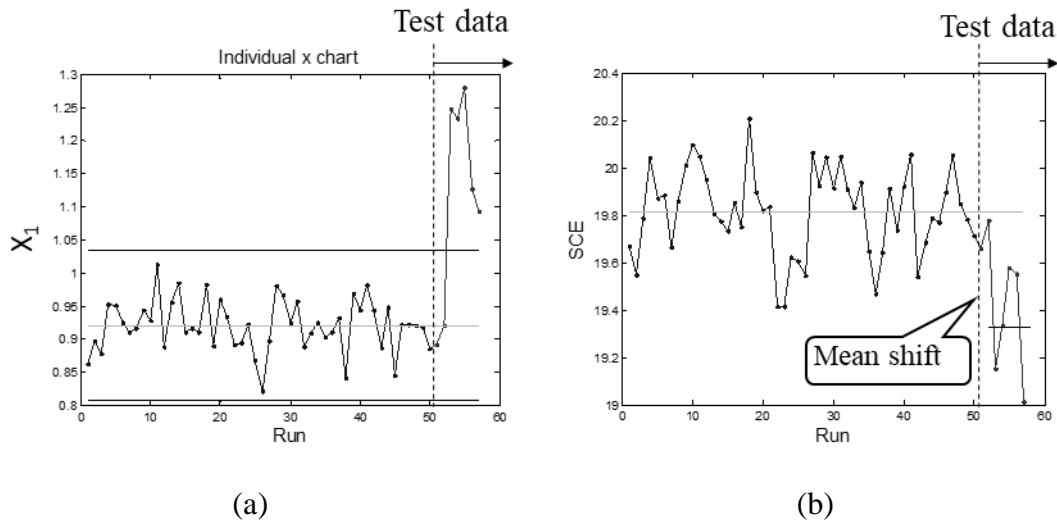


Figure 7 (a) The control chart for X_1 and (b) the control chart for SCE.

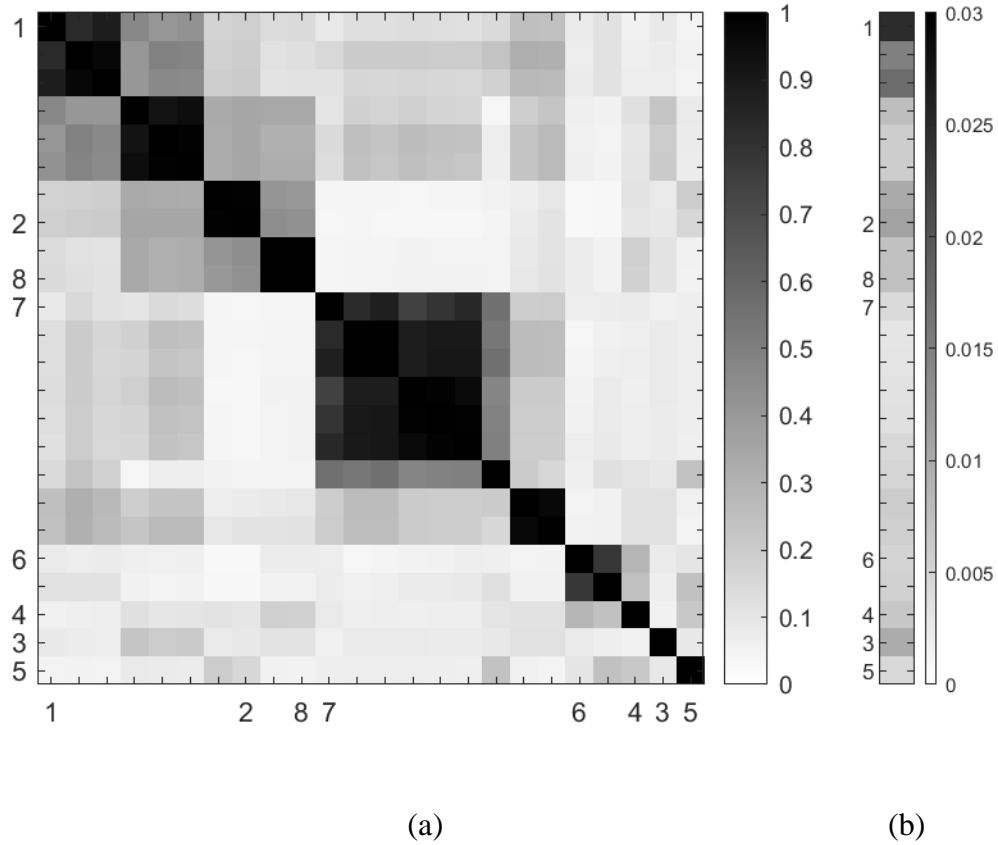


Figure 8 (a) Distance correlations between the two features and (b) the distance correlation between each feature and its quality variable.

We may interpret the results of the other features through Figure 8, which illustrates the sample distance correlation between each pair of features (\mathbf{F}_n) and the sample distance correlation between each feature and the SCE (\mathbf{d}_n). In this figure, the rows and the columns of the matrix of the left diagram and the element of the vector in the right diagram correspond to the feature 1, 2, ..., 24. The numbers marked at the left side and the bottom of the matrix and at the left side of the vector indicate the features' ranks. We can first observe that the features $X_1, X_8, X_{23}, X_{22}, X_{24}, X_{20}, X_{11}, X_{10}$ (whose ranks are marked as numbers at the sides of the matrix \mathbf{F}_n or vector \mathbf{d}_n in the figure) are all moderately dependent on Y , and we can observe that their sequence confirms with the magnitude of

\mathbf{d}_n . From \mathbf{F}_n , we identify further that the remaining 16 tied features (marked with numbers) relate to the former features or barely dependent on Y . Although the pairwise distance correlation values shown in Figure 8 facilitate interpretation of the results, the ranks of the features cannot be obtained directly. The ranks of the features must be obtained based on the procedure in this chapter.

2.6.2 *Lithography process in semiconductor manufacturing*

In a lithography process, the geometric pattern of one layer of microstructures is printed from a reticle onto the wafer surface through an exposure system. The overlay measurements of a wafer refer to the displacement error of the pattern at selected locations on the wafer, and they are regarded as the most important process data from the lithography process. The overlay measurements of an entire wafer are in the format of 2D overlay error map, as illustrated in Figure 9 (a). In this figure, the x - y plane represents the surface of the wafer, and each arrow at a point represents the displacement error of the printed geometric pattern at this location. Therefore, the desired appearance of an overlay error map is that all vectors are short and random.

The root causes in the lithography lead to specific patterns of the overlay error map. In this case study, we use a simulation testbed to generate the overlay errors for 1000 wafers. For the purpose of illustrating diversity requirements, we only generate the types of root causes that lead all vectors within a region to shift simultaneously with a random direction in our experiment, like local bumps of chucking and local lens distortion. A simulation testbed generates the overlay error for 1000 wafers, on which four root causes of this type affect the overlay error in four fixed regions, as illustrated in the shaded region

in Figure 9 (b). The measurements of overlay vectors are taken on a 21×21 grid. As each overlay vector is described by two real values, the entire overlay vector field contains over 842 features of the overlay vectors. For each wafer, the quality variable is the sum of the magnitudes of the underlying shifts in four regions, which is obtained from the testbed. We thereby obtained data matrix $\mathbf{X} \in \mathbb{R}^{1000 \times 842}$ and $\mathbf{y} \in \mathbb{R}^{1000}$.

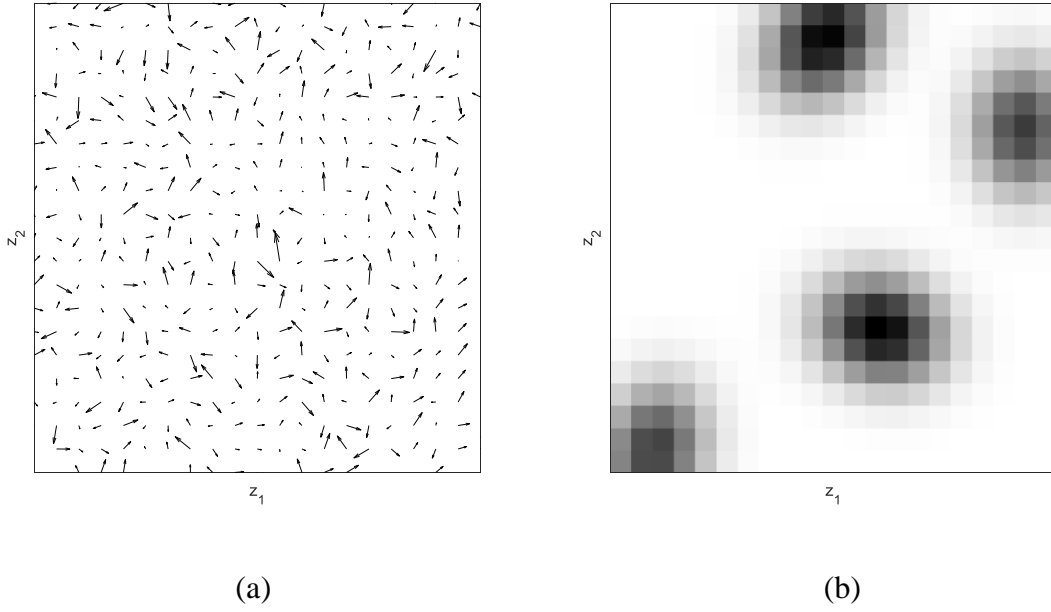


Figure 9 (a) The overlay error on one sample wafer and (b) the locations of the defects.

To automatically reveal the root causes of the overlay process, we rank all overlay features based on their relationship with this quality variable. As we will see later, the quality variable is nonlinearly related to the individual overlay measurement, and meanwhile, the overlay vectors corresponding to each root cause are significantly correlated. Despite these challenges, the leading features obtained from SpaDC can indicate the root causes happen at multiple locations on the wafer: they are all in regions affected by the root cause, and they include features from all four regions.

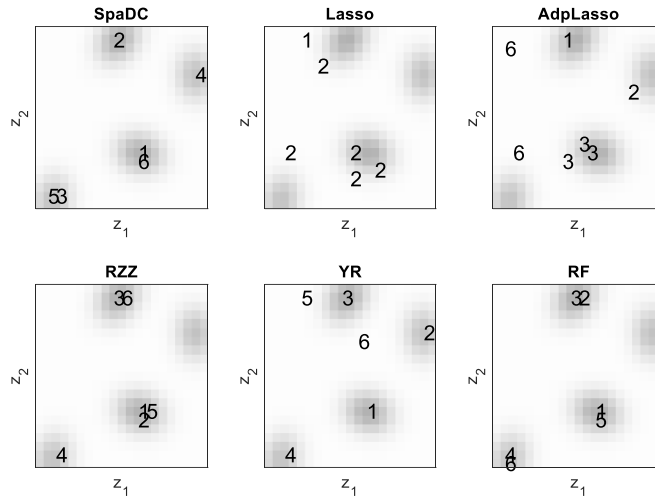


Figure 10 The ranking result of six methods.

We applied the six methods in ranking the overlay features, and the results are illustrated in Figure 10. In each subfigure, the numbers indicate the rank of the leading six features using a given method, and the locations of these numbers indicate the location of the error vector with that rank number. For the SpaDC method, the first six features are obtained with the warm start strategy, with the values $c = 1.00, 1.05, 1.15, 1.20, 1.55, 1.65$. We can see that the features with rank 1-4 correspond to the four root causes or regions of defects. This goal is also achieved by YR method, although in settings 4 and 5 of the simulation studies, YR method does not satisfy the diversity requirement. Actually, YR method does not take the diversity requirement into consideration intentionally. Compared with them, RZZ and RF methods both miss one root cause (the region at the upper-right part of the wafer) within the six leading features. We found that the feature of this missing root cause is ranked as number 7 and 9, respectively. The underlying reason is that RF and RZZ do not take the diversity requirement into consideration, and thus several leading features are dependent and correspond to the same root cause. Finally, Lasso and Adaptive Lasso are based on the linear model. As the magnitude of the disturbance linearly relates

to the lengths of the overlay error instead of its coordinates, there is a nonlinear relationship between the overlay features and quality variable. As a result, the leading features do not correspond to the regions affected by the root causes. This case study shows that the SpaDC method prioritizes the dependent features and simultaneously satisfies the diversity requirement.

2.7 Summary

This chapter considers an automatic root cause analysis method based on process data generated from manufacturing systems. Specifically, we aim to rank the process features based on their relationship with the quality variable. Based on the characteristics of the process data, we proposed two ranking rules to guarantee that the leading features provide useful information for process improvement: 1. the ranking method should be based on general-dependency measure, and 2. the ranking scheme considers diversity requirement. We further proposed SpaDC ranking scheme that satisfies both rules.

Our theoretical investigation and an illustration of the SpaDC indicate that it indeed satisfies the ranking rules we proposed. It is further validated through the simulation and real case studies of semiconductor manufacturing processes.

The SpaDC method may be improved and extended in two aspects. One potential direction is to find distance metrics that both enable feature ranking and take the interaction effect among features into consideration. Second, it is desirable to extend the formulation of SpaDC to further accommodate scalable computation with a large number of features.

CHAPTER 3. HOLISTIC MODELING AND ANALYSIS OF MULTISTAGE MANUFACTURING PROCESSES WITH SPARSE EFFECTIVE INPUTS AND MIXED PROFILE OUTPUTS

3.1 Introduction

Contemporary multistage manufacturing processes (MMPs) are usually equipped with advanced sensing systems that collect both massive process input variables and intermediate product quality measurements from each stage of an MMP. Examples of inputs include the process parameters set by the engineers, the environmental variables, and the external events that occurred to the process. The variation of the process output, the intermediate product quality measurements in every stage, is potentially caused by the variation of certain inputs of the processes. This article performs a root cause analysis of the variability in the outputs of MMPs by associating them with specific process inputs in all stages. Specifically, we aim at answering three inter-related questions: (i) which effective inputs relate to the variations of the outputs? (ii) what are the variation patterns of the outputs caused by these inputs? (iii) how each individual process input affects the manufacturing process? Answering these questions leads to a better understanding of the process variabilities.

The statistical analysis of MMPs has been conducted for decades [1-3]. However, there are two major limitations of the existing analytical methods. First, they are unable to be applied to intermediate product quality measurements of mixed types of data in an MMP, which are increasingly common in data-rich manufacturing environments. Here,

“mixed types of data” means that the data collected from different stages have different dimensions and distinct characteristics. As an example, a semiconductor manufacturing process consists of hundreds of stages, including deposition, lithography, plasma etching, ion-implantations, chemical-mechanical polishing, etc. [44]. In different stages, the corresponding outputs can be image types of data (e.g. spatial data like film thickness of each layer at multiple locations on the wafer [45], multivariate random field like alignment error at hundreds of positions on wafer surface [46], and the image of the etched trenches captured by the scanning electron microscope [47]), and/or assorted functional curves (e.g. the temperature, pressure, and radiofrequency curves during the reaction processes). As will be discussed in the next Section 3.2, Stream of Variation (SOV) modeling approaches [1] based on state space model is generally not suitable. There is a lack of appropriate analytical methods for MMPs where output sensing data is mixed types of data, such as images or functional curves.

Second, existing analytical methods for MMPs cannot handle a large number of inputs associated with each stage of an MMP. In the example of semiconductor manufacturing, tens of control variables adjust the exposure system in a lithography step. For one thing, a large amount of the inputs calls for efficient and parallel implementation of the model estimation algorithm. For another, we need to identify the inputs that are truly related to the outputs and establish the connections between them.

To perform root cause diagnostics for MMPs with mixed profile outputs and a large number of inputs, one may model the relationship between each pair of process input and output individually. However, the underlying relationships among mixed profile outputs and a large number of inputs cannot be effectively revealed due to the potentially complex

interactions among those variables, which may lead to the failure of identifying the effective inputs and output variation patterns. Another common practice for process engineers nowadays is to extract features from the outputs in every stage and model the relationship among the extracted features and the process inputs [48]. However, the selection of features tends to be subjective, and missing important features is always a risk.

In this chapter, we propose a system-level modeling framework to solve the diagnostic problem for MMPs that generate intermediate product quality measurements of mixed types and different dimensions and sparse effective inputs. Based on the characteristics of a MMP, we propose the following assumptions that facilitate our modeling approach:

- (1) Cascading assumption:** An input in one stage only affects the outputs generated from this stage and the downstream stages. Cascading assumption is rooted from the directional error propagation among stages: the input variation from one stage not only affects the quality measurement of the current stage, but its effect can propagate to the next stage, and further downstream stages. However, the input from one downstream stage cannot affect the quality measurement of an upstream stage.
- (2) Mixed data types:** The outputs generated from different stages of a process may be collected through different metrology systems, have different dimensions, and thus have different characteristics. We illustrate our modeling approach by assuming that each stage generates one of two types of outputs: smooth functional curves and smooth images as outputs. Our idea has the potential to be applied to measurements with different structural assumptions (see the discussion in Section 3.3.5).

(3) Sparsity assumption: The potential root causes of the outputs variability shall be driven by a small number of *effective inputs*. We assume that the effective inputs are sparse: they compose a very small portion of all inputs from an MMP.

(4) Low-rank assumption: In reality, the measurements from each stage of an MMS may have multiple sources of variations. Within a short period of time, the potential root causes shall be limited, and thus only a small number of dominant variation patterns significantly impact quality and system performance.

Leveraging these assumptions, we propose a *holistic* modeling framework for MMPs. The word “holistic” means that the model describes the entire manufacturing system composed of all stages, and we estimate the process parameters that represent the relationships between all process inputs and outputs simultaneously. An optimization problem is formulated for the estimation process, and its objective function contains the magnitude of the predictive error of each stage, the smoothness of the functional curves or image outputs, the sparsity of the effective inputs, and the number of the variability patterns caused by the inputs. From the estimation procedure, we can solve the three diagnostic problems for MMPs: identify the effective inputs, identify the variation patterns of the outputs, and describe how each input affects the output of each stage. In Section 3.4, an illustrative example is provided via simulation studies on how the proposed method effectively solves the above three problems.

To our knowledge, this study is the first one that proposes a holistic modeling and analysis framework for an MMP that generates assorted types of data. It simultaneously answers all questions involving the effective inputs, the variation patterns in the outputs,

and their connections. The idea behind the proposed method is extendable to a wide range of MMPs that involve (1) a comprehensive set of inputs, and (2) process outputs of mixed types of data with limited variation patterns from each stage. Also, our model estimation method based on an alternating direction method of multipliers (ADMM) is highly parallelizable and thus guarantees high computational efficiency for an MMP with more stages.

The remaining part of the chapter is organized as follows. In Section 3.2, we review related literature to highlight the necessity of this research. In Section 3.3, we present the mathematical description of our problem, formulate the optimization problem, and propose the algorithm for solving this problem. In Section 3.4, the methods proposed in Section 3 are validated through simulation experiments. Section 3.5 concludes the chapter.

3.2 Literature review

The modeling and statistical analysis of an MMP have been investigated for decades. Since the mid-1990s, state-space models were proposed to describe the stream of variation (SOV) in MMPs for assembly and machining processes [1]. Based on the state-space model, the estimation-based diagnostics methods have been proposed to find the connection between the product quality measurements from each stage and the sources of errors [49]. In most of the literature, the product quality measurements are 3D coordinates of a set of critical points on a fabricated part. Based on the engineering design and physics principles [50], the proposed state-space model can accurately describe the error propagation between stages in an MMP. For modeling complex MMPs with mixed profile outputs, two issues make the state-space modeling approach infeasible. First, the state-space model describes

the status of the manufacturing process by using a state vector. The state vector cannot be not defined when the product quality measures are represented by functional curves or images. Second, the state-space model assumes that the state at stage k is solely determined by the state at stage $k - 1$, and not related to previous stages $k - 2, k - 3, \dots$. However, in MMPs like semiconductor manufacturing processes, the output from one stage may relate to the inputs from more than one previous stage. Therefore, we did not adopt the state-space modeling approach but instead proposed a regressive approach that directly represents the relationship between the output of each stage and the inputs from all previous stages.

In order to model the MMPs that generate profile data like functional curves or images in each stage, we extend the literature on modeling the relationships between profile inputs and outputs. Regression between smooth profile data and scalars is part of functional data analysis [51], and studies like Li, et al. [52] tailor this technique for engineering applications. Recently, some studies [53-55] use tensors to describe the profile of the same size and use tensor regression techniques to model the relationship between profile inputs and outputs. Many studies that apply functional and tensor regression penalize the parametric vectors and matrices for promoting certain characteristics of the input and output profiles, including sparsity [11], continuity and smoothness [56], low-rank [57], and the flatness among neighboring elements [58]. They are used collectively for anomaly detection [59], multiple change point detections based on sparse signal dependency [60], and so forth, to improve the estimation accuracy and identify the effective inputs and the variation patterns caused by them. This chapter also applies penalizations to represent the characteristics of the mixed profile.

To solve the diagnostic problems of the MMPs that generate mixed profile outputs, the above profile data modeling techniques need to be integrated into a model that specifies how the inputs from one stage propagate to the follow-up stages. Currently, there is no well-established theory for modeling and analysis of MMPs with mixed profile outputs. A common practice of analyzing these MMP nowadays is to adopt a two-step procedure: we first extract a set of features from the process output profiles in every stage and then perform the analysis of the process based on these selected features. For example, Zhang, et al. [48] followed this procedure to perform anomaly detection from data from a single stage. However, the second step is usually sensitive to the set of features selected.

Our model estimation process involves solving an optimization problem with multiple penalization terms. We use an alternating direction method of multipliers (ADMM) consensus algorithm for this purpose. Its general framework is introduced in Parikh and Boyd [61] and Boyd, et al. [62]. We cast our problem into the appropriate form and adopt the ADMM consensus method to solve it.

3.3 Holistic Analysis Framework for MMP generating profiles and images

In this section, we first describe the data generated from an MMP, then propose a holistic analysis framework for the MMP. We present how to solve the modeling and estimation problem using an ADMM consensus algorithm. We also discuss the selection of the tuning parameters and the possible variation of the problem formulation.

3.3.1 Data scenario and problem description

We assume that the process outputs from each stage either includes multiple functional curves of the same length or an image. The process output of stage k can be written as $\mathbf{Y}_k \in \mathbb{R}^{m_k \times n_k}$. If the output from stage k is an image, \mathbf{Y}_k represents an image of size $m_k \times n_k$. If stage k generates functional curves, \mathbf{Y}_k represents m_k curves of length n_k . The set \mathcal{I} includes the indices of stages that generate image data, and $\mathcal{S} = \{1, \dots, K\} - \mathcal{I}$ represent the stages that generate functional curves. All outputs for product n are thus described by $\mathbf{y}^{\{n\}} = (\mathbf{Y}_1^{\{n\}}, \dots, \mathbf{Y}_K^{\{n\}})$. Throughout the chapter, we use curly brackets $\{\cdot\}$ to identify the product number.

Note that the structure of $\mathbf{y}^{\{n\}}$ describes the general data structure for the intermediate product quality data generated from an MMP of multiple types and dimensions. This structure is similar to a C struct or MATLAB[®] cell: the data generated from all stages are of different dimensions. If the size of matrices $\mathbf{Y}_1, \dots, \mathbf{Y}_K$ are the same, $\mathbf{y}^{\{n\}}$ can be seen as tensor data [54]. We assume that the data generated from all stages have different structures, in the sense that they may represent either images or functional curves, so that special considerations in data analytics are required. Finally, we note that in other applications, the process outputs can be even more complicated. For example, the data from each stage may include multiple images of different sizes, groups of functional curves of different sizes, or other structured data types such as spatial measurements or point clouds. It will be clear in Section 3.3.5 that the methodology proposed in this chapter can be extended to such scenarios potentially.

We further assume that there are q_k inputs from stage k that may affect the process, represented by $\mathbf{u}_k = (u_{k1}, \dots, u_{kq_k})$, $k = 1, \dots, K$.

For simplicity, we assume that the effect of the inputs on the outputs is always linear. The treatment of nonlinear effects is briefly discussed in Section 3.3.5.2. Based on a linear model, the effect of the process inputs on the process outputs can be described by equation (6):

$$\mathbf{Y}_k = \mathbf{B}_{k0} + \sum_{i=1}^k \sum_{j=1}^{q_i} u_{ij} \mathbf{B}_{ij,k} + \mathbf{E}_k \quad (6)$$

In model (6), the parametric matrix $\mathbf{B}_{ij,k}$ is of size $\mathbb{R}^{m_k \times n_k}$. It is referred to as an *effect matrix* as it describes the effect of the input u_{ij} on the process outputs measured from stage k . The collection of effect matrices is denoted as $\mathcal{B} = \{\mathbf{B}_{ij,k}: 1 \leq i \leq k \leq K, 1 \leq j \leq q_i\}$. The matrix \mathbf{B}_{k0} 's are called *offset matrices*, and the set of all offset matrices is denoted by $\mathcal{B}_0 = \{\mathbf{B}_{k0}: 1 \leq k \leq K\}$. Here \mathbf{E}_k is a matrix representing the modeling error of the stage k , and we assume that every entry of \mathbf{E}_k is with mean 0, and variance $\sigma_{E,k}^2$. The cascading effect of the process inputs is inherently reflected from this model as the input variable u_{ij} from stage i affects the output from stage k , \mathbf{Y}_k only if $i \leq k$. Other assumptions discussed above can be cast into specifications on the model parameter \mathcal{B}_0 and \mathcal{B} .

- (1) The process outputs generated from stage k either represent smooth functional curves or images. As $\mathbf{B}_{ij,k}$ and \mathbf{B}_{k0} represent the effect of parameter u_{ij} on such process outputs, they shall share the same characteristic as the curve or image data in stage k .

Specifically, if the process outputs from stage k are multiple smooth curves, every row in $\mathbf{B}_{ij,k}$ and \mathbf{B}_{k0} corresponds to a curve, and thus two elements whose indices are close in each row should have similar values. If the stage k generates smooth images, any two elements whose indices are close in $\mathbf{B}_{ij,k}$ or \mathbf{B}_{k0} should have similar values.

(2) Only sparse effective inputs affect outputs. It means that most u_{ij} 's ($1 \leq i \leq K$ and $1 \leq j \leq q_i$) are associated with effect matrices $\mathbf{B}_{ij,k} = \mathbf{O}$ for all $k = 1, \dots, K$.

(3) The major variability of the stage k output, caused by all process inputs up to this stage, is of low dimension. In other words, the major variation patterns caused by effective u_{ij} 's lie in a low dimensional subspace. Therefore, the matrix $\mathbf{B}_{\cdot,k}$ is of low rank, where

$$\mathbf{B}_{\cdot,k} = [\text{vec}(\mathbf{B}_{11,k}), \text{vec}(\mathbf{B}_{12,k}), \dots, \text{vec}(\mathbf{B}_{1q_1,k}), \dots, \text{vec}(\mathbf{B}_{kq_k,k})] \in \mathbb{R}^{(m_k n_k) \times (\sum_{i=1}^k q_i)}.$$

Here, $\mathbf{B}_{\cdot,k}$ constitutes the effects of $\sum_{i=1}^k q_i$ inputs from the first k manufacturing stages. The vectorization operator $\text{vec}(\cdot)$ transforms the $m_k \times n_k$ matrix to vectors of size $m_k n_k \times 1$.

3.3.1.1 Interpretation of the model and the usage for root cause diagnostics

The model (6) we proposed above can be illustrated by using Figure 11. In this figure, we can see that every input u_{ij} from stage i only affects the output in stage i and the following stages $i + 1, \dots, K$, indicated by the green arrows and the effect matrices $\mathbf{B}_{ij,k}$'s. Although we do not explicitly model how the output from stage k affects the output of stage $k + 1$ like the SOV modeling approach (shown as dashed blue arrows in Figure 11), we

acknowledge that the error propagation between the output stages exists, and embed this consideration into the cascading assumption: the effective input from stage i may not only affect stage i itself, but it may further influence the following stages $i + 1, i + 2$, etc. The proposed regressive approach has several benefits. First, it enables existing multilinear regression and functional regression techniques applied to the MMP. Second, it also enables an explicit description of how the input from one stage influences the output of a much later stage. Third, the regressive approach can be easily extended for other types of process outputs with different characteristics, as specified in Section 3.3.5.2.

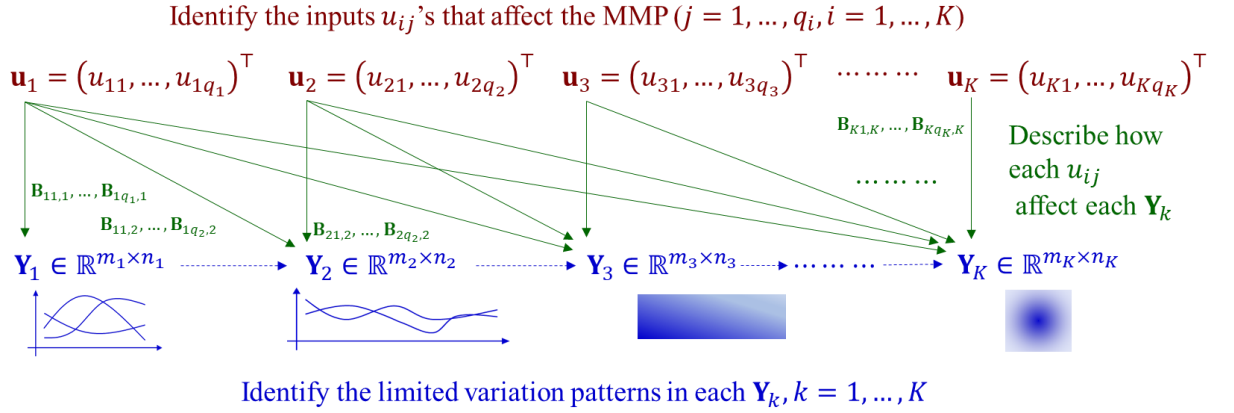


Figure 11 The illustration of the model (6)

We can use the proposed model (6) to solve the diagnostic problem of the MMP. After we obtain the inputs $\mathbf{u}_1^{\{n\}}, \dots, \mathbf{u}_K^{\{n\}}$ and the outputs $\mathbf{y}^{\{n\}}$ for samples $n = 1, \dots, N$, we can estimate the effect matrices in \mathcal{B} in the model (6), such that the cascading assumption, mixed data type assumption, sparsity assumption and low-rank assumption are satisfied. After the estimation is obtained, three questions of the diagnostics can be readily answered. Specifically, we can

- (i) identify the effective inputs (e.g. root causes): The input u_{ij} is identified as an effective input, if $\mathbf{B}_{ij,k} \neq \mathbf{0}$ for some $k \in \{i, i + 1, \dots, K\}$;
- (ii) identify the output variation patterns: The variation driven by all inputs in stage $1, \dots, k$ for the output of stage k is described by the linear subspace spanned by all $\mathbf{B}_{ij,k}: 1 \leq i \leq k, 1 \leq j \leq q_i$, whose dimension is given as the rank of $\mathbf{B}_{\cdot, k}$; and finally,
- (iii) determine the specific inputs on the outputs: How each input u_{ij} affects \mathbf{Y}_k the output of stage k is described by $\mathbf{B}_{ij,k}$.

Therefore, with the estimation of all effect matrices, the diagnostics problems in the introduction can be answered.

From the model, we can see that there are many parameters to be estimated in the modeling efforts. However, the overfitting problem can be avoided by the penalties applied to these parameters based on our model assumptions, as detailed in the Section 3.3.2. If we know that the possible causal structures between input variables and quality measurements in advance from domain knowledge, we may further limit the number of parameters to be estimated. For example, if we know that the inputs u_{ij} may only affect stage $\mathbf{Y}_i, \mathbf{Y}_{i+1}, \dots, \mathbf{Y}_{k_{ij}}$, we can add another constraint $\mathbf{B}_{ij,k} = \mathbf{0}$ for all $k > k_{ij}$.

In the following sections, we describe how to formulate an optimization problem to solve the parameters in \mathcal{B}_0 and \mathcal{B} , and how to solve them numerically.

3.3.2 Problem formulation

The objective of this study is to describe how the inputs u_{11}, \dots, u_{Kq_K} affect the outputs $(\mathbf{Y}_1^{\{n\}}, \dots, \mathbf{Y}_K^{\{n\}})$ in all stages. It is achieved by obtaining an estimation of \mathcal{B} and \mathcal{B}_0 with the characteristics described in the previous subsection. To obtain the estimation, we solve an optimization problem that minimizes the sum of the prediction error of the process outputs and the penalties specified by each assumption. These terms are detailed as follows.

Prediction error of the process outputs. From the model (6), the prediction error for the process outputs of product n from stage k can be represented as

$$\mathbf{E}_k^{\{n\}} = \mathbf{Y}_k^{\{n\}} - \mathbf{B}_{k0} - \sum_{i=1}^k \sum_{j=1}^{q_i} u_{ij}^{\{n\}} \mathbf{B}_{ij,k},$$

and the prediction accuracy can be represented as $\|\mathbf{E}_k^{\{n\}}\|_F^2$, where $\|\cdot\|_F$ is the Frobenius norm. The prediction accuracy across all K stages is then represented as

$$\mathcal{L}(\mathcal{B}, \mathcal{B}_0) = \sum_{n=1}^N \sum_{k=1}^K \|\mathbf{E}_k^{\{n\}}\|_F^2 = \sum_{n=1}^N \sum_{k=1}^K \left\| \mathbf{Y}_k^{\{n\}} - \mathbf{B}_{k0} - \sum_{i=1}^k \sum_{j=1}^{q_i} u_{ij}^{\{n\}} \mathbf{B}_{ij,k} \right\|_F^2 \quad (7)$$

In this expression, we sum up the loss $\|\mathbf{E}_k^{\{n\}}\|_F^2$ corresponding to all stages $1, \dots, K$, to determine the effective inputs and estimating their effects using the information from all stages. Note that every effective input affects the outputs in all later stages. Thus, we need to incorporate the information from all stages to identify them. Taking different magnitudes of the error and different numbers of elements of all stages into consideration, one may

incorporate a weight $1/(\hat{\sigma}_{E,k}^2 m_k n_k)$ to the term corresponding to stage k , where the parameter $\hat{\sigma}_{E,k}^2$ is a rough estimation obtained through smoothing the outputs of a subset of samples from every stage k and calculate the mean-squared error. Under this setting, the formulation is only slightly modified, and the solution framework remains the same.

The effects of each input are smooth. Assume that stage $k \in \mathcal{S}$ generates functional curves. According to Section 3.3.1, the elements on every row of $\mathbf{B}_{ij,k}$ should form a smooth function. To enhance the smooth property of the curves and thus increase the estimation accuracy, we propose the following penalization for these stages, similar to the smooth component in [59].

$$p_1(\mathcal{B}, \mathcal{B}_0) = \sum_{k \in \mathcal{S}} \lambda_{1,k} \left[\sum_{m=0}^{m_k} \|\mathbf{D}_S \mathbf{B}_{k0}(m, :)\|_2^2 + \sum_{i=1}^k \sum_{j=1}^{q_i} \sum_{m=1}^{m_k} \|\mathbf{D}_S \mathbf{B}_{ij,k}(m, :)\|_2^2 \right] \quad (8)$$

In this term, $\mathbf{B}_{ij,k}(m, :)$ describes the effect of u_{ij} on the m th functional curve generated from stage k . \mathbf{D}_S is a modified 1D second-order difference matrix for smoothing curves, with modified Neumann boundary condition [63],

$$\mathbf{D}_S = \begin{bmatrix} -1 & 1 & & & \mathbf{0} \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & \ddots & \ddots & \\ \mathbf{0} & & 1 & -2 & 1 \\ & & & 1 & -1 \end{bmatrix}.$$

The expression $\mathbf{D}_S \mathbf{B}_{ij,k}(m, :)$ gives a discretized approximation of function norm $\|f''(x)\|_2^2$ [56], where $f(x) = \mathbf{B}_{ij,k}(m, x)$. As we will see in Algorithm 4 and Algorithm 5, the choice of the boundary condition (the first and latest row of \mathbf{D}_S) enables efficient

computation. Here $\lambda_{1,k}$ is selected to control the degree of smoothing for the signals in stage k . Motivated by thin-plate splines [64], similar penalization term is defined for the stages that generate a smooth image,

$$p_2(\mathcal{B}, \mathcal{B}_0) = \sum_{k \in \mathcal{J}} \lambda_{2,k} \left[\text{vec}(\mathbf{B}_{k0})^\top \mathbf{R}_I \text{vec}(\mathbf{B}_{k0}) + \sum_{i=1}^k \sum_{j=1}^{q_i} \text{vec}(\mathbf{B}_{ij,k})^\top \mathbf{R}_I \text{vec}(\mathbf{B}_{ij,k}) \right]. \quad (9)$$

Here $\text{vec}(\cdot)$ transforms the image to a $m_i \times n_i$ vector, and \mathbf{R}_I is a discretized version of the operator

$$\mathcal{R}(g) = \int_{\mathbb{R}^2} \left[\left(\frac{\partial^2 g}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 g}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 g}{\partial y^2} \right)^2 \right] dx dy$$

that defines the “roughness” of bivariate function g [65]. The closed-form expression of the roughness matrix \mathbf{R}_I for an $m \times n$ matrix is derived and represented in Section 3 of Buckley [65]:

$$\mathbf{R}_I = (\mathbf{C}_m^\top \otimes \mathbf{C}_n^\top) (\mathbf{M}_m^2 \otimes \mathbf{I}_n + 2\mathbf{M}_m \otimes \mathbf{M}_n + \mathbf{I}_m \otimes \mathbf{M}_n^2) (\mathbf{C}_m \otimes \mathbf{C}_n)$$

where \mathbf{C}_n is discrete cosine transforms of order n , \mathbf{I}_n is an identity matrix of order n , \mathbf{M}_n is a diagonal matrix whose diagonal elements are $\mu_{i,n} = 2[1 - \cos\{\pi(i-1)/n\}]$, $i = 1, \dots, n$, and “ \otimes ” represents the Kronecker product. However, we will see later that the matrix \mathbf{R}_I does not need to be constructed explicitly.

The effects of the inputs are sparse. In model (6), “ $\mathbf{B}_{ij,k} = \mathbf{0}$ for all k ” is satisfied for most (i, j) pairs with $1 \leq i \leq k$ and $1 \leq j \leq q_i$. Motivated by the Group lasso algorithm [66], an ℓ_2 penalty is applied to all elements in \mathcal{B} that involve the input u_{ij} . Specifically, for u_{ij} we define a long vector

$$\mathbf{B}_{ij,\cdot} = \left(\frac{1}{\sqrt{C_i}} \text{vec}(\mathbf{B}_{ij,i}), \frac{1}{\sqrt{C_{i+1}}} \text{vec}(\mathbf{B}_{ij,i+1}), \dots, \frac{1}{\sqrt{C_k}} \text{vec}(\mathbf{B}_{ij,k}) \right)^\top$$

that constitutes all elements in \mathcal{B} , characterizing how u_{ij} affects the outputs. Here, the parameter $C_i = m_i n_i, \dots, C_k = m_k n_k$ are the number of elements in $\mathbf{B}_{ij,i}, \mathbf{B}_{ij,i+1}, \dots, \mathbf{B}_{ij,k}$ that adjusts the weights of the components to make the effects of each stage have comparable norms. The penalization term is then defined based on the $\|\mathbf{B}_{ij,\cdot}\|_2$, given as

$$p_3(\mathcal{B}) = \sum_{i=1}^K \left[\lambda_{3,i} \sum_{j=1}^{q_i} \|\mathbf{B}_{ij,\cdot}\|_2 \right]. \quad (10)$$

Here $\lambda_{3,i}$ controls the level of the sparsity of effective inputs from stage i . With more effective inputs from stage i , $\lambda_{3,i}$ should be selected smaller.

Variation caused by inputs is of low rank. As presented in Section 3.3.1, the matrix $\mathbf{B}_{\cdot,k}$ should be of low rank. A popular heuristic for solving rank minimization problems is by minimizing the nuclear norm of a matrix [67], and the nuclear norm penalization was proposed for reduced-rank regression [57]. We borrow this idea and apply the following penalization term to limit the number of variation patterns of each stage, resulted from all inputs that affect it.

$$p_4(\mathcal{B}) = \sum_{k=1}^K \lambda_{4,k} \|\mathbf{B}_{\cdot,k}\|_* \quad (11)$$

The overall objective function is given as the sum of the prediction error of the process outputs and four regularization terms $p_1(\mathcal{B})$, $p_2(\mathcal{B})$, $p_3(\mathcal{B})$ and $p_4(\mathcal{B})$ listed in (8)–(11). Therefore, our objective is to solve the following optimization problem:

$$\underset{\mathcal{B}, \mathcal{B}_0}{\text{minimize}} \quad \mathcal{L}(\mathcal{B}, \mathcal{B}_0) + p_1(\mathcal{B}, \mathcal{B}_0) + p_2(\mathcal{B}, \mathcal{B}_0) + p_3(\mathcal{B}) + p_4(\mathcal{B}). \quad (12)$$

3.3.3 Problem solution

Note that formulation (12) is a convex problem, lower bounded by zero. Therefore, it has an optimal solution. This problem has two characteristics. First, the problem has many decision variables, and thus a highly parallel algorithm is desired. Second, its objective function contains multiple non-differentiable additive components. For this reason, we apply an alternating direction method of multiplier (ADMM) consensus algorithm to solve this problem [61].

To cast the formulation (12) into the ADMM consensus framework, we introduce four copies of parameter \mathcal{B} , namely $\mathcal{B}^{(1)}$, $\mathcal{B}^{(2)}$, $\mathcal{B}^{(3)}$ and $\mathcal{B}^{(4)}$, and two copies of parameters \mathcal{B}_0 : $\mathcal{B}_0^{(1)}$ and $\mathcal{B}_0^{(2)}$. Then the formulation (12) is equivalent to the formulation (13) below:

$$\min f(\tilde{\mathcal{B}}) + g(\tilde{\mathcal{B}}). \quad (13)$$

In formulation (13), $\tilde{\mathcal{B}} = (\mathcal{B}_0^{(1)}, \mathcal{B}_0^{(2)}, \mathcal{B}^{(1)}, \mathcal{B}^{(2)}, \mathcal{B}^{(3)}, \mathcal{B}^{(4)})$ represents the collection of augmented parameters, the function $f(\tilde{\mathcal{B}}) = \mathcal{L}(\mathcal{B}_0^{(1)}, \mathcal{B}^{(1)}) + p_1(\mathcal{B}_0^{(2)}, \mathcal{B}^{(2)}) + p_2(\mathcal{B}_0^{(2)}, \mathcal{B}^{(2)}) + p_3(\mathcal{B}^{(3)}) + p_4(\mathcal{B}^{(4)})$. The function $g(\tilde{\mathcal{B}}) = I_{\mathcal{B}^{(1)}=\mathcal{B}^{(2)}=\mathcal{B}^{(3)}=\mathcal{B}^{(4)}}(\tilde{\mathcal{B}}) \cdot I_{\mathcal{B}_0^{(1)}=\mathcal{B}_0^{(2)}}(\tilde{\mathcal{B}})$ specifies that the copies are of the same values, where $I_A(x) = \begin{cases} 0 & x \in A \\ +\infty & \text{if } x \notin A \end{cases}$ is the indicator function. The formulation (13) is solved with a general framework of ADMM listed in Algorithm 3.

Algorithm 3: general framework of ADMM

Initialize $\tilde{\mathcal{Z}}$ and $\tilde{\mathcal{U}}$ as structs with the same shape as $\tilde{\mathcal{B}}$. Set all elements to 0.

Do:

Set $\tilde{\mathcal{B}}_{\text{prev}} \leftarrow \tilde{\mathcal{B}}$, $\tilde{\mathcal{Z}}_{\text{prev}} \leftarrow \tilde{\mathcal{Z}}$ and $\tilde{\mathcal{U}}_{\text{prev}} \leftarrow \tilde{\mathcal{U}}$

$\tilde{\mathcal{B}} \leftarrow \text{prox}_{\eta f}[\tilde{\mathcal{Z}} - \tilde{\mathcal{U}}]$ (**Step 1**)

$\tilde{\mathcal{Z}} \leftarrow \text{prox}_{\eta g}[\tilde{\mathcal{B}} + \tilde{\mathcal{U}}]$ (**Step 2**)

$\tilde{\mathcal{U}} \leftarrow \tilde{\mathcal{U}} + \tilde{\mathcal{B}} - \tilde{\mathcal{Z}}$ (**Step 3**)

Until $\|\tilde{\mathcal{U}} - \tilde{\mathcal{U}}_{\text{prev}}\| < \epsilon$ and $\|\tilde{\mathcal{Z}} - \tilde{\mathcal{Z}}_{\text{prev}}\| < \epsilon$.

In this algorithm, the summation and subtraction of two structs are naturally defined as adding and subtracting each corresponding element. The parameter η specifies the step size, and $\text{prox}_h(\mathbf{x})$ is the proximal operator, defined as

$$\text{prox}_h(\mathbf{x}) = \underset{\mathbf{y}}{\text{argmin}} \left\{ h(\mathbf{y}) + \frac{1}{2} \|\text{vec}(\mathbf{x} - \mathbf{y})\|_2^2 \right\}.$$

where the operator $\text{vec}(\cdot)$ in the second term transforms the struct to a long vector.

To perform this optimization algorithm, the proximal operator in **Step 1** and **Step 2** will be evaluated in the following two subsections.

3.3.3.1 Evaluating the proximal operator in Step 1

Recall that

$$f(\tilde{\mathcal{B}}) = \mathcal{L}(\mathcal{B}_0^{(1)}, \mathcal{B}^{(1)}) + \left[p_1(\mathcal{B}_0^{(2)}, \mathcal{B}^{(2)}) + p_2(\mathcal{B}_0^{(2)}, \mathcal{B}^{(2)}) \right] + p_3(\mathcal{B}^{(3)}) + p_4(\mathcal{B}^{(4)}).$$

It is the summation of four components involving non-overlapping elements. The separable property of the proximal operator [61] states that

$$\text{prox}_h(\mathbf{x}_1, \mathbf{x}_2) = \left(\text{prox}_{h_1}(\mathbf{x}_1), \text{prox}_{h_2}(\mathbf{x}_2) \right) \quad (14)$$

if $h(\mathbf{x}_1, \mathbf{x}_2) = h_1(\mathbf{x}_1) + h_2(\mathbf{x}_2)$. Therefore, the proximal operator of $\text{prox}_{\eta f}[\tilde{\mathcal{B}}]$ is determined by that of the following components: $\mathcal{L}(\mathcal{B}_0^{(1)}, \mathcal{B}^{(1)})$, $p_1(\mathcal{B}_0^{(2)}, \mathcal{B}^{(2)}) + p_2(\mathcal{B}_0^{(2)}, \mathcal{B}^{(2)})$, $p_3(\mathcal{B}^{(3)})$ and $p_4(\mathcal{B}^{(4)})$. The procedure of calculating these proximal operators is detailed as follows.

Evaluating the proximal operator of $\mathcal{L}(\mathcal{B}_0^{(1)}, \mathcal{B}^{(1)})$. The first term

$\mathcal{L}(\mathcal{B}_0^{(1)}, \mathcal{B}^{(1)}) = \sum_{n=1}^N \sum_{k=1}^K \left\| \mathbf{Y}_k^{\{n\}} - \mathbf{B}_{k0}^{(1)} - \sum_{i=1}^k \sum_{j=1}^{q_i} \mathbf{u}_{ij}^{\{n\}} \mathbf{B}_{ij,k}^{(1)} \right\|_F^2$ can be decomposed

into the summation of least square components that involve disjoint sets of elements

$\mathcal{B}_{k,v,w}^{(1)} = \left\{ \mathbf{B}_{k0}^{(1)}(v, w) \right\} \cup \left\{ \mathbf{B}_{ij,k}^{(1)}(v, w) : i = 1, \dots, k; j = 1, \dots, q_i \right\}$, corresponding to $k =$

$1, \dots, K; v = 1, \dots, m_k$, and $w = 1, \dots, n_k$:

$$\mathcal{L}(\mathcal{B}_0^{(1)}, \mathcal{B}^{(1)}) = \sum_{k=1}^K \sum_{v=1}^{m_k} \sum_{w=1}^{n_k} S_{k,u,v}(\mathcal{B}_{k,v,w}^{(1)})$$

where

$$S_{k,v,w}(\mathcal{B}_{k,v,w}^{(1)}) = \sum_{n=1}^N \left(\mathbf{Y}_k^{\{n\}}(v,w) - \mathbf{B}_{k0}^{(1)}(v,w) - \sum_{i=1}^k \sum_{j=1}^{q_i} u_{ij}^{\{n\}} \mathbf{B}_{ij,k}^{(1)}(v,w) \right)^2 \quad (15)$$

Therefore, evaluating the *proximal operator* of $\mathcal{L}(\mathcal{B}_0^{(1)}, \mathcal{B}^{(1)})$ reduces to evaluating the proximal operator of each component $S_{k,v,w}(\cdot)$ by using the identity (14) again. Each additive component $S_{k,v,w}(\cdot)$ is a quadratic function of the elements in $\mathcal{B}_{k,v,w}^{(1)}$, whose proximal operator can be calculated using Proposition 5 given in [61].

Proposition 5 If $q(\mathbf{x}) = \frac{1}{2} \mathbf{x}^\top \mathbf{A} \mathbf{x} + \mathbf{b}^\top \mathbf{x} + c$, with \mathbf{A} being a positive semidefinite matrix,

$$\text{prox}_{\eta q(\cdot)}(\mathbf{v}) = (\mathbf{I} + \eta \mathbf{A})^{-1}(\mathbf{v} - \eta \mathbf{b}) \quad (16)$$

where \mathbf{I} is an identity matrix with the same size as \mathbf{A} .

Note that each set $\mathcal{B}_{k,u,v}^{(1)}$ contains no more than $\sum_{i=1}^k q_i + 1$ parameters. Therefore, the inversion of the matrix therein is performed rapidly with little difficulty.

Evaluating the proximal operator of $p_1(\mathcal{B}_0^{(2)}, \mathcal{B}^{(2)}) + p_2(\mathcal{B}_0^{(2)}, \mathcal{B}^{(2)})$. Note that

$$\begin{aligned} p_1(\mathcal{B}_0^{(2)}, \mathcal{B}^{(2)}) &= \sum_{k \in \mathcal{S}} \left(\lambda_{1,k} \sum_{v=0}^{m_k} \left\| \mathbf{D}_S \mathbf{B}_{k0}^{(2)}(v, \cdot) \right\|_2^2 \right) \\ &+ \sum_{k \in \mathcal{S}} \sum_{i=1}^k \sum_{j=1}^{q_i} \left(\lambda_{1,k} \sum_{v=1}^{m_k} \left\| \mathbf{D}_S \mathbf{B}_{ij,k}^{(2)}(v, \cdot) \right\|_2^2 \right), \end{aligned}$$

$$\begin{aligned}
p_2(\mathcal{B}_0^{(2)}, \mathcal{B}^{(2)}) &= \sum_{k \in \mathcal{J}} \left(\lambda_{2,k} \text{vec}(\mathbf{B}_{k0}^{(2)})^\top \mathbf{R}_l \text{vec}(\mathbf{B}_{k0}^{(2)}) \right) \\
&+ \sum_{k \in \mathcal{J}} \sum_{i=1}^k \sum_{j=1}^{q_i} \left(\lambda_{2,k} \text{vec}(\mathbf{B}_{ij,k}^{(2)})^\top \mathbf{R}_l \text{vec}(\mathbf{B}_{ij,k}^{(2)}) \right).
\end{aligned}$$

They are both the summation of multiple components involving disjoint sets of parameters.

Each component corresponds to one parametric matrix, either an offset matrix $\mathbf{B}_{k0}^{(2)}$ or an effect matrix $\mathbf{B}_{ij,k}^{(2)}$. In the summations involving “ $k \in \mathcal{S}$ ”, every term is represented as

$$\lambda_{1,k} \sum_{v=0}^{m_k} \|\mathbf{D}_S \mathbf{T}(v, :)\|_2^2 := \sum_{v=1}^{m_k} p_S(\mathbf{T}(v, :)) \quad \text{where} \quad p_S(\mathbf{x}) = \lambda_{1,k} \|\mathbf{D}_S \mathbf{x}\|_2^2.$$

In the summation involving “ $k \in \mathcal{J}$ ”, every term is then represented as $p_l(\mathbf{T}) =$

$$\lambda_{2,k} \text{vec}(\mathbf{T})^\top \mathbf{R}_l \text{vec}(\mathbf{T}).$$

Therefore, it is sufficient to evaluate the proximal operator of $p_S(\mathbf{x})$ and $p_l(\mathbf{T})$ due to the separable property of the proximal operator (14). We derive the efficient evaluation of these proximal operators in Proposition 6 and 7. The proofs are given in Appendix 0.

Proposition 6 Given a d -dimensional signal $\mathbf{x} \in \mathbb{R}^d$, Algorithm 4 evaluates $\tilde{\mathbf{x}} = \text{prox}_{\eta p_S}(\mathbf{x})$, where $p_S(\mathbf{x}) = \lambda_1 \|\mathbf{D}_S \mathbf{x}\|_2^2$.

Algorithm 4: calculate $\tilde{\mathbf{x}} = \text{prox}_{\eta p_S}(\mathbf{x})$

- 1: Calculate $\mathbf{x}^* = \text{DCT}(\mathbf{x})$, where DCT represents the 1D discrete cosine transform [68].
 - 2: Set $\tilde{x}_i^* \leftarrow x_i^* / \left[1 + 4\lambda_1 \eta \left(1 - \cos\left(\frac{i-1}{d} \pi\right) \right)^2 \right]$ for $i = 1, 2, \dots, d$, where x_i^* is the i -th element of \mathbf{x}^* .
 - 3: Calculate $\tilde{\mathbf{x}} = \text{IDCT}(\tilde{\mathbf{x}}^*)$, where $\tilde{\mathbf{x}}^* = (\tilde{x}_1^*, \dots, \tilde{x}_d^*)^\top$ and IDCT represents the inverse discrete cosine transform.
-

Proposition 7 Given an $m \times n$ signal $\mathbf{T} = (t_{ij})_{m \times n} \in \mathbb{R}^{m \times n}$, Algorithm 5 evaluates $\tilde{\mathbf{T}} = \text{prox}_{\eta p_l}(\mathbf{T})$, where $p_l(\mathbf{T}) = \lambda_2 \text{vec}(\mathbf{T})^\top \mathbf{R}_l \text{vec}(\mathbf{T})$.

Algorithm 5: calculate $\tilde{\mathbf{T}} = \text{prox}_{\eta p_1}(\mathbf{T})$

- 1: Calculate $\mathbf{T}^* \leftarrow \text{DCT2}(\mathbf{T})$, where DCT2 represents the 2D discrete cosine transform.
 - 2: Set $\tilde{t}_{ij}^* \leftarrow t_{ij}^* / \left[1 + 4\lambda_2\eta \left(2 - \cos\left(\frac{i-1}{m}\pi\right) - \cos\left(\frac{j-1}{n}\pi\right) \right)^2 \right]$ for $i = 1, \dots, m$ and $j = 1, \dots, n$, where t_{ij}^* is the (i, j) element of \mathbf{T}^* .
 - 3: Calculate $\tilde{\mathbf{T}} \leftarrow \text{IDCT2}(\tilde{\mathbf{T}}^*)$, where $\tilde{\mathbf{T}}^* = (\tilde{t}_{ij}^*)_{m \times n}$ and IDCT2 represents the inverse 2D discrete cosine transform.
-

Evaluating the proximal operator of $p_3(\mathcal{B}^{(3)})$. The penalty

$p_3(\mathcal{B}^{(3)}) = \sum_{i=1}^K \left[\lambda_{3,i} \sum_{j=1}^{q_i} \left\| \mathbf{B}_{ij,\cdot}^{(3)} \right\|_2 \right]$ is also the summation of multiple components, each involving $\mathbf{B}_{ij,\cdot}^{(3)}$. By the separable property of the proximal operator (14), the proximal operator of $p_3(\cdot)$ can be evaluated by the proximal operators of each term $\lambda_{3,i} \left\| \mathbf{B}_{ij,\cdot}^{(3)} \right\|_2$. Its closed-form expression is given in Chapter 6.5.1, Parikh and Boyd [61].

Proposition 8 $\text{prox}_{\lambda \|\cdot\|_2}(\mathbf{x}) = \begin{cases} \left(1 - \frac{\lambda}{\|\mathbf{x}\|_2}\right) \mathbf{x}, & \text{if } \|\mathbf{x}\|_2 \geq \lambda \\ \mathbf{0}, & \text{if } \|\mathbf{x}\|_2 < \lambda \end{cases}$.

Evaluating the proximal operator of $p_4(\mathcal{B}^{(4)})$. The separable property of the proximal operator (14) can be invoked again to calculate the proximal operator of $p_4(\mathcal{B}^{(4)}) = \sum_{k=1}^K \lambda_{4,k} \left\| \mathbf{B}_{\cdot,\cdot,k}^{(4)} \right\|_*$. The following closed-form expression for the proximal operator of $\lambda_{4,k} \|\cdot\|_*$ is also from Chapter 6.7.3, Parikh and Boyd [61]. With this expression, the proximal operator of $p_4(\mathcal{B}^{(4)})$ can be evaluated.

Proposition 9 Let A be an $m \times n$ matrix with singular value decomposition $A = \sum_{i=1}^{\min\{m,n\}} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$. Then $\text{prox}_{\lambda_{4,k} \|\cdot\|_*}(\mathbf{A}) = \sum_{i=1}^{\min\{m,n\}} (\sigma_i - \lambda_{4,k})_+ \mathbf{u}_i \mathbf{v}_i^\top$, where $x_+ =$

$$\begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

3.3.3.2 Evaluating the proximal operator in Step 2

According to Parikh and Boyd [61], the proximal operator $\text{prox}_{\lambda g}[\cdot]$ involved in Step 2 is a projection onto the subspace $\{\tilde{\mathcal{B}}: \mathcal{B}^{(1)} = \mathcal{B}^{(2)} = \mathcal{B}^{(3)} = \mathcal{B}^{(4)}, \mathcal{B}_0^{(1)} = \mathcal{B}_0^{(2)}\}$. Thus, the update of $\tilde{\mathcal{Z}} = (\mathcal{Z}^{(1)}, \mathcal{Z}^{(2)}, \mathcal{Z}^{(3)}, \mathcal{Z}^{(4)}, \mathcal{Z}_0^{(1)}, \mathcal{Z}_0^{(2)})$ is given by

$$\mathcal{Z}^{(i)} \leftarrow \bar{\mathcal{B}} + \bar{\mathcal{U}}, i = 1, 2, 3, \text{ and } 4; \mathcal{Z}_0^{(i)} \leftarrow \bar{\mathcal{B}}_0 + \bar{\mathcal{U}}_0, i = 1 \text{ and } 2.$$

where $\bar{\mathcal{B}} := \frac{1}{4}(\mathcal{B}^{(1)} + \mathcal{B}^{(2)} + \mathcal{B}^{(3)} + \mathcal{B}^{(4)})$, $\bar{\mathcal{B}}_0 := \frac{1}{2}(\mathcal{B}_0^{(1)} + \mathcal{B}_0^{(2)})$, and $\bar{\mathcal{U}}, \bar{\mathcal{U}}_0$ are defined accordingly. With the above specification of Step 2, $\bar{\mathcal{U}}, \bar{\mathcal{U}}_0$ will remain constant during iterations of the algorithm (though $\mathcal{U}^{(1)}, \dots, \mathcal{U}^{(4)}$ and $\mathcal{U}_0^{(1)}, \mathcal{U}_0^{(2)}$ change during iterations), according to Step 3 of Algorithm 1. If $\bar{\mathcal{U}}, \bar{\mathcal{U}}_0$ are initialized at zeros, the step (2) of Algorithm 3 further reduces to $\mathcal{Z}^{(i)} \leftarrow \bar{\mathcal{B}}, \mathcal{Z}_0^{(i)} \leftarrow \bar{\mathcal{B}}_0$.

3.3.3.3 Summary of the proposed ADMM consensus algorithm

Now we put together the components listed in the above subsections and give a comprehensive optimization procedure in Algorithm 6. The notations $\bar{\mathcal{B}}_{k,v,w}, \bar{\mathbf{B}}_{k0}, \bar{\mathbf{B}}_{ij,k}, \bar{\mathcal{B}}_{i,j}$, and $\bar{\mathbf{B}}_{\cdot,k}$ are vectors or matrices, composed of elements in $(\bar{\mathcal{B}}, \bar{\mathcal{B}}_0)$, according to how $\mathcal{B}_{k,v,w}^{(1)}$ selects a subset of elements in $(\mathcal{B}^{(1)}, \mathcal{B}_0^{(1)})$, how $\mathbf{B}_{k0}^{(2)}$ and $\mathbf{B}_{ij,k}^{(2)}$ select subsets of elements in $(\mathcal{B}^{(2)}, \mathcal{B}_0^{(2)})$, how $\mathcal{B}_{i,j}^{(3)}$ selects a subset of elements in $(\mathcal{B}^{(3)}, \mathcal{B}_0^{(3)})$, and how $\mathbf{B}_{\cdot,k}^{(4)}$ selects a subset of elements in $(\mathcal{B}^{(4)}, \mathcal{B}_0^{(4)})$, respectively. All notations involving the letter ‘‘U’’ corresponds to their counterparts involving the letter ‘‘B’’.

For example, notation $\mathcal{U}_{k,v,w}^{(1)}$ in line (1a) refers to the subset of elements in $\mathcal{U}^{(1)}$, according to how $\mathcal{B}_{k,v,w}^{(1)}$ selects a subset of elements in $\mathcal{B}^{(1)}$.

Algorithm 6: The complete optimization procedure

Initiate $\bar{\mathcal{B}} = \mathcal{B}^{(1)} = \mathcal{B}^{(2)} = \mathcal{B}^{(3)} = \mathcal{B}^{(4)} = \mathcal{U}^{(1)} = \mathcal{U}^{(2)} = \mathcal{U}^{(3)} = \mathcal{U}^{(4)} = \mathcal{O}$ of the same shape as \mathcal{B} . Initiate $\bar{\mathcal{B}}_0 = \mathcal{B}_0^{(1)} = \mathcal{B}_0^{(2)} = \mathcal{U}_0^{(1)} = \mathcal{U}_0^{(2)} = \mathcal{O}$ of the same shape as \mathcal{B}_0 .

Do:

(1) Save $\bar{\mathcal{B}}_{0, \text{prev}} \leftarrow \bar{\mathcal{B}}_0$ and $\bar{\mathcal{B}}_{\text{prev}} \leftarrow \bar{\mathcal{B}}$.

(2a) **For** $k = 1, \dots, K, v = 1, \dots, m_k, w = 1, \dots, n_k$ **do:**

Update $\mathcal{B}_{k,v,w}^{(1)}$ by $\mathcal{B}_{k,v,w}^{(1)} \leftarrow \text{prox}_{\eta \mathcal{S}_{k,v,w}(\cdot)}(\bar{\mathcal{B}}_{k,v,w} - \mathcal{U}_{k,v,w}^{(1)})$ according to Proposition 5.

(2b) **For** $k = 1, \dots, K, i = 1, \dots, k, j = 1, \dots, q_i$ **do:**

If $k \in \mathcal{J}$: update $\mathbf{B}_{k0}^{(2)} \leftarrow \text{prox}_{\eta p_I}(\bar{\mathbf{B}}_{k0} - \mathbf{U}_{k0}^{(2)})$, $\mathbf{B}_{ij,k}^{(2)} \leftarrow \text{prox}_{\eta p_I}(\bar{\mathbf{B}}_{ij,k} - \mathbf{U}_{ij,k}^{(2)})$ based on Proposition 7.

If $k \in \mathcal{S}$: update $\mathbf{B}_{k0}^{(2)}(v, :) \leftarrow \text{prox}_{\eta p_S}(\bar{\mathbf{B}}_{k0}(v, :) - \mathbf{U}_{k0}^{(2)}(v, :))$ and $\mathbf{B}_{ij,k}^{(2)}(v, :) \leftarrow \text{prox}_{\eta p_S}(\bar{\mathbf{B}}_{ij,k}(v, :) - \mathbf{U}_{ij,k}^{(2)}(v, :))$ for all $v = 1, \dots, m_k$ based on Proposition 6.

(2c) **For** $i = 1, \dots, K$ and $j = 1, \dots, q_i$ **do:**

Update $\mathbf{B}_{ij,\cdot}^{(3)} \leftarrow \text{prox}_{\eta \lambda_{3,i} \|\cdot\|_2}(\bar{\mathbf{B}}_{ij,\cdot} - \mathbf{U}_{ij,\cdot}^{(3)})$ based on Proposition 8.

(2d) **For** $k = 1, \dots, K$ **do:**

Update $\mathbf{B}_{\cdot,k}^{(4)} \leftarrow \text{prox}_{\eta \lambda_{4,k} \|\cdot\|_*}(\bar{\mathbf{B}}_{\cdot,k} - \mathbf{U}_{\cdot,k}^{(4)})$ based on Proposition 9.

(3) Update $\bar{\mathcal{B}}$ and $\bar{\mathcal{B}}_0$ via $\bar{\mathcal{B}} \leftarrow \frac{1}{4}(\mathcal{B}^{(1)} + \mathcal{B}^{(2)} + \mathcal{B}^{(3)} + \mathcal{B}^{(4)})$ and $\bar{\mathcal{B}}_0 \leftarrow \frac{1}{2}(\mathcal{B}_0^{(1)} + \mathcal{B}_0^{(2)})$.

(4) Update $\mathcal{U}^{(t)} \leftarrow \mathcal{U}^{(t)} + \mathcal{B}^{(t)} - \bar{\mathcal{B}}$ for $t = 1, \dots, 4$ and $\mathcal{U}_0^{(t)} \leftarrow \mathcal{U}_0^{(t)} + \mathcal{B}_0^{(t)} - \bar{\mathcal{B}}_0$ for $t = 1, 2$.

Until $\max_{i=1,\dots,4} \|\bar{\mathcal{B}} - \mathcal{B}^{(i)}\|$, $\max_{i=1,2} \|\bar{\mathcal{B}}_0 - \mathcal{B}_0^{(i)}\|$, $\max_{i=1,\dots,4} \|\bar{\mathcal{B}} - \bar{\mathcal{B}}_{\text{prev}}\|$ and $\max_{i=1,2} \|\bar{\mathcal{B}}_0 - \bar{\mathcal{B}}_{0,\text{prev}}\|$ are below ϵ .

In Algorithm 6, all updating operations within the four ‘‘for loops’’ in step (2a)-(2d) can be performed in parallel, as they involve distinct groups of elements in $\tilde{\mathcal{B}}$. This notable feature significantly improves the computational efficiency. The variables in the optimization problem include the offset matrices \mathcal{B}_0 and the effect matrices \mathcal{B} listed in the cells of Table 1, which contains $\sum_{k=1}^K q_k$ columns and K rows. In essence, the step (2a)-

(2d) of the algorithms updates $(\mathcal{B}_0^{(1)}, \mathcal{B}^{(1)})$, $(\mathcal{B}_0^{(2)}, \mathcal{B}^{(2)})$, $\mathcal{B}^{(3)}$ and $\mathcal{B}^{(4)}$ through operating on multiple groups of elements in parallel. In step (2a), $(\mathcal{B}_0^{(1)}, \mathcal{B}^{(1)})$ is divided into $\sum_{k=1}^K m_k n_k$ groups. Each group corresponds to a triple (k, v, w) where $k \in \{1, \dots, K\}$, $v \in \{1, \dots, m_k\}$ and $w \in \{1, \dots, n_k\}$, and it consists of the (v, w) -element of \mathbf{B}_{k0} and all (v, w) -element of matrices listed in the k -th row of Table 1. In step (2b), $(\mathcal{B}_0^{(2)}, \mathcal{B}^{(2)})$ is updated by breaking them into $\sum_{i=1}^K (k+1-i)q_i + K$ groups according to the *cells* of Table 1 and the matrices in $\mathcal{B}_0^{(2)}$. In step (2c), $\mathcal{B}^{(3)}$ is divided according to $\sum_{i=1}^K q_i$ columns of Table 1. In step (2d), $\mathcal{B}^{(4)}$ is divided according to K rows of Table 1.

Table 1 Effect matrices estimated in formulation (6)

$\mathbf{B}_{11,1}$...	$\mathbf{B}_{1q_1,1}$				
$\mathbf{B}_{11,2}$...	$\mathbf{B}_{1q_1,2}$	$\mathbf{B}_{21,2}$...	$\mathbf{B}_{2q_2,2}$	
⋮	⋮	⋮	⋮	⋮	⋮	⋮
$\mathbf{B}_{11,K}$...	$\mathbf{B}_{1q_1,K}$	$\mathbf{B}_{21,K}$...	$\mathbf{B}_{2q_2,K}$ $\mathbf{B}_{K1,K}$... $\mathbf{B}_{Kq_k,K}$

As mentioned in previous sections, the major objective of this study is to identify the inputs u_{ij} 's affects the process outputs, the number of variation patterns caused by these inputs, and to understand how an effective u_{ij} affects the output of stage k . Whether the input u_{ij} affects the process can be identified through whether the parameter in \mathbf{B}_{ij} . are all zero and the number of variation patterns in stage k can be observed from the rank of $\mathbf{B}_{\cdot,k}$. When the algorithm terminates, however, such sparsity and low-rank property cannot be

observed from $\bar{\mathcal{B}}$ due to the numerical error. However, $\mathbf{B}_{ij,\cdot}^{(3)}$ s can be all zeros for many pairs of (i, j) given appropriate values of tuning parameters, as it is obtained from the proximal operator for an ℓ_2 -norm. Therefore, we may identify whether each u_{ij} affects the output quality measurements by observing whether $\mathbf{B}_{ij,\cdot}^{(3)} = \mathbf{O}$. Similarly, the $\mathbf{B}_{\cdot,\cdot,k}^{(4)}$ is of low-rank, and its rank specifies the number of variation patterns on stage k that all inputs cause. Finally, $\mathbf{B}_{ij,k}$, the effect of the input u_{ij} on stage k can be visualized through heat maps or multiple curves corresponding to $\mathcal{B}^{(2)}$, to give a visualization of smooth curves or images. As the algorithm converges, note that the difference between $\mathcal{B}^{(1)}, \dots, \mathcal{B}^{(4)}$ is very small, and all of them are close to $\bar{\mathcal{B}}$.

3.3.4 Convergent rate and complexity analysis

The convergence of the ADMM algorithm is guaranteed in the literature [62]. However, the existing theory on the convergence rate of consensus ADMM algorithm relies on the strong convexity assumption of component functions, which does not hold for $p_2(\mathcal{B}_0, \mathcal{B}), p_3(\mathcal{B})$ and $p_4(\mathcal{B})$. Now we only analyze the computation complexity for every iteration and leave the illustration of the empirical convergence behavior of the algorithm in the simulation study.

The computational complexity of the parallel operations in (2a)-(2d) are summarized as follows:

- Step (2a) solves $\sum_{k=1}^K m_k n_k$ linear systems in parallel. Among them, $m_k n_k$ linear systems are of order $\sum_{j=1}^k q_j$, $k = 1, \dots, K$, and each of them involves a computational complexity of $O\left(\left(\sum_{j=1}^k q_j\right)^3 + \left(\sum_{j=1}^k q_j\right)^2 N\right)$.

- Step (2b) involves smoothing images or curves in $\mathbf{B}_{k0}^{(2)}$ and $\mathbf{B}_{ij,k}^{(2)}$, $1 \leq i \leq k \leq K$, $1 \leq j \leq q_i$. If stage k generates curves, smoothing $\mathbf{B}_{k0}^{(2)}$ or $\mathbf{B}_{ij,k}^{(2)}$ involves m_k parallelable operations of complexity $O(n_k \log n_k)$, incurred by the discrete cosine transformation. If stage k generates an image, smoothing $\mathbf{B}_{k0}^{(2)}$ or $\mathbf{B}_{ij,k}^{(2)}$ involves a complexity of $O(m_k \log m_k + n_k \log n_k)$.
- Step (2c) shrinks $\mathbf{B}_{ij}^{(3)}$ for all $1 \leq i \leq k$ and $1 \leq j \leq q_i$ in parallel. The operation on each $\mathbf{B}_{ij}^{(3)}$ involves a computational complexity of $O(\sum_{k=i}^K m_k n_k)$.
- Step (2d) updates $\mathbf{B}_{:,k}^{(3)}$ for all $k = 1, \dots, K$ in parallel, and each step involves an SVD of a matrix of size $(\sum_{i=1}^k q_i) \times m_k n_k$. If we use the R-SVD procedure detailed in Section 8.6.3 of Golub and Van Loan [69], the computational complexity is of order $O\left(\left(\sum_{i=1}^k q_i\right)^3 + \left(\sum_{i=1}^k q_i\right)(m_k n_k)^2\right)$, assuming $\sum_{i=1}^k q_i < m_k n_k$.

We can see that Step (2a) and Step (2d) involves the highest total computational burden. Among them, Step (2d) cannot be parallelized sufficiently as well, as no more than K processes can be utilized. Compared with Steps (2a) and (2d), the computational efforts in Step 3 and Step 4 are negligible.

3.3.5 Discussion

In this section, we discuss various issues on the problem formulation: how to select the tuning parameters and possible variations of the problem formulations.

3.3.5.1 Selection of tuning parameters

In our formulation (12), the values of tuning parameters $\{\lambda_{1,k}, \dots, \lambda_{4,k}\}$ need to be specified. Literature often suggests setting the tuning parameters through a cross-validation (CV) procedure, though our simulation study shows that it under-smooth the signals and the images, and lead to a larger number of effective parameters and variation patterns. The same finding was pointed out by [59]_ENREF_48, and they adopted the Otsu's method based on maximizing inter-class variance. However, this method cannot be extended in our application because classes are not well-defined.

In general, a large value of $\lambda_{1,k}$ leads to smoother response signals of stage k , and a large value $\lambda_{2,k}$ leads to smoother image responses of stage k . A larger value of $\lambda_{3,i}$ leads to a fewer number of effective inputs from stage i , and larger value of $\lambda_{4,k}$ leads to a fewer number of variation patterns in stage k . According to the algorithm in Section 3.3.3, $\lambda_{4,k}$ aims to threshold the singular values of the matrix $\mathbf{B}_{\cdot\cdot,k}^{(4)}$ of size $m_k n_k \times \sum_{i=1}^k q_i$, and the variance of the error of every entry of $\mathbf{B}_{\cdot\cdot,k}^{(4)}$ is proportional to $\sigma_{E,k}^2$. According to [57], if an $p \times q$ matrix is the summation of a rank- r matrix and a matrix of $N(0, \sigma^2)$ error, its the $(r + 1)$ -th singular value is of magnitude $\sigma(\sqrt{p} + \sqrt{q})$. It motivates us to take $\lambda_{4,k} = c_{4,k} \sigma_{E,k} \left(\sqrt{m_k n_k} + \sqrt{\sum_{i=1}^k q_i} \right)$, where $c_{4,k}$ is a prescribed constant, to make the magnitudes of shrinkage applied on the matrices $\mathbf{B}_{\cdot\cdot,k}^{(4)}$'s comparable, even if their shapes differ.

The selection of $\lambda_{3,i}$'s and $c_{4,k}$'s should be regarded as a decision driven by the engineering need. For example, if the practitioners solve the model for identifying a wide range of inputs and output variations for root cause diagnosis, $\lambda_{3,i}$'s and $c_{4,k}$'s should be set to smaller values, which will lead to identifying more effective inputs and variation patterns. If the practitioners are only interested in the inputs that have major effects on the output variation patterns, larger values of $\lambda_{3,i}$'s and $c_{4,k}$'s are preferable.

Finally, we note that depending on an actual physical system, the tuning parameters $\{\lambda_{1,k}, \lambda_{2,k}, \lambda_{3,k}, c_{4,k}: k = 1, \dots, K\}$ corresponding to similar stages may be divided into multiple groups. Each group of parameters can take the same values, or selected using the same policy to reduce the complexity, as illustrated in the simulation study.

3.3.5.2 Variation of problem formulations according to process specifications

We finally note that the analytical framework presented in this section can be extended and configured based on the specific layout of the MMP and sensing system. First, some processes generate both functional curves and images in certain stages or generate curves and images of different sizes. The problem formulation and optimization algorithm can be applied with some minor modifications. Second, if the curves and images have various smoothness properties, different roughness penalties may be applied by discretizing the roughness penalties for functional data (Section 5.3.3 of Ramsay [51]). Third, certain manufacturing stages generate other forms of data, such as the spatial measurements seen in lithography processes, point cloud data in machining processes, as well as electrical signals that jump at discrete time points. Associated penalties based on spatial coordinates and point distance shall be applied based on the structure of such data, instead of using

smoothness penalties presented above. The ADMM consensus algorithm can be adjusted accordingly with minor modifications.

As a limitation of the proposed model, we assumed a linear relationship between the process inputs and the process outputs. If their relationship is not linear, we may include quadratic terms of the input variables into the model. Also, we can usually transform the process outputs from one stage into a set of meaningful features that are linearly related to the inputs. Furthermore, the interaction effect of two or more process inputs can be studied in our analytical framework as well, by including $u_i u_j$ terms in the inputs.

Finally, we note that the least square loss function can be replaced with other types of loss functions, such as Huber or Tukey loss, to yield more robust solutions when the error follows heavy-tail distributions or when outliers exist. If the loss function is convex, and its proximal operator can be evaluated effectively, the ADMM consensus framework can still be applied.

3.4 Simulation studies for performance evaluation

In this section, we set up a simulation platform to validate the methodology proposed in Section 3.3. We will demonstrate how to use the proposed framework to specify the number of variation patterns and effective inputs from the process inputs and intermediate product quality measurements.

3.4.1 Engineering background

A semiconductor manufacturing process involves tens of stages, including multiple chemical-vapor deposition steps, etching steps, lithography steps, and chemical mechanical

polishing. From each stage, multiple control variables, observable environmental variables, and in-situ process features are measured as inputs. For example, in lithography steps, the inputs include the control variables that adjust the alignment between layers. The in-line measurements of the wafer products after every stage include image data like film thickness measurements of the wafer maps from CVD steps and curves such as the profile of the etched trenches in plasma etching steps [44]. After all stages of processing, multiple chips are fabricated on a wafer, and they are tested, cut off from the wafer and packaged. The structure of this manufacturing process motivates our simulation setup.

The semiconductor manufacturers are interested in discovering how the inputs relate to the intermediate product quality measurements. However, there is no effective methods to identify those relationships between them. In practice, only a small number of variation patterns in outputs present in a given time period. This is because each variation pattern of quality data is typically driven by one root cause associated with process inputs. A well-maintained process should have limited variation sources in a short period. As we introduced in Section 3.3.1, the number of variation patterns for the outputs from stage k is represented by the rank of the parametric matrix $\mathbf{B}_{\cdot, k}$. Because only a small portion of the inputs affect the quality data, estimation of the effect matrices can be cast as a low rank, sparse estimation problem.

There are three objectives in our simulation study: (i) to evaluate whether the offset matrices $\{\mathbf{B}_{k0}: k = 1, \dots, K\}$ and the effect matrices $\{\mathbf{B}_{ij, k}\}$ can be estimated accurately; (ii) to identify the inputs related to the outputs; and (iii) to find the number of input-driven variation patterns presented in the outputs. Note that no existing method achieves all three

objectives of the analysis in this chapter. Specifically, if we build a separate model to model the relationship between the output of every stage k and the inputs from stage $1, \dots, k$, these models may result in different sets of effective inputs, and thus they may result in different effective inputs. The detailed set up of the simulation study is described in Section 3.4.2, and the results are discussed in section 3.4.3.

3.4.2 Specifications of simulation settings

Our simulation testbed has a total of $K = 4$ stages, and there are $q_i = 20$ input variables in stage i , $i = 1, \dots, 4$. The type and dimension of data generated from each stage are summarized in Table 2. Our simulation is performed under a relatively low dimensional setting compared with real applications because it enables us to conduct the simulation thoroughly with more replications and under a wider variety of problem settings.

Table 2 The type and dimension of data from each stage

Stage	1	2	3	4
Data type	multiple functional signals	multiple functional signals	images	images
Output Dimension	3 signals of length 10	4 signals of length 20	10×10	20×20
Input Dimension	$q_1 = 20$	$q_2 = 20$	$q_3 = 20$	$q_4 = 20$

We aim at simulating the system that satisfies the following conditions: (1) r_k variation patterns present in the outputs of stage k , (2) only the q_k effective inputs from stage 1 to k relates to the quality measurements of stage k , and (3) the image or multiple functional quality measurements are smooth. We first generate the true values of \mathcal{B}_0 and \mathcal{B} , so that (i) the matrix of $\mathbf{B}_{\cdot, k}$ has a rank of r_k (unless the number of rows or columns of $\mathbf{B}_{\cdot, k}$ is smaller than r_k), (ii) $\mathbf{B}_{ij, k} \neq \mathbf{0}$ if and only if u_{ij} is an effective parameter, and (iii)

the rows of \mathbf{B}_{k0} and $\mathbf{B}_{ij,k}$ form smooth curves if $k \in \mathcal{S}$, and the matrices \mathbf{B}_{k0} and $\mathbf{B}_{ij,k}$ form smooth images if $k \in \mathcal{J}$. For every stage $k = 1, \dots, K$, we first generate \mathbf{B}_{k0} using multiple univariate Gaussian processes (if $k \in \mathcal{S}$) or a bivariate Gaussian Process (if $k \in \mathcal{J}$). Then we generate r_k basis, using the same procedure of generating \mathbf{B}_{k0} . We finally generate $\mathbf{B}_{ij,k}$ corresponding to all effective input u_{ij} using a random linear combination of these r_k basis, such that the rank of $\mathbf{B}_{\cdot,k}$ is r_k . The detailed procedure of generating \mathcal{B}_0 and \mathcal{B} is given in Appendix B. Given \mathcal{B} and \mathcal{B}_0 , we generate the data corresponding to $N = 500$ products. The process inputs u_{ij} 's for each product are independent standard normal random variables, and the process outputs of each product are generated according to model based on the process inputs, where $\mathbf{E}_k^{\{n\}}$ are independent standard normal random variables with variance $\sigma_E^2 = 0.2$. Figure 12 illustrates the data collected from one sample, where the four subfigures are the multiple functional signals and image data collected from each manufacturing stage.

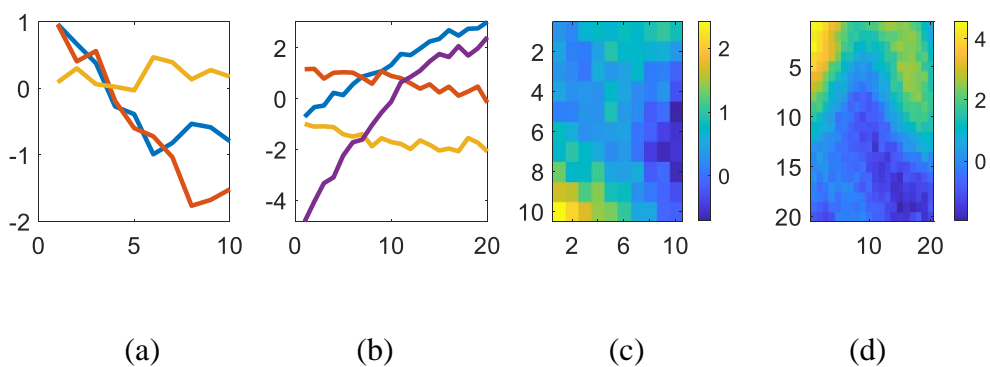


Figure 12 The outputs of curves and images from four stages.

In our simulation studies, we consider the four process setups in which the number of potential root causes from each stage (reflected by r_k , the rank of $\mathbf{B}_{\cdot,k}$, $k = 1, \dots, K$) and

the number of effective process inputs from each stage ($q_{e,k}, k = 1, \dots, K$) are varied: (1) $r_k = 2; q_{e,k} = 3$; (2) $r_k = 5; q_{e,k} = 3$; (3) $r_k = 2; q_{e,k} = 6$ and (4) $r_k = 5; q_{e,k} = 6$ for all $k = 1, \dots, K$. For each simulation setting, we perform the estimation for 300 times, according to 30 different collections of offset matrices \mathcal{B}_0 and effect matrices \mathcal{B} whose detailed generation procedure is reported in Appendix B. Each collection of offset and effect matrices define a specific manufacturing process. For each process, we generated 10 datasets corresponding to different inputs $\mathbf{u}_1, \dots, \mathbf{u}_K$ and random errors $\mathbf{E}_1, \dots, \mathbf{E}_K$, representing 10 datasets collected from the same process. The simulation settings are summarized in Table 3.

3.4.3 Optimization procedure and results of simulation studies

Based on each generated dataset with sample size $N = 500$, we perform the modeling and estimation procedure described in Section 3.3. Although the number of effective inputs $q_{e,k}$ and the number of variation patterns r_k differs, we select the same set of tuning parameters: $\lambda_1 = 1, \lambda_2 = 1, \lambda_3 = 0.25$ and $\lambda_4^k = 0.2(\sqrt{kJ} + \sqrt{m_k n_k}), k = 1, \dots, K$ according to the discussion in Section 3.3.5.1. We fix the step size $\eta = 5$ under all simulation settings.

Our algorithm is implemented in MATLAB, and the simulation study is conducted on a computing cluster. We did not implement the parallel for-loops in the algorithm in a parallel computing framework. To illustrate the speed and convergence property of the algorithm, we perform the estimation for one dataset, generated with $r_k = 2$ and $q_{e,k} = 3$, on a standalone mobile workstation with Intel Xeon E-2176M 2.7GHz CPU and 16GB

memory. The stopping criterion is that both the primal residual $\epsilon_{\text{prim}} = \max \left\{ \max_{i=1,\dots,4} \|\bar{\mathcal{B}} - \mathcal{B}^{(i)}\|, \max_{i=1,2} \|\bar{\mathcal{B}}_0 - \mathcal{B}_0^{(i)}\| \right\}$ and the dual residual $\epsilon_{\text{dual}} = \max \left\{ \max_{i=1,\dots,4} \|\bar{\mathcal{B}} - \bar{\mathcal{B}}_{\text{prev}}\|, \max_{i=1,2} \|\bar{\mathcal{B}}_0 - \bar{\mathcal{B}}_{0,\text{prev}}\| \right\}$ are below 10^{-5} , where $\|\cdot\|$ is the max norm. The algorithm converges in 2133 iterations. On average, each iteration takes 0.61s. We observed that the major computational burden is in Step (2a), where each iteration takes an average of 0.51s. In step (2a) we need to construct and solve $m_1 n_1 = 30$ linear systems of order 20, 80 linear systems of order 40, 100 linear systems of order 60, and 400 linear systems of order 80. However, these linear systems can be solved in parallel. As for the convergence speed, the change of $\log \epsilon_{\text{prim}}$ and $\log \epsilon_{\text{dual}}$ in all iterations are illustrated in Figure 13. From this figure, we can see that the primal and dual residual are consistently dropping. However, the algorithm has a sublinear convergence rate.

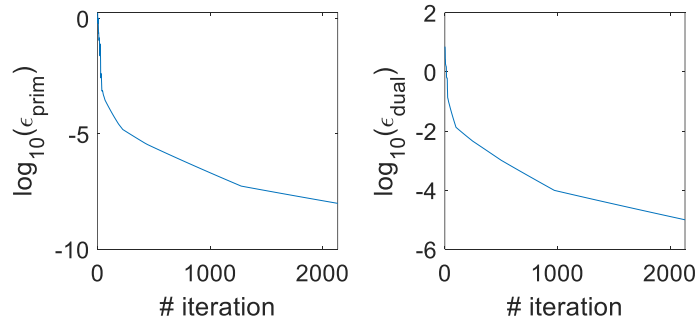


Figure 13 The convergence of ϵ_{prim} and ϵ_{dual} in all iterations.

After the estimation of $\mathcal{B}_0, \mathcal{B}$ are obtained, we observe that the estimation of $\hat{\mathbf{B}}_{ij,k}$ and $\hat{\mathbf{B}}_{k0}$ are either multiple smooth curves when $k = 1, 2$ or smooth images when $k = 3, 4$, which are consistent with the true system parameters (see Figure 14 for the illustration of the estimation of $\hat{\mathbf{B}}_{10}, \dots, \hat{\mathbf{B}}_{40}$, for example). The estimation of these matrices for one run

is shown in Appendix C. It confirms with our structural assumption, i.e., the estimated parametric matrices satisfy the corresponding smoothness property.

Table 3 Summary of the simulation settings

-
- Specifies r_k and $q_{e,k}$ according to Setting (1)-(4).
 - Given each specification of $\{r_k, q_{e,k}: k = 1, \dots, K\}$, generate 30 sets of $\{\mathcal{B}_0, \mathcal{B}\}$ according to the procedure detailed in Appendix B.
 - Generate 10 sets of inputs $\{\mathbf{u}_1, \dots, \mathbf{u}_K\}$ whose elements are all independent and follow standard normal distributions, and generate the error $\{\mathbf{E}_1, \dots, \mathbf{E}_K\}$ whose elements are all independent and follow $N(0, \sigma_{E,k}^2)$, $k = 1, \dots, K$.
 - Given each specification of $\{\mathcal{B}_0, \mathcal{B}\}$, the set of $\{\mathbf{u}_1, \dots, \mathbf{u}_K\}$ and the error $\{\mathbf{E}_1, \dots, \mathbf{E}_K\}$, simulate 10 datasets, each contains $\mathcal{Y}^{\{n\}} = (\mathbf{Y}_1^{\{n\}}, \dots, \mathbf{Y}_K^{\{n\}})$, $n = 1, \dots, N$, based on Equation (1). The sample size $N = 500$.
 - From each dataset, estimate $\widehat{\mathbf{B}}_0$ and $\widehat{\mathbf{B}}$.
-

We then evaluate the effectiveness of the root cause analysis based on the estimations. Specifically, we (1) evaluate their estimation accuracies based on the difference between each estimated parametric matrix $\widehat{\mathbf{B}}_{ij,k}$ or $\widehat{\mathbf{B}}_{k0}$ and their corresponding true value, $\mathbf{B}_{ij,k}$ or \mathbf{B}_{k0} ; (2) identify the effective inputs from each stage by checking whether the entries associated with each u_{ij} of parameter set $\mathbf{B}_{ij,\cdot}^{(3)}$ is non-zero; (3) identify the number of variation patterns of each stage through the number of positive singular values of the estimated $\widehat{\mathbf{B}}_{\cdot,\cdot,k}^{(4)}$ [61].

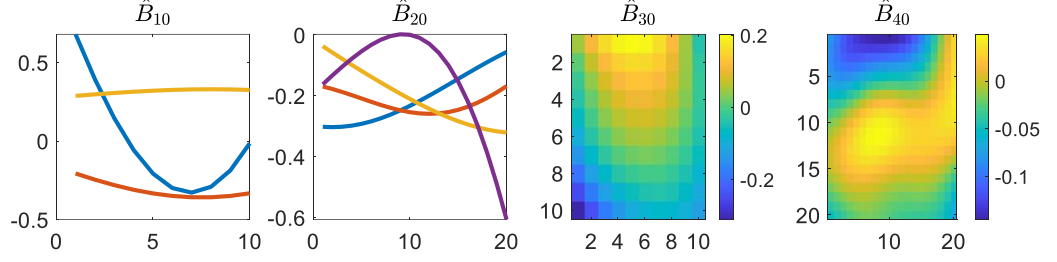


Figure 14. An illustration of an estimation of $\hat{\mathbf{B}}_{10}$, $\hat{\mathbf{B}}_{20}$, $\hat{\mathbf{B}}_{30}$ and $\hat{\mathbf{B}}_{40}$ from one dataset.

3.4.3.1 The estimation accuracy

From the estimation $\hat{\mathbf{B}}_{i,j,k}$ obtained from each dataset, we calculate $d_{i,k} =$

$$\sqrt{\frac{1}{q_i m_k n_k} \sum_{j=1}^{q_i} \|\mathbf{B}_{i,j,k} - \hat{\mathbf{B}}_{i,j,k}\|_F^2} \text{ and } d_{k0} = \sqrt{\frac{1}{m_k n_k} \|\mathbf{B}_{k0} - \hat{\mathbf{B}}_{k0}\|_F^2}$$

to evaluate the estimation error. These quantities correspond to the rooted mean square error associated with every element of the estimated matrices. Under settings (1) to (4), let $d_{i,k}(n_{\text{par}}, n_{\text{rep}})$ and $d_{k0}(n_{\text{par}}, n_{\text{rep}})$ be the values of $d_{i,k}$ and d_{k0} calculated from the dataset according to the n_{par} -th generation of $\{\mathcal{B}_0, \mathcal{B}\}$ and n_{rep} -th generation of inputs and random errors ($n_{\text{par}} \in \{1, \dots, 30\}, n_{\text{rep}} \in \{1, \dots, 10\}$). To understand the average estimation accuracy within each setting and the uncertainty of the estimation error, we further calculate the following summary statistics:

(1) The average error of the setting $\hat{\mu} = \hat{\mathbb{E}}_{n_{\text{par}}} \hat{\mathbb{E}}_{n_{\text{rep}}} [d_{i,k}(n_{\text{par}}, n_{\text{rep}})];$

(2) The variability of the error caused by inputs and random error uncertainty in

$$\text{replications } \hat{\sigma}_{\text{rep}} = \sqrt{\hat{\mathbb{E}}_{n_{\text{par}}} \widehat{\text{var}}_{n_{\text{rep}}} [d_{i,k}(n_{\text{par}}, n_{\text{rep}})]};$$

(3) The variability of the error caused by different generations of parameters $\{\mathcal{B}_0, \mathcal{B}\}$

$$\hat{\sigma}_{\text{par}} = \sqrt{\widehat{\text{var}}_{n_{\text{par}}} \widehat{\text{E}}_{n_{\text{rep}}} [d_{i,k}(n_{\text{par}}, n_{\text{rep}})]},$$

where $\widehat{\text{E}}_{n_{\text{rep}}}, \widehat{\text{E}}_{n_{\text{par}}}$ denotes the average of the following expression for $n_{\text{rep}} = 1, \dots, 10$ or $n_{\text{par}} = 1, \dots, 30$ respectively, and $\widehat{\text{var}}_{n_{\text{rep}}}, \widehat{\text{var}}_{n_{\text{par}}}$ denotes the sample variance of the following expression for $n_{\text{rep}} = 1, \dots, 10$ or $n_{\text{par}} = 1, \dots, 30$ respectively. The summarizing statistics of $\hat{\mu}, \hat{\sigma}_{\text{rep}}$ and $\hat{\sigma}_{\text{par}}$ of all $d_{i,k}$'s and d_{k0}^2 's in four system settings are reported in Table 4 and Table 5. In each cell of this table, the number outside the bracket is the value of $\hat{\mu}$ corresponding to setup 1, ..., 4, and the two numbers separated by the slash in each bracket are $\hat{\sigma}_{\text{rep}}$ and $\hat{\sigma}_{\text{par}}$ that respectively quantifies the uncertainty caused by the inputs and error, and the uncertainty caused by different generations of $\{\mathcal{B}, \mathcal{B}_0\}$.

Table 4 The estimation error $d_{i,k}^2$ ($1 \leq i \leq k \leq 4$) and the associated σ_{rep} and σ_{par} in brackets

Setup 1: $r_k = 2, q_{e,k} = 3$				
	$k = 1$	$k = 2$	$k = 3$	$k = 4$
$i = 1$	1.36e-04 (2.03e-10 / 2.63e-10)	1.37e-04 (8.39e-09 / 4.57e-09)	1.41e-04 (2.87e-08 / 6.98e-08)	2.78e-04 (5.42e-06 / 2.71e-06)
$i = 2$		1.37e-04 (1.10e-08 / 7.70e-09)	1.45e-04 (1.74e-07 / 1.40e-07)	6.18e-04 (1.81e-05 / 1.12e-05)
$i = 3$			1.54e-04 (7.16e-07 / 3.33e-07)	3.39e-03 (1.95e-04 / 8.61e-05)
$i = 4$				1.14e-01 (2.57e-03 / 6.59e-03)
Setup 2: $r_k = 5, q_{e,k} = 3$				
$i = 1$	1.36e-04 (2.77e-10 / 1.73e-10)	1.37e-04 (3.67e-09 / 2.83e-09)	1.41e-04 (2.43e-07 / 4.81e-08)	2.83e-04 (5.06e-06 / 1.84e-06)
$i = 2$		1.37e-04 (1.15e-08 / 6.93e-09)	1.45e-04 (1.75e-07 / 1.31e-07)	6.67e-04 (2.00e-05 / 9.90e-06)
$i = 3$			1.56e-04 (4.27e-07 / 2.01e-07)	3.46e-03 (2.64e-04 / 7.32e-05)
$i = 4$				1.16e-01 (8.28e-03 / 5.54e-03)
Setup 3: $r_k = 2, q_{e,k} = 6$				
$i = 1$	1.36e-04 (2.23e-10 / 2.84e-10)	1.37e-04 (3.67e-09 / 2.91e-09)	1.41e-04 (1.09e-07 / 9.50e-08)	2.89e-04 (3.49e-06 / 2.35e-06)
$i = 2$		1.37e-04 (7.48e-09 / 6.30e-09)	1.45e-04 (2.31e-07 / 1.78e-07)	6.59e-04 (3.27e-05 / 1.26e-05)
$i = 3$			1.56e-04 (5.69e-07 / 2.77e-07)	3.74e-03 (1.81e-04 / 1.10e-04)
$i = 4$				1.46e-01 (7.72e-03 / 8.26e-03)
Setup 4: $r_k = 5, q_{e,k} = 6$				
$i = 1$	1.36e-04 (1.88e-10 / 1.65e-10)	1.37e-04 (2.56e-09 / 2.92e-09)	1.41e-04 (1.90e-07 / 4.65e-08)	2.93e-04 (1.45e-06 / 1.43e-06)
$i = 2$		1.37e-04 (8.59e-09 / 5.00e-09)	1.46e-04 (2.46e-07 / 9.88e-08)	6.76e-04 (2.53e-05 / 8.24e-06)
$i = 3$			1.57e-04 (5.00e-07 / 2.29e-07)	3.92e-03 (2.74e-04 / 7.30e-05)
$i = 4$				1.52e-01 (5.43e-03 / 6.03e-03)

Table 5 The estimation error of d_{k0}^2 ($1 \leq k \leq 4$) and the associated σ_{rep} and σ_{par} in brackets

Setup 1: $r = 2, q_e = 3$			
5.82e-03	6.11e-03	9.58e-03	3.17e-02
(2.38e-04 / 1.12e-03)	(7.79e-04 / 1.87e-03)	(1.48e-03 / 4.56e-03)	(2.74e-03 / 1.58e-02)
Setup 2: $r = 5, q_e = 3$			
5.94e-03	8.12e-03	1.02e-02	3.49e-02
(4.40e-04 / 1.06e-03)	(8.31e-04 / 2.93e-03)	(1.02e-03 / 3.48e-03)	(2.98e-03 / 1.32e-02)
Setup 3: $r = 2, q_e = 6$			
6.11e-03	7.57e-03	8.88e-03	3.57e-02
(4.75e-04 / 9.40e-04)	(8.80e-04 / 2.60e-03)	(1.34e-03 / 4.35e-03)	(3.46e-03 / 1.92e-02)
Setup 4: $r = 5, q_e = 6$			
6.05e-03	8.70e-03	1.31e-02	4.36e-02
(2.18e-04 / 9.78e-04)	(6.42e-04 / 2.13e-03)	(1.02e-03 / 5.18e-03)	(6.24e-03 / 1.92e-02)

From the results in the table, we can observe that the errors are small in general.

We summarize the following findings:

- Among all parametric matrices, the effects of the inputs from stage 1 on the output of stage 1 is estimated most accurately. The magnitude of the error is in the order of 10^{-4} . The reason is that the outputs from stage 1 are only related to the inputs from stage 1, and therefore the relationship between them is clear. Also, the smoothness penalty regularized the matrices of estimation, and thus increase the estimation accuracy.
- As k increases from 1 to 4, the error associated with the estimation $\widehat{\mathbf{B}}_{ij,k}$ generally increases. One of the reasons is that the total number of elements in $\mathbf{B}_{ij,k}$ increases. (Note that when $k = 1, 2, 3$ and 4 , the number of elements in $\mathbf{B}_{ij,k}$ is respectively 30, 80, 100, 400). Apart from this, outputs from later stages are associated with more input variables, and thus the estimation accuracy decreases. Finally, later stages involve larger penalization due to the ranks, as $\lambda_{4,4} > \lambda_{4,3} > \lambda_{4,2} > \lambda_{4,1}$ in our setup. Although such selection of the hyper-parameters is necessary to reduce the ranks of $\mathbf{B}_{\cdot,k}$

involving later stages k containing more elements and associated inputs, it also introduces larger biases for the estimations in later stages.

- When k is fixed, the estimation of $\mathbf{B}_{ij,k}$ become less accurate when i increases.
- Among the four settings, the estimation is more accurate when $r = 2$ and less accurate when $r = 5$. The estimation is also more accurate when $q_e = 3$ and less accurate when $q_e = 6$. This is because our penalization works best when the number of variation patterns of each stage and the number of effective inputs from each stage is not high.

Except for the observations above, we can also see that $\hat{\sigma}_{\text{rep}}, \hat{\sigma}_{\text{par}}$ are typically much smaller than the error $\hat{\mu}$, which indicates that the uncertainty of the estimation error is not high and that the discoveries above are conclusive instead of merely out of chance.

3.4.3.2 Effective inputs

Among simulation setups (1) to (4), the number of effective inputs from each stage is either 3 or 6. Of $30 \times 10 = 300$ replicates corresponding to four setups, all inputs are correctly identified as effective or ineffective ones from stages 1, 2 and 3. In stage 4, on average 0.28 ineffective inputs are falsely selected as effective one under setup 2, 0.18 ineffective inputs are falsely selected as effective one under setup 3 and 0.28 ineffective inputs are falsely selected as effective one under setup 4. This result indicates that the proposed framework generally identifies the significant variables in early stages if the sample size is large enough, the error is not so big, and the hyperparameters are appropriately chosen. However, in later stages like stage 4, the proposed framework is likely to select extra ineffective inputs, because the total number of inputs that may affect this stage is too big (including

all input from the current stage and the previous stages). Consequently, the algorithm is more prone to selecting ineffective variables.

3.4.3.3 Number of variation patterns

Within four simulation setups, the number of variation patterns from each stage is either 2 or 5. Within each simulation setup, we calculate $r(n_{\text{par}}, n_{\text{rep}})$, the rank of $\widehat{\mathbf{B}}_{\cdot, k}^{(4)}$ corresponding to the n_{par} th generation of $\{\mathcal{B}, \mathcal{B}_0\}$ and the n_{rep} th generation of inputs and errors ($n_{\text{par}} \in \{1, \dots, 30\}, n_{\text{rep}} \in \{1, \dots, 10\}$). Similar to the report of estimation accuracy, the average rank among 300 simulation cases ($\hat{\mu}_r$) and their uncertainties $\hat{\sigma}_{\text{par}, r}, \hat{\sigma}_{\text{rep}, r}$ are reported in Table 6.

From the result, we observe that when the true number of variation patterns for the output in each stage is $r = 2$, the algorithm can always correctly identify two variation patterns (because $\hat{\sigma}_{\text{par}} = \hat{\sigma}_{\text{rep}} = 0$ indicates that $r(n_{\text{par}}, n_{\text{rep}})$ are the same for all $n_{\text{par}}, n_{\text{rep}} = 1, \dots, 10$). However, in setup 2 where $q_{e,k} = 3, r_k = 5$, the average rank of $\mathbf{B}_{\cdot, k}^{(4)}$ is 3, 4.60, 4.38 and 4.42 for stages 1, 2, 3 and 4, according to Table 6. The number of the variation patterns for the output from stage 1 is correctly given, as this stage is only affected by 3 inputs from itself and the number of variation patterns cannot exceed 3. The later stages 2, 3 and 4 are influenced by 6, 9 and 12 inputs respectively, but the effects of them are restricted in a 5-dimensional subspace. However, the algorithm does not always identify the rank as 5. We guess that the reason is two-fold: (1) collinearity may exist among the randomly generated 5 variation patterns, and (2) the number of inputs is small to reveal all variation patterns. Under setup 4, $q_{e,k} = 6$ and thus the number of inputs is

larger. As a result, all replications in setup 4 identify that rank of $\mathbf{B}_{\cdot,k}^{(4)}$ is 5 and thus the number of the variation patterns is correctly estimated in stages 2 and 4. In stages 1 and 3, the average rank of $\mathbf{B}_{\cdot,k}^{(4)}$ are 4.60 and 4.80 across all replications. Although error exists, the estimation is closer to the correct value 5 than the results in setup (2).

Table 6 Number of variation patterns and the associated σ_{rep} and σ_{par} in brackets

Setup 1: $r_k = 2, q_{e,k} = 3$			
2 (0 / 0)	2 (0 / 0)	2 (0 / 0)	2 (0 / 0)
Setup 2: $r_k = 5, q_{e,k} = 3$			
3 (0 / 0)	4.60 (0.548 / 0)	4.38 (0.522 / 0.141)	4.42 (0.814 / 0.287)
Setup 3: $r_k = 2, q_{e,k} = 6$			
2 (0 / 0)	2 (0 / 0)	2 (0 / 0)	2 (0 / 0)
Setup 4: $r_k = 5, q_{e,k} = 6$			
4.60 (0.548 / 0)	5 (0 / 0)	4.80 (0.447 / 0)	5 (0 / 0)

3.5 Summary

In a data-rich manufacturing environment, an MMP generates various types of data from different manufacturing stages, which poses a great challenge for data analytics. In this chapter, we propose a novel root cause diagnostic framework for an MMP that satisfies four assumptions: (1) the input from one stage only affect the down-stream stages (e.g., no re-work), (2) the process outputs satisfies smoothness properties, (3) only a small number of inputs affect the process outputs, and (4) the variation patterns caused by the inputs are limited. Based on these assumptions, our approach identifies the effective inputs that relate to the perturbation of the outputs, identifies the variation patterns of the outputs caused by

these inputs, and determines how each individual process input affects the manufacturing process.

The root cause diagnostic framework is based on a model for MMPs that generates mixed profile data, such as functional signals or images. For estimating the model parameters, we proposed a distributed computational scheme. The framework proposed in this chapter is highly extendable: the practitioners may customize it based on the special characteristics of the process, by using appropriate loss functions and structural assumptions on various types of data generated from different stages.

We developed the simulation study based on the scenario of a real semiconductor manufacturing process. In general, with correctly specified tuning parameters, the proposed method can perform well for three tasks of root cause diagnosis: it achieves satisfactory estimation accuracy, can correctly identify the inputs that affect the outputs, and provide a good estimation of the number of variation patterns for the output from each stage.

In this study, our modeling of the MMP focus on the application of root cause diagnosis. How to extend this modeling technique to process control, optimization, and sensor allocation are follow-up questions that need to be studied in the future. We will also extend our current framework to tensor inputs and outputs generated from each stage and modeling the inter-relationship between heterogenous intermediate quality measurement.

CHAPTER 4. MULTIPLE EVENT IDENTIFICATION AND CHARACTERIZATION BY RETROSPECTIVE ANALYSIS OF STRUCTURED DATA STREAMS

4.1 Introduction

In various industries, practitioners increasingly install multiple sensors in complex systems to understand their process conditions. These sensors generate sequences of profile sensing signals data in the form of structured data streams [70, 71]. They contain rich information about the system components and record the system status during its operation. Before using these sensor measurements for real-time monitoring and control of the process, the practitioners need to collect and review the historical data obtained from a period of time. Through the retrospective analysis of the process data, the engineers can gain insight into the process variation in a longer time scale for discovering new root causes.

For a given system, the sensing signals are subject to the impact of multiple system operation conditions. When the system is operating under normal operation conditions, there is a baseline predictable pattern of the sensing signals. However, various faults may occur in the system at particular intervals during the system operation and lead to the changes of associated sensing signals in specific patterns. We refer to those changes related to the same fault as *one event*. When an event occurs in the system, several associated sensing signals will change according to a specific variation pattern. We refer to the variation pattern associated with each event as *event signatures*. In the following examples

of systems, we further illustrate the concepts of events and the corresponding event signatures.

- (a) In the stamping process considered in Jin and Shi [70], the tonnage signals are composed of multiple segments, corresponding to phases of operations or mechanical interactions within one stamping cycle. In this system, each event relates to a specific fault in the stamping process, such as material thickness error, loose tie rods, and worn bushings [72]. Those events lead to respective changes in corresponding segments of tonnage signals, and these changes are event signatures.
- (b) Phasor measurement units, such as frequency disturbance recorders (FDR), are deployed in power grids to achieve situation awareness [73]. There are multiple events that may occur to the grid and affect the associated PMU measurements. For example, the event of generator tripping (or load shedding) may cause a decrease (or an increase) of the FDR signal. The event of line tripping may lead to the damping waveform signal. These effects on the sensor measurements are event signatures.
- (c) In a node of an interconnected cyber system, the usage of CPU, memory, disk, network bandwidth, and power usage are recorded as they reflect the node's working conditions. Examples of events on this node include regular operations (like processing, uploading, and downloading) and abnormal situations (like virus infection or port-scan attacks). According to their event signatures, they cause different behavior on all cyber signals [74].

The systems discussed above have three common characteristics: First, each system generates one or more events that may occur during the system operations. Second, during the time of operations, each event appears sporadically, and each occurrence of the event

tends to last a period of time after it begins. Third, each event is associated with a small and limited number of sensing signals. In practice, there is a large class of systems holding these characteristics in terms of associated events and their event signatures.

This study aims at developing automatic retrospective analysis methods for the signals generated from the systems with those three characteristics. Our objectives are two-fold: First, we aim at characterizing the event signatures that specify how each event affects the signals. This information enables us to extract useful features from the process and perform online monitoring. Second, we aim to identify the periods in which an event occurs and estimate the event's strengths. Both information shed light on the root cause diagnosis of those events.

In literature, some studies tackled similar data analysis problems. However, as discussed in the next literature review section, the existing methods have some limitations in addressing the above two questions simultaneously. For example, some algorithms require pre-defined prototype signals containing individual events apart from the historical data, which involves extra data preprocessing and human labeling efforts [73]. Other methods consider the signal partition problem, while they cannot associate segments with events or generate useful event signatures [75, 76]. Various Phase-I retrospective analysis of the sensing data does not apply to structured data streams. They identify the out-of-control samples without considering either multiple event sequences or their signatures.

This chapter proposes an algorithm that simultaneously *identifies* the events' periods of occurrence and *characterizes* each event with its signature, including the signals associated with each event. This algorithm is called the Multiple Event Identification and

Characterization (MEIC) algorithm. In practice, an event captured by the MEIC algorithm is typically related to some faults that occurred in the process. The identified event signatures will be helpful in finding the root causes of the faults. Therefore, the algorithm serves as an automatic tool that gives practitioners hints to discovering new root causes of the processes.

The MEIC algorithm works by solving an optimization problem that integrates dictionary learning technique [5] with regularization terms specifying the sparsity of the related signals and the temporal smoothness of event strengths. By solving this optimization problem, we can extract useful representation from sensing signals and identify the occurrence of each event. As will be seen from the simulation study and the real case study, the MEIC algorithm performs well if we start it with multiple initial points to avoid suboptimal solutions.

The remaining part of the chapter is organized as follows. In Section 4.2, we review the related literature in greater detail. Section 4.3 proposes the system model, introduces the formulation of the optimization problem, and gives the solution algorithms. After that, we present our simulation study and real case study based on a steel rolling process in Section 4.4 and Section 4.5. The conclusion is given in Section 4.6.

4.2 Literature Review

Identifying underlying events from sensing data has been reported in the literature for decades. For example, Jin and Shi [70] proposed wavelet-based criteria to extract the events-related information from tonnage signals and identify the events through monitoring the compressed coefficients. However, this method focuses on wavelet analysis of tonnage

signals in each stamping cycle. Although they can identify whether an event happens in a stamping cycle, additional efforts are required to associating multiple stamping cycles to a set of events and identify the periods of those events that affect multiple consecutive products. Wang, et al. [73] developed a situation awareness system that recognizes the events from the data generated from phasor measurement units. They proposed a two-step analysis method: first use the k -means clustering approach to form a dictionary of the event signatures, and then formulate an ℓ_1 -penalization approach to identify the offset of the event within the signal. The main drawback of this approach is that it requires to have a dataset of observations corresponding to each single event. Another related method of identifying the events from the structured data streams is using phase partition algorithms. These partition algorithms transform the data within each time window into low-dimensional features. For example, Zhao [76] proposed to calculate the residual of the partial least square (PLS) regression as features for phase partition, and Guo, et al. [75] extracted the covariance matrices of sensor measurements within the window. They then used ad-hoc heuristics or greedy algorithms to find the partition points using these low-dimensional features. Note that these algorithms do not associate each interval with a small set of events and do not consider the possibility of overlapping events during the system operations or the variability of the events' strength.

We also note that the event detection and characterization problem has a close relationship with the Phase-I control chart and root cause diagnosis. One can view the events during the operation of the system as assignable causes that lead to extra variability of the system and regard the samples affected by these events as being out of control. In this sense, identifying the occurrence of events can be achieved by a Phase-I control chart.

Characterizing the event signatures is then a follow-up diagnostic procedure that associates the out-of-control samples with a small set of root causes. Phase-I monitoring for multiple signal data has been identified as an emerging area of statistical process control [77]. Among the few studies in literature, Wang, et al. [78] monitors the principal component scores under a change-point framework. Although they noted the importance of taking the out-of-control information into account when designing control schemes, they did not assume multiple root causes leading to different out-of-control scenarios. Ebrahimi, et al. [79] pointed out that there is a lack of literature on scalable and integrated monitoring and diagnosis approach in the current Phase-I charting schemes. Although they proposed a seamless monitoring and diagnosis framework, they performed the event identification (monitoring) step and the event signature characterization (diagnosis) step sequentially, which hinders the utilization of out-of-control information in the control chart design. In the next section, we will see that the approach proposed in this chapter solves the monitoring and diagnostic problem simultaneously and interactively from one formulation. Finally, we note that control charts usually assume a simple probabilistic description of the system. The out-of-control situations are generally simple, which enable probabilistic quantifications of charts' performance. However, they cannot describe systems with complex event signatures and strength profiles.

The MEIC algorithm we proposed identifies the events associated with each sample and estimates all event signatures. In literature, prototype methods such as k -means clustering and Gaussian Mixture Models [80] achieve a similar goal. However, the difference is that prototype methods do not consider the temporal sequence of the samples when assigning them to different events. As an alternative, we will use the dictionary

learning technique [81] to develop the MEIC algorithm. Unlike the k -means or Gaussian Mixture Model, the dictionary learning method identifies events and describes their effect by directly formulating an optimization criterion. It enables us to incorporate the characteristics of the event signatures and event strength sequences through multiple penalization terms on the associated parameters [59, 82, 83].

4.3 The Method of Multiple Event Identification and Characterization

In this section, we first present the assumptions on the system and the collected data. In each time point t from 1 to T , we obtain a sample containing I signals, and the i -th signal is of length S_i . The s -th measurement obtained from signal i at time t is denoted as $x_{t,i}(s)$, $s = 1, \dots, S_i; i = 1, \dots, I; t = 1, \dots, T$. During this period of time, K events may occur with possible overlaps among them, and let $y_{k,t} \geq 0$ represents the strength of the event k at time t . With the assumption that the events appear, stay, and fade off gradually, $y_{k,t}$ is smooth with the change of time t from 1 to T . The collected data $x_{t,i}(s)$'s and the unknown strengths of underlying events $y_{k,t}$'s are illustrated in Figure 15. In this figure, the background's gray level of each $y_{k,t}$ indicates the magnitude of this value. When no events occur at time t , we assume that the signals $x_{t,1}(s), \dots, x_{t,I}(s)$ are independent and identically distributed. This assumption typically holds if the preprocessing step transforms the sensor signals to residuals from each raw signal by subtracting the fixed or predicted trends.

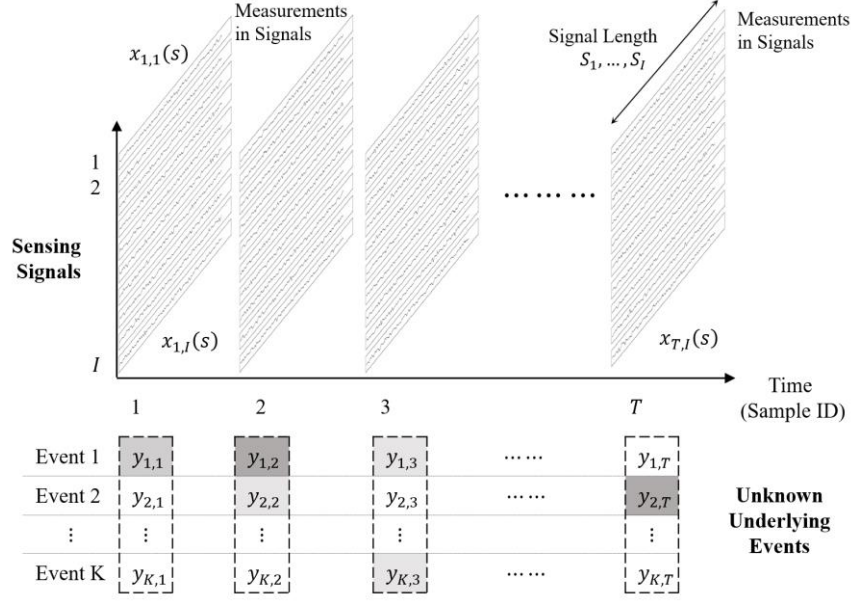


Figure 15 The collected data $x_{t,i}(s)$'s and the strengths of the unknown underlying events $y_{k,t}$'s.

We say that the event k occurs at a time t when $y_{k,t} > 0$. In Figure 15, the background of $y_{k,t}$ is not white. When event k occurs, each signal i contains a smooth variation pattern $\xi_{k,i}(s)$, $i = 1, \dots, I$, and the collection of $\{\xi_{k,i}(s): i = 1, \dots, I\}$ is the signature of event k . Recall that each event is associated with few sensing signals, we have $\xi_{k,i}(s) \equiv 0$ for most i 's. Under the assumption that all events have additive effects on the signals, we thus represent the signal obtained at time t as $x_{t,i}(s) = \sum_{k=1}^K \xi_{k,i}(s)y_{k,t} + \epsilon_{t,i}(s)$.

The measurements from sensor i can be aggregated in a matrix $\mathbf{X}_i \in \mathbb{R}^{S_i \times T}$, with $(\mathbf{X}_i)_{s,t} = x_{t,i}(s)$. All sensor measurements then constitute a data matrix $\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_I \end{bmatrix} \in \mathbb{R}^{S \times T}$, where $S = S_1 + \dots + S_I$ is the total number of measurements from I sensor signals. We have two goals from the inference from \mathbf{X} :

1. Identify the periods that each event k occurs, and estimate the strength profiles. We achieve this by estimating $y_{k,t}$ for all events $k = 1, \dots, K$ at all time points $t = 1, \dots, T$.
2. Characterize each event by its event signature $\xi_{k,i}(s)$ for $k = 1, \dots, K$ on all signals $i = 1, \dots, I$. We achieve this by estimating $\xi_{k,i}(s), k = 1, \dots, K; i = 1, \dots, I$, and $s = 1, \dots, S_i$.

4.3.1 Problem formulation

To facilitate the estimation procedure, we represent each event signature $\xi_{k,i}(s)$ with the wavelet basis $\{h_j(s): j = 1, \dots, J_i\}$,

$$\xi_{k,i}(s) = \sum_{j=1}^{J_i} b_{k,i,j} h_j(s).$$

In this representation, the effect of event k on signal i is a vector of length J_i , $\mathbf{b}_{k,i} = (b_{k,i,1}, \dots, b_{k,i,J_i})^\top$. The event signatures are $\boldsymbol{\xi}_{k,i} = [\xi_{k,i}(1), \dots, \xi_{k,i}(S_i)]^\top$ and we have $\boldsymbol{\xi}_{k,i} = \mathbf{H}_i \mathbf{b}_{k,i}$ with $\mathbf{H}_i = [h_j(s)]_{\substack{s=1, \dots, S_i \\ j=1, \dots, J_i}}$. The overall effect on signal i at time t is then

$$\hat{x}_{t,i}(s) = \sum_{k=1}^K \xi_{k,i}(s) y_k(t), s = 1, \dots, S_i.$$

Represent it in the matrix form, we have $\hat{\mathbf{X}} = \mathbf{H}\mathbf{B}\mathbf{Y}$, where $\mathbf{H} = \begin{bmatrix} \mathbf{H}_1 & & \\ & \ddots & \\ & & \mathbf{H}_I \end{bmatrix} \in \mathbb{R}^{S \times J}$ is

the collection of all basis, $\mathbf{B} = \begin{bmatrix} \mathbf{b}_{1,1} & \cdots & \mathbf{b}_{K,1} \\ \vdots & \ddots & \vdots \\ \mathbf{b}_{1,I} & \cdots & \mathbf{b}_{K,I} \end{bmatrix} \in \mathbb{R}^{J \times K}$ contains all coefficients that

determine the event signatures, and $\mathbf{Y} = (y_{k,t})_{K \times T}$ contains the strengths of all events at

all time points, $J = \sum_{i=1}^I J_i$. The matrix $\widehat{\mathbf{X}} = \begin{bmatrix} \widehat{\mathbf{X}}_1 \\ \vdots \\ \widehat{\mathbf{X}}_I \end{bmatrix} \in \mathbb{R}^{S \times T}$ with $\widehat{\mathbf{X}}_i = (\hat{x}_{t,i}(s))_{s_i \times T}$.

Now, we aim at formulating an optimization problem to solve the values of \mathbf{B} and \mathbf{Y} , which respectively characterize the event signatures and the strength of the events during the T time points. First, the matrix $\widehat{\mathbf{X}}$ provides an approximation to the data matrix \mathbf{X} , and we define the loss as $\|\mathbf{X} - \widehat{\mathbf{X}}\|_F^2$. Besides, we add the following penalization terms to represent the event signatures and the sequence of event strengths.

The event signatures. The wavelet basis usually gives an overcomplete representation of the signals. Motivated by the wavelet shrinkage method [84], we first apply an ℓ_1 regularization $\lambda_1 \|\mathbf{B}\|_{1,1}$ to improve the estimation of the event signature $\xi_{k,i}(s)$'s through overcoming the curse of dimensionality. Recall that the $\xi_{k,i} = \mathbf{0}$ for most event k and signal i , because each event is associated with few signals. Therefore, we have $\mathbf{b}_{k,i} = \mathbf{0}$ for most event k and signal i . We thus add another group lasso penalty $\lambda_2 \sum_{i=1}^I \sum_{k=1}^K \|\mathbf{b}_{k,i}\|_2$.

The sequence of event strengths. We assume that events have continuity property, meaning that they appear, vary, and disappear gradually from time 1 to time T . Therefore,

we apply the smoothness penalty $\lambda_3 \|\mathcal{D}\mathbf{Y}^\top\|_F^2$, where $\mathcal{D} = \begin{bmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{bmatrix}$ is

the second-order smoother that applies to the temporal mode of \mathbf{Y} . Furthermore, every event only occurs sporadically in time, and therefore we add the ℓ_1 -penalty $\lambda_4 \|\mathbf{Y}\|_{1,1}$.

Integrating the loss function and all penalties mentioned above, we derive the following optimization problem

$$\min_{\mathbf{B}, \mathbf{Y}} \|\mathbf{X} - \mathbf{H}\mathbf{B}\mathbf{Y}\|_F^2 + \lambda_1 \|\mathbf{B}\|_{1,1} + \lambda_2 \sum_{i=1}^I \sum_{k=1}^K \|\mathbf{b}_{k,i}\|_2 + \lambda_3 \|\mathcal{D}\mathbf{Y}^\top\|_F^2 + \lambda_4 \|\mathbf{Y}\|_{1,1} \quad (17)$$

$$\text{subject to } \|\mathbf{b}_{k,\cdot}\|_2 = 1, y_{k,t} \geq 0, k = 1, \dots, K; t = 1, \dots, T,$$

where $\mathbf{Y} = [\mathbf{y}_{\cdot,1}, \dots, \mathbf{y}_{\cdot,T}]$. Note that we added another constraint $\|\mathbf{b}_{k,\cdot}\|_2 = 1$ in this

formulation, where $\mathbf{b}_{k,\cdot} = \begin{bmatrix} \mathbf{b}_{k,1} \\ \vdots \\ \mathbf{b}_{k,I} \end{bmatrix}$ is the coefficient vector of event k corresponding to all

signals $1, \dots, I$. Due to the orthogonality of \mathbf{H} , this condition indicates that $\|\boldsymbol{\xi}_{k,\cdot}\|_F^2 = \sum_{i=1}^I \|\boldsymbol{\xi}_{k,i}\|_F^2 = 1$. Essentially, it keeps the scales of all event signatures the same and specifies the unit for measuring the strengths of each event.

Problem (17) is motivated by the dictionary learning problem [81], where \mathbf{B} and \mathbf{Y} simultaneously give a K -dimensional representation of the historical data \mathbf{X} . In our problem setting, the matrix \mathbf{B} represents the wavelet coefficients that defines the events, and \mathbf{Y} represents the sequences of the events' strengths.

Based on the solution of \mathbf{B} and \mathbf{Y} , we can answer the two questions discussed at the beginning of this section. For every event $k = 1, \dots, K$, we can obtain $\mathbf{b}_{k,\cdot}$, which shows

the effect of this event on the signal i as $\xi_{k,i}(s) = \sum_{j=1}^{J_i} b_{k,i,j} h_j(s)$. Also, from $\mathbf{y}_{k,\cdot} \in \mathbb{R}^T$, the k th row of \mathbf{Y} , we can identify the time that the event k occurs with considerable strengths and the time at which event k does not occur. From this perspective, this framework performs events characterization (diagnostics) and event identification (off-line detection) simultaneously. The readers should note that the penalizations may perform two functions here. First, the minimization of the loss involves many coefficients in the matrix \mathbf{B} and \mathbf{Y} . Regularizations make it possible to obtain a solution that satisfies the sparsity and smoothness conditions and improves the estimation accuracy. The term $\lambda_4 \|\mathbf{Y}\|_{1,1}$ enables us to identify the time points that each event occurs by observing if $y_{k,t} = 0$. In this sense, choosing λ_2 and λ_4 enables the practitioners to adjust the number of sensors associated with each event and the number of time points at which the event occurs to facilitate the root cause diagnosis of the system.

Algorithm 7: Multiple Events Identification and Characterization (MEIC) algorithm

Initiate $\mathbf{Y} = \mathbf{Y}^0$.

Loop until converge:

 Update \mathbf{B} given \mathbf{Y} , as detailed in Section 4.3.2.1.

 Update \mathbf{Y} given \mathbf{B} , as detailed in Section 4.3.2.2

4.3.2 Solution algorithms

Problem (17) can be solved through a blockwise coordinate descent (BCD) algorithm, where we iteratively update the matrix \mathbf{B} and the matrix \mathbf{Y} , as shown in Algorithm 7. The steps of updating \mathbf{B} and updating \mathbf{Y} are performed with two ADMM algorithms, detailed in the following two subsections. In Section 4.3.2.3, we discuss how to select the initial value of \mathbf{Y}^0 .

4.3.2.1 Updating \mathbf{B}

To update \mathbf{B} , we need to solve Problem (18).

$$\min_{\mathbf{B}} \|\mathbf{X} - \mathbf{H}\mathbf{B}\mathbf{Y}\|_F^2 + \lambda_1 \|\mathbf{B}\|_{1,1} + \lambda_2 \sum_{i=1}^I \sum_{k=1}^K \|\mathbf{b}_{k,i}\|_2 \quad (18)$$

subject to $\|\mathbf{b}_{k,\cdot}\|_2 = 1$.

This problem can be reformulated as

$$\min_{\mathbf{Z}} \sum_{m=1}^M f_m(\mathbf{Z}) \quad (19)$$

where $M = 4$, $f_1(\mathbf{B}) = \|\mathbf{X} - \mathbf{H}\mathbf{B}\mathbf{Y}\|_F^2$, $f_2(\mathbf{B}) = \lambda_1 \|\mathbf{B}\|_{1,1}$, $f_3(\mathbf{B}) = \lambda_2 \sum_{i=1}^I \sum_{k=1}^K \|\mathbf{b}_{k,i}\|_2$, and $f_4(\mathbf{B}) = \sum_{k=1}^K I_{\|\mathbf{b}_{k,\cdot}\|_2=1}$. Problem (19) can be solved with the ADMM consensus algorithm [61], summarized in Algorithm 8 below.

Algorithm 8 ADMM consensus Algorithm

Initiate replicates $\mathbf{Z}^{(m)}, \mathbf{U}^{(m)} = \mathbf{O}$ for $m = 1, \dots, M$, of the same shape as \mathbf{Z} . Set step size η .

Iterate until convergence:

Update $\mathbf{Z}^{(m)} = \text{prox}_{\eta f_m} [\bar{\mathbf{Z}} - \mathbf{U}^{(m)}]$, for $m = 1, \dots, M$ in parallel.

$\bar{\mathbf{Z}} = \sum_{m=1}^M \mathbf{Z}_m / M$

$\mathbf{U}^{(m)} = \mathbf{U}^{(m)} + \mathbf{Z}^{(m)} - \bar{\mathbf{Z}}$.

To implement Algorithm 8 in solving Problem (18), we need to evaluate the proximal operators of $\eta f_1, \dots, \eta f_4$. The results are in Proposition 10, and the derivation is in Appendix 0.

Proposition 10. Let $f_1(\mathbf{B}) = \|\mathbf{X} - \mathbf{H}\mathbf{B}\mathbf{Y}\|_F^2$, $f_2(\mathbf{B}) = \lambda_1 \|\mathbf{B}\|_{1,1}$, $f_3(\mathbf{B}) = \lambda_2 \sum_{i=1}^I \sum_{k=1}^K \|\mathbf{b}_{k,i}\|_2$ and $f_4(\mathbf{B}) = \sum_{k=1}^K I_{\|\mathbf{b}_{k,\cdot}\|_2=1}$. The proximal operators of $\eta f_1, \dots, \eta f_4$ are given as follows: Let \mathbf{A} and \mathbf{Z} have the same size as \mathbf{B} , and partition them into $\mathbf{A} =$

$$\begin{bmatrix} \mathbf{a}_{1,1} & \cdots & \mathbf{a}_{K,1} \\ \vdots & \ddots & \vdots \\ \mathbf{a}_{1,I} & \cdots & \mathbf{a}_{K,I} \end{bmatrix} \text{ and } \mathbf{Z} = \begin{bmatrix} \mathbf{z}_{1,1} & \cdots & \mathbf{z}_{K,1} \\ \vdots & \ddots & \vdots \\ \mathbf{z}_{1,I} & \cdots & \mathbf{z}_{K,I} \end{bmatrix} \text{ according to } \mathbf{B} = \begin{bmatrix} \mathbf{b}_{1,1} & \cdots & \mathbf{b}_{K,1} \\ \vdots & \ddots & \vdots \\ \mathbf{b}_{1,I} & \cdots & \mathbf{b}_{K,I} \end{bmatrix}. \text{ Similarly,}$$

we let $\mathbf{B}_{\cdot,i} = [\mathbf{b}_{1,i} \dots \mathbf{b}_{K,i}]$, $\mathbf{Z}_{\cdot,i} = [\mathbf{z}_{1,i} \dots \mathbf{z}_{K,i}]$, and $\mathbf{A}_{\cdot,i} = [\mathbf{a}_{1,i} \dots \mathbf{a}_{K,i}]$.

1. If $\mathbf{Z} = \text{prox}_{\eta f_1}[\mathbf{A}]$, we have

$$\text{vec}(\mathbf{Z}_{\cdot,i}) = (\mathbf{E}_{KJ_i \times KJ_i} + \eta(\mathbf{Y}\mathbf{Y}^\top) \otimes \mathbf{E}_{J_i \times J_i})^{-1} [\text{vec}(\mathbf{A}_{\cdot,i}) + \eta \text{vec}(\mathbf{H}_i^\top \mathbf{X}_i \mathbf{Y}^\top)],$$

where \otimes is the Kronecker product, and $\mathbf{E}_{r \times r}$ is the identity matrix of order r .

2. If $\mathbf{Z} = \text{prox}_{\eta f_2}[\mathbf{A}]$, $Z_{l,i} = S_{\lambda_1 \eta}(A_{l,i})$, where $Z_{l,i}, A_{l,i}$ are the (l, i) element of \mathbf{Z} and \mathbf{A}

$$\text{respectively, and } S_{\lambda \eta}(x) = \begin{cases} x + \lambda_1 \eta, & x \leq -\lambda_1 \eta \\ 0, & -\lambda_1 \eta \leq x \leq \lambda_1 \eta \\ x - \lambda_1 \eta, & x > \lambda_1 \eta \end{cases}$$

3. If $\mathbf{Z} = \text{prox}_{\eta f_3}[\mathbf{A}]$, $\mathbf{z}_{k,i} = \left(1 - \frac{\eta \lambda_2}{\|\mathbf{a}_{k,i}\|}\right) \mathbf{a}_{k,i}$.

4. If $\mathbf{Z} = \text{prox}_{\eta f_4}[\mathbf{A}]$, $\mathbf{z}_{k,\cdot} = \frac{\mathbf{a}_{k,\cdot}}{\|\mathbf{a}_{k,\cdot}\|}$. Here $\mathbf{a}_{k,\cdot}$ and $\mathbf{z}_{k,\cdot}$ are the k th column of \mathbf{A} and \mathbf{Z} ,

respectively.

4.3.2.2 Updating \mathbf{Y}

In \mathbf{Y} -update, we need to solve Problem (20)

$$\min_{\mathbf{B}, \mathbf{Y}} \|\mathbf{X} - \mathbf{H}\mathbf{B}\mathbf{Y}\|_F^2 + \lambda_3 \sum_{t=1}^T \|\mathcal{D}\mathbf{Y}^\top\|_F^2 + \lambda_4 \|\mathbf{Y}\|_{1,1} \quad (20)$$

subject to $y_{k,t} \geq 0$, $k = 1, \dots, K$; $t = 1, \dots, T$.

Problem (20) is also in the form of Problem (19), with $M = 3$ and $f_1(\mathbf{Y}) = \|\mathbf{X} - \mathbf{H}\mathbf{B}\mathbf{Y}\|_F^2$, $f_2(\mathbf{Y}) = \lambda_3 \sum_{t=1}^T \|\mathcal{D}\mathbf{Y}^\top\|_F^2$ and $f_3(\mathbf{Y}) = \lambda_4 \|\mathbf{Y}\|_{1,1} + \sum_{t=1}^T \sum_{k=1}^K I_{y_{k,t} \geq 0}$. We use Algorithm 8 again to solve the problem, while the proximal operators of the functions are given in Proposition 11. The derivation is given in Appendix 0.

Proposition 11. Let $f_1(\mathbf{Y}) = \|\mathbf{X} - \mathbf{H}\mathbf{B}\mathbf{Y}\|_F^2$, $f_2(\mathbf{Y}) = \lambda_3 \sum_{t=1}^T \|\mathcal{D}\mathbf{Y}^\top\|_F^2$ and $f_3(\mathbf{Y}) = \lambda_4 \|\mathbf{Y}\|_{1,1} + \sum_{t=1}^T \sum_{k=1}^K I_{y_{k,t} \geq 0}$. The proximal operators of $\eta f_1, \dots, \eta f_3$ are given as follows:

1. If $\mathbf{Z} = \text{prox}_{\eta f_1}[\mathbf{A}]$, $\mathbf{z}_{\cdot,t} = (\mathbf{E}_{K \times K} + \eta \mathbf{B}^\top \mathbf{B})^{-1} [\mathbf{a}_{\cdot,t} + \eta \mathbf{B}^\top \mathbf{H}^\top \mathbf{x}_{\cdot,t}]$, for $t = 1, \dots, T$.
2. If $\mathbf{Z} = \text{prox}_{\eta f_2}[\mathbf{A}]$, $\mathbf{z}_{k,\cdot} = \mathcal{F}^{-1} \left[\mathbf{c} \odot \mathcal{F}[\mathbf{a}_{k,\cdot}] \right]$ where \mathcal{F} and \mathcal{F}^{-1} denotes the Discrete Fourier Transform and Inverse Discrete Fourier Transform, respectively. “ \odot ” represents the elementwise product and $\mathbf{c} \in \mathbb{R}^T$ with $c_t = \frac{1}{1 + 4\lambda_3 \eta \left(1 - \cos \frac{(t-1)\pi}{T}\right)^2}$, $t = 1, \dots, T$.
3. If $\mathbf{Z} = \text{prox}_{\eta f_3}[\mathbf{A}]$, $Z_{k,t} = \max(A_{k,t} - \lambda_4 \eta, 0)$ for all $k = 1, \dots, K$ and $t = 1, \dots, T$.

Here $\mathbf{z}_{k,\cdot}$, $\mathbf{z}_{\cdot,t}$, and $Z_{k,t}$ denotes the row k , column t , and element (k, t) of matrix \mathbf{Z} . The notations $\mathbf{a}_{k,\cdot}$, $\mathbf{a}_{\cdot,t}$, and $A_{k,t}$ are similarly defined. $\mathbf{x}_{\cdot,t}$ represents the t -th column of \mathbf{X} .

4.3.2.3 Initialization

Problem (17) is not convex. Moreover, the problem of \mathbf{B} -update is not convex due to the constraint $\|\mathbf{b}_{k,\cdot}\|_F = 1$, and thus we can only obtain a local optimum. The \mathbf{Y} -update is a convex problem. In general, the MEIC algorithm converges to a local optimum, and therefore a good initialization of \mathbf{Y} is important.

In the MEIC algorithm, there are two intuitive considerations about the initial value \mathbf{Y}^0 . First, we hope that in the first step of \mathbf{B} -update, the columns of the solution \mathbf{B} are significantly different from each other, to capture the information regarding multiple events. Therefore, we want the collection of time points with large $Y_{t,k}^0$ to be different for different k 's. To achieve this, we assign only one event at each time t in the initial event sequences \mathbf{Y}^0 , so that for different k , the collections of time points with large $Y_{t,k}^0$ are disjoint. Furthermore, it is good to assign small consecutive time points in \mathbf{Y}^0 to the same event k , because the true values of $\mathbf{Y}_{\cdot,t}$ and $\mathbf{Y}_{\cdot,t'}$ are similar when time t and t' are close given the continuity of the events in the temporal domain. Second, an event may occur at any time. Therefore, the sequence corresponding to every event should cover the entire sequence because.

Based on the above considerations, we propose the following scheme of initializing \mathbf{Y}^0 . We generate \mathbf{Y}^0 randomly based on a Markovian chain with K states. Specifically, let \mathbf{y}^0 be a Markov chain on state $\{1, \dots, K\}$ with the following transition probability: $p_{k,k} = 1 - \alpha$, and $p_{k,k'} = \frac{\alpha}{K-1}$. Here, α is a tuning parameter that adjusts the frequency of jumps between states. After simulating \mathbf{y}^0 as a path of T time points, we set $Y_{k,t}^0 = 1$ if $y_t^0 = k$, and $Y_{k,t}^0 = 0$ otherwise. Three sample paths of \mathbf{y}^0 corresponding to three values of $\alpha = 1/5, \alpha = 1/10$, and $\alpha = 1/15$ are illustrated in Figure 16 with $K = 3$ and $T = 80$. In practice, α can be selected as the inverse of the expected length of the event periods.

Note that the initialization procedure of \mathbf{Y}^0 is random. It enables us to run Algorithm 7 multiple times with different realizations of \mathbf{Y}^0 , and select the local optimum with the minimal objective value.

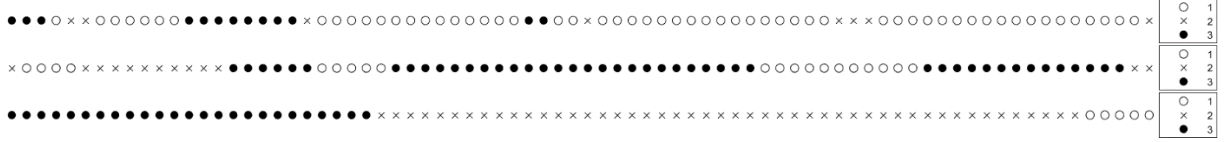


Figure 16 The sample paths of \mathbf{y}^0 in initialization, corresponding to $\alpha = 1/5, 1/10, \text{ and } 1/15$. The circles, crosses, and dots indicate three events.

4.4 Simulation Studies

In this section, we present the following simulation study to investigate the MEIC algorithm. As discussed in the literature review, there is no method that simultaneously characterizes the event signatures and identifies the event sequences based on historical data. Therefore, the simulation study proposed here only aims to evaluate the performance of our method and test if the MEIC algorithm is stable under multiple settings.

4.4.1 Simulation setup

We consider a simulation testbed of $I = 10$ signals, each with length $J_i = 128, i = 1, \dots, 10$. In general, we generate the data \mathbf{X} by simulating the coefficients of the event signatures \mathbf{B} and the event sequences \mathbf{Y} , and then calculating $\mathbf{X} = \mathbf{H}\mathbf{B}\mathbf{Y} + \mathbf{E}$, where $E_{ij} \sim N(0, 0.1^2)$. We select the Haar basis as the wavelet basis \mathbf{H} .

In practice, there is a wide variety of possible scenarios in terms of the number of events, the length of the historical data, as well as the periods in which events happen. In our simulation study, we thereby consider evaluating the algorithm under different scenarios by varying the lengths of the historical data T , the total number of events K , the lengths of time duration covered with at least one event, and the lengths of time duration with overlapping events. Specifically, we selected nine represented scenarios upon which

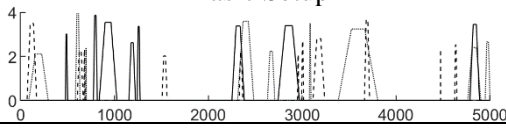
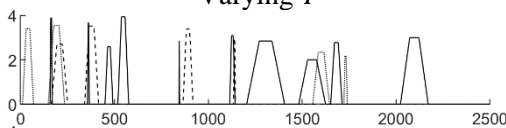
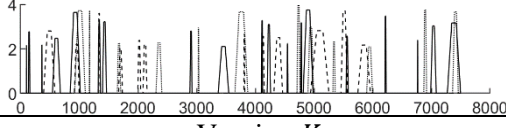
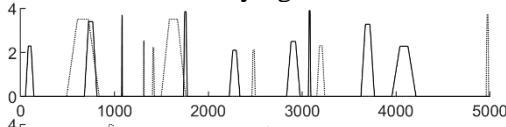
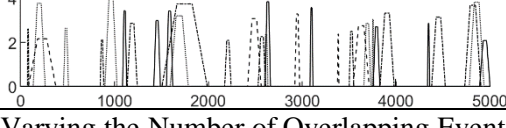
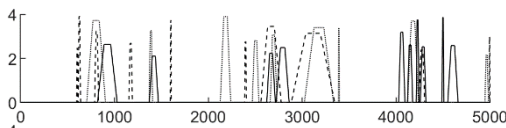
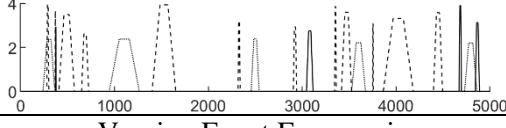
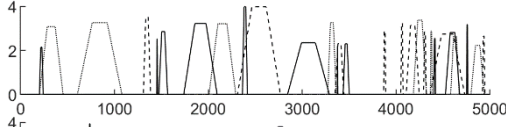
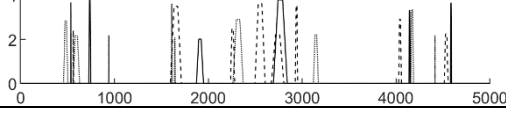
the MEIC algorithms are tested. Their number of events, length of the historical data, associated characteristics, and the event sequences \mathbf{Y} are shown in Table 7, where the letters H, M, and L represent High, Medium, and Low. In the subsection below, we detail the algorithm of generating \mathbf{Y} .

4.4.1.1 Generating the event sequences \mathbf{Y}

The matrices \mathbf{Y} corresponding to the nine setups are generated from the same general randomized procedure. Our strategy is to first use this procedure to generate $R = 1000$ realizations of \mathbf{Y} , and then we select typical realizations \mathbf{Y} for each of the nine setups. In what follows, we first describe the general randomized procedure and then demonstrate how the matrices \mathbf{Y} corresponding to the nine setups are selected.

The general randomized procedure of generating a matrix \mathbf{Y} is as follows. We independently generate the strength sequence $\mathbf{Y}_{k,\cdot}$ for every event $k = 1, \dots, K$. For each sequence $\mathbf{Y}_{k,\cdot}$, we generate it from an alternating renewal process [85], of which the occurrences of event k and the interims appear iteratively. Their lengths are exponentially distributed with $\text{Exp}(100^{-1})$ and $\text{Exp}(500^{-1})$ respectively. In each occurrence of an event, the event strength increases from 0 to a random level following $U(2,4)$ and then decreases gradually to 0. In each interim, $Y_{k,t} = 0$.

Table 7 Simulation setups

Index	Event Sequence	T	K	Overlap	Frequency
Basic Setup					
1		5000	3	M	M
Varying T					
2		2500	3	M	M
3		7500	3	M	M
Varying K					
4		5000	2	M	M
5		5000	4	M	M
Varying the Number of Overlapping Events					
6		5000	3	H	M
7		5000	3	L	M
Varying Event Frequencies					
8		5000	3	M	H
9		5000	3	M	L

For each randomly generated matrix of \mathbf{Y} , we define two indices. The *frequency* index r_f reflects the frequency of time points at which more than zero events occur, and the *overlap* index r_o reflects the proportion of time points with more than one overlapping

event to all time points with at least one event occurs. Here $\#\{A\}$ means the number of elements in set A .

$$r_f = \#\{t: \sum_{k=1}^K 1_{\{Y_{k,t}>0\}} > 0, t = 1, \dots, T\} / T$$

$$r_o = \#\{t: \sum_{k=1}^K 1_{\{Y_{k,t}>0\}} > 1, t = 1, \dots, T\} / \#\{t: \sum_{k=1}^K 1_{\{Y_{k,t}>0\}} > 0, t = 1, \dots, T\}$$

In Setups 1, 6, 7, 8, and 9, the numbers of events are all $K = 3$, and the length of the historical data are all $T = 5000$. We generate $R = 1000$ candidate matrices of \mathbf{Y} with the prescribed values of K and T through the procedure described above. Then, we draw the scatter plot of (r_f, r_o) for these replicates, as shown in Figure 17. We select five points from the scatter plot, corresponding to medium r_o and medium r_f , with high/low r_o and medium r_f , and with medium r_o and high/low r_f , as illustrated by the star points in Figure 17. These points correspond to the matrices \mathbf{Y} 's for Setups 1, 6, 7, 8 and 9.

In Setups 2, 3, 4, and 5, we obtain the \mathbf{Y} in a similar procedure. For each setup, we generate $R = 1000$ candidate matrices of \mathbf{Y} corresponding to the corresponding values of K and T . Among these candidate matrices, we pick the \mathbf{Y} whose r_f and r_o indices are close to their respective average among all 1000 replicates.

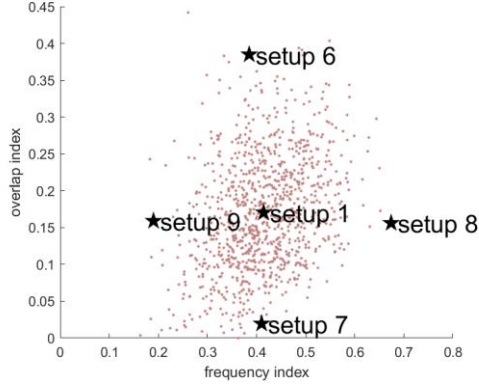


Figure 17 The scatter plot of r_f and r_o and selected event sequences \mathbf{Y} 's.

4.4.1.2 Generate the event signature coefficient \mathbf{B}

In the simulation platform, we assume each event affects three sensing signals. Therefore, there are three signal i 's such that $\mathbf{b}_{k,i} \neq \mathbf{0}$ for every $k = 1, \dots, K$. In the nine setups above, there are at most $K = 4$ events involved in each simulation setup. We let event $k = 1$ affect signal $i = 1, 2, 3$, event 2 affect signals 2, 4, 5, event 3 affect signals 3,6,7, and event 4 affect signals 5, 7, 8. For each event k and one affected signal i , we generate $\mathbf{b}_{k,i}$ by randomly choosing five non-zero elements and sample their values from $U(1,3)$. Finally, each vector $\mathbf{b}_{k,\cdot}$ is scaled to the unit length.

4.4.2 *Estimation results and evaluations*

For each simulation setup, we obtain the estimated event signature $\{\hat{\xi}_{k,i}: k = 1, \dots, K; i = 1, \dots, I\}$ and event sequences $\{\hat{y}_{k,t}: k = 1, \dots, K; t = 1, \dots, T\}$. In the estimation, we run the BCD algorithms based on ten initial \mathbf{Y}_0 's and the same tuning parameters $\lambda_1 = 7, \lambda_2 = 0.3, \lambda_3 = 10$ and $\lambda_4 = 0.3$. The BCD algorithm terminates when the objective value

decreases less than 0.01% in one iteration, and the threshold to primal and dual residual in D-step and Y-step are both set to be 10^{-4} .

We now investigate the convergence behavior of the basic setup. In Setup 1, Figure 18 (a) illustrates the decrease of objective values of the optimization problem after every **B**-update and **Y**-update in the BCD iterations. This figure shows that the objective value decreases monotonically, and the algorithm converges with few BCD iterations. Each line in Figure 18 (b) illustrates the change of primal and dual residual errors in the logarithm scale in each individual **B**-update and **Y**-update process. We can see that the ADMM consensus algorithm converges rapidly in each **B**-update and **Y**-update step. The convergence behavior of the Setup 2-9 is similar.

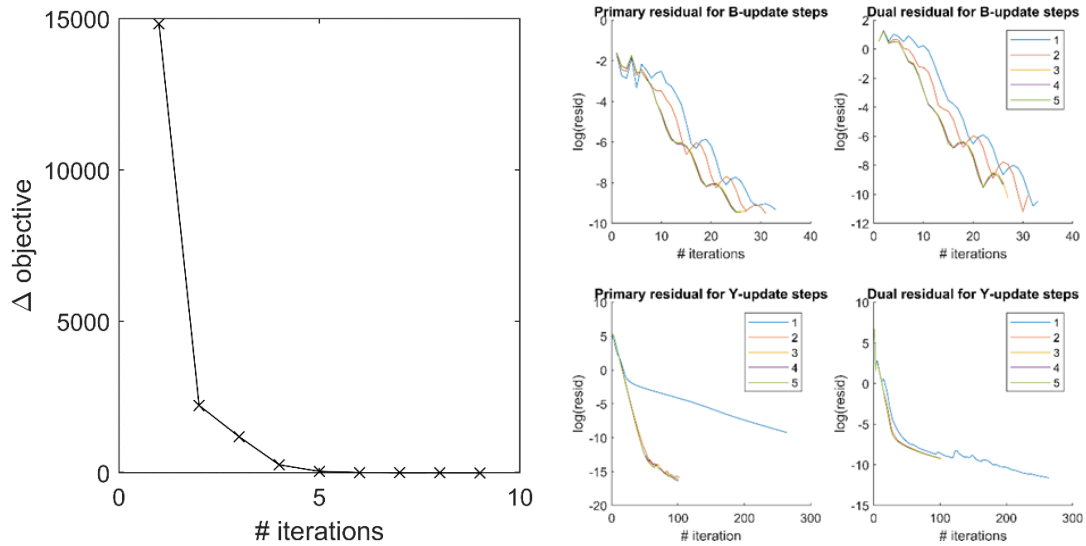


Figure 18 (a) The convergence of the BCD algorithm. (b) The convergence of each B-update and Y-update.

Identification of event sequences

We match the estimated event $k = 1, \dots, K$ with a true event k' by finding $\operatorname{argmin}_{k'} \left\{ \sum_{i=1}^I \|\mathbf{Y}_{k',i} - \widehat{\mathbf{Y}}_{k,i}\|_2^2 \right\}$. Figure 19 shows the estimation of

the event sequences for Setup 1. We can see that the sequences of the estimated event strengths are very similar to their true values, although the magnitudes are slightly lower due to the smoothing effect.

To give a numerical performance measure of the error identification, we find all time points that are identified as associated with event k , i.e. $\hat{E}_k = \{t: \hat{Y}_{k,t} > 0\}$. Then, we compare the set \hat{E}_k with E_k , the set of time points where event k indeed occurs, $E_k = \{t: Y_{k,t} > 0\}$. The time points misidentified as not event k is subject to the type I error, and the time point misclassified as event k is subject to the type II error. They can be represented as $E_k - \hat{E}_k$ and $\hat{E}_k - E_k$ respectively, and thus the type I and type II error rates are

$$V_{Y,1} = \sum_{k=1}^K \# \{E_k - \hat{E}_k\} / TK \text{ and } V_{Y,2} = \sum_{k=1}^K \# \{\hat{E}_k - E_k\} / TK.$$

We list both values and their sum in Table 8.

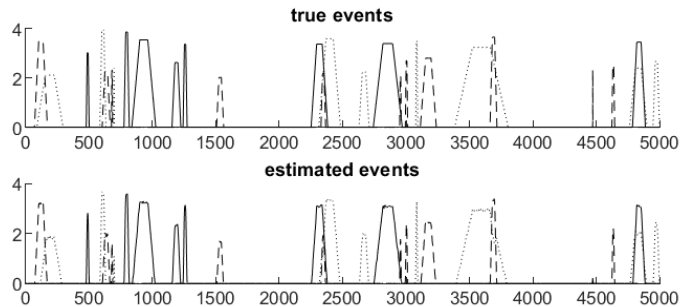


Figure 19 The true event sequence and the estimated event sequence according to Setup 1.

Table 8 The error rate of event sequence identification

Setup	1	2	3	4	5	6	7	8	9
Type I error	0.0076	0.0071	0.0076	0.0089	0.0069	0.0092	0.0062	0.0136	0.0033
Type II error	0.0023	0.0039	0.0023	0.0018	0.0011	0.0019	0.0024	0.0020	0.0040
Total Error	0.0099	0.0109	0.0098	0.0107	0.0080	0.0111	0.0086	0.0156	0.0073

Note that the tuning parameters determine the trade-off between type I and type II error rates. For the fairness of comparison, we select the same parameter λ_4 in all setups, and thus we mainly compare the total error. We can see that the error rate does not exceed 2% for all setups from the result. Setup 8, corresponding to the highest frequency of occurrence, has the largest error rate of 1.56%. In all setups, we found that misidentification occurs when an event appears or disappears. Therefore, a possible reason for the large misidentification rate of Setup 8 is that high frequency of event periods associates with multiple events occurrence and disappearance.

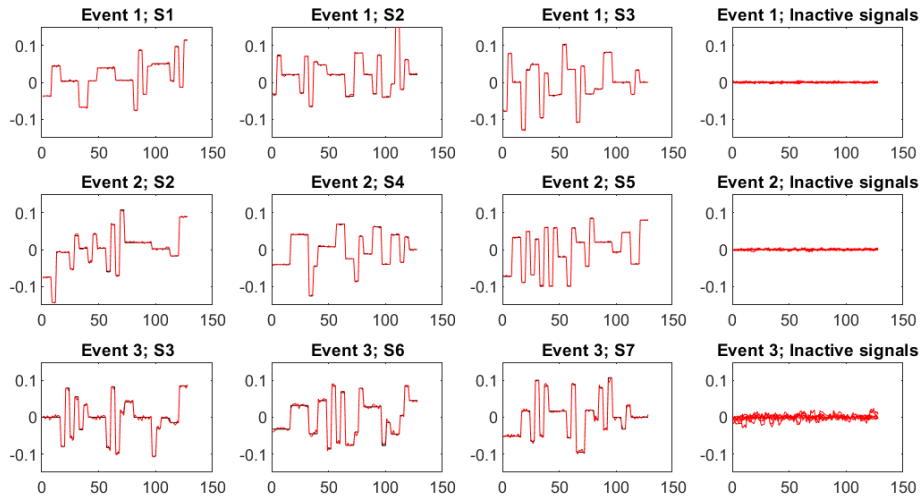


Figure 20 The estimated event signatures on ten sensors of the three events, according to Setup 1. The horizontal axis in each figure represents the measurement points in each signal.

Table 9 The values of V_ξ for nine setups

Setup	1	2	3	4	5	6	7	8	9
V_ξ	0.0010	0.0036	0.0040	0.0010	0.0089	0.0013	0.0012	0.0030	0.0010

Characterization of event signatures

Figure 20 shows the reconstructed event signatures corresponding to events $k = 1, 2, 3$ on curve $i = 1, \dots, 10$ for Setup 1. Plots in each row correspond to an event. The first three subfigures in each row illustrate the estimation of the event signature on three signals that are affected by this event. These signals are marked by S1-S7 in the respective plot titles. We can see that the red lines (representing the estimated event signature on these signals) are very close to the black lines (representing the true event signature on these signals). The fourth subfigure in each row illustrates the estimation of the event signature on signals not affected by an event. The red lines, representing the estimated event signature on irrelevant signals, are all close to zero. This result shows that the event signatures are estimated accurately and that signals that are truly associated with an event can be identified correctly.

After estimating the event signatures, we evaluate the estimation accuracy of those event signatures by the mean squared error $V_\xi = \frac{1}{KI} \sum_{k,i} \|\xi_{k,i} - \hat{\xi}_{k,i}\|_2^2$, as shown in Table 9. For Setup 1, the value of V_ξ is 0.0010. We calculated the value V_ξ for all nine setups, and find that Setup 5 (the case corresponding to $K = 4$) has the largest $V_\xi = 0.0089$, whereas in all other cases, the values of V_ξ do not exceed 0.0040. It shows that the MEIC algorithm characterizes the event signatures accurately.

Importance of multiple starting points In our simulation study, we discover that the result of the algorithm is sensitive to the initial \mathbf{Y}^0 . Occasionally, the algorithm may converge to a local minimum, in which two identified events correspond to one real event and have similar event signatures. Similar behavior also occurs to prototype methods like Gaussian Mixture Model and k -means clustering. Therefore, we perform the optimization algorithm starting from multiple initial values of \mathbf{Y}^0 and pick the solution with minimal objective values. In this way, we find that the MEIC algorithm is able to get excellent identification results in every experiment we performed.

4.5 Case Study

In steel rolling processes, the shape uniformity of the rolling bars is an important quality characteristic [86]. In a rolling production line, a laser gauge is installed for in-situ measurement of the cross-sectional shapes of a rolling bar, which generates six profiles $x_{t,1}(s), \dots, x_{t,6}(s)$ to represent the diameter measurements along six axes of every rolling bar t . Each $x_{t,i}(s)$ ($i = 1, \dots, 6$) is a functional curve that reflects the dimension from the beginning to the end of each rolling bar. The rolling process and the laser gauge measurements are illustrated in Figure 21.

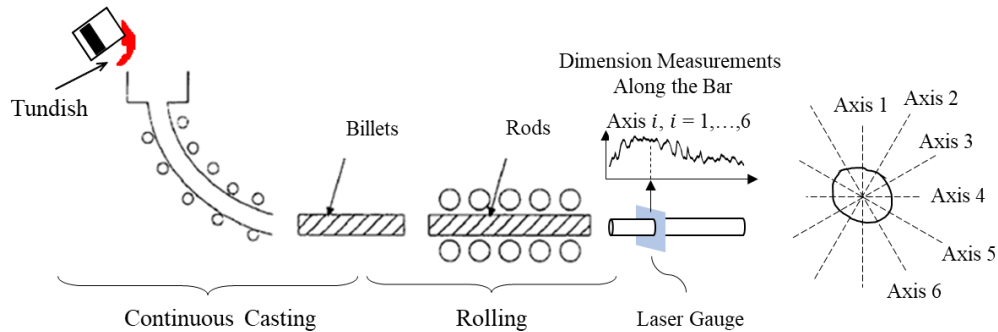


Figure 21 The illustration of a rolling process, where the blue square represents the measurement plane of the laser gauge. The shape at the right side illustrates the cross-sectional shape of a rolling bar and its diameter measurements along six axes.

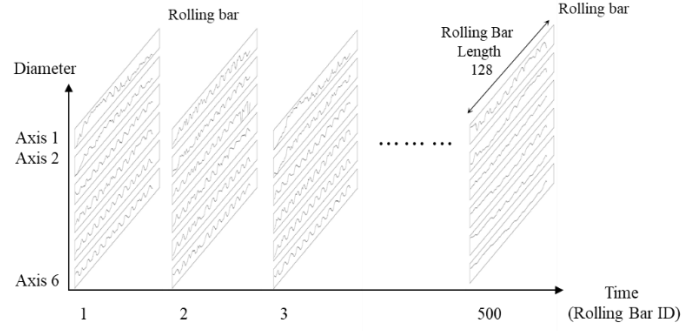


Figure 22 The illustration of the raw data for the case study.

In this case study, the dimensional profile measurements are obtained from a continuous production of $n = 500$ rolling bars. By analyzing those data, we like to find out if there are any special events that occurred during the production and further identify the specific events and their signatures. To serve this purpose, we apply the MEIC algorithm to this dataset.

The data sequence corresponding to each rolling bar has the length $d = 128$. After certain preprocessing steps, we obtain the tensor data $\mathcal{X} \in \mathbb{R}^{128 \times 6 \times 500}$, illustrated in Figure 22, where at each time $t = 1, \dots, 500$ we obtain six curves of length 128, denoting the diameter measurements of the rolling bar along Axes 1-6. The data is then reshaped into the matrix $\mathbf{X} \in \mathbb{R}^{(128 \times 6) \times 500}$.

We set $K = 2$, apply the Haar wavelet as the basis \mathbf{H} , and specify the tuning parameters as $\lambda_1 = 0, \lambda_2 = 0.04, \lambda_3 = 500$, and $\lambda_4 = 0.15$. Here λ_1 is set to 0 because there is no need to specify that an event affects only a subset of sensing signals: the abnormal condition in the rolling process will affect the entire cross-section of a rolling bar, and thus the diameter measurements of all Axes 1-6 will be impacted simultaneously. After using the MEIC algorithm to analyze the data set, we obtain the estimated matrix \mathbf{B}

and \mathbf{Y} . From the matrix \mathbf{B} , we recover the event signatures on six signals, as illustrated in Figure 23, and identify the event sequences as in Figure 24. From Figure 23, we can conclude that two abnormal events have occurred in the rolling process. The first event associates with an increased diameter along Axis 1 near the first quarter of the billet and an increasing diameter along Axis 4. The second event relates to a sharp drop in the diameter value along Axes 1, 3, 4, and 5. In Figure 24, the solid line represents the first event, and the dashed line represents the second event. It tells us that event 1 occurs during the fabrication of the first 150 billets, and event 2 occurs between the 240th to 290th rolling bars.

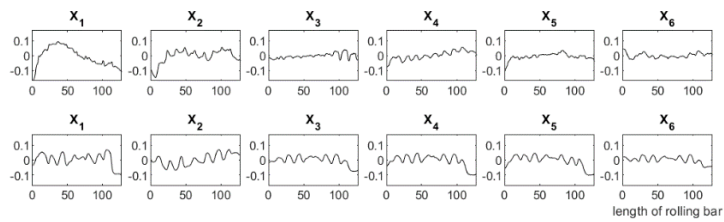


Figure 23 The event signatures on six signals for event 1 (first row) and event 2 (second row).

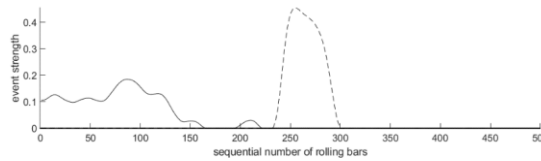


Figure 24 The sequence of event 1 (solid line) and event 2 (dashed line).

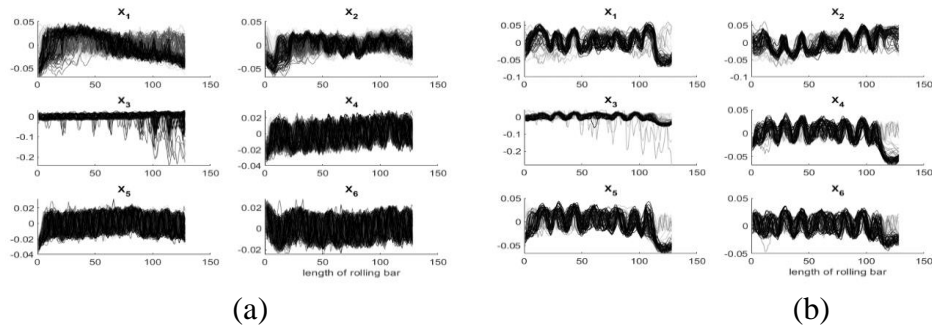


Figure 25 The sample signals that are associated with (a) event 1 and (b) event 2.

To validate the existence of those two events and their characteristics, we draw the preprocessed signals obtained when each event occurs in Figure 25 (a) and (b). Those two figures correspond to event 1 and event 2, respectively. In each plot, a curve represents the measurements along one sample rolling bar. The gray level indicates the estimated strength of events 1 or 2 for this bar. Comparing Figure 25 with Figure 23, we can observe that those two events are indeed associated with the event signatures we estimated because the patterns of the dark curves are very similar to the event signatures.

4.6 Summary

It is common to have multiple sensors installed in a system to faithfully record its operation status and generate structured data streaming. Retrospective analysis of those sensing data enables a better understanding of the system status during its operation and gain insights on new events that affect the system performance. A common question in this retrospective data analysis is to ask if there are some special events (e.g., machine failure, material change, process perturbation, etc.) that occurred during the production time period. If so, what is the event that has occurred? When it occurred? What is the duration of the event? Which sensing signals the event affects? And what are the event signatures shown on the related sensing signals? Answering those questions will enable the development of more effective monitoring and diagnosis tools for process control and quality improvements. These questions motivate us to define *events* and associated *event signatures* in a system and to develop an automatic tool for identifying and characterizing those events from the system operational data.

The multiple event identification and characterization (MEIC) algorithm simultaneously identifies the occurrence of each event during the system operation and characterizes how each event impacts sensing signals by estimating the event signatures. Our approach has two major advantages: First, it does not require labeled observations corresponding to every single event, which typically involves practitioners going through the data stream and label the abnormal segment in sensing data. Second, it allows the anomaly identification and characterization in a single step to streamline the analysis procedure and focus on vital quality issues. Therefore, this approach leads to a deep understanding of the system and its operations based on the sensing signals.

In our simulation study, we verify the validity of the MEIC method in identifying event sequences and characterizing event signatures. The algorithm effectively avoids suboptimal points with our scheme of generating random initial values from the computational aspect. Both the BCD algorithm and the inner loops of the ADMM algorithms converge rapidly. The case study successfully identified two events from a set of dimensional sensing data of rolling bars and simultaneously characterized how each event affects the dimension of the rolling bar.

CHAPTER 5. CONCLUSION

This thesis investigates three problems in the data analytics of a multistage manufacturing system, motivated by the characteristics of the systems and the data characteristics. This chapter first summarizes the unique contributions of the thesis. Then, several future research areas are listed.

5.1 Summary of the contributions

Chapter 2 in this thesis investigates the feature ranking problem for data-driven system diagnostics. One original contribution is that *two requirements for a feature ranking procedure are summarized* from the semiconductor manufacturing application: (1) the ranking should be based on *general* dependency, and (2) a process feature shall be prioritized if it is not dependent with other features strongly related to the quality variable. Furthermore, *a new feature ranking approach is proposed based on distance correlation that satisfies these requirements*. The theoretical study and intuitive illustrations are given to show how this method works, and the effectiveness of the method has been demonstrated in simulation studies and a real case study.

Chapter 3 proposes an analytical framework for multistage manufacturing processes. It is *the first analytical framework for multistage manufacturing processes that enables parallel computation*, and it *simultaneously identifies the actual root causes on every stage of the system, the variation patterns of the outputs in every stage, and the relationship between them*. This framework is also highly customizable for a wide variety of multistage processes generating multiple kinds of data.

The multiple event identification and characterization (MEIC) method in Chapter 4 has the original feature of *performing identification and root cause diagnostics simultaneously. It is enabled by using the dictionary learning method, which has not been introduced to the process data analytics for modeling and diagnosis in MMPs.* The MEIC has been proved as a useful automatic tool in retrospective analysis of sensing signals that gives practitioners hints to discovering new root causes of the processes and leads deep understanding of the system and its operations.

5.2 Future Works

The multistage process can be regarded as a particular type of interconnected system, which includes, but beyond, manufacturing systems. The introduction of this thesis lists four major characteristics of the data generated from multistage processes: heterogeneous data types, multiple root causes of variations, error propagation between stages, and confounding relationship between multiple data sources. Typically, these are also the characteristics of the data generated from general interconnected systems such as the internet-of-things, and therefore the problem formulation and modeling approach used in this thesis can be extended to these systems.

In Section 3.3.1, it has been mentioned that the structure of the data generated from each sample of a multistage manufacturing process is in the form of a C struct or MATLAB[®] cell, which contains multiple components representing the data generated from sensors. It is also the data type generated from the general interconnected Internet-of-Things (IoT), and special statistical modeling and analysis tools are desired for this

structured data type. Two potential research directions on the analysis of this kind of data type for systems improvements are listed as follows.

The first potential research direction is to model the multimodal data generated from interconnected systems. On the one hand, the data from each sensor should be described based on its own characteristics, such as signals, point clouds, multiple categorical data, etc. On the other hand, the interconnections between data from different sensors need to be characterized using graphical models and transfer learning methods. The estimation of the parameters in the model can be achieved by a federated optimization approach. This research will enable and streamline the decision-making process within the engineering processes.

The second potential research direction is the modularized deep learning architectures for the data generated by interconnected systems. Specifically, the system layout of the interconnected system can be used for specifying the deep learning architectures of the heterogeneous data sources in modeling the relationship between the process data from multiple system components and the quality variable. Furthermore, the deep learning modules for similar system components can be applied with the same deep learning architecture. These measures reduce the number of parameters and increase the model's interpretability to engineers.

APPENDIX A.

SUPPLEMENTARY MATERIALS FOR CHAPTER 2

A.1 The derivation of $V_{n,\beta}(\mathbf{X}, \mathbf{Y})$ and $R_n^2(\mathbf{X}, \mathbf{Y})$ in Section 2.3

Based on the notations from Section 2.3,

$$\begin{aligned} V_{n,\beta}(\mathbf{X}, \mathbf{Y}) &= \frac{1}{n^2} \sum_{k=1}^n \sum_{l=1}^n A_{kl} B_{kl} \\ &= \frac{1}{n^2} \sum_{k=1}^n \sum_{l=1}^n [a_{kl} - \bar{a}_{k\cdot} - \bar{a}_{\cdot l} + \bar{\bar{a}}_{\cdot\cdot}] [b_{kl} - \bar{b}_{k\cdot} - \bar{b}_{\cdot l} + \bar{\bar{b}}_{\cdot\cdot}]. \end{aligned}$$

Here $a_{kl} = d_{\beta}(\mathbf{X}_k, \mathbf{X}_l) = \sum_{i=1}^p \beta_i |X_k^{(i)} - X_l^{(i)}|$; $b_{kl} = |Y_k - Y_l|$, and

$$\bar{a}_{k\cdot} = \frac{1}{n} \sum_{l=1}^n a_{kl}, \bar{a}_{\cdot l} = \frac{1}{n} \sum_{k=1}^n a_{kl}, \bar{\bar{a}}_{\cdot\cdot} = \frac{1}{n^2} \sum_{k,l=1}^n a_{kl};$$

$$\bar{b}_{k\cdot} = \frac{1}{n} \sum_{l=1}^n b_{kl}, \bar{b}_{\cdot l} = \frac{1}{n} \sum_{k=1}^n b_{kl}, \bar{\bar{b}}_{\cdot\cdot} = \frac{1}{n^2} \sum_{k,l=1}^n b_{kl}.$$

Let $a_{kl}^{(i)} = |X_k^{(i)} - X_l^{(i)}|$, we have

$$\begin{aligned} V_{n,\beta}(\mathbf{X}, \mathbf{Y}) &= \frac{1}{n^2} \sum_{i=1}^d \sum_{k=1}^n \sum_{l=1}^n \beta_i [a_{kl}^{(i)} - \bar{a}_{k\cdot}^{(i)} - \bar{a}_{\cdot l}^{(i)} + \bar{\bar{a}}_{\cdot\cdot}^{(i)}] [b_{kl} - \bar{b}_{k\cdot} - \bar{b}_{\cdot l} + \bar{\bar{b}}_{\cdot\cdot}] \\ &= \sum_{i=1}^d \beta_i d_{n,i}, \end{aligned}$$

where $d_{n,i} = \frac{1}{n^2} \sum_{k=1}^n \sum_{l=1}^n [a_{kl}^{(i)} - \bar{a}_{k\cdot}^{(i)} - \bar{a}_{\cdot l}^{(i)} + \bar{\bar{a}}_{\cdot\cdot}^{(i)}] [b_{kl} - \bar{b}_{k\cdot} - \bar{b}_{\cdot l} + \bar{\bar{b}}_{\cdot\cdot}] = V_n(\mathbf{X}^{(i)}, \mathbf{Y})$,

the sample distance covariance between two univariate random variables X_i and Y based on the Euclidian distance. Therefore, $V_{n,\beta}(\mathbf{X}, \mathbf{Y})$ can be represented as $\mathbf{d}_n^\top \boldsymbol{\beta}$.

Similarly,

$$\begin{aligned} V_{n,\beta}(\mathbf{X}, \mathbf{X}) &= \frac{1}{n^2} \sum_{k=1}^n \sum_{l=1}^n [a_{kl} - \bar{a}_{k\cdot} - \bar{a}_{\cdot l} + \bar{\bar{a}}_{\cdot\cdot}] [a_{kl} - \bar{a}_{k\cdot} - \bar{a}_{\cdot l} + \bar{\bar{a}}_{\cdot\cdot}] \\ &= \frac{1}{n^2} \sum_{k=1}^n \sum_{l=1}^n \sum_{i=1}^d \sum_{j=1}^d \beta_i \beta_j [a_{kl}^{(i)} - \bar{a}_{k\cdot}^{(i)} - \bar{a}_{\cdot l}^{(i)} + \bar{\bar{a}}_{\cdot\cdot}^{(i)}] [a_{kl}^{(j)} - \bar{a}_{k\cdot}^{(j)} - \bar{a}_{\cdot l}^{(j)} \\ &\quad + \bar{\bar{a}}_{\cdot\cdot}^{(j)}] = \sum_{i=1}^d \sum_{j=1}^d \beta_i \beta_j [\mathbf{F}_n]_{ij} = \boldsymbol{\beta}^\top \mathbf{F}_n \boldsymbol{\beta}, \end{aligned}$$

and thus we can derive

$$R_{n,\beta}^2(\mathbf{X}, \mathbf{Y}) = \frac{V_{n,\beta}(\mathbf{X}, \mathbf{Y})}{\sqrt{V_{n,\beta}(\mathbf{X}, \mathbf{X})} \sqrt{V_n(\mathbf{Y}, \mathbf{Y})}} = \frac{\mathbf{d}_n^\top \boldsymbol{\beta}}{\sqrt{V_n(\mathbf{Y}, \mathbf{Y})} \sqrt{\boldsymbol{\beta}^\top \mathbf{F}_n \boldsymbol{\beta}}}.$$

Therefore, $R_{n,\beta}^2(\mathbf{X}, \mathbf{Y})$ can be represented as $K \cdot \frac{\mathbf{d}_n^\top \boldsymbol{\beta}}{\sqrt{\boldsymbol{\beta}^\top \mathbf{F}_n \boldsymbol{\beta}}}$.

A.2 Proof of the claim on zero weighed population distance correlation

We show that $V_\beta(\mathbf{X}, \mathbf{Y}) = 0$ if and only if $X^{(i)}$ is independent of Y for every i corresponding to $\beta_i > 0$.

Proof: Similar to the derivation in Appendix A, we can show that

$$V_{\boldsymbol{\beta}}(\mathbf{X}, Y) = \sum_{i=1}^n \beta_i V(X^{(i)}, Y),$$

using the expression of *population distance covariance* in Section 2.3.1. We can see that $V_{\boldsymbol{\beta}}(\mathbf{X}, Y) = 0$ if and only if $V(X^{(i)}, Y) = 0$ for all $i \in \{i | \beta_i > 0\}$. By Theorem 3.11 in Lyons [15], $V(X^{(i)}, Y) = 0$ if and only if X_i and Y is independent. Therefore, the conclusion in the property holds.

A.3 Proof of Proposition 1

(1) The “if” part is straightforward. We only prove the “only if” part. Suppose that $\boldsymbol{\beta}$, one of the optimal solution of problem (4), satisfies $\boldsymbol{\beta}^\top \mathbf{F}_n \boldsymbol{\beta} < 1$. We show that problem (3) is infeasible. The Lagrange function of the optimization problem (4) is

$$L(\boldsymbol{\beta}; \xi, \mu, \boldsymbol{\lambda}) = -\boldsymbol{\beta}^\top \mathbf{d}_n + \xi(\boldsymbol{\beta}^\top \mathbf{F}_n \boldsymbol{\beta} - 1) + \mu(\mathbf{1}^\top \boldsymbol{\beta} - c) - \boldsymbol{\lambda}^\top \boldsymbol{\beta}.$$

Therefore, the KKT condition is $\boldsymbol{\beta} \geq 0; \xi \geq 0; \boldsymbol{\lambda} \geq 0; \mu \geq 0$, and

$$2\xi \mathbf{F}_n \boldsymbol{\beta} = \boldsymbol{\lambda} + \mathbf{d}_n - \mu \mathbf{1},$$

$$\mu(\boldsymbol{\beta}^\top \mathbf{1} - c) = 0;$$

$$\boldsymbol{\lambda}^\top \boldsymbol{\beta} = 0;$$

$$\xi(\boldsymbol{\beta}^\top \mathbf{F}_n \boldsymbol{\beta} - 1) = 0.$$

If $\boldsymbol{\beta}^\top \mathbf{F}_n \boldsymbol{\beta} < 1$, then $\xi = 0$, and we have $\boldsymbol{\lambda} = \mu \mathbf{1} - \mathbf{d}_n$. To make $\boldsymbol{\lambda} \geq 0$, we need $\mu \geq$

$\max_{1 \leq i \leq p} \{d_n^{(i)}\} > 0$. With probability 1, $d_n^{(1)}, \dots, d_n^{(p)}$ are p different values, $\boldsymbol{\lambda}$ has one zero

entry if we take $\mu = \max_{1 \leq i \leq p} \{d_n^{(i)}\}$ and has no zero entry if $\mu > \max_{1 \leq i \leq p} \{d_n^{(i)}\}$. Due to complementary slackness, $\boldsymbol{\beta}$ has at most one *non-zero* entry. Without loss of generality, we assume that $\boldsymbol{\beta} = (\beta_1, \mathbf{0}_{p-1})^\top$. In such case, $\beta_1^2 = \beta_1^2 V_n(\mathbf{X}^{(1)}, \mathbf{X}^{(1)}) = \boldsymbol{\beta}^\top \mathbf{F}_n \boldsymbol{\beta} < 1$ so $\beta_1 < 1$. Because $\mu > 0$, we have $c = \boldsymbol{\beta}^\top \mathbf{1} = \beta_1 < 1$ due to complementary slackness. For every $\tilde{\boldsymbol{\beta}} \geq 0$, if $\tilde{\boldsymbol{\beta}}^\top \mathbf{1} \leq c$, from the definition of distance correlation we have $V_n(\mathbf{X}^{(i)}, \mathbf{X}^{(j)}) \leq \sqrt{V_n(\mathbf{X}^{(i)}, \mathbf{X}^{(i)}) V_n(\mathbf{X}^{(j)}, \mathbf{X}^{(j)})} = 1$, so $\tilde{\boldsymbol{\beta}}^\top \mathbf{F}_n \tilde{\boldsymbol{\beta}} \leq \tilde{\boldsymbol{\beta}}^\top [\mathbf{1}\mathbf{1}^\top] \tilde{\boldsymbol{\beta}} < c^2 < 1$. Therefore, problem (3) is not feasible.

(2) Suppose that $\boldsymbol{\beta}'$ and $\boldsymbol{\beta}''$ both achieve the optimal value of problem (4). Because $\mathbf{d}_n^\top \boldsymbol{\beta}' = \mathbf{d}_n^\top \boldsymbol{\beta}''$, and the feasible region of problem (4) is convex, we can check that $\tilde{\boldsymbol{\beta}} = \frac{1}{2}(\boldsymbol{\beta}' + \boldsymbol{\beta}'')$ also achieves the optimal value of problem (4). By Proposition 1 we have $\boldsymbol{\beta}'^\top \mathbf{F}_n \boldsymbol{\beta}' = \boldsymbol{\beta}''^\top \mathbf{F}_n \boldsymbol{\beta}'' = \tilde{\boldsymbol{\beta}}^\top \mathbf{F}_n \tilde{\boldsymbol{\beta}} = 1$, which derives $(\boldsymbol{\beta}' - \boldsymbol{\beta}'')^\top \mathbf{F}_n (\boldsymbol{\beta}' - \boldsymbol{\beta}'') = 0$. We conclude $\boldsymbol{\beta}' = \boldsymbol{\beta}''$ from the fact that \mathbf{F}_n is positive definite with probability 1.

Without loss of generality, we assume that $d_n^{(1)}$ is the maximal number among $d_n^{(1)}, d_n^{(2)}, \dots, d_n^{(p)}$. From the proof of (1) above, we see that the optimal solution to formulation (4) is in the form of $\boldsymbol{\beta} = (\beta_1, \mathbf{0}_{p-1})^\top$. Because $\boldsymbol{\beta}^\top \mathbf{1} = c$, $\boldsymbol{\beta} = (c, \mathbf{0}_{p-1})^\top$ is the only optimal solution of formulation (4).

A.4 Problem (4) is conic quadratic programming problem

Proof: We can directly check that problem (4) can be rearranged in the standard form of conic quadratic form in Lobo, et al. [87], equation (1). In fact, formulation (4) is transformed to the following standard form:

$$\min_{\boldsymbol{\beta}} \mathbf{k}^\top \boldsymbol{\beta}$$

$$\text{subject to } \|\mathbf{A}_i \boldsymbol{\beta} + \mathbf{b}_i\| \leq \mathbf{c}_i^\top \boldsymbol{\beta} + a_i, i = 1, \dots, p + 2,$$

where

- $\mathbf{k} = -\mathbf{d}_n$;
- $\mathbf{A}_i = \mathbf{0}_{p \times p}$ for $i = 1, \dots, p + 1$ and $\mathbf{A}_{p+2} = \mathbf{F}^{\frac{1}{2}}$;
- $\mathbf{b}_i = \mathbf{0}$ for $i = 1, \dots, p + 2$;
- $\mathbf{c}_i = \mathbf{e}_i$ for $i = 1, \dots, p$, where \mathbf{e}_i denotes the i^{th} unit vector; $\mathbf{c}_{p+1} = \mathbf{1}$ and $\mathbf{c}_{p+2} = \mathbf{0}$;
- $a_i = 0$ for $i = 1, \dots, p$, $a_{p+1} = -c$ and $a_{p+2} = 1$.

A.5 Proof of Proposition 2

Proof: (1) The feasible region of formulation (3) is

$$S_c = \{\boldsymbol{\beta}: \boldsymbol{\beta}^\top \mathbf{F}_n \boldsymbol{\beta} = 1; \beta_i \geq 0 \text{ for all } i = 1, \dots, p; \boldsymbol{\beta}^\top \mathbf{1} \leq c\}.$$

Let $A_c = \{\boldsymbol{\beta}: \boldsymbol{\beta}^\top \mathbf{1} \leq c; \beta_i \geq 0\} = \text{conv}\{\mathbf{0}, c\mathbf{e}_1, \dots, c\mathbf{e}_p\}$, and let $B = \{\boldsymbol{\beta}: \boldsymbol{\beta}^\top \mathbf{F}_n \boldsymbol{\beta} < 1\}$. Here $\mathbf{e}_1, \dots, \mathbf{e}_p$ are p standard unit vectors.

a. If $c < 1$, we have $\mathbf{0} \in B$ and $c\mathbf{e}_i \in B$ for $i = 1, \dots, p$. Because B is convex,

$$A_c \subset B. \text{ So } S_c \subset A_c - B = \emptyset.$$

b. If $c \geq 1$, $\{\mathbf{e}_1, \dots, \mathbf{e}_p\} \subset S_c \neq \emptyset$.

(2) Because all diagonal elements of \mathbf{F}_n are 1 and all off-diagonal elements of \mathbf{F}_n are non-negative, $\|\boldsymbol{\beta}\|_2^2 \leq \boldsymbol{\beta}^\top \mathbf{F}_n \boldsymbol{\beta} \leq 1$. Then $\|\boldsymbol{\beta}\|_1 \leq \sqrt{p} \|\boldsymbol{\beta}\|_2 \leq \sqrt{p}$, which means that the constraint $\boldsymbol{\beta}^\top \mathbf{1} \leq c$ is inactive as $c > \sqrt{p}$. Therefore, the solution of formulation (4) with $c > \sqrt{p}$ is the same with that of $c = \sqrt{p}$.

(3) Let $I = \{c \geq 1: \boldsymbol{\beta}(c)^\top \mathbf{1} = c\}$ be the set of c that makes $\boldsymbol{\beta}^\top \mathbf{1} \leq c$ an active constraint in formulation (4).

Lemma: With probability 1, there is some $c_0 \leq \sqrt{p}$ such that $I = [1, c_0]$.

Proof of the Lemma: The maximum value of I is bounded by $\max\{\sum_{i=1}^n \beta_i : \boldsymbol{\beta}^\top \mathbf{F}_n \boldsymbol{\beta} \leq 1; \beta_i \geq 0, i = 1, \dots, n\}$. To show that $I = [1, c_0]$ for some $c_0 \geq 0$, we only need that $c' \in I$ indicates $c'' \in I$ for all $c'' < c'$. Suppose that this statement is not true. That is, $\boldsymbol{\beta}(c')^\top \mathbf{1} = c'$ and $\boldsymbol{\beta}(c'')^\top \mathbf{1} < c''$. Then we can select $\eta \in [0, 1]$ such that $\tilde{\boldsymbol{\beta}} = \eta \boldsymbol{\beta}(c') + (1 - \eta) \boldsymbol{\beta}(c'')$ satisfies $\tilde{\boldsymbol{\beta}}^\top \mathbf{1} = c''$. $\tilde{\boldsymbol{\beta}}$ is a feasible solution for problem (3) with parameter c'' . Also, the feasible region for the problem (4) with parameter c' contains the feasible region for the problem (4) with c'' , so the objective function $-\boldsymbol{\beta}(c')^\top \mathbf{d}_n \leq -\boldsymbol{\beta}(c'')^\top \mathbf{d}_n$, and thus we have

$$-\boldsymbol{\beta}(c')^\top \mathbf{d}_n \leq -\tilde{\boldsymbol{\beta}}^\top \mathbf{d}_n \leq -\boldsymbol{\beta}(c'')^\top \mathbf{d}_n.$$

The second inequality above indicates that $\tilde{\boldsymbol{\beta}}$ also achieves the optimal value of the problem (4) with parameter c'' , which contradicts with the uniqueness of $\boldsymbol{\beta}(c'')$ indicated by the corollary of Proposition 1.

Proof of (3) in Proposition 2:

Case 1: $c_1, c_2 \in I$.

Let $\boldsymbol{\beta}(c_1) = (\boldsymbol{\beta}_1, \mathbf{0})^\top$, and $\boldsymbol{\beta}(c_2) = (\boldsymbol{\beta}_2, \mathbf{0})^\top$, where $\boldsymbol{\beta}_1 > \mathbf{0}$, $\boldsymbol{\beta}_2 > \mathbf{0}$ are of the same dimension. From the KKT condition, there exists $\xi_1 > 0, \boldsymbol{\lambda}_1 \geq \mathbf{0}, \mu_1 \geq 0$ and $\xi_2 > 0, \boldsymbol{\lambda}_2 \geq \mathbf{0}, \mu_2 > 0$ such that

$$2\xi_1 \mathbf{F}_n \begin{bmatrix} \boldsymbol{\beta}_1 \\ \mathbf{0} \end{bmatrix} + \mu_1 \mathbf{1} = \mathbf{d}_n + \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\lambda}_1 \end{bmatrix},$$

$$2\xi_2 \mathbf{F}_n \begin{bmatrix} \boldsymbol{\beta}_2 \\ \mathbf{0} \end{bmatrix} + \mu_2 \mathbf{1} = \mathbf{d}_n + \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\lambda}_2 \end{bmatrix}.$$

(Note that from (1), formulation (3) is feasible when $c \geq 1$. From proposition 1, formulation (4) satisfies $\boldsymbol{\beta}^\top \mathbf{F}_n \boldsymbol{\beta} = 1$. So $\xi_1, \xi_2 \neq 0$.)

We can see that for any $\tilde{\boldsymbol{\beta}}(t) = t\boldsymbol{\beta}_1 + (1-t)\boldsymbol{\beta}_2, t \in [0,1], \tilde{\boldsymbol{\beta}}(t) > \mathbf{0}$ and

$$\mathbf{F}_n \begin{bmatrix} \tilde{\boldsymbol{\beta}}(t) \\ \mathbf{0} \end{bmatrix} + \left(\frac{\mu_1}{2\xi_1} t + \frac{\mu_2}{2\xi_2} (1-t) \right) \mathbf{1} = \left(\frac{1}{2\xi_1} t + \frac{1}{2\xi_2} (1-t) \right) \mathbf{d}_n + \begin{bmatrix} 0 \\ \frac{1}{2\xi_1} \boldsymbol{\lambda}_1 + \frac{1}{2\xi_2} \boldsymbol{\lambda}_2 \end{bmatrix}.$$

By taking

$$\tilde{\boldsymbol{\beta}}(t) = \frac{\tilde{\boldsymbol{\beta}}(t)}{\sqrt{\tilde{\boldsymbol{\beta}}(t)^\top \mathbf{F}_{11,n} \tilde{\boldsymbol{\beta}}(t)}}; \tilde{\xi}(t) = \sqrt{\tilde{\boldsymbol{\beta}}(t)^\top \mathbf{F}_{11,n} \tilde{\boldsymbol{\beta}}(t)} / \left(\frac{1}{2\xi_1} t + \frac{1}{2\xi_2} (1-t) \right);$$

$$\tilde{\mu}(t) = \left(\frac{\mu_1}{2\xi_1} t + \frac{\mu_2}{2\xi_2} (1-t) \right) / \left(\frac{1}{2\xi_1} t + \frac{1}{2\xi_2} (1-t) \right) > 0 \text{ and } \tilde{\boldsymbol{\lambda}}(t) = \begin{bmatrix} 0 \\ \frac{\frac{1}{2\xi_1} \boldsymbol{\lambda}_1 + \frac{1}{2\xi_2} \boldsymbol{\lambda}_2}{\frac{1}{2\xi_1} t + \frac{1}{2\xi_2} (1-t)} \end{bmatrix},$$

we can see that $\begin{bmatrix} \tilde{\boldsymbol{\beta}}(t) \\ \mathbf{0} \end{bmatrix}$ satisfies the KKT condition, and thus it is a solution to formulation

(4). Here $\mathbf{F}_{11,n}$ is the upper-left block of \mathbf{F}_n . Let $c(t) = \tilde{\boldsymbol{\beta}}(t)^\top \mathbf{1}$.

Because $\tilde{\boldsymbol{\beta}}(t)$ is a continuous function with $\tilde{\boldsymbol{\beta}}(0) = \boldsymbol{\beta}_1$ and $\tilde{\boldsymbol{\beta}}(1) = \boldsymbol{\beta}_2$, and $c(t)$ is a continuous function on $[0,1]$ with $c(0) = c_1$ and $c(1) = c_2$. For all $\tilde{c} \in (c_1, c_2)$, there exists $t' \in [0,1]$ such that $c(t') = \tilde{c}$, and the corresponding $\begin{bmatrix} \tilde{\boldsymbol{\beta}}(t') \\ 0 \end{bmatrix}$ is the solution of formulation (4).

Case 2: $c_1, c_2 \notin I$.

In formulation (4), the constraint $\boldsymbol{\beta}^\top \mathbf{1} < c$ is inactive for $c = c_i, i = 1, 2$. Therefore, $\boldsymbol{\beta}(c_1) = \boldsymbol{\beta}(c_2) = \boldsymbol{\beta}(\tilde{c})$ when $\tilde{c} \in (c_1, c_2)$.

Case 3: $c_1 \in I$ and $c_2 \notin I$.

Let $c_0 = \boldsymbol{\beta}(c_2)^\top \mathbf{1}$, then $\boldsymbol{\beta}(c_2) = \boldsymbol{\beta}(c_0)$. If $\tilde{c} \in [c_0, c_2]$, then the solution $\boldsymbol{\beta}(c_2) = \boldsymbol{\beta}(\tilde{c})$ and thus $J(\tilde{c}) = J(c_2)$. If $\tilde{c} \in [c_1, c_0]$, we know $J(c_1) = J(c_2) = J(c_0)$. Then $J(c_0) = J(c_2) = J(\tilde{c})$ is derived from Case 1.

A.6 Proof of Proposition 3

Proof: Let $\mathbf{d}_n = (\mathbf{d}_{1,n}, \mathbf{d}_{2,n})^\top$ where $\mathbf{d}_{1,n} \in \mathbb{R}^{m \times 1}$ and $\mathbf{d}_{2,n} \in \mathbb{R}^{(p-m) \times 1}$. Also, partition

\mathbf{F}_n as $\mathbf{F}_n = \begin{bmatrix} \mathbf{F}_{11,n} & \mathbf{F}_{12,n} \\ \mathbf{F}_{21,n} & \mathbf{F}_{22,n} \end{bmatrix}$, where $\mathbf{F}_{11,n} \in \mathbb{R}^{m \times m}$, and partition \mathbf{F} similarly as $\mathbf{F} =$

$$\begin{bmatrix} \mathbf{F}_{11} & \mathbf{F}_{12} \\ \mathbf{F}_{21} & \mathbf{F}_{22} \end{bmatrix}.$$

If $A(\mathbf{X}_n, \mathbf{Y}_n)$ is true, formulation (4) has a unique optimal solution $\boldsymbol{\beta} = [\boldsymbol{\beta}_1, 0]$ where $\boldsymbol{\beta}_1 > 0$ from Proposition 1. By the KKT condition, $A(\mathbf{X}_n, \mathbf{Y}_n)$ is equivalent with the following statement: there exists $\mu, \xi \geq 0, \boldsymbol{\beta}_1 > 0$ and $\lambda_2 > 0$, such that

$$\begin{cases} 2\xi \mathbf{F}_{11,n} \boldsymbol{\beta}_1 + \mu \mathbf{1}_m = \mathbf{d}_{1,n} \\ 2\xi \mathbf{F}_{21,n} \boldsymbol{\beta}_1 + \mu \mathbf{1}_{p-m} = \mathbf{d}_{2,n} + \boldsymbol{\lambda}_2 \end{cases}$$

From condition (A), we have $\gamma_0 > 0$ and $\mu_0 > 0$ such that

$$\mathbf{F}_{11} \boldsymbol{\gamma}_0 + \mu_0 \mathbf{1}_m = \mathbf{d}_1.$$

By the proof of Proposition 2.6 in Lyons [15], a sample distance covariance for one-dimensional random variables $X^{(i)}, Y$ is a U-statistic. From $E|X^{(i)}|^{2v} < \infty$ and $E|Y|^{2v} < \infty$, we know that the v^{th} moment of the kernel of this U-statistic is finite. By Theorem B in chapter 5.4 in Serfling [88], we have the following result on the deviation between \mathbf{d}_n and \mathbf{d} : for any $\delta > 0$,

$$P(|(\mathbf{d})_i - (\mathbf{d}_n)_i| > \delta) \leq O(n^{1-v}).$$

Similarly, we have

$$P(|(\mathbf{F})_{ij} - (\mathbf{F}_n)_{ij}| > \delta) \leq O(n^{1-v}).$$

Therefore,

$$P(\|\mathbf{d}_1 - \mathbf{d}_{1,n}\| > \delta) \leq O(n^{1-v}),$$

$$P(\|\mathbf{d}_{2,n}\| > \delta) \leq O(n^{1-v}), \text{ and}$$

$$P(\|\mathbf{F}_{11} - \mathbf{F}_{11,n}\| > \delta) \leq O(n^{1-v}).$$

Here and thereafter all norms are denoted as the ℓ_∞ norm. Let $\boldsymbol{\gamma}_0$ be the solution of $\mathbf{F}_{11} \boldsymbol{\gamma}_0 = \mathbf{d}_1 - \mu_0 \mathbf{1}_m$. When $\|\mathbf{F}_{11} - \mathbf{F}_{11,n}\| \leq \delta$ and $\|\mathbf{d}_1 - \mathbf{d}_{1,n}\| \leq \delta$, consider the linear system $\mathbf{F}_{11,n} \boldsymbol{\gamma}_n = \mathbf{d}_{1,n} - \mu_0 \mathbf{1}_m$. From Franklin [89], Section 6.10, equation (16), the solution to

the linear system $\boldsymbol{\gamma}_n$ satisfies $\|\boldsymbol{\gamma}_n - \boldsymbol{\gamma}_0\| \leq K\delta$, where K is a constant independent with n .

Therefore, for any $\delta > 0$ we have $P(\|\boldsymbol{\gamma}_n - \boldsymbol{\gamma}_0\| > \delta) = O(n^{1-v})$.

Now let $D_1 = \left\{ \boldsymbol{\gamma}_n: \|\boldsymbol{\gamma}_n - \boldsymbol{\gamma}_0\| \geq \frac{\|\boldsymbol{\gamma}_0\|}{2} \right\}$, and we have $P(D_1) = O(n^{1-v})$. Also let $D_2 = \{ \mathbf{F}_{21,n}\boldsymbol{\gamma}_n + \mu_0\mathbf{1}_{p-m} \not\geq \mathbf{d}_{2,n} \}$, we have $P(D_2) \leq P(\mu_0\mathbf{1}_{p-m} \not\geq \mathbf{d}_{2,n}) \leq \sum_{i=m+1}^p P(d_{i,n} > \mu_0) = O(n^{1-v})$. In summary, $P(D_1 \cup D_2) = O(n^{1-v})$.

If event D_1 is false, we have $\mathbf{F}_{11,n}\boldsymbol{\gamma}_n + \mu_0\mathbf{1}_m = \mathbf{d}_{1,n}$ with $\boldsymbol{\gamma}_n > 0; \mu_0 > 0$; if D_2 is false, we have $\mathbf{F}_{21,n}\boldsymbol{\gamma}_n + \mu_0\mathbf{1}_{p-m} > \mathbf{d}_{2,n}$. By letting $\hat{\boldsymbol{\lambda}}_2 = \mathbf{F}_{21,n}\boldsymbol{\gamma}_n + \mu_0\mathbf{1}_{p-m} - \mathbf{d}_{2,n} > 0$, $\hat{\boldsymbol{\beta}}_n = \frac{\boldsymbol{\gamma}_n}{\sqrt{\boldsymbol{\gamma}_n^\top \mathbf{F}_n \boldsymbol{\gamma}_n}}$, and $\hat{\xi}_n = \frac{1}{2}\sqrt{\boldsymbol{\gamma}_n^\top \mathbf{F}_n \boldsymbol{\gamma}_n}$, we find that $\hat{\boldsymbol{\beta}}_n, \hat{\xi}_n, \mu_0$ and $\hat{\boldsymbol{\lambda}}_2$ satisfy the KKT condition. In such case, $\hat{\boldsymbol{\beta}}_n$ is the solution of the optimization problem with $c = \hat{\boldsymbol{\beta}}_n^\top \mathbf{1}$. Therefore, the probability that there exist a c such that the optimization solution is consistent is $1 - O(n^{1-v})$.

A.7 Proof of Proposition 4

Proof: First, we claim that if there exists a value $c > 1$ such that formulation (4) has a solution $\boldsymbol{\beta}_n = (\boldsymbol{\beta}_{1,n}^\top 0)^\top$ with $\boldsymbol{\beta}_{1,n} > 0$ then there exists $\boldsymbol{\gamma}_{1,n} \geq 0$ and $\mu_n \geq 0$ such that

$$\mathbf{d}_{1,n} = \mathbf{F}_{11,n}\boldsymbol{\gamma}_{1,n} + \mu_n\mathbf{1}_m.$$

By the KKT condition, formulation (4) has a solution $\boldsymbol{\beta}_n = (\boldsymbol{\beta}_{1,n}^\top 0)^\top$ with $\boldsymbol{\beta}_{1,n} > 0$ if and only if the following equations hold:

$$\begin{cases} 2\xi_n \mathbf{F}_{11,n}\boldsymbol{\beta}_{1,n} = \mathbf{d}_{1,n} - \mu_n\mathbf{1}_m \\ 2\xi_n \mathbf{f}_{1,n}^\top \boldsymbol{\beta}_{1,n} = d_{*,n} - \mu_n + \lambda_{*,n} \\ 2\xi_n \mathbf{F}_{12,n}\boldsymbol{\beta}_{1,n} = \mathbf{d}_{2,n} - \mu_n\mathbf{1}_{p-m-1} + \boldsymbol{\lambda}_{2,n} \end{cases},$$

with $\xi_n \geq 0$; $\mu_n \geq 0$; $\lambda_{*,n} \geq 0$ and $\lambda_{2,n} \geq 0$. Here $\lambda_{*,n}$ is the dual variable to constraint $\beta_{m+1} \geq 0$. As $c > 1$, we know $\xi_n > 0$ from the proof of Propositions 1 and Proposition 2.

The conclusion of (1) is equivalent to that \mathbf{d}_1 is within the cone spanned by the columns of \mathbf{F}_{11} and $\mathbf{1}_m$. Suppose that this is not true, then by separation theorem, there exists some $\mathbf{u} \in \mathbb{R}^m$ such that $\mathbf{d}_1^\top \mathbf{u} > 0$ and $\mathbf{F}_{11}(:, m)^\top \mathbf{u} < 0$, which means that there exists some N , so that $\mathbf{d}_{1,n}^\top \mathbf{u} > 0$ and $\mathbf{F}_{11,n}(:, m)^\top \mathbf{u} < 0$ after $n > N$ with probability 1, which further indicates that for some N , $\mathbf{d}_{1,n}$ is not within the cone spanned the columns of $\mathbf{F}_{11,n}$ when $n > N$ with probability 1. This contradicts the first equation. Now we are clear that $\lim_{n \rightarrow \infty} P(\exists \boldsymbol{\gamma}_{1,n} > 0, \mu_n \geq 0 \text{ s.t. } \mathbf{d}_{1,n} = \mathbf{F}_{11,n} \boldsymbol{\gamma}_{1,n} + \mu_n \mathbf{1}_m) = 1$. We further show that $\exists \boldsymbol{\gamma}_1 > 0, \mu \geq 0$, such that $\mathbf{d}_1 = \mathbf{F}_{11} \boldsymbol{\gamma}_1 + \mu \mathbf{1}_m$.

Let $G_1 = \{(\mathbf{x}_1, \dots, \mathbf{x}_m, \boldsymbol{\gamma}): \exists \lambda_1, \dots, \lambda_m > 0, \mu \geq 0, \text{ s.t. } \mathbf{y} = \sum \lambda_i \mathbf{x}_i + \mu \mathbf{1}\}$. From that $P\left\{\left(\mathbf{F}_{11,n}^{(1)}, \dots, \mathbf{F}_{11,n}^{(m)}, \mathbf{d}_{1,n}\right) \in G_1\right\} \rightarrow 1$, and that $\mathbf{d}_{1,n} \rightarrow \mathbf{d}_1, \mathbf{F}_{11,n} \rightarrow \mathbf{F}_{11}$ with probability 1, we know that $\left(\mathbf{F}_{11}^{(1)}, \dots, \mathbf{F}_{11}^{(m)}, \mathbf{d}_1\right)$ lies in G_1 , i.e., there exist $\boldsymbol{\gamma}_1 \geq 0$ and $\mu \geq 0$ such that $\mathbf{d}_1 = \mathbf{F}_{11} \boldsymbol{\gamma}_1 + \mu \mathbf{1}$.

(2) With assumption 2, the result from (1) indicates

$$\begin{bmatrix} \mathbf{F}_{11}^{(1)} \\ 0 \end{bmatrix} \gamma_1^{(1)} + \begin{bmatrix} \mathbf{F}_{11}^{(2)} \\ 0 \end{bmatrix} \gamma_1^{(2)} + \dots + \begin{bmatrix} \mathbf{F}_{11}^{(m)} \\ 0 \end{bmatrix} \gamma_1^{(m)} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \gamma_* + \mathbf{1}_{m+1} \mu = \begin{bmatrix} \mathbf{d}_1 \\ d_* \end{bmatrix}.$$

with $\boldsymbol{\gamma}_1 = \left[\gamma_1^{(1)}, \dots, \gamma_1^{(m)}\right]^\top \geq 0, \mu \geq 0$. and $\gamma_* = d_* - \mu > 0$ from assumption 3.

From assumption 3, we also have that $\mathbf{F}_{21}\boldsymbol{\gamma}_1 + d_*\mathbf{f}_2 > \mathbf{d}_2$. By letting $\boldsymbol{\lambda} = \mathbf{F}_{12}^\top\boldsymbol{\gamma}_1 + \gamma_*\mathbf{f}_2 - \mathbf{d}_2 + \mu\mathbf{1} = \mathbf{F}_{12}^\top\boldsymbol{\gamma}_1 + d_*\mathbf{f}_2 - \mathbf{d}_2 + \mu(\mathbf{1} - \mathbf{f}_2) > \mathbf{0}$, we have $\boldsymbol{\gamma}_1 \geq \mathbf{0}, \mu \geq 0, \boldsymbol{\lambda} > \mathbf{0}$, such that

$$\begin{bmatrix} \mathbf{F}_{11}^{(1)} \\ \mathbf{0} \\ \mathbf{F}_{12}^{\top(1)} \end{bmatrix} \gamma_1^{(1)} + \dots + \begin{bmatrix} \mathbf{F}_{11}^{(m)} \\ \mathbf{0} \\ \mathbf{F}_{12}^{\top(m)} \end{bmatrix} \gamma_1^{(m)} + \begin{bmatrix} 0 \\ 1 \\ \mathbf{f}_2 \end{bmatrix} \gamma_* = \begin{bmatrix} \mathbf{d}_1 \\ d_* \\ \mathbf{d}_2 \end{bmatrix} - \mu\mathbf{1} + \begin{bmatrix} \mathbf{0} \\ 0 \\ \boldsymbol{\lambda} \end{bmatrix}.$$

With probability one, we have $\mathbf{F}_n^{(k)} \rightarrow \mathbf{F}^{(k)}$ for all $k = 1, 2, \dots, m$, and $\mathbf{d}_n \rightarrow \mathbf{d}$ as $n \rightarrow \infty$. The probability that there exists $\boldsymbol{\gamma}_{1,n} = (\gamma_n^{(1)}, \dots, \gamma_n^{(m)}) \geq \mathbf{0}$ and $\gamma_{*,n} > 0, \mu \geq 0$ and $\boldsymbol{\lambda} > \mathbf{0}$ such that

$$\begin{bmatrix} \mathbf{F}_{11,n}^{(1)} \\ \mathbf{0} \\ \mathbf{F}_{21,n}^{(1)} \end{bmatrix} \gamma_n^{(1)} + \dots + \begin{bmatrix} \mathbf{F}_{11,n}^{(m)} \\ \mathbf{0} \\ \mathbf{F}_{21,n}^{(m)} \end{bmatrix} \gamma_n^{(m)} + \begin{bmatrix} 0 \\ 1 \\ \mathbf{f}_{2,n} \end{bmatrix} \gamma_{*,n} = \begin{bmatrix} \mathbf{d}_{1,n} \\ d_{*,n} \\ \mathbf{d}_{2,n} \end{bmatrix} - \mu\mathbf{1} + \begin{bmatrix} \mathbf{0} \\ 0 \\ \boldsymbol{\lambda} \end{bmatrix}$$

goes to 1.

From $\boldsymbol{\gamma}_n = (\boldsymbol{\gamma}_{1,n}, \gamma_{*,n}, \mathbf{0}_{p-m-1})$, we can calculate $\boldsymbol{\beta}_n = \frac{\boldsymbol{\gamma}_n}{\sqrt{\boldsymbol{\gamma}_n^\top \mathbf{F}_n \boldsymbol{\gamma}_n}}$, and $\xi_n = \frac{1}{2} \sqrt{\boldsymbol{\gamma}_n^\top \mathbf{F}_n \boldsymbol{\gamma}_n}$. Because $(\boldsymbol{\beta}_n, \xi_n, \mu, \boldsymbol{\lambda})$ satisfies the KKT condition of problem (4), $\boldsymbol{\beta}_n$ is a solution to formulation (4).

APPENDIX B.

SUPPLEMENTARY MATERIALS FOR CHAPTER 3B.1 Proofs for

Proposition 6 and Proposition 7

Proof for Proposition 6

The proof follows a similar procedure in [63]. The matrix \mathbf{D}_S can be represented by $\mathbf{C}_d^\top \mathbf{\Lambda} \mathbf{C}_d$, where $\mathbf{\Lambda}$ is a diagonal matrix with

$$\mathbf{\Lambda}_{i,i} = s_{i,d} := 2(1 - \cos[(i-1)\pi/d]); [\mathbf{C}_d]_{i,j} = \begin{cases} \sqrt{\frac{2}{d}} \cos((i-0.5)(j-1)\pi), & j > 1 \\ 1/\sqrt{d}, & j = 1 \end{cases}.$$

Note that \mathbf{C}_d is the type-2 Discrete Cosine Transformation matrix according to [90], and it is an orthogonal matrix. Therefore, we have $\mathbf{D}_S^{-1} = \mathbf{C}_d^\top \mathbf{\Lambda}^{-1} \mathbf{C}_d$. Note that $p_S(\mathbf{x}) = \lambda_1 \mathbf{x}^\top \mathbf{D}_S^\top \mathbf{D}_S \mathbf{x}$ is a quadratic function, and thus $\text{prox}_{\eta p_S}[\mathbf{x}] = (\mathbf{I} + \lambda_1 \mathbf{D}_S^\top \mathbf{D}_S)^{-1} \mathbf{x} = \mathbf{C}_d^\top (\mathbf{I} + \lambda_1 \mathbf{\Lambda}^2)^{-1} \mathbf{C}_d \mathbf{x}$. Therefore, the proximal operator can be calculated by first applying the type-2 discrete cosine transform on \mathbf{x} , then shrink the i -th element of the resulted vector by a factor

$$(\mathbf{I} + \lambda_1 \mathbf{\Lambda}^2)_{i,i}^{-1} = [1 + \lambda_1 s_{i,d}^2]^{-1},$$

and finally apply inverse cosine transform on the signal.

Proof for Proposition 7

The proof of Proposition 7 follows a similar idea of the proof in Proposition 6. According to [65], the discretized matrix \mathbf{R}_I has the following form of eigen value decomposition

$$\mathbf{R}_I = \mathbf{C}_{mn}^\top \mathbf{\Lambda}_{mn} \mathbf{C}_{mn},$$

where $\mathbf{\Lambda}_{mn}^2$ is a $mn \times mn$ diagonal matrix whose diagonal element corresponding to the (i, j) pixel on the image is given as $(s_{i,m} + s_{j,n})^2$, and $\mathbf{C}_{mn} = \mathbf{C}_m \otimes \mathbf{C}_n$ is the matrix that performs 2D type-2 discrete cosine transform over the image signal. Therefore,

$$\text{prox}_{\eta p_S}[\text{vec}(\mathbf{T})] = (\mathbf{I} + \lambda_1 \mathbf{R}_I)^{-1} \mathbf{x} = \mathbf{C}_{mn}^\top (\mathbf{I} + \lambda_1 \mathbf{\Lambda}_{mn}^2)^{-1} \mathbf{C}_{mn} \text{vec}(\mathbf{T})$$

where the $[(\mathbf{I} + \lambda_1 \mathbf{\Lambda}_{mn}^2)^{-1}]_{i,j} = \left(1 + \lambda_1 (s_{i,m} + s_{j,n})^2\right)^{-1}$. Proposition 7 immediately follows.

B.2 The specifications of simulating images and curves in Section 3.4.2

The detailed procedure for generating $\mathbf{B}_{ij,k}$ and \mathbf{B}_{k0} that corresponding to multiple curve or image quality measurements in stage $k = 1, \dots, 4$ are detailed as follows.

Step 1: generate offset matrices \mathbf{B}_{10} and \mathbf{B}_{20}

- First, generate a $m_k \times m_k$ positive definite matrix $\mathbf{\Sigma} = \mathbf{\Gamma}^\top \mathbf{S} \mathbf{\Gamma}$, denoting the cross-covariance between curves generated from stage k .
 - \mathbf{S} is a diagonal matrix, whose m_k diagonal entries are generated from $U(0,1)$.
 - $\mathbf{\Gamma}$ is a random orthogonal matrix of order m_k .
- Then, we generate \mathbf{B}_{k0} from m_k -channel multivariate Gaussian processes (GP) of length n_k , with (1) mean zero; (2) Matérn correlation function with parameter $\nu = 10, \ell = 20$ [91]; and (3) cross-covariance $\mathbf{\Sigma}$.

Step 2: generate offset matrices \mathbf{B}_{30} and \mathbf{B}_{40}

- \mathbf{B}_{k0} is generated from a two-dimensional Gaussian process on a grid $\{(i, j) \in \mathbb{Z}^2, 1 \leq i \leq m_k, 1 \leq j \leq n_k\}$, with (1) mean 0 and (2) Gaussian covariance function with parameter θ_k, σ^2 . We selected $\theta_3 = 0.01, \theta_4 = 0.02$.

Step 3: generate effect matrices $\mathbf{B}_{ij,k}$'s

- First, we generate r_k variation patterns on stage k . Each variation pattern $v = 1, \dots, r_k$ is represented as a matrix $\mathbf{A}_{k,v} \in \mathbb{R}^{m_k \times n_k}$, with the same size of \mathbf{Y}_k . $\mathbf{A}_{k,v}$ is generated with the same procedure as is done for \mathbf{B}_{k0} in the previous two steps corresponding to $k = 1, 2$ or $k = 3, 4$.
- The effect matrix $\mathbf{B}_{ij,k}$ corresponding to every non-effective input u_{ij} is specified as O.
- The effect matrix $\mathbf{B}_{ij,k}$ corresponding to every non-effective input u_{ij} is specified as a fixed linear combination of latent variation patterns $\mathbf{A}_{k,v}$. Therefore, for all effective u_{ij} 's, $\mathbf{B}_{ij,k}$ is generated by $\mathbf{B}_{ij,k} = \sum_{v=1}^{r_k} \xi_{i,j,k,v} \mathbf{A}_{k,v}$, where the coefficients $\xi_{i,j,k,v}$ are randomly selected from $U(0,1)$ random variables.

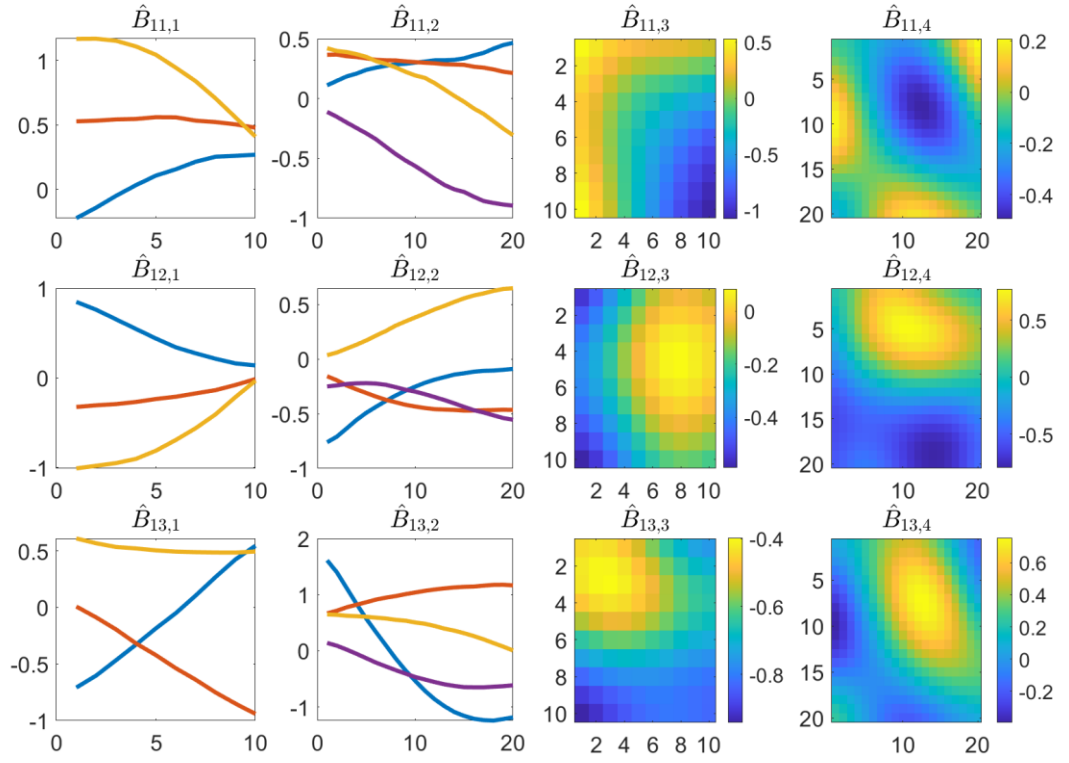
B.3 Illustrations of the estimated parameters in Section 3.4.3

The Figure 0.1 (a)-(d) illustrates the estimated $\widehat{\mathbf{B}}_{ij,k}, 1 \leq i \leq k \leq 4, j = 1, 2, 3$ for the parameter $\mathcal{B} = \{\mathbf{B}_{ij,k}: 1 \leq i \leq k \leq 4; j = 1, \dots, 10\}$ in model (1),

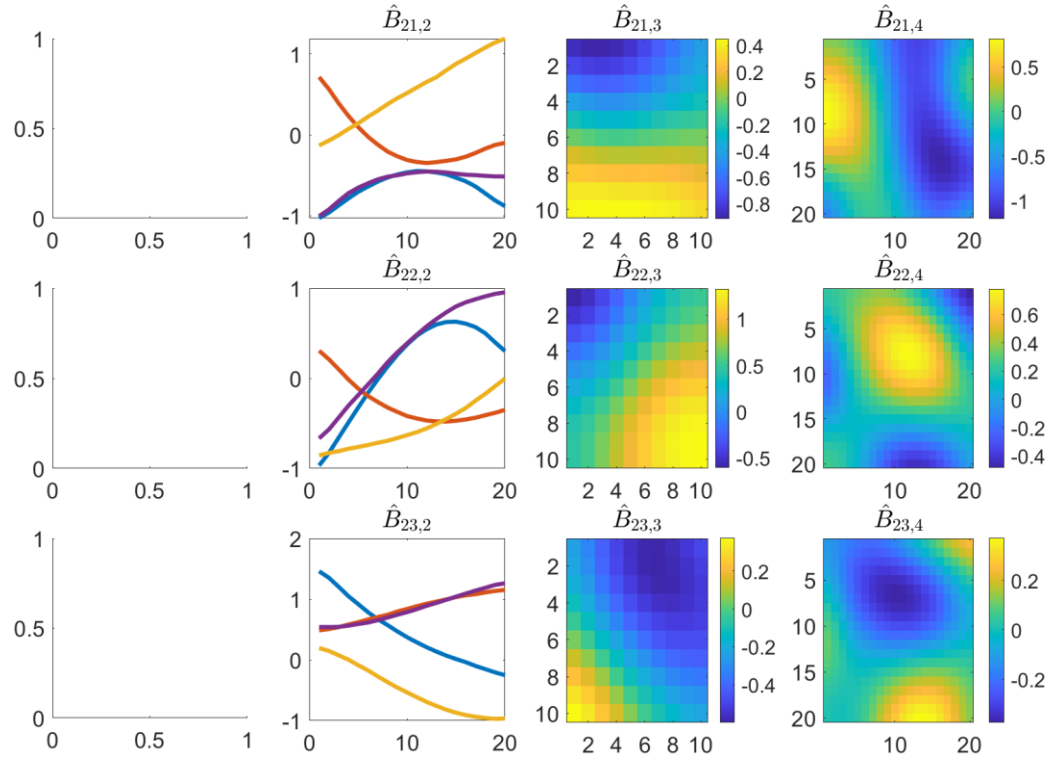
$$\mathbf{Y}_k = \mathbf{B}_{k0} + \sum_{i=1}^k \sum_{j=1}^{q_i} u_{ij} \mathbf{B}_{ij,k} + \mathbf{E}_k \quad (1)$$

for one run of the simulation Setting (1). Here, every matrix $\mathbf{B}_{ij,k}$ represents how the j th process inputs from stage i relate to the process output at stage k . In this run of the simulation, $\widehat{\mathbf{B}}_{ij,k} = \mathbf{0}$ for all $j = 4, \dots, 10$, and thus all corresponding u_{ij} 's are correctly identified as ineffective inputs. Above each subfigure in Figure 0.1 (a)-(d) is the parametric matrix that it illustrates. Specifically, Figure 0.1 (a) illustrates $\mathbf{B}_{1j,k}$ for $j = 1, 2, 3$ and $k = 1, \dots, 4$, representing how the first three inputs in stage 1 affect stages 1, ..., 4. The subfigures in the first two columns in Figure 0.1 (a) correspond to the effect matrices for outputs from stages 1 and 2, and thus they are in the form of smooth curves. The last two columns in Figure 0.1 (a) correspond to the effect matrices for the output from stages 3 and 4, and they are in the form of smooth images. These forms are consistent with the true system parameter in \mathcal{B} .

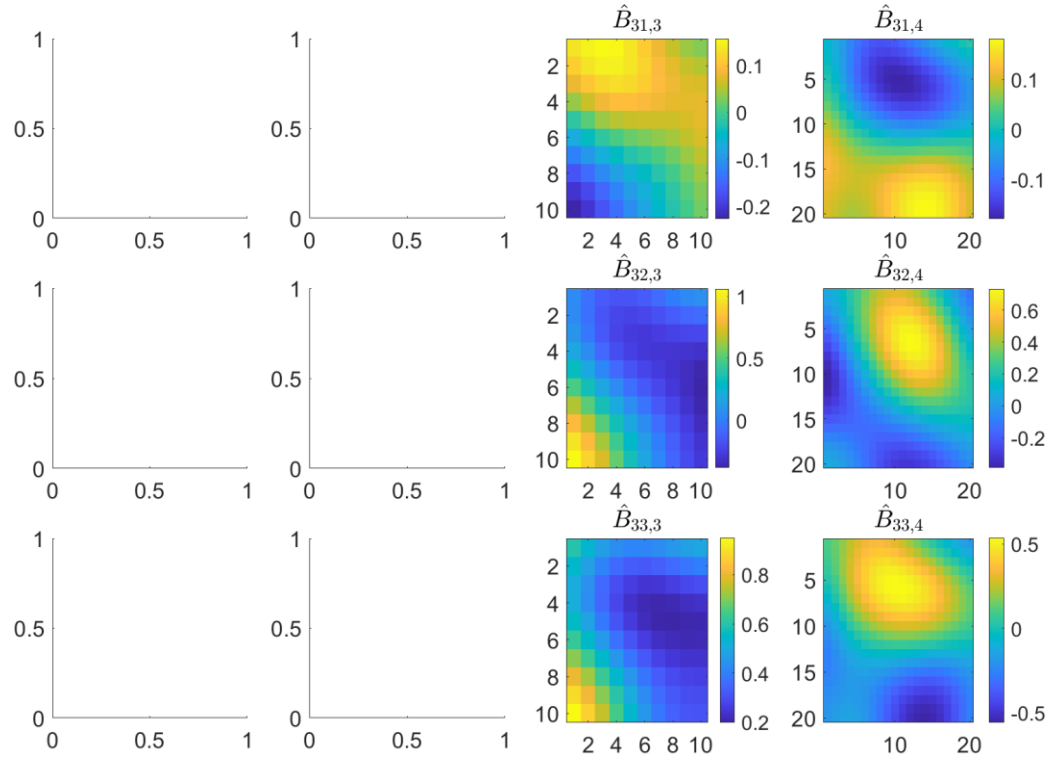
The interpretation of Figure 0.1 (b)-(d) are similar. For example, Figure 0.1 (b) illustrates the effect of the process inputs from stage 2 on the process output for other stages. Due to the cascading assumption, the process inputs from stage 2 have no effect on the process outputs for stage 1, and thus the subfigures in the first column of Figure 0.1 (b) are blank. The effect of process inputs from stage 2 on process outputs of stage 2 is in the form of multiple smooth curves, and the effect of process inputs from stage 2 on process outputs of stages 3 and 4 are in the form of smooth images, as shown in the second, third and fourth columns of Figure 0.1 (b). Figures 0.1 (c) and (d) have two and three columns, respectively, that are blank. The reason is that the process inputs from stage 3 only affect the outputs for stage 3 and 4, and process inputs from stage 4 only affect the outputs for stage 4.



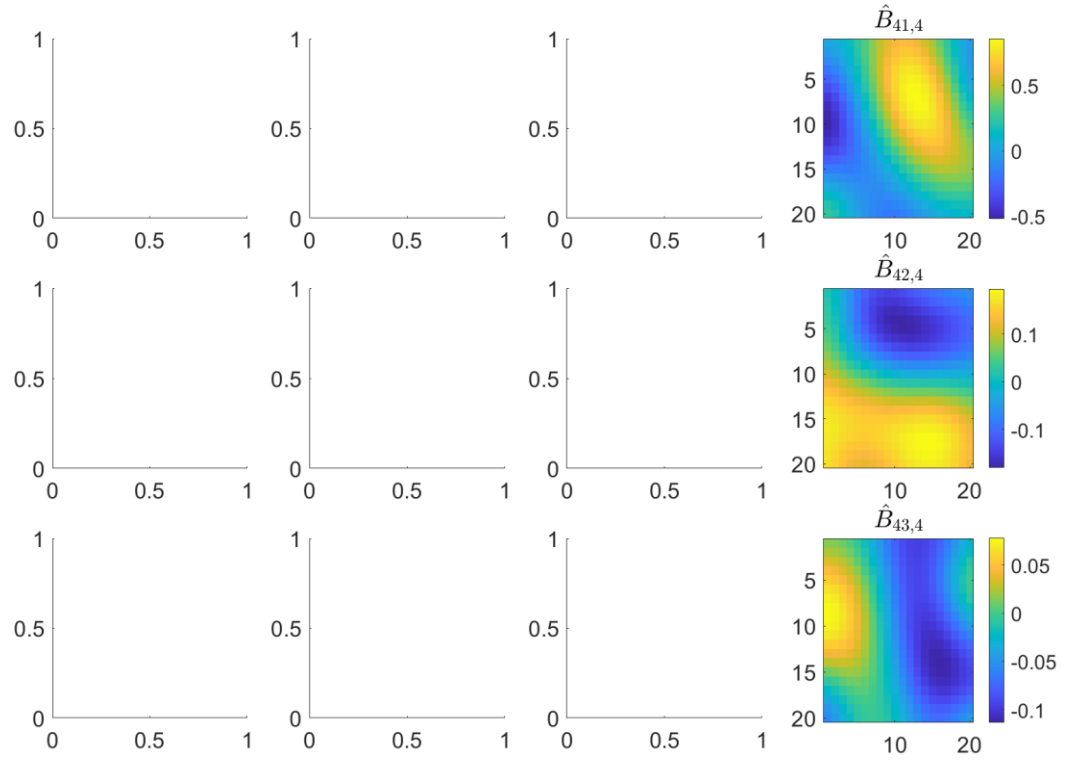
(a) The matrices $\hat{\mathbf{B}}_{1j,k}$ for $j = 1, 2, 3$ and $k = 1, 2, 3$ and 4, representing the effects of effective inputs from stage 1 to the outputs in all stages.



(b) The matrices $\hat{\mathbf{B}}_{2j,k}$ for $j = 1, 2, 3$ and $k = 2, 3$ and 4, representing the effects of effective inputs from stage 2 to the outputs in the last three stages.



(c) The matrices $\hat{\mathbf{B}}_{3j,k}$ for $j = 1,2,3$ and $k = 3,4$, representing the effects of effective inputs from stage 3 to the outputs in the last two stages.



(d) The matrices $\hat{\mathbf{B}}_{4j,k}$ for $j = 1,2,3$ and $k = 4$, representing the effects of the inputs from the last stage to the outputs in the last stage.

Figure 0.1 The estimated matrices $\hat{\mathbf{B}}_{ij,k}$, for $i = 1, 2, 3$ and 4 respectively.

APPENDIX C.

SUPPLEMENTARY MATERIALS FOR CHAPTER 4

C.1 Derivation of Proposition 10

The proximal operator of ηf_1 The $f_1(\mathbf{B})$ can be written as a least-square form using

$$f_1(\mathbf{B}) = \|\mathbf{X} - \mathbf{H}\mathbf{B}\mathbf{Y}\|_F^2 = \sum_{i=1}^I \|\mathbf{X}_i - \mathbf{H}_i \mathbf{B}_{\cdot,i} \mathbf{Y}\|_F^2 = \sum_{i=1}^I \|\text{vec}(\mathbf{X}_i) - (\mathbf{Y}^\top \otimes \mathbf{H}_i) \text{vec}(\mathbf{B}_{\cdot,i})\|_2^2$$

According to Parikh and Boyd [61], the proximal operator for $q(\mathbf{x}) = \|\mathbf{c} - \mathbf{P}\mathbf{x}\|_2^2$ is

$$\text{prox}_{\eta q}[\mathbf{x}] := \underset{\mathbf{t}}{\text{argmin}} \left\{ \eta q(\mathbf{t}) + \frac{1}{2} \|\mathbf{t} - \mathbf{x}\|_2^2 \right\} = (\mathbf{E} + \eta \mathbf{P}^\top \mathbf{P})^{-1} (\mathbf{x} + \eta \mathbf{P}^\top \mathbf{c}). \quad (1)$$

In our setting, we let $\mathbf{P}_i = \mathbf{Y}^\top \otimes \mathbf{H}_i$, $\mathbf{c} = \text{vec}(\mathbf{X}_i)$, and thus

$$\mathbf{P}_i^\top \mathbf{P}_i = (\mathbf{Y}\mathbf{Y}^\top) \otimes \mathbf{E}_{J_i \times J_i}; \mathbf{P}_i^\top \mathbf{c} = (\mathbf{Y} \otimes \mathbf{H}_i^\top) \text{vec}(\mathbf{X}_i) = \text{vec}(\mathbf{H}_i^\top \mathbf{X}_i \mathbf{Y}^\top)$$

Due to the separable property of the proximal operator, we know if $\mathbf{Z} = \text{prox}_{\eta f_1}[\mathbf{A}]$, we have

$$\text{vec}(\mathbf{Z}_{\cdot,i}) = (\mathbf{E}_{KJ_i \times KJ_i} + \eta \mathbf{P}_i^\top \mathbf{P}_i)^{-1} [\text{vec}(\mathbf{A}_{\cdot,i}) + \eta \mathbf{P}_i^\top \mathbf{c}]$$

Thus, the result follows.

The proximal operator of ηf_2 This term can be obtained from the proximal operator of ℓ_1 norm, and it is given in Parikh and Boyd [61].

The proximal operator of ηf_3 This term is the sum of ℓ_2 norms. By the separable property of the proximal operator, the result follows from the proximal operator of ℓ_2 norms given in Parikh and Boyd [61]: $\text{prox}_{\eta \|\cdot\|_2}[\mathbf{x}] = \left[1 - \frac{\lambda}{\|\mathbf{x}\|_2}\right]_+ \mathbf{x}$.

The proximal operator of ηf_4 The term $f_4(\mathbf{B}) = \sum_{k=1}^K I_{\|\mathbf{b}_{k,\cdot}\|_2=1}$ is separable for all $\mathbf{b}_{1,\cdot}, \dots, \mathbf{b}_{K,\cdot}$. The result follows from $\text{prox}_{I_{\|\cdot\|_2=1}}[\mathbf{x}] = \text{proj}_{\{\mathbf{z}:\|\mathbf{z}\|_2=1\}}[\mathbf{x}] = \frac{\mathbf{x}}{\|\mathbf{x}\|_2}$.

C.2 Proof of Proposition 11

The proximal operator of ηf_1 Let $\mathbf{y}_{\cdot,t}$ be the t -th column of \mathbf{Y} , and we have $f_1(\mathbf{Y}) = \sum_{t=1}^T \|\mathbf{x}_{\cdot,t} - \mathbf{H}\mathbf{B}\mathbf{y}_{\cdot,t}\|_2^2$. By the separable property of the proximal operator, we take $\mathbf{P}_t = \mathbf{H}\mathbf{B}, \mathbf{c} = \mathbf{x}_{\cdot,t}$. Using Equation (1), we derive that if $\mathbf{Z} = \text{prox}_{\eta f_1}[\mathbf{A}]$,

$$\mathbf{Z}_{\cdot,t} = (\mathbf{E}_{K \times K} + \eta \mathbf{B}^\top \mathbf{B})^{-1} [\mathbf{A}_{\cdot,t} + \eta \mathbf{B}^\top \mathbf{H}^\top \mathbf{x}_{\cdot,t}], \text{ for } t = 1, \dots, T.$$

The proximal operator of ηf_2 See the proof of Proposition 2 in Wang and Shi [8].

The proximal operator of ηf_3 Note that $f_3(\mathbf{Y}) = \lambda_4 \|\mathbf{Y}\|_{1,1} + \sum_{t=1}^T \sum_{k=1}^K I_{y_{k,t} \geq 0} = \sum_{k,t} [|y_{k,t}| + I_{y_{k,t} \geq 0}]$. For $q(y) = |y| + I_{y \geq 0}$, we have

$$\begin{aligned} \text{prox}_{\eta q}[y] &= \underset{t}{\text{argmin}} \left\{ \eta |t| + \eta I_{t \geq 0} + \frac{1}{2} |t - y|^2 \right\} \\ &= \underset{t \geq 0}{\text{argmin}} \{ \eta t + (t - y)^2 \} = \max(y - \eta, 0). \end{aligned}$$

Due to the separable property of f_3 , we know that if $\mathbf{Z} = \text{prox}_{\eta f_3}[\mathbf{A}]$, $Z_{k,t} = \max(A_{k,t} - \lambda_4 \eta, 0)$ for all $k = 1, \dots, K$ and $t = 1, \dots, T$.

REFERENCES

- [1] J. Shi, *Stream of variation modeling and analysis for multistage manufacturing processes*. CRC Press, 2006.
- [2] R. Jin and J. Shi, "Reconfigured piecewise linear regression tree for multistage manufacturing process control," *IIE Trans*, vol. 44, no. 4, pp. 249-261, 2012.
- [3] J. Li and J. Shi, "Knowledge discovery from observational data for process control using causal Bayesian networks," *IIE Trans*, vol. 39, no. 6, pp. 681-690, 2007.
- [4] D. C. Montgomery, *Introduction to statistical quality control*. John Wiley & Sons, 2007.
- [5] K. Abidin, K. Lee, I. Ibrahim, and A. Zainudin, "Problem Analysis at a Semiconductor Company: A Case Study on IC Packages," *Journal of Applied Sciences*, pp. 1-8, 2011.
- [6] V. Vakharia, V. Gupta, and P. Kankar, "A comparison of feature ranking techniques for fault diagnosis of ball bearing," *Soft Computing*, vol. 20, no. 4, pp. 1601-1619, 2016.
- [7] C. Shao *et al.*, "Feature selection for manufacturing process monitoring using cross-validation," *Journal of Manufacturing Systems*, vol. 32, no. 4, pp. 550-555, 2013.
- [8] A. Wang and J. Shi, "Holistic modeling and analysis of multistage manufacturing processes with sparse effective inputs and mixed profile outputs," *IIE Transactions*, pp. 1-15, 2020, doi: 10.1080/24725854.2020.1786197.
- [9] A. Ben-Tal and A. Nemirovski, *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*. Siam, 2001.
- [10] S. Weisberg, *Applied linear regression*. John Wiley & Sons, 2005.
- [11] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267-288, 1996.

- [12] U. Grömping, "Relative importance for linear regression in R: the package relaimpo," *Journal of statistical software*, vol. 17, no. 1, pp. 1-27, 2006.
- [13] N.-H. Choi, K. Shedden, G. Xu, X. Zhang, and J. Zhu, "Comment: Ridge Regression, Ranking Variables and Improved Principal Component Regression," *Technometrics*, vol. 62, no. 4, pp. 451-455, 2020.
- [14] R. Steuer, J. Kurths, C. O. Daub, J. Weise, and J. Selbig, "The mutual information: detecting and evaluating dependencies between variables," *Bioinformatics*, vol. 18, no. Suppl. 2, pp. S231-S240, 2002.
- [15] R. Lyons, "Distance Covariance in Metric Spaces," (in English), *Ann Probab*, vol. 41, no. 5, pp. 3284-3305, Sep 2013.
- [16] G. J. Székely, M. L. Rizzo, and N. K. Bakirov, "Measuring and testing dependence by correlation of distances," *The annals of statistics*, vol. 35, no. 6, pp. 2769-2794, 2007.
- [17] A. Gretton, R. Herbrich, A. Smola, O. Bousquet, and B. Schölkopf, "Kernel methods for measuring independence," *J Mach Learn Res*, vol. 6, no. Dec, pp. 2075-2129, 2005.
- [18] A. Gretton *et al.*, "Kernel Constrained Covariance for Dependence Measurement," in *AISTATS*, 2005, vol. 10, pp. 112-119.
- [19] G. J. Székely and M. L. Rizzo, "Energy statistics: A class of statistics based on distances," *Journal of statistical planning and inference*, vol. 143, no. 8, pp. 1249-1272, 2013.
- [20] J. D. Huling and S. Mak, "Energy Balancing of Covariate Distributions," *arXiv preprint arXiv:2004.13962*, 2020.
- [21] D. Sejdinovic, B. Sriperumbudur, A. Gretton, and K. Fukumizu, "Equivalence of Distance-Based and Rkhs-Based Statistics in Hypothesis Testing," (in English), *Ann Stat*, vol. 41, no. 5, pp. 2263-2291, Oct 2013, doi: 10.1214/13-Aos1140.
- [22] L. Song, A. Smola, A. Gretton, J. Bedo, and K. Borgwardt, "Feature Selection via Dependence Maximization," (in English), *J Mach Learn Res*, vol. 13, pp. 1393-1434, May 2012. [Online]. Available: <Go to ISI>://WOS:000305456600005.

- [23] R. Z. Li, W. Zhong, and L. P. Zhu, "Feature Screening via Distance Correlation Learning," *J Am Stat Assoc*, vol. 107, no. 499, pp. 1129-1139, Sep 2012.
- [24] J. Kong, S. J. Wang, and G. Wahba, "Using distance covariance for improved variable selection with application to learning genetic risk models," (in English), *Stat Med*, vol. 34, no. 10, pp. 1708-1720, May 10 2015, doi: 10.1002/sim.6441.
- [25] C. D. Yenigun and M. L. Rizzo, "Variable selection in regression using maximal correlation and distance correlation," (in English), *J Stat Comput Sim*, vol. 85, no. 8, pp. 1692-1705, May 24 2015, doi: 10.1080/00949655.2014.895354.
- [26] A.-A. Christidis, L. Lakshmanan, E. Smucler, and R. Zamar, "Split Regularized Regression," *Technometrics*, vol. 62, no. 3, pp. 330-338, 2020.
- [27] H. C. Peng, F. H. Long, and C. Ding, "Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy," (in English), *Ieee Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226-1238, Aug 2005, doi: Doi 10.1109/Tpami.2005.159.
- [28] X. Huo and G. J. Székely, "Fast computing for distance covariance," *Technometrics*, vol. 58, no. 4, pp. 435-447, 2016.
- [29] G. J. Székely and M. L. Rizzo, "Testing for equal distributions in high dimension," *InterStat*, vol. 5, no. 16.10, pp. 1249-1272, 2004.
- [30] G. J. Székely and M. L. Rizzo, "The energy of data," *Annual Review of Statistics and Its Application*, vol. 4, pp. 447-479, 2017.
- [31] N. Simon and R. Tibshirani, "Comment on" Detecting Novel Associations In Large Data Sets" by Reshef Et Al, Science Dec 16, 2011," *arXiv preprint arXiv:1401.7645*, 2014.
- [32] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of statistics*, vol. 32, no. 2, pp. 407-499, 2004.
- [33] T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu, "The entire regularization path for the support vector machine," *J Mach Learn Res*, vol. 5, no. Oct, pp. 1391-1415, 2004.

- [34] S. Rosset and J. Zhu, "Piecewise linear regularized solution paths," *The Annals of Statistics*, vol. 35, no. 3, pp. 1012-1030, 2007.
- [35] R. J. Tibshirani and J. Taylor, "The Solution Path of the Generalized Lasso," (in English), *Ann Stat*, vol. 39, no. 3, pp. 1335-1371, Jun 2011.
- [36] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani, "Pathwise coordinate optimization," *The annals of applied statistics*, vol. 1, no. 2, pp. 302-332, 2007.
- [37] R. H. Lindeman, "Introduction to bivariate and multivariate analysis," 1980.
- [38] H. Zou, "The adaptive lasso and its oracle properties," (in English), *J Am Stat Assoc*, vol. 101, no. 476, pp. 1418-1429, 2006.
- [39] J. Huang, S. Ma, and C.-H. Zhang, "Adaptive Lasso for sparse high-dimensional regression models," *Stat Sinica*, pp. 1603-1618, 2008.
- [40] W. H. McIlhagga, "penalized: A MATLAB toolbox for fitting generalized linear models with penalties," 2016.
- [41] A. Altmann, L. Toloşi, O. Sander, and T. Lengauer, "Permutation importance: a corrected feature importance measure," *Bioinformatics*, vol. 26, no. 10, pp. 1340-1347, 2010.
- [42] A. J. McEvoy, L. Castaner, and T. Markvart, *Solar cells: materials, manufacture and operation*. Academic Press, 2012.
- [43] J. Du and X. Zhang, "Online Multichannel Sensing Data Monitoring for Solar Cell Manufacturing Process," Chinese Patent ZL 201710580158.4, 2019.
- [44] Y. Nishi and R. Doering, *Handbook of semiconductor manufacturing technology*. CRC Press, 2000.
- [45] H. De Witte *et al.*, "In-line electrical metrology for high-K gate dielectrics deposited by atomic layer CVD," *Journal of The Electrochemical Society*, vol. 150, no. 9, pp. F169-F172, 2003.

- [46] C.-C. K. Huang and D. Tien, "Overlay goes high order," *Microlithography World*, 2008.
- [47] T. Y. Lee *et al.*, "Study of critical dimension and overlay measurement methodology using SEM image analysis for process control," in *Metrology, Inspection, and Process Control for Microlithography XX*, 2006, vol. 6152: International Society for Optics and Photonics, p. 61522E.
- [48] C. Zhang, H. Yan, S. Lee, and J. Shi, "Multiple profiles sensor-based monitoring and anomaly detection," *Journal of Quality Technology*, vol. 50, no. 4, pp. 344-362, 2018.
- [49] D. W. Apley and J. Shi, "Diagnosis of multiple fixture faults in panel assembly," *Journal of manufacturing science and engineering*, vol. 120, no. 4, pp. 793-801, 1998.
- [50] Y. Ding, D. Ceglarek, and J. Shi, "Modeling and diagnosis of multistage manufacturing processes: part I: state space model," in *Proceedings of the 2000 Japan/USA symposium on flexible automation*, 2000, pp. 23-26.
- [51] J. O. Ramsay, "Functional data analysis," *Encyclopedia of Statistical Sciences*, vol. 4, 2004.
- [52] Y. Li, H. Sun, X. Deng, C. Zhang, H.-P. Wang, and R. Jin, "Manufacturing quality prediction using smooth spatial variable selection estimator with applications in aerosol jet® printed electronics manufacturing," *IISE Transactions*, vol. 52, no. 3, pp. 321-333, 2020.
- [53] H. Yan, K. Paynabar, and J. Shi, "Image-based process monitoring using low-rank tensor decomposition," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 1, pp. 216-227, 2014.
- [54] X. Yue, J. G. Park, Z. Liang, and J. Shi, "Tensor Mixed Effects Model with Application to Nanomanufacturing Inspection," *Technometrics*, pp. 1-14, 2019.
- [55] M. R. Gahrooei, H. Yan, K. Paynabar, and J. Shi, "Multiple Tensor-on-Tensor Regression: An Approach for Modeling Processes With Heterogeneous Sources of Data," *Technometrics*, pp. 1-23, 2019, doi: 10.1080/00401706.2019.1708463.

- [56] J. O. Ramsay, "Monotone regression splines in action," *Statistical science*, vol. 3, no. 4, pp. 425-441, 1988.
- [57] M. Yuan, A. Ekici, Z. Lu, and R. Monteiro, "Dimension reduction and coefficient estimation in multivariate linear regression," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 69, no. 3, pp. 329-346, 2007.
- [58] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 1, pp. 91-108, 2005.
- [59] H. Yan, K. Paynabar, and J. Shi, "Anomaly detection in images with smooth background via smooth-sparse decomposition," *Technometrics*, vol. 59, no. 1, pp. 102-114, 2017.
- [60] C. Zhang, H. Yan, S. Lee, and J. Shi, "Dynamic Multivariate Functional Data Modeling via Sparse Subspace Learning," *Technometrics*, pp. 1-14, 2020.
- [61] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127-239, 2014.
- [62] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1-122, 2011.
- [63] F. O'Sullivan, "Discretized Laplacian smoothing by Fourier methods," *J Am Stat Assoc*, vol. 86, no. 415, pp. 634-642, 1991.
- [64] G. Wahba, *Spline models for observational data*. SIAM, 1990.
- [65] M. Buckley, "Fast computation of a discretized thin-plate smoothing spline for image data," *Biometrika*, vol. 81, no. 2, pp. 247-258, 1994.
- [66] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," (in English), *J Roy Stat Soc B*, vol. 68, pp. 49-67, 2006.

- [67] M. Fazel, H. Hindi, and S. P. Boyd, "A rank minimization heuristic with application to minimum order system approximation," in *Proceedings of the American control conference*, 2001, vol. 6: Citeseer, pp. 4734-4739.
- [68] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE transactions on Computers*, vol. 100, no. 1, pp. 90-93, 1974.
- [69] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU press, 2012.
- [70] J. Jin and J. Shi, "Feature-preserving data compression of stamping tonnage information using wavelets," *Technometrics*, vol. 41, no. 4, pp. 327-339, 1999.
- [71] C. Zhang, H. Yan, S. Lee, and J. Shi, "Weakly correlated profile monitoring based on sparse multi-channel functional principal component analysis," *IISE Transactions*, vol. 50, no. 10, pp. 878-891, 2018.
- [72] C. K. Koh, J. Shi, W. Williams, and J. Ni, "Multiple Fault Detection and Isolation Using the Haar Transform, Part 2: Application to the Stamping Process," *AMSE Transactions, Journal of Manufacturing Science and Engineering-transactions*, vol. 121, pp. 295-299, 1999.
- [73] W. Wang *et al.*, "Multiple event detection and recognition through sparse unmixing for high-resolution situational awareness in power grid," *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 1654-1664, 2014.
- [74] F. Li, Y. Shi, A. Shinde, J. Ye, and W. Song, "Enhanced cyber-physical security in internet of things through energy auditing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5224-5231, 2019.
- [75] R. Guo, K. Guo, and J. Dong, "Phase partition and online monitoring for batch process based on multiway BEAM," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 4, pp. 1582-1589, 2016.
- [76] C. Zhao, "A quality-relevant sequential phase partition approach for regression modeling and quality prediction analysis in manufacturing processes," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 4, pp. 983-991, 2013.

- [77] W. H. Woodall and D. C. Montgomery, "Some current directions in the theory and application of statistical process monitoring," *Journal of Quality Technology*, vol. 46, no. 1, pp. 78-94, 2014.
- [78] Y. Wang, Y. Mei, and K. Paynabar, "Thresholded multivariate principal component analysis for phase I multichannel profile monitoring," *Technometrics*, vol. 60, no. 3, pp. 360-372, 2018.
- [79] S. Ebrahimi, C. Ranjan, and K. Paynabar, "Monitoring and root-cause diagnostics of high-dimensional data streams," *Journal of Quality Technology*, pp. 1-24, 2020.
- [80] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning* (no. 10). Springer series in statistics New York, 2001.
- [81] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Advances in neural information processing systems*, 2007, pp. 801-808.
- [82] H. Yan, K. Paynabar, and J. Shi, "Real-time monitoring of high-dimensional functional data streams via spatio-temporal smooth sparse decomposition," *Technometrics*, vol. 60, no. 2, pp. 181-197, 2018.
- [83] S. Mou, A. Wang, C. Zhang, and J. Shi, "Additive Tensor Decomposition Considering Structural Data Information," *arXiv preprint arXiv:2007.13860*, 2020.
- [84] D. L. Donoho and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *J Am Stat Assoc*, vol. 90, no. 432, pp. 1200-1224, 1995.
- [85] S. M. Ross, *Stochastic processes*, 2nd ed. New York: John Wiley, 1996.
- [86] W. L. Roberts, *Hot rolling of steel*. CRC Press, 1983.
- [87] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Le Bret, "Applications of second-order cone programming," *Linear algebra and its applications*, vol. 284, no. 1-3, pp. 193-228, 1998.
- [88] R. J. Serfling, *Approximation theorems of mathematical statistics*. John Wiley & Sons, 1980.

- [89] J. N. Franklin, *Matrix theory*. Courier Corporation, 2012.
- [90] G. Strang, "The discrete cosine transform," *Siam Rev*, vol. 41, no. 1, pp. 135-147, 1999.
- [91] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*, 2003: Springer, pp. 63-71.