SPARSE MACHINE LEARNING METHODOLOGY AND ITS APPLICATIONS TO SEMICONDUCTOR MANUFACTURING PROCESSES

By

JEONGSUB CHOI

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Industrial and Systems Engineering

Written under the direction of

Myong K. Jeong

And approved by

_____

_____

_____

_____

_____

New Brunswick, New Jersey

October, 2020

ABSTRACT OF THE DISSERTATION

SPARSE MACHINE LEARNING METHODOLOGY AND ITS APPLICATIONS TO SEMICONDUCTOR MANUFACTURING PROCESSES

by JEONGSUB CHOI

Dissertation Director:

Dr. Myong K. Jeong

In this dissertation, we present new methodologies in machine learning for sparse solutions and the applications to semiconductor manufacturing processes. First, we present a new variant of relevance vector machine, called restricted relevance vector machine (RRVM), for incomplete data. Imputation is a common remedy to handle incomplete data that hinders from training a relevance vector machine (RVM) model. Imputation in kernel space for RVM leads to its prediction performance superior to imputation in original space but causes the loss of model sparsity. RRVM restricts its basis to be from complete instances incorporating incomplete instances for training. The experimental results show that RRVM performs prediction with a competitive accuracy, maintaining its model sparsity.

Next, we propose a new estimation method for Gaussian kernels with incomplete data. Gaussian kernels have been extensively used in kernel methods. A recent study proposes the estimation of the Gaussian kernels with incomplete data based on a function of the squared Euclidean distance between incomplete instances that is the sum of

independent squared unit-dimensional distances, and it overlooks the correlations between missing unit-dimensional distances. In the proposed method, we model the squared Euclidean distance between incomplete instances as the sum of correlated squared unit-dimensional distances and estimate the Gaussian kernel from the expected kernel function under the distribution for the squared Euclidean distance between the instances. The experimental results show that the proposed method improves the prediction performance in a kernel method when missing components are correlated.

Furthermore, we present a new autoencoder for feature extraction from multistep process signals. Autoencoder is a neural network that reconstructs an input while representing the input in a lower-dimensional space from which features are obtained. The nature of the input from multistep process signals, however, is neglected by the autoencoder. The proposed autoencoder aims to extract features with smooth reconstruction by a fusion regularization on neighboring signals and with clipped penalties caused by the transient changes of the signals between consecutive subprocesses. A case study for virtual metrology at an etching process shows that the proposed method provides features for superior prediction performance.

Finally, we propose a new regularization, group-exclusive group lasso (GGL), in deep neural networks for automatic exclusive feature group selection. With group-level sparsity, group lasso facilitates the selection of feature groups, but it is difficult to avoid the coincident selection of the feature groups that are group-level correlated and that share their predictability to a response. GGL aims to enforce exclusive sparsity at an inter-group level to select salient feature groups. The experimental results show that GGL leads to higher feature group sparsity, maintaining competitive prediction accuracy.

Acknowledgement

First, I owe my deepest gratitude to my academic advisor, Dr. Myong K. Jeong, for his guidance, motivation, and encouragement throughout my graduate study at Rutgers University. Dr. Jeong is a patient and supportive mentor and has kindly considered my academic improvements and helped me with his precision and unfailing support since the beginning of my studies at Rutgers University.

I am also grateful to my dissertation committee members, Dr. Hoang Pham, Dr. Jie Gong, Dr. Weihong Guo, and Dr. Zhimin Xi for their insightful comments, support and encouragement.

My thanks also go to my friends and colleagues, Jinho, Suyeon, Byunghoon, Youngdoo, Minkoo, HsinYi, Wonbin, Behnam, Ali, and Mengmeng, who helped me go through tough times and made my life in Rutgers enjoyable and memorable.

Finally, I would not have been able to finish my dissertation and degrees without unwavering support from my parents and brother who have been behind me all the time.

Dedication

To my family

Table of Contents

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

1.1 Overview

Wafers for integrated circuits (ICs) are processed through complex fabrication processes (Fenner et al. 2005). According to the increase of the use of ICs, wafer fabrication is performed with highly automated facilities and the process equipment for mass production, and it is critical to assure reliable wafer fabrication and therefore in-control product qualities from process variations (Diebold, 1995; Hung et al., 2007).

A key to achieving successful wafer fabrication is building an accurate monitoring model that predicts wafer qualities. Modeling to monitor wafers is typically based on physical measurements on quality characteristics from processed wafers (He and Wang, 2007; Hung et al., 2007; Weiss et al., 2010), but data collection is a common issue facing investigators due to costs for production processing and measurement equipment (Yugma et al., 2015). On the other hand, wafer quality prediction can be performed with fabrication information from process-equipment sensors, which is called virtual metrology (VM). VM is a tool that describes processed wafer qualities by the soft sensor signals from process-equipment as in-process information, instead of physical measurements, (Vallejo et al., 2019), and so reliable VM can lead to cost reduction and time-saving, implementing predictive and/or prospective maintenances based on process health status provided by VM and replacing costly physical measurements and metrology devices (Cheng et al., 2011; Chen et al., 2005). For successful VM modeling, many features are extracted from high-

dimensional data collected from numerous sensors (Hirai and Kano, 2015; Hwang et al., 2014; Kang et al., 2011; Susto et al., 2015). In this dissertation, we focus on the development of methodologies for sparse machine learning models considering data properties and the applications to wafer quality prediction in semiconductor manufacturing processes.

For accurate prediction of wafer qualities, a great deal of research has been conducted to exploit available information from the partially observed measurements of incomplete instances (Chien et al., 2007; He and Wang, 2007; Hsu and Chien, 2007; Purwins et al., 2014), wherein missing components in data can occur from various reasons such as sensing failures, mistakes in measuring and storing data, and refusals of respondents (Kadlec et al., 2009). For the inclusion of incomplete data, imputation is commonly adopted to make incomplete instances available as complete data (Hung et al., 2007; Lee et al., 2019; Pan and Tai, 2009; Wu et al., 2011; Zeng and Spanos, 2009), and imputation in feature space for nonlinear prediction models outperforms imputation in original data space (Von Hippel, 2009). The imputation in feature space, however, leads to the loss of model sparsity in the predictive distribution of the relevance vector machine. Therefore, we proposed a new relevance vector machine that performs prediction with a competitive accuracy, maintaining model sparsity, by restricting model basis to be from complete instances.

A recent study (Mesquita et al., 2019) proposes the estimation of Gaussian kernels with incomplete data as the expected Gaussian kernel (EGK) aiming to the estimation of the kernels in feature space. However, EGK overlooks the correlation between missing components in the estimation of the squared Euclidean distance between two incomplete

instances for the estimation of the Gaussian kernel of the instances. Therefore, we propose a new estimation method, the expected Gaussian kernel with correlated variables (EGKC), considering the correlations among missing components. In EGKC, a Gamma distribution for the squared Euclidean distance between incomplete instances is estimated by the approximation of the sum of the correlated Gamma distributions based on the approximation in (Feng et al., 2016) for the sum of its correlated squared unit-dimensional distances. Then, the Gaussian kernel between the instances is obtained from the expected Gaussian kernel under the probability distribution for the squared distance between the instances.

Autoencoder is a neural network that reconstructs an input while representing the input in a lower-dimensional space from which features are obtained. The nature of the input from multi-step process signals is neglected by the autoencoder, and so the reconstructed input signals fluctuate although the original signals are not. Therefore, we propose autoencoder with a fusion regularization (Land and Friedman, 1996) on neighboring reconstructed signals aiming to extract features with smooth reconstruction. Also, we further consider autoencoder with clipping fusion regularization to alleviate indiscriminate penalization caused by the application of the fusion regularization on the transient changes between the signals of consecutive subprocesses.

For group-level sparsity, group lasso (Yuan and Lin, 2006) has been a widely adopted regularization. However, group lasso leads to the coincident selection of the feature groups that are group-level correlated and that share their predictability to a response. Aiming at exclusive sparsity at an inter-group level to select salient feature groups, we propose a new regularization, group-exclusive group lasso (GGL), that

penalizes the feature groups that are active simultaneously and are correlated to each other. We evaluate GGL in deep neural networks on a case study for automatic exclusive sensor selection in VM modeling.

1.2 Dissertation Outline

The rest of this dissertation is organized as follows. In Chapter 2, we present a new relevance vector machine for incomplete data. In Chapter 3, we propose a new method to estimate the expected Gaussian kernels of incomplete data with correlated variables. In Chapter 4, we present an autoencoder with a new regularization for multi-step process signals. In Chapter 5, we propose a new group-level regularization for automatic exclusive feature group selection in deep neural networks. Finally, in Chapter 6, we summarize the research results and describe future research.

CHAPTER 2

RESTRICTED RELEVANCE VECTOR MACHINE FOR INCOMPLETE DATA

2.1 Introduction

Process monitoring in semiconductor manufacturing is a key to achieving reliable wafer fabrication and high yield in production. Fabricating wafers is complex, and researchers have investigated diverse approaches to monitoring the fabricating processes and the quality of the wafer outputs (Martínez-Costa et al., 2014; Negahban and Smith, 2014). A typical approach to monitoring wafer fabrication is based on the physical metrology of wafer quality characteristics (He and Wang, 2007; Weiss et al., 2010). A few wafers in batches are randomly selected as a sample, and their quality characteristics are measured assuming that the characteristics of any given sample of all the wafers in production will be representative of the full batch (Hung et al., 2007). However, monitoring production processes based on such measurements can be limited because of the additional costs for measuring devices and human resources and delays in production (Kurz et al., 2014; Yiqi et al., 2013).

Virtual metrology (VM), considered soft metrology, predicts wafer quality characteristics based on data about in-process wafers from fabrication equipment sensors. In recent decades, VM models have been studied to overcome the limitations of the metrology-based approaches (Vallejo et al., 2019) and reduce manufacturing costs by enhancing or substituting metrology steps while improving the manufacturing quality by providing information about all wafers in production (Kang et al., 2016), thereby increasing manufacturing efficiency (Zeng and Spanos, 2009).

There is the possibility of missing VM data during wafer fabrication. Signals are collected from different sensors on the fabrication equipment as inputs in VM models to collect in-process wafer information, and sensor failure is a common cause of missing data (Kadlec et al., 2009). Difficulties can arise in detecting sensor signals because of aging, miscalibration, or drift without regular maintenance and quality testing (Park et al., 2003), and measurements from malfunctioning sensors are rejected or disregarded in model building.

In studies on semiconductor manufacturing, researchers have commonly used two approaches to building predictive models with incomplete data. Many investigators discarded incomplete instances, measurements that were missing or partially observed, and built their models using only complete, fully observed measurements (Chien et al., 2007; He and Wang, 2007; Hsu and Chien, 2007; Purwins et al., 2014). Other researchers considered imputation to make it possible to include incomplete instances as complete data, replacing the missing values with appropriate values based on available instances such as the mean (Hung et al., 2007; Wu et al., 2011; Zeng and Spanos, 2009).

Kernel-based machine learning methods have become popular in data analysis. Using the "kernel trick," complex nonlinear relationships in data can be represented in a high-dimensional space by such kernel-based models such as Gaussian processes (Rasmussen and Williams, 2006), support vector machines (Vapnik, 1998), and the relevance vector machine (RVM) (Tipping, 2001). RVM is a sparse Bayesian kernel machine that can be used for regression with high sparsity of a trained prediction model and the Bayesian property, and it has shown successful performance on predictive tasks in semiconductor manufacturing (Hwang and Jeong, 2018; Hwang et al., 2014) as well as

other complex manufacturing (Bastani et al, 2012; Caesarendra et al., 2010; Chang et al., 2017; Di Maio et al., 2012).

The presence of missing data hinders training an RVM model because the kernel between two instances cannot be calculated by a kernel function if any of the instances has missing components. For kernel methods, imputation can be commonly applied to include partially observed, incomplete instances. That is, to compute the kernels of incomplete data given a kernel function, the missing values are replaced in the original space to make the original data complete. However, as discussed in (Von Hippel, 2009), variable transformation after direct imputation in the original space, i.e. *impute-and-transform*, can lead to inaccurate estimation of coefficients in an analysis model, and therefore poor prediction performance in VM, whereas imputation in the feature space after transformation, i.e., *transform-and-impute*, does not. For the direct treatment of missing kernels, imputation can be considered in the kernel space (Belanche et al., 2014; Nebot-Troyano and Belanche-Muñoz, 2010), which can lead to losing the sparsity of the predictive function of a trained RVM model by selecting RVs from among all the instances, including incomplete ones, in a data set.

In this study, we propose a new RVM model for incomplete data, called a restricted relevance vector machine (RRVM). RRVM handles incomplete data, imputing the missing values in the kernel space to consider complex data nonlinearity. RRVM restricts model basis functions to complete instances and maintains the model sparsity incorporating incomplete instances. RRVM can explore nonlinearity by using any kind of kernel functions to fit data in the presence of missing values. We describe the proposed RRVM adopting a method for imputation in kernel space (Nebot-Troyano and Belanche-Muñoz,

2010), and we derive the predictive distribution for RRVM, presenting the marginal likelihood optimization for RRVM.

The remainder of this chapter is organized as follows. In Section 2.2, we review the related literature on missing treatment methods and the RVM for regression. In Section 2.3, we describe the proposed method in detail, and, in Section 2.4, we present experimental results from a toy example and a real-life case. Finally, we conclude this work and suggest future work in Section 2.5.

2.2 Related Work

2.2.1 Missing treatments

Incomplete or missing data has long been studied in the literature. A widely adapted approach for missing values is imputation, which entails filling in missing values with reasonable surrogate values induced from available given data. In another approach, some methods deal with missing values within algorithms in order to incorporate such instances into model fitting. In this section, we briefly review the literature on missing values.

One of the most basic imputation methods is single value imputation which replaces each missing piece of data with a certain value. Mean imputation uses the mean of the variable corresponding to an incomplete instance's missing coordinate, and median imputation uses the median of the variables as a variant of mean imputation. However, these processes have the limitation that they underestimate the variance of the variables (Little, 1992).

Hot-deck imputation methods (Andridge and Little, 2010; Sande, 1983) replace missing values with similar observed values in given data. However, hot-deck imputation may not be robust because it relies on a single nearest point(s). In response, advanced imputation methods have been suggested that take into account the multiple nearest neighbor points (Cotton, 1991), a set of neighboring points with weights (Kim and Fuller, 2004), and a set of neighboring points including partially observed instances (Van Hulse and Khoshgoftaar, 2014).

Multiple imputation methods, introduced in (Rubin, 1978), simultaneously produce multiple data sets in which the missing values are replaced with different values randomly sampled from the distribution that the data including the missing values are assumed to follow. For multiple imputation, some methods in the literature entail generating data sets from the distributions whose parameters are estimated by employing the expectation-maximization algorithm (Dempster et al., 1977; Schafer, 1997) and Markov chain Monte Carlo (Lin, 2010) and from iterative hot-deck imputation (Siddique and Belin, 2008).

However, other researchers have developed missing data imputation methods using model-based techniques. In particular for kernel methods, researchers have studied implicitly incorporating partial information in analysis models: modifying the objective functions of Gaussian process and SVM models in the probability estimation of exponential families (Smola et al., 2005) and minimizing the risk caused by incomplete instances in SVM modeling (Pelckmans et al., 2005).

For some kernel method models, researchers have handled missing data in original space with values inferred in a kernel space. For general kernel functions, the kernel extension (Belanche et al., 2014; Nebot-Troyano and Belanche-Muñoz, 2010) considers

imputation in the kernel space without estimating any missing values in the original space, which allows for treating missing values in testing data. However, the kernel extension overlooks the information from partially observed values of incomplete instances, and hence the kernel extension is extended to the extended heterogeneous kernels (EHKs) (Belanche et al., 2014; Nebot-Troyano and Belanche-Muñoz, 2010) to make full use of available values in incomplete data as an implementation of the kernel extension with empirical distributions. The EHK for inputs in a *d*-dimensional original space is estimated by *d* univariate kernel extensions, each of which exploits all the available values in the original space.

2.2.2 Relevance vector machine for regression

RVM, originally proposed in (Tipping, 2001), is a Bayesian kernel machine whose coefficient vector has an automatic relevance determination (ARD) prior distribution. RVM has been applied to various regression and classification tasks because it usually leads to sparser models than the other kernel methods such as support vector machines and Gaussian process regression that originated from the ARD prior (Tipping, 2001; Tipping and Faul, 2003) with comparable performance and computational complexities.

A kernel-based sparse model for predicting a target variable $y \in \mathbb{R}$ with input variable vector $\mathbf{x} \in \mathbb{R}^D$ can be given in the form of a general linear regression model as

$$y = \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}) + \varepsilon \tag{2.1}$$

where $\mathbf{w} = [w_1, w_2, \dots, w_M]^{\mathrm{T}}$ is an M-dimensional model coefficient vector, $\boldsymbol{\phi}(\mathbf{x}) = [\phi_1(\mathbf{x}), \dots, \phi_M(\mathbf{x})]^{\mathrm{T}}$ is a vector of M basis functions, and ε is a zero-mean Gaussian noise

with variance σ^2. The basis funcitons $\phi_j(\mathbf{x})$ can be defined with a kernel function with training instances such that $k(\mathbf{x}, \mathbf{x}_j)$ for $j = 1, \ldots, N$.

Given a data set of N instances, $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, the target values in $\mathbf{y} = [y_1, \ldots, y_N]^{\mathrm{T}}$ are predicted by a RVM model as

$$\mathbf{y} = \mathbf{\Phi w} + \boldsymbol{\epsilon} \tag{2.2}$$

where $\mathbf{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_M]$ is the design matrix whose j-th column is $\boldsymbol{\phi}_j = [\phi_j(\mathbf{x}_1), \ldots, \phi_j(\mathbf{x}_N)]^{\mathrm{T}}$ for $j = 1, \ldots, M$, and $\boldsymbol{\epsilon} = [\epsilon, \ldots, \epsilon_N]^{\mathrm{T}}$ is a vector of the errors assumed as probabilistically independent samples of ε. Accordingly, the multivariate Gaussian likelihood for the target vector $\mathbf{y}$ is given as $p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \sigma^2) \sim \mathcal{N}(\mathbf{y}|\hat{\mathbf{y}}, \sigma^2)$ where $\hat{\mathbf{y}} = \mathbf{\Phi w}$. The coefficient vector $\mathbf{w}$ has an ARD prior of $p(\mathbf{w}|\mathbf{A}) \sim \mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{A}^{-1}) = \prod_{j=1}^M \mathcal{N}(w_j|0, A_{jj}^{-1})$ where the precision matrix $\mathbf{A}$ is an M-by-M diagonal matrix whose j-th diagonal element is A_{jj}.

The hyperparameters for the prior distribution on $\mathbf{w}$ in $\mathbf{A}$ and σ^2 can be determined by type-II maximum likelihood estimation, and the likelihood is

$$\begin{aligned} p(\mathbf{y}|\mathbf{X}, \mathbf{A}, \sigma^2) &= \int p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \sigma^2) p(\mathbf{w}|\mathbf{A}) d\mathbf{w} \\ &\sim \mathcal{N}(\mathbf{0}, \mathbf{\Phi A}^{-1}\mathbf{\Phi}^{\mathrm{T}} + \sigma^2 \mathbf{I}) \end{aligned} \tag{2.3}$$

where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^{\mathrm{T}}$ is an input matrix and $\mathbf{x}_i$ is the i-th instance vector. Due to the ARD prior, most of the diagonal elements in $\mathbf{A}$ become infinite after the likelihood maximization and the corresponding coefficients collapse to zero. Therefore, the model in (2.1) can be represented by a small number of kernel functions that center at the input points corresponding to the nonzero coefficients, and these points are called relevance

vectors (RVs). Tipping (2001) explained the sparsity brought by the ARD prior using the vector alignment, and Wipf and Nagarajan (2008) reformulated the ARD prior and showed that the models related to the prior can be considered as a series of re-weighted ℓ_1 problems. More detailed description for the hyperparmaeter estimation of RVM and the sparsity can be found in (Tipping, 2001; Tipping and Faul, 2003; Wipf and Nagarajan, 2008). Then, the predictive distribution of RVM for a new test point $\mathbf{x}^*$ can be obtained as follows:

$$p(y^*|\mathbf{x}^*, \mathbf{y}, \mathbf{X}, \mathbf{A}, \sigma) \sim \mathcal{N}\left(\overline{f}(\mathbf{x}^*), \sigma^2 + \boldsymbol{\phi}(\mathbf{x}^*)^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{\phi}(\mathbf{x}^*)\right) \tag{2.4}$$

where

$$m(\mathbf{x}^*) = \sigma^{-2}\boldsymbol{\phi}(\mathbf{x}^*)^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{y} \tag{2.5}$$

$$\boldsymbol{\Sigma} = (\mathbf{A} + \sigma^{-2}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi})^{-1}. \tag{2.6}$$

Sparsity can also be found from the predictive distribution in (2.4). Because most diagonal entries of $\mathbf{A}$ are infinite, most elements in $\boldsymbol{\Sigma}$ become zero except those for which the row and column coincide with the finite diagonal entries of $\mathbf{A}$ (i.e., Σ_{ij} has nonzero values if and only if both A_{ii} and A_{jj} are finite). Therefore, $\boldsymbol{\Sigma}$ can be contracted to an r-by-r matrix where r is the number of RVs, and only the kernels between $\mathbf{x}^*$ and RVs are required to find the predictive distribution in (2.4).

2.3 Proposed Method

We first describe the restricted kernel matrix formed with basis functions only from complete instances. Then, we propose a novel variant of RVM, the restricted RVM to handle incomplete data.

2.3.1 Restricted kernel matrix

Given a data index set of N instances $\mathcal{X} = \{1, \dots, N\}$, let $\mathcal{X}_o \subset \mathcal{X}$ be an index set of L complete instances in $\mathcal{X}$ and let $\mathcal{X}_m = \mathcal{X} \backslash \mathcal{X}_o$ be an index set of $(N - L)$ incomplete instances that have missing values. Given a kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$ for a pair of two instance vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, it is insufficient to train a RVM model with the kernel matrix $\mathbf{\Phi}$ wherein $k(\mathbf{x}_i, \mathbf{x}_j)$ cannot be computed if either $\mathbf{x}_i$ or $\mathbf{x}_j$, or both is in $\mathcal{X}_m$. Therefore, all the elements in the kernel matrix, including the kernels from incomplete instances, needs to be determined.

Missing values in an incomplete instance can be inferred from available information in given data, and this can be written as $\hbar(\mathbf{x}_i, \mathbf{x}_j | \mathcal{X})$ where $\hbar$ is a function for computing $k(\mathbf{x}_i, \mathbf{x}_j)$ with either instance i or instance j or with both in $\mathcal{X}_m$. Following (Nebot-Troyano and Belanche-Muñoz, 2010), the complete kernel matrix $\tilde{\mathbf{\Phi}} = [\tilde{\boldsymbol{\phi}}_1, \tilde{\boldsymbol{\phi}}_2, \dots, \tilde{\boldsymbol{\phi}}_N]$ can be defined imputing the missing components where $\tilde{\boldsymbol{\phi}}_j = [\tilde{k}(\mathbf{x}_1, \mathbf{x}_j), \dots, \tilde{k}(\mathbf{x}_N, \mathbf{x}_j)]^{\mathrm{T}}$ for $j \in \mathcal{X}$ and $\tilde{k}(\mathbf{x}_i, \mathbf{x}_j)$ for instances $i, j \in \mathcal{X}$ is given by

$$\tilde{k}(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} k(\mathbf{x}_i, \mathbf{x}_j), & \text{if } i, j \in \mathcal{X}_o \\ \int p(\mathbf{x}_i) k(\mathbf{x}_i, \mathbf{x}_j) d\boldsymbol{x}_i, & \text{if } i \in \mathcal{X}_m \text{ and } j \in \mathcal{X}_o \\ \int p(\mathbf{x}_j) k(\mathbf{x}_i, \mathbf{x}_j) d\boldsymbol{x}_j, & \text{if } i \in \mathcal{X}_o \text{ and } j \in \mathcal{X}_m \\ \int\int p(\mathbf{x}_i) p(\mathbf{x}_j) k(\mathbf{x}_i, \mathbf{x}_j) d\mathbf{x}_i d\mathbf{x}_j, & \text{if } i, j \in \mathcal{X}_m \end{cases}. \quad (2.7)$$

To compute $\tilde{k}(\mathbf{x}_i, \mathbf{x}_j)$, we adopt the EHKs of (Belanche et al., 2014) as follows:

$$\tilde{k}_{EHK}(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} \frac{1}{D}\sum_{d=1}^{D} k_d(\mathbf{x}_i, \mathbf{x}_j), & \text{if } i, j \in \mathcal{X}_o \\ \frac{1}{D}\sum_{d=1}^{D} \hbar_{d,mo}(\mathbf{x}_i, \mathbf{x}_j | \mathcal{X}), & \text{if } i \in \mathcal{X}_m \text{ and } j \in \mathcal{X}_o \\ \frac{1}{D}\sum_{d=1}^{D} \hbar_{d,om}(\mathbf{x}_i, \mathbf{x}_j | \mathcal{X}), & \text{if } i \in \mathcal{X}_o \text{ and } j \in \mathcal{X}_m \\ \frac{1}{D}\sum_{d=1}^{D} \hbar_{d,mm}(\mathbf{x}_i, \mathbf{x}_j | \mathcal{X}), & \text{if } i, j \in \mathcal{X}_m \end{cases} \quad (2.8)$$

where k_d, $\hbar_{d,mo}$, $\hbar_{d,om}$, and $\hbar_{d,mm}$ are univariate kernel functions defined on d-th input features of both instances, respectively, as

$$k_d(\mathbf{x}_i, \mathbf{x}_j) = k(x_{id}, x_{jd})$$

$$\hbar_{d,mo}(\mathbf{x}_i, \mathbf{x}_j | \mathcal{X}) = \mathbb{I}\left(i \in \mathcal{X}_o^{(d)}\right) k_d(\boldsymbol{x}_i, \boldsymbol{x}_j) + \frac{1}{L_d}\mathbb{I}\left(i \in \mathcal{X}_m^{(d)}\right) \sum_{s \in \mathcal{X}_o^{(d)}} k_d(\boldsymbol{x}_s, \boldsymbol{x}_j)$$

$$\hbar_{d,om}(\mathbf{x}_i, \mathbf{x}_j | \mathcal{X}) = \mathbb{I}\left(j \in \mathcal{X}_o^{(d)}\right) k_d(\boldsymbol{x}_i, \boldsymbol{x}_j) + \frac{1}{L_d}\mathbb{I}\left(j \in \mathcal{X}_m^{(d)}\right) \sum_{r \in \mathcal{X}_o^{(d)}} k_d(\boldsymbol{x}_i, \boldsymbol{x}_r)$$

$$\begin{aligned}\hbar_{d,mm}(\mathbf{x}_i, \mathbf{x}_j|\mathcal{X}) = \mathbb{I}\left(i \in \mathcal{X}_o^{(d)}, j \in \mathcal{X}_o^{(d)}\right) k_d(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ + \frac{1}{L_d}\mathbb{I}\left(i \in \mathcal{X}_m^{(d)}, j \in \mathcal{X}_o^{(d)}\right) \sum_{s \in \mathcal{X}_o^{(d)}} k_d(\boldsymbol{x}_s, \boldsymbol{x}_j) \\ + \frac{1}{L_d}\mathbb{I}\left(i \in \mathcal{X}_o^{(d)}, j \in \mathcal{X}_m^{(d)}\right) \sum_{r \in \mathcal{X}_o^{(d)}} k_d(\boldsymbol{x}_i, \boldsymbol{x}_r) \\ + \frac{1}{L_d^2}\mathbb{I}\left(i \in \mathcal{X}_m^{(d)}, j \in \mathcal{X}_m^{(d)}\right) \sum_{s \in \mathcal{X}_o^{(d)}} \sum_{r \in \mathcal{X}_o^{(d)}} k_d(\boldsymbol{x}_s, \boldsymbol{x}_r),\end{aligned}$$

$\mathbb{I}$ is an indicator function, $\mathcal{X}_m^{(d)}$ and $\mathcal{X}_o^{(d)}$ are the sets of instances of which d-th input features are missing and observed, respectively, and L_d is the number of instances that belong to $\mathcal{X}_o^{(d)}$.

However, direct application of the imputed kernel matrix $\widetilde{\boldsymbol{\Phi}}$ to an RVM model may cause issues. First, the basis functions can contain uncertainties from incomplete instances. By employing the entire matrix $\widetilde{\boldsymbol{\Phi}}$ for a RVM model, a RV can be chosen from the instance $\mathbf{x}_{RV} \in \mathcal{X}_m$, and accordingly, the model is built on this basis function whose values are artificially determined with uncertainties. If the basis function has uncertainties from missing imputation, it can cause imprecise coefficient estimation in regression such that the measurement errors in independent variables causes bias in regression coefficients in error-in-variables regression.

Furthermore, the RVM model with $\widetilde{\Phi}$ may lose its sparsity properties if a RV is selected that has missing values. Predicting a new testing instance $\mathbf{x}^*$ requires computing kernel values between $\mathbf{x}^*$ and $\mathbf{x}_{RV}$ in order to find the predictive distribution in (2.4). If $\mathbf{x}_{RV} \in \mathcal{X}_m$, the kernel $\tilde{k}(\mathbf{x}^*, \mathbf{x}_{RV})$ implicitly requires calculating the kernel between $\mathbf{x}^*$ and

the instances associated with $k(\mathbf{x}^*, \mathbf{x}_{RV}|\mathcal{X}_o)$ as in (2.7) or (2.8). Analogously, if $\mathbf{x}^*$ is missing, the predictive distribution requires such computation for the instances associated with $\hbar(\mathbf{x}^*, \mathbf{x}_{RV}|\mathcal{X}_o)$.

To avoid building a RVM model with RVs chosen from incomplete instances, we construct an N-by-L matrix $\widetilde{\boldsymbol{\Phi}}_R = \left[\widetilde{\boldsymbol{\phi}}_{R,1}, \ldots, \widetilde{\boldsymbol{\phi}}_{R,L}\right]$ whose basis functions are restricted to the functions of the complete instances as $\widetilde{\boldsymbol{\phi}}_{R,l} = \left[\tilde{k}_R(\boldsymbol{x}_1, \boldsymbol{x}_l), \ldots, \tilde{k}_R(\boldsymbol{x}_N, \boldsymbol{x}_l)\right]^T$ for $l \in \mathcal{X}_o$. Then, the kernel in (2.7) and (2.8) restricting the basis can be represented, respectively, as

$$\tilde{k}_R(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} k(\mathbf{x}_i, \mathbf{x}_j), & \text{if } i, j \in \mathcal{X}_o \\ \int p(\mathbf{x}_i) k(\mathbf{x}_i, \mathbf{x}_j) d\boldsymbol{x}_i, & \text{if } i \in \mathcal{X}_m \text{ and } j \in \mathcal{X}_o \end{cases} \tag{2.9}$$

and

$$\tilde{k}_{EHK,R}(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} \dfrac{1}{D}\displaystyle\sum_{d=1}^{D} k_d\left(\mathbf{x}_i^{(d)}, \mathbf{x}_j^{(d)}\right), & \text{if } i, j \in \mathcal{X}_o \\ \dfrac{1}{D}\displaystyle\sum_{d=1}^{D} \hbar_{d,mo}\left(\mathbf{x}_i, \mathbf{x}_j \middle| \mathcal{X}\right), & \text{if } i \in \mathcal{X}_m \text{ and } j \in \mathcal{X}_o \end{cases} . \tag{2.10}$$

2.3.2 Restricted relevance vector machine for regression with missing data

We here propose the RRVM to construct a sparse Bayesian kernel regression with missing data, restricting relevance vectors to fully observed points.

Given N pairs of input vectors and target values $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, suppose inputs vectors of L instances are complete ($0 < L < N$) and the target values of all N instances

are available. With the restricted kernel matrix $\widetilde{\mathbf{\Phi}}_R$ from (2.9) or (2.10), the proposed RRVM regression model is defined as

$$\mathbf{y} = \widetilde{\mathbf{\Phi}}_R \mathbf{w}_R + \boldsymbol{\varepsilon} \tag{2.11}$$

where $\mathbf{w}_R$ is an L-dimensional coefficient vector, $\boldsymbol{\varepsilon}$ is an N-dimensional noise vector that follows $\mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_N)$, and $\mathbf{I}_N$ is an N-by-N identity matrix. The basis functions of the RRVM model are restricted to the kernel functions centered at the fully observed instances. The proposed RRVM also employs an ARD prior over $\mathbf{w}_R$ in order to obtain a sparse solution as the regular RVM model in Section 2.2.2. That is, the coefficient vector $\mathbf{w}_R$ has the ARD prior as $p(\mathbf{w}_R|\mathbf{A}_R) \sim \mathcal{N}(\mathbf{0}, \mathbf{A}_R^{-1}) = \prod_{j=1}^{L} \mathcal{N}\left(w_j \middle| 0, A_{jj}^{-1}\right)$ where the precision matrix $\mathbf{A}_R$ is an L-by-L diagonal matrix with diagonal vector $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_L]^T$.

Given the ARD prior, the posterior distribution over the coefficient vector $\mathbf{w}_R$ is Gaussian:

$$p(\mathbf{w}_R|\mathbf{y}, \mathbf{X}, \mathbf{A}_R, \sigma^2) \sim \mathcal{N}(\mathbf{m}_R, \mathbf{\Sigma}_R) \tag{2.12}$$

where $\mathbf{m}_R = \sigma^{-2} \mathbf{\Sigma}_R \widetilde{\mathbf{\Phi}}_R^{\mathrm{T}} \mathbf{y}$ and $\mathbf{\Sigma}_R = \left(\mathbf{A}_R + \sigma^{-2} \widetilde{\mathbf{\Phi}}_R^{\mathrm{T}} \widetilde{\mathbf{\Phi}}_R\right)^{-1}$. The hyperparameters of the proposed model (i.e., $\mathbf{A}_R$ and σ^2) can be found by maximum likelihood estimation where the marginal likelihood is

$$\begin{aligned} p(\mathbf{y}|\mathbf{X}, \mathbf{A}_R, \sigma^2) &= \int p(\mathbf{y}|\mathbf{X}, \mathbf{w}_R, \sigma^2) p(\mathbf{w}_R|\mathbf{A}_R) d\mathbf{w}_R \\ &\sim \mathcal{N}\left(\mathbf{0}, \widetilde{\mathbf{\Phi}}_R \mathbf{A}_R^{-1} \widetilde{\mathbf{\Phi}}_R^{\mathrm{T}} + \sigma^2 \mathbf{I}_N\right). \end{aligned} \tag{2.13}$$

The optimization can be conducted iteratively as in (Tipping and Faul, 2003) and the detailed procedure is as follows. First, we need to find update formula for iterative

process. The covariance matrix in (2.13), $\boldsymbol{\Lambda} = \widetilde{\boldsymbol{\Phi}}_R \mathbf{A}_R^{-1} \widetilde{\boldsymbol{\Phi}}_R^{\mathrm{T}} + \sigma^2 \mathbf{I}_N$, can be decomposed as follows:

$$\begin{aligned} \boldsymbol{\Lambda} &= \alpha_i^{-1} \widetilde{\boldsymbol{\phi}}_i \widetilde{\boldsymbol{\phi}}_i^{\mathrm{T}} + \sum_{l \neq i} \alpha_l^{-1} \widetilde{\boldsymbol{\phi}}_l \widetilde{\boldsymbol{\phi}}_l^{\mathrm{T}} + \sigma^2 \mathbf{I}_N \\ &= \alpha_i^{-1} \widetilde{\boldsymbol{\phi}}_i \widetilde{\boldsymbol{\phi}}_i^{\mathrm{T}} + \boldsymbol{\Lambda}_{-i} \end{aligned} \tag{2.14}$$

where $\boldsymbol{\Lambda}_{-i}$ contains the sum of the contribution of the basis vectors l from the complete instances for $l = 1, \ldots, L$ and $l \neq i$. The determinant and inverse of $\boldsymbol{\Lambda}$ are given, respectively, by

$$|\boldsymbol{\Lambda}| = |\boldsymbol{\Lambda}_{-i}| \cdot \left|1 + \alpha_i^{-1} \widetilde{\boldsymbol{\phi}}_i^{\mathrm{T}} \boldsymbol{\Lambda}_{-i}^{-1} \widetilde{\boldsymbol{\phi}}_i\right| \tag{2.15}$$

and

$$\boldsymbol{\Lambda}^{-1} = \boldsymbol{\Lambda}_{-i}^{-1} - \frac{\boldsymbol{\Lambda}_{-\mathrm{i}}^{-1} \widetilde{\boldsymbol{\phi}}_i \widetilde{\boldsymbol{\phi}}_{\mathrm{i}}^{\mathrm{T}} \boldsymbol{\Lambda}_{-\mathrm{i}}^{-1}}{\alpha_i + \widetilde{\boldsymbol{\phi}}_i^{\mathrm{T}} \boldsymbol{\Lambda}_{-i}^{-1} \widetilde{\boldsymbol{\phi}}_i}. \tag{2.16}$$

Then, letting $\boldsymbol{\alpha}_{-i}$ be the $(L-1)$-dimensional vector that is $\boldsymbol{\alpha}$ whose i-th component is excluded, the log marginal likelihood $\mathcal{L}(\boldsymbol{\alpha}) = \log p(\mathbf{y}|\mathbf{X}, \mathbf{A}_R, \sigma^2)$ is given by

$$\mathcal{L}(\boldsymbol{\alpha}) = \mathcal{L}(\boldsymbol{\alpha}_{-i}) + \ell(\alpha_i) \tag{2.17}$$

where

$$\mathcal{L}(\boldsymbol{\alpha}_{-i}) = -\frac{1}{2}[N \log(2\pi) + \log|\boldsymbol{\Lambda}_{-i}| + \mathbf{y}^{\mathrm{T}} \boldsymbol{\Lambda}_{-i}^{-1} \mathbf{y}] \tag{2.18}$$

and

$$\ell(\alpha_i) = \frac{1}{2}\left[\log \alpha_i - \log(\alpha_i + s_i) + \frac{q_i^2}{\alpha_i + s_i}\right] \tag{2.19}$$

with the sparsity factor of the i-th instance, $s_i = \widetilde{\boldsymbol{\phi}}_i^{\mathrm{T}} \boldsymbol{\Lambda}_{-i}^{-1} \widetilde{\boldsymbol{\phi}}_i$, that shows the extent to which the corresponding basis vector $\widetilde{\boldsymbol{\phi}}_i$ overlaps the basis vectors already involved in the model, and the quality factor, $q_i = \widetilde{\boldsymbol{\phi}}_i^{\mathrm{T}} \boldsymbol{\Lambda}_{-i}^{-1} \mathbf{y}$, that shows the extent of the alignment of $\widetilde{\boldsymbol{\phi}}_i$ with the error in the model without that basis.

Because the log marginal likelihood in (2.17) as the objective function of the optimization is decomposed into the part that is related to α_i and those that are not as in (2.17), α_i has an unique maximal solution if all other variables are fixed as follows:

$$\alpha_i = \begin{cases} \dfrac{s_i^2}{q_i^2 - s_i}, & \text{if } \; q_i^2 > s_i \\ \infty, & \text{otherwise} \end{cases} \tag{2.20}$$

by solving

$$\frac{\partial \ell(\alpha_i)}{\alpha_i} = \frac{1}{\alpha_i} - \frac{1}{\alpha_i + s_i} - \frac{q_i^2}{(\alpha_i + s_i)^2} = 0.$$

Hence, the point $\mathbf{x}_i \in \mathcal{X}_o$ is included in the set of RVs, $\mathcal{S}_{RV}$, if $q_i^2 > s_i$. If σ^2 is not given as a fixed value, it can also be updated as $\sigma^2 = \frac{\|\mathbf{y} - \hat{\mathbf{y}}\|^2}{N - L + \sum_{l=1}^{L}(\alpha_l \sigma^2 \Sigma_{R,ll})}$ (Tipping, 2001) where $\hat{\mathbf{y}} = \widetilde{\boldsymbol{\Phi}}_R \mathbf{m}_R$ is the predictive mean of the training points and $\Sigma_{R,ll}$ is the l-th diagonal element of $\boldsymbol{\Sigma}_R$.

Following the algorithm in (Tipping and Faul, 2003), the optimization above is performed as shown in Algorithm 2.1. This algorithm has the time complexity $\mathcal{O}(\eta N M^2)$ (Tipping and Faul, 2003; Son and Lee 2016) where M is the maximum number of basis

Algorithm 2.1 Marginal Likelihood Optimization for RRVM

Given $\mathbf{X} \in \mathbb{R}^{N\times D}$, $\mathbf{y} \in \mathbb{R}^{N}$, $k(\cdot)$, and $\mathcal{S}_{RV} = \emptyset$

1: Initialize σ^2 with an appropriate value.

2: Choose a single basis vector $\widetilde{\boldsymbol{\phi}}_r = \left[\tilde{k}_R(\mathbf{x}_1, \mathbf{x}_r), \ldots, \tilde{k}_R(\mathbf{x}_N, \mathbf{x}_r)\right]^{\mathrm{T}}$ where $r \in \mathcal{X}_o$, and include r in $\mathcal{S}_{RV}$, setting

- $\alpha_r \leftarrow \frac{\|\widetilde{\boldsymbol{\phi}}_r\|_2^2}{\|\widetilde{\boldsymbol{\phi}}_r^{\mathrm{T}}\mathbf{y}\|_2^2/\|\widetilde{\boldsymbol{\phi}}_r\|_2^2 - \sigma^2}$
- $\alpha_i \leftarrow \infty$ for all $i \in \mathcal{X}_o$ and $i \notin \mathcal{S}_{RV}$.

3: Calculate $\mathbf{m}_R = \sigma^{-2}\boldsymbol{\Sigma}_R\widetilde{\boldsymbol{\Phi}}_R^{\mathrm{T}}\mathbf{y}$ and $\boldsymbol{\Sigma}_R = \left(\mathbf{A}_R + \sigma^{-2}\widetilde{\boldsymbol{\Phi}}_R^{\mathrm{T}}\widetilde{\boldsymbol{\Phi}}_R\right)^{-1}$ with given initial values.

4: **Repeat**

5: Select a candidate basis vector $\widetilde{\boldsymbol{\phi}}_i$ for $i \in \mathcal{X}_o$.

6: Compute q_i and s_i.

7: **if** $q_i^2 - s_i > 0$ and $\alpha < \infty$ (i.e., already i in $\mathcal{S}_{RV}$),

8: Re-estimate α_i

9: **else if** $q_i^2 - s_i > 0$ and $\alpha = \infty$,

10: Add i to the set $\mathcal{S}_{RV}$ and compute α_i

11: **else if** $q_i^2 - s_i \leq 0$ and $\alpha_i < \infty$,

12: Remove i from $\mathcal{S}_{RV}$ and set $\alpha_i = \infty$

13: **end**

14: Update $\sigma^2 \leftarrow \frac{\|\mathbf{y}-\hat{\mathbf{y}}\|^2}{N-L+\sum_{l=1}^{L}(\alpha_l\sigma^2\Sigma_{R,ll})}$.

15: Recalculate $\mathbf{m}_R$, $\boldsymbol{\Sigma}_R$.

16: **Until** converged.

vectors included during the algorithm and η is the number of iterations until convergence.

Finally, the posterior predictive distribution of the RRVM for a new instance $\mathbf{x}^*$ can be derived using the Sherman-Morrison-Woodbury matrix identity as follows:

$$p(y^*|\mathbf{x}^*, \mathbf{y}, \mathbf{X}, \mathbf{A}_R, \sigma^2) = \int p(y^*|\mathbf{x}^*, \mathbf{w}_R, \sigma^2) p(\mathbf{w}_R|\mathbf{y}, \mathbf{X}, \mathbf{A}_R, \sigma^2) d\mathbf{w}_R$$

$$\sim \mathcal{N}\left(m_{new}(\mathbf{x}^*), \sigma^2_{new}(\mathbf{x}^*)\right) \tag{2.21}$$

where

$$m_{new}(\mathbf{x}^*) = \mathbf{m}_R^{\mathrm{T}} \widetilde{\boldsymbol{\phi}}(\mathbf{x}^*) \tag{2.22}$$

$$\sigma^2_{new}(\mathbf{x}^*) = \widetilde{\boldsymbol{\phi}}(\mathbf{x}^*)^{\mathrm{T}} \boldsymbol{\Sigma}_R \widetilde{\boldsymbol{\phi}}(\mathbf{x}^*) + \sigma^2. \tag{2.23}$$

The detailed derivation of the predictive distribution in (2.19) can be found in Appendix A. The proposed RRVM has all advantages of the kernel extension noted in Section 2.2.2, maintaining the sparsity of the solution in the predictive posterior distribution.

Similar to the original RVM, a sparse solution of the proposed RRVM can be obtained by employing the ARD prior distribution for the regression coefficient vector $\mathbf{w}_R$. Most of the diagonal entries in $\mathbf{A}_R$ go to infinity after maximizing the likelihood in (2.17) with respect to the hyperparameters $\mathbf{A}_R$ and σ^2 so that only a small number of the coefficients have nonzero values. While the original RVM takes all the instances of nonzero regression coefficients as the RVs, the RRVM selects the instances of nonzero coefficients of $\mathbf{w}_R$ only from the fully observed instances wherein the basis vector consists only of the kernel functions centered at the fully observed instances.

Analogously, most of the elements in $\boldsymbol{\Sigma}_R$ become zero except for those for which row and column together coincide with the finite diagonal entries of $\mathbf{A}_R$. In particular, the

ij-th entry of $\mathbf{\Sigma}_R$ is nonzero if and only if both α_i and α_j are finite. Hence, $\mathbf{\Sigma}_R$ can be contracted to $|\mathcal{S}_{RV}|$-by-$|\mathcal{S}_{RV}|$ matrix where $|\mathcal{S}_{RV}|$ is the number of RVs in the set, $\mathcal{S}_{RV}$, after the hyperparameter estimation. The kernels between a new point $\mathbf{x}^*$ and RVs only from fully observed instances are required to find the predictive distribution in (2.19). To be specific, in the case of EHK, the trained RRVM requires calculating $D|\mathcal{S}_{RV}|$ kernel functions if $\mathbf{x}^*$ that has no missing and $(D-u+uL)|\mathcal{S}_{RV}|$ kernel functions if $\mathbf{x}^*$ has missing data in u dimensions ($u < D$). Meanwhile, the RVM with the EHK, if one of its RVs has missing values in v dimensions ($v < D$), requires computing the $D(|\mathcal{S}_{RV}| - v + vL)$ and $(D-u+uL)(|\mathcal{S}_{RV}| - v + vL)$ functions, respectively, for complete and incomplete $\mathbf{x}^*$. Therefore, the predictive distribution of the proposed RRVM also implies a sparser solution than that of the conventional RVM.

2.4 Experiments

We evaluate the performance of RRVM-EHK on a toy data set and a real-world data set in this section.

2.4.1 Toy data

In this experiment, we generated artificial data consisting of two-dimensional input and output variables. The input data were from the equation $[X_1, X_2]^{\mathrm{T}} = [R\cos\theta\,,\ R\sin\theta]^{\mathrm{T}}$ where $R = 4 + 0.5U$, $\theta = \pi V$, and $U, V \in [0,1]$. We generated 100 instances of U and V sampled from a uniform distribution with the range between 0 and 1 as a training set and 101 instances by sampling U from a uniform distribution with the range between 0 and 1

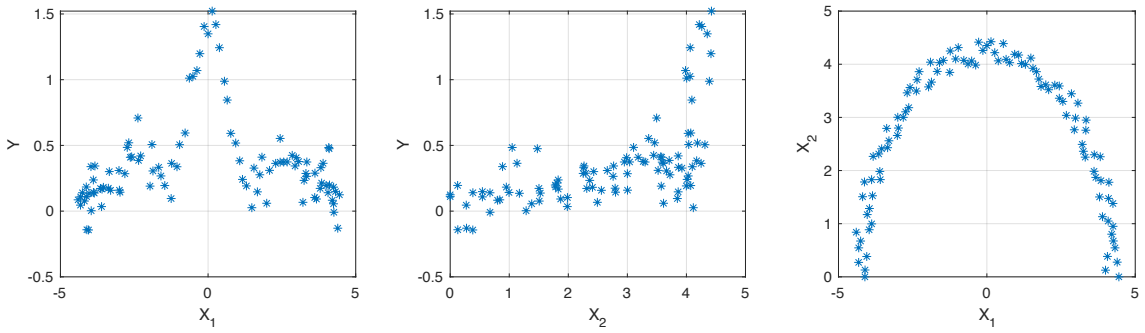


Figure 2.1 Toy data with a noise variance of 0.1.

and V linearly spaced in the same range as a testing set. The output data were from $Y = \text{sinc}(X_1) + 0.1X_2 + \varepsilon$ where $\text{sinc}(x) = \sin(x)/x$ and $\varepsilon \sim \mathcal{N}(0, \sigma_{noise}^2)$ with a noise variance σ_{noise}^2. Figure 2.1 illustrates the data generated with a noise variance $\sigma_{noise}^2 = 0.01$. We randomly selected instances in the training set and generated missing values for either X_1 or X_2 of the instances.

We evaluated the performance of RRVM with EHK for a Gaussian kernel (*RRVM-EHK*), comparing with that of RVM with EHK for the same kernel (*RVM-EHK*). We measured the performance of the models by computing each model's root mean squared error, $RMSE = \sqrt{\sum_{i=1}^{N_{te}} e_i^2 / N_{te}}$ and maximum absolute error (MAXAE) as $MAXAE = \max\{|e_1|, |e_2|, \dots, |e_{N_{te}}|\}$ where $e_i = y_i - \hat{y}_i$ is the prediction error of the i-th instance in the testing set with the predicted value $\hat{y}_i$ and N_{te} is the number of testing instances.

Figures 2.2 and 2.3 show the averaged difference between the testing performances of *RVM-EHK* and *RRVM-EHK* in 20 replications under different levels of uncertainty (i.e.,

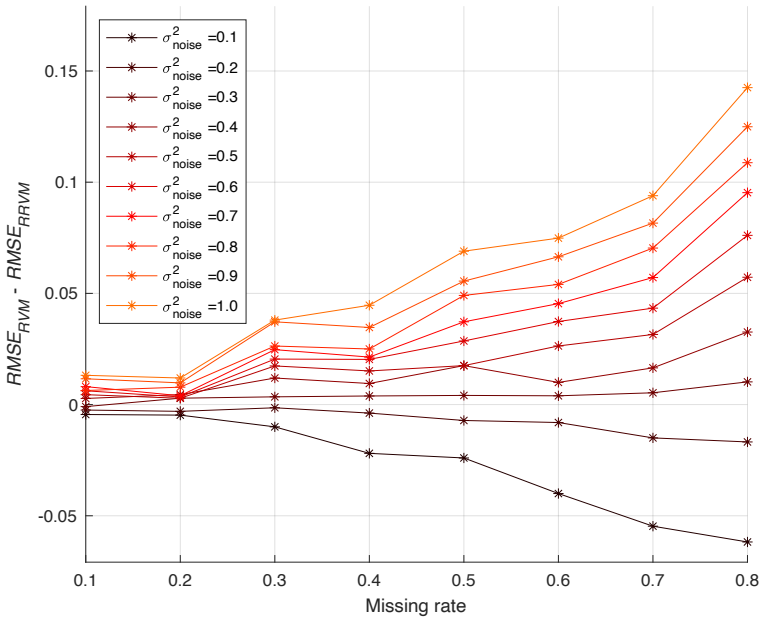


Figure 2.2 RMSE difference between *RVM-EHK* and *RRVM-EHK* in toy data analysis.

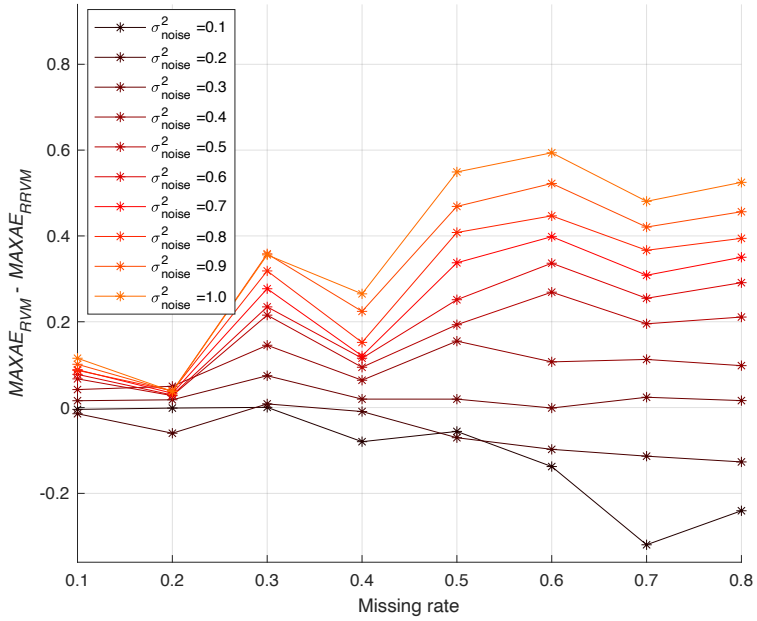


Figure 2.3 MAXAE difference between *RVM-EHK* and *RRVM-EHK* in toy data analysis.

σ_{noise}^2) and missing rates given a fixed Gaussian kernel parameter $\tau = 0.01$, and Figures 2.4 and 2.5 present the numbers of RVs chosen by *RVM-EHK* and *RRVM-EHK*, respectively. As the uncertainty level in data (i.e., σ_{noise}^2) increased, *RRVM-EHK* improved the prediction accuracy from *RVM-EHK*, and such improvement was amplified as the missing rate increased as shown in Figures 2.2 and 2.3. Besides, the number of RVs of *RVM-EHK* increased for higher missing rate in Figure 2.4, and the proportion of the RVs from incomplete instances increased as the missing rate increased in Table 2.1. In the other words, the directly application of imputation in feature space to RVM led to losing the sparsity of a RVM model wherein more RVs were chosen from highly missing data. Meanwhile, the number of RVs of *RRVM-EHK* maintained lower than the *RVM-EHK* as sparser models, which led to the improved prediction performance by avoiding overfitting.

2.4.2 Case study

We now evaluate the proposed method for VM modeling in plasma etching for semiconductor wafer fabrication. For the etching process, surface films layered on wafers are removed by plasmas in a chamber as illustrated in Figure 2.6. The process is monitored by various sensors in/on the chamber that measure different physical parameters such as gas supply pressure, temperature, and chemical levels. After the completion of the process, critical dimensions (CDs) of wafer quality characteristics are recorded as the output of the VM model.

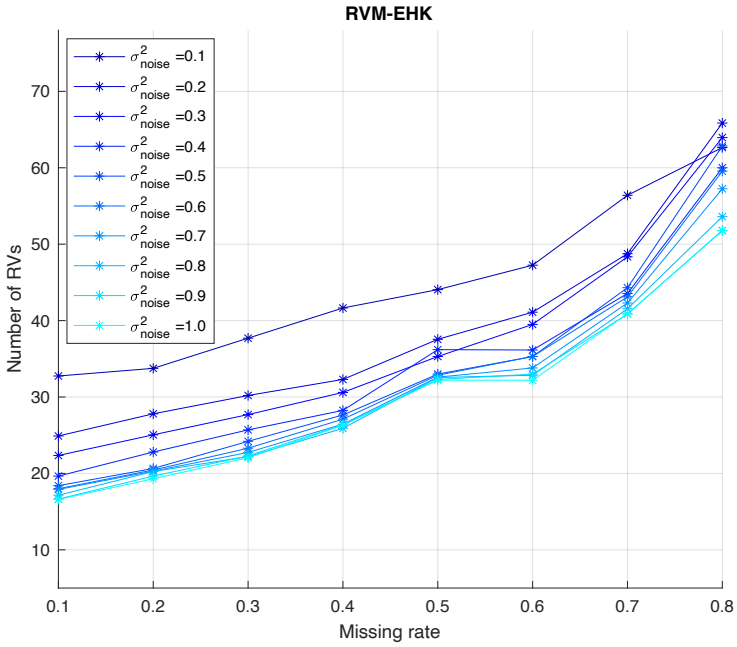


Figure 2.4 Number of selected RVs of *RVM-EHK* in toy data analysis.

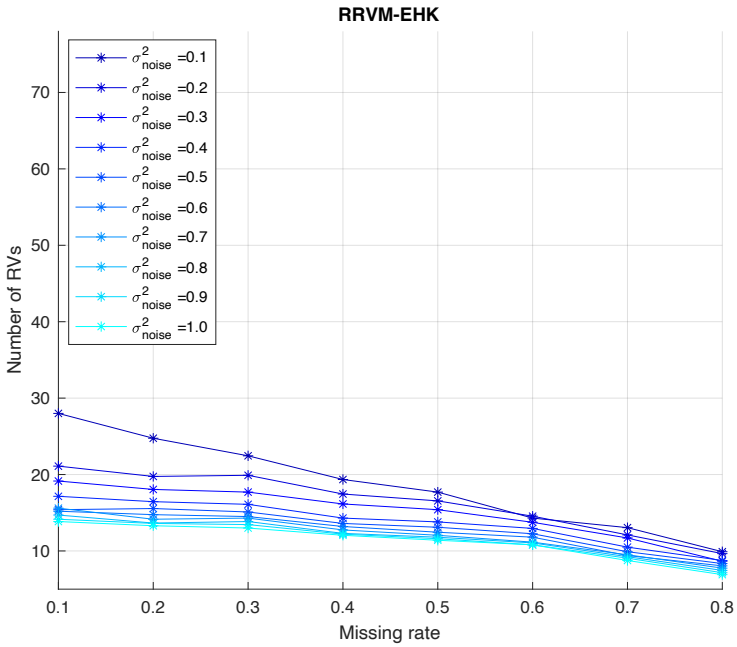


Figure 2.5 Number of selected RVs of *RRVM-EHK* in toy data analysis.

Table 2.1 Ratio of RVs of *RVM-EHK* selected from the incomplete instances in toy data analysis.

Missing rates	σ^2_{noise} 0.10	0.20	0.50	1.00
0.1	0.147	0.164	0.178	0.201
0.2	0.247	0.331	0.364	0.334
0.3	0.373	0.419	0.432	0.429
0.4	0.465	0.532	0.528	0.510
0.5	0.552	0.595	0.617	0.585
0.6	0.641	0.704	0.702	0.697
0.7	0.736	0.768	0.763	0.764
0.8	0.840	0.834	0.839	0.853

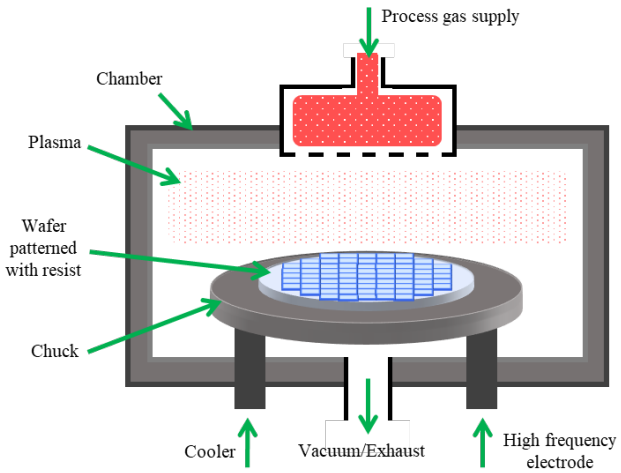


Figure 2.6 Illustration of a plasma etching process for wafer fabrication and its equipment.

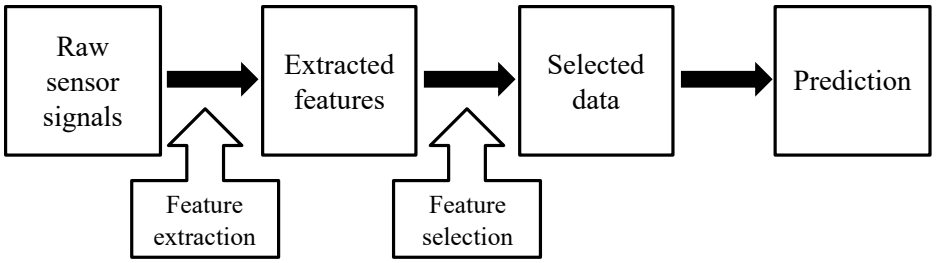


Figure 2.7 A general procedure of VM modeling using raw process signals.

The data set we employed in this experiment consisted of 299 instances from the etching process. During the production of identical wafer products, lots of 25 wafers were randomly sampled, and instances were collected from the wafers randomly selected from the wafers in the sampled lots. Each wafer instance consisted of the signals from 85 sensors[1] and a CD of the wafer. Wafer i for i = 1, …, 299 was processed during the processing time T_i, and the signals from sensor k, $\mathbf{x}_{ik}$, for k = 1, …, 85 were stored.

To build a VM model, the collected data were preprocessed as illustrated in Figure 2.7. The features for VM were obtained by computing the process statistics from each signal shown in Table 2.2; then, 10 process features were selected based on importance scores from the conditional permutation for random forest (Strobl et al., 2008). A summary of the data is given in Table 2.3. In the data set, CD measurements of all the sampled wafers were available, and the features of 120 wafers were fully available; there were missing values from the features of the remaining wafers.

1 Among more than 200 sensors, 85 were initially chosen by fab engineers screening out irrelevant sensors.

Table 2.2 Statistics extracted from the signal from sensor k for the process of wafer i.

Statistic	Expression
Length	T_i
Max	$\max(x_{ik1}, x_{ik2}, \dots, x_{ikT_i})$
Min	$\min(x_{ik1}, x_{ik2}, \dots, x_{ikT_i})$
Mean	$\bar{x}_{ik} = \frac{1}{T_i}\sum_{j=1}^{T_i} x_{ikj}$
Variance	$s_k^2 = \frac{1}{T_i}\sum_{j=1}^{T_i}(x_{ikj} - \bar{x}_{ik})^2$
Skewness	$\sum_{j=1}^{T_i}\left(\frac{x_{ikj}-\bar{x}_{ik}}{s_k}\right)^3$
Kurtosis	$\sum_{j=1}^{T_i}\left(\frac{x_{ikj}-\bar{x}_{ik}}{s_k}\right)^4$

Table 2.3 A summary of selected process features.

Sensor index	Statistics	Missing rate (%)
3	Min	12.04
10	Max	13.38
21	Mean	0.00
22	Min	0.00
35	Mean	12.37
35	Skewness	12.37
66	Max	0.00
68	Max	7.02
68	Skewness	7.02
72	Variance	29.48

We evaluated the proposed RRVM (*RRVM-EHK*), comparing with RVM with the incomplete data imputed with the unconditional mean of each dimension (*RVM-SMI*), RVM with the incomplete data imputed with the conditional mean from a multivariate normal distribution (*RVM-CMI*), and RVM with the incomplete data and EHK (*RVM-EHK*). The performance of each model was measured from a ten-fold cross-validation. We employed all the available complete instances and incomplete instances that were randomly sampled from the given data set as follows. We divided both the complete and incomplete data into ten individual data sets. We trained a prediction model with a training set that consisted of nine complete and nine incomplete sets, and we measured its performance using the rest of the complete set. We employed the Gaussian kernel, determining the parameter τ for each predictive model from a five-fold cross-validation over the training data set.

Table 2.4 presents the testing performance results with the pairwise t-test for the difference between the testing results. Using EHK for imputation in kernel space (*RVM-EHK* and *RRVM-EHK*) outperformed the imputation in original space (*RVM-SMI* and *RVM-CMI*). *RRVM-EHK* showed the competitive performance with higher sparsity over that of RVM-EHK where RVM-EHK selected more RVs from the data including incomplete instances as seen in Table 2.4.

2.5 Conclusion

We proposed a novel sparse Bayesian kernel model, the restricted relevance vector machine, for incorporating incomplete data into virtual metrology in semiconductor manufacturing. RRVM restricts the basis functions to be chosen only from complete

Table 2.4 Testing accuracy, the number of RVs, and the ratio of RVs from the incomplete instances in VM data analysis.

	RMSE	MAXAE	Number of RVs	Ratio†
RVM-SMI	0.808*	1.760***	86.4***	0.48
RVM-CMI	0.807*	1.666**	89.1***	0.43
RVM-EHK	0.744*	1.388	40.7***	0.48
RRVM-EHK	**0.693**	**1.287**	**25.7**	**-**

† Ratio of RVs from incomplete instances

* p-value < 0.10; ** p-value < 0.05; *** p-value < 0.01

instances in order to prevent the potential loss of the sparsity in the predictive RVM distribution. We described the proposed method with EHK, and we evaluated RRVM with both a toy and a real-life data set for VM in the etching process for wafer fabrication. The experimental results showed that the predictive performance improved when we included incomplete instances whose missing values were imputed by EHK in the kernel space, and RRVM-EHK achieved this improvement while maintaining model sparsity.

The proposed method can be extended to problems with different types of data in semiconductor manufacturing processes. The information from incomplete data can be incorporated considering special data properties such as spatial or time attributes of sampled instances. Additionally, it would be worth studying classification problems with missing data in semiconductor manufacturing such as faulty detection.

CHAPTER 3

GAUSSIAN KERNEL WITH CORRELATED VARIABLES FOR INCOMPLETE DATA

3.1 Introduction

Kernel method is a popular approach for machine learning tasks. Mapping data into a high-dimensional feature space, kernels facilitate a model to represent complex nonlinearity in the original space as linear patterns in feature space (Schölkopf et al., 2002; Shawe-Taylor and Cristianini, 2004). The mapping is performed with a kernel function that computes inner products of pairs of instance vectors, and diverse types of kernels are considered in the literature such as the linear kernel, the polynomial kernel and the radial basis function kernel where the selection of the kernel should be based on the nature of problems.

The presence of incomplete instances that have missing components, however, hinders training a kernel method model as the kernel between two instances cannot be calculated through a kernel function if any of the instances have missing components. There are diverse approaches to handling incomplete data with kernel methods.

For kernel methods, incomplete data can be treated implicitly in analysis models. Some methods in the literature, for example, are proposed such as the kernel partial least square for both the estimation of missing values and the classification thereof (Nguyen and Tsoy, 2017), the support vector machine minimizing the risk caused by incomplete instances (Pelckmans et al., 2005), and the support vector machines and Gaussian process with the estimation in feature space (Smola et al., 2005). However, these are model-specific methods that are limited to classification tasks and the analysis models employed.

As a simple approach, incomplete data for kernel methods can be handled based on missing imputation directly in the original space. As a preprocessing step to make incomplete data as complete (Jurado et al., 2017), one may employ imputation techniques to replace missing components with appropriate values inferred from observed components in the data, and then kernels are estimated with complete instances the data whose missing components are imputed. Imputation in the original space, however, leads to inaccurate estimation of analysis model coefficients when the missing imputation is performed for nonlinear terms from the transformation of input variables before the variable transformation (Von Hippel, 2009).

In recent years, there have been studies that address the estimation of kernels for incomplete data estimating transformed variables, in particular, for the Gaussian kernel that is one of the most popular choices because of its advantages (Bae and Park, 2019; Khellat-Kihel et al., 2016; Kim et al., 2018; Zhong et al., 2018). Instead of estimating the missing components in the original space, the expected squared distance with incomplete instances is considered to estimate the Gaussian kernel (Eirola et al., 2013) and extend their work using a Gaussian mixture distribution (Eirola et al., 2014).

A recent study in (Mesquita et al., 2019) proposes a method to estimate the Gaussian kernel with incomplete data as the expected Gaussian kernel (EGK). EGK estimates the Gaussian kernel of incomplete instances by computing the expectation of the Gaussian function under the probability distribution of the squared Euclidean distance between the instances. Following the assumption in (Mesquita et al., 2017), the squared distance between two instances in EGK is modeled as a random variable from the sum of the squared unit-dimensional distances between the instances, and the squared distance

between the instances is approximated by a Gamma random variable from the sum of independent Gamma random variables. However, the independence assumption among the variables for the squared unit-dimensional distances can lead to a poor approximation of the squared distance between two instances when the correlations among the squared unit-dimensional distances are ignored, and such poor approximation can be amplified when more squared unit-dimensional distances are missing as more missing components involve in the approximation. Consequently, this leads to inaccurate estimation of Gaussian kernels and therefore poor performance of kernel method models.

In this work, we propose a new method to estimate the Gaussian kernels with incomplete data considering the correlations among squared unit-dimensional distances, called the expected Gaussian kernel with correlated variables (EGKC). The proposed EGKC generalizes EGK considering the correlations among squared unit-dimensional distances of incomplete instances in the Gaussian kernel function. By incorporating the correlations among the unit-dimensional distances, EGKC leads to a better approximation of the squared distance between two instances i) when stronger correlations among the squared unit-dimensional distances exist and ii) when more squared unit-dimensional distances are missing. We model the squared Euclidean distance between two incomplete instances as the sum of the correlated squared unit-dimensional distances. In this modeling, we derive the distribution of the Gaussian kernel of incomplete instances using the approximation of the squared distance as the sum of the correlated squared unit-dimensional distances by a Gamma variable as the sum of correlated Gamma variables, adopting the approximation in (Feng et al., 2016). We show the parameter estimation of the Gamma, and we prove that a necessary condition for the approximation in (Feng et al.,

2016) is satisfied by the parameter estimation for squared unit-dimensional distance variables using the moments of random variables for missing components in original space. Then, the Gaussian kernel is obtained from the expected value of the Gaussian kernel function under the probability distribution for the squared distance between the instances.

The remainder of this chapter is organized as follows. In Section 3.2, we review the missing treatment methods for the Gaussian kernel for incomplete data. In Section 3.3, the proposed method is described in detail, and, in Section 3.4, experiments on synthetic data and a real-life case in semiconductor manufacturing are presented. Finally, we conclude this chapter and suggest future work in Section 3.5.

3.2 Related Work

Let $\mathbf{x}_i = [x_{i1}, \dots, x_{iD}]^{\mathrm{T}}$ be a D-dimensional vector for the i-th instance and let $\mathcal{m}_i$ be an index set of the missing dimensions of $\mathbf{x}_i$. For a pair of two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, the Gaussian kernel is given by

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left\{-\frac{1}{2\tau}\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2\right\} \tag{3.1}$$

where τ is a kernel parameter ($\tau > 0$) and $\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2 = \sum_{p=1}^{D}\left(x_{ip} - x_{jp}\right)^2$ is the squared Euclidean distance between the vectors. The presence of any missing in the two vectors (i.e., $\mathcal{m}_i \cup \mathcal{m}_j \neq \emptyset$), however, precludes the computation in (3.1).

To make available the computation of the Gaussian kernel with incomplete instances, missing components can be replaced with appropriate values which are typically estimated using the observed components of incomplete instances and other complete

instances. One may directly handle incomplete instances imputing the missing components in the original space of data, and then obtain the value from the calculation in (3.1) with the imputed intances as complete data. As a simple approach, the expected values of missing components as random variables are estimated from the conditional distribution given the observed values of the incomplete instance in the original space. That is, the missing component on x_{ip} in $\mathbf{x}_i$ as $p \in m_i$ is regarded as a random variable X_{ip} according to missing at random (Little and Rubin, 1986), and its expectation $E\left[X_{ip}\middle|\mathbf{x}_i^{(o)}\right]$ is employed to replace the missing component where $\mathbf{x}_i^{(o)}$ is the vector that consists of only the observed components of $\mathbf{x}_i$. Then, we obtain the imputed vector $\tilde{\mathbf{x}}_i$ whose p-th component is given by

$$\tilde{x}_{ip} = \begin{cases} E\left[X_{ip}\middle|\mathbf{x}_i^{(o)}\right] & \text{if } p \in m_i \\ x_{ip} & \text{otherwise} \end{cases}. \tag{3.2}$$

The missing imputation in (3.2) requires estimating the conditional distribution parameters. In the case of data that follow a multivariate normal distribution, the expectation conditional maximization method (Meng and Rubin, 1993; Sexton and Swensen 2000) can be a fine option (Elora et al., 2013). Then, the Gaussian kernel in (3.1) is computed with the imputed vectors $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}_j$ as $k\left(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j\right)$ where $\tilde{\mathbf{x}}_i = [\tilde{x}_{i1}, \dots, \tilde{x}_{iD}]^{\mathrm{T}}$.

EGK in (Mesquita, 2019) is a method to estimate the Gaussian kernel with incomplete instances directly in kernel space using the expectation of the Gaussian kernel function. For the pair of the instances i and j, the Gaussian kernel function in (3.1) can be written as a function of the squared distance:

$$f(\zeta_{ij};\tau) = \exp\left\{-\frac{\zeta_{ij}}{2\tau}\right\} \tag{3.3}$$

where ζ_{ij} is the squared Euclidean distance between the instance vectors $\mathbf{x}_i$ and $\mathbf{x}_j$. In EGK, the squared distance ζ_{ij} is modeled with the sum of the squared unit-dimensional distances over all the dimensions as $\zeta_{ij} = \sum_{p=1}^{D} \gamma_{ijp}$ where $\gamma_{ijp} = \left(x_{ip} - x_{jp}\right)^2$ is the squared unit-dimensional distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ in the p-th dimension for $p = 1, \ldots, D$.

In the presence of missing components in any of $\mathbf{x}_i$ and $\mathbf{x}_j$, the Gaussian kernel in (3.3) becomes a function of random variables. For example, if $p \in m_i$ and $p \in m_j$, the corresponding squared unit-dimensional distance γ_{ijp} is defined as a random variable as a function of the random variables for missing components in $\mathbf{x}_i$ and $\mathbf{x}_j$ such that $\gamma_{ijp} = \left(X_{ip} - X_{jp}\right)^2$.

In EGK, it is assumed that the squared unit-dimensional distance in the p-th dimension, γ_{ijp}, follows a Gamma distribution with the shape and scale parameters, $k_{ijp} > 0$ and $\theta_{ijp} > 0$, respectively, wherein Gamma distributions are employed for modeling with squared random variables (Mesquita et al., 2017) as well as non-negative random variables of skewed distributions (Covo and Elalouf, 2014; Genton, 2004; Johnson et al, 1970; Roberts and Geisser, 1966). Also, it is assumed that the random variables γ_{ijp} for $p = 1, \ldots, D$ are independent of each other.

The squared distance ζ_{ij} is approximated as Gamma distributed with the shape and scale parameters, $k_{ij} > 0$ and $\theta_{ij} > 0$, respectively, using the sum of independent Gamma random variables, γ_{ijp} for $p = 1, \ldots, D$ based on the approximation in (Covo and Elalouf, 2014). The parameters of ζ_{ij} are estimated by using its moments as

$$k_{ij} = \frac{E\left[\zeta_{ij}\right]^2}{Var\left(\zeta_{ij}\right)} \tag{3.4}$$

$$\theta_{ij} = \frac{Var\left(\zeta_{ij}\right)}{E\left[\zeta_{ij}\right]}. \tag{3.5}$$

Using the estimates from the original data, the expected squared distance $E\left[\zeta_{ij}\right]$ (Eirola et al., 2013) can be estimated as

$$E\left[\zeta_{ij}\right] = \sum_{p=1}^{D} \left\{\left(\tilde{x}_{ip} - \tilde{x}_{jp}\right)^2 + \sigma_{pp,i} + \sigma_{pp,j}\right\} \tag{3.6}$$

where $\sigma_{pp,i}$ is the conditional variance of X_{id} defined as

$$\sigma_{pp,i} = \begin{cases} Var\left(X_{id} \middle| \mathbf{x}_i^{(o)}\right) & \text{if } d \in m_i \\ 0 & \text{otherwise} \end{cases}, \tag{3.7}$$

and, similarly, the variance of the squared distance $Var\left(\zeta_{ij}\right)$ (Mesquita et al., 2019) can be estimated as

$$Var\left(\zeta_{ij}\right) = \sum_{p=1}^{D} \left\{2\left(\sigma_{pp,i} + \sigma_{pp,j}\right)^2 + 4\left(\sigma_{pp,i} + \sigma_{pp,j}\right)\left(\tilde{x}_{ip} - \tilde{x}_{jp}\right)^2\right\}. \tag{3.8}$$

Then, the Gaussian kernel for a pair of instances $(i, j) \in \left\{(i, j) \middle| m_i \cup m_j \neq \emptyset, i \neq j\right\}$ is obtained by computing the expected value of the Gaussian kernel function in (3.3) under the probability distribution of ζ_{ij} with the parameters in (3.4) and (3.5).

3.3 Proposed Method

We propose a new method to estimate the Gaussian kernel with incomplete data, called the expected Gaussian kernel with correlated variables (EGKC). In the proposed method, we consider a missing component x_{ip} (i.e., $p \in m_i$) as a random variable X_{ip} according to missing at random (Little and Rubin, 1986), and, for brevity, we use the notation X_{ip} also for observed components as $E[X_{ip}] = x_{ip}$ and $Var(X_{ip}) = 0$ if $p \notin m_i$.

3.3.1 Formulation

Consider a pair of D-dimensional vectors $\mathbf{x}_i = [x_{i1}, \ldots, x_{iD}]^{\mathrm{T}}$ and $\mathbf{x}_j = [x_{j1}, \ldots, x_{jD}]^{\mathrm{T}}$, and let m_i and m_j be index sets of the missing dimensions of $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively. For the pair of $\mathbf{x}_i$ and $\mathbf{x}_j$, the Gaussian kernel function in (3.1) can be written as a function of the squared distance:

In the proposed method, we assume that the squared unit-dimensional distance in the p-th dimension, γ_{ijp}, is Gamma distributed for $p \in m_i \cup m_j$ as $\gamma_{ijp} \sim Gamma(k_{ijp}, \theta_{ijp})$ with the shape and scale parameters, $k_{ijp} \geq \frac{1}{2}$ and $\theta_{ijp} > 0$, respectively, where its probability density function of γ_p is given by

$$p_{\gamma_{ijp}}(x) = \frac{x^{k_{ijp}-1}}{\Gamma(k_{ijp})\theta_{ijp}^{k_{ijp}}} \exp\left\{-\frac{x}{\theta_{ijp}}\right\}.$$

The variables, γ_{ijp} for $p = 1, \ldots, D$, are not necessarily distributed with identical parameters. Therefore, in the presence of the correlations among γ_{ijp} for all p, the variable

for the squared distance ζ_{ij}^{*} is from the sum of the correlated Gamma random variables whose distributions have unequal shape and scale parameters.

We approximate the squared distance ζ_{ij}^{*} as Gamma distributed with the shape and scale parameters, $k_{ij}^{*} > 0$ and $\theta_{ij}^{*} > 0$, respectively, using the sum of the correlated Gamma random variables, γ_{ijp} for $p = 1, \ldots, D$, based on the approximation in (Feng et al., 2016) as follows.

Let $\boldsymbol{\gamma}_{ij} = \left[\gamma_{ij1}, \ldots, \gamma_{ijD}\right]^{\mathrm{T}}$ be a D-dimensional vector of the squared unit-dimensional distances and $\mathbf{R}_{ij}$ be the covariance matrix of $\boldsymbol{\gamma}_{ij}$ whose (p, q)-th element is $R_{ij,pq} = Cov\left(\gamma_{ijp}, \gamma_{ijq}\right)$. Let $\boldsymbol{\omega}_{ij} = \left[\omega_{ij1}, \ldots, \omega_{ijD}\right]^{\mathrm{T}}$ be a D-dimensional vector of independent Gamma random variables $\omega_{ijk} \sim Gamma\left(a_{ijp}, b_{ijp}\right)$ for $p = 1, \ldots, D$ where the covariance matrix of $\boldsymbol{\omega}$ is a D-by-D identity matrix. The original Gamma random variable γ_{ijp} can be approximated using the sum of weighted independent Gamma random variable ω_{ijk} for $k = 1, \ldots, D$ (Zhang et al., 2004) as

$$\gamma_{ijp} = \sum_{k=1}^{p} l_{ijpk}\omega_{ijk} \tag{3.9}$$

where l_{ijpk} the (p, k)th element of $\mathbf{L}_{ij}$ that is a unique lower triangular matrix from the Cholesky factorization on $\mathbf{R}_{ij}$ given by

$$\mathbf{R}_{ij} = \mathbf{L}_{ij}\mathbf{L}_{ij}^{T}. \tag{3.10}$$

Using (3.9), we can obtain the variable ζ_{ij} from the sum of weighted independent Gamma random variables as

$$\zeta_{ij} = \sum_{p=1}^{D} \sum_{k=1}^{p} l_{ijpk} \omega_{ijk} = \sum_{p=1}^{D} \omega_{ijp} \sum_{k=p}^{D} l_{ijkp} = \sum_{p=1}^{D} \omega'_{ijp} \tag{3.11}$$

where $\omega'_{ijp} \sim G\left(a_{ijp}, b'_{ijp}\right)$ with $b'_{ijp} = b_{ijp} \sum_{k=p}^{D} l_{ijkp}$. Then, the distribution for ζ_{ij} can be obtained from the approximation from the sum of independent Gamma random variables (Nakagami, 1960), and its shape and scale parameters, k^*_{ij} and θ^*_{ij}, respectively, are given as

$$k^*_{ij} = \frac{\left(\sum_{p=1}^{D} a_{ijp} b'_{ijp}\right)^2}{\sum_{p=1}^{D} a_{ijp} \left(b'_{ijp}\right)^2} = \frac{\left(\sum_{p=1}^{D} a_{ijp} b_{ijp} \sum_{k=p}^{D} l_{ijkp}\right)^2}{\sum_{p=1}^{D} \left(\sum_{k=p}^{D} l_{ijkp}\right)^2} \tag{3.12}$$

$$\theta^*_{ij} = \frac{\sum_{p=1}^{D} a_{ijp} \left(b'_{ijp}\right)^2}{\sum_{p=1}^{D} a_{ijp} b'_{ijp}} = \frac{\sum_{p=1}^{D} \left(\sum_{k=p}^{D} l_{ijkp}\right)^2}{\sum_{p=1}^{D} a_{ijp} b_{ijp} \sum_{k=p}^{D} l_{ijkp}} \tag{3.13}$$

where $Var\left(\omega_p\right) = a_{ijp} b^2_{ijp} = 1$. Based on (3.9) and (3.10), the distribution of ζ^*_{ij} is characterized with the approximated shape and scale parameters, k^*_{ij} and θ^*_{ij}, respectively, rewriting (3.12) and (3.13) as

$$k^*_{ij} = \frac{\left(\sum_{p=1}^{D} k_{ijp} \theta_{ijp}\right)^2}{\sum_{p=1}^{D} \sum_{q=1}^{D} R_{ijpq}} \tag{3.14}$$

$$\theta^*_{ij} = \frac{\sum_{p=1}^{D} \sum_{q=1}^{D} R_{ijpq}}{\sum_{p=1}^{D} k_{ijp} \theta_{ijp}} \tag{3.15}$$

where $k_{ijp} \theta_{ijp} = a_{ijp} b_{ijp} \sum_{k=p}^{D} l_{ijkp}$ and $\sum_{p=1}^{D} \left(\sum_{k=p}^{D} l_{ijkp}\right)^2 = \sum_{p=1}^{D} \sum_{q=1}^{D} R_{ijpq}$.

Finally, EGKC for the pair of the instances (i, j), $k_{EGKC}\left(\mathbf{x}_i, \mathbf{x}_j\right)$, can be obtained by estimating the expected Gaussian kernel function $f\left(\zeta_{ij}; \tau\right)$ in (3.3) under the probability

distribution of ζ_{ij} with the parameters in (3.14) and (3.15) as the expectation of $k_{EGKC}(\mathbf{x}_i, \mathbf{x}_j)$ and its variance are, respectively,

$$E_{\zeta_{ij}^*}\left[k(\mathbf{x}_i, \mathbf{x}_j)\right] = \left(\frac{2\tau}{2\tau + \theta_{ij}^*}\right)^{k_{ij}^*} \tag{3.16}$$

$$Var\left(k(\mathbf{x}_i, \mathbf{x}_j)\right) = \left(\frac{\tau}{\tau + \theta_{ij}^*}\right)^{k_{ij}^*} \left(1 - \frac{2\tau}{2\tau + \theta_{ij}^*}\right)^{2k_{ij}^*}. \tag{3.17}$$

3.3.2 Estimation of EGKC

To estimate the expectation and variance of the Gaussian kernel in (3.17) and (3.18), we first estimate the parameters of γ_{ijp} by the expected squared unit-dimensional distance in the p-th dimension $E\left[\gamma_{ijp}\right]$ and its variance $Var(\gamma_{ijp})$. Similar to the methods for the distances between two vectors shown in (3.6) and (3.8), $E\left[\gamma_{ijp}\right]$ and $Var(\gamma_{ijp})$ can be obtained by matching the moments with the conditional means in (3.2) and the conditional variances in (3.7) of the missing components in original space:

$$E\left[\gamma_{ijp}\right] = \left(\tilde{x}_{ip} - \tilde{x}_{jp}\right)^2 + \sigma_{pp,i} + \sigma_{pp,j} \tag{3.18}$$

$$Var(\gamma_{ijp}) = 2\left(\sigma_{pp,i} + \sigma_{pp,j}\right)\left\{\sigma_{pp,i} + \sigma_{pp,j} + 2\left(\tilde{x}_{ip} - \tilde{x}_{jp}\right)^2\right\}. \tag{3.19}$$

Proposition 1 shows the estimation in (3.18) and (3.19) for γ_{ijp} is sufficient to the condition of the Gamma approximiation described in Section 3.3.1, and the proof of Proposition 1 is in Appendix B. Besides, we may notice that the sum of the expectation in (3.18) for $p =$

1, …, D and the sum of variance in (3.19) for $p = 1, \ldots, D$ become equivalent to (3.6) and (3.8), resepectively, if the independence among γ_{ijp} for $p = 1, \ldots, D$ is assumed.

Proposition 1 *Let X_{ip} and X_{ip} be a random variable for the missing component of instance i in the p-th and q-th dimensions, respectively, and let $\gamma_{ijp} = \left(X_{ip} - X_{jp}\right)^2 \sim Gamma\left(k_{ijp}, \theta_{ijp}\right)$ be a random variable for the squared unit-dimensional distance between instances i and j for $i \neq j$ and $\sigma_{pp,i} + \sigma_{pp,i} \neq 0$. Then, the parameters estimated using the moments of X_{ip} and X_{jp} in (3.18) and (3.19) satisfy the condition of the parameters, $k_{ijp} \geq \frac{1}{2}$ and $\theta_{ijp} > 0$, in the Gamma approximation using the sum of the correlated Gammas.*

Also, for the approxmiation of the Gamma distribution for ζ_{ij}^*, it is required to estimate the covariances between γ_{ijp} and γ_{ijq}, R_{ijpq}, for $p, q \in \{1, \ldots, D\}$. However, it is unavailable to directly estimate the covariance where the joint distribution of γ_{ijp} and γ_{ijq} is not given. Similar to the estimation in (3.18) and (3.19), we estimate the covariance $R_{ijpq} = Cov\left(\gamma_{ijp}, \gamma_{ijq}\right)$, by means of the moments of the missing components in original space.

The covariance between γ_{ijp} and γ_{ijq} can be written in terms of the random variables for missing components in original space as

$$\begin{aligned} &Cov\left(\gamma_{ijp}, \gamma_{ijq}\right) \\ &= E\left[\left(X_{ip} - X_{jp}\right)^2 \left(X_{iq} - X_{jq}\right)^2\right] - E\left[\left(X_{ip} - X_{jp}\right)^2\right] E\left[\left(X_{iq} - X_{jq}\right)^2\right]. \end{aligned} \tag{3.20}$$

Under the assumption of the independence between two instances $\mathbf{x}_i$ and $\mathbf{x}_j$, it becomes

$$
\begin{aligned}
&Cov(\gamma_{ijp}, \gamma_{ijq}) \\
&= E[X_{ip}^2 X_{iq}^2] - 2E[X_{ip}^2 X_{iq}]E[X_{jq}] - 2E[X_{ip} X_{iq}^2]E[X_{jp}] + 4E[X_{ip} X_{iq}]E[X_{jp} X_{jq}] \\
&-2E[X_{ip}]E[X_{jp} X_{jq}^2] - 2E[X_{iq}]E[X_{jp}^2 X_{jq}] + E[X_{jp}^2 X_{jq}^2] + 2E[X_{ip}^2]E[X_{iq}]E[X_{jq}] \\
&-E[X_{ip}^2]E[X_{jq}^2] + 2E[X_{ip}]E[X_{jp}]E[X_{iq}^2] - 4E[X_{ip}]E[X_{iq}]E[X_{jp}]E[X_{jq}] \\
&+2E[X_{ip}]E[X_{jp}]E[X_{jq}^2] + 2E[X_{jp}^2]E[X_{iq}]E[X_{jq}] - E[X_{jp}^2]E[X_{jq}^2]
\end{aligned} \tag{3.21}
$$

where the two terms in (3.20) are

$$
\begin{aligned}
&E\left[\left(X_{ip} - X_{jp}\right)^2 \left(X_{iq} - X_{jq}\right)^2\right] \\
&= E[X_{ip}^2 X_{iq}^2] - 2E[X_{ip}^2 X_{iq}]E[X_{jq}] + E[X_{ip}^2]E[X_{jq}^2] - 2E[X_{ip} X_{iq}^2]E[X_{jp}] \\
&+4E[X_{ip} X_{iq}]E[X_{jp} X_{jq}] - 2E[X_{ip}]E[X_{jp} X_{jq}^2] + E[X_{iq}^2]E[X_{jp}^2] - 2E[X_{iq}]E[X_{jp}^2 X_{jq}] \\
&+E[X_{jp}^2 X_{jq}^2]
\end{aligned}
$$

and

$$
\begin{aligned}
&E\left[\left(X_{ip} - X_{jp}\right)^2\right] E\left[\left(X_{iq} - X_{jq}\right)^2\right] \\
&= E[X_{ip}^2]E[X_{iq}^2] - 2E[X_{ip}^2]E[X_{iq}]E[X_{jq}] + E[X_{ip}^2]E[X_{jq}^2] - 2E[X_{ip}]E[X_{jp}]E[X_{iq}^2] \\
&+4E[X_{ip}]E[X_{iq}]E[X_{jp}]E[X_{jq}] - 2E[X_{ip}]E[X_{jp}]E[X_{jq}^2] + E[X_{iq}^2]E[X_{jp}^2] \\
&-2E[X_{jp}^2]E[X_{iq}]E[X_{jq}] + E[X_{jp}^2]E[X_{jq}^2].
\end{aligned}
$$

To compute the high-order moments of the i-th instance in (3.21), let $\mathbf{x}_{i(pq)} = [X_{ip}, X_{iq}]^{\mathrm{T}}$ be the bivariate normal distribution, as a subset of the variables in $\mathbf{x}_i$, with the mean $\tilde{\mathbf{x}}_{i(pq)} = [\tilde{x}_{ip}, \tilde{x}_{iq}]^{\mathrm{T}}$ and covariance matrix $\tilde{\mathbf{S}}_{i(pq)} = \begin{bmatrix} \sigma_{pp,i} & \sigma_{pq,i} \\ \sigma_{pq,i} & \sigma_{qq,i} \end{bmatrix}$ where

$$\sigma_{pq,i} = \begin{cases} Cov\left(X_{ip}, X_{iq} \middle| \mathbf{x}_i^{(o)}\right) & \text{if } p, q \in m_i \\ 0 & \text{otherwise} \end{cases}. \quad (3.22)$$

Let $M(\mathbf{t})$ be the moment generating function of $\mathbf{x}_{i(pq)}$ with a variable vector $\mathbf{t} = \left[t_p, t_q\right]^{\mathrm{T}}$ as

$$M(\mathbf{t}) = \exp\left\{\mathbf{t}^T \tilde{\mathbf{x}}_{i(pq)} + \frac{1}{2}\mathbf{t}^T \tilde{\mathbf{S}}_{i(pq)} \mathbf{t}\right\}.$$

High-order raw cross moments of $\mathbf{x}_{i(pq)}$ are given by

$$E\left[X_{ip}^{k_1} X_{iq}^{k_2} \middle| \tilde{\mathbf{x}}_{i(pq)}, \tilde{\mathbf{S}}_{i(pq)}\right] = \left.\frac{\partial^{k_1+k_2} M(\mathbf{t})}{\partial t_p^{k_1} t_q^{k_2}}\right|_{\boldsymbol{t}=\mathbf{0}},$$

and, accordingly, we have

$$E\left[X_{ip}^2\right] = \tilde{x}_{ip}^2 + \sigma_{pp,i} \quad (3.23)$$

$$E\left[X_{ip} X_{iq}\right] = \sigma_{pq,i} + \tilde{x}_{ip}\tilde{x}_{iq} \quad (3.24)$$

$$E\left[X_{ip}^2 X_{iq}\right] = \sigma_{pp,i}\mu_{iq} + 2\sigma_{pq,i}\tilde{x}_{ip} + \tilde{x}_{ip}^2\tilde{x}_{iq} \quad (3.25)$$

$$E\left[X_{ip}^2 X_{iq}^2\right] = \sigma_{pp,i}\sigma_{qq,i} + \sigma_{pp,i}\tilde{x}_{iq}^2 + \sigma_{qq,i}\tilde{x}_{ip}^2 + 2\sigma_{pq,i}^2 + 4\sigma_{pq,i}\tilde{x}_{ip}\tilde{x}_{iq} + \tilde{x}_{ip}^2\tilde{x}_{iq}^2. \quad (3.26)$$

From (3.23) - (3.26), we estimate the covariance in (3.21) with the moments of the variables for missing components in original space as

$$Cov\left(\gamma_{ijp}, \gamma_{ijq}\right) = 2\left(\sigma_{pq,i} + \sigma_{pq,j}\right)\left\{\sigma_{pq,i} + \sigma_{pq,j} + 2\left(\tilde{x}_{ip} - \tilde{x}_{jp}\right)\left(\tilde{x}_{iq} - \tilde{x}_{jq}\right)\right\}. \quad (3.27)$$

Note that the covariance in (3.27) becomes equivalent to the variance in (3.19) in the case of $p = q$. Finally, the parameters of ζ_{ij}^* are estimated using (3.18) and (3.26) as

$$k_{ij}^* = \frac{E\left[\zeta_{ij}^*\right]^2}{Var\left(\zeta_{ij}^*\right)} \tag{3.28}$$

$$\theta_{ij}^* = \frac{Var\left(\zeta_{ij}^*\right)}{E\left[\zeta_{ij}^*\right]} \tag{3.29}$$

with

$$E\left[\zeta_{ij}^*\right] = \sum_{p=1}^{D} \left\{\left(\tilde{x}_{ip} - \tilde{x}_{jp}\right)^2 + \sigma_{pp,i} + \sigma_{pp,j}\right\} \tag{3.30}$$

$$Var\left(\zeta_{ij}^*\right) = \sum_{p=1}^{D}\sum_{q=1}^{D} 2\left(\sigma_{pq,i} + \sigma_{pq,j}\right)\left\{\sigma_{pq,i} + \sigma_{pq,j} + 2\left(\tilde{x}_{ip} - \tilde{x}_{jp}\right)\left(\tilde{x}_{iq} - \tilde{x}_{jq}\right)\right\}. \tag{3.31}$$

3.3.3 Implementation

Given a dataset of N intances, $\{\mathbf{x}_i\}_{i=1}^N$ and its index set $\mathcal{X} = \{1, \ldots, N\}$, let $\mathcal{M}$ be a set of incomplete instance indexes ($\mathcal{M} \subset \mathcal{X}$) where $i \in \mathcal{M}$ if $m_i \neq \emptyset$. The Gaussian kernels with incomplete data and EGKC are obtained according to the procedure described in Algorithm 3.1.

Algorithm 3.1 Estimation of the Gaussian kernels with EGKC

Input: a data set $\{\mathbf{x}_i\}_{i=1}^N$, a instance index set $\mathcal{X}$, and an incomplete instance index set $\mathcal{M}$

Estimate mean $\boldsymbol{\mu}_{\mathcal{X}}$ and covariance $\boldsymbol{\Sigma}_{\mathcal{X}}$ from data
Repeat for all $i \in \mathcal{X}$
  Obtain $\tilde{x}_{ip}$ in (3.2) for all p and $\sigma_{pq,i}$ in (3.22) for all (p, q)
End
Repeat for $i \in \mathcal{X}$,
  Repeat for $j \in \mathcal{X}$,
    If $i = j$,
      Set $k(\mathbf{x}_i, \mathbf{x}_j) = 1$
    Else if $i, j \notin \mathcal{M}$,
      Obtain $k(\mathbf{x}_i, \mathbf{x}_j)$ with the standard Gaussian kernel
    Else
      Compute $E[\zeta_{ij}^*]$ in (3.30) and $Var(\zeta_{ij}^*)$ in (3.31)
      Estimate the parameters k_{ij}^* in (3.28) and θ_{ij}^* in (3.29)
      Compute $k_{EGKC}(\mathbf{x}_i, \mathbf{x}_j)$ from the expectation in (3.16)
    End
  End
End

Output: a kernel matrix $\mathbf{K} \in \mathbb{R}^{N \times N}$ where $[\mathbf{K}]_{ij} = k_{EGKC}(\mathbf{x}_i, \mathbf{x}_j)$

3.4 Experiments

We evaluate the performance of EGKC by conducting experiment with synthetic data and real-life case data. Using synthetic data from multivariate normal distribution, we compare the Gaussian kernels estimated by EGKC with those estimated by EGK. Then, we conduct a comparative experiment using a real-life data from a multi-pattern photolithography process in wafer fabrication**.**

For the following experiments, we employed the relevance vector machine (RVM) (Tipping, 2001), which is a kernel-based machine learning method for sparse Bayesian regression. RVM chooses relevance vectors from the instances in a given training data set, and a RVM model with the relevance vectors as the basis can perform prediction tasks with high sparsity, maintaining the Bayesian property. The applications of RVM has been shown with successful performance in predictive tasks in manufaturing (Caesarendra et al., 2010; Chang et al., 2017; Di Maio et al., 2012; Hwang et al., 2014).

3.4.1 Synthetic data: Multivariate normal data

We evaluate the estimation quality of the kernels from EGKC using the difference from true data. Let $\mathbf{x} = [X_1, X_2, \dots, X_6]$ be an input variable vector that follows a multivariate normal distribution as $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with a zero mean vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$ whose diagonal elements Σ_{ii} for $i = 1, \dots, 6$ are one and whose off-diagonal elements Σ_{ij} are c_{corr} to control for the correlations of variables of interest for $(i, j) \in \{(1,2), (3,4), (5,6)\}$ and 0 otherwise. Then, an output variable $Y \in \mathbb{R}$ is defined by a function of the input variables as

$$Y = u_1 X_1 \sin X_2 + u_2 X_3 \sin X_4 + u_3 X_5 \sin X_6 + \varepsilon$$

where $u_j \in \{-10,10\}$ is a coefficient with the randomly chosen sign for $j = 1,2,3$ and $\varepsilon \sim N(0,0.2)$.

Based on the distribution, we generated training data of $N_{tr} = 100$ instances at random as an input matrix $\mathbf{X}_{tr} \in \mathbb{R}^{N_{tr} \times M}$ and an output vector $\mathbf{y} \in \mathbb{R}^{N_{tr}}$. We divided the data randomly into two sets of N_o instances as complete data $\mathbf{X}_{tr}^{(o)} = \left[\mathbf{x}_1, \dots, \mathbf{x}_{N_o}\right]^{\mathrm{T}}$ and N_m instances as incomplete data $\mathbf{X}_{tr}^{(m)} = \left[\mathbf{x}_{N_o+1}, \dots, \mathbf{x}_{N_{tr}}\right]^{\mathrm{T}}$ where $\mathbf{x}_i$ is the i-th instance vector and $N_{tr} = N_o + N_m$. We made missing the values for the pair of variables of interest together as $\mathbf{Z}_{tr}^{(m)} = \left[\mathbf{z}_{N_o+1}, \dots, \mathbf{z}_{N_{tr}}\right]^{\mathrm{T}}$ where $\mathbf{z}_i$ is the vector $\mathbf{x}_i$ with missing components. Then, we trained RVM models using the incomplete data $\mathbf{Z}_{tr} = \left[\left(\mathbf{Z}_{tr}^{(o)}\right)^{\mathrm{T}}, \left(\mathbf{Z}_{tr}^{(m)}\right)^{\mathrm{T}}\right]^{\mathrm{T}}$ where $\mathbf{Z}_{tr}^{(o)} = \mathbf{X}_{tr}^{(o)}$. We generated testing data of $N_{te} = 100$ complete instances at random from the same distribution as $\mathbf{X}_{te} \in \mathbb{R}^{N_{te} \times M}$. We measured the performance of the prediction models computing the mean absolute error (MAE) as $MAE_{EGKC} = \sum_{i=1}^{N_{te}} |e_i| / N_{te}$ and the root mean square error (RMSE) as $RMSE = \sqrt{\sum_{i=1}^{N_{te}} e_i^2 / N_{te}}$.

Figures 3.1 and 3.2 present the averaged prediction results of EGK and EGKC over 10 replications with the Gaussian kernel parameter τ and the missing rate η. Roughly speaking, EGKC outperformed EGK in the varying kernel parameters and missing rates. EGKC improved the prediction performance as the correlation coefficient c_{corr} increased. Also, we may notice that such improvement by EGKC was amplified by the increase of missing ratio wherein the more missing components, the more the estimation of the kernels affected the prediction model.

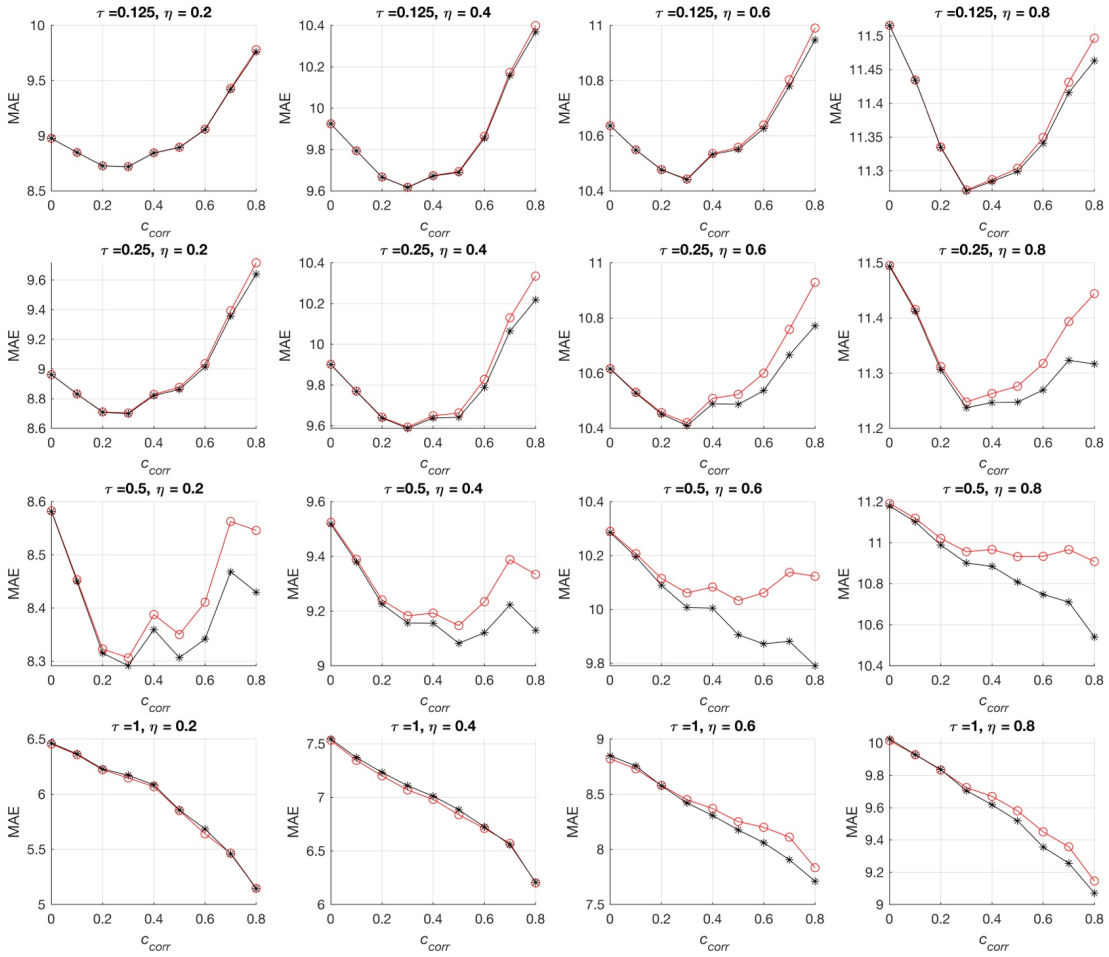


Figure 3.1 Testing MAE of EGK (red circles) and EGKC (black asterisks) in synthetic data.

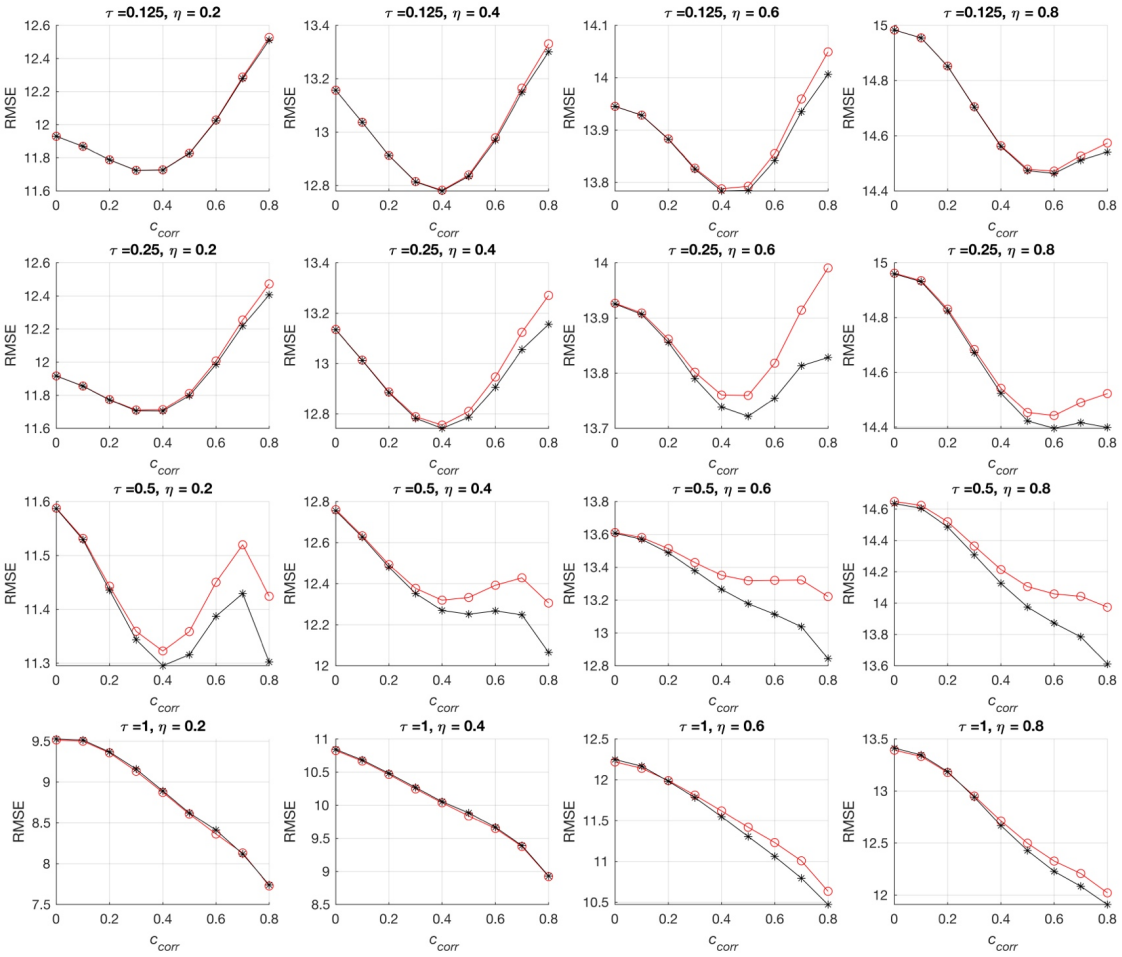


Figure 3.2 Testing RMSE of EGK (red circles) and EGKC (black asterisks) in synthetic data.

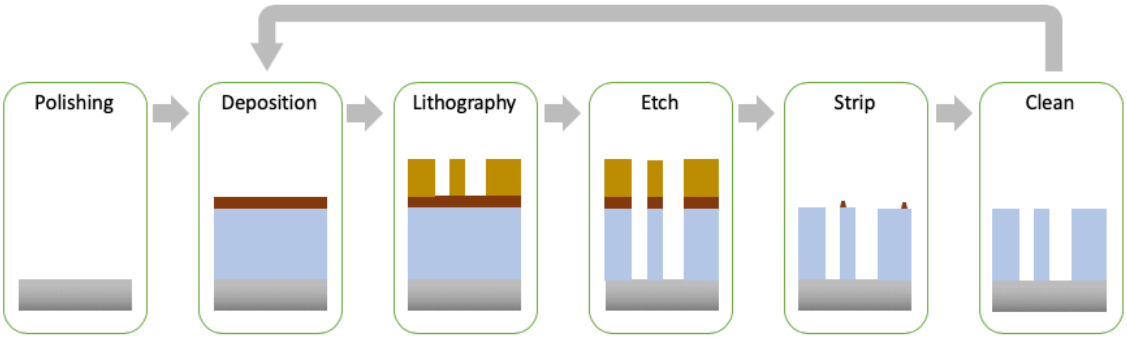


Figure 3.3 A schematic procedure of multistage wafer fabrication process.

3.4.2 Case study: Prediction in wafer quality at an etching process

We consider a multi-pattern photolithography process in wafer fabrication. The wafer fabrication is typically performed in a complex multistage process that consists of numerous stages. At each stage a specified unit process such as chemical-mechanical polishing (CMP), chemical vapor deposition (CVD), planarization and lithography (PHOTO), and etching (ETCH) is performed, and wafers in a batch go through the set of stages predefined along with a fabrication recipe for final products as shown in Figure 3.3.

For the process control in semiconductor manufacturing processes, monitoring models are built typically using wafers quality characteristics measured at each stage after the completion of the corresponding operation. A few critical dimensions for wafer qualities are selected based on engineers' knowledge such as the thickness of the remaining silicon on wafers at a unit etching process stage, and then the physical measurements of the quality characteristics are obtained from sampled lots of wafers.

A goal in a multistage process is to monitor the process with wafer qualities at a critical stage and thereby to control the final product volume according to a production plan considering undesirable defective chips on the wafers. The product qualities are determined not only by the corresponding unit process stage but also by the results from the preceding stages as accumulated antecedents on the products.

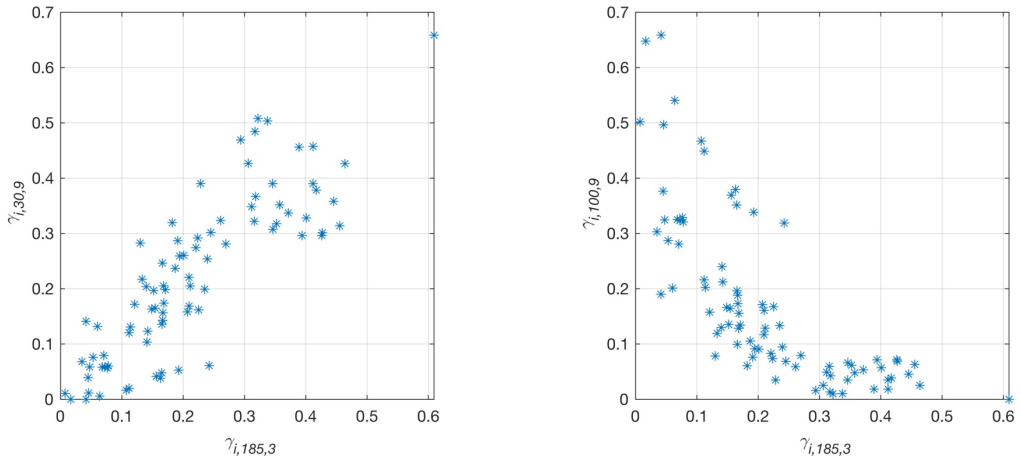
Collecting data for process analysis at an inter-stage level requires much time and cost. Although it is available to readily monitor the qualities of wafers at a unit process stage taking the measurements from sampled wafers, the sampled wafers need to be traced along the preceding/subsequent stages based on the fabrication recipe in order to get the complete measurements for the inter-stage analysis. Besides, it is cumbersome to collect such inter-stage data wherein the relevant unit processes are interrupted to obtain measurements for the sampled wafers each time. In these regards, the number of the wafers observed fully at all the stages is often limited comparing to that of the wafer partially observed in a multistage process.

This experiment aims to predict the quality of wafers at an etching process using the information of its preceding unit processes, determined by technical relationships based on engineers' knowledge, in a multi-pattern fabrication process. The target etching process is performed after the completion of eight preceding unit processes, and wafer quality at each unit process is measured on a predetermined measurement such as thickness (THK) and critical dimensions (CDs). The data collected from the nine processes consists of 204 instances. All the measurements at the target process were available, and the measurements of 75 instances were fully observed at all the preceding processes whereas those of the rest

Table 3.1 Unit processes preceding to the target process.

	Variable	Process type	Measurement type	Number of observed instances
Preceding unit process	X_1	CMP	THK	78
	X_2	CVD	THK	117
	X_3	CVD	THK	162
	X_4	PHOTO	THK	68
	X_5	PHOTO	THK	141
	X_6	PHOTO	THK	120
	X_7	ETCH	THK	86
	X_8	CVD	THK	119
	X_9	ETCH	THK	71
	X_{10}	CVD	THK	163
	X_{11}	ETCH	CD	140
	X_{12}	CVD	THK	144
	X_{13}	PHOTO	THK	110
	X_{14}	ETCH	CD	102
	X_{15}	ETCH	THK	63
	X_{16}	ETCH	THK	127
Target process	Y	ETCH	CD	316

were partially observed. Table 3.1 presents the basic information about the processes and the observations therefrom, and the squared unit-dimensional distances of the observed instances presented linear relationships with the others as shown in Figure 3.4.



(a) (b)

Figure 3.4 Scatter plots of the squared unit-dimensional distance in the multi-pattern photolithography process data: (a) $\gamma_{i,30,9}$ against $\gamma_{i,185,3}$ for all i and (b) $\gamma_{i,100,9}$ against $\gamma_{i,185,3}$ for all i.

We evaluated the performance of the proposed EGKC using RVM (*RVM-EGKC*) over ten-fold cross validation. The predictive performance with the kernels from EGKC was compared with the performances with the kernels from the instances imputed with simple unconditional means in respective dimensions (*RVM-SMI*), the performances with the kernels from the instances imputed with conditional means in (3.2) (*RVM-CMI*), and from the instances imputed with EGK (*RVM-EGK*). The Gaussian kernel parameter τ for each of the comparing models was chosen through the five-fold cross validation in the training data.

Table 3.2 shows the prediction performance of the RVM models and the statistical significance in the two-side *t*-test between the results of *RVM-EGKC* and the others. First,

Table 3.2 Testing results in photolithography process data.

	RVM-SMI	*RVM-CMI*	*RVM-EGK*	*RVM-EGKC*
MAE ($\times 10^2$)	10.422^{**}	9.510^{***}	8.975^{*}	**8.792**
RMSE ($\times 10^2$)	11.850^{***}	10.944^{**}	9.533^{*}	**8.388**

* p-value < 0.10; ** p-value < 0.05; *** p-value < 0.01

we notice that the imputation in kernel space (*RVM-EGK* and *RVM-EGKC*) outperformed with the imputation in original space (*RVM-SMI* and *RVM-CMI*). Furthermore, *RVM-EGKC* led to more accurate prediction than *RVM-EGK* in the multi-pattern photolithography process data of high missing ratio and correlations among the squared unit-dimensional distances.

3.5 Conclusion

We proposed a new method to estimate the expected Gaussian kernel for incomplete data. considering the correlation between missing variables. The proposed method generalized the existing EGK considering the correlations among the squared unit-dimensional distances of incomplete instances in the Gaussian kernel. We derived the distribution of the Gaussian kernel for incomplete data, modeling the squared distance between two instance vectors as the sum of the correlated squared unit-dimensional distances and approximating the squared distance as the sum of the correlated Gammas by a Gamma.

We evaluated the proposed method on synthetic data and real-life case of high missing ratio and correlations in the squared unit-dimensional distances for the prediction of wafer quality characteristic at an etching process using a sparse Bayesian kernel machine for regression. The proposed method with showed competitive performances in the cases of correlated variables.

As this work has focused on modeling variables for the components of Gaussian kernels with Gamma variables, future work should investigate different distributions and kernels. Also, the proposed method may be extended to handle missing components of different types of data such as spatial data and time series.

CHAPTER 4

DEEP AUTOENCODER WITH CLIPPING FUSION REGULARIZATION ON MULTI-STEP PROCESS SIGNALS

4.1 Introduction

Virtual metrology (VM) in semiconductor manufacturing is a tool to monitor wafers in processes through data from fabrication equipment. Unlike conventional approaches based on physical measurements, quality characteristics of wafers in VM are predicted based on sensory data on process equipment. Therefore, cost reduction can be achieved in process and production control by dynamic monitoring wherein numerous stages are involved in wafer fabrication (Gazzola et al., 2018).

Feature extraction is a key for building a successful VM model. As extensive data are collected from numerous sensors on process equipment, features are required not only to explain latent relationships with outputs (i.e., wafer quality characteristics), but also to compress high-dimensional data from the sensors into a feature space as well. For VM modeling, diverse feature extraction methods are employed in the literature: summary statistics from each sensor signal over a whole process (Kang et al., 2011) and over the subprocesses of a whole process (Hirai and Kano, 2015; Hwang et al., 2014) and principal component analysis (PCA) (Susto et al., 2015).

In wafer fabrication, a sensor signal at a process stage may consist of several heterogeneous signals from subprocesses. To be specific, the whole process can be divided into subprocess steps according to a product recipe, and the setup may differ from subprocess to subprocess to meet requirements on the product recipe. The signals from the

sensors on the equipment are reflected by such changes in the process equipment setup according to subprocesses. Thus, it is frequently observed that the signals have transient changes between subprocesses. This characteristic of data, however, is considered in no existing feature extraction method.

In this study, we aim to extract features from high-dimensional signals of process-equipment sensors for VM considering the characteristics of the signals that consist of multiple sub-processes. To do so, we present a new unsupervised deep autoencoder (AE) with the clipping fusion regularization (Choi and Jeong, 2018). The proposed model is evaluated by conducting a comparative experiment using a real-life dataset from an etching process for wafer fabrication.

The remainder of this chapter is organized as follows. In Section 4.2, we review the related literatures on deep learning models and regularizatino techniques. In Section 4.3, the proposed method is described in detail, and, in Section 4.4, experiments on a real-life case in semiconductor manufacturing are presented. Finally, we conclude this chapter and suggest future work in Section 4.5.

4.2 Related Work

AE is an unsupervised feature extraction method that compresses inputs into latent variables (i.e., neurons) on a hidden layer by reconstructing the inputs in a neural network. In the network, inputs are encoded typically into a lower dimensional latent space, and features from the space are decoded reconstructing the inputs. In the neural network for AE, nonlinear activation functions, such as logistic sigmoid and hyper-tangent functions, enable us to extract features considering the nonlinearity of data.

In recent years, AEs with deep architectures have been studied in the literature for feature extraction tasks with complex data (Khatab et al., 2018) as deep learning-based approaches show great performance in many applications (Längkvist et al., 2014; Schmidhuber, 2015), as well as in semiconductor manufacturing modeling (Lee et al., 2017). Regularization techniques are widely used in modeling deep structured neural network (Schmidhuber, 2015). Such models employ regularization terms to achieve diverse purposes while training the models. For example, the ℓ_1 or ℓ_2-norm regularization on a neural network model is commonly used to penalize weight values so that it helps the model avoid overfitting by taking critical weights only. Also, critical features can be extracted by suppressing a group of weights associated with a neuron (Scardapane et al., 2017).

In regression modeling with signal data, the fusion regularization (Land and Friedman, 1996) and the fused LASSO regularization elaborated therefrom (Tibshirani et al., 2005) are developed to take the nature of sequential/spatial data into account. With the fusion regularization, the neighboring variables from a signal are forced to be similar to each other, which leads the model to be trained with denoising the signal (Hoefling, 2010).

Recently, researchers have been using the technique of clipping values (like max-norm constraints) in deep learning models on diverse purposes such as for gradients to train models efficiently (Abadi et al., 2016), for activations to stabilize the model (Gupta et al., 2015), and for feature values to denoise (Yosinski et al., 2015), and weights for binarization (Courbariaux et al., 2016). In this study, we clip penalizing values in training an AE model to control the intensity of the fusion regularization on the transient changes in signals between subprocesses.

4.3 Proposed Model

Consider an input vector, $\mathbf{x} \in \mathbb{R}^{m_0}$, from S sensors, each of which collects a signal of a wafer in process during T processing time, ($m_0 = S \times T$) and a target value, $y \in \mathbb{R}$, from a quality characteristic of the wafer processed.

First, we extract features from the raw signals using an AE model as a feed-forward network f with K hidden layers. The activations of m_k neurons on the k-th layer for $k = 1, \ldots, (K+1)$ in f, $\mathbf{a}^{(k)} \in \mathbb{R}^{m_k}$, is given by

$$\mathbf{a}^{(k)} = g_{(k)}\left(\mathbf{W}^{(k)}\mathbf{a}^{(k-1)} + \mathbf{b}^{(k)}\right) \tag{4.1}$$

where $\mathbf{W}^{(k)} \in \mathbb{R}^{m_{(k-1)} \times m_k}$, $\mathbf{b}^{(k)} \in \mathbb{R}^{m_k}$, and $g_{(k)}$ are the weight matrix, the bias vector, and element-wise activation function, respectively, on the k-th layer. In (4.1), $\mathbf{a}^{(0)} = \mathbf{x}$ and $\mathbf{a}^{(K+1)} = f(\mathbf{x}; \boldsymbol{\theta})$ where $\boldsymbol{\theta}$ indicates a set of all the parameters in f. The target vector in f is set to the original input $\mathbf{x}$, and the model is trained to obtain the reconstructed input $\hat{\mathbf{x}} = f(\mathbf{x}; \boldsymbol{\theta})$ minimizing the reconstruction loss (i.e., the prediction error of the AE), $\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}})$, where $\mathcal{L}(\cdot)$ is a loss function between two vectors. In such f, features are extracted on the pre-determined K_L-th layer ($0 < K_L < K + 1$). The architecture of AE is typically in a symmetric shape with the same number of layers and the same number neurons thereof for the encoder and decoder in f. Then, as shown in Figure 4.1, a VM model is built to predict the water quality y with the features extracted by the AE, $\mathbf{v} = \mathbf{a}^{(K_L)}$ and a predictive model h as

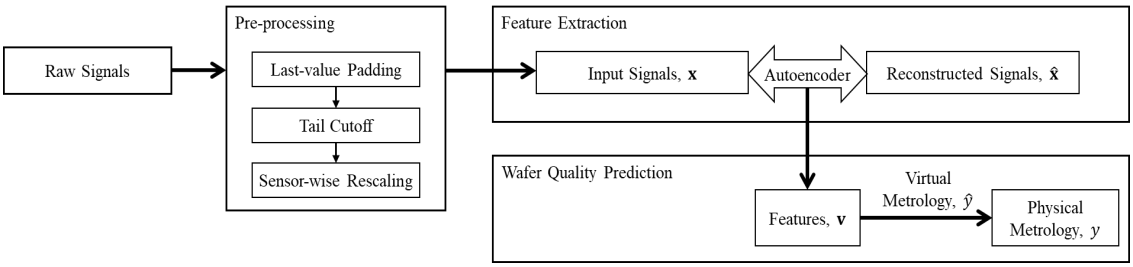


Figure 4.1 Flowchart of VM modeling with AE-based feature extraction.

$$y = h(\mathbf{v}) + \epsilon. \tag{4.2}$$

where ϵ is a prediction error.

Given a dataset of N instances $\{\mathbf{x}_i, y_i\}_{i=1}^{N}$ where $\mathbf{x}_i$ is the input vector of signals from the i-th processing wafer and y_i is the output value of processed the i-th wafer, respectively, the AE model f is trained by minimizing the objective function (Goodfellow et al., 2016) as

$$\mathcal{J}(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} \left\{ \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\mathbf{x}_i, \hat{\mathbf{x}}_i) \; + \mathcal{R}(\mathbf{x}; \boldsymbol{\theta}) \right\} \tag{4.3}$$

where $\hat{\mathbf{x}}_i = f(\mathbf{x}_i; \boldsymbol{\theta})$ is the ith input signals reconstructed by f, and and $\mathcal{R}(\cdot)$ is a regularizer. For continuous outputs, the squared error loss is employed as $\mathcal{L}(\mathbf{z}, \mathbf{z}') = \|\mathbf{z} - \mathbf{z}'\|_2^2$ here $\|\cdot\|_2$ is an ℓ_2 vector norm.

Considering the high dimensionality of multiple sensor data, for efficient model training, we employ the weight decay method with the ℓ_1 regularization on weights as

$$\mathcal{R}_{L1}(\boldsymbol{\theta}) = \lambda_{L1} \sum_{k=1}^{K+1} \left\| \mathbf{W}^{(k)} \right\|_1 \tag{4.4}$$

where λ_{L1} is a non-negative constant for the ℓ_1 regularization.

In VM modeling, diverse types of sensors, such for pressure, temperature, and voltage in chamber, are considered in order to derive critical information about in-process wafers from process equipment. Accordingly, different shapes of signals are collected from those sensors (Park et al., 2014). Unsurprisingly, stable regions are often manifested in parts of the signals as a result that the process is controlled to follow the predefined wafer fabrication recipe at the respective subprocesses. However, the signals reconstructed by AE turn into fluctuating shapes coarsely reflecting the nature of the original data. Thus, to preserve such characteristics of multi-step process signals by suppressing the fluctuation, we propose the fusion regularization on the reconstructed signals as $\mathcal{R}_F(\mathbf{x}_i; \boldsymbol{\theta}) = \lambda_F \left\| \hat{\mathbf{x}}_{i,-1} - \hat{\mathbf{x}}_{i,-p} \right\|_1$ where $\hat{\mathbf{x}}_{i,-a}$ is the vector that exclude the a-th element in $\hat{\mathbf{x}}_i$.

Furthermore, the signals over a whole fabrication process consist of several heterogeneous regions as the wafer is processed along with fabrication recipes for the respective sub-processes. That is, for each of the sub-processes, the process equipment is differently set up, which results in the significant difference between the signals of the sub-processes. The direct application of the fusion regularization, however, overlooks such data characteristic suppressing the difference between the signals of two consequent sub-processes. Thus, to alleviate such indiscriminate penalization from the fusion regularization, we propose the clipping fusion regularization as

$$\mathcal{R}_{CF}(\mathbf{x}_i;\boldsymbol{\theta}) = \lambda_{CF}\min\left(\delta, \left\|\hat{\mathbf{x}}_{i,-1} - \hat{\mathbf{x}}_{i,-p}\right\|_1\right) \tag{4.5}$$

where λ_{CF} is a non-negative constant and the maximum threshold δ is a non-negative constant. Restricting the maximal penalty from the excessive difference between consequent sub-processes, the clipping fusion regularization facilitates to preserve the advantage of the fusion regularization.

The proposed AE model is trained by minimizing the objective function in (4.3) with the regularizations in (4.4) and (4.5) as:

$$\mathcal{R}(\mathbf{x};\boldsymbol{\theta}) = \mathcal{R}_{L1}(\boldsymbol{\theta}) + \sum_{i=1}^{N}\mathcal{R}_{CF}(\mathbf{x}_i;\boldsymbol{\theta}). \tag{4.6}$$

4.4 Experiment

We evaluate the proposed model using a real-life dataset from a plasma etching process in a semiconductor manufacturing industry. The etching process for wafer fabrication is to remove complex surface films layered on a silicon wafer by plasmas such as fluorine and chlorine plasmas. The information about each in-process wafer is monitored in a chamber by equipment sensors that trace the process status such about gas flow, power, pressure, and temperature. Then, critical dimensions (CD) of wafer quality characteristics, such as the etching rate from the thickness of the remaining film of the material of interest on the process wafer, are measured after completion of the process for the wafer.

For this experiment, the data in total were collected from 298 wafers. For VM modeling as illustrated in Figure 4.2, we employed the signals from five sensors on the

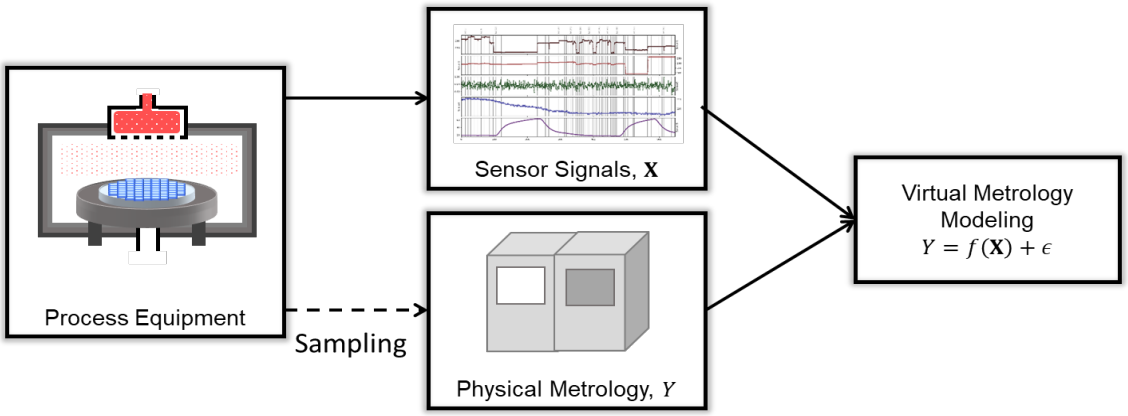
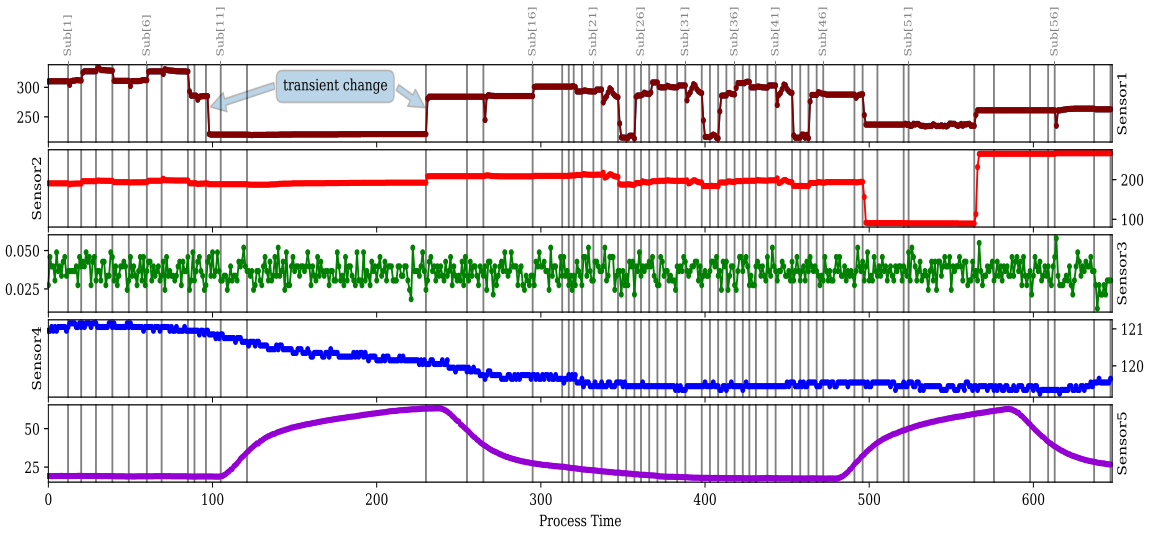


Figure 4.2 VM modeling with sensor signals as inputs and physical measurements as outputs.

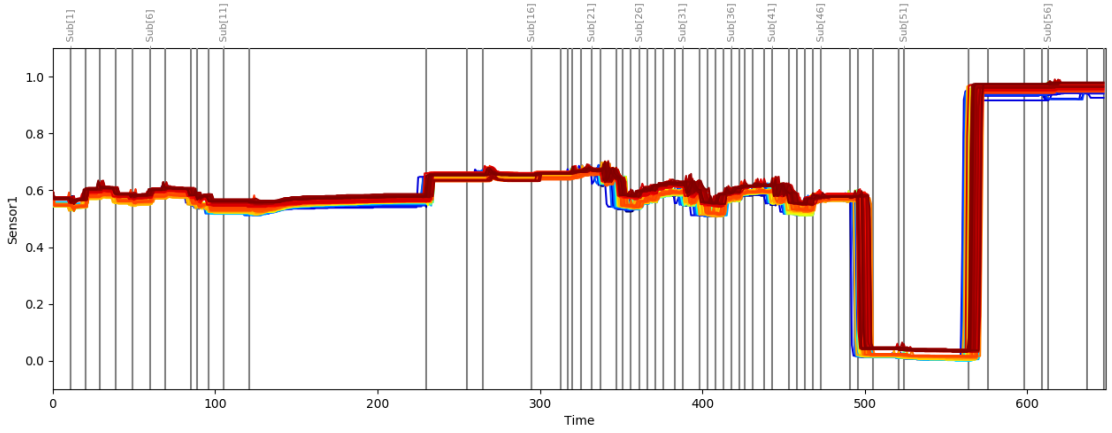
process equipment at the etching process as model input and a CD of wafers as the output. The heterogeneous signals from the five different sensors were recorded for each wafer while the wafer is processed as shown in Figure 4.3(a). The signals for different wafers varied as shown in Figure 4.3(b) due to the variation of the processing time at each different subprocesses. In Figure 4.3, the grey lines indicate the completion time of the subprocesses in the process of a chosen wafer, and the completion time of the first of every five subprocesses is labeled on the top of the plots.

For a comparative experiment, the following approaches are considered to obtain features from the raw signals as conventional methods and the proposed method:

1) *RAW*: the raw signal data;
2) *STEP:* 1740 features (6×5×58) from the six statistics of the signal for each subprocess from each sensor;



(a)



(b)

Figure 4.3 (a) The sensor signals about an observed wafer and transient changes on the signals. (b) The signals at Sensor 2 from 200 instances.

3) *PCA:* 30 features from principal components of the signals for the whole process in PCA;

4) *AE:* 30 features from the activations on the hidden layer in the middle of the *AE* with no regularization;

5) *AE-L1:* 30 features from *AE* whose individual weights are regularized with the ℓ_1 norm;
6) *AE-FL:* 30 features from *AE-L1* whose reconstructed inputs are regularized with the fusion norm;
7) *AE-CFL:* 30 features from *AE-L1* whose reconstructed inputs are regularized with the fusion norm clipping the panelized values.

The AE models in this experiment are structured with hyper-tangent activation functions on the layers, and the architecture of the AEs is identically employed as shown in Figure 4.4 where the architecture was decided based on the network complexity in which *AE* can reconstruct the given signals sufficiently as the reconstruction errors in different architectures are shown in Appendix C.

Each instance, albeit for the same type of wafer products, was collected from different lots, and its processing time varied where the numbers of the regularly observed time points are from 644 to 654. For the implementation of the AE models, we preprocessed the signals, as seen in Figure 4.1, by last-value padding for all the signals shorter than the maximal length (i.e., 654), and then, the signals were cut off to 648 time points assuming no loss of critical information from the signals with the length. Finally, all the signals of each sensor were rescaled in the range from 0.00 to 0.99 to avoid the extreme activations on the reconstruction layer.

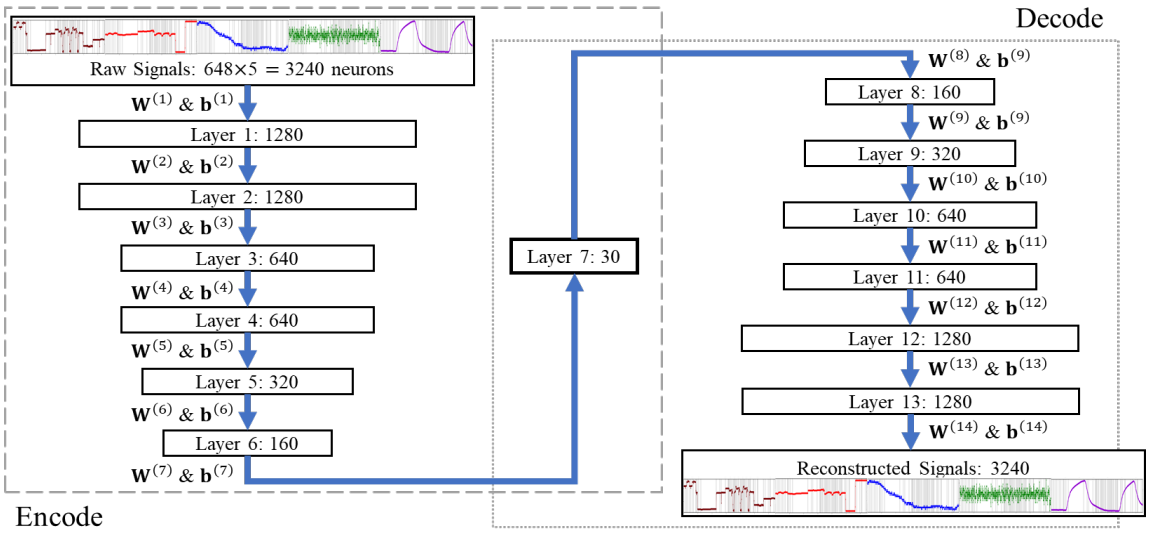


Figure 4.4 Architecture of AEs.

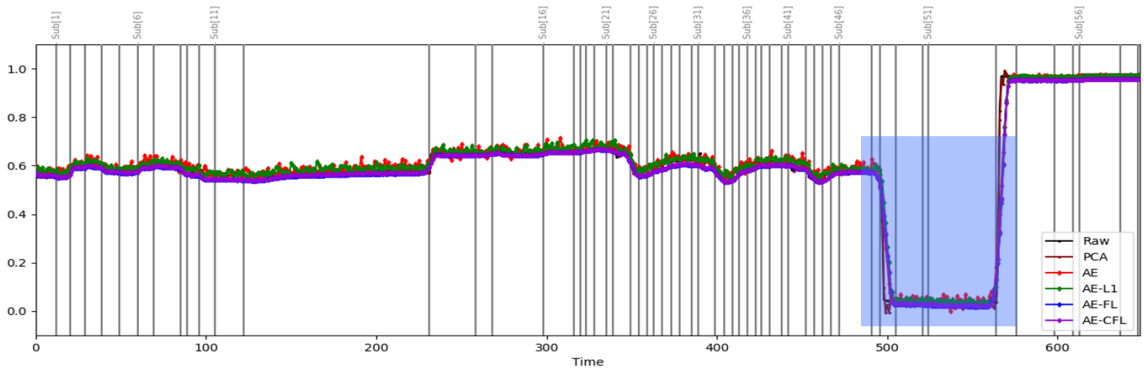
The testing performance of a predictive model with features was evaluated by conducting ten-fold cross-validation. To measure the performance of the prediction models, we compute the R^2 metric in the testing set R^2_{Test} to show the improvement with the features from the null model (i.e., prediction without any input features) as $R^2_{test} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$ where $\bar{y}$ is the mean of the outputs (i.e., the CDs of wafers) and $\hat{y}_i$ is the predicted ith instance's output.

In the experiment, the deep AE models are trained as follows. The objective function of each model is optimized using the RMSprop (Tieleman and Hinton, 2012) with the ratio of the moving average of squared gradients as 0.9 for preventing the gradient exploding and vanishing by balancing the learning step size, the batch size as 8, and a random initialization. To prevent the model from overfitting during the optimization, the early stopping is applied with 10% of the training instances for validating the model given

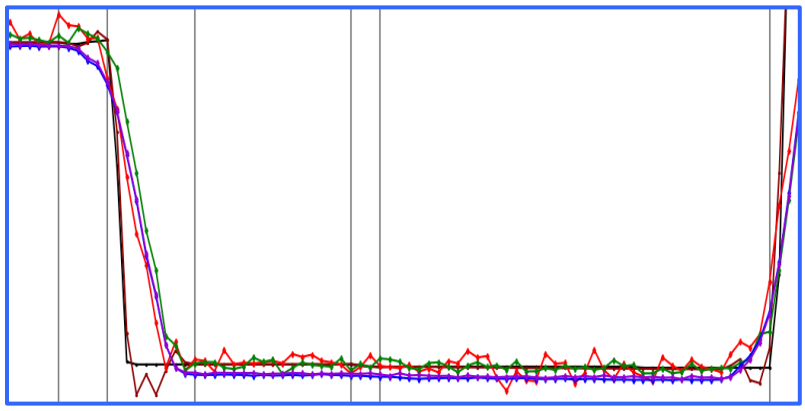
the maximum iteration number as 1000 and the tolerance epochs as 100. Then, the other hyper-parameters are chosen by testing the predictive performance with the features of the *AE* from the combinations of the parameters in the sets for the initial learning rate $\{10^{-4}, 10^{-3}, 10^{-2}\}$ and the learning rate decay per epoch $\{10^{-3}, 10^{-2}, 10^{-1}\}$. With the chosen hyper-parameters by the *AE*, the *AE-L1* is trained choosing the parameter λ_{L1} from a set of parameters $\{5 \times 10^{-7}, 2 \times 10^{-7}, 10^{-7}, 5 \times 10^{-8}, 2 \times 10^{-8}, 10^{-8}\}$. Similarly, with the chose parameters by the previous models, the *AE-FL* is trained choosing the parameter λ_F from a set of parameters $\{5 \times 10^{-7}, 2 \times 10^{-7}, 10^{-7}, 5 \times 10^{-8}, 2 \times 10^{-8}, 10^{-8}\}$. Finally, with the chosen parameters setting $\lambda_F = \lambda_{CF}$, the *AE-CFL* is trained choosing the clipping threshold parameter δ from a set of parameters $\{0.25, 0.2, 0.1, 0.05, 0.02, 0.01, 0.001\}$.

Figure 4.5 presents the reconstructed signals for an observed wafer based on the comparing models. The existing methods, *PCA*, *AE*, and *AE-L1*, produced the reconstructed signals fluctuating around the original signal, whereas the proposed *AE-FL* and *AE-CFL* produced the stable reconstructed signals.

With the features obtained from the aforementioned methods, the predictive performances are compared using the following models: the LASSO regression (LAS) (Tibshirani, 1996) as a linear model and the support vector machine for regression (SVR) (Smola and Schölkopf, 2004) as a kernel-based model. The parameters of the predictive models are determined from the result of ten-fold cross validation from the sets of parameters: in LAS, $\{2^{-10}, 2^{-9}, \ldots, 2^{5}\}$ for the coefficient penalty; in SVR, combination of $\{2^{-5}, 2^{-4}, \ldots, 2^{5}\}$ for the error penalty and $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$ for the radial basis function kernel.



(a)



(b)

Figure 4.5 (a) The signal reconstructed by the comparing models. (b) The enlargement of the highlighted part in (a).

Table 4.1 presents the experimental results about the prediction with the features obtained by the comparing methods from ten-fold cross validation. The prediction for VM was improved mostly by employing feature extraction methods comparing to the prediction with the raw signals. In particular, the prediction with the features from *AE* was improved by introducing ℓ_1 regularization as seen from the performance of *AE-L1*. Also, the prediction performance decreased by additionally employing the fusion regularization in the *AE-L1*, but it rebounded by clipping the regularization penalty.

Table 4.1 Testing performance of the predictive models with the features extracted by different methods.

	Prediction model	
Method	LAS	SVR
RAW	0.1498	0.3120
STEP	0.3079	0.2918
PCA	0.3161	0.3216
AE	0.3070	0.3178
AE-L1	0.3180	0.3238
AE-FL	**0.3214**	0.2920
AE-CFL	0.3177	**0.3318**

4.5 Conclusion

We proposed a deep autoencoder with clipping fusion regularization for feature extraction from multi-step process signals. By clipping the penalizing values from fusion regularization for the process signals, the proposed model reduced the penalties from the heterogeneous subprocess signals.

We conducted a comparative experiment for the predictive performance for virtual metrology with the features extracted by the proposed model and the features of the existing methods. The results demonstrated that the prediction with the features extracted by the proposed model outperformed the prediction with the others where the limitation of the fusion regularization for multi-step process signals was overcome.

In future work, it would be of interest to consider feature extraction with regularization on multi-step process signals classifying the different types of transient changes into subprocess changes and process noise.

CHAPTER 5

GROUP-EXCLUSIVE GROUP LASSO IN DEEP NEURAL NETWORKS

5.1 Introduction

In recent years, deep learning has been a powerful tool for machine learning problems. Based on high flexibility and complexity, deep neural networks (DNNs) enable to achieve remarkable accuracies in prediction tasks (Schmidhuber, 2015; LeCun et al., 2015), including problems with high dimensional data such as image, video, and sensor signals (Litjens et al., 2017; Zhao et al., 2019; Wang et al., 2019). Such properties of DNNs, however, often 'overpower' a model (Scardapane et al., 2017), which can lead to model overfitting and become unsuitable for devices that low-power devices (Sainath et al., 2013). Accordingly, researchers have proposed techniques to efficiently train DNNs with numerous layers and parameters therefrom, for example, considering delicate architectures (He et al., 2016; Xie et al., 2017) and regularization such as parameter penalties and dropout (Srivastava et al., 2014).

Regularization is a strategy widely employed to improve the prediction performance of machine learning models, including DNNs. A common approach to regularization is sparse modeling based on parameter penalties that limit the capacity of models, and structural sparsity induced by such regularization is a key element for better generalization to unseen data with a finite training set or in an imperfect optimization (Goodfellow et al., 2016). In these regards, regularization in DNN models has been employed for modeling with high dimensional data where high dimensionality of data often

causes excessive numbers of model parameters and the improper results in model training like overfitting therefrom.

Sparse modeling has been considered for feature learning with diverse forms of regularization. In particular, the relations in features are incorporated in sparse modeling. Group lasso (Yuan and Lin, 2006) and its variants (Wang and Leng, 2008; Simon and Tibshirani, 2012; Simon et al., 2013) enforce the sparsity of groups at an inter-group level as variables in a group compete with variables in the other groups to survive, and such group-level sparsity has drawn attention for the sparsity of structured features in machine learning (Hastie et al., 2015) including DNN (Wen et al., 2016; Scardapane et al., 2017; Yoon and Hwang, 2017). Exclusive lasso (Zhou et al., 2010) enforces the sparsity of variables at an intra-group level as an exclusive variable (or a few) is selected among the covarying variables in the same group. Exclusive group lasso (Kong et al., 2014) enforces the sparsity of variables at inter-group and intra-group levels as each group that consists of covarying variables compete with each other. However, these methods aim at group-level sparsity, assuming the orthogonality among groups (Simon and Tibshirani, 2012), or focusing on feature-level sparsity (Simon et al., 2013) and their exclusivity (Kong et al., 2014; Kong et al., 2016) at an intra-group level. Thus, the existing methods penalize the coefficients without considering the similarity between groups, and no study considers group-level exclusivity for groups (and their features) that are similar with each other at an inter-group level. That is, a group may consist of the features correlated to the features in the other group, and the features in both groups are significant to the prediction of an output.

On the other hand, some studies that incorporate group-level relations: A study considers the smoothness between the coefficients of neighboring groups in regression (Liu

et al., 2012) and the others rely on lower-dimensional representations of feature groups (Lin et al., 2013; Yan et al., 2011; Zhang et al., 2012). These, however, require models to perform prediction using the information from all the available groups.

In this study, we propose a new regularization for higher group-level sparsity to penalize active groups that are similar with each other and develop a deep neural network using the proposed regularization for automatic exclusive feature group selection. Based on the group lasso (Yuan and Lin, 2006), we incorporate the correlations between feature groups in the formulation introducing group-exclusive group lasso (GGL). The proposed regularization aims to achieve higher group-level sparsity by discouraging a model from employing similar groups, maintaining competitive prediction performance.

The rest of this chapter is organized as follows. In Section 5.2, we briefly review DNN models and relevant studies. In Section 5.3, we describe the proposed feature extraction model. In Section 5.4, we present experimental results using synthetic datasets and a case study. Finally, in Section 5.5, we conclude this chapter with future research.

5.2 Related Work

Consider a general fully connected feedforward neural network f of K hidden layers with M_k neurons on the k-th layer for $k = 1, \ldots, K$, respectively for the prediction of target $T \in \mathbb{R}$ using input $\mathbf{x} \in \mathbb{R}^{M_0}$. Let $\mathbf{x}$ be on the 0-th layer and the predicted target $\hat{t} = f(\mathbf{x}; \boldsymbol{\theta})$ on the $(K+1)$-th layer (i.e., $M_{k+1} = 1$) where $\boldsymbol{\theta}$ is a set of all parameters in f. The output of the k-th layer $\mathbf{z}^{(k)} \in \mathbb{R}^{M_k}$ for $k = 1, \ldots, K+1$ is given by

$$\mathbf{y}^{(k)} = \mathbf{b}^{(k)} + \mathbf{W}^{(k)}\mathbf{z}^{(k-1)} \tag{5.1}$$

$$\mathbf{z}^{(k)} = \sigma_{(k)}\left(\mathbf{y}^{(k)}\right) \tag{5.2}$$

where $\sigma_{(k)}$ is an element-wise activation function, $\mathbf{b}^{(k)} \in \mathbb{R}^{M_k}$ is a bias vector, and $\mathbf{W}^{(k)} \in \mathbb{R}^{M_k \times M_{(k-1)}}$ is a weight parameter matrix on the k-th layer for $k = 1, \ldots, K$.

Given a dataset $(\mathbf{X}, \mathbf{t})$ of N instances where $\mathbf{X} \in \mathbb{R}^{M_0 \times N}$ and $\mathbf{t} \in \mathbb{R}^{N}$ are the input matrix and the target vector, respectively, a neural network f is trained through minimizing the cost as follows:

$$\mathcal{I}(\boldsymbol{\theta}) = \mathcal{L}(\mathbf{t}, \hat{\mathbf{t}}) + \mathcal{R}(\boldsymbol{\theta}) \tag{5.3}$$

where $\mathcal{L}(\mathbf{a}, \mathbf{a}')$ is a loss function between two vectors $\mathbf{a}$ and $\mathbf{a}'$, $\mathcal{R}(\cdot)$ is a regularizer, and $\hat{\mathbf{t}} = f(\mathbf{X}; \boldsymbol{\theta})$ is a vector of N predicted instances. To solve the problem in (5.3), a widely adopted approach is the backpropagation (Rumelhart et al., 1985). The backpropagation is based on updating parameters through the gradients of weight units in a neural network model. The gradients of $w_{pq}^{(k)}$ that is the (p, q)-th element in $\mathbf{W}^{(k)}$ is obtained using the chain rule as

$$\frac{\partial \mathcal{I}(\boldsymbol{\theta})}{\partial w_{pq}^{(k)}} = \frac{\partial \mathcal{I}(\boldsymbol{\theta})}{\partial z_p^{(k)}} \frac{\partial z_p^{(k)}}{\partial y_p^{(k)}} \frac{\partial y_p^{(k)}}{\partial w_{pq}} \tag{5.4}$$

where $z_p^{(k)}$ is the p-th element of $\mathbf{z}^{(k)}$, and $y_p^{(k)}$ is the p-th element of $\mathbf{z}^{(k)}$. The gradients computed in (5.4) are used for iteratively updating the parameters as

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - \Delta\boldsymbol{\theta}(t) \tag{5.5}$$

where t is the index of iterations, $\boldsymbol{\theta}(t)$ is the parameters in $\boldsymbol{\theta}$ at the t-th iteration, and $\Delta\boldsymbol{\theta} = \eta\partial\mathcal{J}(\boldsymbol{\theta})/\partial\boldsymbol{\theta}$ with a learning rate η. Furthermore, there are many methods proposed for effective model training such as the Adam algorithm (Kingma and Ba, 2014) as an adaptive gradient-based optimization algorithm and batch normalization (Ioffe and Szegedy, 2015).

DNN results in outstanding performance if the network were successfully trained. As noted in (Schmidhuber, 2015), the backpropagation, however, encounters some critical issues that hinder efficient training a DNN model. One of the issues is model overfitting due to a large number of parameters. To alleviate the issue from model overfitting, sparse modeling with regularization has been considered, and diverse approaches has been proposed considering different levels of sparsity in DNN models.

Let us consider the regularization on the weight parameters connecting the $(k-1)$-th layer to the k-th layer for generality in the following description.

For the sparse modeling of DNN, an approach is to employ an element-level regularizer as the sparsity is addressed based on the number of active parameters. Popular methods are the ℓ_2 regularization (also called weight decay) on individual weight parameters such that, for the weight parameters on the k-th layer,

$$\mathcal{R}_{\ell_2}^{(k)}(\boldsymbol{\theta}) = \sum_{p,q}\left(w_{pq}^{(k)}\right)^2. \tag{5.6}$$

Similarly, the other popular method is the ℓ_1 (also called lasso) regularization such that

$$\mathcal{R}_{\ell_1}^{(k)}(\boldsymbol{\theta}) = \sum_{p,q}\left|w_{p,q}^{(k)}\right|. \tag{5.7}$$

Besides, the Kullback-Leibler divergence between the average of the activations of the instances on each hidden layer and the given target activation level is used to control the

sparsity level of a DNN model (Ranzato et al., 2006). Furthermore, the regularization on the Jacobian matrix of hidden features is imposed for invariant/robust results induced by suppressing the small variation from the features that have the low values of the first partial derivatives (Rifai et al., 2011).

On the other hand, higher-level sparsity of DNN models has been addressed by grouping parameters for regularization. Based on the number of active neurons, the neuron-level sparsity is considered by imposing the regularization on all the outward weight parameters associated with each neuron (Scardapane et al., 2017) such that, on the k-th layer,

$$\mathcal{R}_N^{(k)} = \sum_{h=1}^{M_{k-1}} c_h^{(k)} \left\| \mathbf{w}_h^{(k)} \right\|_2 \tag{5.8}$$

where $c_h^{(k)}$ is a coefficient for the weight parameter group size, $\|\mathbf{a}\|_2 = \left(\sum_{j=1}^{M} a_j\right)^{1/2}$ is a ℓ_2 norm of a vector $\mathbf{a} = [a_1, a_2, \ldots, a_M]^{\mathrm{T}}$, and $\mathbf{w}_h^{(k)} \in \mathbb{R}^{M_k}$ is the h-th column of $\mathbf{W}^{(k)}$ as the outward weights from the h-th input unit.

Other studies address the sparsity of neuron groups as input or feature groups. Similar to the neuron-level regularization in (5.8), the weight parameter groups are penalized as

$$\mathcal{R}_G^{(k)} = \sum_{r=1}^{g_{k-1}} c_r^{(k)} \left\| \boldsymbol{\omega}_r^{(k)} \right\|_2 \tag{5.9}$$

where g_{k-1} is the total number of predetermined groups on the $(k-1)$-th layer and $\boldsymbol{\omega}_r^{(k)}$ is the vectorized $\mathbf{W}_r^{(k)}$ that consists of $\mathbf{w}_h^{(k)}$ for all h in group r. For the feature-group-level

sparsity, the regularizations in (5.9) are employed on the groups of the neurons on the first hidden layer as the grouped input features (Yu and Lin, 2011), and on the features individually extracted from submodels in a multimodal DNN (Zhao et al., 2015).

Furthermore, recent studies (Zhu et al., 2018; Yoon and Hwang, 2017) employ the exclusive group regularization (EG) firstly introduced in (Zhou et al., 2010) for multi-task learning problems. EG is to penalize for intra-group sparsity with ℓ_1 norm and inter-group sparsity with ℓ_2 norm as

$$\mathcal{R}_{EG}^{(k)} = \sum_{r=1}^{g_{k-1}} c_r^{(k)} \left\| \boldsymbol{\omega}_r^{(k)} \right\|_1^2 . \tag{5.10}$$

5.3 Proposed Model

5.3.1 Group-exclusive group lasso regularization

Let $\mathcal{G}^{(k)}$ be an index set of all the features on the k-th layer and let $\mathcal{G}_r^{(k)}$ be the r-th feature group set as a subset of $\mathcal{G}^{(k)}$ for r = 1, ..., g_k where $\mathcal{G}^{(k)} = \bigcup_{r=1}^{g_k} \mathcal{G}_r^{(k)}$ and $\mathcal{G}_r^{(k)} \cap \mathcal{G}_s^{(k)} = \emptyset$ for r and s = 1, ..., g_k and $r \neq s$. Denoting m_{rk} to the number of neurons in group r on the k-th layer where $1 \leq m_{rk} \leq M_k$ and $M_k = \sum_{r=1}^{g_k} m_{rk}$, in this work, we consider an equal group size on the k-th layer as $m_{rk} = m_k$ for r = 1, ..., g_k.

Suppose that the information about g_0 groups for M_0 features in input $\mathbf{x}$ (i.e., $\mathbf{x} = \mathbf{z}^{(0)}$) is given as $\mathcal{G}^{(0)}$. Let $\mathbf{x}_r$ be a m_0-dimensional subvector for the input features in $\mathcal{G}_r^{(0)}$.

The linear operation in (5.1) on the first hidden layer can be written with respect to the groups as

$$\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)} = \mathbf{b}^{(1)} + \sum_{r=1}^{g_0} \mathbf{W}_r^{(1)}\mathbf{x}_r \tag{5.11}$$

where $\mathbf{W}_r^{(1)} \in \mathbb{R}^{M_1 \times m_0}$ is a matrix whose columns are $\mathbf{w}_h^{(1)}$ for $h \in \mathcal{G}_r^{(0)}$. For a group sparsity, penalizing the weight parameters in $\mathbf{W}_r^{(1)}$ for groups $\mathcal{G}_r^{(0)}$ for r = 1, ..., g_0 with a rescaling factor $c_r^{(1)} = \sqrt{M_1 \cdot m_0}$ using the $\ell_{2,1}$ norm is equivalent to the regularization for $k = 1$ in (5.9). Given a dataset $(\mathbf{X}, \mathbf{t})$, the input matrix $\mathbf{X}$ are divided as $\mathbf{X} = \left[\mathbf{X}_1^{\mathrm{T}}, \mathbf{X}_2^{\mathrm{T}}, \ldots, \mathbf{X}_{g_0}^{\mathrm{T}}\right]^{\mathrm{T}}$ accordingly.

To avoid the redundant information from different groups, we first measure the association ρ_{rs} between the r- and s-th input feature groups. In this work, we adopt the $\mathrm{R_V}$ coefficient (Escoufier, 1973) to measure the linear association between the feature groups over all the instances, given $\mathbf{X}_r, \mathbf{X}_s \in \mathbb{R}^{m_0 \times N}$ are data matrices for the features in the r- and s-th groups as

$$\rho_{rs} = \frac{\mathrm{tr}(\mathbf{X}_r\mathbf{X}_r^{\mathrm{T}}\mathbf{X}_s\mathbf{X}_s^{\mathrm{T}})}{\sqrt{\mathrm{tr}(\mathbf{X}_r\mathbf{X}_r^{\mathrm{T}})^2\mathrm{tr}(\mathbf{X}_s\mathbf{X}_s^{\mathrm{T}})^2}} \tag{5.12}$$

where $\mathrm{tr}(\mathbf{X}_r\mathbf{X}_r^{\mathrm{T}})$ is a trace of $\mathbf{X}_r\mathbf{X}_r^{\mathrm{T}}$. The association of two groups may be defined with the other measures as summarized in (Josse and Holmes, 2016).

We now introduce a new regularization for the exclusivity of feature groups. For simplicity, we consider imposing the proposed regularization on the weights of the first layer and drop the superscripts of group penalty terms as layer indexes in the following

description for the proposed regularization, accordingly. We define the group-exclusive group penalty for the r- and s-th groups that are linearly associated with each other as

$$\mathcal{R}_{GG}(r,s) \triangleq \delta_{rs}(\|\boldsymbol{\omega}_r\|_2 \cdot \|\boldsymbol{\omega}_s\|_2)^{\frac{1}{2}} \tag{5.13}$$

where $\delta_{rs} = \rho_{rs}(c_r c_s)^{1/2}$. The penalty from (5.13) will go to zero if two feature groups have no association and/or if any of the two groups or both were excluded. Otherwise, the penalty will be imposed if the two feature groups are being simultaneously selected in a DNN model. It would be worth noting that $\mathcal{R}_{GG}(r,s)$ becomes equivalent to $\mathcal{R}_G(s)$ when $r = s$. Then, we define the group-exclusive group regularization (GG):

$$\begin{aligned}\mathcal{R}_{GG}(\boldsymbol{\theta};\mathbf{X}) &= \sum_{s\in\mathcal{G}}\sum_{r\in\mathcal{G}} \mathcal{R}_{GG}(r,s) \\ &= \sum_{s\in\mathcal{G}} \mathcal{R}_{GG}(s,s) + \sum_{s\in\mathcal{G}}\sum_{r\in\mathcal{G}_{(-s)}} \mathcal{R}_{GG}(r,s)\end{aligned} \tag{5.14}$$

where $\mathcal{G}_{(-s)} = \mathcal{G} \setminus \{s\}$.

Also, as noted in Section 5.2, the element-wise regularization on the weight parameters in a DNN model is a great tool to have the model sparse and to discourage model overfitting. Thus, we employ the lasso regularizer in (5.7) for the element-wise sparsity

$$\mathcal{R}_{EL}(\boldsymbol{\theta}) = \sum_{k=1}^{K} \mathcal{R}_{\ell_2}^{(k)}(\boldsymbol{\theta}). \tag{5.15}$$

Finally, with the regularizers in (5.14) and (5.15), we define the group-exclusive group lasso regularization (GGL) as follows:

$$\mathcal{R}_{GGL}(\mathbf{\theta};\mathbf{X}) \triangleq \lambda_{GG}\mathcal{R}_{GG}(\mathbf{\theta};\mathbf{X}) + \lambda_{EL}\mathcal{R}_{EL}(\mathbf{\theta}) \tag{5.16}$$

where λ_{GG} and λ_{EL} are nonnegative regularization coefficients.

5.3.2 Formulation and model training

Given a dataset $(\mathbf{X}, \mathbf{t})$ and input feature groups, the proposed DNN model for a group-exclusive group-level sparsity is formulated in (5.3) with the term in (5.16) as

$$\mathcal{J}(\mathbf{\theta}) = \arg\min_{\mathbf{\theta}} \mathcal{L}\big(\mathbf{t}, f(\mathbf{X};\mathbf{\theta})\big) + \mathcal{R}_{GGL}(\mathbf{\theta};\mathbf{X}). \tag{5.17}$$

For model training, we use a popular gradient-based optimization algorithm, the Adam (Kingma and Ba, 2014), to train the proposed model. As noted in Section 5.2, the model is trained by iteratively updating the parameters in $\mathbf{\theta}$ with their gradients. In particular, we need to compute the gradients of $\mathcal{R}_{GG}(\mathbf{\theta};\mathbf{X})$ in (5.14). The GG regularization term is convex, and its gradient is not defined when $\boldsymbol{\omega}_r = \mathbf{0}$ for any r. We, therefore, use the subgradient of the GG regularization term with respect to $\boldsymbol{\omega}_s$:

$$\frac{\partial \mathcal{R}_{GG}(\mathbf{\theta};\mathbf{X})}{\partial \boldsymbol{\omega}_s} = \begin{cases} \dfrac{\boldsymbol{\omega}_s}{\|\boldsymbol{\omega}_s\|_2} \displaystyle\sum_{r\in\mathcal{G}} \frac{\delta_{rs}}{2} \left(\frac{\|\boldsymbol{\omega}_r\|_2}{\|\boldsymbol{\omega}_s\|_2}\right)^{\frac{1}{2}} & \text{if } \boldsymbol{\omega}_s \neq \mathbf{0} \\ \mathbf{v} \in \{\mathbf{v} : \|\mathbf{v}\|_2 \leq \delta_{ss}\} & \text{if } \boldsymbol{\omega}_s = \mathbf{0} \end{cases}, \tag{5.18}$$

and the derivation is in Appendix D.

5.4 Experiments

We evaluate the proposed method on different datasets from artificial examples and a real-life case. We first illustrate general behavior of GGL in a prediction model with toy data and then test its performance with real-life data from the virtual metrology in semiconductor wafer fabrication.

5.4.1 Experimental setup

In this work, we performed all experiments with feedforward neural networks. We set up a network structure with different numbers of hidden layers for each experiment according to its complexity and an output layer with a single neuron for regression. For hidden layers, we employed an element-wise hyperbolic tangent activation $\sigma_{\tanh}(a) = \frac{e^{2a}-1}{e^{2a}+1}$. On each layer, the weight parameters were initialized with random values from a uniform distribution and the bias parameters were set to be zero following the method in (Glorot and Bengio, 2010). Randomly selecting 10% of training data as a validation set, a network was trained using the Adam algorithm (Kingma and Ba, 2014) with the learning rate as 0.001, the maximum number of training epochs as 10000, and the number of patience epochs for the early stopping criterion as 20.

Given a network structure in each experiment, we compared the proposed DNN with GGL (*DNN-GGL*) in (5.16) with the following models: a DNN with the neuron-wise group lasso (*DNN-NL*) from the terms in (5.8) and (5.15) as $\mathcal{R}_{NL} = \lambda_N \mathcal{R}_N^{(1)}(\boldsymbol{\theta}; \mathbf{X}) + \lambda_{EL}\mathcal{R}_{EL}(\boldsymbol{\theta})$; a DNN with the feature-group-wise group wise lasso (*DNN-GL*) from the terms in (5.9) and (5.15) as $\mathcal{R}_{GL} = \lambda_G \mathcal{R}_G^{(1)}(\boldsymbol{\theta}; \mathbf{X}) + \lambda_{EL}\mathcal{R}_{EL}(\boldsymbol{\theta})$; and a network with the

exclusive-neuron-group-wise group lasso (*DNN-EGL*) from the terms in (5.10) and (5.15) as $\mathcal{R}_{EGL} = \lambda_{EG}\mathcal{R}_{EG}^{(1)}(\boldsymbol{\theta}; \mathbf{X}) + \lambda_{EL}\mathcal{R}_{EL}(\boldsymbol{\theta})$.

Given a prediction error vector $\mathbf{e} = [e_1, \ldots, e_N]^T$, we evaluate the prediction performance using mean square error (MSE), $MSE(\mathbf{e}) = \frac{1}{N}\sum_{i=1}^{N} e_i^2$, and mean absolute error (MAE), $MAE(\mathbf{e}) = \frac{1}{N}\sum_{i=1}^{N}|e_i|$, and maximum absolute error (MAXAE) as $MAXAE = \max\{|e_1|, |e_2|, \ldots, |e_{N_{te}}|\}$. We measure the model sparsity induced by the regularizations by counting the number of active feature groups and the number of active neurons on the first hidden layer as $\sum_{s=1}^{g_0}\mathbb{I}\{\sum_k \mathbb{I}\{|\omega_{ks}| > 10^{-3}\} \neq 0\}$ and $\sum_{j=1}^{M_0}\mathbb{I}\{\sum_k \mathbb{I}\{|w_{kj}| > 10^{-3}\} \neq 0\}$, respectively, where $\mathbb{I}(\cdot)$ is an indicator function and ω_{ks} is the k-th element of $\boldsymbol{\omega}_s$.

5.4.2 Synthetic data

We generated $n = 200$ sample instances for $L = 30$ feature groups with $m = 10$. As input data, we independently sampled instances of each group r, $\mathbf{X}_r \in \mathbb{R}^{m\times N}$, for $r = 1, \ldots, L$, from a central multivariate normal $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_r)$ where $[\boldsymbol{\Sigma}_r]_{ii} = 1.0$ for all i and $[\boldsymbol{\Sigma}_r]_{ij} = 0.1$ for all (i, j) and $i \neq j$. As output data, we sampled the responses for which the features of the first 4 groups are significant as

$$\mathbf{y} = \sum_{r\in\{1,2,3,4\}} \mathbf{X}_r\boldsymbol{\beta}_r + \boldsymbol{\varepsilon}$$

where $\boldsymbol{\beta}_r = [-2, -1, 0, 0, 0, 1, 2, 0, 0, 0]^{\mathrm{T}}$ for $s = 1, \ldots, 4$. Let x_{ijr} be the (i, j)-th element in $\mathbf{X}_r$. Then, we construct an input matrix of 32 groups with $\mathbf{X} = [\mathbf{X}_1', \mathbf{X}_1'', \mathbf{X}_2', \mathbf{X}_2'', \mathbf{X}_3, \ldots, \mathbf{X}_{40}]$ where $[\mathbf{X}_r']_{ij} = x_{ijr}'$ is given by $x_{ijr}' = x_{ijr} + \epsilon_1$ with noises $\epsilon_1 \sim \mathcal{N}(0, 0.1)$ and

$[\mathbf{X}_r'']_{ij} = x_{ijr}''$ is given by $x_{ijr}'' = -x_{ijr} + \epsilon_2$ with $\epsilon_2 \sim \mathcal{N}(0,0.2)$. Figure 5.1 shows the pairwise group similarities in the generated data estimated using the R_V coefficient in (5.12). Unsurprisingly, the R_V coefficient between groups 1 and 2 was high as they were generated from the same original data $\mathbf{X}_1$ as well as the R_V coefficient between groups 3 and 4 from $\mathbf{X}_2$.

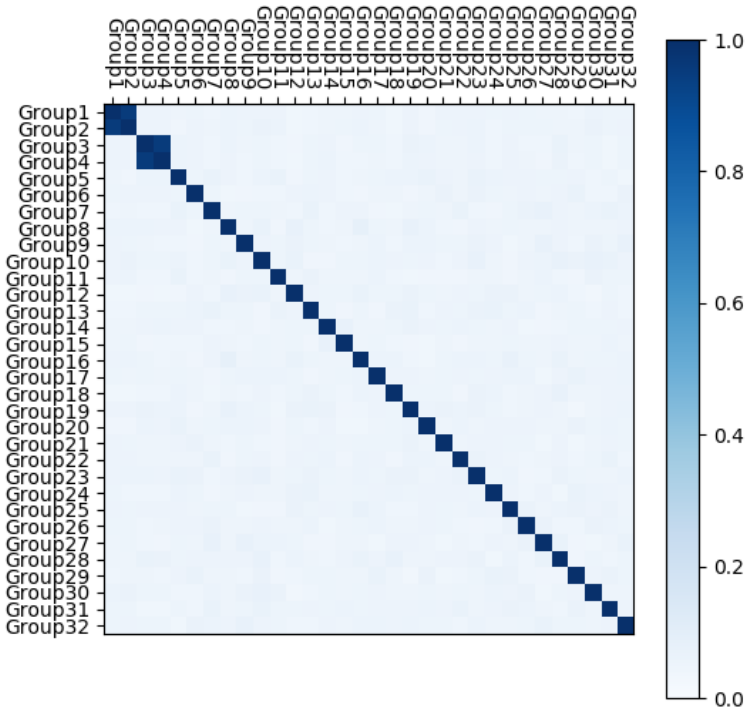


Figure 5.1 Pairwise group similarities of the synthetic data.

We built network models with a hidden layer of 8 neurons and an output layer of 1 neuron with the comparing regularization terms. The model performance was evaluated over a five-fold cross validation. In the training set, the parameters for the group

regularization as $\lambda_N = \lambda_G = \lambda_{EG} = \lambda_{GG} = 10^{-a/4}$ with $a = 0, 1, \ldots, 24$ and for the element-wise regularization $\lambda_{EL} = 10^{-a/4}$ with $a = 0, 1, \ldots, 24$ were chosen based on the validation MSE over three-fold cross validations.

Table 5.1 shows the testing accuracy and the model sparsity in the synthetic data and the pairwise *t*-test results between the proposed *DNN-GGL* and the comparing models. *DNN-GGL* showed the superior accuracy to the other models. Furthermore, the models

Table 5.1 Testing results and the numbers of active neurons and active groups in the synthetic data.

	MSE	MAE	MAXAE	Active neurons	Active groups
DNN-NL	0.0014**	0.0293**	0.0249**	107.27***	29.13***
DNN-GL	0.0018***	0.0335***	0.0285***	131.20***	30.27***
DNN-EGL	0.0164***	0.1024***	0.0874***	145.27***	32.00***
DNN-GGL	**0.0007**	**0.0217**	**0.0185**	**42.13**	**7.47**

* p-value < 0.10; ** p-value < 0.05; *** p-value < 0.01

with the existing regularizations were trained with higher numbers of neurons and groups, and the proposed *DNN-GGL* had the smallest number of active groups that is closest to the true number of groups.

5.4.3 Case study: Sensor selection for virtual metrology in semiconductor manufacturing

We evaluate the proposed method on a real-life case about virtual metrology (VM) in semiconductor manufacturing processes. VM aims to predict the quality of a wafer using data from process and production equipment, without physical metrology on wafers. The measurement of a wafer's quality from VM can be provided immediately after being processed, without an additional operation for gauging the quality. Typically, a number of sensors are employed to obtain salient information about processed wafers, which can facilitate monitoring and controlling the process in real-time, and to improve production

efficiency (Khan et al., 2007; Qin et al., 2006) and reduce process costs by replacing costly physical metrologies and metrology devices and by planning predictive/prospective maintenance based on process health status provided by VM (Chen et al., 2005; Cheng et al., 2011 Kang et al., 2009).

In this case study, we aimed to build a VM model at a plasma etching process for wafer fabrication. The process was monitored by various sensors in/on the chamber that records different physical parameters about the process status such as levels of chemicals, power, gas supply, and temperature, and a critical dimension (CD) as a quality characteristic of the wafer was measured after the completion of the process. During the production of identical products, lots were sampled at random, and the instances were collected from the wafers selected at random in a lot of 25 wafers. For our experiment, 299 instances were collected. The signals from 28 sensors during the processing time T_i were recorded for wafer i. Figure 5.2 shows raw process signals from the first ten sensors for a wafer, some of which provided similar information as the same types of sensors. Besides, the etching was processed through a series of 58 sub-processes where the grey vertical lines in Figure 5.2 indicate the completion time of the subprocesses.

As input variables in VM models, we extracted 3248 features ($2 \times 58 \times 28$) from the raw signals of 28 sensors by calculating two statistics of the signals for each of 58 subprocesses: mean $\bar{x}_{ijk} = \frac{1}{T_{ij}} \sum_{l=1}^{T_{ij}} x_{ijkl}$ and variance $s_{ijk}^2 = \frac{1}{(T_{ij}-1)} \sum_{l=1}^{T_{ij}} \left(x_{ijkl} - \bar{x}_{ijk}\right)^2$ where x_{ijkl} is the value at the l-th time point of the i-th wafer's signals from the j-th subprocess on the k-th sensor and T_{ij} is the processing time for the i-th wafer at the j-th subprocess ($T_i = \sum_j T_{ij}$). We formed 28 feature groups, each of which consists of $2 \times 58 = 116$ features from a sensor. Figure 5.3 shows the similarities of the feature groups and,

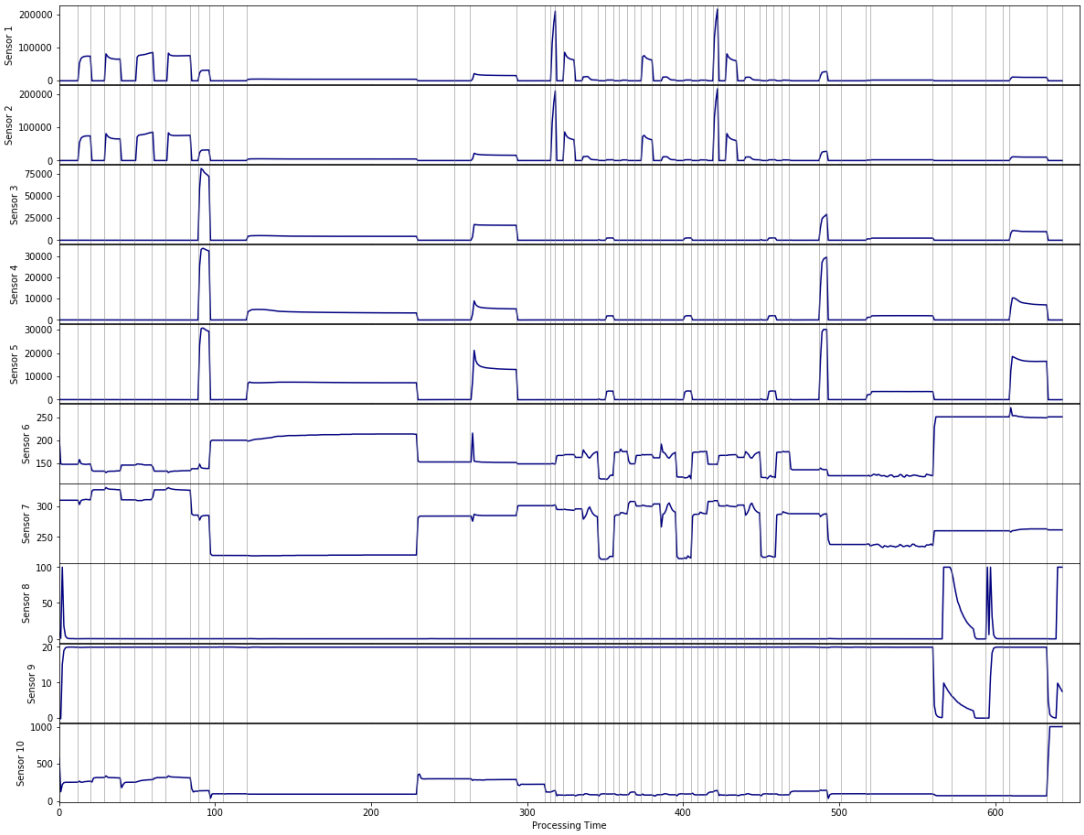


Figure 5.2 Raw signals of a wafer from ten sensors.

unsurprisingly, there were highly correlated feature groups as some of the features were from the sensors of similar signals.

We set up the network architecture of four hidden layers with 64, 32, 16, and 8 neurons. The hyper-parameters for the regularization penalties were explored during this experiment by grid search with $\lambda_{GRP} = \lambda_{GG} = \lambda_{EG} = \lambda_G = 10^{-a}$ for $a = 2, \ldots, 5$ and $\lambda_{EL} = 10^{-b}$ for $b = 2, \ldots, 7$ and were chosen in terms of MSE among the models in which all the groups were not active, considering the use of less number of sensors for VM.

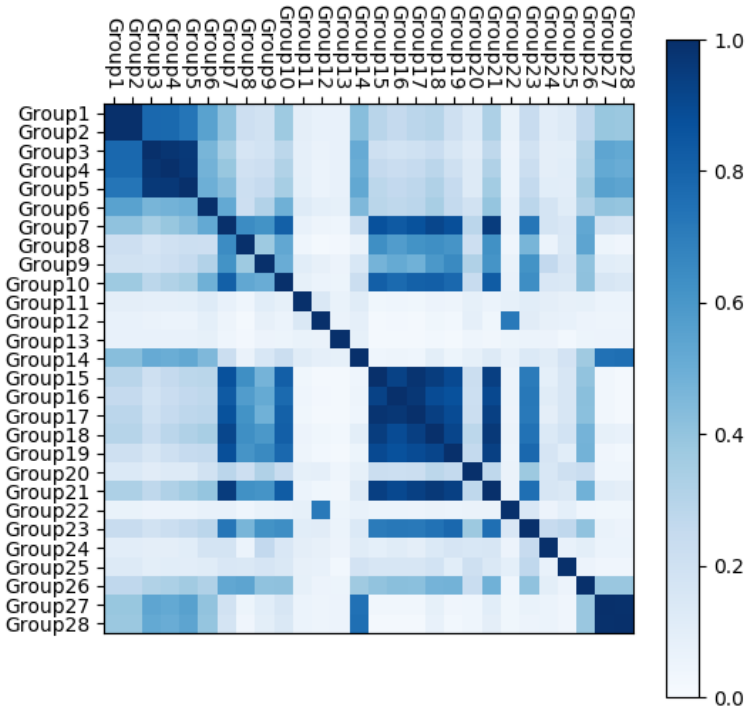


Figure 5.3 Pairwise group similarities of VM data.

Table 5.2 Testing accuracy and the numbers of active neurons and active groups in the case study data.

	MSE	MAE	MAXAE	Active neurons	Active groups
DNN-NL	0.1232	0.2812	0.2349	656.4	26.8
DNN-GL	0.1097	0.2710	0.2390	851.6	23.8
DNN-EGL	0.1146	0.2719	0.2352	547.4	27.8
DNN-GGL	0.1043	0.2609	0.2189	67.8	5.2

Table 5.2 presents the testing result from a five-fold cross-validation. DNN-GGL showed the best prediction performance in terms of three performance measures and the highest sparsity in terms of both the active neurons and active groups. Although *DNN-GL* showed the competitive accuracy, it required much more active groups and input neurons, comparing to *DNN-GGL*. Unsurprisingly, *DNN-NL* and *DNN-EGL* barely achieved group-level sparsity as only few feature groups were inactive.

5.5 Conclusion

We have proposed the group-exclusive group lasso regularization in deep neural networks for group selection. Extending from the group lasso regularization for individual groups, the proposed regularization additionally penalized the active groups that are similar with each other, which leads to the automatic selection of salient feature groups from a solution of higher group sparsity. We developed a deep neural network using the proposed regularization. The experimental results showed its superior group sparsity maintaining its competitive prediction performance.

The proposed method can be extended in several ways. First, we should extend the proposed group-exclusive group lasso to generalized groups of different numbers of features. Second, different similarities/associations at a group level can be considered with the other measures of different properties as reviewed in (Josse and Holmes, 2016). Also, it would be interesting to investigate a method for smoothing the proposed regularization in order to overcome computational obstacles, such as the oscillation of gradients at the origin, caused by the group lasso regularization (Wang et al., 2017). Finally, it is of interest

to explore the proposed regularization for the other types of neural networks, including convolutional neural networks and recurrent neural networks.

CHAPTER 6

CONCLUSION AND FUTURE RESEARCH

6.1 Conclusion

In this dissertation, we proposed sparse machine learning methods for the prediction of wafer qualities in semiconductor manufacturing processes.

In Chapter 2, we proposed the restricted relevance vector machine for incomplete data. The proposed model that restricts its basis in order to prevent the potential loss of the sparsity in its predictive distribution when the missing components were imputed in kernel space. The experimental results using toy and real-life data demonstrated that the proposed method maintained the model sparsity while the imputation of missing components in kernel space improved the prediction performance.

In Chapter 3, we proposed the expected Gaussian kernel with correlated variables for the sparse Bayesian kernel machine regression with incomplete data. Considering the correlation between the variables whose instances are missing, the parameters of the probability distributions are estimated for the computation of the expected Gaussian kernels. The experimental results revealed the proposed method led to superior prediction performance to the existing EGK when the more incomplete instances are considered and when more strong correlation between missing variables is manifested.

In Chapter 4, we proposed a deep learning-based feature extraction using the clipping fusion regularization for multistep process signals. By clipping the penalties from the fusion regularization for the stable signals reconstructed, the proposed model reduces the undesired penalties from the heterogeneous subprocess signals. We conducted a

comparative experiment for the predictive performance for virtual metrology with the features extracted by the proposed model and the features of the existing methods. The results demonstrated that the potential of the predictive performance with the features extracted by the proposed model over the performances by the existing methods.

In Chapter 5, we proposed a new group-exclusive group lasso in a deep neural network for automatic exclusive feature group selection. With feature groups and their correlations, the proposed model aims to avoid the coincident selection of the feature groups that correlated to each other and share the predictive powers to responses. The experimental results from synthetic and real-life data sets showed that the proposed method achieved higher group sparsity with competitive prediction accuracy.

6.2 Future Research

In future work, a meaningful line of research would be further investigation of the restricted relevance vector machine extending to incorporating different data types and solving classification problems. In addition, it would be worth investigating the relevance vector machine with weights from the uncertainties of incomplete instances.

Next, the expected Gaussian kernels for incomplete data can be extended by incorporating different distributions, kernels, and data types. Also, it would be worth extending the expected Gaussian kernels for the other types of machine learning problems such as classification and feature extraction.

Furthermore, future studies with the clipping fusion regularization can be feature extraction on multistep process signals classifying the types of transient changes. Also, it

would be interesting to employ the regularization in the other types of machine learning methods that reconstruct original signals.

Finally, we can extend the group-exclusive group lasso in several approaches. We should generalize the group-exclusive group lasso in terms of group sizes, group similarities, and predictive models. Also, the development of methods for the efficient computation of the group-exclusive lasso can be future works. We believe that more advanced methodologies will bring improvements in this research area.

APPENDIX A Derivation of the predictive distribution of RRVM

To find the predictive distribution of the proposed RRVM in (2.17), we need to calculate the posterior distribution over the coefficient vector $\mathbf{w}_R$ in (2.11) as:

$$p(\mathbf{w}_R|\mathbf{y},\mathbf{X},\mathbf{A}_R,\sigma^2) = \frac{p(\mathbf{y}|\mathbf{X},\mathbf{w}_R,\sigma^2)p(\mathbf{w}_R|\mathbf{A}_R)}{p(\mathbf{y}|\mathbf{X},\mathbf{A}_R,\sigma^2)}$$

$$= (2\pi)^{-\frac{N+1}{2}}|\boldsymbol{\Sigma}|^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}(\mathbf{w}_R-\mathbf{m}_R)^{\mathrm{T}}{\boldsymbol{\Sigma}_R}^{-1}(\mathbf{w}_R-\mathbf{m}_R)\right\}$$

where

$$p(\mathbf{y}|\mathbf{X},\mathbf{w}_R,\sigma^2) = (2\pi\sigma^2)^{-\frac{N}{2}}\exp\left\{-\frac{1}{2\sigma^2}\left\|\mathbf{y}-\widetilde{\boldsymbol{\Phi}}_R\mathbf{w}_R\right\|^2\right\}$$

$$p(\mathbf{w}_R|\mathbf{A}_R) = (2\pi)^{-\frac{L}{2}}|\mathbf{A}_R|^{\frac{1}{2}}\exp\left\{-\frac{1}{2}\mathbf{w}_R^{\mathrm{T}}\mathbf{A}\mathbf{w}_R\right\}$$

$$p(\mathbf{y}|\mathbf{X},\mathbf{A}_R,\sigma^2) = \int (2\pi\sigma)^{-\frac{N}{2}}(2\pi)^{-\frac{L}{2}}|\mathbf{A}_R|^{\frac{1}{2}}\exp\left\{-\frac{1}{2\sigma^2}\left\|\mathbf{y}-\widetilde{\boldsymbol{\Phi}}_R\mathbf{w}_R\right\|^2-\frac{1}{2}\mathbf{w}_R^{\mathrm{T}}\mathbf{A}\mathbf{w}_R\right\}d\mathbf{w}_R$$

$$= (2\pi)^{-\frac{N}{2}}\left|\widetilde{\boldsymbol{\Phi}}_R\mathbf{A}_R^{-1}\widetilde{\boldsymbol{\Phi}}_R^{\mathrm{T}}+\sigma^2\mathbf{I}_N\right|^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}\mathbf{y}^T\left(\widetilde{\boldsymbol{\Phi}}_R\mathbf{A}_R^{-1}\widetilde{\boldsymbol{\Phi}}_R^{\mathrm{T}}+\sigma^2\mathbf{I}_N\right)^{-1}\mathbf{y}\right\}.$$

Then, the predictive distribution for y_{new} given a new instance $\mathbf{x}_{new}$ is obtained as a convolution of two Gaussian distributions:

$$p(y_{new}|\mathbf{x}_{new},\mathbf{y},\mathbf{X},\mathbf{A}_R,\sigma^2) = \int p(y_{new}|\mathbf{x}_{new},\mathbf{w}_R,\sigma^2)p(\mathbf{w}_R|\mathbf{y},\mathbf{X},\mathbf{A}_R,\sigma^2)d\mathbf{w}_R$$

$$\sim \mathcal{N}\left(\mathbf{m}^{\mathrm{T}}\widehat{\boldsymbol{\phi}}_R(\mathbf{x}_{new}),\sigma^2+\widehat{\boldsymbol{\phi}}_R(\mathbf{x}_{new})^{\mathrm{T}}\boldsymbol{\Sigma}_R\widehat{\boldsymbol{\phi}}_R(\mathbf{x}_{new})\right).$$

APPENDIX B Proof of Proposition 1

For the squared distance between two real vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, a Gamma variable ζ_{ij} is approximated from the sum of correlated Gamma variables $\gamma_{ijp} \sim Gamma(k_{ijp}, \theta_{ijp})$ for $p = 1, \ldots, D$ based on the approximation in (Feng et al., 2016). The shape parameter k_{ijp}, which is estimated using $E[\gamma_{ijp}]$ in (3.18) and $Var(\gamma_p)$ in (3.19) from the moments of the missing components in original space, satisfies the condition $k_{ijp} \geq \frac{1}{2}$ if $\sigma_{pp,i} + \sigma_{pp,j} > 0$:

$$
\begin{aligned}
& k_{ijp} = \frac{E[\gamma_{ijp}]^2}{Var(\gamma_{ijp})} \geq \frac{1}{2} \\
& \Leftrightarrow \frac{\left\{(\tilde{x}_{ip} - \tilde{x}_{jp})^2 + \sigma_{pp,i} + \sigma_{pp,j}\right\}^2}{2(\sigma_{pp,i} + \sigma_{pp,j})\left\{\sigma_{pp,i} + \sigma_{pp,j} + 2(\tilde{x}_{ip} - \tilde{x}_{jp})^2\right\}} \geq \frac{1}{2} \\
& \Leftrightarrow (\tilde{x}_{ip} - \tilde{x}_{jp})^4 \geq 0.
\end{aligned}
\tag{B.1}
$$

Similarly, the scale parameter θ_{ijp} satisfies the condition $\theta_{ijp} > 0$ if $\sigma_{pp,i} + \sigma_{pp,j} > 0$:

$$
\begin{aligned}
& \theta_{ijp} = \frac{Var(\gamma_{ijp})}{E[\gamma_{ijp}]} > 0 \\
& \Leftrightarrow \frac{2(\sigma_{pp,i} + \sigma_{pp,j})\left\{\sigma_{pp,i} + \sigma_{pp,j} + 2(\tilde{x}_{ip} - \tilde{x}_{jp})^2\right\}}{(\tilde{x}_{ip} - \tilde{x}_{jp})^2 + \sigma_{pp,i} + \sigma_{pp,j}} > 0
\end{aligned}
\tag{B.2}
$$

APPENDIX C Preliminary analysis result for model architecture

Table C.1 presents the reconstruction errors of the AE models in different architectures. The first column of Table C.1 indicates the front-end architecture as the numbers of hidden neurons on the layers following the input layer where all the architectures were designed in a symmetric shape.

Table C.1 Reconstruction MSE in different architectures.

Front-end architecture	Reconstruction MSE
Input – 160 –30	0.003839
Input – 320 – 30	0.002815
Input – 640 –30	0.002410
Input – 1280– 30	0.001798
Input – 1280 – 160 – 30	0.001711
Input – 1280 – 320 – 160 – 30	0.001686
Input – 1280 – 640 – 160 – 30	0.001650
Input – 1280 – 1280 – 160 – 30	0.001604
Input – 1280 – 1280 – 320 – 160 – 30	0.001586
Input – 1280 – 1280 – 320 – 320 – 160 – 30	0.001618
Input – 1280 – 1280 – 640 – 320 – 320 – 30	0.001599
Input – 1280 – 1280 – 640 – 640 – 160 – 160 – 30	0.001607
Input – 1280 – 1280 – 640 – 640 – 320 – 160 – 30	0.001562

APPENDIX D Derivation of the subgradient of group-exclusive group lasso

Given an index set of groups, $\mathcal{G}$, let $\boldsymbol{\omega}_s = \left[\omega_{1s}, \ldots, \omega_{p_s,s}\right]^{\mathrm{T}} \in \mathbb{R}^{p_s}$ be a vector of the weights for group $s \in \mathcal{G}$. For a pair of groups r and s, the derivative of (5.18) for $\boldsymbol{\omega}_s \neq 0$ is

$$\frac{\partial \mathcal{R}_{GG}(r,s)}{\partial \omega_{js}} = \frac{\delta_{rs}}{2} \frac{\omega_{js}}{\|\boldsymbol{\omega}_s\|_2} \left(\frac{\|\boldsymbol{\omega}_r\|_2}{\|\boldsymbol{\omega}_s\|_2}\right)^{1/2}. \tag{D.1}$$

Accordingly, the derivative of the GG regularization in (5.14) is given, using (D.1), as

$$\begin{aligned}\frac{\partial \mathcal{R}_{GG}}{\partial \omega_{js}} &= \frac{\partial}{\partial \omega_{js}} \left(\sum_{s\in\mathcal{G}} \mathcal{R}_{GG}(s,s)\right) + \frac{\partial}{\partial \omega_{js}} \left(\sum_{s\in\mathcal{G}} \sum_{r\in\mathcal{G}_{(-s)}} \mathcal{R}_{GG}(r,s)\right) \\ &= \frac{\omega_{js}}{\|\boldsymbol{\omega}_s\|_2} \sum_{r\in\mathcal{G}} \delta_{rs} \left(\frac{\|\boldsymbol{\omega}_r\|_2}{\|\boldsymbol{\omega}_s\|_2}\right)^{1/2}\end{aligned} \tag{D.2}$$

where

$$\frac{\partial}{\partial \omega_{js}} \left(\sum_{s\in\mathcal{G}} \mathcal{R}_{GG}(s,s)\right) = \delta_{ss} \frac{\omega_{js}}{\|\boldsymbol{\omega}_s\|_2} \tag{D.3}$$

$$\begin{aligned}\frac{\partial}{\partial \omega_{js}} \left(\sum_{s\in\mathcal{G}} \sum_{r\in\mathcal{G}_{(-s)}} \mathcal{R}_{GG}(r,s)\right) &= 2 \frac{\partial}{\partial \omega_{js}} \left(\sum_{r\in\mathcal{G}_{(-s)}} \mathcal{R}_{GG}(r,s)\right) \\ &= \frac{\omega_{js}}{\|\boldsymbol{\omega}_s\|_2} \sum_{r\in\mathcal{G}_{(-s)}} \delta_{rs} \left(\frac{\|\boldsymbol{\omega}_r\|_2}{\|\boldsymbol{\omega}_s\|_2}\right)^{1/2}.\end{aligned} \tag{D.4}$$

For $\boldsymbol{\omega}_s = 0$, the regularization term is non-differentiable at $\mathbf{0}$ and we consider the subdifferential of $f(\boldsymbol{\omega}_s) = \delta_{ss}\|\boldsymbol{\omega}_s\|_2 + 2\sum_{r\in\mathcal{G}_{(-s)}} \delta_{rs}(\|\boldsymbol{\omega}_r\|_2 \cdot \|\boldsymbol{\omega}_s\|_2)^{1/2}$, following the general form, as

$$\begin{aligned}\partial f(\boldsymbol{\omega}_s) &= \{\mathbf{v} \in \mathbb{R}^{p_s} \mid f(\boldsymbol{\omega}_s') \geq f(\boldsymbol{\omega}_s) + \mathbf{v}^{\mathrm{T}}(\boldsymbol{\omega}_s' - \boldsymbol{\omega}_s),\ \forall \boldsymbol{\omega}_s' \in \mathbb{R}^{p_s}\} \\ &= \{\mathbf{v} \in \mathbb{R}^{p_s} \mid f(\boldsymbol{\omega}_s') \geq \mathbf{v}^{\mathrm{T}}\boldsymbol{\omega}_s',\ \forall \boldsymbol{\omega}_s' \in \mathbb{R}^{p_s}\}.\end{aligned} \tag{D.5}$$

At any $\boldsymbol{\omega}_s' \in \mathbb{R}^{p_s}$, it holds

$$f(\boldsymbol{\omega}_s') \geq \delta_{ss}\|\boldsymbol{\omega}_s'\|_2 \geq 0, \tag{D.6}$$

and we define a set $\partial f^*(\boldsymbol{\omega}_s) \subseteq \partial f(\boldsymbol{\omega}_s)$ from the inequality in (D.6) as

$$\partial f^*(\boldsymbol{\omega}_s) = \{\mathbf{v} \in \mathbb{R}^{p_s} \mid \delta_{ss}\|\boldsymbol{\omega}_s'\|_2 \geq \mathbf{v}^{\mathrm{T}}\boldsymbol{\omega}_s',\ \forall \boldsymbol{\omega}_s' \in \mathbb{R}^{p_s}\}. \tag{D.7}$$

The inequality condition in (D.7) becomes the subdifferential of the standard group lasso with a coefficient. Thus, a subgradient vector $\mathbf{v}$ of $f(\boldsymbol{\omega}_s)$ at $\boldsymbol{\omega}_s = 0$ needs to satisfy $\|\mathbf{v}\| \leq \delta_{ss}$ and we define the subdifferential $\partial f^*(\boldsymbol{\omega}_s)$ as

$$\partial f^*(\boldsymbol{\omega}_s) = \{\mathbf{v} \in \mathbb{R}^{p_s} \mid \|\mathbf{v}\| \leq \delta_{ss}\}. \tag{D.8}$$

REFERENCE

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016, October). Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (pp. 308-318). ACM.

Andridge, R. R., & Little, R. J. (2010). A review of hot deck imputation for survey non-response. *International Statistical Review*, *78*(1), 40-64.

Bae, J., & Park, J. (2019). Count-based change point detection via multi-output log-Gaussian Cox processes. *IISE Transactions*, 1-16.

Bastani, K., Kong, Z., Huang, W., Huo, X., & Zhou, Y. (2012). Fault diagnosis using an enhanced relevance vector machine (RVM) for partially diagnosable multistation assembly processes. *IEEE Transactions on Automation Science and Engineering*, *10*(1), 124-136.

Belanche, L. A., Kobayashi, V., & Aluja, T. (2014). Handling missing values in kernel methods with application to microbiology data. *Neurocomputing*, *141*, 110-116.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and Regression Trees*. New York: CRC Press.

Caesarendra, W., Widodo, A., & Yang, B. S. (2010). Application of relevance vector machine and logistic regression for machine degradation assessment. *Mechanical Systems and Signal Processing*, *24*(4), 1161-1171.

Chang, M. H., Kang, M., & Pecht, M. (2017). Prognostics-based LED qualification using similarity-based statistical measure with RVM regression model. *IEEE Transactions on Industrial Electronics*, *64*(7), 5667-5677.

Chen, P., Wu, S., Lin, J., Ko, F., Lo, H., Wang, J., Yu, C. H., & Liang, M. (2005, September). Virtual metrology: A solution for wafer to wafer advanced process control. In *ISSM 2005, IEEE International Symposium on Semiconductor Manufacturing, 2005.* (pp. 155-157). IEEE.

Chen, X., Yuan, X., Yan, S., Tang, J., Rui, Y., & Chua, T. S. (2011, November). Towards multi-semantic image annotation with graph regularized exclusive group lasso. In *Proceedings of the 19th ACM International Conference on Multimedia* (pp. 263-272). ACM.

Cheng, F. T., Chang, J. Y. C., Huang, H. C., Kao, C. A., Chen, Y. L., & Peng, J. L. (2011). Benefit model of virtual metrology and integrating AVM into MES. *IEEE Transactions on Semiconductor Manufacturing*, *24*(2), 261-272.

Chien, C. F., Wang, W. C., & Cheng, J. C. (2007). Data mining for yield enhancement in semiconductor manufacturing and an empirical study. *Expert Systems with Applications*, *33*(1), 192-198.

Choi, J., & Jeong, M. K. (2018). Deep autoencoder with clipping fusion regularization on multistep process signals for virtual metrology. *IEEE sensors letters*, *3*(1), 1-4.

Cotton, C. (1991). Functional description of the generalized edit and imputation system. business survey methods division. *Statistics Canada*, *59*, 447-461.

Courbariaux, M., Hubara, I., Soudry, D., El-Yaniv, R., & Bengio, Y. (2016). Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1. *arXiv preprint arXiv:1602.02830*.

Covo, S., & Elalouf, A. (2014). A novel single-gamma approximation to the sum of independent gamma variables, and a generalization to infinitely divisible distributions. *Electronic Journal of Statistics*, *8*(1), 894-926.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1-38.

Di Maio, F., Tsui, K. L., & Zio, E. (2012). Combining relevance vector machines and exponential regression for bearing residual life estimation. *Mechanical Systems and Signal Processing*, *31*, 405-427.

Diebold, A. C. (1995, November). Overview of metrology requirements based on the 1994 National Technology Roadmap for semiconductors. In *Proceedings of SEMI Advanced Semiconductor Manufacturing Conference and Workshop* (pp. 50-60). IEEE.

Eirola, E., Doquire, G., Verleysen, M., & Lendasse, A. (2013). Distance estimation in numerical data sets with missing values. *Information Sciences*, *240*, 115-128.

Eirola, E., Lendasse, A., Vandewalle, V., & Biernacki, C. (2014). Mixture of Gaussians for distance estimation with missing data. *Neurocomputing*, 131, 32-42.

Escoufier, Y. (1973). Le traitement des variables vectorielles. *Biometrics*, 751-760.

Feng, Y., Wen, M., Zhang, J., Ji, F., & Ning, G. X. (2016, February). Sum of arbitrarily correlated Gamma random variables with unequal parameters and its application in wireless communications. In *2016 International Conference on Computing, Networking and Communications* (pp. 1-5). IEEE.

Gazzola, G., Choi, J., Kwak, D. S., Kim, B., Kim, D. M., Tong, S. H., & Jeong, M. K. (2018). Integrated variable importance assessment in multi-stage processes. *IEEE Transactions on Semiconductor Manufacturing*, *31*(3), 343-355.

Genton, M. G. (Ed.). (2004). *Skew-elliptical distributions and their applications: A journey beyond normality*. New York: CRC Press.

Glorot, X., & Bengio, Y. (2010, March). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics* (pp. 249-256). JMLR.

Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning*. Cambridge: MIT press.

Gupta, S., Agrawal, A., Gopalakrishnan, K., & Narayanan, P. (2015, June). Deep learning with limited numerical precision. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning* (Vol. 37, pp. 1737-1746). JMLR.

Hastie, T., Tibshirani, R., & Wainwright, M. (2015). *Statistical learning with sparsity: the lasso and generalizations*. Boca Raton: CRC press.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770-778). IEEE.

He, Q. P., & Wang, J. (2007). Fault detection using the k-nearest neighbor rule for semiconductor manufacturing processes. *IEEE Transactions on Semiconductor Manufacturing*, *20*(4), 345-354.

Hirai, T., & Kano, M. (2015). Adaptive virtual metrology design for semiconductor dry etching process through locally weighted partial least squares. *IEEE Transactions on Semiconductor Manufacturing*, *28*(2), 137-144.

Hoefling, H. (2010). A path algorithm for the fused lasso signal approximator. *Journal of Computational and Graphical Statistics*, *19*(4), 984-1006.

Hsu, S. C., & Chien, C. F. (2007). Hybrid data mining approach for pattern extraction from wafer bin map to improve yield in semiconductor manufacturing. *International Journal of Production Economics*, *107*(1), 88-103.

Hung, M. H., Lin, T. H., Cheng, F. T., & Lin, R. C. (2007). A novel virtual metrology scheme for predicting CVD thickness in semiconductor manufacturing. *IEEE/ASME Transactions on Mechatronics*, *12*(3), 308-316.

Hwang, S., & Jeong, M. K. (2018). Robust relevance vector machine for classification with variational inference. *Annals of Operations Research*, *263*(1-2), 21-43.

Hwang, S., Jeong, M. K., & Yum, B. J. (2014). Robust relevance vector machine with variational inference for improving virtual metrology accuracy. *IEEE Transactions on Semiconductor Manufacturing*, *27*(1), 83-94.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Johnson, N. L., Kotz, S., & Balakrishnan, N. (1970). *Continuous univariate distributions*. Boston: Houghton Mifflin.

Jurado, S., Nebot, À., Mugica, F., & Mihaylov, M. (2017). Fuzzy inductive reasoning forecasting strategies able to cope with missing data: A smart grid application. *Applied Soft Computing*, 51, 225-238.

Josse, J., & Holmes, S. (2016). Measuring multivariate association and beyond. *Statistics Surveys*, *10*, 132.

Kadlec, P., Gabrys, B., & Strandt, S. (2009). Data-driven soft sensors in the process industry. *Computers & Chemical Engineering*, *33*(4), 795-814.

Kang, P., Kim, D., & Cho, S. (2016). Semi-supervised support vector regression based on self-training with label uncertainty: An application to virtual metrology in semiconductor manufacturing. *Expert Systems with Applications*, *51*, 85-106.

Kang, P., Kim, D., Lee, H. J., Doh, S., & Cho, S. (2011). Virtual metrology for run-to-run control in semiconductor manufacturing. *Expert Systems with Applications*, *38*(3), 2508-2522.

Kang, P., Lee, H. J., Cho, S., Kim, D., Park, J., Park, C. K., & Doh, S. (2009). A virtual metrology system for semiconductor manufacturing. *Expert Systems with Applications*, *36*(10), 12554-12561.

Khan, A. A., Moyne, J. R., & Tilbury, D. M. (2007). An approach for factory-wide control utilizing virtual metrology. *IEEE Transactions on Semiconductor Manufacturing*, *20*(4), 364-375.

Khatab, Z. E., Hajihoseini, A., & Ghorashi, S. A. (2018). A fingerprint method for indoor localization using autoencoder based deep extreme learning machine. *IEEE Sensors Letters*, *2*(1), 1-4.

Khellat-Kihel, S., Abrishambaf, R., Monteiro, J. L., & Benyettou, M. (2016). Multimodal fusion of the finger vein, fingerprint and the finger-knuckle-print using Kernel Fisher analysis. *Applied Soft Computing*, 42, 439-447.

Kim, J. K., & Fuller, W. (2004). Fractional hot deck imputation. *Biometrika*, *91*(3), 559-578.

Kim, J., Lee, Y., & Kim, H. (2018). Detection and clustering of mixed-type defect patterns in wafer bin maps. *IISE Transactions*, *50*(2), 99-111.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kong, D., Liu, J., Liu, B., & Bao, X. (2016, February). Uncorrelated group LASSO. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence* (pp. 1765-1771). AAAI Press.

Kong, D., Fujimaki, R., Liu, J., Nie, F., & Ding, C. (2014, December). Exclusive feature learning on arbitrary structures via $\ell_{1,2}$-norm. In *Proceedings of the 27th International Conference on Neural Information Processing Systems* (Vol. 1, pp. 1655-1663). MIT Press.

Kurz, D., De Luca, C., & Pilz, J. (2014). A sampling decision system for virtual metrology in semiconductor manufacturing. *IEEE Transactions on Automation Science and Engineering*, *12*(1), 75-83.

Land, S. and Friedman, J. (1996) Variable fusion: a new method of adaptive signal regression. Technical Report. Department of Statistics, Stanford University, Stanford.

Längkvist, M., Karlsson, L., & Loutfi, A. (2014). A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters*, *42*, 11-24.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436-444.

Lee, D. H., Yang, J. K., Lee, C. H., & Kim, K. J. (2019). A data-driven approach to selection of critical process steps in the semiconductor manufacturing process considering missing and imbalanced data. *Journal of Manufacturing Systems*, *52*, 146-156.

Lee, H., Kim, Y., & Kim, C. O. (2017). A deep learning model for robust wafer fault monitoring with sensor measurement noise. *IEEE Transactions on Semiconductor Manufacturing*, *30*(1), 23-31.

Lin, D., Zhang, J., Li, J., Calhoun, V. D., Deng, H. W., & Wang, Y. P. (2013). Group sparse canonical correlation analysis for genomic data integration. *BMC bioinformatics*, *14*(1), 245.

Lin, T. H. (2010). A comparison of multiple imputation with EM algorithm and MCMC method for quality of life missing data. *Quality & Quantity*, *44*(2), 277-287.

Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., Van Der Laak, J.A., Van Ginneken, B., & Sánchez, C.I. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis*, *42*, 60-88.

Little, R. J. (1992). Regression with missing X's: a review. *Journal of the American Statistical Association*, *87*(420), 1227-1237.

Little, R. J., & Rubin, D. B. (1986). *Statistical analysis with missing data*. New York: John Wiley and Sons.

Liu, J., Huang, J., Ma, S., & Wang, K. (2012). Incorporating group correlations in genome-wide association studies using smoothed group Lasso. *Biostatistics*, *14*(2), 205-219.

Martínez-Costa, C., Mas-Machuca, M., Benedito, E., & Corominas, A. (2014). A review of mathematical programming models for strategic capacity planning in manufacturing. *International Journal of Production Economics*, *153*, 66-85.

Meng, X. L., & Rubin, D. B. (1993). Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, *80*(2), 267-278.

Nakagami, M. (1960, June). The m-distribution—A general formula of intensity distribution of rapid fading. In *Statistical Methods in Radio Wave Propagation* (pp. 3-36). Pergamon.

Nebot-Troyano, G., & Belanche-Muñoz, L. A. (2010). A kernel extension to handle missing data. In *Research and Development in Intelligent Systems XXVI* (pp. 165-178). Springer.

Negahban, A., & Smith, J. S. (2014). Simulation for manufacturing system design and operation: Literature review and analysis. *Journal of Manufacturing Systems*, *33*(2), 241-261.

Nguyen, T. T., & Tsoy, Y. (2017). A kernel PLS based classification method with missing data handling. *Statistical Papers*, *58*(1), 211-225.

Pan, J. C. H., & Tai, D. H. (2009). Implementing virtual metrology for in-line quality control in semiconductor manufacturing. *International Journal of Systems Science*, *40*(5), 461-470.

Park, E. L., Park, J., Yang, J., Cho, S., Lee, Y. H., & Park, H. S. (2014). Data based segmentation and summarization for sensor data in semiconductor manufacturing. *Expert Systems with Applications*, *41*(6), 2619-2629.

Park, H. M., Grimard, D. S., & Grizzle, J. W. (2003). Sensor fault detection in etch based on broadband rf signal observation. *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films*, *21*(3), 814-824.

Pelckmans, K., De Brabanter, J., Suykens, J. A., & De Moor, B. (2005). Handling missing values in support vector machine classifiers. *Neural Networks*, *18*(5), 684-692.

Purwins, H., Barak, B., Nagi, A., Engel, R., Höckele, U., Kyek, A., Cherla, S. Lenz, B., Pfeifer, G., & Weinzierl, K. (2014). Regression methods for virtual metrology of layer

thickness in chemical vapor deposition. *IEEE/ASME Transactions on Mechatronics*, *19*(1), 1-8.

Qin, S. J., Cherry, G., Good, R., Wang, J., & Harrison, C. A. (2006). Semiconductor manufacturing process control and monitoring: A fab-wide framework. *Journal of Process Control*, *16*(3), 179-191.

Ranzato, M. A., Poultney, C., Chopra, S., & LeCun, Y. (2006, December). Efficient learning of sparse representations with an energy-based model. In *Proceedings of the 19th International Conference on Neural Information Processing Systems* (pp. 1137-1144). MIT Press.

Rasmussen, C. E., & Williams, C. K. (2006). *Gaussian processes for machine learning*. Cambridge: MIT press.

Rifai, S., Vincent, P., Muller, X., Glorot, X., & Bengio, Y. (2011, June). Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on International Conference on Machine Learning* (pp. 833-840). Omnipress.

Roberts, C., & Geisser, S. (1966). A necessary and sufficient condition for the square of a random variable to be gamma. *Biometrika*, *53*(1/2), 275-278.

Rubin, D. B. (1978). Multiple imputations in sample surveys-a phenomenological Bayesian approach to nonresponse. In *Proceedings of the Survey Research Methods Section of the American Statistical Association* (Vol. 1, pp. 20-34). American Statistical Association.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986, January). Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (Vol. 1, pp. 318-362). MIT Press.

Sainath, T. N., Kingsbury, B., Sindhwani, V., Arisoy, E., & Ramabhadran, B. (2013, May). Low-rank matrix factorization for deep neural network training with high-dimensional output targets. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 6655-6659). IEEE.

Sande, I. G. (1983). Hot-deck imputation procedures. *Incomplete Data in Sample Surveys*, *3*, 339-349.

Scardapane, S., Comminiello, D., Hussain, A., & Uncini, A. (2017). Group sparse regularization for deep neural networks. *Neurocomputing*, *241*, 81-89.

Schafer, J. L. (1997). *Analysis of incomplete multivariate data*. CRC press.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85-117.

Schölkopf, B., Smola, A. J., & Bach, F. (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Sexton, J., & Swensen, A. R. (2000). ECM algorithms that converge at the rate of EM. *Biometrika*, *87*(3), 651-662.

Shawe-Taylor, J., & Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge university press.

Siddique, J., & Belin, T. R. (2008). Multiple imputation using an iterative hot-deck with distance-based donor selection. *Statistics in Medicine*, *27*(1), 83-102.

Simon, N., Friedman, J., Hastie, T., & Tibshirani, R. (2013). A sparse-group lasso. *Journal of Computational and Graphical Statistics*, *22*(2), 231-245.

Simon, N., & Tibshirani, R. (2012). Standardization and the group lasso penalty. *Statistica Sinica*, *22*(3), 983.

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, *14*(3), 199-222.

Smola, A. J., Vishwanathan, S. V. N., & Hofmann, T. (2005, January). Kernel Methods for Missing Variables. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics* (pp. 325-332). Society for Artificial Intelligence and Statistics.

Son, Y., & Lee, J. (2016). Active learning using transductive sparse Bayesian regression. *Information Sciences*, *374*, 240-254.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929-1958.

Strobl, C., Boulesteix, A. L., Kneib, T., Augustin, T., & Zeileis, A. (2008). Conditional variable importance for random forests. *BMC Bioinformatics*, *9*(1), 307.

Susto, G. A., Pampuri, S., Schirru, A., Beghi, A., & De Nicolao, G. (2015). Multi-step virtual metrology for semiconductor manufacturing: A multilevel and regularization methods-based approach. *Computers & Operations Research*, *53*, 328-337.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 267-288.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., & Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *67*(1), 91-108.

Tieleman, T., & Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, *4*(2), 26-31.

Tipping, M. E. (2001). Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, *1*, 211-244.

Tipping, M. E., & Faul, A. C. (2003, January). Fast marginal likelihood maximisation for sparse Bayesian models. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics* (pp. 1-13). Society for Artificial Intelligence and Statistics.

Vallejo, M., de la Espriella, C., Gómez-Santamaría, J., Ramírez-Barrera, A. F., & Delgado-Trejos, E. (2019). Soft metrology based on machine learning: a review. *Measurement Science and Technology*, *31*(3), 032001.

Van Hulse, J., & Khoshgoftaar, T. M. (2014). Incomplete-case nearest neighbor imputation in software measurement data. *Information Sciences*, *259*, 596-610.

Vapnik, V. (1998). *The nature of statistical learning theory*. New York: John Wiley and Johns.

Von Hippel, P. T. (2009). 8. How to impute interactions, squares, and other transformed variables. *Sociological methodology*, *39*(1), 265-291.

Wang, J., Chen, Y., Hao, S., Peng, X., & Hu, L. (2019). Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters*, *119*, 3-11.

Wang, H., & Leng, C. (2008). A note on adaptive group lasso. *Computational Statistics & Data Analysis*, *52*(12), 5277-5286.

Wang, J., Xu, C., Yang, X., & Zurada, J. M. (2017). A novel pruning algorithm for smoothing feedforward neural networks based on group lasso method. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(5), 2012-2024.

Weiss, S. M., Baseman, R. J., Tipu, F., Collins, C. N., Davies, W. A., Singh, R., & Hopkins, J. W. (2010). Rule-based data mining for yield improvement in semiconductor manufacturing. *Applied Intelligence*, *33*(3), 318-329.

Wen, W., Wu, C., Wang, Y., Chen, Y., & Li, H. (2016, December). Learning structured sparsity in deep neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems* (pp. 2082-2090). ACM.

Wipf, D., & Nagarajan, S. (2007, December). A new view of automatic relevance determination. In *Proceedings of the 20th International Conference on Neural Information Processing Systems* (pp. 1625-1632). Curran Associates Inc.

Wipf, D. P., & Nagarajan, S. S. (2008). A new view of automatic relevance determination. In *Advances in Neural Information Processing Systems* (pp. 1625-1632).

Wu, W. M., Cheng, F. T., Lin, T. H., Zeng, D. L., & Chen, J. F. (2011). Selection schemes of dual virtual-metrology outputs for enhancing prediction accuracy. *IEEE Transactions on Automation Science and Engineering*, *8*(2), 311-318.

Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1492-1500). IEEE.

Yan, J., Zheng, W., Zhou, X., & Zhao, Z. (2011). Sparse 2-D canonical correlation analysis via low rank matrix approximation for feature extraction. *IEEE Signal Processing Letters*, *19*(1), 51-54.

Yiqi, L., Daoping, H., & Zhifu, L. (2013). A SEVA soft sensor method based on self-calibration model and uncertainty description algorithm. *Chemometrics and Intelligent Laboratory Systems*, *126*, 38-49.

Yoon, J., & Hwang, S. J. (2017, August). Combined group and exclusive sparsity for deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (pp. 3958-3966). JMLR.

Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., & Lipson, H. (2015). Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579.*

Yu, K., & Lin, Y. (2011, June). Learning image representations from the pixel level via hierarchical sparse coding. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1713-1720). IEEE Computer Society.

Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *68*(1), 49-67.

Yugma, C., Blue, J., Dauzère-Pérès, S., & Obeid, A. (2015). Integration of scheduling and advanced process control in semiconductor manufacturing: review and outlook. *Journal of Scheduling*, *18*(2), 195-205.

Zeng, D., & Spanos, C. J. (2009). Virtual metrology modeling for plasma etch operations. *IEEE Transactions on Semiconductor Manufacturing*, *22*(4), 419-431.

Zhang, K., Song, Z., & Guan, Y. L. (2004). Simulation of Nakagami fading channels with arbitrary cross-correlation and fading parameters. *IEEE Transactions on Wireless Communications*, *3*(5), 1463-1468.

Zhang, Z., Zhao, M., & Chow, T. W. (2012). Binary-and multi-class group sparse canonical correlation analysis for feature extraction and classification. *IEEE Transactions on Knowledge and Data Engineering*, *25*(10), 2192-2205.

Zhao, L., Hu, Q., & Wang, W. (2015). Heterogeneous feature selection with multi-modal deep neural networks and sparse group lasso. *IEEE Transactions on Multimedia*, *17*(11), 1936-1948.

Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, *30*(11), 3212-3232.

Zhong, Y., Ma, A., soon Ong, Y., Zhu, Z., & Zhang, L. (2018). Computational intelligence in optical remote sensing image processing. *Applied Soft Computing*, *64*, 75-93.

Zhou, Y., Jin, R., & Hoi, S. C. H. (2010, March). Exclusive lasso for multi-task feature selection. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (pp. 988-995). PMLR.

Zhu, X., Zhou, W., & Li, H. (2018, July). Improving deep neural network sparsity through decorrelation regularization. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence* (pp. 3264-3270). ACM.