

Sparse Model Selection using Information Complexity

A Dissertation Presented for the
Doctor of Philosophy
Degree

The University of Tennessee, Knoxville

Yaojin Sun

May 2022

© by Yaojin Sun, 2022
All Rights Reserved.

This work is dedicated to my wife, Dr. Jiali Yu, my parents, and advisor Dr. Hamparsum Bozdogan.

Acknowledgments

I want to thank my academic advisor and mentor, Prof. Hamparsum Bozdogan, for his continued help during my Ph.D. studies. In addition, I am grateful for the support, feedback, and expertise.

I want to thank the members of my Ph.D. committee, Dr. Haileab Hilafu, Dr. Emre Demirkaya, Dr. Chuanren Liu, and Dr. Matthew Harris. They gave me lots of ideas to help with my dissertation for serving on my Ph.D. committee and providing valuable comments.

I would like to thank Dr. Hong Guo, a UTK professor I respect. He has always supported me during my difficult times. His wise advise and suggestions helped me tremendously, which I'm grateful. And thank you to Dr. Qin Xu for his help when I needed it most. I want to express my gratitude to Harrison Hicks, my former classmate, who was kind enough to help me during my most challenging times as a student.

It would be remiss of me not to thank my parents for all their help over the years. Specifically, I would like to thank them for teaching me right from wrong and encouraging me always to do my best.

I am grateful to my wife, Dr. Jiali Yu. We met and fell in love at UTK and have been through thick and thin ever since. She has been my last line of defense during a low point in my life, and I wouldn't be who I am without her by my side.

I would like to thank Tan Li. As a friend for over a decade, she provided me with guidance and support. Her knowledge and experience have given me a new perspective on how strong friendships can be.

I am also grateful to Fenghao Ding, who has shown me that some things can travel through time and remain the same throughout history.

Thank the many professors and colleges at UTK who have provided me with invaluable insight, instruction, friendliness, and more.

Abstract

This dissertation studies and uses the application of information complexity to statistical model selection through three different projects. Specifically, we design statistical models that incorporate sparsity features to make the models more explanatory and computationally efficient.

In the first project, we propose a Sparse Bridge Regression model for variable selection when the number of variables is much greater than the number of observations if model misspecification occurs. The model is demonstrated to have excellent explanatory power in high-dimensional data analysis through numerical simulations and real-world data analysis.

The second project proposes a novel hybrid modeling method that utilizes a mixture of sparse principal component regression (MIX-SPCR) to segment high-dimensional time series data. Using the MIX-SPCR model, we empirically analyze the S&P 500 index data (from 1999 to 2019) and identify two key change points.

The third project investigates the use of nonlinear features in the Sparse Kernel Factor Analysis (SKFA) method to derive the information criterion. Using a variety of wide datasets, we demonstrate the benefits of SKFA in the nonlinear representation and classification of data. The results obtained show the flexibility and the utility of information complexity in such data modeling problems.

Table of Contents

1	Introduction	1
1.1	Statistical Model Selection	2
1.2	High-dimensional Data	4
1.3	Sparse Model	4
1.4	Information Criterion and Information Complexity	6
2	Selection of Variables in Sparse Bridge Regression Under Misspecification Using Information Complexity	8
2.1	Introduction	9
2.2	Sparse Bridge Regression Model	12
2.2.1	A Brief Background: The Least Squares Regression Model	12
2.2.2	Sparse Bridge Regression Under Misspecification	14
2.3	Misspecification Resistant Information Complexity	17
2.4	New Proposed Estimation of the Sparse Bridge Regression Model Parameters	18
2.4.1	Simulated Annealing and Particle Swarm Optimization: SA-PSO . .	19
2.4.2	Smoothed Covariance Estimators (SCEs)	20
2.4.3	Regularized Local Quadratic Approximation (RLQA)	22
2.5	Monte Carlo Simulation Results	23
2.5.1	Simulation Protocol	24
2.5.2	Effects of Smoothed Covariance Estimator	24
2.5.3	Comparative Performance of Sparse Bridge Regression with Lasso . .	25
2.6	Real Data Example: Riboflavin (Vitamin B2) Dataset	30
2.7	Conclusions and Discussion	31

3	Segmentation of Time Series Data Using the Mixture of Sparse Principal Component Regression Model with Information Complexity	36
3.1	Introduction	37
3.2	Model and Methods	40
3.2.1	Sparse Principal Component Analysis (SPCA)	42
3.2.2	Mixtures of SPCR Model for Time Series Data	44
3.3	Regularized Entropy-Based EM Clustering Algorithm	46
3.3.1	The Entropy of EM Mixture Probability	46
3.3.2	E-Step (Expectation)	47
3.3.3	M-Step (Maximization)	48
3.4	Information Complexity Criteria	49
3.4.1	Derivation of Information Complexity in <i>MIX-SPCR</i> Model for Time-Series Data	51
3.5	Monte Carlo Simulation Study	54
3.5.1	Simulation Protocol	55
3.5.2	Detection of Structural Change Point	56
3.5.3	A Large-Scale Monte Carlo Simulation	58
3.6	Case Study: Segmentation of the S&P 500 Index	61
3.6.1	Description of the Data	61
3.6.2	Computational Results	62
3.6.3	Interpretation of Computational Results	64
3.7	Conclusion and Discussion	70
4	Sparse Kernel Factor Analysis In High Dimensional Data Using The Information Complexity Criterion	72
4.1	Introduction	72
4.2	Kernel Modeling in Feature Space	74
4.3	Sparse Kernel Factor Analysis	76
4.3.1	Factor Analysis (FA) Model	76
4.3.2	Kernel Factor Analysis (KFA) and Sparse Kernel Factor Analysis (SKFA)	79

4.3.3	Maximum Likelihood Estimation	80
4.3.4	SVD-based Parameter Estimation Approach	81
4.3.5	EM-based Parameter Estimation Approach	84
4.3.6	Estimating Sparse Kernel Factor Scores	86
4.3.7	Choosing the Number of Sparse Kernel Factors	88
4.4	Information Complexity Criterion	89
4.5	Numerical Examples and Computational Results	90
4.5.1	A Toy Example Using Iris Data	90
4.5.2	Monte Carlo Evidence: Stability Study of SKFA Algorithm via Pre- image Learning	98
4.5.3	Real Benchmark Examples Using High Dimensional Wide-Datasets .	101
4.6	Conclusion	108
5	Conclusion and Future Research	110
	Bibliography	113
	Appendix	130
A	Acronyms	131
B	SA-PSO Algorithm	132
C	EM Algorithm for SKFA (M-Step)	134
	Vita	135

List of Tables

2.1	Parameters of SA-based Particle Swarm Optimization Algorithm.	21
2.2	Comparison of SBR with Lasso as the number of variable sets increases. . . .	29
2.3	Variable selection for riboflavin production data.	32
3.1	Frequency of the choice of the true model with information criteria	59
3.2	The ICOMP(IFIM) values of segmentation results	63
3.3	SPCR coefficients (β_g) of three different segments.	65
3.4	Sparse Principal Component (SPC) of Segment 1 (1999-01-01 $\sim$ 2007-10-26). .	66
3.5	Sparse Principal Component (SPC) of Segment 2 (2007-10-27 $\sim$ 2016-12-29). .	66
3.6	Sparse Principal Component (SPC) of Segment 3 (2016-12-30 $\sim$ 2019-12-31). .	67
4.1	Selected Kernel Functions and their hyperparameters	92
4.2	ICOMP values using Gaussian kernel.	92
4.3	ICOMP values using Exponential kernel.	92
4.4	ICOMP values using Laplace kernel.	93
4.5	ICOMP values of the Sparse KFA and Non-sparse KFA using the Iris Data. .	93
4.6	Neural network model	102
4.7	Hit frequencies of choosing number of factors using Gaussian kernel in 10 replication of the experiment with SKFA and Pre-image analysis with SKFA. .	102
4.8	ICOMP values for Gauss Kernel SKFA (Ovarian Cancer).	104
4.9	ICOMP values for Exponential Kernel SKFA (Ovarian Cancer).	105
4.10	ICOMP values for Laplace Kernel SKFA (Ovarian Cancer).	105
4.11	Wide datasets and SKFA results.	107
4.12	Classification Error of Brain Data.	107

4.13	Classification Error of Colon Data.	107
4.14	Classification Error of Leukemia Data.	109
4.15	Classification Error of Prostate Data.	109
1	Abbreviations.	131

List of Figures

2.1	Plot of the penalty $\sum_{j=1}^p \beta_j ^q$ for different values of $0 < q < 1$	21
2.2	Comparison of eigenvalues w/o OAS smoothed covariance estimator of the Gram matrix $(X'X)$ for the simulated dataset with $\gamma = 0.3$, $n = 100$ and $p = 1000$	26
2.3	Estimated coefficients w/o OAS smoothed covariance estimators in SBR using LQA algorithm.	27
2.4	The behavior of $\max(\hat{\beta}^{(k+1)} - \hat{\beta}^{(k)}) < \delta = 10^{-4}$	27
2.5	Comparison of Sparse Bridge Regression (SBR) with Lasso.	29
2.6	Comparison of SA-PSO and grid-search results for Riboflavin data	32
2.7	ICOMP values of the entire model space from SBR of Riboflavin data using SA-PSO algorithm.	33
2.8	Plot of predicted y values against real values and QQ-plot of the residuals. .	33
2.9	Hierarchically-clustered heatmap of selected genes	34
3.1	The flowchart of the MIX-SPCR method.	41
3.2	The plot of two-segment simulated time-series data.	57
3.3	Sample path of information criteria for the simulated time-series data.	57
3.4	Plot of average and 1SD (standard deviation) of information criterion values	60
3.5	Normalized S&P 500 index and stock prices from January 1999 to December 2019.	63
3.6	Segmented periods and probability	65
3.7	Stem plot of SPC coefficients $\mathbf{P}_g$ for financial companies	67
3.8	Stem plot of SPC coefficients $\mathbf{P}_g$ for real estate companies	68

3.9	Stem plot of SPC coefficients $\mathbf{P}_g$ for energy companies	68
3.10	Stem plot of SPC coefficients $\mathbf{P}_g$ for information technology companies	69
3.11	Overlay plot of the SPCR coefficients before/after 2008 financial crisis. . . .	69
4.1	Mesh plot of ICOMP values for Gaussian Kernel.	93
4.2	Mesh plot of ICOMP values for Exponential Kernel.	94
4.3	Mesh plot of ICOMP values for Laplace Kernel.	94
4.4	Comparison of ICOMP values for different kernel functions.	95
4.5	Sparse vs. Non-sparse KFA using the Gaussian kernel.	95
4.6	Sparse vs. Non-sparse KFA using the Exponential kernel.	96
4.7	Sparse vs. Non-sparse KFA using the Laplace kernel.	96
4.8	t-Distributed Stochastic Neighbor Embedding (tSNE) plot of iris data. . . .	97
4.9	Flow chart of simulation using neural network for pre-image learning.	102
4.10	Comparison of ICOMP of original simulated kernel and reconstructed pre- image kernel under Gaussian kernel function.	104
4.11	SKFA vs. KFA using Gauss Kernel (Ovarian Cancer).	105
4.12	SKFA vs. KFA using Exponential Kernel (Ovarian Cancer).	106
4.13	SKFA vs. KFA using Laplace Kernel (Ovarian Cancer).	106

Chapter 1

Introduction

Statistical models attempt to understand the relationship between variables by modeling the data. In the many of statistical papers cited, statisticians place a great deal of emphasis on how to design model structures for performing understandable analyses of data (see [43, 102, 65]). According to [101], for example, the importance of model building significantly outweighs parameter estimation accuracy.

Many scientists have introduced more complex models over the past few years due to the popularity of machine learning and deep neural networks. For example, ResNets contains 19.4 million parameters [69], and OpenAI’s GPT-3 model contains 175 billion parameters [32], etc. These models’ parameter size and structural complexity far exceed those of classical statistical models which are currently practiced. Due to their excellent prediction performance, these models have achieved surprising results in parameter estimation and are generally considered with excellent generalization ability. However, these models cannot be interpreted from a statistical modeling perspective because we cannot directly connect the individual parameters to the corresponding domain knowledge [100].

This dissertation argues that constructing an interpretable model comes down to model selection. In particular, we want the model to extract as many relationships between variables as possible from the data (also known as domain knowledge) while including as few irrelevant variables as possible. We discuss how to perform model selection with the statistical model to achieve this goal. In particular, we use information criterion as a tool to trade-off model complexity and goodness of fit in different application areas.

1.1 Statistical Model Selection

In the area of model selection, it is noteworthy that perspectives from the machine learning field can enhance our toolbox while advancing our understanding of traditional models. In this discussion, we will cross-pollinate basic concepts from the fields of statistical models and machine learning and then explore interpretability, sparsity, model selection, information criterion, and information complexity.

Statistical models of the classical kind address the following issues: model structure, the meaning of the parameters, likelihood function, and parameter estimation methods. One can improve a model by giving it greater explanatory power or using it for prediction purposes under various scenarios. The researcher determines the parameters based on domain knowledge. In addition, the statistician must provide a probabilistic basis for the model based on a description of the data distribution. For instance, in the case of a simple linear regression, we assume that the error distribution is a Gaussian distribution, which allows us to construct the likelihood function. Finally, we optimize the likelihood function to obtain the specific parameter estimates.

If a model does not include all the relevant variables, it is considered underfitting because it is too simple with respect to the data. A model that contains too many variables that are irrelevant to the dependent variable is subject to overfitting, resulting in a weak prediction. Then the question is to determine which parameters need to be added to the model to avoid overfitting and underfitting. Numerous papers discuss how to decide which variables should be included in a model. Researchers have examined various approaches, including calculating p -values or assigning weights [14, 157, 109].

In recent years, an alternative approach to preventing overfitting or underfitting has gained popularity: the introduction of regularization into statistical models. For example, Lasso [134] imposes an L_1 penalty to adjust the number of variables with zero coefficient, thereby adjusting the complexity of the model. In Lasso's parameter estimation, variable selection is performed passively since the coefficients of irrelevant variables are all zero. Due to this guiding principle, many models emerge rapidly. The idea of introducing regularization to control the complexity of the model also appears in the field of machine learning research.

A regularization term such as L_1 or L_2 is vital when it comes to creating neural networks in order to stabilize the computational results, and neural networks are not strictly probabilistic in nature [97, 141].

However, regularization poses a new problem: determining the coefficients of the penalty term. This process is known as hyperparameter optimization. Similar to traditional variable selection, the determination of the coefficients of the penalty term falls under the category of model selection. Model selection is typically determined by cross-validation in much of the literature. In applying the Lasso model, for instance, researchers plot minimum cross-validation error curves to find the most appropriate coefficients for the penalty terms. Another solution is obtained by comparing the information criterion. Contrary to cross-validation, the information criterion can take advantage of the whole data, which is valuable in specific areas [74].

Throughout the paper, we follow the following steps throughout the dissertation when dealing with various statistical models.

Step 1 Define the model structure and find parameters related to domain knowledge.

Step 2 Introduce penalty terms and the corresponding hyperparameter to ensure that the complexity of the model can be adjusted.

Step 3 Find an optimization method to estimate the parameters accurately.

Step 4 Determine the appropriate information criterion to measure the model's performance.

Step 4 Search the hyperparameter to perform model selection.

Using three different statistical models, we demonstrate how to select statistical models using information criteria (specifically, the information complexity criterion) in this dissertation. To begin with, we need to discuss one scenario where model selection is most likely to occur: high-dimensional data.

1.2 High-dimensional Data

A dataset is considered high-dimensional if its number of dimensions p is of the same order of magnitude as or greater than the size of its observable sample. Further, Fan et al. [58] divides high-dimensionality into moderately high-dimensionality, high-dimensionality, and ultra-high-dimensionality.

Researchers have found that Ordinary Least Squares (OLS) regression methods cannot perform well when dealing with high-dimensional data. In particular, estimating matrix parameters can cause large deviations in parameter estimates due to a minimal error at each observation level. There are two types of errors caused by high-dimensional data [58, 57, 150].

1. The model will mistakenly believe that the two variables are highly correlated, even if they are independent and identically distributed.
2. Noise accumulation: This is especially common in regression models. Errors will accumulate in the parameter estimation algorithm until they are completely invalid.

1.3 Sparse Model

The current solution to the problem posed by high dimensional data is based on the assumption that very few factors significantly impact the dependent variable (relative to the number of variables). In other words, we can present the statistical model in a low-dimensional manifold even when the data are high-dimensional.

Identifying the “significant variables” is the so-called variable selection problem if chosen from the original variables. Because most regression coefficients are zeros (i.e., no effect), we call it a sparse model. For example, scientists believe that the mechanism of disease is influenced by a small number of genes or chemical molecules when performing genomic analyses. Therefore, researchers construct a sparse model to identify these factors.

When the original variables are transformed to a subspace (e.g., PCA), they are commonly referred to as dimensionality reduction. Bidirectional Encoder Representations from Transformers (BERT) [50], for example, compresses the word vector (with more than 30k dimensions) into 768 dimensions, each value representing a combination of effects from

the original data space [132]. Thus, it belongs to the category of dimensionality reduction [139]. Interestingly, Fan and Lv [57] argues in his paper that dimensionality reduction is a particular case of sparse modeling because the number of effective parameters is less than the number of original variables.

Because many of the coefficients in the model’s fitted results are zero in a sparse model, we can streamline the effected variables to a few relevant variables. This sparsity provides the overall model with substantially more explanatory power. For example, Chapter 3 applies a sparsity approach to the S&P 500 data time series analysis. The model accurately determines the few companies with the most significant impact on stock volatility over time.

Another benefit of sparsity is that it makes model parameter estimation more reliable, and the unidentifiability issue is eliminated. For example, in the factor analysis model, because the result of the decomposition can be rotated, results from operations that conform to this model can theoretically consist of an infinite number. This issue is referred to as the “unidentifiability problem.” If we introduce the concept of sparsity, then we can better identify what is going on in the model. Therefore, sparsity allows us to find a unique result for our decomposition.

There are many ways to construct sparse models. The specific operation methods can be divided into three categories: 1) selection-based, 2) penalty-based, and 3) screening-based. We will focus on the first two methods in this thesis. If the reader is interested in the screening-based method, we recommend papers [56] and [87].

From the selection point of view, we need to judge whether each variable is worthy of being selected for the model. The most common way to account for several explanatory variables is by using the stepwise regression method [30]. The stepwise regression requires first calculating the value of the regression coefficient and the p -value and selecting a confidence interval. Then, we use the p -value as a criterion to decide which of these explanatory variables will be added or removed. Bozdogan [22] proposes to use the Genetic Algorithm (GA) method for variable selection, which is essentially a derivative of the selection-based method using a different information criterion. Selection-based methods need to explore all possible solutions to find the global optimal solution. Thus, the searching process is a combinational problem with NP-complexity.

From the perspective of penalties, we can directly perform parameter estimation to obtain the chosen variables by adjusting the coefficients corresponding to regularization. Moreover, if the penalty term itself is a convex problem, we can solve the problem exceptionally quickly using various efficient optimization methods.

However, we also need to emphasize that the selection and penalty views are two different perspectives on the same concept. They are not exclusive, and we can embrace both with equal success. For example, the combination of variables can also be described as a penalty optimization problem for L_0 . However, since the L_0 problem lacks efficient solving algorithms, this change in view does not help propose new algorithms for parameter estimation.

1.4 Information Criterion and Information Complexity

In 1973, Akaike first proposed an information criterion (called AIC) for trade-offs between the goodness of fit and the number of model parameters [4]. The number of model parameters reflects to some extent the complexity of the model, so the proposed criterion also represents a fine-tuning of the model’s generalization ability. According to [121], the AIC is asymptotically efficient in identifying the order of the model. Moreover, the AIC also provides an asymptotically unbiased estimator of the expected Kullback discrepancy [35]. However, we also note two significant problems with AIC. The first is that it is an inconsistent criterion [35], and the second is that the “number of effective parameters” is not clearly defined in many models [127]. For example, in a Bayesian model, if the model’s prior distributions are informative, then the parameter space is equivalent to imposing a specific limit, so the number of parameters no longer truly reflects the real complexity of the model.

The emergence of BIC [117] solves the inconsistency of AIC estimation results. Standing from the Bayesian point of view, BIC’s prior is assumed to be uniformly distributed [37]. However, the uniform distribution is no longer applicable when working with high dimensional data, so BIC is not advised. Nevertheless, Chen et al. believed that prior

distributions were not accurate when dealing with small- n -large- p problems, so they came up with EBIC [38], which proved to be very useful for similar cases in the future research [59].

ICOMP (Information Complexity Criterion) is an information criterion proposed by Bozdogan [16, 17, 21]. The problem that ICOMP tries to solve is the inability to determine the model complexity accurately. By default, AIC assumes no correlation between the variables, and they are all effective parameters. In contrast, ICOMP assumes the parameters interacting by default and calculates the model complexity by using its covariance matrix or Fisher information matrix (and its inverse). We introduce three statistical models in this thesis, accompanied by the derivation of the corresponding ICOMP formulas. We use ICOMP to perform model selection, throughout.

The organization of the rest of the dissertation is as follows. Chapter 2 describes how to achieve the effect of sparsity using bridge regression and derives ICOMP expressions to search the optimal hyperparameters. Chapter 3 analyzes the S&P 500 index data using the MIX-SPCR model, and the sparse model provides good explanatory power for high-dimensional time series data. The fourth chapter introduces nonlinear features through the kernel and then constructs Sparse Kernel Factor Analysis models in the feature space. Finally, in the last Chapter, we discuss future research on sparsity, model selection, and ICOMP.

Chapter 2

Selection of Variables in Sparse Bridge Regression Under Misspecification Using Information Complexity

This chapter proposes a novel Sparse Bridge Regression (SBR) model for subset selection of variables under model misspecification when the number of variables is much larger than the number of observations. The misspecification resistant information complexity criterion is developed as the fitness function for subset selection that allows both the robust statistical inference and proper interpretation of the results. To estimate the shrinkage factor q , tuning parameter λ , and model coefficients jointly, a hybridized Simulated Annealing and Particle Swarm Optimization algorithm, abbreviated as SA-PSO, is proposed. We introduce a smoothed covariance estimator to resolve the problem of ill-conditioned Gram matrix. The proposed new approach enables simultaneous model selection in Sparse Bridge Regression. It eliminates the costly and time-consuming cross-validation in high dimensions when the number of observations is limited. We show numerical examples using Monte Carlo simulation and a high-dimensional data to illustrate the model's sparsity and the efficiency of the proposed methodological and computational approach.

2.1 Introduction

In high-dimensional data modeling, a statistical technique’s success depends on identifying and selecting the most informative predictor variables. High dimensional data often have many redundant variables (or features) and a small number of relevant variables. It is crucial to identify and choose the relevant variables. Therefore, in high dimensions, the principle of parsimony or the Occam’s razor, that is, “*entities should not be multiplied without necessity*,” attributed to the English Franciscan monk Occam (1287 - 1347), is necessary to data modeling and machine learning to understand the underlying data generation process.

In the literature, many penalized regression models have been proposed and used for variable selection. It has been shown that penalized regression methods yield consistent variable selection and oracle parameter estimation under correct model specification. However, the robustness of the penalized regression methods has not been convincingly studied. Many of the penalized regression methods still depend on the Gram matrix $\hat{\Sigma} := (X'X)$ to estimate the model coefficients. Given that the dataset is high dimensional ($p \gg n$), the Gram matrix is singular, and we cannot calculate its inverse. In other words, because of the singularity of the Gram matrix, the regression model cannot provide reliable predictive inference.

Under singularity, one solution is to use a regularized covariance matrix instead of a raw Gram matrix to compute the coefficients of the variables. Therefore, further data-adaptive regularization is required on covariance matrix of the estimated coefficients to allow the selection of the best subset of predictors in the model. This approach achieves consistent variable selection, and allows dimension reduction further and eliminates irrelevant variables.

To address and resolve the existing problems in currently practiced penalized-regression modeling, we propose a new Sparse Bridge Regression (SBR) modeling approach under model misspecification based on the generalization of the work of [23, 28, 103, 85, 86]. The primary assumption in the literature is that the regression model is almost always correctly specified. For high-dimensional data, even if we assume Gaussianity on the model, this assumption may not be accurate in practice.

The sparsity of statistical models is important when the original dataset includes large variables, and statisticians cannot interpret the relationship between predictors and the response variable. To find the best subset of variables, we develop the misspecification resistant information complexity (ICOMP) criterion of Bozdogan, and Bozdogan and Howe [21, 27] for variable selection in SBR modeling. To derive the ICOMP criterion, we utilize the celebrated Fisher information matrix $\mathcal{F}$ in the Hessian form and the outer-product form $\mathcal{R}$ to obtain the robust covariance matrix. The robust covariance matrix is also known as a sandwich covariance matrix. We can use the sandwich covariance matrix in SBR model to guard against distributional misspecification and to obtain standard error estimates of the coefficients for further inference such as constructing confidence intervals.

There are two hyperparameters, the shrinkage parameter q , and the tuning parameter λ . We propose and use a hybridized Simulated Annealing and Particle Swarm Optimization algorithm, abbreviated as SA-PSO, to search the best combination of q and λ .

To further resolve the ill-conditioned Gram matrix $\hat{\Sigma} := (X'X)$ in high dimensional datasets, we introduce the idea of Smoothed Covariance Estimators (SCEs) of the predictor variables. By regularizing the commonly used Local Quadratic Approximation (LQA) algorithm, we provide a Regularized Local Quadratic Approximation (RLQA) algorithm using one of the SCE's to make the Gram matrix $\hat{\Sigma} := (X'X)$ well-behaved to estimate the coefficients of the SBR model. We then apply variable selection via the ICOMP criterion as our fitness function.

There are several forms of SCEs available in the literature, as discussed in Mohebbi et al. [103]. Here, we only use one type of SCEs for space consideration. However, our proposed approach is flexible to use other forms of SCEs.

In summary, our goals and contributions in this paper are multifaceted.

- We develop the misspecification-resistant information complexity ICOMP criterion for the selection of best variables in the SBR models.
- We propose a new hybridized Simulated Annealing-based Particle Swarm Optimization (SA-PSO) algorithm to jointly estimate the hyperparameters.

- To resolve the ill-conditioning of the Gram matrix $\hat{\Sigma} := (X'X)$, we introduce Smoothed Covariance Estimators (SCEs) of the predictor variables. We use SCE in the Local Quadratic Approximation (LQA) algorithm and provide a Regularized Local Quadratic Approximation (RLQA) algorithm in the SBR model.

We organize the rest of the chapter as follows. In Section 2.2, we present the statistical background of the SBR model. In particular, Section 2.2.1 has a brief introduction of the least-squares (LS) regression model under misspecification. As a continuation, we discuss the use of the SBR model under misspecification setting and provide the derived form of the estimated sandwich covariance matrix (Section 2.2.2). In Section 2.3, we discuss the general form of the misspecification resistant information complexity ICOMP criterion. Section 2.4 presents our newly proposed estimation of SBR model parameters simultaneously and provides computational algorithms. In Section 2.4.1, we present the hybridized SA-PSO algorithm to obtain the λ 's optimal values, the tuning parameter, and q , the shrinkage factor using the ICOMP criterion. In Section 2.4.2, we present the idea of Smoothed Covariance Estimators (SCEs), which has been largely used to resolve the “large p , small n ” problem in the SBR model. Section 2.4.3 presents the Regularized Local Quadratic Approximation (RLQA) algorithm and provides the necessary steps to obtain the estimated coefficients $\hat{\beta}_{Bridge}$ of the SBR model. By combining the results from Sections 2.4.1, 2.4.2, and 2.4.3, we score the misspecification resistant information complexity ICOMP criterion for the SBR model. In Section 2.5, we provide our numerical examples. In Section 2.5.1, we provide the structure of the Monte Carlo Simulation protocol. In Section 2.5.2, we conduct several Monte Carlo simulations to illustrate the performance of our proposed approach using the OAS smoothed covariance estimator. In Section 2.5.3, we compare the SBR model with that of Lasso using different simulated data sets as we vary the number of predictor variables from small 32 to large 10,000 variables. We provide the results for variable performance metrics such as Precision, Accuracy, Recall, and F-1 Score, respectively.

Further, in Section 2.6, a real-world benchmark dataset, Riboflavin (Vitamin B_2), is analyzed. This benchmark dataset has $n = 71$ observations and has $p = 4,088$ dimensional normalized gene-expressions (predictor variables). As can be seen, this benchmark data set is extremely undersized. Our results from both simulated numerical examples and real

Vitamin B_2 data show the efficiency and accuracy of the proposed SBR model using the misspecification resistant information complexity criterion. Section 2.7 concludes the paper with a discussion for future research.

2.2 Sparse Bridge Regression Model

2.2.1 A Brief Background: The Least Squares Regression Model

We first consider the usual linear regression model

$$y = X\beta + \varepsilon, \quad (2.1)$$

where $y = (y_1, y_2, \dots, y_n)$ is a response variable vector, $X_{n \times p} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]'$ represents non-stochastic predetermined variables. Further, we assume that ε is a $(n \times 1)$ vector of random noise term that is distributed as uncorrelated Gaussian. That is,

$$\varepsilon \sim N(0, \sigma^2 I). \quad (2.2)$$

The log likelihood of $(y_1, y_2, \dots, y_n)$, conditional on $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$, under Gaussian assumption is given by

$$\log L(\beta, \sigma^2) = -\frac{n}{2} \log(2\pi) - \frac{n}{2} \log(\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - \mathbf{x}_i' \beta). \quad (2.3)$$

The least-squares method is used to estimate the unknown β by minimizing the residual sum of squares

$$\hat{\beta}_{OLS} = \arg \min_{\beta} \|y - X\beta\|_2^2. \quad (2.4)$$

Under the Gauss-Markov assumptions, specifically, X is of full rank, the least-squares estimator of β is

$$\hat{\beta}_{OLS} = (X'X)^{-1} X'y \quad (2.5)$$

with $\mathbb{E}(\hat{\beta}_{OLS}) = \beta$ and the covariance matrix $\text{Cov}(\hat{\beta}_{OLS}) = \sigma^2 (X'X)^{-1}$.

If the entries of $\varepsilon_1 \dots \varepsilon_n$ of the noise vector ε are uncorrelated and have common variance σ^2 , then

$$\mathbb{E} \left\| X \left(\hat{\beta}_{OLS} - \beta \right) \right\|_2^2 = \sigma^2 p, \quad (2.6)$$

where $\| X \left(\hat{\beta}_{OLS} - \beta \right) \|_2^2 / n$ is referred to as the *prediction error*.

Because of the simplicity, unbiasedness, and having a minimum variance, the least square estimator is widely applied to estimate coefficients and the covariance. However, Godfrey [66] and White [148] pointed out that least squares estimator can be affected when the linear model is misspecified. Misspecification occurs when there is high collinearity, heteroskedasticity, and autocorrelation. Following the research of Huber [78], White [147], and Kauermann and Carroll [84], we use the “robust variance matrix” estimation to deal with misspecification.

We define the two forms of the Fisher information matrix to check the misspecification of classical regression models. From Bozdogan [23], these are given as follows.

The Hessian form of the estimated Fisher information matrix (in inner-product form) is

$$\hat{\mathcal{F}} = \begin{bmatrix} \frac{1}{\hat{\sigma}^2} (X'X) & 0 \\ 0' & \frac{n}{2\hat{\sigma}^4} \end{bmatrix}, \quad (2.7)$$

and the estimated outer-product form of the Fisher information matrix is

$$\hat{\mathcal{R}} = \begin{bmatrix} \frac{1}{\hat{\sigma}^4} X' D^2 X & X' \mathbf{1} \frac{Sk}{2\hat{\sigma}^3} \\ (X' \mathbf{1} \frac{Sk}{2\hat{\sigma}^3})' & \frac{(n-q)(Kt-1)}{4\hat{\sigma}^4} \end{bmatrix}. \quad (2.8)$$

where $D = \text{diag}(\hat{\varepsilon}_1, \dots, \hat{\varepsilon}_n)$, and Sk denotes the estimated residual skewness, and Kt is the estimated kurtosis. The symbol $\mathbf{1}$ is a $(n \times 1)$ vector of ones. Sk and Kt are given by

$$Sk = \frac{(\frac{1}{n} \sum_{i=1}^n \hat{\varepsilon}_i^3)}{\hat{\sigma}^3}, \text{ and} \quad (2.9)$$

$$Kt = \frac{(\frac{1}{n} \sum_{i=1}^n \hat{\varepsilon}_i^4)}{\hat{\sigma}^4}.$$

Under model misspecification, we denote the sandwich covariance matrix as

$$\text{Cov}(\beta, \sigma_k^2) = \mathcal{F}^{-1} \mathcal{R} \mathcal{F}^{-1}. \quad (2.10)$$

The consistent covariance estimator of the sandwich matrix is given by

$$\widehat{\text{Cov}}(\hat{\beta}, \hat{\sigma}^2)_{\text{Miss}} = \hat{\mathcal{F}}^{-1} \hat{\mathcal{R}} \hat{\mathcal{F}}^{-1}. \quad (2.11)$$

2.2.2 Sparse Bridge Regression Under Misspecification

One of the most popular classes of estimators is the L_q -regularized least squares (LQLS), known as the Bridge regression (BR). Frank and Friedman [60] proposed the Bridge regression (BR) with no solution.

The optimization of Bridge Regression is through minimizing both the residual sum of squares and the penalty term given by

$$\hat{\beta}_{\text{Bridge}} = \arg \min_{\beta} \left\{ \|y - X\beta\|^2 + \lambda \sum_{j=1}^p |\beta_j|^q \right\}, \quad (2.12)$$

where $\|\cdot\|$ is the L_2 norm of the vector, $\lambda > 0$ is the tuning parameter, $q > 0$ is the shrinkage or bridge parameter, and $\hat{\beta}_{\text{Bridge}}$ is called the bridge estimator.

When $0 < q \leq 1$, the Bridge regression (BR) tends to shrinkage the variable coefficients leading to variable selection and estimation simultaneously. It is therefore called the Sparse Bridge Regression (SBR), in which the feature of sparsity is the focus of the paper. In Figure 2.1 we show the structure of the penalty $\sum |\beta_j|^q$ for different values of $0 < q < 1$.

When the range of q is $(0, 1]$, we obtain a sparse model where the coefficients corresponding to many uncorrelated variables are zero. For example, [76] chooses to use the L_0 penalty term to select a few relevant variables from the data. However, we must note that the L_0 penalty term still lacks a general optimization algorithm, so it is computationally inefficient and challenging [105].

Lasso (Least Absolute Shrinkage and Selection Operator) uses the penalty term L_1 [134]. Because of the use of the LARS algorithm [53], its parameter estimation algorithm runs

very efficiently and has become a standard method for constructing sparse models today. However, according to [160], Lasso tends to select one from a set of highly correlated variables. Furthermore, Lasso's high-dimensional data analysis results have been controversial because of the prevalent *spurious correlation* problem in high-dimensional data [55, 58].

When $q = 2$, we obtain the Ridge Regression (RR) introduced by Hoerl and Kennard [72, 71]. Assuming a squared error loss function, $\hat{\beta}_{Ridge}$ minimizes:

$$\hat{\beta}_{Ridge} = \arg \min_{\beta} \|y - X\beta\|^2 + \lambda \|\beta\|_2^2. \quad (2.13)$$

Ridge Regression tends to retain all the predictors in the model. Thus the model cannot maintain the sparsity.

In general, one can set $q \in (0, \infty]$. However, when $q > 2$, bridge regression does not provide new insights for sparsity or variable selection.

As previously pointed out, the bridge regression belongs to a broad class of the penalized regression model that deserves close attention and further study for improvements.

During the past two decades, in the literature, we see that there has been active research and studies toward understanding the mechanism of the bridge regression since it was initially introduced by Frank and Friedman [60] without a provided solution. For example, Fu [61] was one of the first who studied the bridge estimator and developed an algorithm to solve the bridge estimator for any fixed $q \geq 1$. Fu [61] derived the covariance of the bridge estimator using the delta method. Knight and Fu [90] later identified the asymptotic properties of the bridge estimator when the number of predictors, p , is finite or fixed. These authors showed that, for $0 < q \leq 1$, under certain regularity conditions, when the actual value of the parameter is zero, the limiting distributions of bridge estimators can have positive probability mass at zero. Huang et al. [75] extended the work of Knight and Fu [90] to study the asymptotic properties of bridge estimators for $0 < q < 1$, given the situation that the number of variables p might be larger than the sample size n . Huang et al. [75] showed that for $0 < q < 1$, the bridge regression estimators can correctly select variables with nonzero coefficients. The conclusion is derived under the assumption that all the nonzero coefficients $(\hat{\beta}_j \neq 0)$ share the same asymptotic distribution and all the predictors with zero

coefficients were known in advance. Thus these studies showed that the bridge estimator for $0 < q \leq 1$ provides a way to combine variable selection and parameter estimation at the same time. There are many other studies published on bridge regression. For space considerations, we will not cite all these research papers but do acknowledge their contributions.

Coupled with the misspecification problem in the SBR model, when we have wide data sets, that is, when $p \gg n$ ("large p , small n "), to our knowledge, there is no research work carried in the SBR model.

Under the misspecified SBR model, following the work of Bozdogan [21, 23] and generalizing the results of Konishi et al. [92], and Kawano [86], we provide the analytical form of the *sandwich covariance matrix* for the SBR model to guard against misspecification. We are only interested in those cases where the probabilistic assumption is away from the Gaussian distribution under the SBR framework.

Let $\mathcal{A} = \{j : \hat{\beta}_j \neq 0\}$ be an active set of the coefficients β , and $\mathcal{A}_C = \{j : \hat{\beta}_j = 0\}$ be the non-active set or number of coefficients equal to zero.

From Kawano [86], the estimated Fisher information $\hat{\mathcal{F}}$, in Hessian form, is a $(|\mathcal{A}| + 1) \times (|\mathcal{A}| + 1)$ matrix given by

$$\hat{\mathcal{F}} = \frac{1}{n\hat{\sigma}^2} \begin{bmatrix} X'_{\mathcal{A}}X_{\mathcal{A}} + n\lambda\hat{\sigma}^2q(q-1)K_1 & \frac{1}{\hat{\sigma}^2}X'_{\mathcal{A}}D\mathbf{1}_n \\ \frac{1}{\hat{\sigma}^2}\mathbf{1}'_nDX_{\mathcal{A}} & \frac{n}{2\hat{\sigma}^2} \end{bmatrix}, \quad (2.14)$$

where $D = \text{diag}(\hat{\varepsilon}_1, \dots, \hat{\varepsilon}_n)$ a $n \times n$, $K_1 = \text{diag}(|\hat{\beta}_{k_1}|^{q-1}/2, \dots, |\hat{\beta}_{k_r}|^{q-r}/2)$ is a $|\mathcal{A}| \times |\mathcal{A}|$ diagonal matrices.

The estimated Fisher information matrix in outer-product form is given by

$$\hat{\mathcal{R}} = \frac{1}{n\hat{\sigma}^2} \begin{bmatrix} \frac{1}{\hat{\sigma}^2}X'_{\mathcal{A}}D^2X_{\mathcal{A}} - \lambda K_2\mathbf{1}_n^TDX_{\mathcal{A}} & \frac{1}{2\hat{\sigma}^4}X'_{\mathcal{A}}D^3\mathbf{1}_n - \frac{1}{2\hat{\sigma}^2}X'_{\mathcal{A}}D\mathbf{1}_n \\ \frac{1}{2\hat{\sigma}^4}\mathbf{1}'_nD^3X_{\mathcal{A}} - \frac{1}{2\hat{\sigma}^2}\mathbf{1}_nDX_{\mathcal{A}} & \frac{1}{4\hat{\sigma}^6}\mathbf{1}'_nD^4\mathbf{1}_n - \frac{n}{4\hat{\sigma}^2} \end{bmatrix}, \quad (2.15)$$

where, K_2 is also a $|\mathcal{A}| \times |\mathcal{A}|$ diagonal matrix given by

$$K_2 = \text{diag}\left(\frac{|\hat{\beta}_{k_1}|^{q-1}\text{sgn}(\hat{\beta}_{k_1})}{2}, \dots, \frac{|\hat{\beta}_{k_r}|^{q-1}\text{sgn}(\hat{\beta}_{k_r})}{2}\right), \quad (2.16)$$

and where sgn denotes the sign function.

Following the work of Huber [78] and White [147], we construct the *sandwich covariance matrix* for the SBR model given by

$$\widehat{\text{Cov}}(\hat{\beta}, \hat{\sigma}^2)_{\text{MISS}} = \hat{\mathcal{F}}^{-1} \hat{\mathcal{R}} \hat{\mathcal{F}}^{-1}, \quad (2.17)$$

which is the correct form of the covariance matrix under misspecification and not just $\hat{\mathcal{F}}^{-1}$. Note that using the sandwich covariance matrix in equation (2.11) for the usual regression model and the equation in (2.17) for the SBR model penalizes presence of skewness and kurtosis in the data set.

2.3 Misspecification Resistant Information Complexity

The issue of model misspecification has been a challenge for statisticians during the model fitting and selection process. In order to drive valid variable selection, we generalize the information complexity criterion given that the Gaussian assumption is not valid for the original input data. We formalize the work of Bozdogan, Koc and Bozdogan [21, 23, 91] to the case of SBR model under misspecification.

$$\begin{aligned} \text{ICOMP}(\text{SBR Model})_{\text{MISS}} &= -2 \log L(\hat{\beta}, \hat{\sigma}^2 | y) + 2C_1(\widehat{\text{Cov}}(\hat{\beta}, \hat{\sigma}^2)_{\text{MISS}}) \\ &= -2 \log L(\hat{\beta}, \hat{\sigma}^2 | y) + 2C_1(\hat{\mathcal{F}}^{-1} \hat{\mathcal{R}} \hat{\mathcal{F}}^{-1}). \end{aligned} \quad (2.18)$$

$L(\hat{\beta}, \hat{\sigma}^2 | y)$ is the maximized likelihood function of the probability density function $f(x | \beta, \sigma^2)$. The second term $C_1(\widehat{\text{Cov}}(\hat{\beta}, \hat{\sigma}^2)_{\text{MISS}})$ denotes the maximal information-theoretic complexity of the sandwich covariance matrix $\widehat{\text{Cov}}(\hat{\beta}, \hat{\sigma}^2)_{\text{MISS}} = \hat{\mathcal{F}}^{-1} \hat{\mathcal{R}} \hat{\mathcal{F}}^{-1}$ given by

$$C_1(\widehat{\text{Cov}}(\hat{\theta})_{\text{MISS}}) := \frac{s}{2} \log \left(\frac{\text{tr} \left((\widehat{\text{Cov}}(\hat{\beta}, \hat{\sigma}^2)_{\text{MISS}}) \right)}{s} \right) - \frac{1}{2} \log | (\widehat{\text{Cov}}(\hat{\beta}, \hat{\sigma}^2)_{\text{MISS}}) |, \quad (2.19)$$

where $s := \text{rank} \left(\widehat{\text{Cov}}(\hat{\beta}, \hat{\sigma}^2)_{\text{Miss}} \right)$. Then we have

$$\text{ICOMP}(\text{SBR Model})_{\text{Correct}} = -2 \log L(\hat{\beta}, \hat{\sigma}^2 | y) + 2C_1(\hat{\mathcal{F}}^{-1}). \quad (2.20)$$

where $\mathcal{F}^{-1}$ is from equation (2.14). Note that when

$$\text{ICOMP}(\text{SBR Model})_{\text{Miss}} \neq \text{ICOMP}(\text{SBR Model})_{\text{Correct}}, \quad (2.21)$$

or, equivalently, if

$$C_1(\widehat{\text{Cov}}(\hat{\theta})_{\text{Miss}}) \neq C_1(\hat{\mathcal{F}}^{-1}), \quad (2.22)$$

then we say that the two models are misspecified. There are other forms of ICOMP. These different forms of information complexity are derived from both the likelihood function and the covariance matrix of the estimated parameters of the proposed model. The various derivation of ICOMP also takes consideration of datasets from finite sampling distributions and Bayesian justification, widely used in linear and nonlinear models. We refer the readers more on this and other applications of ICOMP to Bozdogan [26].

Throughout the following of the paper, we use and score $\text{ICOMP}(\text{Model})_{\text{Miss}}$ as our fitness function for variable selection in the SBR model under misspecification.

2.4 New Proposed Estimation of the Sparse Bridge Regression Model Parameters

In this section, we present our new proposed estimation of the SBR model parameters to answer the following questions.

- How do we efficiently estimate the regularization (or tuning) parameter λ ?
- What is the optimal value of bridge (or shrinkage) parameter $q \in (0, 1]$?
- How do we simultaneously estimate λ , q , and the SBR coefficients β ?

Researchers typically utilize the grid search method or cross-validation method to search for hyperparameters. Here, we present an alternative search method that combines simulated annealing and particle swarm optimization methods to quickly and efficiently estimate the values of the hyperparameters in the SBR model.

2.4.1 Simulated Annealing and Particle Swarm Optimization: SA-PSO

In this section, we propose an alternative approach to select the optimal values of the regularization (or tuning) parameter λ , and bridge (or shrinkage) parameter q , simultaneously in the penalty function of the SBR model by hybridizing Simulated Annealing (SA) and Particle Swarm Optimization (PSO), abbreviated as SA-PSO. Simulated Annealing (SA) is a method for solving unconstrained and bound-constrained optimization problems. Particle Swarm Optimization (PSO), on the other hand, is a new method of evolutionary computation. Since both SA and PSO are highly efficient optimization algorithms, we borrow the strengths from these two algorithms and, by hybridizing them as SA-PSO, one can create a new family of optimization algorithms for function optimization in high dimensions. This proposed approach enables us to objectively identify appropriate values of the adjusted hyperparameters of the SBR model.

PSO can find globally optimal results, and SA can find local search optimization. By hybridization of the two, the SA-PSO algorithm achieves a balance between global and local optimization. This idea of a hybridized algorithm was introduced by Wang and Li [145] and further developed by Zhan et al. [156]. Hybridization provides a better global search efficiency that narrows down the search space and speeds up the convergence rate.

The full implementation of SA-PSO is provided in Algorithms 2, 3 and 4. Algorithm 2 shows the initialization of SA-PSO analogs to the genetic algorithm (GA) and sets up the parameters in Table 2.1. Algorithms 3 and 4 are the main bodies of SA-PSO. For each generation of particles, we select individuals with the best performance and store the coordinates (estimates of each variable coefficient). By comparing the best candidate with other particles, we calculate the velocity to update each particle's position in the next

generation. The position of each candidate would be updated randomly to ensure the searching result is a globally optimal rather than locally optimal result.

Throughout the numerical examples in this paper, we set the parameter values of SA-PSO algorithm given in Table 2.1 to optimally learn the parameter values of the SBR model by minimizing $\text{ICOMP}_{\text{Miss}}$. In Table 2.1, the user determines the size of the population and the number of generations. However, c_1 and c_2 are recommended by Clerc and Kennedy [42] based on repeated experimentation. For further details, we refer the readers to their paper.

2.4.2 Smoothed Covariance Estimators (SCEs)

The last two decades have witnessed an increase in research on penalized regression methods, including the bridge regression, due to the emergence of high-dimensional data with different structures and increasing complexity with a small number of observations and with high-dimensions. Such situations created new challenges to existing statistical models and covariance matrix estimation when $p \gg n$. In these scenarios, obtaining the inverse covariance matrix under singularity conditions became a severe challenge to statistical modelers.

Many smoothed covariance estimators have been proposed, including Ledoit-Wolf (LW) estimator by Ledoit and Wolf [94] and the optimal shrinkage covariance estimator by Ollila [106]. Chen et al. [39] improved the LW estimator by conditioning on a sufficient statistic, and proposed the oracle approximating shrinkage (OAS) estimator. Bozdogan and Pamukcu [28] and Mohebbi et al. [103] gave a list of Smoothed Covariance Estimators. For space consideration, in this paper, we will use only the OAS estimator to make the Gram matrix well-conditioned for under-sample data.

Let $\hat{\mathbf{S}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^T$ the gram matrix, then

$$\hat{\rho}_{OAS} = \min \left(1, \frac{(1 - 2/p) \text{tr}(\hat{\mathbf{S}}^2) + \text{tr}^2(\hat{\mathbf{S}})}{(n + 1 - \frac{2}{p})(\text{tr}(\hat{\mathbf{S}}^2) - \text{tr}^2(\hat{\mathbf{S}})/p)} \right). \quad (2.23)$$

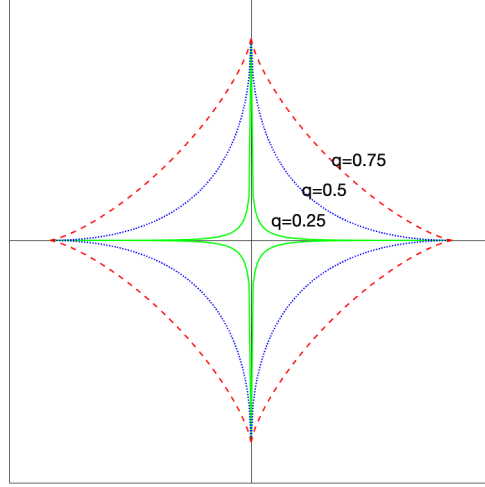


Figure 2.1: Plot of the penalty $\sum_{j=1}^p |\beta_j|^q$ for different values of $0 < q < 1$.

Table 2.1: Parameters of SA-based Particle Swarm Optimization Algorithm.

Parameters	Values
Fitness Function	ICOMP _{MISS}
Size of Population (N)	50
c_1	2.05
c_2	2.05
Number of Generations	50
Dimension of Inputs	2

Chen et al. [39] proved that $\hat{\rho}_{OAS}$ is bounded within $(0, 1]$. Then, the final form of the SCE is given by

$$\hat{\Sigma}_{SCE} = (1 - \hat{\rho}_{OAS})\hat{\mathbf{S}} + \hat{\rho}_{OAS}\frac{tr(\hat{\mathbf{S}})}{p}I_p. \quad (2.24)$$

We use $\hat{\Sigma}_{SCE}$ in our computational results.

2.4.3 Regularized Local Quadratic Approximation (RLQA)

When we select values of λ and q , we must also find an efficient algorithm to calculate regression coefficients. When $0 < q \leq 1$, the sparse bridge penalty is non-concave, which means traditional convex optimization could not be applied. Fan and Li [55] proposed the Local Quadratic Approximation (LQA) method to estimate the model coefficients. The method can determine the estimated coefficients within a limited number of steps by repeatedly using the Gram matrix. However, the Gram matrix is not invertible under singularity situations. As a result, the LQA might not be reliable when $p \gg n$.

We propose replacing the Gram matrix with one of the smoothed covariance estimators to regularize the Local Quadratic Approximation (LQA) method. We call this the Regularized Local Quadratic Approximation (RLQA) method. Here, we only considered using the oracle approximating shrinkage (OAS) estimator. Furthermore, our computational method is flexible enough to study the results of the other SCEs, which can be of further study in the SBR model.

In what follows, we show the steps of RLQA by modifying the steps of LQA in Fan and Li [55] and Kawano [86] to obtain the estimated values of the SBR model coefficients.

The main purpose of the algorithm is to transform a non-concave optimization problem to a concave one. Under some mild conditions, we initialize the estimated values of coefficients first, which are denoted as $\beta_{OLD} = (\beta_{1,OLD}, \beta_{2,OLD}, \dots, \beta_{p,OLD})$. Then we update the values of β_{NEW} by using β_{OLD} and partial derivatives.

We can write the parameter update function as

$$\hat{\beta}_{NEW} = (\hat{\Sigma}_{SCE} + \Sigma_{\lambda,q}(\hat{\beta}_{OLD}))^{-1}\mathbf{X}'\mathbf{y}. \quad (2.25)$$

In equation (2.25), once a coefficient is set to zero, then in all the subsequent steps, the estimated values would be zero. Following Kawano [86], our proposed Regularized Local Quadratic Approximation (RLQA) steps are outlined in algorithm 1.

We define $\text{ICOMP}_{\text{MISS}}$ for the SBR model given by

$$\text{ICOMP}_{\text{MISS}}(\text{SBR}) = -2\log L(\hat{\theta}|y) + 2C_1(\widehat{\text{Cov}}(\hat{\sigma}^2, \hat{\beta})_{\text{MISS}}), \quad (2.26)$$

where both the likelihood function and sandwich covariance matrix in closed-form expression are given in Section 2.2.

To summarize, we use the SA-PSO algorithm and the ICOMP criteria to find hyperparameters. Then, with the λ and q in hand, we can use the RLQA algorithm to estimate the parameter. As a sparse model, SBR selects variables by considering only those parameters with non-zero coefficients.

In our investigation, there are not many statistical criteria in the literature that handles misspecification. However, existing publications on this area include the Takeuchi [130] information criterion (TIC), AIC_T , also known as GAIC discussed in Kawano [85, 86], where the $2\text{tr}(\hat{\mathcal{F}}^{-1}\hat{\mathcal{R}})$ is used in the penalty term of GAIC and Demirkaya et al. [46] in high dimensional setting.

Next, we show our numerical examples.

2.5 Monte Carlo Simulation Results

In this section, we compare the performance of the SA-PSO based SBR model to the Lasso regression using simulated datasets. For Lasso, there are multiple methods for choosing λ adaptively from the data. These methods include, but are not limited to: 1) performing grid-search, and 2) using $(\min(\text{eigenvalues}(X'X)))^{-1}$ values and other parameter tuning techniques. In our computations, we use the OAS estimator in equation (2.24) for tuning λ values in Lasso. All our computations are carried out using a newly developed MATLAB[®] computational routines.

2.5.1 Simulation Protocol

In our simulations, we focus on comparing performances on variable selection under different scenarios of complex data structures. In order to create different levels of collinearity among the variables, we use the following simulation protocol. This simulation protocol considers both pairwise correlations among actual variables and autoregressive (AR) structure among irrelevant variables. Roozbeh [111] and Mohebbi et al. [103] suggested and used this simulation protocol to obtain X_{ij} , reflecting the convoluted collinear relationships among variables for $j = 1, 2, \dots, p$ and $i = 1, 2, \dots, n$ observations. We denote independent explanatory variables by Z_{ij} as follows.

$$X_{ij} = (1 - \gamma^2)^{\frac{1}{2}} Z_{ij} + \gamma Z_{ip}, \quad (2.27)$$

where Z_{ij} has zero mean and unit variance. To answer the question of whether SBR outperforms the Lasso in variable selection, given the complicated data structures, we fix a set of variables ($X^{(1)}$) as the ground truth and $X^{(2)}$ as redundant set of variables in

$$y = X^{(1)}\beta_1 + X^{(2)}\beta_2 + \epsilon, \quad (2.28)$$

$$\epsilon \sim N_n(0, \sigma^2 \Sigma), \quad (2.29)$$

$$\Sigma_{i,j} = \exp(-7|i - j|). \quad (2.30)$$

The true coefficient values is set as $\beta_1 = (-1.25, 1, 2.5, 4, -3, -5)^T$, where $p_1 = 6$ and $n = 100$ observations. To create the sparse model with redundant variables, $p_2 = p - p_1$, β_2 is sampled from multivariate Gaussian distribution $N_{p_2}(0, 0.01I_p)$. In our simulation, we vary the total number of predictor variable sets $p = 32$ to $10,000$.

2.5.2 Effects of Smoothed Covariance Estimator

Next, we analyze the effects of oracle approximating shrinkage (OAS) estimator by answering two questions,

1. How does OAS help adjust the eigenvalues of the Gram matrix when $p \gg n$?

2. Can OAS stabilize the calculation of SBR coefficients when $p \gg n$?

We first illustrate the eigenvalues of the Gram matrix with and without utilizing the OAS smoothed covariance estimator. In Figure 2.2, we see that the OAS method smooths the eigenvalue and regularizes them. In other words, OAS is a data-adaptive method for dealing with ill-conditioned Gram matrix $\hat{\Sigma} := (X'X)$. One can use other SCEs as well.

Figure 2.3 shows the visualization of the SBR coefficients given different values of λ . The first panel displays the results using the usual LQA algorithm. The second panel shows how the RLQA algorithm performs with the usage of the OAS smoothed covariance estimator. We can see that the change of coefficient values become smoother when using the OAS, and estimated results are stable when $\log(\lambda) \leq -4$.

In Figure 2.4, we show the results of our approach and demonstrate why one needs to regularize further the LQA algorithm employing SCE, using the data generated from our simulation protocol. Looking at Figure 2.4, we can see that the utilization of the usual Gram matrix in the SBR model, adopted by many authors, exhibits an erratic behavior and is not smooth because there are singularities in the Gram matrix. However, when we use one of the smoothed covariance estimators and its shrinkage intensity, $\hat{\rho}$, for example, using the $\hat{\rho}$ of OAS, we see much improved regularized results. Therefore, we propose and recommend the usage of the SCEs within the SBR model for the best subset selection of the predictors.

2.5.3 Comparative Performance of Sparse Bridge Regression with Lasso

In this section, we compare and study the performance metrics of variable selection between the SBR model and the Lasso. To this end, we generate multiple datasets using the simulation protocol in Section 2.5.1. We fix the ground truth variables in each experiment and change the number of irrelevant variables $X^{(2)}$ through different simulated datasets.

The Lasso estimators are calculated using cross validation (picking λ with the minimum mean square error). For SBR, we use the SA-PSO algorithm to find the optimal solution of the tuning parameters (λ, q) . RLQA is adopted to calculate the SBR coefficient values. Table 2.1 lists the parameters of the SA-PSO algorithm.

Algorithm 1 Regularized Local Quadratic Approximation (RLQA).

- 1: Input: $\lambda > 0$, $0 < q < 1$, X , y , $\delta = 10^{-4}$, $\gamma = 10^{-5}$, **maxSteps** = 1000
 - 2: $\hat{\beta}_{\text{OLD}} = \hat{\beta}_{\text{NEW}} = (\hat{\Sigma}_{SCE} + n\gamma I_p)^{-1} \mathbf{X}' \mathbf{y}$
 $\triangleright$ To improve the calculation efficiency, the initialized estimated coefficients should be close to the MLE estimation by [55].
 - 3: $\hat{\sigma}_{\text{OLD}}^2 = \hat{\sigma}_{\text{NEW}}^2 = \frac{1}{n} (\mathbf{y} - \mathbf{X} \hat{\beta}_{\text{NEW}})' (\mathbf{y} - \mathbf{X} \hat{\beta}_{\text{NEW}})$
 - 4: **for** $i = 1 : \text{maxSteps}$ **do**
 - 5: $\Sigma_{\lambda,q}(\hat{\beta}_{\text{OLD}}) = \text{diag}(n\lambda\hat{\sigma}_{\text{OLD}}^2 q |\hat{\beta}_{1,\text{OLD}}|^{q-2}/4, \dots, n\lambda\hat{\sigma}_{\text{OLD}}^2 q |\hat{\beta}_{p,\text{OLD}}|^{q-2}/4)$
 - 6: $\hat{\beta}_{\text{NEW}} = (\hat{\Sigma}_{SCE} + \Sigma_{\lambda,q}(\hat{\beta}_{\text{OLD}}))^{-1} \mathbf{X}' \mathbf{y}$
 $\triangleright \hat{\Sigma}_{SCE}$ represents the smoothed covariance estimators
 - 7: $\sigma_{\text{NEW}}^2 = \frac{1}{n} (\mathbf{y} - \mathbf{X} \hat{\beta}_{\text{NEW}})' (\mathbf{y} - \mathbf{X} \hat{\beta}_{\text{NEW}})$
 - 8: **if** $\max(|\hat{\beta}_{\text{NEW}} - \hat{\beta}_{\text{OLD}}|) < \delta$ **then**
 - 9: Break
 - 10: **end if**
 - 11: **end for**
 - 12: **return** $\hat{\beta}_{\text{NEW}}$
-

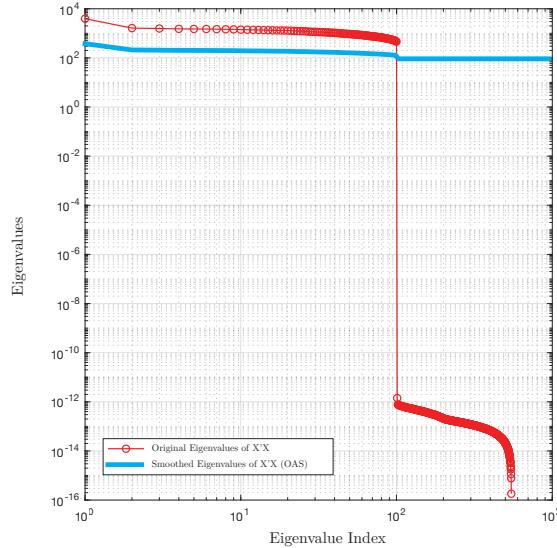
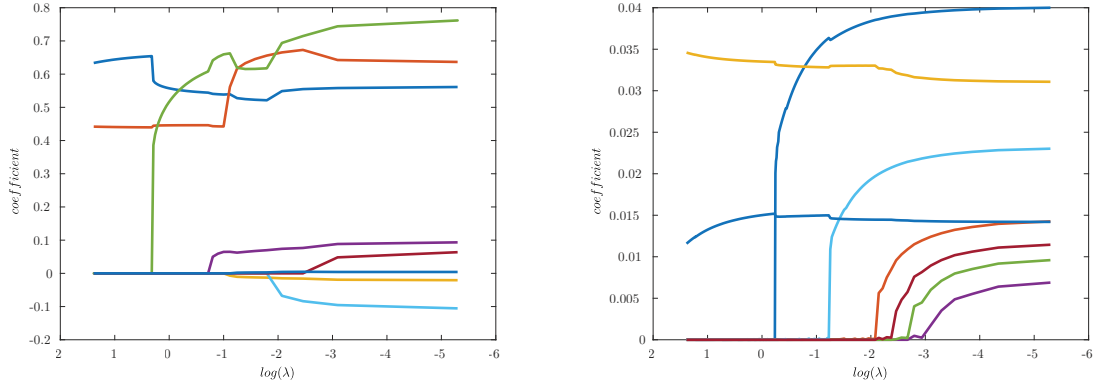


Figure 2.2: Comparison of eigenvalues w/o OAS smoothed covariance estimator of the Gram matrix ($X'X$) for the simulated dataset with $\gamma = 0.3$, $n = 100$ and $p = 1000$.



(a) Coefficients of variables from original $(X'X)$.

(b) Coefficients of variables with OAS.

Figure 2.3: Estimated coefficients w/o OAS smoothed covariance estimators in SBR using LQA algorithm.

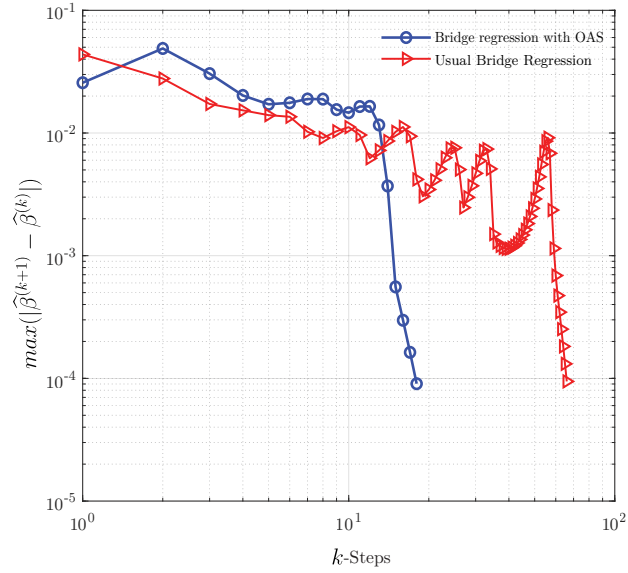


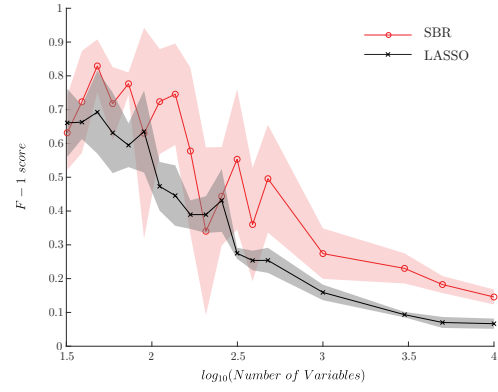
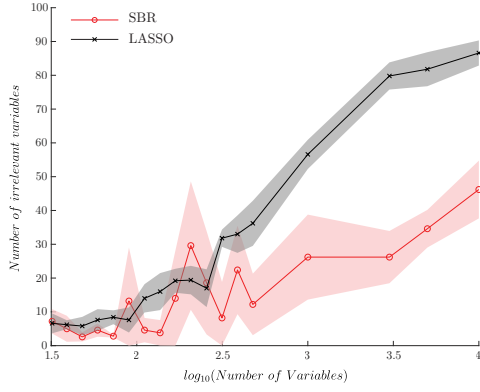
Figure 2.4: The behavior of $\max(\hat{\beta}^{(k+1)} - \hat{\beta}^{(k)}) < \delta = 10^{-4}$.

We use four classification performance metrics, 1) Precision, 2) Accuracy, 3) Recall, and 4) F-1 Score to evaluate the classification performance of the proposed SBR model and Lasso. The accuracy measures the proportion of correctly selected variables among all the selected factors. Recall stands for choosing all the relevant variables within the dataset. To seek a balance between precision and recall, we include the F-1 score in our comparison.

Results of the number of incorrectly selected variables (false positive) under SBR and Lasso are presented in Figure 2.5a. The gray color band indicates a 1SE (one standard error) region of Lasso, and the red color band indicates a 1SE (one standard error) region of the SBR model. A striking feature of Figure 2.5a is that, compared with Lasso, SBR selects fewer irrelevant variables as the number of irrelevant variable sets grows. When the number of variables increases to more than the number of observations ($n = 100$), Lasso selects two times the irrelevant variables than does the SBR model. In other words, SBR's performance is two times better than Lasso. The comparison clearly shows that despite its popularity, when $p \gg n$, the Lasso tends to over-shrink significant coefficients and include unimportant variables in order to compensate for this over-shrinkage [55].

In terms of F-1 scores shown in Figure 2.5b, both Lasso and SBR decrease as the number of variables increases. When it comes to 10,000 variables with 100 observations in the dataset, the covariance matrix becomes highly ill-conditioned. Even though the penalized regression models (Lasso and SBR model) can still select the real coefficients, both models select some unnecessary variables in the final results. Since SBR has a more robust performance in false-positive detection (FPD), the F-1 scores of SBR also outperform the Lasso results.

In summary, from the results in Table 2.2, we see that the SA-PSO based SBR outperforms the Lasso in terms of false-positive rates and F-1 scores for all eighteen simulations. In contrast, the SBR model does not perform well as the Lasso model in the recall. The recall value is used to control the number of False-negative variables. The SBR model tends to discard variables when it is uncertain whether to include a variable in the model, but Lasso assigns non-zero coefficients.



(a) Number of incorrectly selected variables (false positive) under SBR and Lasso. (b) Comparison of F-1 scores of SBR model with Lasso.

Figure 2.5: Comparison of Sparse Bridge Regression (SBR) with Lasso.

Table 2.2: Comparison of SBR with Lasso as the number of variable sets increases.

Number of variables	Precision		Accuracy		Recall		F-1 score	
	Lasso	SBR	Lasso	SBR	Lasso	SBR	Lasso	SBR
32	0.502	0.485	0.794	0.769	1.000	0.967	0.661	0.632
39	0.498	0.603	0.841	0.867	1.000	0.967	0.663	0.723
48	0.545	0.716	0.879	0.946	1.000	1.000	0.693	0.829
59	0.474	0.576	0.871	0.919	1.000	0.967	0.631	0.718
73	0.426	0.668	0.885	0.956	1.000	0.933	0.595	0.777
90	0.477	0.625	0.916	0.844	1.000	0.867	0.635	0.630
111	0.313	0.618	0.874	0.955	1.000	0.933	0.473	0.724
137	0.291	0.666	0.883	0.968	1.000	0.900	0.446	0.746
168	0.243	0.456	0.886	0.917	1.000	1.000	0.389	0.578
207	0.243	0.268	0.906	0.852	1.000	0.833	0.389	0.340
256	0.280	0.316	0.934	0.926	1.000	0.933	0.431	0.444
315	0.159	0.553	0.899	0.968	1.000	0.667	0.275	0.553
388	0.147	0.250	0.914	0.940	0.933	0.833	0.254	0.360
477	0.146	0.405	0.924	0.972	1.000	0.833	0.254	0.496
1000	0.087	0.172	0.943	0.972	0.900	0.767	0.159	0.274
3000	0.050	0.141	0.973	0.991	0.700	0.667	0.093	0.230
5000	0.038	0.106	0.983	0.993	0.533	0.667	0.070	0.183
10000	0.036	0.082	0.991	0.995	0.533	0.667	0.067	0.146

2.6 Real Data Example: Riboflavin (Vitamin B2) Dataset

In order to demonstrate the performance of SA-PSO based SBR model and to show the importance of tuning parameters on model selection, the Riboflavin data is analyzed in detail to identify the relationship between production rate and thousands of gene expression measures. Tännler et al. [131] originally collected the Riboflavin dataset. Furthermore, Bühlmann et al. [34] made it publicly available for research purposes.

Many researchers analyzed the Riboflavin production data as a benchmark dataset over the past several years. This dataset has $n = 71$ observations, and each observation includes $p = 4,088$ gene expressions (predictor variables). The riboflavin production rate in the logarithm scale is the response variable. Our primary purpose is to identify a subset of genes that is relevant to predict the Riboflavin production rate. We also want to maintain the sparsity of the model, ensuring the interpretation of gene expression data is clean and bright.

We now elaborate on the selection performance of SA-PSO based SBR for the Riboflavin production data.

Our analysis shows that the SA-PSO algorithm can search the model space and identifies the optimal values of (λ, q) automatically and efficiently. Figure 2.7 illustrates the model space of SBR for the Riboflavin production data. The first panel is the surface plot of $\text{ICOMP}_{\text{Miss}}$ values over $(\log(\lambda), q)$. Specifically, a ridge-like area is formed when $\log(\lambda) \in [-7, -3]$. SA-PSO algorithm generates particles for each generation (marked as blue circles in Figure 2.7). It develops that particle swarms left behind ridges and spread with fluidity, as illustrated in the black line in both panels. The second panel shows the top-down view of the surface plot.

SA-PSO based SBR represents the right balance between precision and recall and identifies 54 genes. Compared with other penalized regression models, such as the adaptive elastic net (AEN) based regression model reported by Mohebbi et al. [103]. In Table 2.3, we select fewer variables compare to EN, AEN models.

In Figure 2.8, we show our predicted y values against the real values in panel (a) and the QQ-plot of the residuals in panel (b). We can observe the shrinkage effects because some predicted values get closer to zero compared with real values. However, despite this shrinkage effect, the predicted values are still close to the original values. The QQ-plot of residuals also confirmed this observation.

Figure 2.6 shows the efficiency of the SA-PSO algorithm based on the $\text{ICOMP}_{\text{Miss}}$ results. Compared with the grid-search method, SA-PSO is data-adaptive in searching for the optimal value of λ and q . As a by-product of SBR model, we present Figure 2.9 illustrating the hierarchical structure of selected genes based on the correlation matrix of the 54 relevant genes. This result can be used in building classification models.

2.7 Conclusions and Discussion

In this work, we introduced and proposed a novel Sparse Bridge Regression (SBR) model for subset selection of best predictors when the number of observations is less than the number of predictors. We developed the misspecification resistant information complexity ($\text{ICOMP}_{\text{Miss}}$) criterion as a fitness function for subset selection to allow robust statistical inference and their proper interpretations for the Sparse Bridge Regression (SBR) model.

In the literature, researchers mostly ignore the model misspecification issue. To resolve this issue, we provided the closed-form expression of the robust covariance matrix by using the derived form of the celebrated Fisher information matrix $\mathcal{F}$ in the Hessian form, and the outer-product form $\mathcal{R}$. This robust covariance matrix is useful to calculate the standard errors of the model parameters for further frequentist inference.

We also proposed and used a novel hybridized Simulated Annealing (SA) and Particle Swarm Optimization (PSO) algorithm, SA-PSO. We showed that this new approach enables us to determine appropriate values of the estimated tuning parameter λ and the shrinkage parameter q jointly by eliminating the costly and time-consuming cross-validation (CV) in such problems.

We introduced one of the Smoothed Covariance Estimators (SCEs), namely the OAS covariance estimator, to regularize the commonly used Local Quadratic Approximation

Table 2.3: Variable selection for riboflavin production data.

Model	Covariance Estimator	No. of selected variables
EN, Mohebbi et al.[103]	MLE	68
AEN, Mohebbi et al.[103]	OAS	74
AEN, Mohebbi et al.[103]	BCSE	62
Sparse Bridge Regression (SBR)	OAS	54

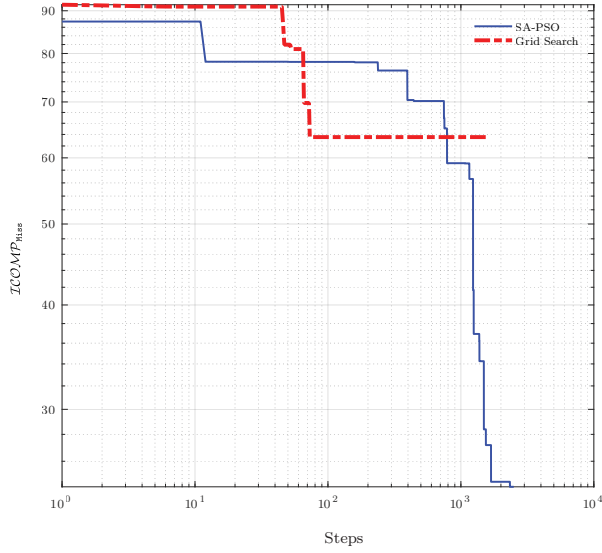


Figure 2.6: Comparison of SA-PSO and grid-search results for Riboflavin data for $0 < q \leq 1$, $0 < \lambda \leq 4$. This comparison is labeled on a 40×40 grid search and SA-PSO with population size 50 and 50 generations.

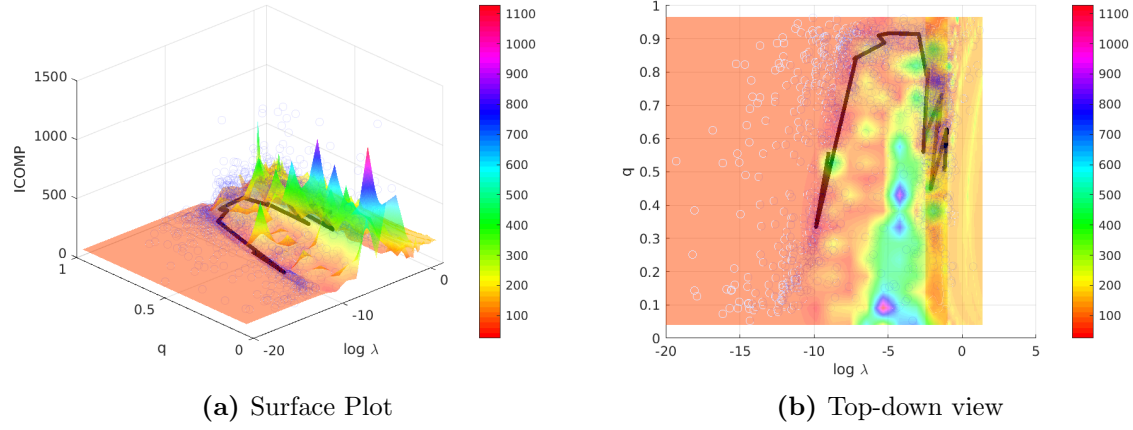


Figure 2.7: ICOMP values of the entire model space from SBR of Riboflavin data using SA-PSO algorithm. The model space is illustrated over $(\log(\lambda), q)$. The black solid line represents the path of the best particle over each generation of the SA-PSO algorithm.

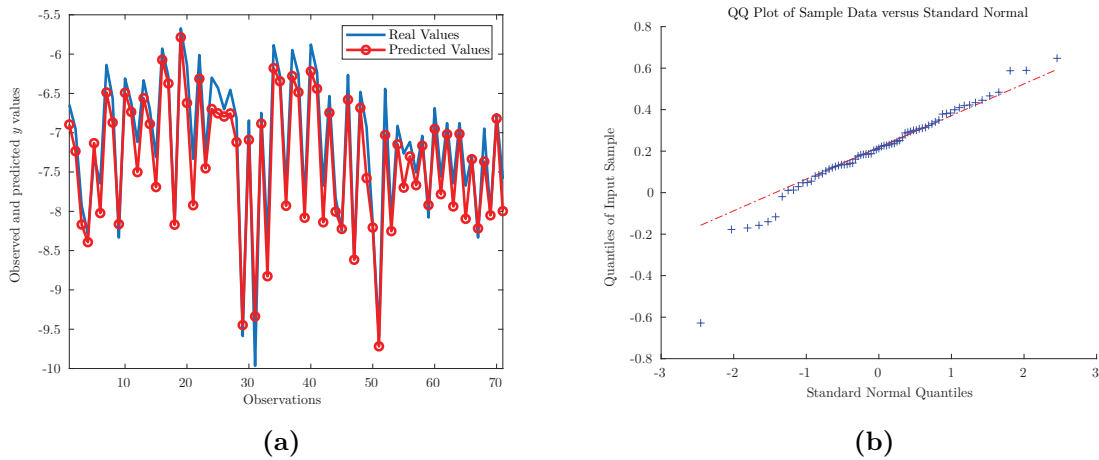


Figure 2.8: Plot of predicted y values against real values and QQ-plot of the residuals.

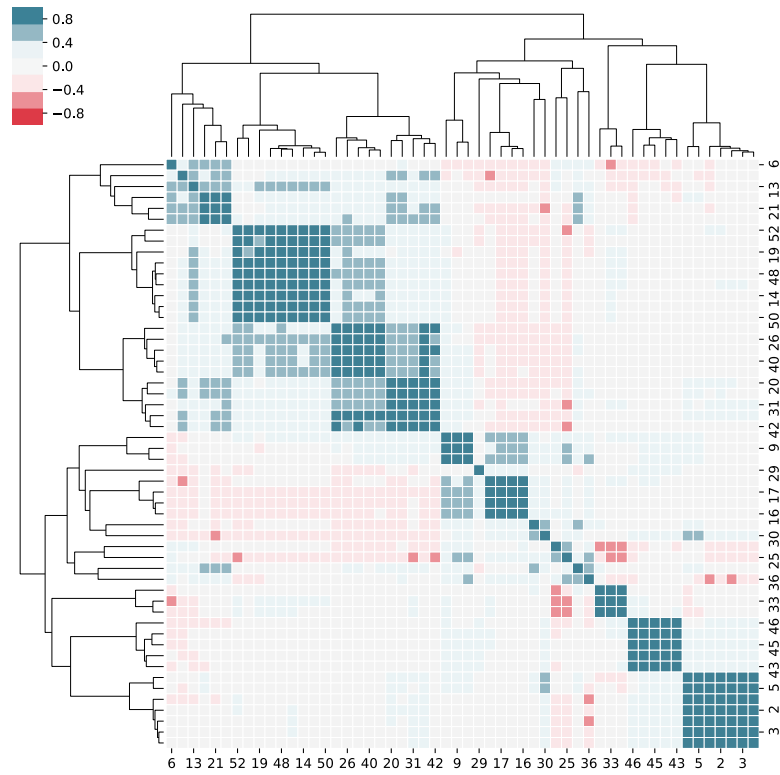


Figure 2.9: Hierarchically-clustered heatmap of selected genes. The heatmap plot identifies the correlational relationship between the selected 54 genes. The hierarchical tree shows the similarities and differences arising between selected genes of SBR.

(LQA) algorithm. The regularized LQA algorithm (RLQA) and SCE can resolve the ill-conditioned Gram matrix.

Finally, the SBR model can select the best subset of predictors via the $\text{ICOMP}_{\text{Miss}}$ criterion as our fitness function.

Under a large scale Monte Carlo misspecified simulation setting, we studied and compared the performance metrics of our proposed SBR approach to the Lasso. Through its application to a high-dimensional real dataset, we also presented some evidence that the SBR model is a worthy competitor to Lasso, to Elastic Net (EN) regression, and other penalized regression methods.

There are still unresolved problems that remain in SBR research deserving of further study. Areas, including the SBR model from a Bayesian perspective under misspecification, kernel SBR, and robust SBR, lack the rigorous solutions. We intend to conduct future research in some of these areas and will report our results under separate work.

Chapter 3

Segmentation of Time Series Data Using the Mixture of Sparse Principal Component Regression Model with Information Complexity

This chapter has been published as Sun, Yaojin, and Hamparsum Bozdogan. “Segmentation of high dimensional time-series data using mixture of sparse principal component regression model with information complexity.” Entropy 22.10 (2020): 1170.

This chapter presents a new and novel hybrid modeling method for the segmentation of high dimensional time-series data using the mixture of the sparse principal components regression (*MIX-SPCR*) model with information complexity (ICOMP) criterion as the fitness function. Our approach encompasses dimension reduction in high dimensional time series data and, at the same time, determines the number of component clusters (i.e., number of segments across time series data) and selects the best subset of predictors. A large-scale Monte Carlo simulation is performed to show the capability of the *MIX-SPCR* model to identify the correct structure of the time-series data successfully. *MIX-SPCR* model is also applied to a high dimensional S&P 500 index data to uncover the time series’s hidden structure and identify the structure change points. The approach presented in this paper

determines both the relationships among the predictor variables and how various predictor variables contribute to the explanatory power of the response variable through the sparsity settings cluster wise.

3.1 Introduction

This paper presents a new and novel method for the segmentation and dimension reduction of high dimensional time series data. We develop hybrid modeling between *mixture-model cluster analysis* and *sparse principal components regression* (MIX-SPCR) model as an expert unsupervised classification methodology with *information complexity* (ICOMP) criterion as the fitness function. This new approach performs dimension reduction in high dimensional time series data and, at the same time, determines the number of component clusters.

The research of time series segmentation and change point positioning has been a hot topic of research for a long time. Different research groups have provided solutions with various approaches in this area, including, but not limited to, Bayesian methods [12], fuzzy systems [1], and complex system modeling [125, 137, 112, 52]. We group these approaches into two branches, one based on complex systems modeling and the other on the statistical model through parameter estimation and inference. Among the complex systems-based modeling approaches, it is worth noting a series of papers that use the stochastic volatility model by [125]. For example, these authors used a nonlinear Hestone model to analyze 1071 stocks on the New York Stock Exchange (1987-1998). After accounting for the stochastic nature of volatility, the model is well suited to extracting the escape time distribution from financial time series data. The authors also identified the NES (Noise Enhanced Stability) effect to measure market dynamics' stabilizing effect. The approach we propose in this paper belongs to another branch of using a statistical model on time scales. Along with the empirical analysis, we show a broader view of how different companies/sectors behaved across different periods. In particular, we use a mixture-model based statistical methodology to segment the time series and determine change points.

The mixture-model cluster analysis of regression models is not new. These models are also known as “*cluster-wise regression*”, “*latent models*”, and “*latent structure models of choice*”.

These models have been well-studied among statisticians, machine learning researchers, and econometricians in the last several decades to construct time series segmentation models and identify change points. They have many useful theoretical and applied properties. Mixture-model cluster analysis of regression models is a natural extension of the standard multivariate Gaussian mixture-model cluster analysis. These models are beneficial to study heterogeneous data sets that involve not just one response variable but can have several responses or target-dependent variables simultaneously with a given set of independent variables. Recently, they have been proven to be a precious class of models in various disciplines in *behavioral and economic research, ecology, financial engineering, process control, and monitoring, market research, transportation systems*. Besides, we also witness the mixture model's usage in the *analysis of scanner panel, survey, and other choice data to study consumer choice behavior and dynamics* Dillon et al. [51].

In reviewing the literature, we note that Quandt and Ramsey [108] and Kiefer [88] studied data sets by applying a mixture of two regression models using moment generating function techniques to estimate the unknown model parameters. Later, De Veaux [45] developed an Expectation-Maximization (EM) algorithm to fit a mixture of two regression models. DeSarbo and Cron [48] used similar estimating equations and extended the earlier work done on a mixture of two regression models to a mixture of K-component regression models. For an excellent review article on this problem, we refer the reviewers to DeSarbo et al. [49].

In terms of these models' applications in the segmentation of time series can be seen in the early work of Sclove [118], where the author applied the mixture model to the segmentation of US gross national product, a high dimensional time series data. Specifically, Sclove [118] used the statistical model selection criteria to choose the number of classes.

With the currently existing challenges in mind in the segmentation of time series data, in this paper, our objective and goal are to develop a new methodology which can:

- identify and select variables that are sparse in the *MIX-SPCR* model,
- treat each time segment continuously in the process with some specified probability density function (pdf),

- determine the number of time series segments and the number of sparse variables and estimate the structural change points simultaneously,
- develop a robust and efficient algorithm for estimating model parameters.

We aim to achieve these objectives by developing the information complexity (ICOMP) criteria as our fitness function throughout the paper for the segmentation of high-dimensional time series data.

Our approach involves a two-stage procedure. We first make a variable selection by using SPCA with the benefit of sparsity. We then fit the sparse principal component regression (SPCR) model by transforming the original high dimensional data into several main principal components and estimating relationships between the sparse component loadings and the response variable. In this way, the mixture model not only handles the curse of dimensionality but also maintains the model’s excessive explanatory power. In this manner, we choose the best subset of predictors and determine the number of time series segments in the *MIX-SPCR* model simultaneously using ICOMP.

The rest of the paper is organized as follows. In Section 3.2, we present the model and methods. In particular, we first briefly explain sparse principal component analysis (SPCA) due to Zou et al. [161] in Section 3.2.1. In Section 3.2.2, we modify SPCA and develop mixtures of sparse principal component regression (*MIX-SPCR*) model for the segmentation of time-series data. In Section 3.3, we present a regularized entropy-based EM clustering algorithm. As is well known, the EM algorithm performs through maximizing the likelihood of the mixture models. However, to make the conventional EM algorithm robust (not sensitive to initial values) and converge to global optimum, we use the robust version of the EM algorithm for the *MIX-SPCR* model based on the work of Yang et al. [154]. These authors addressed the robustness issue by adding an entropy term of mixture proportions to the conventional EM algorithm’s objective function. While our EM algorithm is in the same spirit of the Yang et al. [154] approach, there are significant differences between our approach and theirs. Yang’s robust EM algorithm merely deals with the usual clustering problem without involving any response (or dependent) variable or time factor in the data. We extend it to the case of the *MIX-SPCR* model in the context of time-series data. In Section

3.4, we discuss various information criteria, specifically the information complexity based criteria (ICOMP). We derive the ICOMP for the *MIX-SPCR* model based on Bozdogan’s previous research [17, 18, 19, 20, 24]. In Section 3.5, we present our Monte Carlo simulation study. Section 3.5.2 involves an experiment on the detection of structural points, and Section 3.5.3 presents a large scale Monte Carlo simulation verifying the advantage of the *MIX-SPCR* with statistical information criteria. We provide a real data analysis in Section 3.6 using the daily adjusted closed S&P 500 index and stock prices from the Yahoo Finance database that spans the period from January 1999 to December 2019. Finally, our conclusion and discussion are presented in Section 3.7.

3.2 Model and Methods

In this section, we briefly present the *sparse principal component analysis* (SPCA), *sparse principal component regression* (SPCR) as a background. Then, by hybridizing these two methods within the mixture model, we propose the *mixture-model cluster analysis of sparse principal component regression* (abbreviated as *MIX-SPCR* model hereafter), for segmentation of high dimensional time-series datasets. Compared with a simple linear combination of all explanatory variables (i.e., the dense PCA model), the new approach interprets better because it maintains a sparsity specification.

Referring to Figure 3.1, we first show the overall structure of the model in this paper. The overall processing flow is that we clean and standardize the data after obtaining the time series data. Subsequently, we specify the number of time-series segments and how many Sparse Principal Components (SPCs) each segment contains. Using the Robust EM algorithm (Section 3.3), we estimate the model parameters, especially the boundaries (also known as *change points*) of each time segment. The information criterion values are calculated using the method of Section 3.4. By testing different numbers of time segments/SPCs, we obtain multiple criterion values. According to the calculated information criterion values, we choose the most appropriate model with the estimated parameters.

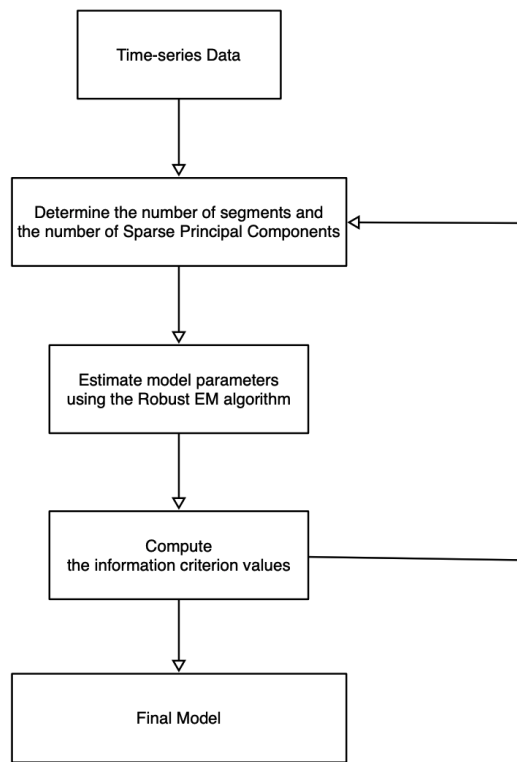


Figure 3.1: The flowchart of the MIX-SPCR method.

3.2.1 Sparse Principal Component Analysis (SPCA)

Given the input data matrix, $\mathbf{X}$ with n number of observations and p variables, we decompose $\mathbf{X}$ using the singular value decomposition (SVD). We write the decomposition procedure as $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T$, where $\mathbf{D}$ is a diagonal matrix of singular values and orthogonal columns $\mathbf{U}$ and $\mathbf{V}$ as the left and right singular vectors. When we perform SVD of a data matrix $\mathbf{X}$ that has been centered, by subtracting each column's mean, the process is the well-known *principal component analysis* (PCA). As discussed by Zou et al. [161], PCA has several advantages as compared with other dimensionality reduction techniques. For example, the PCA can sequentially identify the source of variability by considering the linear combination of all the variables. Because of the orthonormal constraint during the computation, all the calculated *principal components* (PCs) have clear geometrical interpretation corresponding to the original data space as a dimension reduction technique. Because PCA can deal with “the curse of dimensionality” of high-dimensional data sets, it has been widely used in real-world scenarios, including biomedical and financial applications.

Even though PCA has excellent properties that are desirable in real-world applications and statistical analysis, the interpretation of PCs is often difficult since it includes all the variables as linear combinations of all the original variables in each of the PCs. In practice, the principal components always have a large number of non-zero coefficient values for corresponding variables. To resolve this drawback, researchers proposed various improvements focusing on PCA's sparsity while maintaining the minimal loss of information. Shen and Huang [120] designed an algorithm to iteratively extract top PCs using the so-called *penalized least sum of square* (PLSS) criterion. Zou et al. [161] utilized the Lasso penalty (via Elastic Net) to maintain a sparse loading of the principal components, which is named *sparse principal component analysis* (SPCA).

In this paper, we use the sparse principal component analysis (SPCA) proposed by Zou et al. [161]. Given the data matrix $\mathbf{X}$, we minimize the objective function to obtain the SPCA results:

$$(\hat{\mathbf{A}}, \hat{\mathbf{B}}) = \arg \min_{\mathbf{A}, \mathbf{B}} \sum_{i=1}^n \|\mathbf{x}_i^T - \mathbf{A}\mathbf{B}^T \mathbf{x}_i^T\|^2 + \sum_{j=1}^k \lambda_{1,j} \|\mathbf{B}_{(j)}\|_1 + \lambda_2 \sum_{j=1}^k \|\mathbf{B}_{(j)}\|_2^2, \quad (3.1)$$

subject to

$$\mathbf{A}^T \mathbf{A} = I_k. \quad (3.2)$$

where I_k is the identify matrix. We maintain the hyperparameters $\lambda_{1,j}$ and λ_2 to be nonnegative. The $\mathbf{A}$ and $\mathbf{B}$ matrices of size $(p \times k)$ are given by

$$\mathbf{B} = \begin{bmatrix} B_{1,1} & \cdots & B_{1,k} \\ \vdots & \ddots & \vdots \\ B_{p,1} & \cdots & B_{p,k} \end{bmatrix} = [\mathbf{B}_{(1)} \mid \cdots \mid \mathbf{B}_{(k)}] = \begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_p \end{bmatrix}, \quad (3.3)$$

and

$$\mathbf{A} = \begin{bmatrix} A_{1,1} & \cdots & A_{1,k} \\ \vdots & \ddots & \vdots \\ A_{p,1} & \cdots & A_{p,k} \end{bmatrix} = [\mathbf{A}_{(1)} \mid \cdots \mid \mathbf{A}_{(k)}] = \begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_p \end{bmatrix}. \quad (3.4)$$

If we choose the first k principal components from the data matrix $\mathbf{X}$, then the estimate $\hat{\mathbf{B}}_{(j)}$ contains the sparse loading vectors, which are no longer orthogonal.

A bigger $\lambda_{1,j}$ means a greater penalty for having non-zero entries in $\hat{B}_{(j)}$. By using different $\lambda_{1,j}$, we control the number of zeros in the j th loading vector. If $\lambda_{1,j} = 0$ for $j = 1, 2, \dots, k$, this problem reduces to usual PCA.

Zou et al. [161] proposed a generalized SPCA algorithm to solve the optimization problem in equation (3.1). The algorithm applies the Elastic Net (EN) to estimate $\mathbf{B}_{(j)}$ iteratively and update matrix $\mathbf{A}$. However, this algorithm is not the only available approach for extracting principal components with sparse loadings. The SPCA could also be computed through dictionary learning by Mairal et al. [99]. By introducing the probability model of principal component analysis, SPCA is equivalent to the *sparse probabilistic principal component analysis* (SPPCA) if the prior is Laplacian distribution for each weight matrix element (Guan and Dy [67], Williams [149]). For further discussion on SPPCA, we refer readers to those related publications for more details.

Next, we introduce the *MIX-SPCR* model for the segmentation of time-series data.

3.2.2 Mixtures of SPCR Model for Time Series Data

Suppose the continuous response variable is denoted as $\mathbf{y} = \{y_i | 1 \leq i \leq n\}$, where n represents the number of observations (time points). Similarly, we have the predictors denoted as $\mathbf{X} = \{\mathbf{x}_i | 1 \leq i \leq n\}$. Each observation $\mathbf{x}_i$ has p dimensions and is represented as $\mathbf{x}_i = [x_{1,i}, x_{2,i}, \dots, x_{p,i}]^T$. Both the response variable and independent variables are collected sequentially labeled by time points $T = [t_1, t_2, \dots, t_n]$.

The finite mixture model allows applying cluster analysis on conditionally dependent data into several classes. In the time-series data scenario, researchers cluster the data $((t_1, \mathbf{x}_1, y_1), (t_2, \mathbf{x}_2, y_2), \dots, (t_n, \mathbf{x}_n, y_n))$ into several homogeneous groups where the number of groups G is unknown in general. Within each group, we apply the SPCA to extract top k principal components that each of them has a sparse loading of p variable coefficients. The extracted top k PCs are denoted as matrix $\mathbf{P}_{p \times k}$. We also use $\mathbf{P}_g$ to represent the principal component matrix obtained from the group indexed by $g = 1, 2, \dots, G$.

The SPCR model assumes that each pair $(\mathbf{x}_i, y_i)$ is independently drawn from a cluster using both the SPCA and the regression model as follows.

$$y_i = \mathbf{x}_i^T \mathbf{P}_g \beta_g + \epsilon_{i,g}, i = 1, 2, \dots, n, \quad (3.5)$$

where $\beta_g = [\beta_{g,1}, \beta_{g,2}, \dots, \beta_{g,k}]^T$. For each group g , the random error is assumed to be Gaussian distributed. That is, $\epsilon_{i,g} \sim \mathcal{N}(0, \sigma_g^2)$. If the response variable is multivariate, then the random error is usually assumed to be also a multivariate Gaussian distribution. Thus the probability density function (pdf) of the SPCR model is

$$f(y_i | \mathbf{x}_i, \mathbf{P}_g, \beta_g) = \mathcal{N}(y_i | \mathbf{x}_i^T \mathbf{P}_g \beta_g, \sigma_g^2). \quad (3.6)$$

We emphasize here that the noise (i.e., the error term) included in the statistical model is drawn from a normal distribution independent for each time-series segment, with different values of σ_g^2 for each period. Since we use the EM algorithm to estimate the parameters of the model, the noise parameter σ_g^2 can be estimated accurately as well. Future studies will

consider introducing different noise distributions, such as α -stable Lévy noise [68], and other non-Gaussian noise distributions to further extend the current model.

We also consider time factor t_i in the SPCR model of time-series data to be continuous. The pdf of the time factor is

$$f(t_i|v_g, \sigma_g^{2,\text{time}}) = \mathcal{N}(t_i|v_g, \sigma_g^{2,\text{time}}), \quad (3.7)$$

where v_g is the mean, and $\sigma_g^{2,\text{time}}$ is the variance of the time segment g . Apart from the normal distribution, our approach can also be generalized to other distributions for the time factor, such as skewed distributions, Student's t-distribution, ARCH, GARCH time series models, and so on.

As a result, if we use the *MIX-SPCR* model to perform segmentation of time-series data, the likelihood function of the whole data $((t_1, \mathbf{x}_1, y_1), (t_2, \mathbf{x}_2, y_2), \dots, (t_n, \mathbf{x}_n, y_n))$ with G number of clusters (or segments) is given by

$$L = \prod_{i=1}^n \prod_{g=1}^G [\pi_g f(y_i|\mathbf{x}_i, \mathbf{P}_g, \beta_g) f(t_i|v_g, \sigma_g^{2,\text{time}})]^{z_{g,i}}, \quad (3.8)$$

where the π_g is the mixing proportion with the constraint that $\pi_g \geq 0$ and $\sum_{g=1}^G \pi_g = 1$. We follow the definition of missing values by Yang et al. [154] and let $\mathbf{Z} = \{Z_1, Z_2, \dots, Z_n\}$. If $Z_i = g$, then $z_{g,i} = 1$, otherwise, $z_{g,i} = 0$. Then the log-likelihood function of the *MIX-SPCR* model models is

$$\begin{aligned} \mathcal{L}_{\text{mix}} &= \log(L) \\ &= \sum_{i=1}^n \sum_{g=1}^G z_{g,i} \log [\pi_g f(y_i|\mathbf{x}_i, \mathbf{P}_g, \beta_g) f(t_i|v_g, \sigma_g^{2,\text{time}})] \\ &= \sum_{i=1}^n \sum_{g=1}^G z_{g,i} [\log \pi_g + \log f(y_i|\mathbf{x}_i, \mathbf{P}_g, \beta_g) + \log f(t_i|v_g, \sigma_g^{2,\text{time}})] \\ &= \underbrace{\sum_{i=1}^n \sum_{g=1}^G z_{g,i} \log \pi_g}_{\mathcal{L}_{\pi}} + \underbrace{\sum_{i=1}^n \sum_{g=1}^G z_{g,i} \log f(y_i|\mathbf{x}_i, \mathbf{P}_g, \beta_g)}_{\mathcal{L}_{\text{SPCR}}} + \underbrace{\sum_{i=1}^n \sum_{g=1}^G z_{g,i} \log f(t_i|v_g, \sigma_g^{2,\text{time}})}_{\mathcal{L}_{\text{time}}}. \end{aligned} \quad (3.10)$$

We denote $\mathbf{z} = [z_{g,i}]$ where $g = 1, 2, \dots, G$ and $i = 1, 2, \dots, n$.

Given the number of segments, researchers usually apply the EM algorithm to determine the optimal segmentation by setting the objective function as $\mathcal{J}_{\text{EM}} = \mathcal{L}_{\text{mix}}$ (Gaffney and Smyth [64], Esling and Agon [54], Gaffney [63]).

3.3 Regularized Entropy-Based EM Clustering Algorithm

The EM algorithm is a method for iteratively optimizing the objective function. As discussed in Section 3.2.2, by setting the objective function as the log-likelihood function, we can use the EM algorithm to identify optimal segmentation of time series.

However, in practice, the EM algorithm is sensitive to model initialization conditions and cannot estimate the number of clusters appropriately. To deal with the initialization problem, in 2012, Yang et al. [154] proposed using an entropy penalty to stabilize the computation of each step. The improved method is called the *robust EM algorithm*. In this paper, we extend the robust EM algorithm to deal with time-series data for the *MIX-SPCR* model.

In Section 3.3.1, we discuss the entropy term of the robust EM algorithm. Then, we show the extension of the robust EM algorithm for the *MIX-SPCR* model in Sections 3.3.2 and 3.3.3.

3.3.1 The Entropy of EM Mixture Probability

As introduced in equation (3.8), the π_g represents the mixture probability of each cluster or segment. In other words, the value of π_g is the probability that a data point belongs to group g . The clustering complexity is determined by the number of clusters and corresponding probability values, which could be obtained using entropy. Given $\{\pi_g | 1 \leq g \leq G\}$, the entropy of Z_i is

$$H(Z_i | \{\pi_g | 1 \leq g \leq G\}) = - \sum_{g=1}^G \pi_g \log(\pi_g), \text{ for } i = 1, 2, \dots, n. \quad (3.11)$$

Then the entropy of $\mathbf{Z}$ is written as,

$$\begin{aligned}
H(\mathbf{Z} | \{\pi_g | 1 \leq g \leq G\}) &= \sum_{i=1}^n H(Z_i | \{\pi_g | 1 \leq g \leq G\}) \\
&= - \sum_{i=1}^n \sum_{g=1}^G \pi_g \log(\pi_g) \\
&= -n \sum_{g=1}^G \pi_g \log(\pi_g).
\end{aligned} \tag{3.12}$$

The objective function of the robust EM algorithm is

$$\mathcal{J}_{\text{Robust-EM}} = \mathcal{L}_{\text{mix}} - \lambda_{\text{Robust-EM}} H(\mathbf{Z} | \{\pi_g | 1 \leq g \leq G\}), \tag{3.13}$$

where $\lambda_{\text{Robust-EM}} \geq 0$. The log-likelihood term $\mathcal{L}_{\text{mix}}$ is from equation (3.9), which gives the goodness-of-fit.

Next, we present the steps of the EM algorithm for maximizing the objective function in equation (3.13).

3.3.2 E-Step (Expectation)

From a Bayesian perspective, we let $\hat{z}_{g,i}$ denote the posterior probability of the true cluster membership that a dataset triplet $(t_i, \mathbf{x}_i, y_i)$ is drawn from group g . Using the Bayes theorem, we have

$$\hat{z}_{g,i} = \mathbb{E}(Z_i = g | y_i, \mathbf{x}_i, \mathbf{P}_g, \beta_g) \tag{3.14}$$

$$= \frac{\pi_g \mathcal{N}(y_i; \mathbf{x}_i \mathbf{P}_g \beta_g, \sigma_g^2) \mathcal{N}(t_i | v_g, \sigma_g^{2, \text{time}})}{\sum_{h=1}^G \pi_h \mathcal{N}(y_i; \mathbf{x}_i \mathbf{P}_h \beta_h, \sigma_h^2) \mathcal{N}(t_i | v_h, \sigma_h^{2, \text{time}})}. \tag{3.15}$$

In the next section, we discuss the mathematical details of the M-step in more detail. In fact, the M-step is the core of the robust EM algorithm that distinguishes it from traditional algorithms.

3.3.3 M-Step (Maximization)

Using the robustified derivation of $\hat{\pi}_g$, the estimated mixture proportion, we have

$$\hat{\pi}_g^{\text{new}} = \hat{\pi}_g^{\text{EM}} + \hat{\lambda}_{\text{Robust-EM}} \hat{\pi}_g^{\text{old}} \left(\log(\hat{\pi}_g^{\text{old}}) - \sum_{h=1}^G (\hat{\pi}_h^{\text{old}} \log(\hat{\pi}_h^{\text{old}})) \right), \quad (3.16)$$

where

$$\hat{\pi}_g^{\text{EM}} = \frac{\sum_{i=1}^n \hat{z}_{g,i}}{n}. \quad (3.17)$$

We follow Yang et al. [154] recommendation on the value of $\hat{\lambda}_{\text{Robust-EM}}^{\text{new}}$ as

$$\hat{\lambda}_{\text{Robust-EM}}^{\text{new}} = \min \left\{ \frac{\sum_{h=1}^G \exp(-\eta n |\hat{\pi}_g^{\text{new}} - \hat{\pi}_g^{\text{old}}|)}{G}, \frac{1 - \max \left\{ \sum_{i=1}^n \hat{z}_{h,i}^{\text{old}} / n \mid h = 1, 2, \dots, G \right\}}{-\max \left\{ \hat{\pi}_h^{\text{old}} \mid h = 1, 2, \dots, G \right\} \sum_{h=1}^G \hat{\pi}_h^{\text{old}} \log \hat{\pi}_h^{\text{old}}} \right\}, \quad (3.18)$$

where

$$\eta = \min \{1, 0.5^{\lfloor p/2-1 \rfloor}\}, \quad (3.19)$$

and p is the number of variables in the model.

We iterate E-step and M-step several times until convergence to obtain the parameter estimates. In particular, the β_g values get updated by maximizing the $\mathcal{J}_{\text{Robust-EM}}$ from equation (3.13). Since we fix the number of segments and principal components during each E-step and M-step, the updated values of β_g and σ_g can be calculated using $\mathcal{L}_{\text{mix}}$ directly. The estimated values of β_g and σ_g are given as follows.

$$\hat{\sigma}_g^{2,\text{new}} = \sum_{i=1}^n \hat{z}_{g,i}^{\text{old}} \left\| y_i - \mathbf{x}_i^T \mathbf{P}_g \hat{\beta}_g^{\text{new}} \right\|_2^2 / \sum_{i=1}^n \hat{z}_{g,i}^{\text{old}}. \quad (3.20)$$

$$\begin{aligned}
\hat{\beta}_g^{\text{new}} &= \left[\sum_{i=1}^n \hat{z}_{g,i}^{\text{old}} (\mathbf{x}_i^T \mathbf{P}_g)^T (\mathbf{x}_i^T \mathbf{P}_g) \right]^{-1} \sum_{i=1}^n \hat{z}_{g,i}^{\text{old}} (\mathbf{x}_i^T \mathbf{P}_g)^T y_i \\
&= \left[\sum_{i=1}^n \hat{z}_{g,i}^{\text{old}} \mathbf{P}_g^T \mathbf{x}_i \mathbf{x}_i^T \mathbf{P}_g \right]^{-1} \sum_{i=1}^n \hat{z}_{g,i}^{\text{old}} \mathbf{P}_g^T \mathbf{x}_i y_i.
\end{aligned} \tag{3.21}$$

For the time factor, the estimated mean $\hat{v}_g$ and variance $\hat{\sigma}_g^{2,\text{time}}$ are

$$\hat{v}_g = \frac{\sum_{i=1}^n \hat{z}_{g,i} t_i}{\sum_{i=1}^n \hat{z}_{g,i}}, \tag{3.22}$$

$$\hat{\sigma}_g^{2,\text{time}} = \frac{\sum_{i=1}^n \hat{z}_{g,i} (t_i - \hat{v}_g)^2}{\sum_{i=1}^n \hat{z}_{g,i}}. \tag{3.23}$$

As discussed above, our approach is flexible in considering other distributional models for the time series factor, which we will pursue in separate research work.

3.4 Information Complexity Criteria

Recently, the statistical literature recognized the necessity of introducing model selection as one of the technical areas. In this area, the entropy and the Kullback-Leibler [93] information (or KL distance) play a crucial role and serve as an analytical basis to obtain the forms of model selection criteria. In this paper, we use information criteria to evaluate a portfolio of competing models and select the best-fitting model with minimum criterion values.

One of the first information criteria for model selection in the literature is due to the seminal work of Akaike [4]. Following the entropy maximization principle (EMP), Akaike developed the Akaike's Information Criterion (AIC) to estimate the expected KL distance or divergence. The form of AIC is

$$\text{AIC} = -2 \log L(\hat{\theta}) + 2k, \tag{3.24}$$

where $L(\hat{\theta})$ is the maximized likelihood function, and k is the number of estimated free parameters in the model. The model with minimum AIC value is chosen as the best model to fit the data.

Motivated by Akaike's work, Bozdogan [17, 18, 19, 20, 21, 24] developed a new information complexity (ICOMP) criteria based on Van Emden's [140] entropic complexity index in parametric estimation. Instead of penalizing the number of free parameters directly, ICOMP penalizes the covariance complexity of the model. There are several forms of ICOMP. In this section, we present the two general forms of ICOMP criteria based on the estimated inverse Fisher information matrix (IFIM). The first form is

$$\begin{aligned}\text{ICOMP}(\text{IFIM}) &= -2 \log L(\hat{\theta}) + 2\mathcal{C}(\hat{\Sigma}_{model}) \\ &= -2 \log L(\hat{\theta}) + 2\mathcal{C}_1(\hat{\mathcal{F}}^{-1}),\end{aligned}\tag{3.25}$$

where $L(\hat{\theta})$ is the maximized likelihood function, and $\mathcal{C}_1(\hat{\mathcal{F}}^{-1})$ represents the entropic complexity of IFIM. We define $\mathcal{C}_1(\hat{\mathcal{F}}^{-1})$ as

$$\mathcal{C}_1(\hat{\mathcal{F}}^{-1}) = \frac{s}{2} \log \left(\frac{\text{tr} \hat{\mathcal{F}}^{-1}}{s} \right) - \frac{1}{2} \log |\hat{\mathcal{F}}^{-1}|,\tag{3.26}$$

and where $s = \text{rank}(\hat{\mathcal{F}}^{-1})$. We can also give the form of $\mathcal{C}_1(\hat{\mathcal{F}}^{-1})$ in terms of eigenvalues,

$$\mathcal{C}_1(\hat{\mathcal{F}}^{-1}) = \frac{s}{2} \log \left(\frac{\bar{\lambda}_a}{\bar{\lambda}_g} \right),\tag{3.27}$$

where $\bar{\lambda}_a$ is the arithmetic mean of the eigenvalues, $\lambda_1, \lambda_2, \dots, \lambda_s$, and $\bar{\lambda}_g$ is the geometric mean of the eigenvalues.

We note that ICOMP penalizes the lack of parsimony and the profusion of the model's complexity through IFIM. It offers a new perspective beyond the simple counting of estimated model parameters. Instead, ICOMP considers the interaction and correlation among the estimated parameters through the model fitting process.

We define the second form of ICOMP as

$$\text{ICOMP}(\text{IFIM})_{\text{C}_{1\text{F}}} = -2 \log L(\hat{\theta}) + 2\text{C}_{1\text{F}}(\hat{\mathcal{F}}^{-1}), \quad (3.28)$$

where $\text{C}_{1\text{F}}(\hat{\mathcal{F}}^{-1})$ is given by

$$\text{C}_{1\text{F}}(\hat{\mathcal{F}}^{-1}) = \frac{s}{4} \frac{\frac{1}{s} \text{tr} \left(\left(\hat{\mathcal{F}}^{-1} \right)^T \left(\hat{\mathcal{F}}^{-1} \right) \right) - \left(\frac{\text{tr}(\hat{\mathcal{F}}^{-1})}{s} \right)^2}{\left(\frac{\text{tr}(\hat{\mathcal{F}}^{-1})}{s} \right)^2}. \quad (3.29)$$

In terms of the eigenvalues of IFIM, we write $\text{C}_{1\text{F}}(\hat{\mathcal{F}}^{-1})$ as

$$\text{C}_{1\text{F}}(\hat{\mathcal{F}}^{-1}) = \frac{1}{4\bar{\lambda}_a^2} \sum_{j=1}^s (\lambda_j - \bar{\lambda}_a)^2. \quad (3.30)$$

We want to highlight some features of $\text{C}_{1\text{F}}(\hat{\mathcal{F}}^{-1})$ here. The term $\text{C}_{1\text{F}}(\hat{\mathcal{F}}^{-1})$ is a second-order equivalent measure of complexity to the original term $\text{C}_1(\hat{\mathcal{F}}^{-1})$. Also, we note that $\text{C}_{1\text{F}}(\hat{\mathcal{F}}^{-1})$ is scale-invariant and $\text{C}_{1\text{F}}(\hat{\mathcal{F}}^{-1}) \geq 0$ with $\text{C}_{1\text{F}}(\hat{\mathcal{F}}^{-1}) = 0$ only when all $\lambda_j = \bar{\lambda}_a$. Furthermore, $\text{C}_{1\text{F}}(\hat{\mathcal{F}}^{-1})$ measures the relative variation in the eigenvalues.

These two forms of ICOMP provide us an easy to use computational means in high dimensional modeling. Next, we derive the analytical forms of ICOMP in the *MIX-SPCR* model.

3.4.1 Derivation of Information Complexity in *MIX-SPCR* Model for Time-Series Data

We first consider the log-likelihood function of the *MIX-SPCR* model given in equation (3.9),

$$\mathcal{L}_{\text{mix}} = \mathcal{L}_{\pi} + \mathcal{L}_{\text{SPCR}} + \mathcal{L}_{\text{time}}. \quad (3.31)$$

After some work, the estimated inverse Fisher information matrix (IFIM) of the mixture probabilities is

$$\hat{\mathcal{F}}_{\pi}^{-1} = \begin{bmatrix} \hat{\pi}_1^{-1} & 0 & 0 & 0 \\ 0 & \hat{\pi}_2^{-1} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \hat{\pi}_G^{-1} \end{bmatrix}. \quad (3.32)$$

Similarly, for each segment g , the estimated IFIM, $\hat{\mathcal{F}}_{g,\text{SPCR}}^{-1}$, is

$$\hat{\mathcal{F}}_{g,\text{SPCR}}^{-1} = \begin{bmatrix} \hat{\sigma}_g^2 \left[\sum_{i=1}^n \hat{z}_{g,i} (\mathbf{x}_i^T \mathbf{P}_g)^T (\mathbf{x}_i^T \mathbf{P}_g) \right]^{-1} & \mathbf{0} \\ \mathbf{0}^T & 2\hat{\sigma}_g^4 (\sum \hat{z}_{g,i})^{-1} \end{bmatrix}, \quad g = 1, 2, \dots, G. \quad (3.33)$$

Note that the IFIM should include both the SPCR models $\hat{\mathcal{F}}_{g,\text{SPCR}}^{-1}$ and the time factor $\hat{\mathcal{F}}_{g,\text{time}}^{-1}$ for each segment.

For each segment g , the time factor is under the univariate Gaussian distribution. As a result, the IFIM of the time factor is

$$\hat{\mathcal{F}}_{g,\text{time}}^{-1} = \begin{bmatrix} \hat{\sigma}_g^{2,\text{time}}/n & 0 \\ 0 & \frac{2}{n} \hat{\sigma}_g^{4,\text{time}} \end{bmatrix}. \quad (3.34)$$

By combining the two IFIMs for the SPCR model and the time factor, we have the inverse Fisher information

$$\hat{\mathcal{F}}_g^{-1} = \begin{bmatrix} \hat{\mathcal{F}}_{g,\text{SPCR}}^{-1} & \mathbf{0} \\ \mathbf{0}^T & \hat{\mathcal{F}}_{g,\text{time}}^{-1} \end{bmatrix}. \quad (3.35)$$

Overall, the inverse of the estimated Fisher information matrix (IFIM) for the *MIX-SPCR* model becomes

$$\widehat{\mathcal{F}}^{-1} \cong \begin{bmatrix} \widehat{\mathcal{F}}_{\pi}^{-1} & 0 & 0 & \cdots & 0 \\ 0 & \widehat{\mathcal{F}}_1^{-1} & 0 & \cdots & 0 \\ 0 & 0 & \widehat{\mathcal{F}}_2^{-1} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \widehat{\mathcal{F}}_G^{-1} \end{bmatrix}. \quad (3.36)$$

Using the above definition of ICOMP(IFIM) and the properties of block-diagonal matrices with their trace and determinant, we have

$$\text{ICOMP(IFIM)} = -2\mathcal{L}_{\text{mix}} + 2\mathbf{C}_1(\widehat{\mathcal{F}}^{-1}), \quad (3.37)$$

where

$$\mathbf{C}_1(\widehat{\mathcal{F}}^{-1}) = \frac{s}{2} \log \left[\frac{\text{tr}(\widehat{\mathcal{F}}_{\pi}^{-1}) + \sum_{g=1}^G \text{tr}(\widehat{\mathcal{F}}_g^{-1})}{s} \right] - \frac{1}{2} \left[\log \left| \widehat{\mathcal{F}}_{\pi}^{-1} \right| + \sum_{g=1}^G \log \left| \widehat{\mathcal{F}}_g^{-1} \right| \right], \quad (3.38)$$

and where $s = \text{rank}(\widehat{\mathcal{F}}^{-1}) = r_{\pi} + \sum_{g=1}^G r_g = \dim(\widehat{\mathcal{F}}^{-1})$.

Similarly, we derive the second equivalent form of $\text{ICOMP(IFIM)}_{\text{C}_{1\text{F}}}$ as

$$\text{ICOMP(IFIM)}_{\text{C}_{1\text{F}}} = -2\mathcal{L}_{\text{mix}} + 2\mathbf{C}_{1\text{F}}(\widehat{\mathcal{F}}^{-1}). \quad (3.39)$$

Using the properties of the block-diagonal matrices, we have

$$\text{tr} \left(\left(\widehat{\mathcal{F}}^{-1} \right)^T \left(\widehat{\mathcal{F}}^{-1} \right) \right) = \text{tr} \left(\widehat{\mathcal{F}}_{\pi}^{-1} \right)^2 + \sum_{g=1}^G \text{tr} \left(\widehat{\mathcal{F}}_g^{-1} \right)^2. \quad (3.40)$$

Thus, we integrate the previous discussion and the mathematical derivation to obtain the final tool for measuring the complexity of the model.

The open computational form of $\text{ICOMP}(\text{IFIM})_{\text{C}_{\text{IF}}}$ becomes

$$\text{ICOMP}(\text{IFIM})_{\text{C}_{\text{IF}}} = -2\mathcal{L}_{\text{mix}} + \frac{s}{2} \frac{\frac{1}{s} \left[\text{tr}(\hat{\mathcal{F}}_{\pi}^{-1})^2 + \sum_{g=1}^G \text{tr}(\hat{\mathcal{F}}_g^{-1})^2 \right] - \left[\frac{\text{tr}(\hat{\mathcal{F}}_{\pi}^{-1}) + \sum_{g=1}^G \text{tr}(\hat{\mathcal{F}}_g^{-1})}{s} \right]^2}{\left[\frac{\text{tr}(\hat{\mathcal{F}}_{\pi}^{-1}) + \sum_{g=1}^G \text{tr}(\hat{\mathcal{F}}_g^{-1})}{s} \right]^2}. \quad (3.41)$$

We note that in computing both forms of ICOMP above, we do not need to build the full inverse of the estimated Fisher information matrix (IFIM) for the *MIX-SPCR* model given in equation (3.36). All one requires is the computation of IFIM for each segment, which is appealing.

We also use AIC and CAIC (Bozdogan [15]) for comparison purposes given by

$$\text{AIC} = -2\mathcal{L}_{\text{mix}} + 2s^*, \quad (3.42)$$

$$\text{CAIC} = -2\mathcal{L}_{\text{mix}} + s^* (\log n + 1), \quad (3.43)$$

where $s^* = G(k + 3)$ is the number of estimated parameters in the *MIX-SPCR* model and $\log$ denotes the natural logarithm of the sample size n .

Next, we show our numerical examples starting with a detailed Monte Carlo simulation study.

3.5 Monte Carlo Simulation Study

We perform numerical experiments in a unified computing environment: Ubuntu 18.04 operating system, Intel i7-8700, and 32 GB of RAM. We use the programming language Python and the scientific computing package NumPy [142] to build a computational platform. The size of the input data directly affects the running time of the program. At $n = 4000$ time-series observations, the execution time for each EM iteration is about 0.9 seconds. Parameter

estimation can reach convergence within 40 steps of iterations, with a total machine run time of 37 seconds.

3.5.1 Simulation Protocol

In this section, we present the performance of the proposed *MIX-SPCR* model using synthetic data generated from a segmented regression model. Our simulation protocol has $p = 12$ variables and four actual latent variables. Two segmented regression models determine the dependent variable y , and each segment is continuous and has its own specified coefficients (β_1 and β_2). Our simulation set up is as follows:

$$\Lambda = \begin{bmatrix} 1.8 & 0 & 0 & 0 \\ 1.8 & 0 & 0 & 0 \\ 1.8 & 0 & 0 & 0 \\ 0 & 1.7 & 0 & 0 \\ 0 & 1.7 & 0 & 0 \\ 0 & 1.7 & 0 & 0 \\ 0 & 0 & 1.6 & 0 \\ 0 & 0 & 1.6 & 0 \\ 0 & 0 & 1.6 & 0 \\ 0 & 0 & 0 & 1.5 \\ 0 & 0 & 0 & 1.5 \\ 0 & 0 & 0 & 1.5 \end{bmatrix}, \quad (3.44)$$

$$\psi = \text{diag}(1.27, 0.61, 0.74, 0.88, 0.65, 0.81, 0.74, 1.3, 1.35, 0.74, 0.92, 1.32), \quad (3.45)$$

$$\Sigma = \Lambda\Lambda^T + \psi, \quad (3.46)$$

$$\mathbf{x}_t \sim \text{MVN}(\mathbf{0}, \Sigma), t = 1, 2, \dots, 4000, \quad (3.47)$$

$$\beta_1 = (-10, 0.1, 0.1, 0.1, 2.1, 0, 0, 0.1, 0.1, 0, 0, 0), \quad (3.48)$$

$$\beta_2 = (0, 0, 0, 0, 0, 0.5, 0.3, 0.1, 2.1, 1, 2, 20), \quad (3.49)$$

$$y_{t,g=1} = \mathbf{x}_{1,t}\beta_1 + \varepsilon_{1,t}, t = 1, 2, \dots, 2800, \quad (3.50)$$

$$y_{t,g=2} = \mathbf{x}_{2,t}\beta_2 + \varepsilon_{2,t}, t = 2801, 2802, \dots, 4000, \quad (3.51)$$

We set the total number of time-series observations, $n = 4000$. The first segment has $n_1 = 2800$, and the second segment has $n_2 = 1200$ time-series observations. We randomly draw error term from a Gaussian distribution with zero mean and $\sigma^2 = 9$. Among all the variables, the first six observable variables explain the first segment, and the remaining six explanatory variables primarily determine the second segment. We set the mixing proportions $\pi_1 = 0.7$ and $\pi_2 = 0.3$ for two time-series segments, respectively.

3.5.2 Detection of Structural Change Point

In the first simulation study, we limit the actual number of segments equal to two, which means that the first segment expands from the starting point to a structural change point, and the second segment expands from the change point to the end. By design, each segment is continuous on the time scale, and different sets of independent variables explain the trending and volatility. We run the *MIX-SPCR* model to see if it can successfully determine the position of the change point using the information criteria. If a change point is correctly selected, we expect that the information criteria is minimized at this change point.

Figures 3.2 and 3.3 show our results from the *MIX-SPCR* model. Specifically, it shows the sample path of the information criteria at each time point. We note that all the information criteria values are minimized from $t = 2800$ to $t = 3000$, which covers the time series's actual change point position. As the *MIX-SPCR* model selects different change points, the penalty term of AIC and CAIC remain the same because both the number of model parameters and the number of observations do not change. In this simulation scenario, the fixed penalty term means that the AIC and CAIC reflect the changes only in the "lack of fit" term of various models without considering model complexity. This indicates that using AIC-type criteria just counting and penalizing the number of parameters may be necessary but not sufficient in model selection. As a comparison, however, we note that the penalty term of information complexity-based criteria, C_1 and C_{1F} , are adjusted in selecting different change points. They are varying but not fixed.

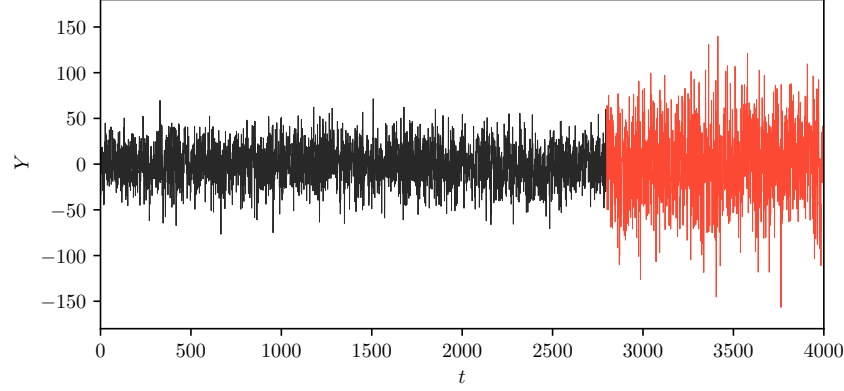


Figure 3.2: The plot of two-segment simulated time-series data. We show the plot of the simulated time-series data through the whole-time scale. Note that the first segment is from the starting point $t = 1$ to the change point $t = 2800$, and the second time segment expands from the change point $t = 2801$ to the end $t = 4000$.

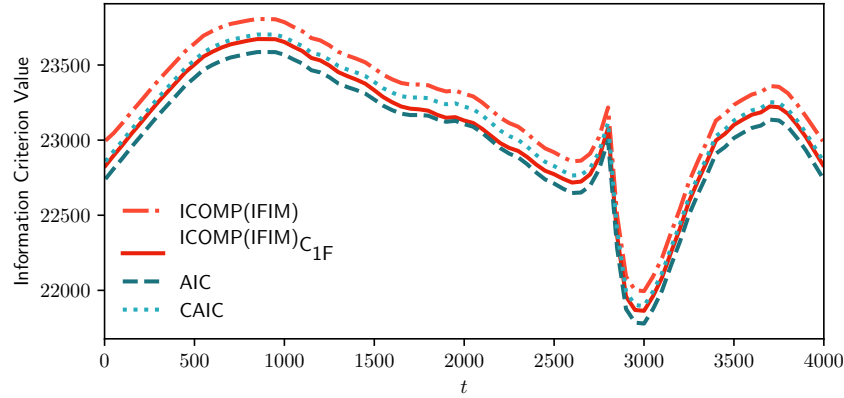


Figure 3.3: The horizontal coordinate represents the position of the possible change points, and the vertical coordinate represents the corresponding information criterion (IC) values. The lower the IC value, the more likely the selected position of the change point is the real position. The real change point is $t = 2800$.

3.5.3 A Large-Scale Monte Carlo Simulation

Next, we perform a large-scale Monte Carlo simulation to illustrate the *MIX-SPCR* model's performance in choosing the correct number of segments and the number of latent variables. A priori, in this simulation, we pretend that we do not know the actual structure of the data and use the information criteria to recover the actual construction of the *MIX-SPCR* model. To achieve this, we follow the above simulation protocol using a different number of time points by varying $n = 1000, 2000, 4000$. As before, there are twelve explanatory variables drawn from four latent variable model generated from a multivariate Gaussian distribution given in equation (3.47). The simulated data again consist of two time-series segments with mixing proportions $\pi_1 = 0.7$ and $\pi_2 = 0.3$, respectively. For each data generating process, we replicate the simulation one hundred times and record both information complexity-based criteria ($\text{ICOMP}(\text{IFIM})$ & $\text{ICOMP}(\text{IFIM})_{\text{C}_{\text{IF}}}$) and classic AIC-type criteria (AIC & CAIC). In Table 3.1, we present how many times the *MIX-SPCR* model selects different models in the one hundred simulations. In this way, we can assess different information criteria by measuring the hit rates.

Looking at Table 3.1, we see that when the sample size $n = 1000$ (small) AIC selects the correct model ($G = 2, k = 4$) 69 times, CAIC selects 80 times, $\text{ICOMP}(\text{IFIM})$ selects 48 times, and $\text{ICOMP}(\text{IFIM})_{\text{C}_{\text{IF}}}$ selects 76 times, respectively, in 100 replications of the Monte Carlo simulation. When the sample size is small, $\text{ICOMP}(\text{IFIM})$ tends to choose a sparser regression model sensitive to the sample size. However, as the sample size increases, when $n = 2000$ and $n = 4000$, $\text{ICOMP}(\text{IFIM})$ consistently outperforms other information criteria in terms of hit rates. The percentage of the correctly identified model is above 90%, as reported above.

Our results show that the *MIX-SPCR* model works well in all settings to estimate the number of time-series segments and the number of latent variables.

Figure 3.4 illustrates how the *MIX-SPCR* model performs if the number of segments and the number of sparse principal components are unknown beforehand.

The choice of the number of segments (G) has a significant impact on the results. For all the simulation scenarios, the correct choice of the number of segments ($G = 2$) has

Table 3.1: Frequency of the choice of the true model with information criteria in 100 replications of the experiment for each sample size (n) of time series observations. The true model is $G = 2$ and $k = 4$.

		$n = 1000$		$n = 2000$		$n = 4000$	
		G=2	G=3	G=2	G=3	G=2	G=3
AIC	k=2	0	0	0	0	0	0
	k=3	0	6	0	3	0	1
	k=4	69	0	77	0	75	0
	k=5	24	1	20	0	24	0
CAIC	k=2	1	0	0	0	0	0
	k=3	1	3	0	1	0	1
	k=4	80	0	96	0	93	0
	k=5	14	1	3	0	6	0
ICOMP(IFIM)	k=2	31	2	1	0	0	0
	k=3	2	5	0	2	0	1
	k=4	48	0	96	0	96	0
	k=5	11	1	1	0	3	0
ICOMP(IFIM) _{C_{1F}}	k=2	2	1	0	0	0	0
	k=3	0	7	0	3	0	1
	k=4	76	0	93	0	93	0
	k=5	13	1	4	0	6	0

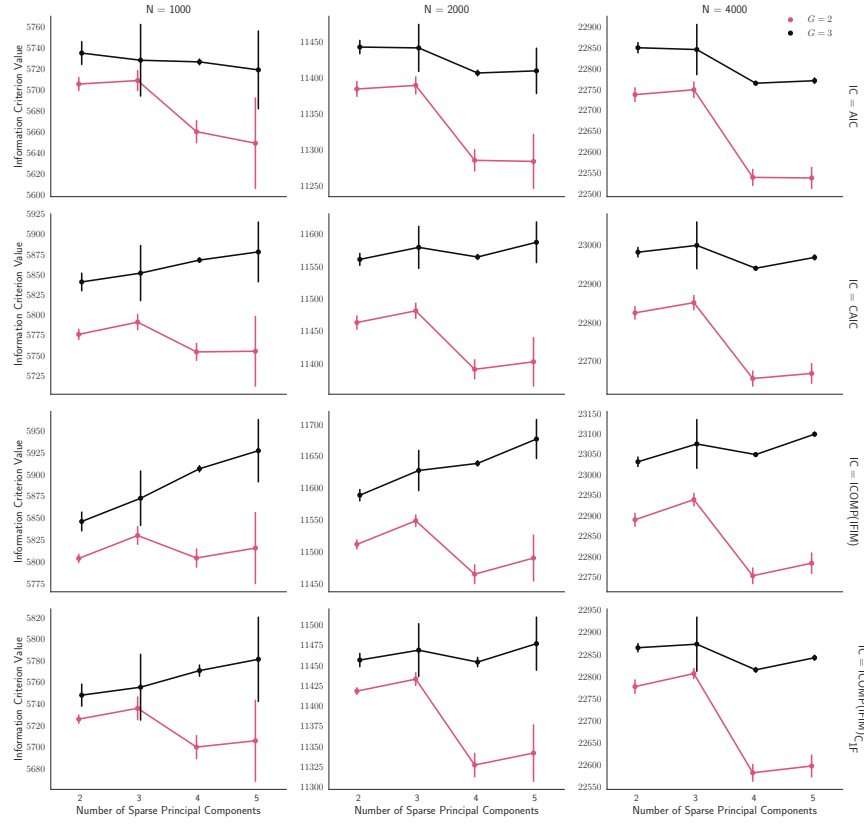


Figure 3.4: Plot of average and 1SD (standard deviation) of information criterion values over different sample sizes in all simulations with three SPCs and $G = 2$ segments. The red line indicates the estimated *MIX-SPCR* model based on two groups ($G = 2$). Correspondingly, the black line indicates the estimated *MIX-SPCR* model for three groups ($G = 3$). Horizontal coordinates represent different numbers of SPCs.

information criterion values less than the incorrect choice ($G = 3$). This pattern emerges consistently among all the sample sizes, both the classical ones and information-complexity based criteria.

In summary, the large-scale Monte Carlo simulation analysis highlights the performance of the *MIX-SPCR* model. As the sample size increases, the *MIX-SPCR* model improves its performance. As shown in Figure 3.3, the *MIX-SPCR* model can efficiently determine the structural change point and estimate the mixture proportions when the number of segments is unknown beforehand. Another key finding is that, by using the appropriate information criteria, the *MIX-SPCR* model can correctly identify the number of segments and the number of latent variables from the data. In other words, our approach can extract the main factors not only from the intercorrelated variables but also classify the data into several clearly defined segments on the time scale.

3.6 Case Study: Segmentation of the S&P 500 Index

3.6.1 Description of the Data

The financial market often generates a large amount of time-series data, and in most cases, the generated data is high-dimensional. In this paper, we use the S&P 500 index and its related hundreds of company stocks categorized into eleven sectors, which are high dimensional time-series data. The index value is the response variable mixed by plenty of companies' variations at each time point. These long time-series values often consist of different regimes and states. For example, the stock market experienced a boom period from 2017 to 2019, which is a dramatic change compared with the stock market during the 2008 financial crisis. If we analyze each sector or company, some industries perform more actively than others during a particular period.

In this section, we implement the *MIX-SPCR* model on the adjusted closed price of the S&P 500 ($\sim$ GSPC) as a case study. We extract the daily adjusted closed prices from the Yahoo Finance database ¹ that spans the period from January 1, 1999, to December 31,

¹<https://finance.yahoo.com/>

2019. By removing weekends and holidays, there are $n = 5292$ tradable days in total. The main focus of this section is to split the time series into several self-contained segments. Besides, we expect the extracted sparse principal components to explain the variance and volatility in each segment.

3.6.2 Computational Results

To have a big picture of how the S&P 500 index values reflect the changes of 506 company stock prices, Figure 3.5 shows the plot of the normalized values of adjusted closing prices. We use the *MIX-SPCR* model with the information criteria to determine the number of segments and the number of sparse principal components. To achieve interpretable results, we limit our search space to a maximum of seven time-series and six sparse principal components. Table 3.2 shows the optimal combination of three self-contained segments and three sparse principal components for each of the segments by using the information complexity ICOMP(IFIM). The other three information criteria also choose this combination as the best-fitting model. Figure 3.6 illustrates the probability and time range of each segment. We can see that the first segment is from January 1, 1999, to October 26, 2007. The second time-series segment spans from October 29, 2007, to the end of 2016. The last segment extends from December 30, 2016, to December 31, 2019.

We emphasize that many factors may explain the stock market variation, and this is not a research on how the socioeconomic events influence the S&P 500 index. However, it does raise our interest in the distribution of two structural change points from the segmentation results. The first change point is October 2007, which is the early stage of the 2008 financial crisis. The second structural change point is December 2016, the transitional period of the USA presidential election. Identification of these two change points shows our proposed method can detect the underlying physical and structural change from the available time-series data.

Table 3.3 lists the estimated coefficients (β_g) from sparse principal component regression. Because all the collected stock prices and S&P 500 index values are standardized before implementing the *MIX-SPCR* model, we make dimension reduction, remove the constant

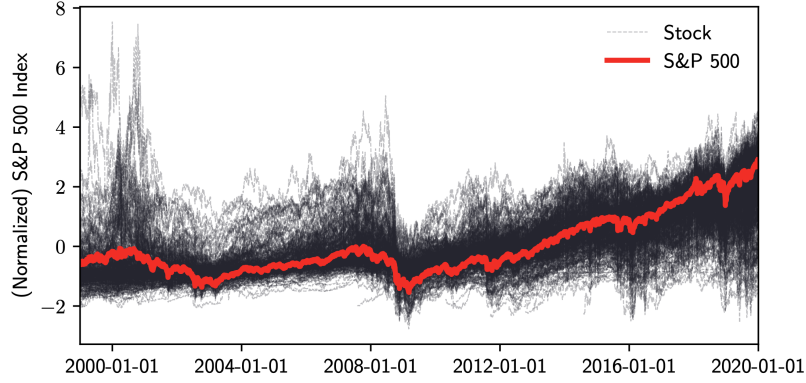


Figure 3.5: Normalized S&P 500 index and stock prices from January 1999 to December 2019.

Table 3.2: The ICOMP(IFIM) values of segmentation results for S&P 500 index data (Lower is better).

		Number of Sparse Principal Components				
		1	2	3	4	5
Number of Segments	1	30097.04	30092.45	30106.50	30121.64	30145.13
	2	29975.01	30058.40	30293.55	30234.65	30347.94
	3	30010.70	30062.19	29241.52	30453.74	30526.20
	4	29877.27	29825.73	29811.53	30571.39	30628.61
	5	29904.35	29973.47	30011.18	30311.52	30554.82
	6	30111.35	30361.39	30388.47	30665.26	30581.29
	7	30031.39	30564.65	30597.14	30823.76	31057.54

term, and perform regression analysis using the SPCR model. The R^2 values are above 0.8 across all three different time segments.

3.6.3 Interpretation of Computational Results

One may ask a question, “can the *MIX-SPCR* model identify the key variables from the hundreds of companies?” If the constructed model is dense, the selected companies would include all the sectors whereby the dense model is limiting the interpretation of the data. Our analysis identifies all the companies with non-zero coefficient values and maps them back to each of the sectors in Tables 3.4, 3.5 and 3.6. Each calculated sparse principal component vector consists of around fifty companies, much less than the original data dimension ($p = 506$). We observe that these selected companies are grouped into a few sectors within different time segments. For example, energy companies load in the first sparse principal component vector from 1999 to 2007 (segment 1) and diminish after that.

To have a detailed analysis of how different sectors perform across three segments, we do the stem plot to show the sparse principal component coefficients $\mathbf{P}_g$ of four sectors, namely financials, real estate, energy, and Information technology. Figures 3.7 and 3.8 indicate a similar behavior that happened in financial and real estate companies. Both sectors play an essential role in the first two time-series segments but have no contribution in the third segment, which is the period after December 2016. Notice that in Figure 3.9, energy companies act as an essential player before 2016. However, during the recession in 2008, energy company loadings are negated from the first SPC to the second SPC. Compared with other industries, the variation of energy company stock prices does not contribute to the S&P 500 index after 2016.

Another question is that what sector/industry is the main contributing factor after the 2016 United States presidential election? A possible answer is from Figure 3.10, the SPC coefficients of information technology companies. From 1999 to the recession in 2008, IT companies work mainly on the second SPC and the third SPC, which do not contribute much to the main variation. After the recession, the variations of IT companies do not contribute compared with other sectors. However, after December 2016, companies from the IT industry plays an essential role in the primary stock price volatility.

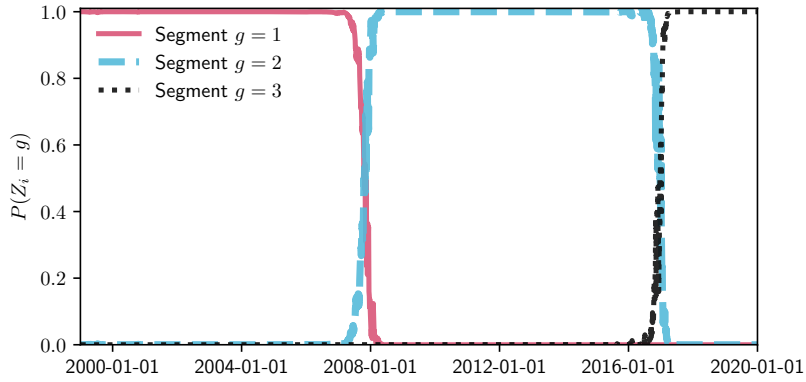


Figure 3.6: Segmented periods and probability. The plot's vertical coordinate indicates the probability that an individual time-series data point belongs to each segment.

Table 3.3: SPCR coefficients (β_g) of three different segments.

	Segment 1 ($R^2 = 0.82$) 1999-01-01 ~ 2007-10-26	Segment 2 ($R^2 = 0.94$) 2007-10-27 ~ 2016-12-29	Segment 3 ($R^2 = 0.97$) 2016-12-30 ~ 2019-12-31
SPC1	0.0964	0.1240	0.1512
SPC2	0.0729	-0.0439	0.0359
SPC3	0.0079	0.0191	-0.0051

Table 3.4: Sparse Principal Component (SPC) of Segment 1 (1999-01-01 ~ 2007-10-26).

	SPC1		SPC2		SPC3	
	Count	Percentage	Count	Percentage	Count	Percentage
Health Care	4	6.56	6	9.84	1	1.64
Industrials	6	8.57	3	4.29	6	8.57
Utilities	5	17.86	3	10.71	3	10.71
Materials	2	7.14	1	3.57	1	3.57
Consumer Discretionary	5	7.81	6	9.38	6	9.38
Energy	13	46.43	1	3.57	3	10.71
Financials	5	7.58	10	15.15	15	22.73
Real Estate	5	16.13	2	6.45	5	16.13
Consumer Staples	2	6.06	0	0.00	1	3.03
Communication Services	1	3.85	2	7.69	1	3.85
Information Technology	2	2.82	16	22.54	8	11.27

Table 3.5: Sparse Principal Component (SPC) of Segment 2 (2007-10-27 ~ 2016-12-29).

	SPC1		SPC2		SPC3	
	Count	Percentage	Count	Percentage	Count	Percentage
Health Care	7	11.48	2	3.28	1	1.64
Industrials	5	7.14	6	8.57	4	5.71
Utilities	0	0.00	0	0.00	5	17.86
Materials	6	21.43	2	7.14	3	10.71
Consumer Discretionary	7	10.94	14	21.88	3	4.69
Energy	2	7.14	14	50.00	9	32.14
Financials	12	18.18	3	4.55	16	24.24
Real Estate	2	6.45	3	9.68	6	19.35
Consumer Staples	0	0.00	0	0.00	0	0.00
Communication Services	5	19.23	3	11.54	1	3.85
Information Technology	4	5.63	3	4.23	2	2.82

Table 3.6: Sparse Principal Component (SPC) of Segment 3 (2016-12-30 ~ 2019-12-31).

	SPC1		SPC2		SPC3	
	Count	Percentage	Count	Percentage	Count	Percentage
Health Care	10	16.39	14	22.95	2	3.28
Industrials	9	12.86	4	5.71	4	5.71
Utilities	1	3.57	0	0.00	0	0.00
Materials	1	3.57	3	10.71	6	21.43
Consumer Discretionary	3	4.69	10	15.63	8	12.50
Energy	0	0.00	3	10.71	1	3.57
Financials	5	7.58	1	1.52	10	15.15
Real Estate	0	0.00	0	0.00	2	6.45
Consumer Staples	0	0.00	6	18.18	3	9.09
Communication Services	2	7.69	5	19.23	5	19.23
Information Technology	19	26.76	4	5.63	9	12.68

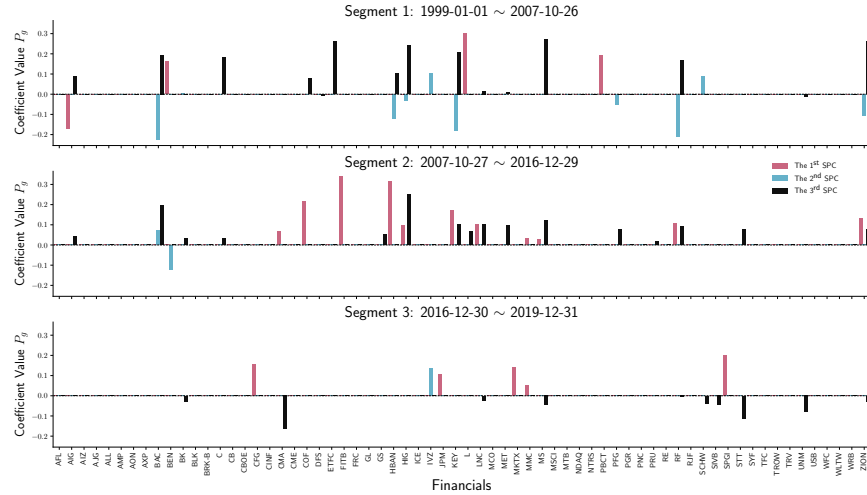


Figure 3.7: Stem plot of SPC coefficients $\mathbf{P}_g$ for financial companies within each time segment. From top to bottom, the three panels represent different segmented periods, respectively. The horizontal axis of each panel indicates the company in the industrial sector. The vertical axis shows the SPC coefficient values.

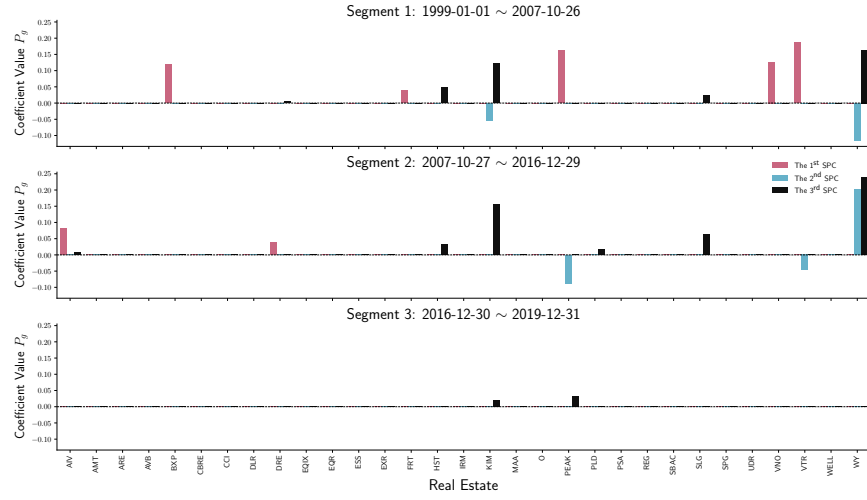


Figure 3.8: Stem plot of SPC coefficients $\mathbf{P}_g$ for real estate companies within each time segment. From top to bottom, the three panels represent different segmented periods, respectively. The horizontal axis of each panel indicates the company in the industrial sector. The vertical axis shows the SPC coefficient values.

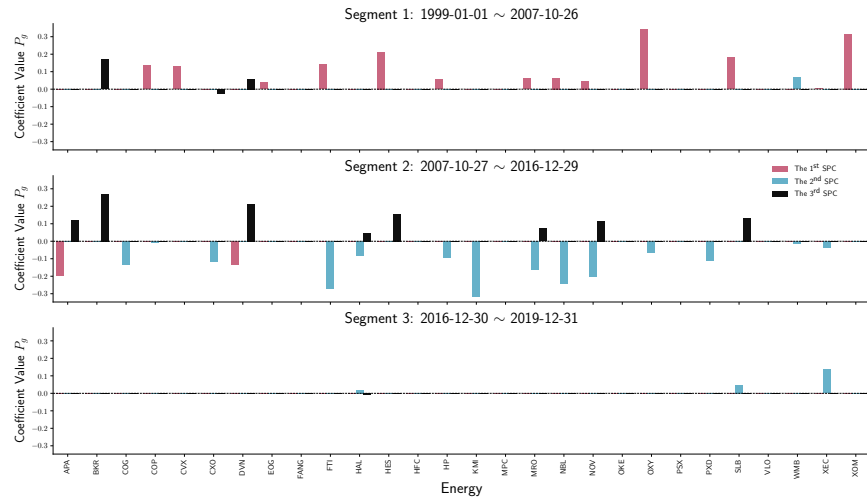


Figure 3.9: Stem plot of SPC coefficients $\mathbf{P}_g$ for energy companies within each time segment. From top to bottom, the three panels represent different segmented periods, respectively. The horizontal axis of each panel indicates the company in the industrial sector. The vertical axis shows the SPC coefficient values.

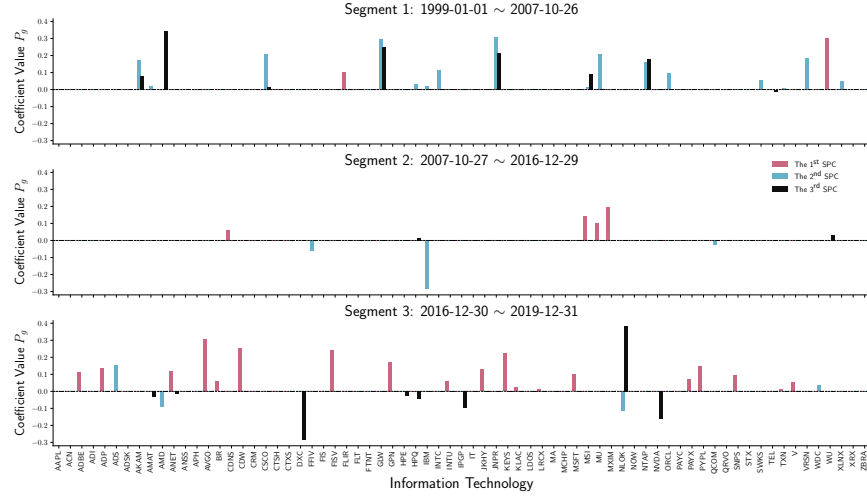


Figure 3.10: Stem plot of SPC coefficients $\mathbf{P}_g$ for information technology companies within each time segment. From top to bottom, the three panels represent different segmented periods, respectively. The horizontal axis of each panel indicates the company in the industrial sector. The vertical axis shows the SPC coefficient values.

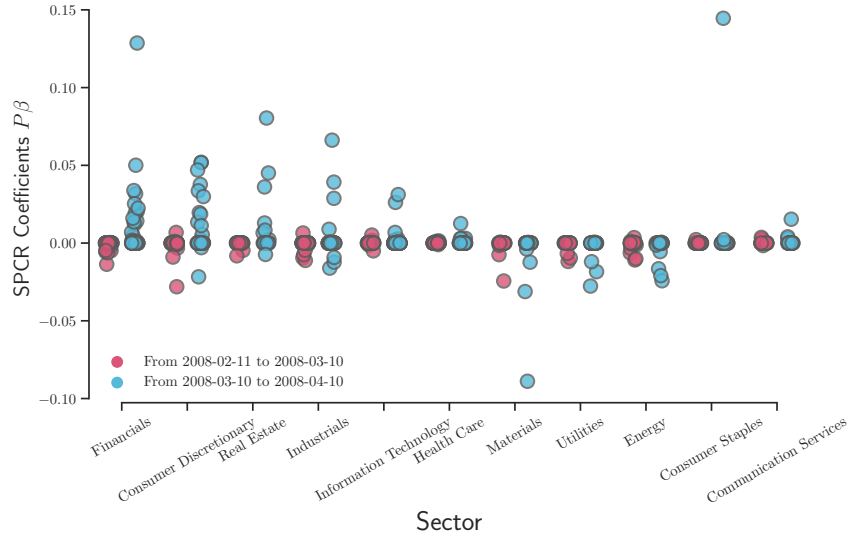


Figure 3.11: Overlay plot of the SPCR coefficients before/after 2008 financial crisis.

As discussed above, Figures 3.7 to 3.10 provide a clear picture of how different sectors perform (via coefficient $\mathbf{P}_g$) without considering the effects on the S&P 500 index. It might raise the interest in how SPCR coefficient $\mathbf{P}_g\beta_g$ changes before/after certain socioeconomic events. We follow the research implemented by Aït-Sahalia and Xiu [3] about how Federal Reserve addressing heightened liquidity from March 10 to March 14, 2008, affects the stock market. The data analyzed by Aït-Sahalia and Xiu [3] is the S&P 100 index values using the traditional PCA, and the authors grouped stocks into financial and non-financial categories. Instead of PCA, we apply the SPCR model on the S&P 500 index and analyze how eleven sectors react before/after Federal Reserve operations. Figure 3.11 shows that financials, consumer discretionary, real estate, and industrials experienced more significant perturbations than other sectors in terms of SPCR coefficients $\mathbf{P}_g\beta_g$. This conclusion is consistent with results from Aït-Sahalia and Xiu [3] that the average loadings of first and second principal components of financial companies are distinct from non-financial companies. However, considering that we have 506 companies in the raw data and make a sparse loading of companies for comparison, the excessive explanatory power is still maintained in this high-dimensional case using the SPCR model, which is more interpretable.

3.7 Conclusion and Discussion

In this paper, we presented a new and novel method to segment high-dimensional time-series data into different clusters or segments using the mixture model of the sparse principal components model (*MIX-SPCR*). The *MIX-SPCR* model considers both the relationships among the predictor variables and how various predictor variables contribute the explanatory power to the response variable through the sparsity settings. Information criteria have been introduced and derived for the *MIX-SPCR* model. These criteria are applied to study their performance under different sample sizes and to select the best-fitting model.

Our large-scale Monte Carlo simulation exercise showed that the *MIX-SPCR* model could successfully identify the real structure of the time-series data using the information criteria as the fitness function. In particular, based on our results, the information complexity-based criteria, i.e., $\text{ICOMP}(\text{IFIM})$ and $\text{ICOMP}(\text{IFIM})_{\text{C}_{\text{IF}}}$, outperformed the conventional standard

information criteria such as the AIC-type criteria as the data dimension and the sample size increase.

Later, we empirically applied the *MIX-SPCR* model to uncover the S&P 500 index data (from 1999 to 2019) and identify two change points of this data set.

We observe that the first change point physically coincides with the early stages of the 2008 financial crisis. The second change point is immediately after the 2016 United States presidential election. This structural change point coincides with the election of President Trump and his transition.

Our findings showed how the S&P 500 index and company stock prices react within each time-series segment. The *MIX-SPCR* model presents the excessive explanatory power by identifying how different sectors fluctuated before/after the Federal Reserve’s addressing heightened liquidity from March 10 to March 14, 2008.

Although this is not a traditional event study paper, it is the first paper to use the sparse principal component regression model with mixture models in the time-series analysis. The proposed new and novel *MIX-SPCR* model enlightens us to explore more interpretable results on how macroeconomic factors/events influence the stock prices on the time scale. Later, in a separate paper, we will incorporate the event study in the *MIX-SPCR* model as our future research initiative.

This paper’s time segmentation model builds on time series data, constructs likelihood functions, and performs parameter estimation by introducing error information unique to each period. Researchers have recently realized that environmental background noise can positively affect the model building and analysis under certain circumstances ([126, 138, 124, 119, 9, 2, 95]). For example, in Azpeitia and Wagner [9], the authors highlighted that the introduction of noise is necessary to obtain information about the system. In our next study, we would like to explore this positive effect of environmental noise even further and use it to build better statistical models for analyzing high-dimensional time series data.

Chapter 4

Sparse Kernel Factor Analysis In High Dimensional Data Using The Information Complexity Criterion

This paper presents a novel statistical model, namely Sparse Kernel Factor Analysis (SKFA). The SKFA model allows for the analysis of nonlinear features of data by combining the kernel method with the Sparse Factor Analysis. We describe the structure of the model and the parameter estimation method in the paper and examine systematically how sparsity affects the interpretability and computational stability of the model. Additionally, we introduce Information Complexity Criterion (ICOMP) to search for hyperparameters in the model selection process. Finally, the simulations and real-world data analysis confirm the excellent performance of SKFA in wide data analysis.

4.1 Introduction

The paper presents a new analysis method that combines kernel methods and Sparse Factor Analysis (SFA). The kernel method extracts the nonlinear features of the data, and the SFA method allows for a reasonable interpretation of the results and reduces dimension. We discuss in this paper the parameter estimation, hyperparameter search, and model selection for the new model and attempt to show the entire model processing steps. Finally, we

illustrate the proposed model with a Monte Carlo simulation and several real-world data sets.

Kernel models are a common method for constructing nonlinear models. In recent years, kernel-based methods have enriched traditional statistical modeling methods and machine learning methods by providing nonlinear features to improve model learning. The most well known example is the Support Vector Machine (SVM) [116, 122]. When the raw data cannot be classified appropriately in the original space, SVM uses kernels to construct a binary classification surface in feature space. Applications of kernel methods include digital signal processing [110], genomics [113], statistical process monitoring [8], stock price prediction [80], etc. It is also worth noting that machine learning has attempted to further combine deep learning and kernel methods to expand the potential applications of kernel methods in data modeling [40]. By introducing kernels, this paper extends the use of SFA model to perform fine-grained classification and clustering in the feature space. Consequently, the model's classification accuracy is improved in data analysis.

Uncommonly discussed is the fact that the number of observations determines the dimensionality of kernelized data. Based on this property, the kernel approach has a natural advantage for wide data sets (i.e., the number of data items is far less than the number of variables).

Factor analysis (FA) models were initially proposed to reduce data dimensionality and provide better model interpretation [44]. Researchers use FA models to develop linear statistical models most of the time [13, 33]. There has also been interest in applying FA models to nonlinear problems in recent years, with encouraging results. Schölkopf et al. published the Kernel Principal Component Analysis (KPCA) model in 1998, a simplified adaptation of the FA model in a specific setting. Michael Tipping further extended it to a Sparse Kernel Principal Component Analysis (SKPCA) model [135]. Additionally, several papers have been published on the theory and application of the Kernel Factor Analysis (KFA) model [151, 36]. However, our literature review found almost no article on Sparse Kernel Factor Analysis (SKFA) model construction and specific parameter estimation in model fitting process. It is for this reason that we have undertaken this study.

By introducing sparsity in this paper, we aim to ensure the interpretability of the model, as well as to improve the model’s computation efficiency. The FA model has a so-called *unidentifiability* problem, i.e., rotate factor loading to obtain an infinite number of legitimate outcomes. It has been proposed that there are several ways to solve the *unidentifiability* problem [104], and sparsity is one of the most feasible approaches. (This problem and its solution are discussed in detail in Section 4.3.1.)

In response to the previous discussion, we construct the Sparse Kernel Factor Analysis (SKFA) model with the following contributions

- Extraction of the nonlinear characteristics of the data itself
- Modeling of high-dimensional data to achieve dimensionality reduction
- Interpretability of the model.

There are three types of hyperparameters in the SKFA model, i.e., a) hyperparameters in the kernel functions, b) the number of sparse factors, and c) penalty coefficients to control the sparsity. The determination of hyperparameters is a problem in the field of model selection. A general solution is to use information criteria combined with specific search algorithms. In this paper, we propose to use the Information Complexity criterion (ICOMP) for model selection while taking into account goodness of fit, lack of parsimony, and model complexity.

We organize the chapter as follows. First, Section 4.2 describes the primary usage of kernel methods and the kernel functions. Then, Section 4.3 presents the statistical basis of SKFA, including parameter estimation and likelihood calculation. Next, Section 4.4 provides the derivation ICOMP for the SKFA model and uses it to construct the data analysis pipeline. Finally, Section 4.5 presents the results of the analysis of simulated data and the analysis of real-world data to assess the performance of the SKFA model.

4.2 Kernel Modeling in Feature Space

Kernel methods transform the $(n \times p)$ original data $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]$ into another data space (called feature space) by applying a nonlinear transformation. The most widely

used kernel methods are distance and similarity-based nonlinear transformation methods. In order to measure similarity between data points, the kernel function is of the form $k(.,.) : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$. Generally, the output of the kernel function is non-negative and has symmetric property $k(x_1, x_2) = k(x_2, x_1)$. Often, the concept of “similarity” is used in conjunction with the concept of “distance.” The more similar the two data points are, the closer they are to the data distribution space without further specification.

In machine learning, researchers often apply model to process the original data to create a feature vector (also called a feature embedding). For example, models can be more complex neural network models [143] or simpler models, such as tree models [144]. Along this line, if we use the kernel function to process the data, we can also obtain a feature vector as $\phi(\mathbf{x}') = [k(\mathbf{x}', \mathbf{x}_1), \dots, k(\mathbf{x}', \mathbf{x}_n)]$, corresponding to the data vector $\mathbf{x}'$.

There are no universally accepted methods for choosing the appropriate kernel function in the literature since it depends heavily on the substantive research question and data. Therefore, we present the information-theoretic model selection criteria for choosing an appropriate kernel function for a given dataset.

Table 4.1 below lists the most common and popular kernel functions. We note that the adjustable parameter σ in the above kernel functions plays a significant role in the performance of the kernel and should be carefully tuned based on the problem. For example, the exponential kernel will behave almost linearly if we overestimate the σ , and the higher-dimensional projection will lose its nonlinear properties. On the other hand, if we underestimate σ , the function will be less regularized, and the decision boundary will be very sensitive to noise.

The kernel function forms a Gram matrix K , known as a kernel matrix, which are inner products between transformed datasets. That is, given n data points (or observations) with p variables (or dimensions) in the dataset X , we define the kernel matrix as

$$K = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \cdots & k(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & \cdots & k(\mathbf{x}_n, \mathbf{x}_n) \end{bmatrix} \quad (4.1)$$

which is a $(n \times n)$ matrix. In other words, if we have an original dataset $X_{n \times p}$, the kernel function transforms it to a kernel matrix $K_{n \times n}$ in feature space, where all the further computational operations take place.

Another essential property of the kernel is that it allows us to model undersized data (or wide data). The undersized data has a limited number of observations but large dimensions ($n \ll p$). In this case of undersized data, many traditional statistical models will no longer work because of degeneracy. However, using the kernel approach can effectively handle similar challenges because the dimensionality of the kernel matrix is $n \times n$. This point has been overlooked in the literature in machine learning and in statistics. Therefore, this is also one of the advantages of using kernel-based methods in dimension reduction, classification, and clustering. In the numerical analysis section of this paper, we extensively use undersized data to support the validity of the kernel-based approach.

4.3 Sparse Kernel Factor Analysis

4.3.1 Factor Analysis (FA) Model

Charles Spearman proposed the first Factor Analysis model in 1904 for psychological analysis [44]. Because FA models are easily interpreted and can identify the sources of variation in a set of variables, they are widely used in economics [96], sociology [31], and finance [82], to mention a few.

From a statistical perspective, FA forms the view that the visible variables are developed through the linear combination of latent variables. In other words, factor analysis can reveal hidden dimensions in the data, which offers a different perspective on it, and may result in a much simpler and more informative representation. In addition, if we modify the assumptions/conditions of the FA model, we can obtain the probabilistic principal component analysis (PPCA) and the independent component analysis (ICA).

Below we give the matrix representation of the FA model. Assume an observation $\mathbf{z}$ has n visible variables. We have

$$\underset{(n \times 1)}{\mathbf{z}} = \underset{(n \times m)(m \times 1)}{\Lambda} \underset{(m \times 1)}{f} + \underset{(n \times 1)}{\mu} + \underset{(n \times 1)}{\epsilon} \quad (4.2)$$

where f represents the unobserved latent variable, and the Λ matrix shows the linear relationship between the latent variables and the visible variables. The μ in equation (4.2) denotes the mean value of $\mathbf{z}$ and ϵ is the model's error term.

In the FA model, we note that we do not know the factor loading Λ or factor scores f . Therefore, we need to estimate them from the dataset. Both f and ϵ are assumed to be multivariate Gaussian distributed.

The latent variables or the common factors in f are of zero mean, independent of each other with unit variance:

$$\mathbb{E}(f) = 0, \quad (4.3)$$

$$\Sigma_f = \text{Cov}(f) = I, \quad (4.4)$$

$$f \sim N(0, I_m). \quad (4.5)$$

The random error vector or the noise in ϵ is zero-mean, independent of a diagonal covariance matrix:

$$\mathbb{E}(\epsilon) = 0, \quad (4.6)$$

$$\Sigma_\epsilon = \text{Cov}(\epsilon) = \Psi = \text{diag}(\psi_1, \dots, \psi_n), \quad (4.7)$$

$$\epsilon \sim N_n(0, \Psi). \quad (4.8)$$

The error term in equation (4.7) is a diagonal matrix, and the diagonal elements are different. The FA model reduces to a principal component analysis (PCA) model if all diagonal elements of the error term is spherical, i.e., $\Psi = \sigma^2 I$.

The latent variables and the random noise are independent of each other

$$\Sigma_{f\epsilon} = \text{Cov}(f, \epsilon) = 0. \quad (4.9)$$

Under the multivariate Gaussian assumption, the FA model's covariance matrix is expressed as

$$\Sigma = \Lambda\Lambda' + \Psi \quad (4.10)$$

$$p(z \mid \Lambda, \Psi) = N(\mu, \Lambda\Lambda' + \Psi) \quad (4.11)$$

Let R be an arbitrary orthogonal rotation matrix, satisfying $RR' = I$. Then, as long as Λ is a reasonable solution, ΛR is also a solution satisfying equation (4.10). In other words, because the Σ matrix decomposition does not provide deterministic results, we cannot compute the Λ matrix using parameter estimation methods without adding other constraints. This problem is known as the *unidentifiability* issue in the FA model that we mentioned above.

In order to ensure that the parameter estimation methods can yield unique answers, we must introduce certain restrictions that limit the flexibility of the variables. There are two mainstream solutions currently available.

The first solution is to impose restrictions on factor loading. One of the more common restrictions is requiring that the factor loading Λ be an orthonormal matrix. This approach commonly appears in solving PCA models. Another idea is that Λ has a specific structure. For example, the Λ must be a lower triangular matrix. By forcing the elements of the upper triangular part of the loading factor to be zero, the result of the matrix decomposition of Σ is deterministic.

The second approach is to use the idea of the Bayesian model. We can assume that the elements of the Λ matrix all come from the same prior distribution, which removes the freedom of the matrix factorization. For example, Guan introduced Laplace distribution as the prior distribution of factor loading in the Sparse Probabilistic Principal Component Analysis (SPPCA) [67], and the final computational result is a sparse matrix. We can also

use Bayesian model for factor scores. For example, assuming that factor scores f are from a non-Gaussian prior distribution, the FA model transforms into an Independent Component Analysis model [79].

Our paper aims to analyze kernelized data using sparse factor analysis. The function of sparsity in SFA is to ensure that the results are well interpreted and to limit the solution space. Based on this viewpoint, SKFA is similar to the usage of Laplace prior. Nevertheless, the parameter estimation methods we present in Sections 4.3.4 and 4.3.5 are not the same as Bayesian parameter estimation. Bayesian parameter estimation methods use MCMC and Gibbs sampling, whereas our method uses SVD and EM algorithms.

4.3.2 Kernel Factor Analysis (KFA) and Sparse Kernel Factor Analysis (SKFA)

Various methods for analyzing kernelized data have emerged in the past few years, including the KPCA [115] and KCCA [62]. Moreover, the SKFA method we propose in this paper is consistent with what we have seen in the previous studies, namely, using the standard SFA algorithm for the kernelized data.

To model factor analysis (FA) in the feature space, we replace the original variables of interest with their transformations via the kernel function and define the factor analysis (KFA) model in the feature space as

$$\underset{(n \times 1)}{\mathbf{z}} = \underset{(n \times 1)}{\phi(x)} = \underset{(n \times m)(m \times 1)}{\Lambda^\phi f^\phi} + \underset{(n \times 1)}{\mu^\phi} + \underset{(n \times 1)}{\epsilon^\phi} \quad (4.12)$$

where superscript ϕ indicates the association with the transformation in the feature space.

We will now drop the superscript ϕ in our explanation for the sake of simplicity and practicality. This dropping is because the exact computations will occur in the feature space as they did in the original data space. However, in the feature space, the dimensions of the data points are n , the number of data points, instead of p , the number of variables (or dimensions) of the original dataset.

The KFA model in (4.12) can also be conceptualized as a multivariate regression (MVR) model in matrix form given by

$$\underset{(n \times n)}{K} = \underset{(n \times m)}{F} \underset{(m \times n)}{\Lambda'} + \underset{(n \times n)}{E}, \quad (4.13)$$

where K is an $(n \times n)$ kernel data matrix, F is an $(n \times m)$ unobserved factor score matrix, Λ' is the transpose of $\Lambda(n \times m)$ factor loading matrix, and E an $(n \times n)$ random error or residual matrix. We further assume that $E \sim N(O, \Sigma \otimes I)$, and $\text{Corr}(K, E) = 0$. More on matrix formulation, see Yates (1987) [155], and Yang and Bozdogan (2011) [153].

We constructed the SKFA algorithm since the data after kernelization is well suited for FA and SFA methods. Taking an example from the original data, if two points are located very close to each other, then after kernelization, one could expect that the feature vectors corresponding to the two points will be similar. Since the kernel matrix has symmetric properties, two similar feature vectors imply that two columns are also highly similar. Therefore, after FA and SFA analysis, the model will construct a single latent variable from the two similar columns. Moreover, we can deduce that each latent variable represents a cluster in the feature space. Therefore, the SFA algorithm is well suited to the kernelized data, i.e., the SKFA model we propose in this paper.

4.3.3 Maximum Likelihood Estimation

Our presentation in this section closely follows that of Anderson (1984) [7], Magnus and Neudecker (1999) [98], and Zhao et al. (2008) [158]. In what follows, we use more commonly accepted notation from the statistical literature.

The log-likelihood function for the factor analysis model is

$$\log L(\mu, \Lambda, \Psi) = -\frac{np}{2} \log(2\pi) - \frac{n}{2} \log |\Sigma| - \frac{1}{2} \sum_{i=1}^n (z_i - \mu)' \Sigma^{-1} (z_i - \mu) \quad (4.14)$$

where

$$\Sigma = \Lambda \Lambda' + \Psi. \quad (4.15)$$

Maximizing equation (4.14) for μ yields the maximum likelihood estimator (MLE) $\hat{\mu}$:

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n z_i. \quad (4.16)$$

By substituting $\hat{\mu}$ in equation (4.14), we obtain the concentrated log-likelihood function

$$\log_c L(\Lambda, \Psi) = -\frac{np}{2} \log(2\pi) - \frac{n}{2} [\log |\Sigma| + \text{tr}(\Sigma^{-1}S)], \quad (4.17)$$

where

$$S = \frac{1}{n} \sum_{i=1}^n (z_i - \bar{z})(z_i - \bar{z})' \quad (4.18)$$

is the sample covariance matrix. For practical reasons we drop the constant term $-\frac{np}{2} \log(2\pi)$ in equation (4.17) and obtain

$$\log_c L(\Lambda, \Psi) = -\frac{n}{2} [\log |\Sigma| + \text{tr}(\Sigma^{-1}S)]. \quad (4.19)$$

Using the matrix differential calculus via Magnus and Neudecker [98], we have

$$\frac{\partial \log_c L(\Lambda, \Psi)}{\partial \Lambda} = -\frac{n}{2} [\Sigma^{-1}\Lambda - \Sigma^{-1}S\Sigma^{-1}\Lambda] = 0, \quad (4.20)$$

$$\frac{\partial \log_c L(\Lambda, \Psi)}{\partial \Psi} = -\frac{n}{2} \text{diag}(\Sigma^{-1} - \Sigma^{-1}S\Sigma^{-1}) = 0. \quad (4.21)$$

Jöreskog [83] noted that it is difficult to solve equations (4.20) and (4.21) simultaneously for Λ and Ψ . Thus, it is necessary to apply iterative numerical procedures to maximize log-likelihood $\log_c L(\Lambda, \Psi)$.

4.3.4 SVD-based Parameter Estimation Approach

In this section, we first describe the procedure for estimating the parameters of FA using SVD based on Zhao [158] and Barber [11]. Additionally, we compute the likelihood in terms of eigenvalues. Moreover, the eigenvalue also facilitates the calculation of the information complexity criterion of the model later.

Parameter estimation using the SVD method involves two steps:

- 1 Fix Ψ , calculate Λ by maximizing log-likelihood $\log_c L(\Lambda, \Psi)$.
- 2 Calculate the new Ψ .

This algorithm is called the SVD-based method because calculating the eigenvalues, which is the most vital part of the algorithm, relies on SVD in its first step and can be calculated using a deterministic process. If the Ψ (which quantifies how confident we are in our prediction) is determined, we avoid problems such as unidentifiability.

In Section 4.3.3, we have obtained the equation (4.14) for likelihood and further simplified it to equation (4.19). For computing likelihood directly using equation (4.19), it is necessary to compute $|\Sigma|$ and Σ^{-1} beforehand. Due to the possibility of a singularity, these two components are computationally intensive and prone to producing imaginary results. However, the SVD-based parameter estimation eliminates the two issues inherent to the usual Maximum Likelihood Estimation (MLE) approach. Here we describe how to reparameterize the likelihood according to its eigenvalues, simplifying estimation and producing more accurate estimates.

First, we solve equation (4.20), obtaining

$$\Lambda = S\Sigma^{-1}\Lambda \quad (4.22)$$

We define

$$\tilde{\Lambda} \equiv \Psi^{-\frac{1}{2}}\Lambda, \quad (4.23)$$

$$\tilde{S} = \Psi^{-\frac{1}{2}}S\Psi^{-\frac{1}{2}}. \quad (4.24)$$

We can write $\tilde{\Lambda}$ as $U_m L W'$ if we assume it is a thin SVD. Moreover, it is possible to decompose the SVD here using only $\min(p, n)$, which saves computing time and space when the rows and columns of the original matrix are very different (e.g., $p \gg n$). The $U_m (n \times m)$ and $W (m \times m)$ are orthogonal matrices with $U_m' U_m = I_m$ and $W' W = I_m$. The L represents $\text{diag}(l_1, \dots, l_m)$ which includes singular values of $\tilde{\Lambda}$.

Using new parameterization,

$$\tilde{S} = UDU' \quad (4.25)$$

$$\Sigma = \Psi^{\frac{1}{2}}(\tilde{\Lambda}\tilde{\Lambda}' + I)\Psi^{\frac{1}{2}} \quad (4.26)$$

where $U = (u_1, u_2, \dots, u_m, u_{m+1}, \dots, u_p)$ is $p \times p$ orthogonal matrix.

Then,

$$tr(\Sigma^{-1}S) = tr((\tilde{\Lambda}\tilde{\Lambda}' + I)^{-1}\tilde{S}) \quad (4.27)$$

$$= tr((I + U_m L^2 U_m')^{-1}\tilde{S}) \quad (4.28)$$

$$= tr([U \text{diag}(\lambda_1, \dots, \lambda_m, 1, \dots, 1)U']^{-1}\tilde{S}) \quad (4.29)$$

$$= tr([U \text{diag}(\lambda_1^{-1}, \dots, \lambda_m^{-1}, 1, \dots, 1)U']\tilde{S}) \quad (4.30)$$

$$= tr(U \text{diag}(\lambda_1, \dots, \lambda_m, 1, \dots, 1)U' U \text{diag}(\lambda_1, \dots, \lambda_p)U') \quad (4.31)$$

$$= tr(U \text{diag}(1, \dots, 1, \lambda_{m+1}, \dots, \lambda_p)U') \quad (4.32)$$

$$= m + \sum_{i=m+1}^p \lambda_i. \quad (4.33)$$

Similarly, we derive the $\log(|I + \tilde{\Lambda}\tilde{\Lambda}'|)$ as

$$\log \det(I + \tilde{\Lambda}\tilde{\Lambda}') = \log \det(U \text{diag}(\lambda_1, \dots, \lambda_m, 1, \dots, 1)U') \quad (4.34)$$

$$= \log(UU' \prod_{i=1}^m \lambda_i) \quad (4.35)$$

$$= \sum_{i=1}^m \log \lambda_i. \quad (4.36)$$

Thus, the eigenvalue representation of likelihood is

$$-\frac{2}{n} \log L(\Lambda, \Psi) = \sum_{j=1}^m \log(\lambda_j) + m + \sum_{j=m+1}^n \lambda_j + \log |2\pi\Psi| \quad (4.37)$$

In other words, we can reliably compute the likelihood value with a Ψ matrix by calculating the eigenvalues. Moreover, this calculation does not require $\log \det(\Sigma)$ or Σ^{-1} as part of the equation.

There are several ways to calculate Ψ . Commonly, the conditional likelihood is maximized by first creating a random diagonal matrix (the diagonal elements must be positive) and then updating each diagonal element one at a time. (See [158] for details.) In our study, the Ψ is computed in advance by the EM algorithm. Then we use the Ψ to compute eigenvalues, the log-likelihoods, and the ICOMP. For more information about EM, see Section 4.3.5.

4.3.5 EM-based Parameter Estimation Approach

Aside from SVD-based parameter estimation, researchers in the field of machine learning utilize Dempster et al.'s (1977) [47] Expectation-Maximization (EM) algorithm to estimate the unknown parameters of the SKFA model. The EM algorithm is a widely applicable method of maximum computing likelihood (ML) estimates iteratively and is helpful in many incomplete-data problems. In the following section, we will show that, compared to SVD-based algorithms, the EM algorithm has the capability of customizing the objective function to incorporate log-likelihood along with penalty terms.

Here is a summary of the EM algorithm for the Gaussian KFA model. Consider the KFA model $z = \phi(x) = \Lambda^\phi f^\phi + \mu^\phi + \epsilon^\phi$ in equation (4.12). Given Λ_{init} and Ψ_{init} , the initial values of the loading matrix λ and the error covariance matrix Ψ , respectively.

E-Step

Compute $\mathbb{E}(f \mid z)$ and $\mathbb{E}(ff' \mid z)$ for the data vector z in the feature space

$$\mathbb{E}(f \mid z) = \Lambda'(\Lambda\Lambda' + \Psi)^{-1}z = \beta z \quad (4.38)$$

$$\mathbb{E}(ff' \mid z) = \Sigma_{f|z} + \mathbb{E}(f \mid z)\mathbb{E}(f \mid z)' = I - \beta\Lambda + \beta zz'\beta' \quad (4.39)$$

along with the expected value of the log-likelihood function of the KFA model parameters $\theta = \{\Lambda, \Psi\}$ given by

$$Q = \mathbb{E} [\log L(\Lambda, \Psi)] \quad (4.40)$$

$$= c - \frac{n}{2} \log |\Psi| - \frac{1}{2} \sum_{i=1}^n \mathbb{E} [z_i' \Psi^{-1} z_i - f' \Lambda' \Psi^{-1} z_i - z_i' \Psi^{-1} \Lambda f + f' \Lambda' \Psi^{-1} \Lambda f]. \quad (4.41)$$

M-Step

Based on the data and the previous parameters in the E-step, update the parameters to maximize the expected likelihood (or the log-likelihood function).

$$\Lambda^{\text{new}} = \left(\sum_{i=1}^n z_i \mathbb{E}[f | z_i]' \right) \left(\sum_{i=1}^n \mathbb{E}[f f' | z_i]' \right)^{-1} \quad (4.42)$$

$$\Psi^{\text{new}} = \frac{1}{n} \text{diag} \left(\sum_{i=1}^n z_i z_i' - \Lambda^{\text{new}} \mathbb{E}(f | z_i) z_i' \right) \quad (4.43)$$

where the diag denotes the diagonals of the Ψ^{new} matrix with off-diagonal elements equal to zero.

Throughout this discussion, it is worth noting that equations (4.42) and (4.43) are derivations derived from updated formula computed to maximize log-likelihood. These formulas no longer hold if the objective function is not log-likelihood but a penalized version.

As a result, the objective function with penalty term (i.e., the Q function in the M-step) is

$$Q = \mathbb{E} [\log L(\Lambda, \Psi)] - \frac{1}{2} P(\Lambda), \quad (4.44)$$

where $P(\Lambda)$ denotes a non-negative penalty term. Due to this paper's objective of computing sparse factor loadings, we can apply the L_1 penalty term for factor loading, i.e., $P(\Lambda) = \gamma \sum_{i=1}^n \sum_{j=1}^m |\Lambda_{ij}|$.

Choi et al. [41] use the Lasso-based algorithm to calculate Ψ and Λ to update the M-step. However, our analysis of Choi et al.'s paper revealed severe errors that led to an incorrect

update formula for Ψ . Once the errors are corrected, the Ψ update process becomes

$$\Psi_{jj}^{\text{new}} = \Sigma_{jj}^2 - 2\Sigma[j,]\delta\Lambda^{\text{old}}[j] + (\Lambda^{\text{old}}[j])' [\Delta + \delta'\Sigma\delta] \Lambda^{\text{old}}[j] \quad (4.45)$$

$$\delta = (\Psi^{\text{old}} + \Lambda^{\text{old}}(\Lambda^{\text{old}})')^{-1}(\Lambda^{\text{old}})' \quad (4.46)$$

$$\Delta = I - (\Lambda^{\text{old}})'\delta \quad (4.47)$$

The corresponding formula for updating Λ is

$$\Lambda'[j] = \arg \min_{\beta} \|(\Sigma[j,]\delta Z^{-1})' - Z\Lambda'\|_2^2 + \gamma \frac{\Psi_{jj}}{n} |\Lambda|, \quad (4.48)$$

where Z is from Cholesky decomposition $Z'Z = \Delta + \delta'\Sigma\delta$, and γ is a hyperparameter that controls the intensity of the penalty (see Appendix C).

4.3.6 Estimating Sparse Kernel Factor Scores

So far, we have studied and derived the estimation of the parameters of the KFA and SKFA models. The following section aims to reverse the problem, show how the factors depend on the features, and further calculate the factor scores. Using the SKFA model, we can reduce the feature space dimension and compute factor scores as the new explanatory variables. Later, we can construct supervised models using these new explanatory variables.

From feature space, one can estimate factor scores in several different ways. Here we will focus on two of the most popular methods, namely (a) the weighted least squares method and (b) regression analysis. Using the EM algorithm described in Section 4.3.5 above, we assume we already have the estimated factor loading matrix $\hat{\Lambda}$ and the uniqueness covariance matrix $\hat{\Psi}$.

Weighted Least Square Method

To determine the factor scores, $f^\phi = (f_1, f_2, \dots, f_m)'$ for different feature observations z , we consider the KFA model again

$$z = \phi(x) = \Lambda^\phi f^\phi + \mu^\phi + \epsilon^\phi. \quad (4.49)$$

Supposing that factor loading matrix $\hat{\Lambda}$ estimates and the uniqueness covariance matrix $\hat{\Psi}$ are available from the EM algorithm, we minimize the sum of squares of the weighted residuals

$$\sum_{i=1}^n \frac{\epsilon_i^2}{\psi_i} = \epsilon' \Psi^{-1} \epsilon = (z - \Lambda f - \mu)' \Psi^{-1} (z - \Lambda f - \mu). \quad (4.50)$$

Then we have

$$\hat{f} = (\Lambda' \Psi^{-1} \Lambda)^{-1} \Lambda' \Psi^{-1} (z - \mu). \quad (4.51)$$

An estimate of the factor score for the i -th observation z_i is, therefore,

$$\hat{f}_i^{\text{WLS}} = \left(\hat{\Lambda}' \hat{\Psi}^{-1} \hat{\Lambda} \right)^{-1} \hat{\Lambda}' \hat{\Psi}^{-1} (z_i - \bar{z}), \quad i = 1, \dots, n. \quad (4.52)$$

The Regression Method

Using the the joint Gaussian distribution of z and f :

$$p \left(\begin{bmatrix} z \\ f \end{bmatrix} \right) = \mathcal{N} \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \Lambda \Lambda' + \Psi & \Lambda \\ \Lambda' & I \end{bmatrix} \right) \quad (4.53)$$

we calculate the conditional expectation of the factor score f_i

$$\mathbb{E}(f_i | z_i) = \Lambda' (\Lambda \Lambda' + \Psi)^{-1} (z_i - \mu), \quad (4.54)$$

$$\Sigma_{f|z} = \text{Cov}(f | z) = I_m - \Lambda' (\Lambda \Lambda' + \Psi)^{-1} \Lambda. \quad (4.55)$$

Equation (4.54) gives the regression of f on z . For $z = z_i$, substituting the maximum likelihood estimates of μ , Λ , and Ψ in equation (4.54), the factor score f_i is, therefore, obtained from

$$\hat{f}_i^{\text{R}} = \hat{\Lambda}' \left(\hat{\Lambda}' \hat{\Psi}^{-1} \hat{\Lambda} \right)^{-1} (z_i - \bar{z}), \quad i = 1, \dots, n. \quad (4.56)$$

There is no doubt that equation (4.54) is the more fundamental calculation method. Plugging the Λ and Ψ from the MLE estimation, the original formula is simplified to form the equation (4.56). However, in this study, the objective function is not a plain log-likelihood but a version with the L_1 penalty term. Hence, we should use the Λ and Ψ estimates calculated by the EM algorithm directly in equation (4.54) to compute factor scores.

4.3.7 Choosing the Number of Sparse Kernel Factors

In many cases, the number of factors m is unknown to researchers, or we may not even be able to postulate the specific value/range of m . As a result, estimating the appropriate number of factors is a common problem we need to solve in factor analysis. Because m is a hyperparameter, the statistical or machine learning problem here is not hypothesis testing or parameter estimation. Instead, it is a model selection problem determining the appropriate number of factors.

There is a correlation between determining the number of factors and the unidentifiability problem of the FA model itself. This correlation can aid in determining the upper bound of m the number of factors. According to [29, 104], after considering the uniqueness condition $\hat{\Lambda}'\hat{\Psi}^{-1}\hat{\Lambda} \equiv \Delta$ and the $(mp + p)$ parameters, the covariance matrix contains

$$s = (mp + p) - \frac{1}{2}m(m - 1) \quad (4.57)$$

free parameters. Furthermore, they calculate that the number of factors must coincide with the following bound

$$m \leq \left\lfloor \frac{1}{2} [2p + 1 - (8p + 1)^{1/2}] \right\rfloor \quad (4.58)$$

This upper bound is called the *big factor*. It is the most number of factors we can allow while maintaining the number of free parameters in the factor model being less than the number of parameters in the covariance matrix.

This paper uses the information-theoretic measure of complexity (ICOMP) criterion due to [17, 18, 21, 25] for model selection to choose the number of KFA and SKFA in the next section.

4.4 Information Complexity Criterion

A long-standing problem is how to choose the number of factors required to construct the FA model. Literature has extensively explored the problem of poor performance and interpretation of models when the number of factors is inappropriately selected. Hubbard & Allen [77] argues in their historical studies that this problem of determining the total number of factors is thorny and that there is no authoritative method of measuring it. Zwick & Velicer [162, 163] emphasizes that this is crucial in constructing FA models.

Researchers argue that number-of-factor problem is complex for two reasons: 1) lack of consistent criteria for determining training results, and 2) the number of factors is not a model parameter; therefore, we cannot obtain an exact value through parameter estimation.

In recent studies, methods related to model selection have been increasingly used to determine the number of factors (e.g., [107]). The model selection process involves comparing the performance and complexity of several different statistical models. The concept of model selection, in a broad sense, can be applied to any statistical model, regardless of the relationship between them. However, our paper uses the model selection in a narrow sense, i.e., selecting models from the same design that differ only in hyperparameter values.

The researchers generally agree that a criterion should consider how well the model fits the current data and its ability to generalize to unknown future samples. This criterion is also known as the bias-variance tradeoff in machine learning [133]. For example, when a statistical model consists of relatively few parameters, it is considered a simpler model with better generalization ability but a limited ability to fit the current data set. AIC [5, 6] and BIC [70] are the first information criteria developed under this concept. Bozdogan improved the AIC/BIC by proposing an information complexity criterion (ICOMP) based on the complexity of the model [17, 18, 19, 20, 21, 24, 153]. In particular, ICOMP needs to assess the model fit, model complexity, and lack of parsimony.

The ICOMP for the SKFA model is

$$\text{ICOMP}(\hat{\Sigma}) = -2 \log L(\hat{\theta}) + 2C_{1F}(\hat{\Sigma}), \quad (4.59)$$

$$C_{1F}(\hat{\Sigma}) = \frac{1}{4\bar{\lambda}_a^2} \sum_{j=1}^s (\lambda_j - \bar{\lambda}_a)^2, \quad (4.60)$$

$$s = \text{rank}(\hat{\Sigma}) = \dim(\hat{\Sigma}), \quad (4.61)$$

where λ represents the eigenvalues of $\hat{\Sigma}$ and $\bar{\lambda}_a$ is the average of all eigenvalues.

Contrary to other Factor Analysis models, the SKFA model we propose in this paper involves two hyperparameters, the number of factors and the coefficient of L_1 penalty term. It is possible to simultaneously search the two hyperparameters using grid search, random search, and the SA-PSO method (see Chapter 2). In this paper, for the sake of simplicity, we use grid search to determine the optimal hyperparameters. The ICOMP value is minimized when a set of hyperparameters corresponds to the optimal model.

4.5 Numerical Examples and Computational Results

4.5.1 A Toy Example Using Iris Data

As a first demonstration, we use IRIS data to illustrate the use of SKFA. To determine the most appropriate number of factors and the coefficient value γ of the L_1 penalty term, we perform a grid search of the hyperparameter based on the ICOMP values. On a logarithmic scale, the search range for the penalty coefficient is $[10^{-0.35}, 10^{0.35}]$. Tables 4.2, 4.3 and 4.4 show a summary of the search results. The tables correspond to different kernel functions, and the values in the tables define the ICOMP values. Therefore, we select the model with the lowest ICOMP value as the optimal model.

Table 4.5 shows that KFA extracts more factors than SKFA. SKFA, for example, obtains the minimum ICOMP value (i.e., the model performs optimally) using eight factors, whereas KFA requires twenty-six elements under the same conditions. The number of factors extracted with an Exponential kernel and a Laplace kernel is also lower than those obtained

with a non-sparse KFA, indicating that our proposed method can lead to a more sparse statistical model; therefore, the model demonstrates greater explanatory power.

Figure 4.1 shows the ICOMP mesh plot for Gaussian kernel functions. Our observations indicate that when γ is set too high, the ICOMP computation degenerates (collapses on the mesh plot). For instance, when $\gamma > 1.8$, the value of ICOMP does not increase but decreases as the model includes more factors. Specifically, too large γ values cause the model's coefficients to be equal to zero, thereby deteriorating the goodness of fit. At the same time, C_{1F} prefers the sparse model, and the values corresponding to it also diminish. Different kernel functions exhibit this collapse phenomenon (see Figures 4.2 and 4.3).

The different kernel functions in Figure 4.4 are selected separately for comparison purposes. The curves first involve choosing the global optimum then drawing the corresponding ICOMP curve. We observe that the Gaussian kernel has the smallest ICOMP value, corresponding to the global optimum. Moreover, when the number of factors is too large, both the Laplace kernel and the Exponential kernel become non-smooth.

Figures 4.5, 4.6 and 4.7 display the ICOMP curves corresponding to non-sparse KFA and SKFA. The Gaussian kernel in Figure 4.5 shows the most obvious difference, with twenty-six factors selected for non-sparse KFA, a typical dense model. On the other hand, SKFA picks the sparse model with only eight factors, but the corresponding ICOMP values are higher.

Fisher iris data has four-dimensional measurements of irises in the original data space and related species classifications. Figure 4.8 shows the original raw data of iris groups in the first panel of the tSNE plot. The tSNE plot corresponds to Gaussian SKFA scores in the second panel, and in the third panel, it corresponds to Exponential SKFA scores. The fourth tSNE plot corresponds to Laplace SKFA scores. SKFA factor scores can improve the separation of Versicolor and Virginia in the feature space. The two groups are now clearly separated.

Table 4.1: Selected Kernel Functions and their hyperparameters from [123].

Kernel Functions	$k(\mathbf{x}_i, \mathbf{x}_j)$	Hyperparameters
Polynomial	$(\langle x_i, x_j \rangle + c)^d$	$c = \text{bias}, d = \text{degree}$
Linear	$\langle x_i, x_j \rangle$	$c = 0, d = 1$
Gaussian RBF (GRBF)	$\exp(-\frac{\ \mathbf{x}_i - \mathbf{x}_j\ ^2}{2\sigma^2})$	$\sigma = \text{scale}$
Exponential RBF (ERBF)	$\exp(-\frac{\ \mathbf{x}_i - \mathbf{x}_j\ }{2\sigma^2})$	$\sigma = \text{scale}$
Laplace	$\exp(-\frac{\ \mathbf{x}_i - \mathbf{x}_j\ }{\sigma})$	$\sigma = \text{scale}$
Cauchy	$(1 + \frac{\ \mathbf{x}_i - \mathbf{x}_j\ ^2}{\sigma^2})^{-1}$	$\sigma = \text{scale}$
Sigmoid	$\tanh[a \langle \mathbf{x}_i, \mathbf{x}_j \rangle + c]$	$a = \text{scale}; c = \text{bias}$

Table 4.2: ICOMP values using Gaussian kernel.

		γ				
		0.76	0.91	1.09	1.31	1.56
Number of Factors	6	-139.19	-158.82	-146.27	-153.35	-158.94
	7	-164.86	-129.48	-156.33	-168.95	-162.46
	8	-161.82	-173.52	-176.57	-165.26	-166.14
	9	-171.90	-137.21	-151.27	-153.87	-149.34
	10	-146.91	-152.74	-141.82	-135.39	-151.49

Table 4.3: ICOMP values using Exponential kernel.

		γ				
		0.76	0.91	1.09	1.31	1.56
Number of Factors	6	-74.83	-90.80	-87.70	-89.00	-87.57
	7	-92.14	-95.32	-94.24	-97.10	-95.43
	8	-96.14	-97.33	-98.38	-96.07	-96.01
	9	-86.64	-90.44	-90.90	-89.80	-88.92
	10	-77.15	-84.06	-82.92	-82.08	-78.66

Table 4.4: ICOMP values using Laplace kernel.

		γ				
		0.76	0.91	1.09	1.31	1.56
Number of Factors	6	8.27	9.66	7.79	31.45	6.47
	7	-1.02	0.33	0.83	3.65	0.85
	8	-1.21	-1.60	-4.55	0.70	-3.03
	9	-2.41	-1.14	-1.30	13.71	4.88
	10	5.91	6.35	6.47	-2.33	-2.00

Table 4.5: ICOMP values of the Sparse KFA and Non-sparse KFA using the Iris Data.

Kernel Functions	Sparse KFA			Non-Sparse KFA	
	Number of Factors	γ	ICOMP	Number of Factors	ICOMP
Gauss	8	1.09	-176.57	26	-1179.74
Exponential	10	0.91	-98.38	14	-261.12
Laplace	10	1.56	-4.55	14	-88.47

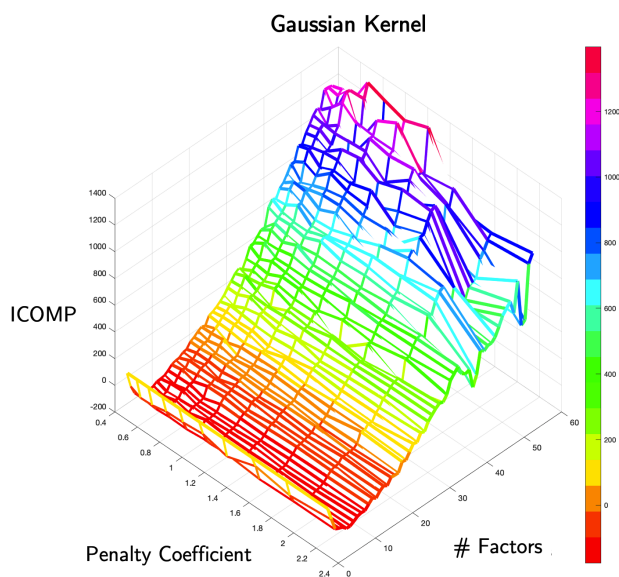


Figure 4.1: Mesh plot of ICOMP values for Gaussian Kernel.

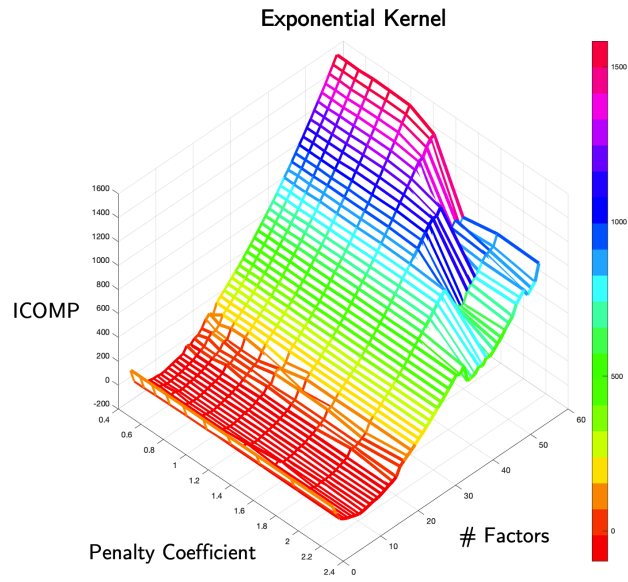


Figure 4.2: Mesh plot of ICOMP values for Exponential Kernel.

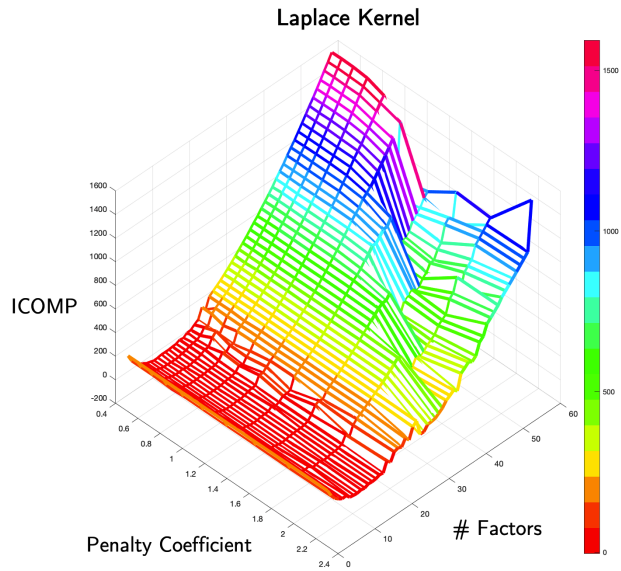


Figure 4.3: Mesh plot of ICOMP values for Laplace Kernel.

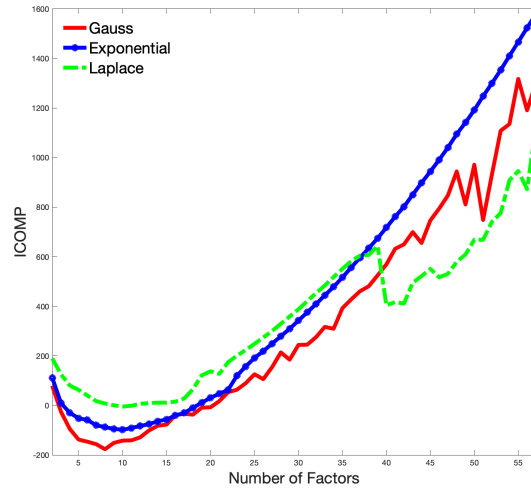


Figure 4.4: Comparison of ICOMP values for different kernel functions.

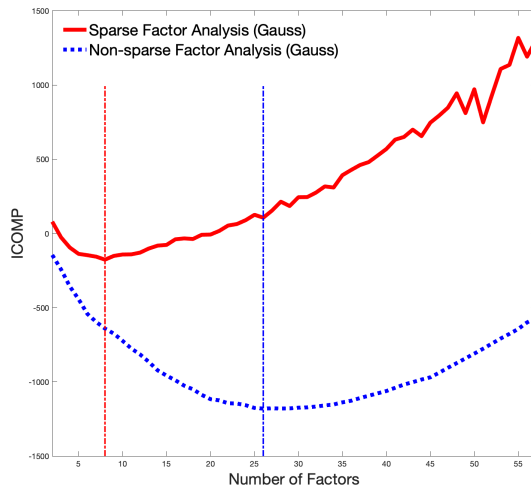


Figure 4.5: Sparse vs. Non-sparse KFA using the Gaussian kernel.

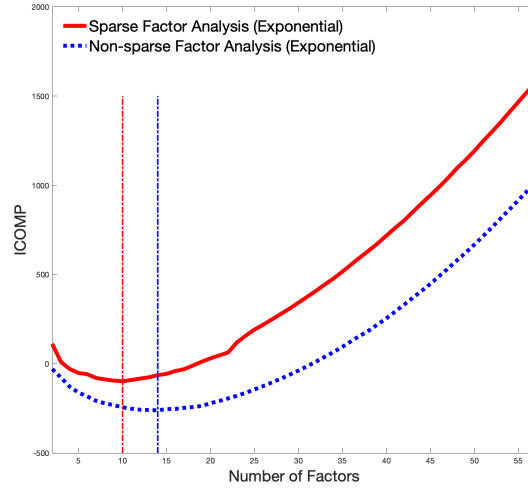


Figure 4.6: Sparse vs. Non-sparse KFA using the Exponential kernel.

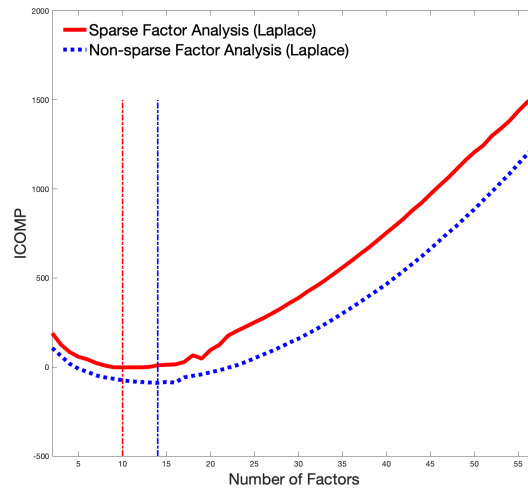


Figure 4.7: Sparse vs. Non-sparse KFA using the Laplace kernel.

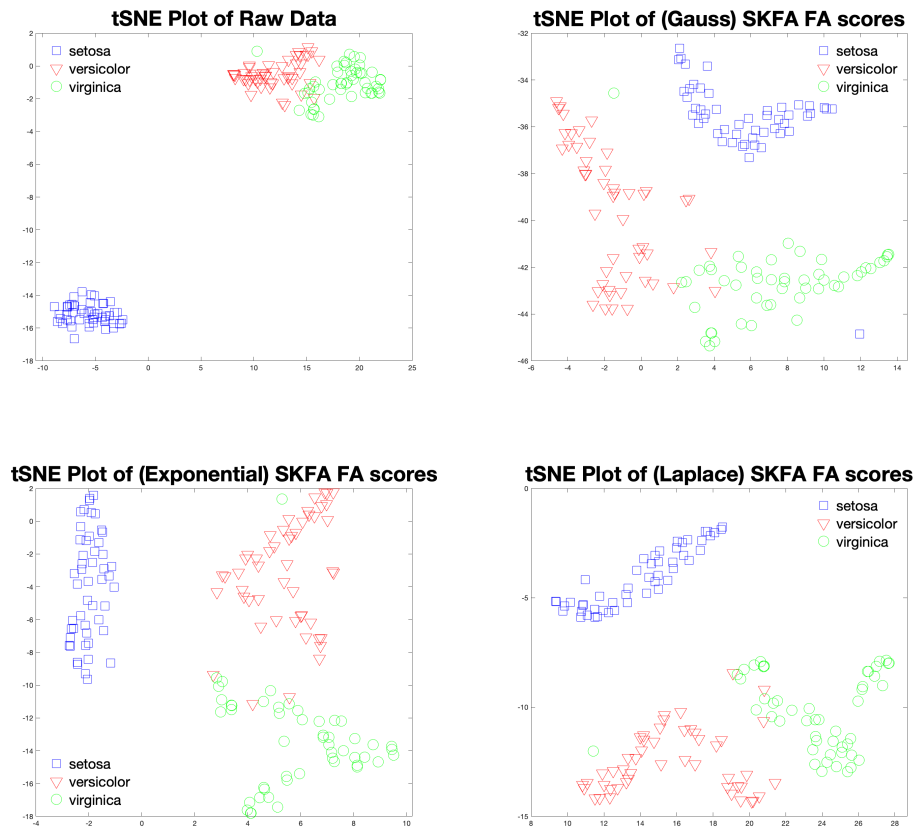


Figure 4.8: t-Distributed Stochastic Neighbor Embedding (tSNE) plot of iris data.

4.5.2 Monte Carlo Evidence: Stability Study of SKFA Algorithm via Pre-image Learning

We will demonstrate numerically that SKFA is a consistent algorithm throughout this section. The SKFA algorithm extracts influencing factors using the kernel space of the data, and we want to verify that the extraction process results are consistent. Specifically, we intend to impose some noise into kernel space. If the number of extracted factors is not affected by the noise for the same data set, we consider the results of the SKFA algorithm to be consistent and stable.

Simulation Protocol Setup

From [129], we design the simulation protocol of the synthetic data generation process. The setup consists of $p = 12$ variables and comprises $m = 4$ true number factors in the original data space. The factor loading matrix is

$$\Lambda = \begin{bmatrix} 1.8 & 0 & 0 & 0 \\ 1.8 & 0 & 0 & 0 \\ 1.8 & 0 & 0 & 0 \\ 0 & 1.7 & 0 & 0 \\ 0 & 1.7 & 0 & 0 \\ 0 & 1.7 & 0 & 0 \\ 0 & 0 & 1.6 & 0 \\ 0 & 0 & 1.6 & 0 \\ 0 & 0 & 1.6 & 0 \\ 0 & 0 & 0 & 1.5 \\ 0 & 0 & 0 & 1.5 \\ 0 & 0 & 0 & 1.5 \end{bmatrix} \quad (4.62)$$

and the specific factor covariance is

$$\Psi = \text{diag}(1.27, 0.61, 0.74, 0.88, 0.65, 0.81, 0.74, 1.3, 1.35, 0.74, 0.92, 1.32). \quad (4.63)$$

Hence, the covariance matrix of the FA model is

$$\Sigma = \Lambda\Lambda^T + \Psi. \quad (4.64)$$

Under the assumption of Gaussianity, i.e.,

$$x_i \sim \text{MVN}(0, \Sigma), i = 1, 2, \dots, n = 100. \quad (4.65)$$

We generate $n = 100$ observations from the above simulation protocol to study the consistency and stability of the SKFA Algorithm via pre-image learning in what follows.

Noisy Pre-image Learning

We need to introduce the pre-image technique to add noise to the kernel space. An example of pre-image learning is a learning task in which, given a point in the kernel space (designated by Φ), we attempt to find its corresponding point (represented by the symbol z) in the original data space ($\mathcal{X}$). Then, we can describe the loss function of the whole problem in terms of the optimization algorithm as

$$z = \arg \min ||\Phi - \Phi(z)||_2^2. \quad (4.66)$$

The pre-image learning problem has many applications in image denoising and signal processing. From the viewpoint of numerical simulation, pre-image learning maximizes the preservation of information in the kernel space. By introducing a certain amount of noise into the kernel space, our objective is to test the consistency and stability of the SKFA algorithm by using pre-image learning in the simulation protocol. Researchers have proposed different algorithms and computational methods to solve the loss function in equation (4.66). Jason Weston et al. [146] presented Kernel Dependency Estimation (KDE) based on computing the similarity of inputs and outputs. Gokhan et al. [10] introduced kNN and proposed using a linear combination of the k closest points to infer the location of the actual points. We have also seen more articles exploring nonlinear and iterative approaches to optimize the loss function [73].

However, most of these methods suffer from two limitations: dependence on application-specific scenarios and a lack of generalization capabilities. For instance, the exact pre-image approach requires a linear kernel function. On the other hand, the KDE method [146] relies upon a linear approximation between inputs and outputs.

In this paper, for the first time to our knowledge, we propose using a neural network approach to learn how to infer the location of the original data-space from the kernel space. This approach has a strong generalization capability extending to different kernel functions. The first step is to determine the input and output array dimensions. The neural network input is a kernel matrix $k(x_i, x'_i)$ (dimension $n \times n$), and the output is a predicted data matrix $\hat{x}_i(n \times p)$. The training data set is $\{(k(x_i, x'_i), x_i)\}$. Due to the fact that the kernel function may be called an infinite number of times, our training data set may theoretically contain any number of $(k(x_i, x_i^T), x_i)$ pairs. The training data for the neural network model involves 4,096 points.

Because the neural network model does not know the range of the actual data distribution, the optimization process of the neural network becomes infeasible if the output values are not constrained. Therefore, during training, we limit all output data to the interval $[0, 1]$ so that the model can rapidly converge.

Gradient disappearance and gradient explosion are other problems we encounter. Finally, we refer to the widely used regularization techniques, including L_1 and L_2 penalty terms and the dropout approach [128]. The combination of these algorithms significantly improves the prediction of the model.

With the loss function in equation (4.66), we can see that the model eventually learns how to reconstruct the kernel matrix in the feature space and make it similar to the given kernel matrix. The model, however, does not directly guarantee that the original data matrix x and the learned data $\hat{x}$ are one-to-one. For example, our numerical experiments reveal that x and $\hat{x}$ are structurally different, but their kernel matrices $k(x, x')$ and $k(\hat{x}, \hat{x}')$ are structurally similar.

Our neural network model structure is in Table 4.6 below.

Figure 4.9 summarizes the neural network pre-image learning based on the Monte Carlo simulation to verify the consistency and stability of the SKFA algorithm. As described

previously, the output $\hat{x}'$ of the neural network preserves the spatial information of the kernel space. We then approximate a data point that is close to $k(x, x')$ using $k(\hat{x}, \hat{x}')$. A neural network approach makes it possible to obtain a kernel matrix that keeps the original structure. Moreover, we can find its corresponding $\hat{x}$ to verify whether the SKFA algorithm is robust and consistent for dimension reduction.

The neural network reconstructs the kernel matrix and maintains its original structure, as shown in figure 4.10. Using ICOMP, SKFA can identify thirteen factors in the original kernel matrix and fourteen in the reconstructed kernel matrix, which is pretty close. ICOMP values are more aggregated in the reconstructed kernel matrices than in the original kernel while still following a similar trend.

Because of the computational intensity, we performed the numerical experiment ten times and recorded the number of SKFs associated with the minimum ICOMP value. In Table 4.7, we summarize our results from this experiment. Compared with the reconstructed pre-image kernel, the number of SKFs varies within a relatively large range. By contrast, we obtained consistent results with the reconstructed pre-image kernel.

4.5.3 Real Benchmark Examples Using High Dimensional Wide-Datasets

This section uses large datasets from different sources to illustrate SKFA’s sparsity and interpretability compared to KFA’s. We also show how to identify hyperparameters and parameters using ICOMP.

Datasets used in this section have characteristics known as “wide data,” i.e., the data have more dimensions than the observable measurements. In this case of such high-dimensional datasets, traditional statistical models require considerable pre-processing (e.g., dimension reduction, feature selection, etc.). On the other hand, SKFA is well suited to handling high-dimensional data, as shown in this section, because SKFA relies on the computation of the kernel matrix. And a direct projection of the high-dimensional data into the nonlinear space would be more straightforward for classification purposes.

Table 4.6: Neural network model with 65,508,700 parameters in total.

Layer	Output Shape	# Parameter
Input Layer	(100,100)	0
Reshape	10,000	0
Dense Layer 1	5,000	50,005,000
Dense Layer 2	2,500	12,501,250
Sigmoid Layer	1,200	3,001,200
Reshape Layer	(100,12)	0

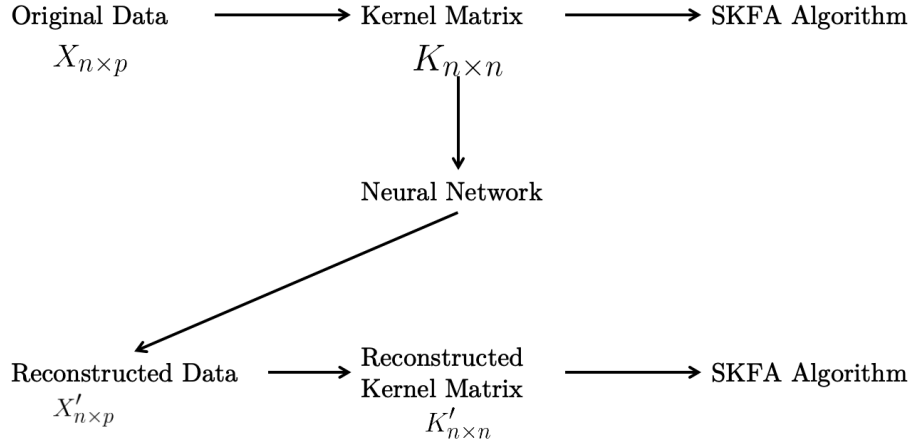


Figure 4.9: Flow chart of simulation using neural network for pre-image learning.

Table 4.7: Hit frequencies of choosing number of factors using Gaussian kernel in 10 replication of the experiment with SKFA and Pre-image analysis with SKFA.

Number of Factors Fitted	SKF (No Pre-image)	Pre-image analysis
2-10	0	0
11	2	0
12	2	0
13	3	0
14	1	10
15	0	0
16	0	0
17	2	0
18-30	0	0

According to the previous discussion, minimum ICOMP values indicate better model performance. With the grid search shown in Tables 4.8, 4.9 and 4.10, we can determine how many SKFA components the model chooses and the penalty coefficient to reach optimality. Theory suggests that the number of sparse kernel factors (SKFs) can get the so-called "big factor," and the penalty factor can be any other non-negative number. However, the ICOMP value is exceptionally high in actual numerical experiments once the number of SKFA components exceeds fifty. In addition, the model has a substantial increase in complexity that makes it nearly impossible to interpret.

Figures 4.11, 4.12 and 4.13 illustrate the calculation of ICOMP values for SKFA and KFA with different numbers of components for three kernel functions (using the ovarian cancer data). For example, SKFA selects fewer factors than KFA when using a Gauss kernel. In contrast, SKFA and KFA select the same number of factors when using exponential and Laplace kernel functions.

Table 4.11 summarises the analysis results of the prostate, brain, colon, ovarian, and leukemia data. Our study indicated that the number of SKFs identified from SKFA was much smaller than the size/dimension of the data.

According to the previous introduction, the SKFA method presents significant advantages when dealing with high-dimensional data. Based on the minimum ICOMP values, our assessment is validated with SKFA over wide datasets to determine the sparse kernel factors. These factors are then fed into different classifiers to observe the outcome. Here we utilize three different classifiers: decision tree, naive Bayes, and kNN. Every classifier is cross-validated ten times to ensure the accuracy of the results. As shown in Tables 4.12, 4.13 and 4.14, the classification of the brain, leukemia, and prostate datasets has been improved partly due to the SKFA algorithm's introduction. For example, SKFA reduces the classification error from 40.2% to 17.65% for the Prostate data analysis.

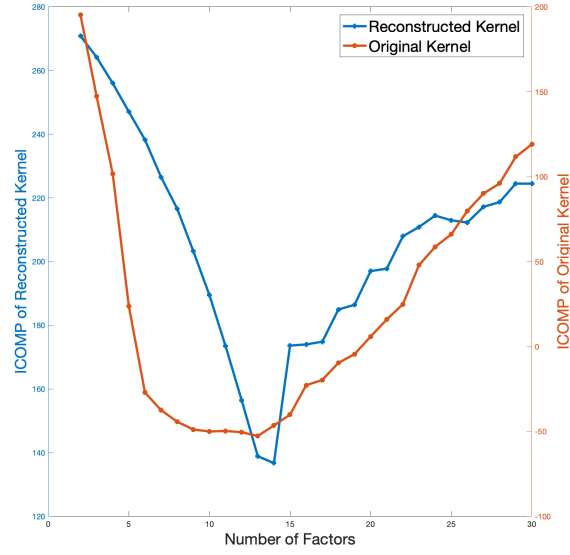


Figure 4.10: Comparison of ICOMP of original simulated kernel and reconstructed pre-image kernel under Gaussian kernel function.

Table 4.8: ICOMP values for Gauss Kernel SKFA (Ovarian Cancer).

		γ				
		0.05	0.10	0.22	0.46	1.00
# SKFA	12	91.21	90.81	91.08	91.94	94.51
	14	76.08	77.30	76.53	76.98	77.18
	16	73.11	69.36	66.49	75.29	73.14
	18	76.10	74.48	70.18	74.68	70.91
	20	79.63	80.49	72.71	81.35	82.17

Table 4.9: ICOMP values for Exponential Kernel SKFA (Ovarian Cancer).

		γ				
		0.10	0.22	0.46	1.00	2.15
# SKFA	4	477.56	477.60	477.64	477.91	477.86
	6	470.98	470.99	471.01	471.02	470.98
	8	467.88	467.88	467.83	467.83	467.95
	10	469.22	469.23	469.23	469.20	469.27
	12	472.72	472.64	472.63	472.63	472.74

Table 4.10: ICOMP values for Laplace Kernel SKFA (Ovarian Cancer).

		γ				
		0.22	0.46	1.00	2.15	4.64
# SKFA	6	117.58	114.87	123.06	114.76	117.08
	8	95.36	95.23	95.84	95.58	95.76
	10	87.34	86.98	86.61	87.37	87.75
	12	88.86	89.08	88.91	89.01	88.81
	14	95.28	95.00	94.75	94.96	95.24

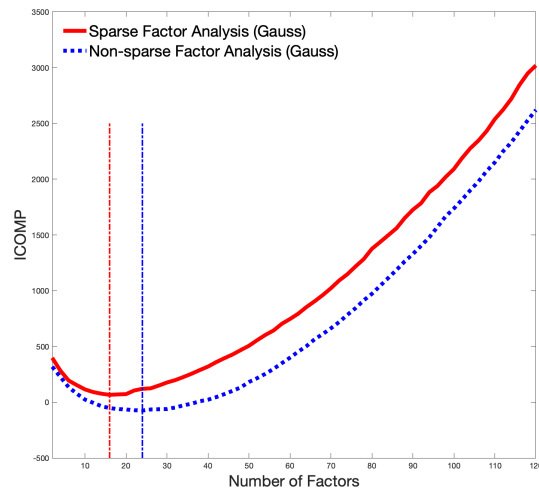


Figure 4.11: SKFA vs. KFA using Gauss Kernel (Ovarian Cancer).

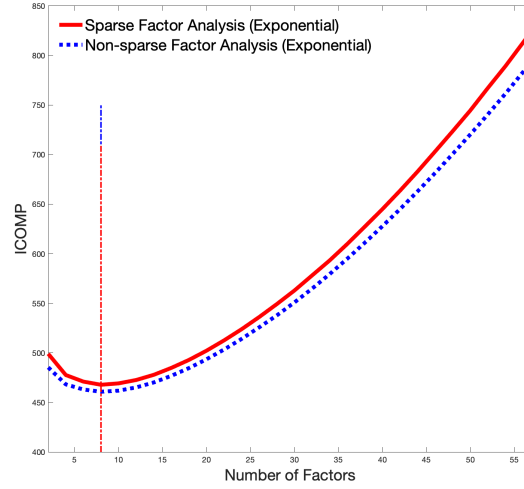


Figure 4.12: SKFA vs. KFA using Exponential Kernel (Ovarian Cancer).

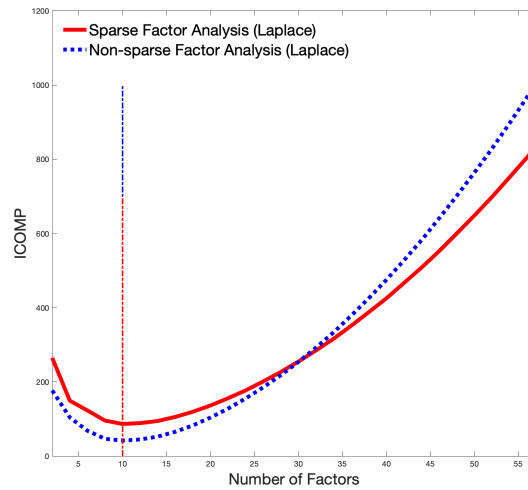


Figure 4.13: SKFA vs. KFA using Laplace Kernel (Ovarian Cancer).

Table 4.11: Wide datasets and SKFA results.

Datasets	n	p	g	n_g	Kernel Functions	Number of Factors	ICOMP
Prostate	102	6033	2	[50,52]	Gaussian	5	-56.70
					Exponential	5	24.06
					Laplace	5	61.49
Brain	42	5597	5	[10,10,10,4,8]	Gaussian	4	104.74
					Exponential	4	110.47
					Laplace	4	108.38
Colon	62	256	2	[22,40]	Gaussian	6	50.37
					Exponential	6	77.05
					Laplace	6	68.93
Leukemia	72	3571	2	[25,47]	Gaussian	9	141.47
					Exponential	8	175.79
					Laplace	8	172.14
Ovarian	216	4000	2	[121,95]	Gaussian	16	66.49
					Exponential	8	467.83
					Laplace	10	86.61

Table 4.12: Classification Error of Brain Data.

Classifier	Classification Error	
	Raw	SKFA(Gauss Kernel)
Naïve Bayes	0.3571	0.2381
Decision Tree	0.4048	0.3095
kNN	0.3333	0.2857

Table 4.13: Classification Error of Colon Data.

Classifier	Classification Error	
	Raw	SKFA(Gauss Kernel)
Naïve Bayes	0.4032	0.3871
Decision Tree	0.3871	0.4839
kNN	0.4839	0.4194

4.6 Conclusion

In this paper, we introduced the SKFA model that combines the kernel method with sparse factor analysis to effectively cluster and classify data by combining the nonlinear properties with the idea of dimensionality reduction.

We presented the SKFA parameter estimation method. Modified EM algorithm is capable of keeping factor loading sparse. Furthermore, the SVD makes the computation of the likelihood of the model more stable.

Introducing the information complexity criterion (ICOMP), we systematically explored how to search for hyperparameters and select models. We used an integrated parameter estimation and hyperparameter search approach (i.e., grid search) to construct a complete data analysis pipeline based on the SKFA model.

We validated the stability of the SKFA model computational results by applying pre-image learning to numerical experiments. For real-world datasets, we used a variety of wide datasets to demonstrate the value of SKFA in the nonlinear representation and classification of data.

We will further explore the nonlinear characteristics of the kernel and expand its unique advantages in data analysis in the future. Specifically, traditional kernel analysis methods cannot cope with considerable amounts of data (due to the large kernel matrix). Therefore, we will explore further model optimizations to support big data.

Table 4.14: Classification Error of Leukemia Data.

Classifier	Classification Error	
	Raw	SKFA(Gauss Kernel)
Naïve Bayes	0.0139	0.0417
Decision Tree	0.1667	0.1667
kNN	0.0278	0.0556

Table 4.15: Classification Error of Prostate Data.

Classifier	Classification Error	
	Raw	SKFA(Gauss Kernel)
Naïve Bayes	0.402	0.1765
Decision Tree	0.1471	0.2843
kNN	0.2157	0.1765

Chapter 5

Conclusion and Future Research

The dissertation presents three case studies illustrating the use of information criteria (particularly the information complexity criterion) for statistical model selection. Furthermore, our specific discussion covers modeling selection with a sparsity setting. It shows that sparsity can stabilize the parameter estimation of high-dimensional data models and improve their interpretability. Next, we will explore further how to extend sparsity, model selection, and ICOMP in the future.

First, we explore how to extend the application of sparsity. The present study focuses on achieving sparsity effects in classical statistical models by selecting predictors. Research in the future will extend the concept of sparsity. We want to select from a group of candidate models or select from a set of data points to achieve sparsity. When building sparse solutions in the model dimension, it is necessary to ensure that more than one candidate model is feasible. Combining different models through boosting [114] and stacking-based ensemble methods [159] builds a stronger learner in classical statistics. However, it is worth noting that current ensemble learning is still heavily dominated by dense models, and there is a lack of research on the sparsification of ensemble learning. Consequently, we will endeavor to fill the research gap in this area in the future.

In the field of deep learning, we have witnessed researchers applying sparsity to Mixture of Experts (MoE) models. These studies have demonstrated the significant improvement that sparsity brings to existing models (probabilistic and non-probabilistic models). Jayakumar et al. point out that sparsity's introduction can significantly improve computational efficiency

and perform better than dense models [81]. Despite maintaining a constant computation cost, sparsity permits deep learning models to scale up to a staggering 10 and 100 billion parameters, with a faster convergence rate [152]. As a well-studied field in statistics, sparsity will provide potential theoretical value for designing these neural networks. Our future research will take this direction as well.

It is also promising to apply sparsity to data. According to [136], in the current state of so-called big data, the vast majority of the dataset is “unforgettable”. Hence, a model with few training steps can achieve successful classification and prediction capabilities for such data sets. In contrast, very few data sets contain complex information or contain much noise. These data sets are known as “forgettable” data. It requires a long training period for the learner to classify and predict forgettable data. Some partial solutions have been proposed in the literature, such as setting up proxies [89]. However, these approaches have not been rigorously analyzed, and they are based on empirical evidence. Combining the proxy approach with sparsity-related algorithms will allow the model to focus more on forgettable data, thus increasing the model’s training efficiency.

In addition, we intend to extend the application of information criteria to deep learning. Currently, there are few information criteria applicable to deep learning models. Following the traditional information criterion, the number of model parameters should not be excessive. However, when the model is complex enough to have billions of parameters, constructing the covariance matrix or the Fisher information matrix will be so challenging that it will be impossible to gauge how complex the model is. As discussed above, this problem can be potentially solved by introducing proxy learning. Our future research will investigate this hypothesis. If successfully applied to the design of deep network models, this criterion will provide excellent statistical interpretation and further improve the interpretability of deep learning models.

Finally, we intend to extend the application of information criteria to deep learning. Currently, there are few information criteria applicable to deep learning models. Following the traditional information criterion, the number of model parameters should not be excessive. However, when the model is complex enough to have billions of parameters, constructing the covariance matrix or the Fisher information matrix will be so challenging

that it will be impossible to gauge how complex the model is. As discussed above, this problem can be potentially solved by introducing proxy learning. Our future research will investigate this hypothesis. If successfully applied to the design of deep network models, this criterion will provide excellent statistical interpretation and further improve the interpretability of deep learning models.

Bibliography

- [1] Abonyi, J. and Feil, B. (2007). *Cluster analysis for data mining and system identification*. Springer Science & Business Media. [37](#)
- [2] Addesso, P., Filatrella, G., and Pierro, V. (2012). Characterization of escape times of josephson junctions for signal detection. *Physical Review E*, 85(1):016708. [71](#)
- [3] Aït-Sahalia, Y. and Xiu, D. (2019). Principal component analysis of high-frequency data. *Journal of the American Statistical Association*, 114(525):287–303. [70](#)
- [4] Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In Petrox, B. and Csaki, F., editors, *Second International Symposium on Information Theory.*, pages 267–281, Budapest. Akademiai Kiado. [6](#), [49](#)
- [5] Akaike, H. (1974). A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723. [89](#)
- [6] Akaike, H. (1987). Factor analysis and aic. In *Selected papers of hirotugu akaike*, pages 371–386. Springer. [89](#)
- [7] Anderson, J. C. and Gerbing, D. W. (1984). The effect of sampling error on convergence, improper solutions, and goodness-of-fit indices for maximum likelihood confirmatory factor analysis. *Psychometrika*, 49(2):155–173. [80](#)
- [8] Apsemidis, A., Psarakis, S., and Moguerza, J. M. (2020). A review of machine learning kernel methods in statistical process monitoring. *Computers & Industrial Engineering*, 142:106376. [73](#)
- [9] Azpeitia, E. and Wagner, A. (2019). The positive role of noise for information acquisition in biological signaling pathways. *bioRxiv*, page 762989. [71](#)
- [10] Bakır, G. H., Weston, J., and Schölkopf, B. (2004). Learning to find pre-images. *Advances in neural information processing systems*, 16:449–456. [99](#)
- [11] Barber, D. (2012). *Bayesian reasoning and machine learning*. Cambridge University Press. [81](#)

- [12] Barber, D., Cemgil, A. T., and Chiappa, S. (2011). *Bayesian time series models*. Cambridge University Press. [37](#)
- [13] Bartholomew, D. J., Knott, M., and Moustaki, I. (2011). *Latent variable models and factor analysis: A unified approach*, volume 904. John Wiley & Sons. [73](#)
- [14] Bendel, R. B. and Afifi, A. A. (1977). Comparison of stopping rules in forward “stepwise” regression. *Journal of the American Statistical association*, 72(357):46–53. [2](#)
- [15] Bozdogan, H. (1987). Model selection and akaike’s information criteria (aic): the general theory and its analytical extensions. *Psychometrika*, 52:317–332. [54](#)
- [16] Bozdogan, H. (1988). Icomp: A new model selection criterion. In Bock, H. H., editor, *Classification and Related Methods of Data Analysis*, pages 599–608. North-Holland, Amsterdam. [7](#)
- [17] Bozdogan, H. (1990). On the information-based measure of covariance complexity and its application to the evaluation of multivariate linear models. *Communications in Statistics-Theory and Methods*, 19(1):221–278. [7](#), [40](#), [50](#), [89](#)
- [18] Bozdogan, H. (1993). Choosing the number of component clusters in the mixture-model using a new informational complexity criterion of the inverse-Fisher information matrix. In *Information and classification*, pages 40–54. Springer. [40](#), [50](#), [89](#)
- [19] Bozdogan, H. (1994a). Choosing the number of clusters, subset selection of variables, and outlier detection in the standard mixture-model cluster analysis. In *New approaches in classification and data analysis*, pages 169–177. Springer. [40](#), [50](#), [89](#)
- [20] Bozdogan, H. (1994b). Mixture-model cluster analysis using model selection criteria and a new informational measure of complexity. In *Proceedings of the first US/Japan conference on the frontiers of statistical modeling: An informational approach*, pages 69–113. Springer. [40](#), [50](#), [89](#)
- [21] Bozdogan, H. (2000). Akaike’s Information Criterion and Recent Developments in Information Complexity. *Journal of Mathematical Psychology*, 44:62–91. [7](#), [10](#), [16](#), [17](#), [50](#), [89](#)

- [22] Bozdogan, H. (2003). Intelligent statistical data mining with information complexity and genetic algorithms hamparsum bozdogan university of tennessee, knoxville, usa. In *Statistical data mining and knowledge discovery*, pages 47–88. Chapman and Hall/CRC. [5](#)
- [23] Bozdogan, H., editor (2004). *Statistical Data Mining and Knowledge Discovery*. Chapman and Hall/CRC. [9](#), [13](#), [16](#), [17](#)
- [24] Bozdogan, H. (2010a). A new class of information complexity (ICOMP) criteria with an application to customer profiling and segmentation. *İstanbul Üniversitesi İşletme Fakültesi Dergisi*, 39(2):370–398. [40](#), [50](#), [89](#)
- [25] Bozdogan, H. (2010b). A new class of information complexity (icomp) criteria with an application to customer profiling and segmentation. *Istanbul University Journal of the School of Business Administration*, 39(2):370–398. [89](#)
- [26] Bozdogan, H. (2020). *Information Complexity and Multivariate Modeling in High Dimensions with Applications*. Forthcoming book. [18](#)
- [27] Bozdogan, H. and Howe, J. A. (2012). Misspecified multivariate regression models using the genetic algorithm and information complexity as the fitness function. *European Journal of Pure and Applied Mathematics*, 5(2):211–249. [10](#)
- [28] Bozdogan, H. and Pamukcu, E. (2016). Novel dimension reduction techniques for high dimensional data using information complexity. In Gupta, A. and Capponi, A., editors, *Optimization Challenges in Complex, Networked, and Risky Systems*, pages 140–170. INFORMS. [9](#), [20](#)
- [29] Bozdogan, H. and Ramirez, D. E. (1987). An expert model selection approach to determine the “best” pattern structure in factor analysis models. In *Multivariate statistical modeling and data analysis*, pages 35–60. Springer. [88](#)
- [30] Breaux, H. J. (1967). On stepwise multiple linear regression. Technical report, Army Ballistic Research Lab Aberdeen Proving Ground MD. [5](#)

- [31] Briggs, S. R. and Cheek, J. M. (1986). The role of factor analysis in the development and evaluation of personality scales. *Journal of personality*, 54(1):106–148. [76](#)
- [32] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901. [1](#)
- [33] Brown, T. A. (2015). *Confirmatory factor analysis for applied research*. Guilford publications. [73](#)
- [34] Bühlmann, P., Kalisch, M., and Meier, L. (2014). High-dimensional statistics with a view toward applications in biology. *Annual Review of Statistics and Its Application*, 1:255–278. [30](#)
- [35] Cavanaugh, J. E. and Neath, A. A. (2019). The akaike information criterion: Background, derivation, properties, application, interpretation, and refinements. *Wiley Interdisciplinary Reviews: Computational Statistics*, 11(3):e1460. [6](#)
- [36] Charles, D. and Fyfe, C. (2000). Kernel factor analysis with varimax rotation. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, volume 3, pages 381–386. IEEE. [73](#)
- [37] Chen, J. and Chen, Z. (2008). Extended bayesian information criteria for model selection with large model spaces. *Biometrika*, 95(3):759–771. [6](#)
- [38] Chen, J. and Chen, Z. (2012). Extended bic for small-n-large-p sparse glm. *Statistica Sinica*, pages 555–574. [7](#)
- [39] Chen, Y., Wiesel, A., Eldar, Y. C., and Hero, A. O. (2010). Shrinkage algorithms for mmse covariance estimation. *IEEE Transactions on Signal Processing*, 58(10):5016–5029. [20](#), [22](#)
- [40] Cho, Y. and Saul, L. (2009). Kernel methods for deep learning. *Advances in neural information processing systems*, 22. [73](#)

- [41] Choi, J., Oehlert, G., and Zou, H. (2010). A penalized maximum likelihood approach to sparse factor analysis. *Statistics and its Interface*, 3(4):429–436. [85](#)
- [42] Clerc, M. and Kennedy, J. (2002). The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE transactions on Evolutionary Computation*, 6(1):58–73. [20](#)
- [43] Cox, D. R. and Wermuth, N. (2014). *Multivariate dependencies: Models, analysis and interpretation*. Chapman and Hall/CRC. [1](#)
- [44] Cudeck, R. and MacCallum, R. C. (2007). *Factor analysis at 100: Historical developments and future directions*. Routledge. [73](#), [76](#)
- [45] De Veaux, R. D. (1986). Parameter estimation for a mixture of linear regressions. *Doctoral Dissertation and Tech. Rept. No. 247, Department of Statistics, Stanford University*. [38](#)
- [46] Demirkaya, E., Feng, Y., Basu, P., and Lv, J. (2022). Large-scale model selection in misspecified generalized linear models. *Biometrika*, 109(1):123–136. [23](#)
- [47] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22. [84](#)
- [48] DeSarbo, W. S. and Cron, W. L. (1988). A maximum likelihood methodology for clusterwise linear regression. *Journal of classification*, 5(2):249–282. [38](#)
- [49] DeSarbo, W. S., Wedel, M., et al. (1993). A review of recent developments in latent class regression models. Technical report. [38](#)
- [50] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. [4](#)
- [51] Dillon, W. R., Böckenholt, U., De Borrero, M. S., Bozdogan, H., De Sarbo, W., Gupta, S., Kamakura, W., Kumar, A., Ramaswamy, B., and Zenor, M. (1994). Issues

- in the estimation and application of latent structure models of choice. *Marketing Letters*, 5(4):323–334. [38](#)
- [52] Ding, W., Wang, B., Xing, Y., and Li, J.-C. (2019). Correlation noise and delay time enhanced stability of electricity futures market. *Modern Physics Letters B*, 33(30):1950375. [37](#)
- [53] Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics*, 32(2):407–499. [14](#)
- [54] Esling, P. and Agon, C. (2012). Time-series data mining. *ACM Computing Surveys (CSUR)*, 45(1):1–34. [46](#)
- [55] Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360. [15](#), [22](#), [26](#), [28](#)
- [56] Fan, J. and Lv, J. (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(5):849–911. [5](#)
- [57] Fan, J. and Lv, J. (2010). A selective overview of variable selection in high dimensional feature space. *Statistica Sinica*, 20(1):101. [4](#), [5](#)
- [58] Fan, J., Lv, J., and Qi, L. (2011). Sparse high-dimensional models in economics. *Annu. Rev. Econ.*, 3(1):291–317. [4](#), [15](#)
- [59] Foygel, R. and Drton, M. (2010). Extended bayesian information criteria for gaussian graphical models. *Advances in neural information processing systems*, 23. [7](#)
- [60] Frank, L. E. and Friedman, J. H. (1993). A statistical view of some chemometrics regression tools. *Technometrics*, 35(2):109–135. [14](#), [15](#)
- [61] Fu, W. J. (1998). Penalized regressions: the bridge versus the lasso. *Journal of computational and graphical statistics*, 7(3):397–416. [15](#)

- [62] Fukumizu, K., Bach, F. R., and Gretton, A. (2007). Statistical consistency of kernel canonical correlation analysis. *Journal of Machine Learning Research*, 8(2). [79](#)
- [63] Gaffney, S. (2004). *Probabilistic curve-aligned clustering and prediction with regression mixture models*. PhD thesis, University of California, Irvine. [46](#)
- [64] Gaffney, S. and Smyth, P. (1999). Trajectory clustering with mixtures of regression models. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 63–72. [46](#)
- [65] Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (1995). *Bayesian data analysis*. Chapman and Hall/CRC. [1](#)
- [66] Godfrey, L. G. (1991). *Misspecification tests in econometrics: the Lagrange multiplier principle and other approaches*. Number 16. Cambridge University Press. [13](#)
- [67] Guan, Y. and Dy, J. (2009). Sparse probabilistic principal component analysis. In *Artificial Intelligence and Statistics*, pages 185–192. [43](#), [78](#)
- [68] Guarcello, C., Valenti, D., Spagnolo, B., Pierro, V., and Filatrella, G. (2019). Josephson-based threshold detector for lévy-distributed current fluctuations. *Physical Review Applied*, 11(4):044078. [45](#)
- [69] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778. [1](#)
- [70] Hirose, K., Kawano, S., Konishi, S., and Ichikawa, M. (2011). Bayesian information criterion and selection of the number of factors in factor analysis models. *Journal of Data Science*, 9(2):243–259. [89](#)
- [71] Hoerl, A. E. and Kennard, R. W. (1970a). Ridge regression: applications to nonorthogonal problems. *Technometrics*, 12(1):69–82. [15](#)
- [72] Hoerl, A. E. and Kennard, R. W. (1970b). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67. [15](#)

- [73] Honeine, P. and Richard, C. (2011). Preimage problem in kernel-based machine learning. *IEEE Signal Processing Magazine*, 28(2):77–88. [99](#)
- [74] Hooten, M. B. and Hobbs, N. T. (2015). A guide to bayesian model selection for ecologists. *Ecological monographs*, 85(1):3–28. [3](#)
- [75] Huang, J., Horowitz, J. L., and Ma, S. (2008). Asymptotic properties of bridge estimators in sparse high-dimensional regression models. *The Annals of Statistics*, 36(2):587–613. [15](#)
- [76] Huang, J., Jiao, Y., Liu, Y., and Lu, X. (2018). A constructive approach to l0 penalized regression. *The Journal of Machine Learning Research*, 19(1):403–439. [14](#)
- [77] Hubbard, R. and Allen, S. J. (1989). On the number and nature of common factors extracted by the eigenvalue-one rule using bmdp vs spssx. *Psychological Reports*, 65(1):155–160. [89](#)
- [78] Huber, P. J. (1967). The behavior of maximum likelihood estimates under nonstandard conditions. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 221–233. University of California Press. [13](#), [17](#)
- [79] Hyvärinen, A. and Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430. [79](#)
- [80] Ince, H. and Trafalis, T. B. (2007). Kernel principal component analysis and support vector machines for stock price prediction. *Iie Transactions*, 39(6):629–637. [73](#)
- [81] Jayakumar, S., Pascanu, R., Rae, J., Osindero, S., and Elsen, E. (2020). Top-kast: Top-k always sparse training. *Advances in Neural Information Processing Systems*, 33:20744–20754. [111](#)
- [82] Jones, C. S. (2006). A nonlinear factor analysis of s&p 500 index option returns. *The Journal of Finance*, 61(5):2325–2363. [76](#)
- [83] Jöreskog, K. G. (1967). Some contributions to maximum likelihood factor analysis. *Psychometrika*, 32(4):443–482. [81](#)

- [84] Kauermann, G. and Carroll, R. J. (2001). A note on the efficiency of sandwich covariance matrix estimation. *Journal of the American Statistical Association*, 96(456):1387–1396. [13](#)
- [85] Kawano, S. (2012). Adaptive bridge regression modeling and selection of the tuning parameters. [9](#), [23](#)
- [86] Kawano, S. (2014). Selection of tuning parameters in bridge regression models via bayesian information criterion. *Statistical Papers*, 55(4):1207–1223. [9](#), [16](#), [22](#), [23](#)
- [87] Ke, T., Jin, J., and Fan, J. (2014). Covariance assisted screening and estimation. *Annals of statistics*, 42(6):2202. [5](#)
- [88] Kiefer, N. M. (1978). Discrete parameter variation: Efficient estimation of a switching regression model. *Econometrica: Journal of the Econometric Society*, pages 427–434. [38](#)
- [89] Kim, S., Kim, D., Cho, M., and Kwak, S. (2020). Proxy anchor loss for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3238–3247. [111](#)
- [90] Knight, K. and Fu, W. (2000). Asymptotics for lasso-type estimators. *The Annals of statistics*, 28(5):1356–1378. [15](#)
- [91] Koc, E. K. and Bozdogan, H. (2015). Model selection in multivariate adaptive regression splines (mars) using information complexity as the fitness function. *Machine Learning*, 101(1-3):35–58. [17](#)
- [92] Konishi, S., Ando, T., and Imoto, S. (2004). Bayesian information criteria and smoothing parameter selection in radial basis function networks. *Biometrika*, 91(1):27–43. [16](#)
- [93] Kullback, A. and Leibler, R. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86. [49](#)
- [94] Ledoit, O. and Wolf, M. (2004). A well-conditioned estimator for large-dimensional covariance matrices. *Journal of multivariate analysis*, 88(2):365–411. [20](#)

- [95] Li, J.-h. and Łuczka, J. (2010). Thermal-inertial ratchet effects: Negative mobility, resonant activation, noise-enhanced stability, and noise-weakened stability. *Physical Review E*, 82(4):041104. [71](#)
- [96] Ludvigson, S. C. and Ng, S. (2007). The empirical risk–return relation: A factor analysis approach. *Journal of Financial Economics*, 83(1):171–222. [76](#)
- [97] Ma, R., Miao, J., Niu, L., and Zhang, P. (2019). Transformed l1 regularization for learning sparse deep neural networks. *Neural Networks*, 119:286–298. [3](#)
- [98] Magnus, J. R. and Neudecker, H. (2019). *Matrix differential calculus with applications in statistics and econometrics*. John Wiley & Sons. [80](#), [81](#)
- [99] Mairal, J., Bach, F., Ponce, J., and Sapiro, G. (2009). Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*, pages 689–696. [43](#)
- [100] Marcus, G. and Davis, E. (2020). Gpt-3, bloviator: Openai’s language generator has no idea what it’s talking about. *Technology Review*. [1](#)
- [101] McCullagh, P. (2002). What is a statistical model? *The Annals of Statistics*, 30(5):1225–1310. [1](#)
- [102] McCullagh, P. and Nelder, J. A. (2019). *Generalized linear models*. Routledge. [1](#)
- [103] Mohebbi, S., Pamukcu, E., and Bozdogan, H. (2019). A new data adaptive elastic net predictive model using hybridized smoothed covariance estimators with information complexity. *Journal of Statistical Computation and Simulation*, pages 1–30. [9](#), [10](#), [20](#), [24](#), [30](#), [32](#)
- [104] Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press. [74](#), [88](#)
- [105] Natarajan, B. K. (1995). Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234. [14](#)

- [106] Ollila, E. (2017). Optimal high-dimensional shrinkage covariance estimation for elliptical distributions. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 1639–1643. IEEE. [20](#)
- [107] Preacher, K. J., Zhang, G., Kim, C., and Mels, G. (2013). Choosing the optimal number of factors in exploratory factor analysis: A model selection perspective. *Multivariate Behavioral Research*, 48(1):28–56. [89](#)
- [108] Quandt, R. E. and Ramsey, J. (1978). Estimating Mixtures of Normal Distributions and Switching Regressions. *Journal of American Statistical Association*, 73. [38](#)
- [109] Rakotomamonjy, A. (2003). Variable selection using svm-based criteria. *Journal of machine learning research*, 3(Mar):1357–1370. [2](#)
- [110] Rojo-Álvarez, J. L., Martínez-Ramón, M., Muñoz-Mari, J., and Camps-Valls, G. (2018). *Digital signal processing with Kernel methods*. John Wiley & Sons. [73](#)
- [111] Roozbeh, M. (2018). Generalized ridge regression estimator in high dimensional sparse regression models. *Statistics, Optimization & Information Computing*, 6(3):415–426. [24](#)
- [112] S Lima, L. (2019). Nonlinear stochastic equation within an itô prescription for modelling of financial market. *Entropy*, 21(5):530. [37](#)
- [113] Schaid, D. J. (2010). Genomic similarity and kernel methods i: advancements by building on mathematical and statistical foundations. *Human heredity*, 70(2):109–131. [73](#)
- [114] Schapire, R. E. (2003). The boosting approach to machine learning: An overview. *Nonlinear estimation and classification*, pages 149–171. [110](#)
- [115] Schölkopf, B., Smola, A., and Müller, K.-R. (1997). Kernel principal component analysis. In *International conference on artificial neural networks*, pages 583–588. Springer. [79](#)
- [116] Scholkopf, B., Sung, K.-K., Burges, C. J., Girosi, F., Niyogi, P., Poggio, T., and Vapnik, V. (1997). Comparing support vector machines with gaussian kernels to radial basis function classifiers. *IEEE transactions on Signal Processing*, 45(11):2758–2765. [73](#)

- [117] Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6:461–464. [6](#)
- [118] Sclove, S. L. (1983). Time-series segmentation: A model and a method. *Information Sciences*, 29(1):7–25. [38](#)
- [119] Serdukova, L., Zheng, Y., Duan, J., and Kurths, J. (2016). Stochastic basins of attraction for metastable states. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 26(7):073117. [71](#)
- [120] Shen, H. and Huang, J. Z. (2008). Sparse principal component analysis via regularized low rank matrix approximation. *Journal of multivariate analysis*, 99(6):1015–1034. [42](#)
- [121] Shibata, R. (1980). Asymptotically efficient selection of the order of the model for estimating parameters of a linear process. *The annals of statistics*, pages 147–164. [6](#)
- [122] Soman, K., Loganathan, R., and Ajay, V. (2009). *Machine learning with SVM and other kernel methods*. PHI Learning Pvt. Ltd. [73](#)
- [123] Souza, C. R. (2010). Kernel functions for machine learning applications. *Creative commons attribution-noncommercial-share alike*, 3(29):1–1. [92](#)
- [124] Spagnolo, B., Guarcello, C., Magazzù, L., Carollo, A., Persano Adorno, D., and Valenti, D. (2017). Nonlinear relaxation phenomena in metastable condensed matter systems. *Entropy*, 19(1):20. [71](#)
- [125] Spagnolo, B. and Valenti, D. (2008). Volatility effects on the escape time in financial market models. *International Journal of Bifurcation and Chaos*, 18(09):2775–2786. [37](#)
- [126] Spagnolo, B., Valenti, D., Guarcello, C., Carollo, A., Adorno, D. P., Spezia, S., Pizzolato, N., and Di Paola, B. (2015). Noise-induced effects in nonlinear relaxation of condensed matter systems. *Chaos, Solitons & Fractals*, 81:412–424. [71](#)
- [127] Spiegelhalter, D. J., Best, N. G., Carlin, B. P., and Van der Linde, A. (2014). The deviance information criterion: 12 years on. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(3):485–493. [6](#)

- [128] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958. [100](#)
- [129] Sun, Y. and Bozdogan, H. (2020). Segmentation of high dimensional time-series data using mixture of sparse principal component regression model with information complexity. *Entropy*, 22(10):1170. [98](#)
- [130] Takeuchi, K. (1976). Distribution of information statistics and a criterion of model fitting. *Suri-Kagaku, Mathematical Sciences*, 153:12–18. [23](#)
- [131] Tännler, S., Zamboni, N., Kiraly, C., Aymerich, S., and Sauer, U. (2008). Screening of bacillus subtilis transposon mutants with altered riboflavin production. *Metabolic Engineering*, 10(5):216–226. [30](#)
- [132] Tenney, I., Das, D., and Pavlick, E. (2019). Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*. [5](#)
- [133] Theodoridis, S. (2015). *Machine learning: a Bayesian and optimization perspective*. Academic press. [89](#)
- [134] Tibshirani, R. (1996). Regression shrinkage and selection via lasso. *Journal of the Royal Statistical Society Series B*, 58:267–288. [2](#), [14](#)
- [135] Tipping, M. (2000). Sparse kernel principal component analysis. *Advances in neural information processing systems*, 13. [73](#)
- [136] Toneva, M., Sordoni, A., Combes, R. T. d., Trischler, A., Bengio, Y., and Gordon, G. J. (2018). An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*. [111](#)
- [137] Valenti, D., Fazio, G., and Spagnolo, B. (2018). Stabilizing effect of volatility in financial markets. *Physical Review E*, 97(6):062307. [37](#)
- [138] Valenti, D., Magazzù, L., Caldara, P., and Spagnolo, B. (2015). Stabilization of quantum metastable states by dissipation. *Physical Review B*, 91(23):235412. [71](#)

- [139] Van Aken, B., Winter, B., Löser, A., and Gers, F. A. (2019). How does bert answer questions? a layer-wise analysis of transformer representations. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1823–1832. [5](#)
- [140] Van Emden, H. M. (1971). An analysis of complexity. In *Mathematical Centre Tracts.*, volume 35. Mathematisch Centrum. [50](#)
- [141] Van Laarhoven, T. (2017). L2 regularization versus batch and weight normalization. *arXiv preprint arXiv:1706.05350*. [3](#)
- [142] Walt, S. v. d., Colbert, S. C., and Varoquaux, G. (2011). The numpy array: a structure for efficient numerical computation. *Computing in science & engineering*, 13(2):22–30. [54](#)
- [143] Wang, D., Cui, P., and Zhu, W. (2016). Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1225–1234. [75](#)
- [144] Wang, X., He, X., Feng, F., Nie, L., and Chua, T.-S. (2018). Tem: Tree-enhanced embedding model for explainable recommendation. In *Proceedings of the 2018 World Wide Web Conference*, pages 1543–1552. [75](#)
- [145] Wang, X.-H. and Li, J.-J. (2004). Hybrid particle swarm optimization with simulated annealing. In *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826)*, volume 4, pages 2402–2405. IEEE. [19](#)
- [146] Weston, J., Chapelle, O., Elisseeff, A., Schölkopf, B., and Vapnik, V. (2002). Kernel dependency estimation. [99](#), [100](#)
- [147] White, H. (1982). Maximum likelihood estimation of misspecified models. *Econometrica*, pages 1–25. [13](#), [17](#)
- [148] White, H. (1994). *Estimation, Inference and Specification Analysis*. Number 22. Cambridge University Press. [13](#)

- [149] Williams, P. M. (1995). Bayesian regularization and pruning using a Laplace prior. *Neural computation*, 7(1):117–143. [43](#)
- [150] Wu, C., Zhou, F., Ren, J., Li, X., Jiang, Y., and Ma, S. (2019). A selective review of multi-level omics data integration using variable selection. *High-throughput*, 8(1):4. [4](#)
- [151] Xia, G.-e. and Shao, P.-j. (2009). Factor analysis algorithm with mercer kernel. In *2009 Second International Symposium on Intelligent Information Technology and Security Informatics*, pages 202–205. IEEE. [73](#)
- [152] Yang, A., Lin, J., Men, R., Zhou, C., Jiang, L., Jia, X., Wang, A., Zhang, J., Wang, J., Li, Y., et al. (2021). M6-t: Exploring sparse expert models and beyond. *arXiv preprint arXiv:2105.15082*. [111](#)
- [153] Yang, H. and Bozdogan, H. (2011). Learning factor patterns in exploratory factor analysis using the genetic algorithm and information complexity as the fitness function. *J. Pattern Recognit. Res*, 6(2):307–325. [80](#), [89](#)
- [154] Yang, M.-S., Lai, C.-Y., and Lin, C.-Y. (2012). A robust EM clustering algorithm for Gaussian mixture models. *Pattern Recognition*, 45(11):3950–3961. [39](#), [45](#), [46](#), [48](#)
- [155] Yates, A. (1987). *Multivariate exploratory data analysis: A perspective on exploratory factor analysis*. Suny Press. [80](#)
- [156] Zhan, Z.-H., Zhang, J., Li, Y., and Chung, H. S.-H. (2009). Adaptive particle swarm optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(6):1362–1381. [19](#)
- [157] Zhang, C.-X., Zhang, J.-S., and Yin, Q.-Y. (2017). A ranking-based strategy to prune variable selection ensembles. *Knowledge-Based Systems*, 125:13–25. [2](#)
- [158] Zhao, J.-H., Philip, L., and Jiang, Q. (2008). Ml estimation for factor analysis: Em or non-em? *Statistics and computing*, 18(2):109–123. [80](#), [81](#), [84](#)
- [159] Zhou, Z.-H. (2021). Ensemble learning. In *Machine learning*, pages 181–210. Springer. [110](#)

- [160] Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320. [15](#)
- [161] Zou, H., Hastie, T., and Tibshirani, R. (2006). Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286. [39](#), [42](#), [43](#)
- [162] Zwick, W. R. and Velicer, W. F. (1982). Factors influencing four rules for determining the number of components to retain. *Multivariate behavioral research*, 17(2):253–269. [89](#)
- [163] Zwick, W. R. and Velicer, W. F. (1986). Comparison of five rules for determining the number of components to retain. *Psychological bulletin*, 99(3):432. [89](#)

Appendix

A Acronyms

Table 1: Abbreviations.

Acronym	Description
FA	Factor Analysis
ICA	Independent Component Analysis
PCA	Principal Component Analysis
PPCA	Probabilistic Principal Component Analysis
SPPCA	Sparse Probabilistic Principal Component Analysis
PC	Principal Component
SPC	Sparse Principal Component
SPCA	Sparse Principal Component Analysis
SPCR	Sparse Principal Component Regression
MIX-SPCR	Mixture of Sparse Principal Component Regression
KFA	Kernel Factor Analysis
SKFA	Sparse Kernel Factor Analysis
KPCA	Kernel Principal Component Analysis
EM	Expectation-Maximization algorithm
GA	Genetic Algorithm
SA-PSO	Simulated Annealing and Particle Swarm Optimization
IC	Information Criterion
ICOMP	Information Complexity Criterion
IFIM	Inverse Fisher Information Matrix
MISS	Misspecification
SVD	Singular Value Decomposition
LASSO	Least Absolute Shrinkage and Selection Operator Model

B SA-PSO Algorithm

Algorithm 2 SA-PSO Initialization .

- 1: Input: N : size of population; M : number of generations; $\text{ICOMP}_{\text{Miss}}$: fitness function;
 X, Y , the dataset
 - 2: Initialize values of $v_0(\lambda)$, $v_0(q)$ and T .
 - 3: $\Lambda_0 = \{\lambda_{i,0} | \lambda_{i,0} \in (0, 4), i = 1, 2, \dots, N\}$
 - 4: $Q_0 = \{q_{i,0} | q_{i,0} \in (0, 1), i = 1, 2, \dots, N\}$
 - 5: $F_0 = \{f_{i,0} | f_{i,0} = \text{ICOMP}_{\text{Miss}}(\lambda_{i,0}, q_{i,0}, X, Y), i = 1, 2, \dots, N\}$
 - 6: $(\lambda^*, q^*) = \text{argmin}_{\Lambda_0, Q_0} \text{ICOMP}_{\text{Miss}}(X, Y)$
 - 7: $f^* = \text{ICOMP}_{\text{Miss}}(\lambda^*, q^*, X, Y)$
 - 8: $Y = (\Lambda_0, Q_0)$ $\triangleright Y_i(\lambda)$ means λ value of i^{th} particle.
 - 9: $P = F_0$ $\triangleright P_i$ records $\text{ICOMP}_{\text{Miss}}$ value of the i^{th} particle.
-

Algorithm 3 SA-PSO Main.

- 1: After initializing the parameters using Algorithm SA-PSO (Initialization).
 - 2: **for** $t = 1 : M$ **do**
 - 3: $K = \{k_i | k_i = \frac{\text{Exp}(-\frac{f_{i,t-1}-f^*}{T})}{\sum_{j=1}^N \text{Exp}(-\frac{f_{j,t-1}-f^*}{T})}\}$
 - 4: $threshold = \text{rand}()$ $\triangleright \text{rand}() \in [0, 1]$ is a (pseudo)random number generator.
 - 5: **for** $e = 1 : N$ **do**
 - 6: **if** $threshold \leq \text{sum}(K(1 : e))$ **then** $\triangleright \text{sum}(K(1 : e))$ represents summation from
the first value to e^{th} value .
 - 7: $\lambda'_i = \lambda_{e,t-1}$
 - 8: $q'_i = q_{e,t-1}$
 - 9: break
 - 10: **end if**
 - 11: Run algorithm SA-PSO (Update)
 - 12: **end for**
 - 13: **end for**
-

Algorithm 4 SA-PSO Update.

```
1:  $C = c_1 + c_2$ 
2:  $\chi = \frac{2}{|2-C-\sqrt{C^2-4C}|}$ 
3: for  $i = 1 : N$  do
4:    $r_1 = rand(), r_2 = rand()$ 
5:    $v_{i,t}(\lambda) = \chi\{v_i(\lambda) + c_1r_1(Y_i(\lambda) - \lambda_{i,t-1})\} + c_2r_2(\lambda'_i - \lambda_{i,t-1})$ 
6:    $v_{i,t}(q) = \chi\{v_i(q) + c_1r_1(Y_i(q) - q_{i,t-1})\} + c_2r_2(q'_i - q_{i,t-1})$ 
7:    $\lambda_{i,t} = \lambda_{i,t} + v_{i,t}(\lambda), q_{i,t} = q_{i,t} + v_{i,t}(q)$ 
8:    $f_{i,t} = \text{ICOMP}_{\text{MISS}}(\lambda_{i,t}, q_{i,t}, X, Y)$ 
9:   if  $f_{i,t} < f^*$  then
10:      $\lambda^* = \lambda_{i,t}, q^* = q_{i,t}$ 
11:   end if
12:   if  $f_{i,t} < P_i$  then
13:      $P_i = f_{i,t}, Y_i(\lambda) = \lambda_i, Y_i(q) = q_i$ 
14:   end if
15: end for
```

C EM Algorithm for SKFA (M-Step)

The following derivation is for maximizing the $Q(\Lambda, \Psi)$ function, namely the M-step. The $Q(\Lambda, \Psi)$ function is

$$Q(\Lambda, \Psi) = \mathbb{E}(\log L(\Lambda, \Psi)) - \frac{1}{2}P(\Lambda) \quad (1)$$

$$= -\frac{1}{2} \sum_{i=1}^n \log \Psi_{jj} - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\Psi_{jj})^{-1} (K_{ij}^2 - 2K_{ij}K[i,]\delta\Lambda'[, j]) \quad (2)$$

$$- \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\Psi_{jj})^{-1} (\Lambda[j,](\Delta + \delta'K[, i]K[i,]\delta)\Lambda[j,]') - \frac{1}{2}P(\Lambda). \quad (3)$$

We first take partial $Q(\Lambda, \Psi)$ function derivatives respecting Ψ and obtain

$$\frac{\partial}{\partial \Psi_{jj}} Q(\Lambda, \Psi) = 0, \quad (4)$$

$$M_{ij} = K_{ij}^2 - 2K_{i,j}K[i,]\delta\Lambda'[, j] + \Lambda[j,](\Delta + \delta'K[, i]K[i,]\delta)\Lambda[j,]', \quad (5)$$

$$\Psi_{ij} = n^{-1} \sum_{i=1}^n M_{ij}. \quad (6)$$

Then, we rewrite the $Q(\Lambda, \Psi)$ function as a quadratic form for Λ

$$Q(\Lambda, \Psi) = A\Lambda' + \Lambda B\Lambda' + P(\Lambda) + \text{constant}, \quad (7)$$

$$A = -2\Psi_{jj}^{-1} \left(\sum_{i=1}^n K_{ij}K[i,] \right) \delta, \quad (8)$$

$$B = (\Psi_{jj})^{-1} (n\Delta + \delta'K'K\delta). \quad (9)$$

The quadratic form can be expressed as an L_1 penalized regression as the following

$$\Lambda'[, j] = \arg \min_{\beta} \left\| \tilde{K} - Z\Lambda' \right\|_2^2 + \gamma \frac{\Psi_{jj}}{n} |\Lambda|, \quad (10)$$

$$\Sigma^s = n^{-1}K'K, \quad (11)$$

$$Z'Z = \Delta + \delta'\Sigma^s\delta, \quad (12)$$

$$\tilde{K} = (\Sigma^s[j,]\delta Z^{-1})'. \quad (13)$$

Vita

In 1993, Yaojin Sun was born in Yongji, Shanxi Province, China. After enrolling at Kangjie High School in 2008, he continued to study at Shanghai Jiao Tong University in 2011. In 2015, Yaojin received his undergraduate degree in Bioinformatics and Biostatistics. After that, he studied Genome Science and Technology at the University of Tennessee, Knoxville, and worked at the Oak Ridge National Laboratory. In 2017, Yaojin Sun changed his major to Dept. Of Business Analytics and Statistics to pursue his Ph.D. in Management Science (concentration: Business Analytics).