Dimension Reduction for Spatially-Misaligned and Multi-Pollutant Data with Missing Observations

Phuong Thu Vu

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

Doctor of Philosophy

University of Washington

2019

Reading Committee:

Adam A. Szpiro, Chair

Noah Simon

Kenneth M. Rice

Program Authorized to Offer Degree:
Department of Biostatistics

University of Washington

Abstract

Dimension Reduction for Spatially-Misaligned
and Multi-Pollutant Data with Missing Observations

Phuong Thu Vu

Chair of the Supervisory Committee:
Associate Professor Adam A. Szpiro
Department of Biostatistics

Accurate predictions of pollutant concentrations at new locations are often of interest in air pollution studies, in which data are usually not measured at all study locations. Ambient air is also a mixture of many chemical components, which can modify the associations between its total mass and various health outcomes. Principal component analysis (PCA) can be incorporated to obtain lower-dimensional representative scores of the multi-pollutant data. Spatial prediction models can then be used to estimate these scores at new locations. Recently developed predictive PCA (PredPCA) modifies the traditional algorithm to improve the overall predictive performance. However, these approaches require complete data, whereas multi-pollutant data tend to have complex missing patterns. In the first part of this dissertation, we propose a probabilistic version of PredPCA that can directly handle incomplete data with flexible model-based imputation accounting for geographic and spatial information. In the second part, we reformulate the PredPCA algorithm into a convex optimization problem by incorporating spatial information into the low-rank matrix completion framework. The advantages of our proposed method include simultaneous estimation of all components, orthogonality, and a mechanism to handle missing data. Finally, we leverage these core ideas to modify existing technique in low-rank tensor approximation to handle misaligned spatiotemporal data.

TABLE OF CONTENTS

Page

List of Figures iii

List of Tables vi

Chapter 1: Introduction 1

Chapter 2: Probabilistic Predictive Principal Component Analysis 4
2.1 Introduction 4
2.2 Motivating example 6
2.3 Review of traditional PCA and predictive PCA 7
2.4 Probabilistic predictive PCA 9
2.5 Simulations 12
2.6 Data application 20
2.7 Discussion 27

Chapter 3: Spatial Matrix Completion 30
3.1 Introduction 30
3.2 Review of low-rank matrix completion (LRMC) 32
3.3 The spatial matrix completion problem 33
3.4 Simulations 36
3.5 Data application 42
3.6 Discussion 50

Chapter 4: Higher-order and spatially predictive dimension reduction 53
4.1 Introduction 53
4.2 Review of some standard operations in tensor algebra 55
4.3 The Tucker decomposition with optimal rank approximation 61

4.4 Proposed algorithm: the spatial Tucker decomposition 62
4.5 Toy simulations 64
4.6 Data application 69
4.7 Discussion 78

Chapter 5: Discussion and Future Work 80

Bibliography 82

Appendix A: Appendix for Chapter 2 92
A.1 The ProPrPCA-Krige model and algorithm 92
A.2 The ProPrPCA-Spline model and algorithm 102
A.3 High-dimensional simulations 106

Appendix B: Appendix for Chapter 3 108
B.1 Proof of lemma 108
B.2 The low-rank matrix completion algorithm 112
B.3 The spatial matrix completion (SMC) algorithm 115
B.4 Additional high-dimensional simulation results 117
B.5 Additional results for 2011 CSN data 125

Appendix C: Appendix for Chapter 4 129
C.1 Proofs for the higher-order orthogonal iteration for Tucker decomposition . 129
C.2 Proofs for our proposed algorithm 134
C.3 Setups for toy simulations 139
C.4 Sensitivity analysis with different combinations of (M, Q) 141

LIST OF FIGURES

Figure Number Page

1.1 Spatial mislignment in health-pollutant cohort studies 1
1.2 Challenges and potential solutions for spatially-misaligned multi-pollutant data 2
2.1 Prediction R^2's and reconstruction errors from simulations with three-dimensional surface generated with spatially correlated noises 13
2.2 Prediction R^2's and reconstruction errors from simulations with three-dimensional surface generated with independent noises 15
2.3 Differences in prediction R^2 values between ProPrPCA-Spline and PredPCA for high-dimensional scenario 1 18
2.4 Differences in prediction R^2 values between ProPrPCA-Spline and PredPCA for high-dimensional scenario 2 19
2.5 Estimated loadings for feature with highly positive weights on SO_4^{2-} and S . 22
2.6 Estimated loadings for feature with highly positive weights on Na, Ni, and V 23
2.7 Estimated loadings for feature with highly positive weights on NO_3^- and Zn . 24
2.8 Prediction R^2's from leave-one-site-out cross-validation on 2010 CSN data . . 26
3.1 Differences in prediction R^2 values between SMC and PredPCA for high-dimensional scenario 1 39
3.2 Differences in prediction R^2 values between SMC and ProPrPCA for high-dimensional scenario 1 39
3.3 Differences in prediction R^2 values between SMC and PredPCA for high-dimensional scenario 2 41
3.4 Differences in prediction R^2 values between SMC and ProPrPCA for high-dimensional scenario 2 41
3.5 Computation time of ProPrPCA and SMC 42
3.6 Estimated loadings for feature with highly positive weights on SO_4^{2-} and S in 2010 CSN data 45
3.7 Estimated loadings for feature with highly positive weights on Na, Ni, and V in 2010 CSN data 46

3.8 Estimated loadings for feature with highly positive weights on NO_3^- and Zn in 2010 CSN data 47

4.1 Spatiotemporal data as third-order tensor 53
4.2 Mode-n unfolding matrix of a tensor 57
4.3 The Tucker decomposition 60
4.4 Distributions of median temporal R^2 for scenario 1 67
4.5 Distributions of median temporal R^2 for scenario 2 69
4.6 Location of analysis sites 70
4.7 Overall temporal trends in 2010 biweekly data 71
4.8 Overall temporal trends in 2011 biweekly data 72
4.9 Differences in temporal R^2 and MSE for 2010 biweekly data 75
4.10 Differences in temporal R^2 and MSE for 2011 biweekly data 77

B.1 Median estimated loadings in complete data scenario 120
B.2 Median estimated loadings in MCAR 35% scenario 121
B.3 Median estimated loadings in MAR scenario 122
B.4 Differences in prediction R^2 values between ProPrPCA and PredPCA for high-dimensional simulations with non-sparse loadings 123
B.5 Differences in prediction R^2 values between SMC and PredPCA for high-dimensional simulations with non-sparse loadings 124
B.6 Differences in prediction R^2 values between SMC and ProPrPCA for high-dimensional simulations with non-sparse loadings 124
B.7 Estimated loadings for feature with highly positive weights on SO_4^{2-} and S in 2011 CSN data 126
B.8 Estimated loadings for feature with highly positive weights on Na, Ni, and V in 2011 CSN data 127
B.9 Estimated loadings for feature with highly positive weights on NO_3^- and Zn in 2011 CSN data 128

C.1 Cross-validation prediction results for 2010 biweekly data with various combinations of (M, Q) 142
C.2 Differences in temporal R^2 and MSE for 2010 biweekly data with $(M = 2, Q = 5)$ 143
C.3 Differences in temporal R^2 and MSE for 2010 biweekly data with $(M = 5, Q = 5)$ 144
C.4 Cross-validation prediction results for 2011 biweekly data with various combinations of (M, Q) 145

C.5 Differences in temporal R^2 and MSE for 2011 biweekly data with $(M = 2, Q = 5)$146
C.6 Differences in temporal R^2 and MSE for 2011 biweekly data with $(M = 5, Q = 5)$147

LIST OF TABLES

Table Number Page

2.1 Estimated PC1 loadings from simulations with three-dimensional surface generated with spatially correlated noises 14
2.2 Estimated PC1 loadings from simulations with three-dimensional surface generated with independent noises 15
2.3 Median prediction R^2's for simulations under high-dimensional scenario 1 . . 17
2.4 Median prediction R^2's for simulations under high-dimensional scenario 2 . . 19

3.1 Median prediction R^2's for simulations under high-dimensional scenario 1 . . 38
3.2 Median prediction R^2's for simulations under high-dimensional scenario 2 . . 40
3.3 Leave-one-site-out cross-validation results for 2010 CSN data 48
3.4 Leave-one-site-out cross-validation results for 2011 CSN data 49
3.5 Evaluation of different approaches with imputation and dimension reduction 51

4.1 Median estimated loadings for scenario 1 66
4.2 Spatio-temporal prediction R^2 for scenario 1 67
4.3 Median estimated loadings for scenario 2 68
4.4 Spatio-temporal prediction R^2 for scenario 2 69
4.5 Cross-validation prediction results for 2010 biweekly data 74
4.6 Cross-validation prediction results for 2010 annual averages 76
4.7 Cross-validation prediction results for 2011 biweekly data 76
4.8 Cross-validation prediction results for 2011 annual averages 78

B.1 Median prediction R^2's for high-dimensional simulations with non-sparse loadings 119

ACKNOWLEDGMENTS

This dissertation and my survival through graduate school would not be possible without the support of many people.

I thank my advisor, Adam Szpiro, for his amazing mentorship, incredible calmness, and constant support over the years. I thank Adam for the many lessons that helped mold me into the capable and independent researcher that I am today. I also thank Adam for allowing me to explore various career paths, for connecting me with professionals, and for supporting me with my internship and job applications. Finally, I thank Adam for trusting in my ability, even when I barely did. I felt bad that he had to deal with my chaotic and anxious nature, and I sincerely hope that my current and future academic siblings will be more tolerable.

I have benefited greatly from the generous funding and great mentorship provided by Bryan Comstock and Patrick Heagerty at the Center for Biomedical Statistics (CBS). Particularly, I thank Bryan for throwing me into the lions den early to learn the arts of communication and managing expectations of collaborators for multiple projects at a time. I also thank Bryan for being brave enough to let me handle meetings with doctors and clinical practitioners alone, even when I was barely a junior researcher with very little experience. Finally, I thank Bryan for his kind words that helped me land my full-time job offer.

I thank Noah Simon for his guidance on my second project as well as his enormous support, both academic and non-academic, throughout my time in the graduate program. I thank Ken Rice for keeping my writing in check, and for being critical of my work but always so fun and considerate whenever I had a chance to meet him in person. I thank the rest of my doctoral committee members, Jennifer Bobb, June Spector, and a former member, Julia Cui, for their willingness to help with my research and other administrative aspects of the

process. I thank Timothy Larson for his collaboration on my first project, and the University of Washington MESA Air team for providing the data used in my dissertation.

I thank Gitana Garofalo for her unwavering support and commitment to our students' success. I am also fortunate to have received so much great personal and professional advice throughout the years from distinguished Biostatistics alumni, so many thanks to Joshua Keller, Andrew Spieker, Katherine Wilson, Katherine Tan, Asad Haris, Rita Shi, Shizhe Chen, and Kean Ming Tan. I thank the current biostatistics students, i.e. Entering Class of 2015 or later, for the friendships and fun conversations we had over the years, no matter how weird they were. Special shout-outs to Travis Hee Wai, for being an annoying but helpful-once-in-a-while academic little brother (hah); Taylor Okonek, for being a genuinely awesome housemate who also has a car; Aaron Hudson, for letting me squat in Slab Lab and indulging me with fun conversations all the time; and Qijun (Kendrick) Li, for being a lovely unofficial mentee.

Thank you to my friends from undergrad and high school, who allowed me to live vicariously through them. Special thanks to Stefan Dumlao for always being there to talk about everything and anything. I thank the Vietnamese friends I got to know over the last few years, Hoang Ly, Minh Nguyen, Nguyen Lam, An Nguyen, and Linh Le, for keeping me grounded and making me go out and have fun once in a while. I thank the Title Boxing Club - Greenwood community for keeping me in shape and allowing me to relieve stress in a socially acceptable way. Special thanks to Kevin Carino for putting up with my nonsense, for making sure I get home after boxing classes, and for enduring my never-ending rants.

Last but certainly not least, many thanks to the Biostatistics Entering Class of 2014, Natalie Gasca, Kelsey Grinde, Arjun Sondhi, Xiaowen Tian, Brian Williamson, Fan Xia, Yuxiang Xie, Chaoyu Yu, and Rui Zhuang, for our friendship and genuine camaraderie that got me through the struggles of coursework, qualifying exams, and the roller coaster years of research.

DEDICATION

I dedicated this dissertation to my lovely parents, Mr. Du Hong Vu and Ms. Lan To Nguyen. I would not be the person I am today if it was not for their open mindsets, valuing education, and occasionally tough love. I also dedicate this to my older sister, Ms. Trang Thu Vu, and my brother-in-law, Mr. Hieu Trung Dang. They both have supported me in their own ways throughout the years, and I wish all the bests to them and my two little troublemaker nephews.

Chapter 1

INTRODUCTION

Environmental studies on the health impacts of exposures to air pollution on human subjects often rely on pollutant data measured at regulatory monitoring locations. In these cohort studies on health-pollutant associations, the locations of study participants and the locations of fixed monitoring sites are often *spatially misaligned*, as illustrated in Figure 1.1. This challenge necessitates an exposure modeling stage, in which accurate predictions of pollutant concentrations at study locations are of interest.

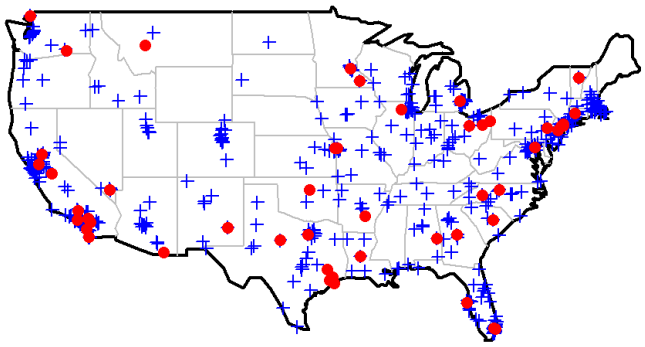

Figure 1.1: Illustration of spatial misalignment in health-pollutant cohort studies. The red dots represent the locations of fixed monitoring sites where pollutant data is available. The blue crosses mark the locations of study participants at which health outcome is available.

These analyses are further complicated by the *high dimension* of the ambient air data, which are often a mixture of many chemical components. Including some or all of these pollutants in statistical models can be problematic due to correlations and potential interactions among these components. Hence, dimension reduction is often necessary to obtain

a lower-dimensional representation of the original data. This challenge is exacerbated when there is a significant amount of *missing data*, with many monitoring locations missing one or more pollutants. Figure 1.2 illustrates an exposure modeling procedure under such presence of spatially-misaligned multi-pollutant monitoring data with missing observations. This often consists of three steps: (1) imputation for missing data, (2) dimension reduction to derive lower-dimensional representation, or scores, of the monitoring data, and (3) spatial prediction to estimate the corresponding scores at cohort locations. This dissertation is comprised of three projects that specifically addresses the first two steps of this procedure. The final goal is to make these steps more cohesive, and subsequently improve the predictive performance in the last stage of spatial prediction.

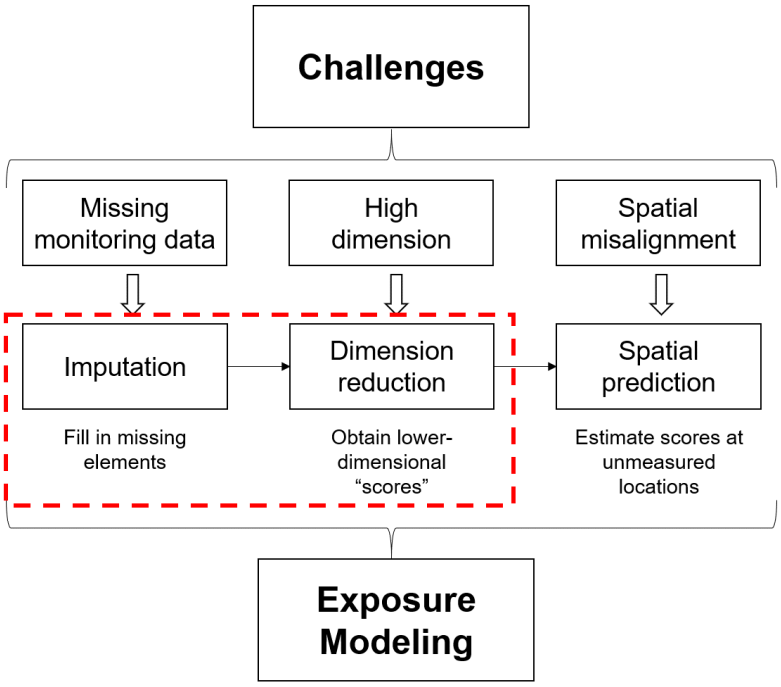

Figure 1.2: An exposure modeling procedure under the presence of spatially-misaligned multi-pollutant monitoring data with missing observations consists of three steps: *imputation* (filling in missing monitoring data), *dimension reduction* (obtaining lower-dimensional scores at monitoring locations), and *spatial prediction* (predicting corresponding scores at cohort locations). This dissertation focuses on the first two steps of this procedure (dashed box).

In Chapter 2, we propose two probabilistic extensions to the predictive principal component analysis algorithm developed by Jandarov et al. (2017). The methods, which we call *probabilistic predictive principal component analysis*, seek to produce lower-dimensional components that can be predicted well at new locations using available geographic and spatial covariates. The probabilistic assumptions allow flexible model-based imputation, which eliminates a separate preprocessing imputation step prior to dimension reduction. The methods are motivated, especially in health-pollution studies, by the scientific need to evaluate the effect modification of the air pollution profile on the main health-pollution associations of interest. The manuscript was first made available on May 1, 2019, as a preprint on arXiv (Vu et al., 2019) and was accepted on October 30, 2019, to be published in *Environmetrics*.

In Chapter 3, we tackle the same problem from a slightly different angle, using existing methods in low-rank matrix approximation and completion. To develop a practical and computationally feasible version of the existing method, we reformulate our problem into a convex optimization problem which we call *spatial matrix completion*. We derive a straightforward algorithm to solve this problem using proximal gradient descent.

In Chapter 4, we extend the entire framework to the setting of spatio-temporal data. Similarly, the goal is to obtain a lower-dimensional representation of the multi-pollutant time series that can be predicted at new locations. We propose a simple algorithm based on our core idea in Chapter 3 and an existing low-rank tensor approximation method.

Finally, we review our proposed methods and discuss potential directions for future research in Chapter 5.

Chapter 2

PROBABILISTIC PREDICTIVE PRINCIPAL COMPONENT ANALYSIS

2.1 Introduction

In recent years, there has been a growing interest in studying the role and health impact of $PM_{2.5}$, which is fine particulate matter with aerodynamic diameter less than 2.5 μm (Brook et al., 2004). $PM_{2.5}$ is a complex mixture of many components, and its chemical profile may vary drastically across time and space (Brook et al., 2004; Bell et al., 2007; Dominici et al., 2010). Obtaining a lower-dimensional representation of $PM_{2.5}$ multi-pollutant data is often necessary, as including many highly correlated pollutants in a statistical model is problematic. Principal component analysis (PCA) (Jolliffe, 1986) is an unsupervised dimension reduction technique that has gained popularity in multi-pollutant analysis (Dominici et al., 2003).

Examples of environmental studies utilizing $PM_{2.5}$ data include studies on the associations between various health outcomes and long-term (Pope III et al., 2002; Künzli et al., 2005; Miller et al., 2007; Chan et al., 2015; Kaufman et al., 2016) or short-term (Gold et al., 2000; Tolbert et al., 2007; Pascal et al., 2014; Achilleos et al., 2017; Hsu et al., 2017; Tian et al., 2017) exposures to $PM_{2.5}$. Many studies have suggested that the associations between $PM_{2.5}$ total mass and various health outcomes can be modified by some specific constituents or the overall chemical composition (Franklin et al., 2008; Bell et al., 2009; Krall et al., 2013; Zanobetti et al., 2014; Dai et al., 2014; Kioumourtzoglou et al., 2015; Wang et al., 2017; Keller et al., 2018).

In the United States, $PM_{2.5}$ studies often rely on data collected from regulatory monitoring networks managed by the Environmental Protection Agency (EPA). Unfortunately, for many pollution-health association studies, these fixed monitoring sites are usually not at the

same locations where health outcomes are available. Such *spatial misalignment* motivates an exposure modeling stage in which a spatial prediction model, such as land-use regression or universal kriging, is often used to estimate the exposure at unmeasured locations where pollutant data is not observed (Brauer et al., 2003; Künzli et al., 2005; Crouse et al., 2010; Bergen et al., 2013; Chan et al., 2015).

Derivation of a lower-dimensional representation of $PM_{2.5}$ multivariate data prior to making these spatial predictions is necessary, as predicting chemically and spatially correlated pollutant surfaces is challenging and intractable in most cases. As PCA is capable of performing dimension reduction without meddling with the health outcomes, it can be easily integrated in the analysis of spatially-misaligned data. Using PCA, lower-dimensional scores of the multi-pollutant data at monitoring locations can be obtained. These monitoring scores, along with geographic covariates, can then be used in a spatial prediction model to estimate the corresponding scores at unmeasured locations. However, PCA does not account for exogenous geographic information and spatial correlations across neighboring locations. Hence, PCA may produce scores that summarize the monitoring data well but are difficult to be predicted at unmeasured locations. A spatially predictive PCA algorithm (Jandarov et al., 2017) was developed to mitigate this issue by producing scores with spatial patterns that can be subsequently predicted well at new locations.

An additional challenge arises in practice where there is often a large amount of missing data, especially for multi-pollutant monitoring data. For example, not all $PM_{2.5}$ components are measured at all monitoring sites, either due to environmental considerations, logistic constraints or lack of resources. The missing patterns can sometimes be complex or spatially informative. Neither traditional PCA nor predictive PCA is well-equipped to deal with missing data, and thus a separate imputation step is required prior to dimension reduction. Existing non-parametric imputation schemes, ranging from simple mean imputation to sophisticated matrix completion, do not account for external spatial information. They may therefore distort the underlying spatial structure in the original data even before the dimension reduction stage, and thus worsen the predictive performance in the final stage.

In this chapter, our goal is to enhance the dimension reduction procedure under the presence of missing data by proposing a probabilistic framework in place of the deterministic algorithm of predictive PCA. Similar to Jandarov et al. (2017), our methods seek to produce principal components that can be predicted well at new locations. The added probabilistic assumptions allow for flexible model-based imputation that takes into account the embedded geographic and spatial information, and thus eliminates the need for a preprocessing stage with imputation.

2.2 Motivating example

To illustrate our proposed methods, we use data collected nationally by the Air Quality System (AQS) network of monitors managed by the EPA. Measurements of annually averaged $PM_{2.5}$ total mass and its components are only collected at a few subnetworks of AQS. For consistency with previous related work (Keller et al., 2017; Jandarov et al., 2017), we choose to use the 2010 data from the Chemical Speciation Network (CSN), of which monitoring sites are located strategically in various urban areas. Data is available for 21 components of $PM_{2.5}$: elemental carbon (EC), organic carbon (OC), sulfate ion (SO_4^{2-}), nitrate ion (NO_3^-), aluminum (Al), arsenic (As), bromine (Br), cadmium (Cd), calcium (Ca), chromium (Cr), copper (Cu), iron (Fe), potassium (K), magnesium (MN), sodium (Na), sulfur (S), silicon (Si), selenium (Se), nickel (Ni), vanadium (V), and zinc (Zn).

Geographic covariates are obtained for all available sites through the Exposure Assessment Core Database by the MESA Air team at the University of Washington. Data on roughly 600 Geographic Information System (GIS) covariates are available, including distances from roads, distances from major pollution sources, land-use information, vegetation indices, etc. The specific sources and attributions of these geographic covariates are carefully described in Bergen et al. (2013).

Data for 2010 is available for 221 CSN sites, with only 130 of those sites having complete data on all 21 components. Overall the amount of missing data in 2010 is roughly 32.1%. Not only do we compare the predictive performances following the application of different

PCA methods, but we also examine how different the chemical profiles are when considering only complete sites versus all available data. The data processing, analysis procedures, and results are discussed in Section 2.6.

2.3 Review of traditional PCA and predictive PCA

We denote $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ as the exposure data with p pollutants observed at n monitoring sites with spatial coordinates $\boldsymbol{s}_1, ..., \boldsymbol{s}_n$. The exposure data $\boldsymbol{X}$ may contain missing elements as some pollutants are not measured at all monitoring sites. Let $\boldsymbol{r}_i$ be a vector of k geographic covariates pertaining to the i-th monitoring sites. Variables corresponding to locations where the exposure is of interest but not measured are distinguished by an asterisk, i.e. $n^*, \boldsymbol{X}^*, \boldsymbol{s}_1^*, ..., \boldsymbol{s}_{n^*}^*, \boldsymbol{r}_1^*, ..., \boldsymbol{r}_{n^*}^*$.

The data of interest, $\boldsymbol{X}^*$, is high-dimensional but inaccessible. If $\boldsymbol{X}^*$ were observed, dimension reduction could be applied directly to obtain a lower-dimensional representation $\boldsymbol{U}^* \in \mathbb{R}^{n^* \times q}$ where $q < p$. Because of spatial misalignment, a spatial prediction model is required to estimate the unobserved exposures. Modeling highly correlated surfaces is challenging and may be excessive, given the final aim of recovering only the lower-dimensional $\boldsymbol{U}^*$. Thus, a sensible modeling procedure under the presence of spatially misaligned multi-pollutant data with missing observations may consist of several steps: (1) imputation for missing data, (2) dimension reduction to derive scores at monitoring sites, and (3) spatial prediction to estimate corresponding scores at new locations. In this paper, we focus on dimension reduction using PCA, an unsupervised technique that is suitable for handling spatially-misaligned data.

Traditional PCA provides a mapping from the original p-dimensional exposure surface to a corresponding q-dimensional representation where $\boldsymbol{X} \approx \boldsymbol{U}\boldsymbol{V}^\mathsf{T}$ for $q < p$. We refer to the orthogonal columns of $\boldsymbol{V} \in \mathbb{R}^{p \times q}$ as the loadings or principal directions. The columns of $\boldsymbol{U} \in \mathbb{R}^{n \times q}$, $\{\boldsymbol{u}_1, ..., \boldsymbol{u}_q\}$, are the principal component (PC) scores. These PC scores can be thought of as linear combinations of the original features of $\boldsymbol{X}$. These newly transformed variables are considered uncorrelated due to orthogonality of the loadings, which is an attractive feature

of PCA. The PCA algorithm is also optimal in the sense that the derived PC scores are conveniently ordered by the amount of variability explained in $\boldsymbol{X}$.

While PCA provides a unique solution in the reduced dimensions, the algorithm can be reformulated into a series of biconvex optimization problems, in which the loading and corresponding score of each PC can be solved in an iterative fashion (Shen and Huang, 2008),

$$\min_{\boldsymbol{u},\boldsymbol{v}} \left\| \boldsymbol{X} - \boldsymbol{u}\boldsymbol{v}^{\mathsf{T}} \right\|_F^2 \text{ s.t. } \|\boldsymbol{v}\|_2 = 1.$$

Utilizing this sort of optimization framework, Jandarov et al. (2017) develop a spatially predictive PCA algorithm (PredPCA hereafter) by directly incorporating spatial information in the objective function:

$$\min_{\boldsymbol{\alpha},\boldsymbol{v}} \left\| \boldsymbol{X} - \left(\frac{\boldsymbol{Z}\boldsymbol{\alpha}}{\|\boldsymbol{Z}\boldsymbol{\alpha}\|_2} \right) \boldsymbol{v}^{\mathsf{T}} \right\|_F^2,$$

where $\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{R} & \tilde{\boldsymbol{R}} \end{bmatrix}$, in which $\boldsymbol{R} \in \mathbb{R}^{n \times k}$ contains k GIS covariates, and $\tilde{\boldsymbol{R}} \in \mathbb{R}^{n \times \tilde{k}}$ contains $\tilde{k}$ thin-plate spline basis functions. The induced PC score, $\boldsymbol{Z}\boldsymbol{\alpha}/\|\boldsymbol{Z}\boldsymbol{\alpha}\|_2$, is constrained to have an underlying smooth spatial structure guided by geographic and spatial information encoded in $\boldsymbol{Z}$. An advantage of PredPCA over PCA is the capability to identify principal directions that lead to spatially predictable PC scores at unmeasured locations. Recent work by Bose et al. (2018) further improves PredPCA by adaptively selecting information to be included in $\boldsymbol{Z}$ for each PC.

When monitoring data is incomplete, simply omitting locations with missing data may reduce the usable sample size substantially; thus, imputation is often required prior to dimension reduction. Non-parametric techniques, ranging from mean imputation to matrix completions, are based on observed pollutant values but not additional spatial information. When the missingness is spatially informative, such imputation schemes may bias the results of these algorithms.

In the next section, we propose a probabilistic framework that aims to derive spatially predictive PC scores, with the ability to handle incomplete monitoring data and induce

flexible model-based imputation that accounts for spatial and geographic information.

2.4 Probabilistic predictive PCA

2.4.1 Probabilistic formulation with a latent variable model: the Krige algorithm

Tipping and Bishop (1999) proposed a probabilistic formulation of PCA based on a Gaussian latent variable model. Their model assumes $\boldsymbol{X} = \boldsymbol{u}\boldsymbol{v}^{\mathsf{T}} + \boldsymbol{E}$, where $\boldsymbol{u} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_n)$, $\boldsymbol{v} \in \mathbb{R}^p$, $\|\boldsymbol{v}\|_2 = 1$, and the elements of $\boldsymbol{E}$ are independently and identically distributed (i.i.d.) with mean zero and variance γ^2. We extend this framework by directly imposing a spatial mean and covariance structure on the latent variable space. That is, given a desired number of PCs, q, our model assumes

$$\boldsymbol{X} = \sum_{l=1}^{q} \left(\boldsymbol{u}_l \boldsymbol{v}_l^{\mathsf{T}} + \boldsymbol{E}_l \right),$$
$$\boldsymbol{u}_l = \boldsymbol{R}\boldsymbol{\beta}_l + \boldsymbol{\eta}_l,$$

where $\boldsymbol{\beta}_l \in \mathbb{R}^k$ includes the coefficients corresponding to the geographic covariates in $\boldsymbol{R}$, while $\boldsymbol{\eta}_l \in \mathbb{R}^n$ has zero mean and spatial covariance $\Sigma(\boldsymbol{\xi}_l)$, with $\boldsymbol{\xi}_l$ denoting the spatial covariance parameters of the latent space. We use similar constraint $\|\boldsymbol{v}_l\|_2 = 1$, and assume that $\Sigma(\boldsymbol{\xi}_l)$ has no nugget effect. The latent score $\boldsymbol{u}_l$ is stochastic with a full spatial distribution.

Let Θ_l be the collection of the model parameters, $\{\boldsymbol{v}_l, \boldsymbol{\beta}_l, \gamma_l^2, \boldsymbol{\xi}_l\}$, corresponding to the l-th PC. When the monitoring data is complete, estimate of the first loading, $\hat{\boldsymbol{v}}_1$, can be obtained using the original data matrix $\boldsymbol{X}$. The corresponding score $\hat{\boldsymbol{u}}_1$ at monitoring locations can then be calculated by projecting $\boldsymbol{X}$ onto the direction of $\hat{\boldsymbol{v}}_1$. In later steps, Θ_l can be estimated using $\boldsymbol{X}_l = \boldsymbol{X}_{l-1} - \hat{\boldsymbol{u}}_{l-1}\hat{\boldsymbol{v}}_{l-1}^{\mathsf{T}}$, where $\boldsymbol{X}_1 = \boldsymbol{X}$. The PC score $\hat{\boldsymbol{u}}_l$ can then be derived by projecting $\boldsymbol{X}_l$ onto $\hat{\boldsymbol{v}}_l$. Note that we use projection of the data matrix to obtain the PC score in each step instead of using model estimate of the latent mean $\boldsymbol{R}\boldsymbol{\beta}_l$. When some elements of $\boldsymbol{X}$ are missing, estimation of Θ_l is based only on the observed elements of $\boldsymbol{X}_l$. Estimated PC score $\hat{\boldsymbol{u}}_l$ can then be made by projecting the model-based imputed

exposure data onto the direction of $\hat{\boldsymbol{v}}_l$.

Our approach to estimate Θ_l in each step is similar to the EM algorithm employed by Tipping and Bishop (1999). We consider the latent variable $\boldsymbol{u}_l$ to be the "missing" portion, and thus the "complete" data consists of the observed $\boldsymbol{X}_l$ and the latent variable $\boldsymbol{u}_l$. The goal is then to maximize the joint likelihood of $\boldsymbol{X}_l$ and $\boldsymbol{u}_l$. The mathematical details and algorithms for both complete and missing monitoring data are described in Appendix A.1. We refer to this framework as the probabilistic predictive PCA, or ProPrPCA, hereafter. Specifically, we call this algorithm ProPrPCA-Krige due to the kriging formulation in the model assumptions.

It turns out that the ProPrPCA-Krige model is connected to the SupSVD model recently proposed by Li et al. (2016). The SupSVD model is expressed as $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^\mathsf{T} + \boldsymbol{E}$ where $\boldsymbol{U} = \boldsymbol{Y}\boldsymbol{B} + \mathbf{F}$. Here $\boldsymbol{U}$ is a the latent score matrix, $\boldsymbol{V}$ is a full-rank loading matrix, $\mathbf{F}$ and $\boldsymbol{E}$ are error matrices. Li et al. (2016) also propose an EM approach to estimate the model parameters. The ProPrPCA-Krige model is also related to the envelope model proposed in Cook et al. (2010), which is a more generalized version than SupSVD. As discussed in Li et al. (2016), the SupSVD model attempts to extract a low-rank representation of the original data based on some auxiliary data, while the envelope model aims to reduce variation in regression coefficient estimation. We note that our model arises in a different analytic problem and need. Particularly, it is motivated by spatial misalignment where data are not observed at cohort locations, but some geographic information is available. The end goal is also different, as we seek to accurately predict a low-rank representation of the data at unmeasured locations. Thus, our model is designed such that patterns of available covariates and spatial structures are properly induced in the latent scores at locations where we have data, so that we can easily predict them at new locations. An additional contribution is that we develop EM algorithms for parameter estimation for both complete and missing data scenarios.

2.4.2 Probabilistic formulation with thin-plate spline basis: the Spline algorithm

While the ProPrPCA-Krige algorithm is cohesive with a prediction stage using universal kriging, the parameter estimation stage appears to be computationally burdensome. In general, the EM algorithm is often computationally expensive and convergence is not always guaranteed. We propose a more simplified version of ProPrPCA,

$$\boldsymbol{X} = \sum_{l=1}^{q} \left((\boldsymbol{Z}\boldsymbol{\beta}_l)\boldsymbol{v}_l^{\mathsf{T}} + \boldsymbol{E}_l \right),$$

where $\boldsymbol{Z}$ contains thin-plate spline functions similar to PredPCA. Compared to the ProPrPCA-Krige model, the latent score $\boldsymbol{u}_l$ no longer has a stochastic component. Instead, $\boldsymbol{u}_l$ is now a smooth structure enriched with spatial patterns included in $\boldsymbol{Z}$.

Algorithm: ProPrPCA-Spline with complete monitoring data

Input $\boldsymbol{X}$, $\boldsymbol{Z}$, q, and t_{max}
  for l in $\{1, ..., q\}$ **do**
    $\boldsymbol{X}_l \leftarrow \boldsymbol{X}_{l-1} - \hat{\boldsymbol{u}}_{l-1}\hat{\boldsymbol{v}}_{l-1}^{\mathsf{T}}$ where $\boldsymbol{X}_0 = \boldsymbol{X}$, $\hat{\boldsymbol{u}}_0 = \boldsymbol{0}$, and $\hat{\boldsymbol{v}}_0 = \boldsymbol{0}$
    Initialize $\boldsymbol{v}_l^{(0)}$, $(\gamma_l^{(0)})^2$, $\boldsymbol{\beta}_l^{(0)}$, and $t = 1$
      while not converged **or** $t < t_{max}$ **do**
        $\boldsymbol{v}_l^{(t+1)} \leftarrow \tilde{\boldsymbol{v}}_l / \|\tilde{\boldsymbol{v}}_l\|_2$ where $\tilde{\boldsymbol{v}}_l \leftarrow \boldsymbol{X}_l^{\mathsf{T}}\boldsymbol{Z}\boldsymbol{\beta}_l^{(t)} / \left\|\boldsymbol{Z}\boldsymbol{\beta}_l^{(t)}\right\|_2^2$
        $\boldsymbol{\beta}_l^{(t+1)} \leftarrow \left(\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{Z}\right)^{-1} \left(\boldsymbol{Z} \otimes \boldsymbol{v}_l^{(t+1)}\right)^{\mathsf{T}} \mathrm{vec}(\boldsymbol{X}_l)$
        $(\gamma_l^{(t+1)})^2 \leftarrow (np)^{-1} \left\|\mathrm{vec}(\boldsymbol{X}_l) - (\boldsymbol{I}_n \otimes \boldsymbol{v}_l^{(t+1)})\boldsymbol{Z}\boldsymbol{\beta}_l^{(t+1)}\right\|_2^2$
        $t \leftarrow t + 1$
      end while
    $\hat{\boldsymbol{v}}_l \leftarrow \boldsymbol{v}_l^{(t)}$, $\hat{\gamma}_l^2 \leftarrow (\gamma_l^{(t)})^2$, $\hat{\boldsymbol{\beta}}_l \leftarrow \boldsymbol{\beta}_l^{(t)}$
    $\hat{\boldsymbol{u}}_l = \boldsymbol{X}_l\hat{\boldsymbol{v}}_l$
  end for
Output $\{\hat{\boldsymbol{v}}_1, ..., \hat{\boldsymbol{v}}_q\}$, $\{\hat{\boldsymbol{u}}_1, ..., \hat{\boldsymbol{u}}_q\}$, $\{\hat{\boldsymbol{\beta}}_1, ..., \hat{\boldsymbol{\beta}}_q\}$, $\{\hat{\gamma}_1^2, ..., \hat{\gamma}_q^2\}$

The overall procedure to obtain PC scores is similar to the Krige algorithm. When some elements of $\boldsymbol{X}_l$ are missing, estimation of $\hat{\Theta}_l = \{\boldsymbol{v}_l, \boldsymbol{\beta}_l, \gamma_l^2\}$ is based on the observed elements of $\boldsymbol{X}_l$, and estimated PC score $\hat{\boldsymbol{u}}_l$ can be derived by projecting the model-based

imputed exposure matrix onto the direction of $\hat{\boldsymbol{v}}_l$. When the monitoring data is complete, the algorithm for parameter estimation at each step is straightforward. The mathematical derivations and the algorithm for missing data are described in Appendix A.2. We refer to this model as ProPrPCA-Spline due to the use of thin-plate spline basis functions.

2.5 Simulations

We conduct two sets of simulations to compare the different PCA approaches. The first set involves a low-dimensional setting with three-pollutant exposure surfaces. The second set illustrates a higher-dimensional setting with 15 generated pollutant surfaces. In both cases, the multi-pollutant data is generated on a 100×100 grid ($N = 10,000$).

In each simulation, we randomly choose 400 training locations and 100 testing locations. We then apply the four competing methods (PCA, PredPCA, ProPrPCA-Krige, and ProPrPCA-Spline) to the training data, $\boldsymbol{X}^{train}$, to obtain the corresponding loading $\hat{\boldsymbol{v}}_l^{train}$ and score $\hat{\boldsymbol{u}}_l^{train}$, for $l = 1, ..., q$ where q is a desired number of PCs. We then use $\hat{\boldsymbol{u}}_l^{train}$ and relevant covariate information to obtain $\hat{\boldsymbol{u}}_l^{test}$, predicted scores at testing locations, in a universal kriging model with an exponential covariance assumption. Finally, we compare the predicted scores to the known scores, $\boldsymbol{u}_l^{test}$, which is defined by projecting $\boldsymbol{X}^{test}$ onto the direction of $\hat{\boldsymbol{v}}_l^{train}$.

We also consider various scenarios in which some training data is missing. These scenarios include missing completely at random (MCAR) , with 30%, 35%, and 40% of missing data, and missing at random (MAR), in which the missing patterns are associated with the generated spatial covariates. When there is missing data, we apply low-rank matrix completion (LRMC) via the SoftImpute algorithm (Mazumder et al., 2010) to fill in the missing entries prior to PCA and PredPCA.

There are several metrics to evaluate the predictive performance. The metric of interest is the prediction R^2 adapted from Szpiro et al. (2011), which reflects the correlation between $\hat{\boldsymbol{u}}_l^{test}$ and $\boldsymbol{u}_l^{test}$. We also look at the reconstruction error (RE), defined as $\|\boldsymbol{X}^{test} - \hat{\boldsymbol{X}}^{test}\|_F$ where $\hat{\boldsymbol{X}}^{test} = \hat{\boldsymbol{U}}^{test}(\hat{\boldsymbol{V}}^{train})^{\mathsf{T}}$, $\hat{\boldsymbol{U}}^{test} = \begin{bmatrix} \hat{\boldsymbol{u}}_1^{test} & ... & \hat{\boldsymbol{u}}_q^{test} \end{bmatrix}$, and $\hat{\boldsymbol{V}}^{train} = \begin{bmatrix} \hat{\boldsymbol{v}}_1^{train} & ... & \hat{\boldsymbol{v}}_q^{train} \end{bmatrix}$.

2.5.1 Three-dimensional exposure surfaces

We simulate three-dimensional surfaces with $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}$, and three independent covariates $\{\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3\}$. Only $\boldsymbol{r}_1 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_N)$ is "observed" and thus used in the universal kriging model. Both $\boldsymbol{r}_2 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_N)$ and $\boldsymbol{r}_3 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_N)$ are unobserved and primarily used to induce correlations across $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}$. We generate data such that $\boldsymbol{x}_1 = 4\boldsymbol{r}_1 + 2\boldsymbol{r}_3 + \boldsymbol{\epsilon}_1$, $\boldsymbol{x}_2 = 3\boldsymbol{r}_2 + \boldsymbol{\epsilon}_2$, and $\boldsymbol{x}_3 = 2\boldsymbol{r}_1 + 4\boldsymbol{r}_2 + \boldsymbol{\epsilon}_3$, where $\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2, \boldsymbol{\epsilon}_3 \sim \mathcal{N}(\boldsymbol{0}, \Sigma)$, where Σ has an exponential structure with partial sill $\sigma^2 = 3.5^2$, nugget $\tau^2 = 1$, and range $\phi = 50$. Under this setting, only $\boldsymbol{x}_1$ and $\boldsymbol{x}_3$ are predictable by $\boldsymbol{r}_1$. While not dependent on $\boldsymbol{r}_1$, $\boldsymbol{x}_2$ is moderately correlated with $\boldsymbol{x}_3$ via $\boldsymbol{r}_2$. We also generate a second set of data in which the errors $\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2, \boldsymbol{\epsilon}_3 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{1})$. For MAR scenarios, $\boldsymbol{x}_1$ is missing at training locations where $\boldsymbol{r}_1$ values are larger than its 80th sample percentile, while $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ have 20% MCAR. We look at the first PC for these simulations, i.e. $q = 1$.

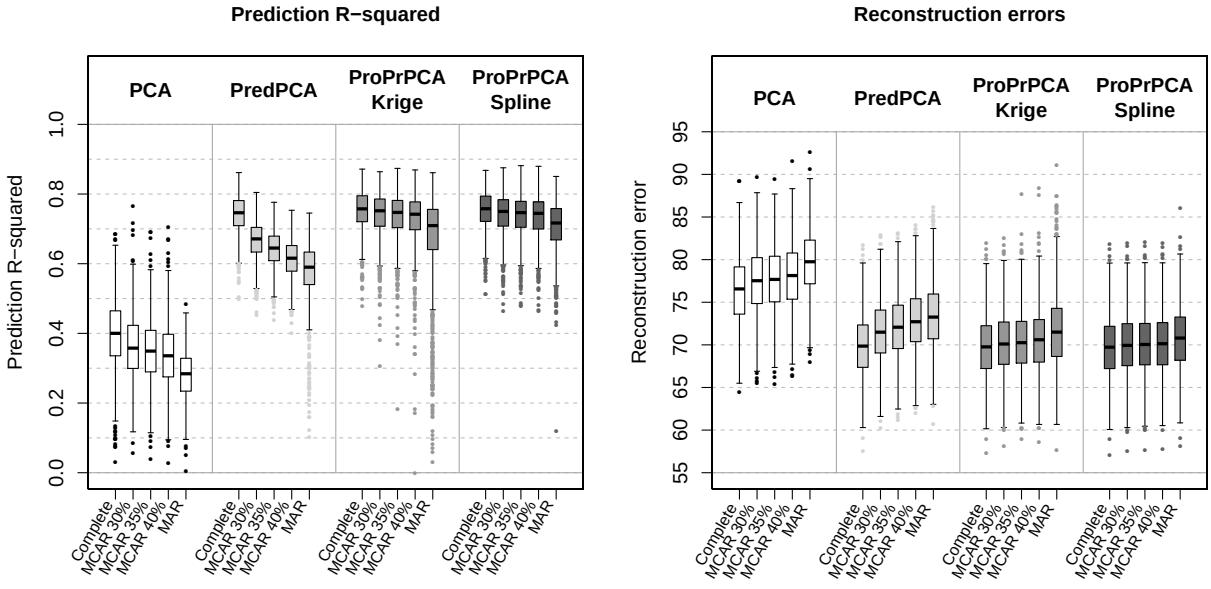

Figure 2.1: Prediction R^2's and reconstruction errors across 1,000 replications with three-dimensional surface generated with spatially correlated noises. Under missing data scenarios, LRMC is used prior to the application of either PCA or PredPCA.

Figure 2.1 shows the prediction R^2's and REs across 1,000 simulations for data generated with spatially correlated noise. Table 2.1 displays the means and standard deviations of the estimated loadings from each method when the training data is complete. The principal direction produced by PCA is loaded heavily on $\boldsymbol{x}_3$ and only moderately on both $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. This leads to poor predictive performance for PCA (median $R^2 = 0.40$). Meanwhile, loadings from the other three methods put the most weight on $\boldsymbol{x}_1$ and some on $\boldsymbol{x}_3$, thus they have higher prediction R^2's (median R^2's are about 0.75) and lower REs.

Table 2.1: Means (standard deviations) of estimated PC1 loadings across 1,000 replications with three-dimensional surface with spatially correlated noise and complete training data.

	X_1	X_2	X_3
PCA	0.40 (0.11)	0.41 (0.09)	0.80 (0.07)
PredPCA	0.88 (0.04)	-0.07 (0.04)	0.46 (0.09)
ProPrPCA-Krige	0.85 (0.04)	-0.11 (0.08)	0.50 (0.08)
ProPrPCA-Spline	0.86 (0.03)	-0.12 (0.07)	0.49 (0.07)

Under MCAR scenarios, prediction R^2's substantially decrease and REs increase for both PCA and PredPCA as the amount of missing data increases. Median R^2 of PredPCA drops to as low as 0.64 when training data is 35% MCAR. On the other hand, there are only some subtle reductions in the predictive performances of both ProPrPCA approaches. Under MAR, the performances of both PCA and PredPCA are significantly worse. While ProPrPCA-Krige performs better than PredPCA on average, the variability in performance is high across simulations. Despite not achieving the same level as when the data is complete, ProPrPCA-Spline has the highest predictive performance among the four competing methods.

Table 2.2 shows the estimated loadings with complete data, while Figure 2.2 shows the prediction R^2's and REs across 1,000 simulations for data generated with independent noise. Similar trends, where ProPrPCA outperforms the rest when missing data is more severe, are also observed in this set of generated data.

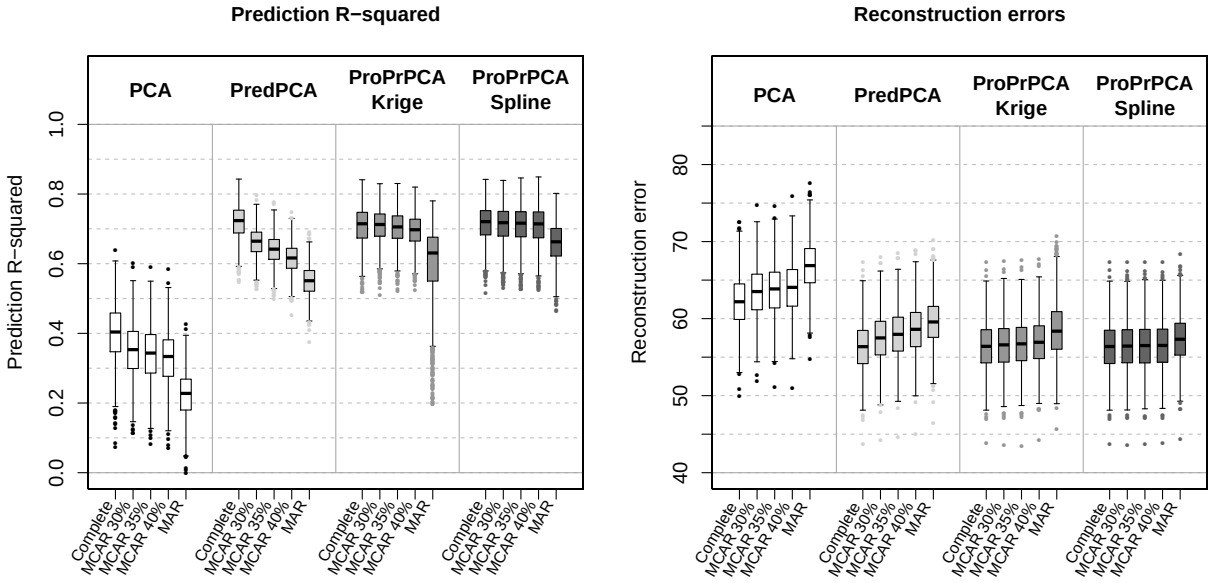

Figure 2.2: Prediction R^2's and reconstruction errors across 1,000 replications with three-dimensional surface generated with independent noises. Under missing data scenarios, LRMC is used prior to the application of either PCA or PredPCA.

Table 2.2: Means (standard deviations) of estimated PC1 loadings across 1,000 replications with three-dimensional surface with independent noise and complete training data.

	X_1	X_2	X_3
PCA	0.53 (0.06)	0.39 (0.04)	0.75 (0.03)
PredPCA	0.89 (0.02)	0.01 (0.02)	0.45 (0.04)
ProPrPCA-Krige	0.88 (0.02)	0.03 (0.04)	0.47 (0.04)
ProPrPCA-Spline	0.89 (0.02)	0.01 (0.03)	0.46 (0.04)

2.5.2 High-dimensional exposure surfaces

We also demonstrate the performance of ProPrPCA algorithms via simulations with 15 generated pollutants. The full setup is described in Appendix A.3. Overall, the high-dimensional exposure surfaces are generated from three underlying scores, $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, and $\boldsymbol{u}_3$. The data generating mechanism is such that $\boldsymbol{u}_1$ is the most spatially predictable, $\boldsymbol{u}_2$ is moderately

predictable, and $\boldsymbol{u}_3$ is not predictable by any covariates used in the universal kriging model. The loadings used to generate the data are sparse, in order to clearly identify the behaviors of the PCA methods. That is, the first five pollutants, $(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_5)$, are generated from $\boldsymbol{u}_1$. Meanwhile, $(\boldsymbol{x}_6, \boldsymbol{x}_7, \boldsymbol{x}_8, \boldsymbol{x}_9, \boldsymbol{x}_{10})$ are generated from $\boldsymbol{u}_2$, and $(\boldsymbol{x}_{11}, \boldsymbol{x}_{12}, \boldsymbol{x}_{13}, \boldsymbol{x}_{14}, \boldsymbol{x}_{15})$ are generated from $\boldsymbol{u}_3$. For MAR scenario, we induce a mild spatial pattern in the missing data for the first five pollutants. In these simulations, we evaluate the predictive performance based on two PCs, i.e. $q = 2$.

We create two scenarios: scenario 1 with $Var(\boldsymbol{u}_1) = 10$, $Var(\boldsymbol{u}_2) = 7.5$, and $Var(\boldsymbol{u}_3) = 5$, and scenario 2 with $Var(\boldsymbol{u}_3) = 10$, $Var(\boldsymbol{u}_1) = 7.5$, and $Var(\boldsymbol{u}_2) = 5$. In scenario 1, where the order of variance contribution is the same as the order of spatial predictability, we expect all methods to identify linear combinations of $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ as the first two PCs when training data is complete. In scenario 2, the non-predictable score $\boldsymbol{u}_3$ has the highest variance contribution. Thus we expect PCA to identify linear combinations of $\boldsymbol{u}_3$ and $\boldsymbol{u}_1$ for the first two PCs, with a large contribution of $\boldsymbol{u}_3$ for the first PC. Meanwhile, we anticipate the other predictive methods to still pick linear combinations of $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$.

Table 2.3 shows the results for the prediction R^2's across 1,000 simulations under scenario 1. As expected under scenario 1, all methods perform comparably when the training data is complete. While the results for MCAR 30% and 40% are not shown in this chapter, we observed similar patterns to the three-dimensional simulations where the performance of PCA and PredPCA decreases steadily as the amount of MCAR missing data increases. Under MCAR 35% setting, ProPrPCA-Spline has the best median R^2's for both PCs.

Under MAR, data among the first five pollutants are more likely to be missing at locations with extreme geographic covariate values. This setup effectively has an impact on the actual variance contributions of the underlying scores in a given sample, and particularly lowers the variability contributed by $\boldsymbol{u}_1$. As a result, for PC1, PCA is likely to produce loadings with higher contribution from $\boldsymbol{u}_2$ than before. As the predictive methods (PredPCA and ProPrPCA) attempt to balance out the trade-off between data representativeness and spatial predictability, these methods will also likely to obtain linear combinations with more weights

from $\boldsymbol{u}_2$ for PC1 than before. Subsequently, linear combinations obtained for PC2 will have more weights from $\boldsymbol{u}_1$ than before. This explains the decreases in median R^2's of PC1 for all methods but slight increases for PC2. ProPrPCA-Spline notably has the best median R^2 for PC1.

Table 2.3: The median prediction R^2's across 1,000 simulations for high-dimensional scenario 1. Under missing data scenarios, LRMC is used prior to either PCA or PredPCA.

PC1	Complete	MCAR 35%	MAR
PCA	0.83	0.80	0.61
PredPCA	0.84	0.81	0.63
ProPrPCA-Krige	0.83	0.83	0.64
ProPrPCA-Spline	0.84	0.83	0.69
PC2	Complete	MCAR 35%	MAR
PCA	0.60	0.58	0.67
PredPCA	0.60	0.58	0.68
ProPrPCA-Krige	0.60	0.60	0.69
ProPrPCA-Spline	0.60	0.60	0.68

We further compare the differences in R^2 values between ProPrPCA-Spline and PredPCA in Figure 2.3. With complete training data, ProPrPCA-Spline outperforms PredPCA for only less than 60% of the simulations, and the magnitude of the difference between the two methods is rather negligible. Under MCAR 35%, ProPrPCA-Spline outperforms PredPCA for both PCs in 69.7% of the 1,000 simulations, and, for 28.5% of the time, ProPrPCA-Spline is better in one of the PCs. Finally, under MAR, there are only 2.5% of the simulations in which ProPrPCA-Spline is worse than PredPCA for both PCs. There are 38.7% of the simulations where ProPrPCA-Spline is better for only PC1 (blue top-left quadrant). Particularly for points lying in this quadrant, the greater spread along the y-axis implies that a higher increase in R^2 for PC1 is often accompanied by a smaller decrease in R^2 for PC2. Thus ProPrPCA-Spline shows more prominent benefits for PC1 without trading off too much in predictability of PC2.

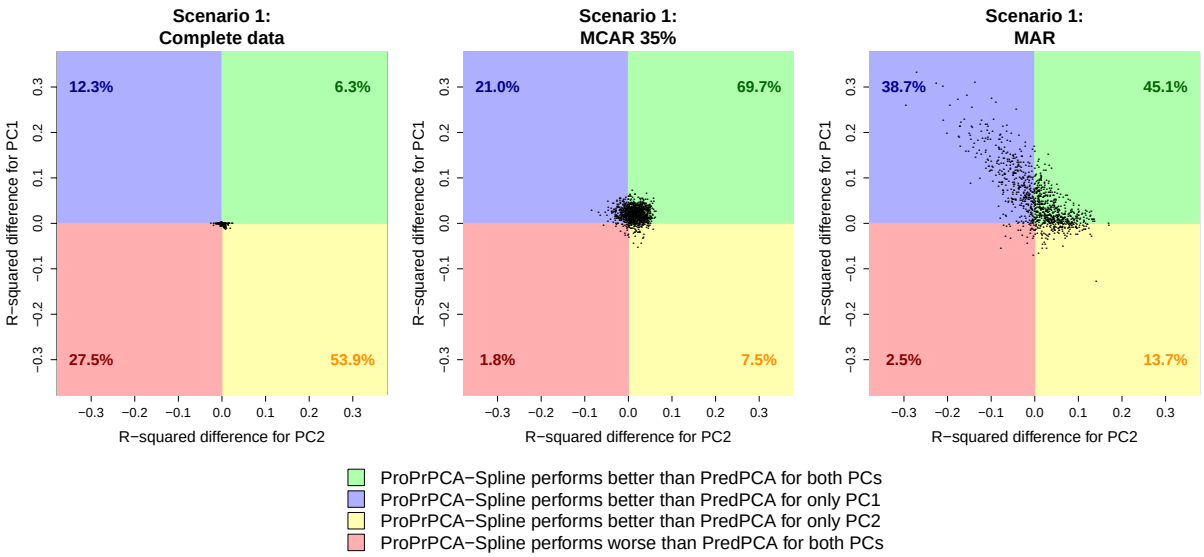

Figure 2.3: Differences in prediction R^2 values between ProPrPCA-Spline and PredPCA for high-dimensional scenario 1. Each dot represents result from one simulation. Percentages indicate the proportion out of 1,000 simulations.

Table 2.4 and Figure 2.4 show the corresponding results under scenario 2. In this scenario, as expected, PCA often identifies linear combinations of $\boldsymbol{u}_3$ and $\boldsymbol{u}_1$ as the first two PCs, and thus the predictive performance is generally poor, especially for PC1. ProPrPCA-Krige severely underperforms compared to PredPCA and ProPrPCA-Spline, even with complete data. Both PredPCA and ProPrPCA-Spline produce similar median R^2's with complete data. Similar to scenario 1, ProPrPCA-Spline performs consistently well with an increasing amount of MCAR, while the performance of PredPCA deteriorates. ProPrPCA-Spline shows clear benefits under MAR, particularly for PC1 (0.72) compared to PredPCA (0.63). The visualization of the differences in prediction R^2's between ProPrPCA-Spline and PredPCA in Figure 2.4 further supports similar conclusions to those of scenario 1.

Table 2.4: The median prediction R^2's across 1,000 simulations for high-dimensional scenario 2. Under missing data scenarios, LRMC is used prior to either TradPCA or PredPCA.

PC1	Complete	MCAR 35%	MAR
PCA	0.01	0.01	0.00
PredPCA	0.81	0.78	0.63
ProPrPCA-Krige	0.70	0.66	0.41
ProPrPCA-Spline	0.81	0.80	0.72

PC2	Complete	MCAR 35%	MAR
PCA	0.78	0.74	0.60
PredPCA	0.56	0.54	0.62
ProPrPCA-Krige	0.30	0.26	0.23
ProPrPCA-Spline	0.56	0.56	0.59

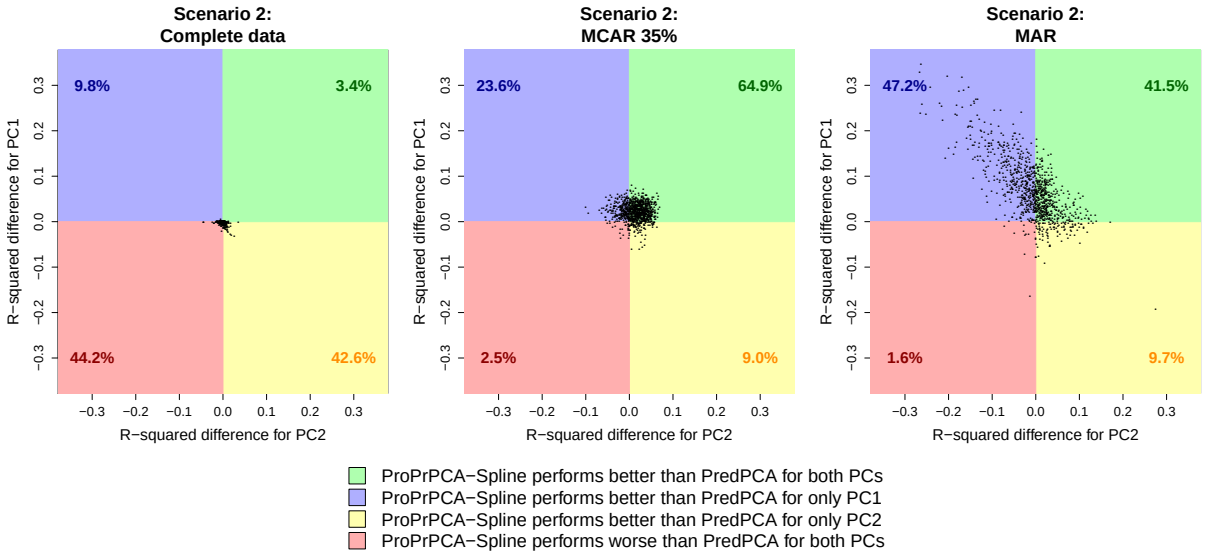

Figure 2.4: Differences in prediction R^2 values between ProPrPCA-Spline and PredPCA for high-dimensional scenario 2. Each dot represents result from one simulation. Percentages indicate the proportion out of 1,000 simulations.

2.6 Data application

2.6.1 Methods

In this section, we first compare the pollutant profiles obtained by different dimension reduction methods to the annual average 2010 CSN data. Prior to our analysis, we take a similar approach to Keller et al. (2017) and convert the mass concentrations of $PM_{2.5}$ components to proportions by dividing by the total mass of $PM_{2.5}$, and then log-transform these proportions. We also follow a similar preprocessing procedure as described in Keller et al. (2017) and Jandarov et al. (2017) to the GIS covariates to be used in the predictive algorithms and spatial prediction model. That is, we remove covariates that are missing at all chosen sites, have the same values in at least 80% of the sites, or have at least 2% of their values being more than five standard deviations away from the sample mean. We also remove land-use covariates whose maximal value is only 10% among all chosen sites. Finally, we apply PCA on the processed GIS data and use the first five PCs in later stages.

After the preprocessing procedure, we end up with a total of 221 CSN sites, only 130 of which have complete data on all 21 $PM_{2.5}$ components. We first apply three methods, PCA, PredPCA, and ProPrPCA-Spline, on the 130 sites with complete data (the "complete" set). We then proceed to apply these methods on all 221 CSN sites (the "full" set), where LRMC is applied prior to PCA and PredPCA. The goal is to assess how the estimated loadings and PC scores change when using only sites with complete data compared with using all available sites. The design matrix, $\boldsymbol{Z}$, used in PredPCA and ProPrPCA-Spline includes the five PCs of GIS covariates and thin-plate spline basis functions generated from the spatial coordinates, similar to Jandarov et al. (2017). We do not use ProPrPCA-Krige in our comparison because of its computational cost and inferior performance compared to ProPrPCA-Spline in our previously described simulations.

We also conduct leave-one-site-out cross-validation to compare the predictive performances among these methods. In each round of cross-validation, we leave out one site among the complete sites as test data. We then perform dimension reduction and fit a uni-

versal kriging model on training data comprised of either only the remaining complete sites (the "complete" training data), or all remaining sites (the "full" training data), while the testing data in each round stays the same. The goal is to assess the predictive performance of different methods with both complete and missing data.

2.6.2 Results

The multi-pollutant profile

Figure 2.5 shows the estimated loadings and the spatial distributions of corresponding scores of the first PC for four combinations of method and dataset: PCA applied to the complete set, PredPCA applied to the complete set, imputation followed by PredPCA applied to the full set, and ProPrPCA-Spline applied to the full set. The results for ProPrPCA-Spline when using the complete set (not shown here) are essentially identical to PredPCA results.

The estimated PC1 loadings are similar across PredPCA applied to either sets and to ProPrPCA-Spline, with highly positive weights on SO_4^{2-} and S and highly negative weights on Al, Ca, Na, and Si. Highly positive scores are observed in the east and part of the Midwest, probably due to sulfur emissions from coal combustion (Thurston et al., 2011; Hand et al., 2012). Negative scores are observed in the west and southwest, and have a classic resuspended soil profile (Thurston et al., 2011; Tong et al., 2012; Clements et al., 2017). While the spatial distribution of PCA scores looks similar to other methods, loadings obtained by PCA applied to the complete set are fundamentally different than the rest, with much weaker positive weights on SO_4^{2-} and S, and strongly negative weights on many additional elements, including Cr, Cu, Fe, Mn, Ni, Zn.

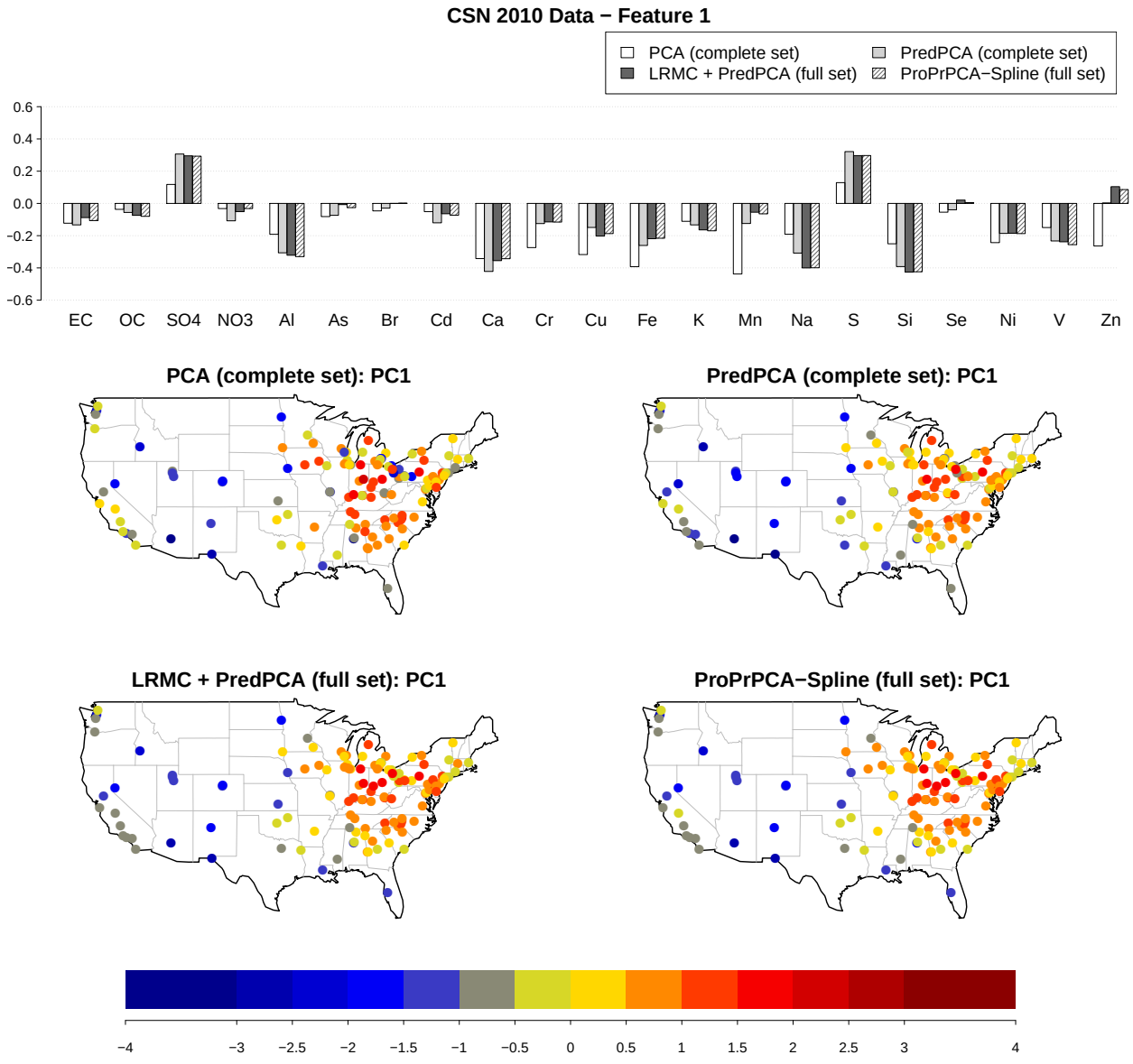

Figure 2.5: Estimated loadings for feature with highly positive weights on SO_4^{2-} and S, and corresponding scores, obtained from different PCA algorithms applied to 2010 CSN data: PCA and PredPCA applied to the complete set (130 sites with complete data), PredPCA and ProPrPCA-Spline applied to the full set (all 221 available sites).

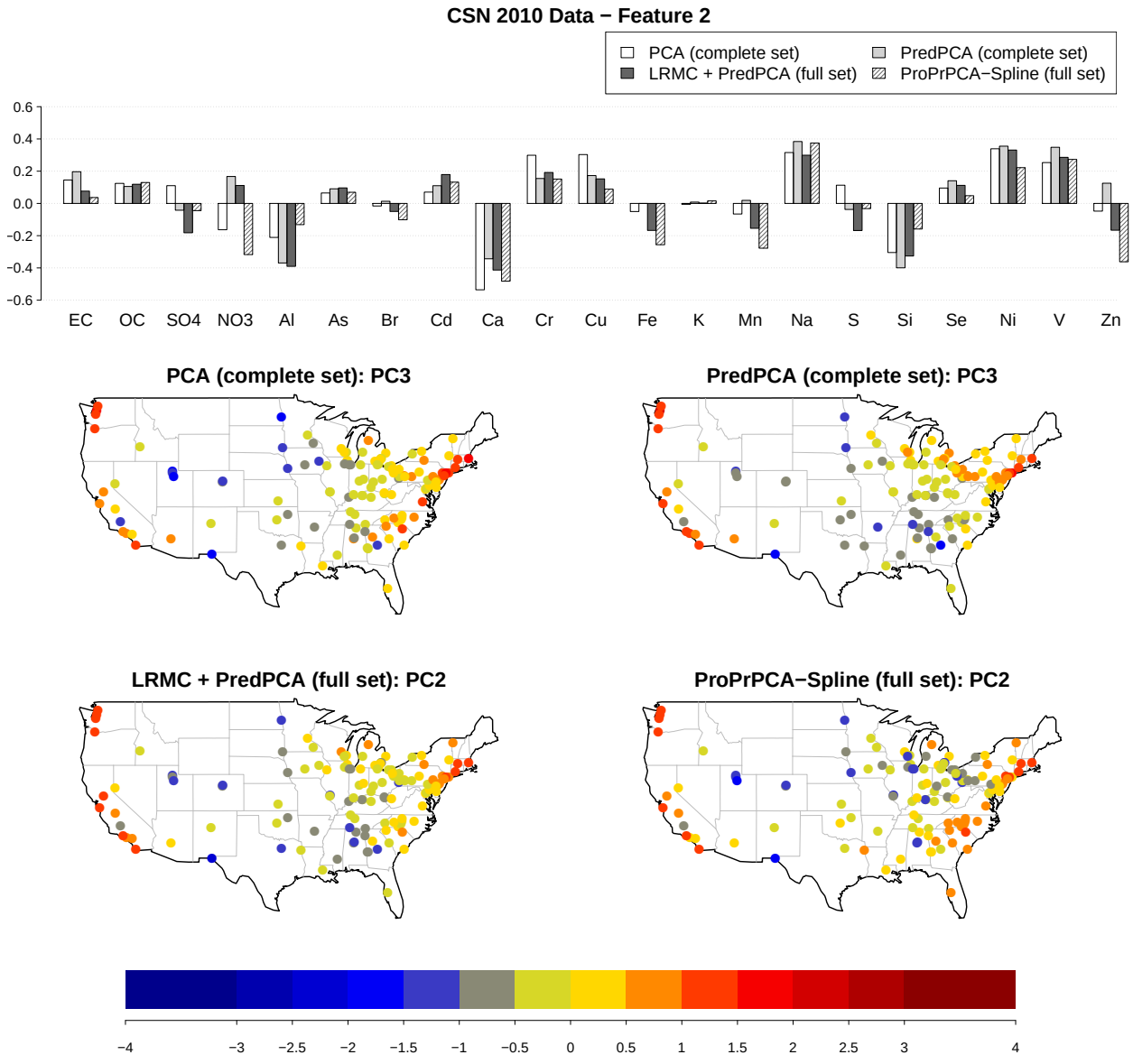

Figure 2.6: Estimated loadings for feature with highly positive weights on Na, Ni, and V, and corresponding scores, obtained from different PCA algorithms applied to 2010 CSN data: PCA and PredPCA applied to the complete set (130 sites with complete data), PredPCA and ProPrPCA-Spline applied to the full set (all 221 available sites).

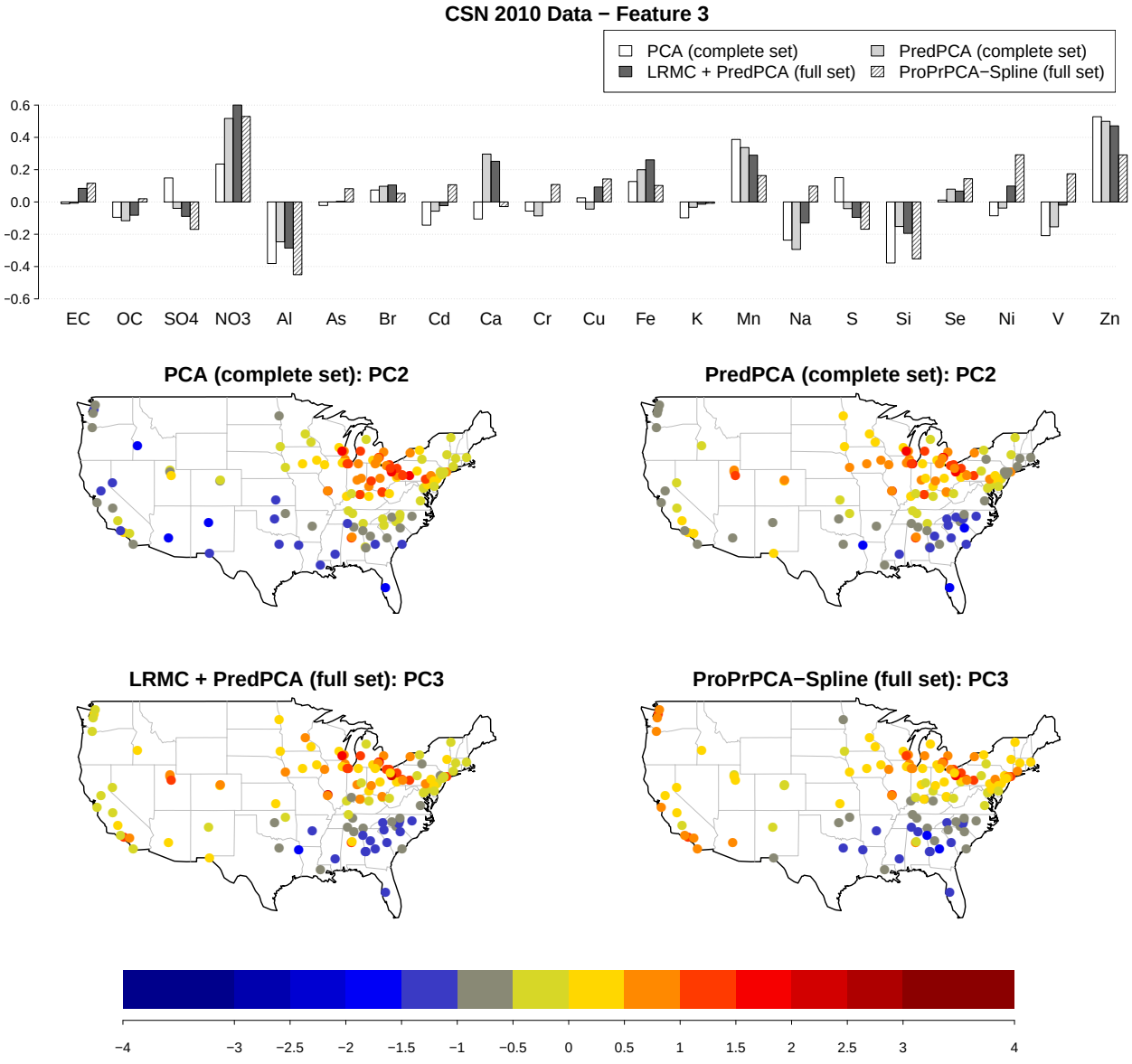

Figure 2.7: Estimated loadings for feature with highly positive weights on NO_3^- and Zn, and corresponding scores, obtained from different PCA algorithms applied to 2010 CSN data: PCA and PredPCA applied to the complete set (130 sites with complete data), PredPCA and ProPrPCA-Spline applied to the full set (all 221 available sites).

Figure 2.6 shows the estimated loadings and the score distributions for the PC that has a highly positive composition of Na, Ni, and V. This feature corresponds to PC3 ob-

tained by PCA or PredPCA applied to the complete set, and PC2 obtained by PredPCA or ProPrPCA-Spline applied to the full set. ProPrPCA-Spline results in highly positive scores along the west coast, the east coast, and southeast region, possibly due to residual oil combustion (Thurston et al., 2011), and marine aerosol (Thurston et al., 2011; Kotchenruther, 2017). ProPrPCA-Spline also identifies pronounced negative loadings on Zn and NO_3^-. The remaining three combinations of methods and datasets are able to produce fairly similar maps with strongly positive scores along the west coast and across the northern east coast, although they fail to highlight some relevant coastal locations in the southeast region.

Figure 2.7 shows the results for features highly positive in NO_3^- and Zn, which corresponds to PC2 obtained by PCA or PredPCA applied to the complete set, and PC3 obtained by PredPCA or ProPrPCA-Spline applied to the full set. For all methods, highly positive scores are observed in the northern Midwest, possibly due to nitrate hazes (Coutant et al., 2003; Pitchford et al., 2009; Hand et al., 2012). Additionally, loadings produced by ProPrPCA-Spline are also strongly positive in Ni, V, and negative in Al, Si, with greater magnitude compared to other methods. Thus, moderately positive scores are also observed along the west coast. ProPrPCA-Spline also results in highly positive scores in the southeast region due to the calcium poor soils in that region compared to Al and Si content (Shacklette and Boerngen, 1984).

Cross-validation results

Finally, we look at the predictive performances in leave-one-site-out cross-validations, and the results are shown in Figure 2.8. While having decent performance for PC2 and PC3 ($R^2 = 0.51$), using PCA applied to the complete training data yields a poor result for PC1 ($R^2 = 0.24$). PredPCA has similar performances for PC1 with either complete or full training data. However, there is a substantial trade-off in performances between PC2 and PC3, which can potentially be explained by the switching between PC2 and PC3 observed in the pollutant profile. ProPrPCA-Spline applied on the full training data shows the highest predictive performance for PC1 ($R^2 = 0.57$) and PC3 ($R^2 = 0.69$), but suffers from a decrease

in the ability to predict PC2 well ($R^2 = 0.35$).

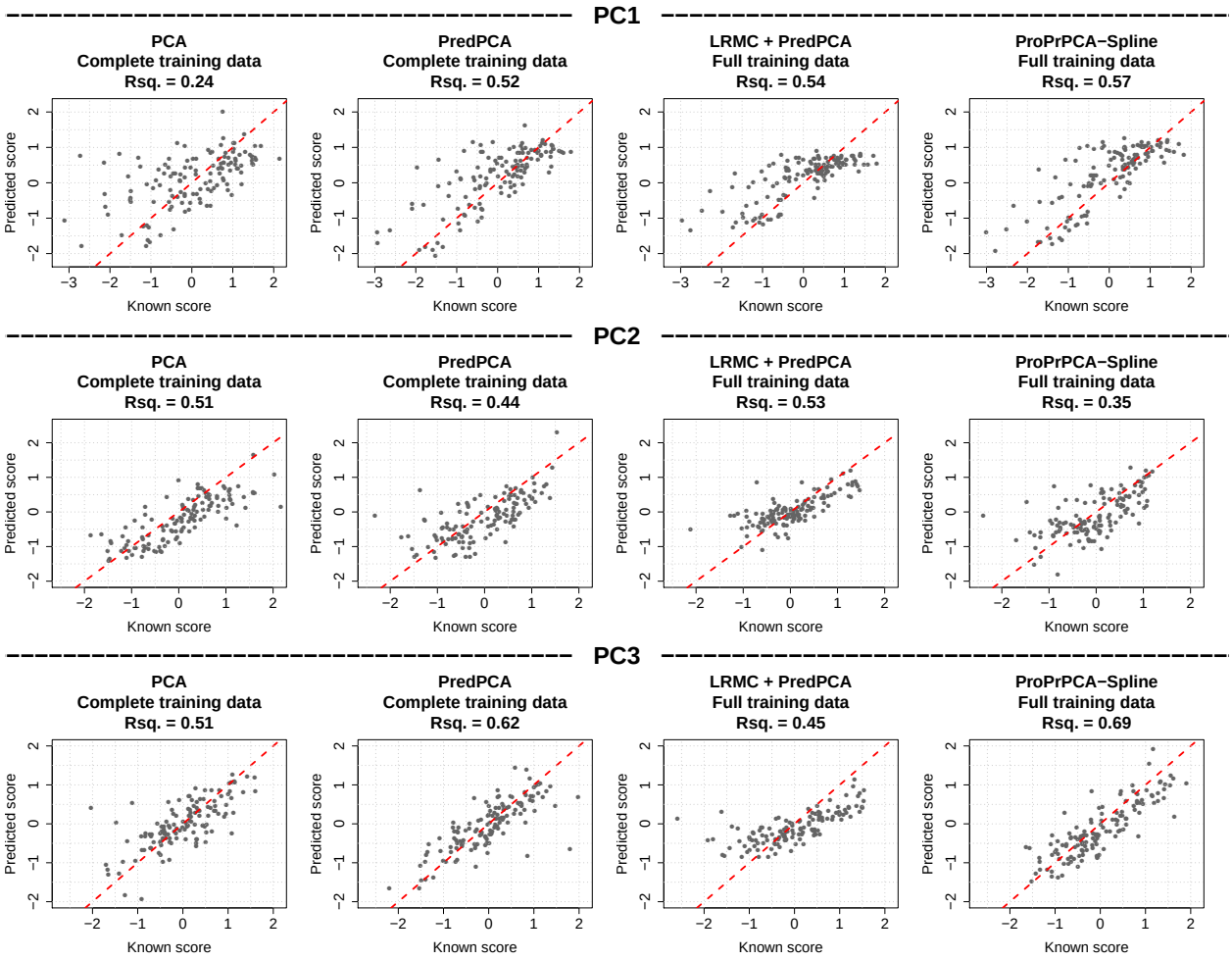

Figure 2.8: Prediction R^2's from leave-one-site-out cross-validation on 2010 CSN data. Sites with complete $PM_{2.5}$ component data are used as testing data. Training data may include only complete sites, or all available sites.

A possible explanation to the overall relatively low R^2's for all methods is that we use the same pre-specified spatial information encoded in $\boldsymbol{Z}$ to characterize the spatial variability across all PCs, which may not be effective. A potential solution, which is beyond the scope of this paper, is adaptive selection of features to be included in $\boldsymbol{Z}$, which is proposed and discussed in Bose et al. (2018).

2.7 Discussion

In this chapter, we propose a probabilistic extension to the PredPCA algorithm developed by Jandarov et al. (2017). The proposed ProPrPCA algorithms can be applied to misaligned multi-pollutant data with missing observations. The ultimate goal is to improve the predictive performance of the exposure modeling stage that is often required in air pollution studies that rely on fixed site monitoring data. In spite of their simplicity, these probabilistic extensions are effective in mitigating the impact of missing data on the predictive performance of the exposure model. The proposed methods also eliminate the necessity of a separate imputation procedure prior to dimension reduction. The scientific motivation, especially in health-pollution studies on $PM_{2.5}$ and its components, includes the ability to use estimated PC scores at study locations as effect modifiers for the main health associations of interest.

We have demonstrated via simulations that ProPrPCA-Spline consistently outperforms its competitors under various missing observation scenarios. Its computational speed is on par with both PCA and PredPCA, which are non likelihood-based methods. The complex version, ProPrPCA-Krige, assumes a universal kriging formulation for the latent variable, with the mean model enriched by spatial covariates, and spatial correlations among the residuals. ProPrPCA-Spline incorporates thin-plate spline basis functions, which can be regarded as an alternative to a fixed low-rank kriging model (Kammann and Wand, 2003). Intuitively, the latent specification of ProPrPCA-Krige would have been cohesive with the later prediction stage using universal kriging. Possible explanations for the inferior performance of the Krige algorithm in simulations include the difficult nature of the numerical optimization for spatial variance parameters, the number of parameters to estimate, and no guaranteed convergence to the global optima using the EM algorithm.

PCA is closely related to factor analysis (Harman, 1976), k-mean clustering (MacQueen, 1967), or positive matrix factorization (Paatero and Tapper, 1994), which have recently been used as source apportionment or dimension reduction for exposure data prior to health analyses (Sarnat et al., 2008; Ostro et al., 2011; Zanobetti et al., 2014; Ljungman et al., 2016).

These applications, however, have been limited to time-series analysis in specific regions, without the challenge of spatial misalignment and severe missing data. Recent work by Keller et al. (2017) and Jandarov et al. (2017) has modified the traditional clustering and PCA methods, respectively, to the setting of spatially-misaligned multi-pollutant data, where the products of the dimension reduction procedure are desired to be spatially predictable. We further extend these frameworks by considering the realistic challenge of missing monitoring data. Our proposed framework essentially performs model-based imputation, which is cohesive and complementary to the spatial prediction stage. While one can impute the original data with sophisticated low-rank matrix completion techniques, which also operate based on the assumption of a latent variable structure, such methods only rely on observed measures. Therefore, if the missing patterns depend on external geographic covariates, such imputation schemes cannot recover the correct data structure.

In the literature, spatial latent variable models have been explored under the Bayesian framework. For example, Wang and Wall (2003) proposed a generalized common spatial factor model using MCMC techniques. Hogan and Tchernis (2004) formulated a Bayesian factor analysis model, which was later extended by Liu et al. (2005) to motivate a generalized spatial structural equations model, and by Zhu et al. (2005) to deal with spatiotemporal data. These rich modeling approaches have not been utilized in the setting of multi-pollutant analysis with spatial misalignment. The main goal of these models is often to explain the associations between the original variables and the underlying factors. Here the goal of an improved PCA algorithm is to obtain a lower-dimensional representation of the data in a spatially predictive way for subsequent use in spatial prediction and health regression.

In analyzing health-pollution associations under spatial misalignment, the multi-stage procedure is a common and pragmatic approach (Crouse et al., 2010; Bergen et al., 2013; Chan et al., 2015). However, it is important to be mindful of the potential implications of measurement errors and model uncertainty of the spatial prediction stage on the health inference model, a topic which has been discussed extensively in Szpiro and Paciorek (2013). These authors additionally emphasized that the spatially structured components of the co-

variates used in the health model should be included in the exposure modeling stage to guarantee a consistent estimation of the health effects. In the multi-pollutant setting with missing observations, additional stages of imputation and dimension reduction lead to more complicated layers of uncertainty. Our proposed methods eliminate the need of a separate imputation step prior to dimension reduction, as these two steps are handled simultaneously using a model-based approach. A possible alternative to the multi-stage paradigm is a unified approach where both exposure and health data are considered simultaneously in a joint model, while leveraging the factor analysis framework to perform dimension reduction. Szpiro and Paciorek (2013) point out several disadvantages of such models, including sensitivity to influential or outlying health data, vulnerability to model mis-specifications, and computational burden, especially with multi-pollutant data.

While we focus our discussion in this chapter exclusively on studies involving data on $PM_{2.5}$ and its components, our proposed method is both appropriate for other multi-pollutant studies and applicable to other fields in general where spatial misalignment necessitates an exposure modeling procedure.

Chapter 3

SPATIAL MATRIX COMPLETION

3.1 Introduction

In multi-pollutant studies, a dataset is often represented as an $(n \times p)$ matrix $\boldsymbol{X}$, in which the concentrations of p pollutants are collected at n monitoring locations. When evaluating the associations between health outcomes and exposures to air pollution, including some or all of these pollutants in a statistical model can be problematic due to correlations and potential interactions among these components. Hence, dimension reduction is often necessary to obtain a lower-dimensional representation of the original data.

Principal component analysis (PCA) (Jolliffe, 1986) is an unsupervised technique for dimension reduction that has been used in multi-pollutant analysis (Dominici et al., 2003). PCA essentially provides a mapping of the original data $\boldsymbol{X}$ to a low-rank approximation $\boldsymbol{U}\boldsymbol{V}^{\mathsf{T}}$, in which $\boldsymbol{U} \in \mathbb{R}^{n \times q}$ and $\boldsymbol{V} \in \mathbb{R}^{p \times q}$ $(q < p)$ are usually referred to as PC scores and loadings. The product $\boldsymbol{U}\boldsymbol{V}^{\mathsf{T}}$ can be considered to be the best rank-q approximation to $\boldsymbol{X}$. One can also derive this quantity by solving $\min_{\boldsymbol{W}} \left\{ \frac{1}{2} \left\| \boldsymbol{X} - \boldsymbol{W} \right\|_F^2 + \lambda \left\| \boldsymbol{W} \right\|_0 \right\}$, for a value of λ dependent on q. Here $\left\| \boldsymbol{W} \right\|_0$ denotes the number of non-zero singular values of $\boldsymbol{W}$. Another approach is to replace the L0 penalty with the nuclear norm,

$$\min_{\boldsymbol{W}} \left\{ \frac{1}{2} \left\| \boldsymbol{X} - \boldsymbol{W} \right\|_F^2 + \lambda \left\| \boldsymbol{W} \right\|_* \right\}. \tag{3.1}$$

Here $\left\| \boldsymbol{W} \right\|_*$ denotes the nuclear norm of $\boldsymbol{W}$, which is equal to the sum of its singular values. When closed-form solutions exist for both problems, the L0 penalty is not convex. When some elements of $\boldsymbol{X}$ are missing, extension to matrix completion with L0 norm is NP-hard, and existing algorithms would require time doubly exponential in the matrix's dimensions

(Candès and Recht, 2009). With missing data, the convex relaxation of the nuclear norm allows optimization via semidefinite programming, and leads to efficient low-rank matrix completion (LRMC) algorithms (Cai et al., 2010; Mazumder et al., 2010). LRMC is a powerful tool to reduce dimension while being able to utilize incomplete information.

In cohort studies on health-pollutant associations, the locations of study participants with health data and the locations of fixed monitoring sites with pollutant data are often spatially misaligned. One solution to this problem is to use an exposure modeling stage, in which accurate predictions of pollutant concentrations at new locations are often of interest. In this stage, a spatial prediction model is often used to estimate the unobserved exposures at the locations of interest (Künzli et al., 2005; Crouse et al., 2010; Bergen et al., 2013; Chan et al., 2015). When dealing with multi-pollutant data, modeling many correlated pollutant surfaces can be intractable. A potential solution is to obtain PC scores of the monitoring data, and then use a spatial prediction model to estimate these scores at cohort locations.

It is particularly challenging to predict PC scores using spatial models in situations where some constituents may exhibit very little spatial structure but dominate the PC loadings. That is, PCA does not take spatial structure or any auxiliary information into account. Without accounting for external geographic information and spatial correlations across neighboring locations, PCA may produce scores that summarize the monitoring data well but are difficult to predict at locations of interest. This challenge is exacerbated when there are often significant amounts of missing data. While LRMC can recover a low-rank structure using incomplete data, this technique does not take into account spatial information either.

A spatially predictive PCA algorithm (PredPCA) (Jandarov et al., 2017) was developed to produce scores with spatial patterns that can be subsequently predicted well at new locations. However, an additional step of imputation is required when some data is missing. Vu et al. (2019) proposed a probabilistic version of PredPCA (ProPrPCA) that is capable of performing model-based imputation and dimension reduction at the same time. In their current forms, however, neither PredPCA nor ProPrPCA can estimate all desired PCs simultaneously. The first PC is estimated using the original data matrix. The next PC

is then estimated based on a residual matrix, which is obtained by subtracting the rank-1 approximation constructed by the first PC loading and score. This procedure is repeated until a desired number of PCs has been reached. Subsequently, these algorithms may not guarantee orthogonality among the estimated PC scores. Orthogonality is often preferred, as orthogonal PCs cane be considered to be uncorrelated, unlike the original variables.

In this chapter, our objective is to develop a practical and computationally feasible version of PCA based on LRMC that can (i) accommodate complex spatial missing patterns; (ii) lead to PCs that can be accurately predicted at unmeasured locations; and (iii) obtain all PCs simultaneously. We formulate this as a convex optimization problem, and derive a straightforward algorithm to solve it using proximal gradient descent.

3.2 Review of low-rank matrix completion (LRMC)

3.2.1 Problem formulation

The optimization problem in (3.1) can be referred to as a low-rank matrix approximation, or the "fully observed" version of low-rank matrix completion (LRMC). When some entries of $\boldsymbol{X}$ are missing, the low-rank structure of $\boldsymbol{X}$ can be recovered by minimizing the residuals over only the observed indices,

$$\min_{\boldsymbol{W}} \ \frac{1}{2}\left\{\left\|P_{\Omega}(\boldsymbol{X}) - P_{\Omega}(\boldsymbol{W})\right\|_F^2 + \lambda\left\|\boldsymbol{W}\right\|_*\right\}, \tag{3.2}$$

where Ω denotes the set of observed indices in $\boldsymbol{X}$, and $[P_{\Omega}(\boldsymbol{X})]_{ij} = X_{ij}$ if $(i,j) \in \Omega$ and zero otherwise.

3.2.2 Optimization

Mazumder et al. (2010) prove that the "fully observed" problem in (3.1) has a closed-form solution $\hat{\boldsymbol{W}}$ that uses the *soft-thresholding* operator,

$$\hat{\boldsymbol{W}} = \tilde{\boldsymbol{U}} S_{\lambda}(\boldsymbol{D}) \tilde{\boldsymbol{V}}^{\mathsf{T}}.$$

Here $\tilde{\boldsymbol{U}}\boldsymbol{D}\tilde{\boldsymbol{V}}^{\mathsf{T}}$ is the singular value decomposition (SVD) of $\boldsymbol{X}$, with $\boldsymbol{D} = \text{diag}\{\sigma_1, ..., \sigma_r\}$, with σ_i being the $i-$th largest singular value of $\boldsymbol{X}$, and r being the column rank of $\boldsymbol{X}$. We assume that the columns of $\boldsymbol{X}$ have been properly centered. The soft-thresholding operator (Donoho et al., 1995) is defined as $S_\lambda(\boldsymbol{D}) = \text{diag}\{(\sigma_1 - \lambda)_+, ..., (\sigma_r - \lambda)_+\}$, where $t_+ = \max(0, t)$. This result is closely related to PCA, in that, if PCA were to be applied onto $\boldsymbol{X}$, $\tilde{\boldsymbol{V}}$ would be returned as the loadings.

Mazumder et al. (2010) propose an algorithm that uses proximal gradient descent (Rockafellar, 1976) to solve (3.2), which is given below. Proof of the proximal gradient descent and derivation of this LRMC algorithm are given in Appendix B.2. The algorithm consists of two major steps: a gradient descent update, and solving the proximal problem to (3.2). In particular, the gradient update is simply filling missing entries with the corresponding entries of the current estimate. The proximal problem turns out to be exactly the low-rank approximation problem (3.1), which has a closed-form solution using soft-thresholding.

Algorithm 1: LRMC adapted from Mazumder et al. (2010)

Input $\boldsymbol{X}$, q, λ, and t_{max}
  Initialize $\boldsymbol{W}^{(0)} = \boldsymbol{0}$, $t = 1$
    while not converged or $t < t_{max}$ **do**
      $\tilde{\boldsymbol{W}}^{(t)} \leftarrow P_\Omega(\boldsymbol{X}) + P_\Omega^\perp(\boldsymbol{W}^{(t)})$, where $P_\Omega^\perp(\boldsymbol{W}^{(t)}) = \boldsymbol{W}^{(t)} - P_\Omega(\boldsymbol{W}^{(t)})$
      $\boldsymbol{W}^{(t+1)} \leftarrow \tilde{\boldsymbol{U}} S_\lambda(\tilde{\boldsymbol{D}}) \tilde{\boldsymbol{V}}^{\mathsf{T}}$, where $\tilde{\boldsymbol{U}}\tilde{\boldsymbol{D}}\tilde{\boldsymbol{V}}^{\mathsf{T}}$ is the SVD of $\tilde{\boldsymbol{W}}^{(t)}$
      $t \leftarrow t + 1$
    end while
Output $\hat{\boldsymbol{W}} = \boldsymbol{W}^{(t)}$, $\hat{\boldsymbol{X}} = P_\Omega(\boldsymbol{X}) + P_\Omega^\perp(\hat{\boldsymbol{W}})$

3.3 The spatial matrix completion problem

3.3.1 Proposed optimization problem

The LRMC algorithm is a powerful tool to recover a low-rank structure of the data even though only a sampling of the entries is observed. Once the missing entries are filled, the PC scores can be easily obtained by projecting the imputed data $\hat{\boldsymbol{X}}$ onto the direction of its first

q right singular vectors. However, under spatial misalignment, the ultimate goal is not merely to summarize the pollutant data well. It is actually more important to produce accurate predictions of these PC scores at cohort locations where pollution data is unavailable. A multi-stage procedure is often employed in these cohort studies: 1) imputation to fill in missing elements of the data matrix, 2) dimension reduction to obtain lower-dimensional representations (scores) of the data, and 3) spatial prediction to estimate these scores at locations of interest.

In the second stage with dimension reduction, ideally we would like to identify principal directions such that the resulting PC scores would retain important characteristics and spatial structure. Having these spatial patterns, the PC scores could be predicted well at new locations in the spatial prediction stage. As a result, we propose the following convex optimization problem for the "fully-observed" scenario,

$$\min_{\boldsymbol{M}} \left\{ \frac{1}{2} \left\| \boldsymbol{X} - \boldsymbol{Z}\boldsymbol{M} \right\|_F^2 + \lambda \left\| \boldsymbol{Z}\boldsymbol{M} \right\|_* \right\}, \tag{3.3}$$

where $\boldsymbol{Z} \in \mathbb{R}^{n \times k}$ contains geographic covariates used in later prediction stage and thin-plate spline basis functions. The basis functions are included to capture any underlying spatial patterns that may not have been explained by other covariates.

When $\boldsymbol{X}$ has missing entries, we propose the following optimization problem

$$\min_{\boldsymbol{M}} \frac{1}{2} \left\{ \left\| P_\Omega(\boldsymbol{X}) - P_\Omega(\boldsymbol{Z}\boldsymbol{M}) \right\|_F^2 + \lambda \left\| \boldsymbol{Z}\boldsymbol{M} \right\|_* \right\}, \tag{3.4}$$

3.3.2 Optimization

First, we look at the complete data scenario and the proposed problem in (3.3). While $\boldsymbol{M}$ is the unknown quantity of the objective function, it is important to keep in mind that we are more interested in the quantity $\hat{\boldsymbol{W}} = \boldsymbol{Z}\hat{\boldsymbol{M}}$ where $\hat{\boldsymbol{M}}$ is the optimal solution for (3.3). This quantity is the low-rank approximation of $\boldsymbol{X}$. We give the closed-form solution of $\hat{\boldsymbol{W}}$ in the following lemma, with the detailed proof in Appendix B.1.

Lemma. *If* $(\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z})^{-1}$ *exists, then the approximation* $\hat{\boldsymbol{W}} = \boldsymbol{Z}\hat{\boldsymbol{M}}$ *of* $\boldsymbol{X}$*, where* $\hat{\boldsymbol{M}}$ *is the optimal solution for (3.3), has a closed-form expression,* $\hat{\boldsymbol{W}} = \tilde{\boldsymbol{U}}S_\lambda(\boldsymbol{D})\tilde{\boldsymbol{V}}^\mathsf{T}$*, where* $\tilde{\boldsymbol{U}}\boldsymbol{D}\tilde{\boldsymbol{V}}^\mathsf{T}$ *is SVD of* $\tilde{\boldsymbol{X}} = \boldsymbol{Z}(\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z})^{-1}\boldsymbol{Z}^\mathsf{T}\boldsymbol{X}$*.*

When $\boldsymbol{X}$ has missing entries, we prove that $\hat{\boldsymbol{W}} = \boldsymbol{Z}\hat{\boldsymbol{M}}$, where $\hat{\boldsymbol{M}}$ is the optimizer of (3.4), can be derived via a proximal algorithm similar to the LRMC algorithm. The steps are laid out below. Our algorithm is similar to the LRMC algorithm, with the insertion of an additional projection step involving $\boldsymbol{Z}$. The full derivation and proof are given in Appendix B.3. We refer to this as the **S**patial **M**atrix **C**ompletion (SMC) algorithm.

Algorithm 2: Spatial matrix completion (SMC)

Input $\boldsymbol{X}$, $\boldsymbol{Z}$, q, λ, and t_{max}
  Initialize $\boldsymbol{W}^{(0)} = \boldsymbol{0}$, $t = 1$
    while not converged or $t < t_{max}$ **do**
      $\breve{\boldsymbol{W}}^{(t)} \leftarrow P_\Omega(\boldsymbol{X}) + P_\Omega^\perp(\boldsymbol{W}^{(t)})$
      $\tilde{\boldsymbol{W}}^{(t)} \leftarrow \boldsymbol{Z}(\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z})^{-1}\boldsymbol{Z}^\mathsf{T}\breve{\boldsymbol{W}}^{(t)}$
      $\boldsymbol{W}^{(t+1)} \leftarrow \tilde{\boldsymbol{U}}S_\lambda(\tilde{\boldsymbol{D}})\tilde{\boldsymbol{V}}^\mathsf{T}$, where $\tilde{\boldsymbol{U}}\tilde{\boldsymbol{D}}\tilde{\boldsymbol{V}}^\mathsf{T}$ is the SVD of $\tilde{\boldsymbol{W}}^{(t)}$
      $t \leftarrow t + 1$
    end while
Output $\hat{\boldsymbol{W}} = \boldsymbol{W}^{(t)}$, $\hat{\boldsymbol{X}} = P_\Omega(\boldsymbol{X}) + P_\Omega^\perp(\hat{\boldsymbol{W}})$

3.3.3 Connection with existing methods

Another direct approach to induce spatial patterns into the PC scores was the PredPCA algorithm proposed by Jandarov et al. (2017). The PredPCA algorithm also employs the same matrix $\boldsymbol{Z}$ of covariates and spline basis functions in its objective function. The algorithm uses a biconvex formulation of PCA where the PCs are estimated sequentially.

We take a closer look into the objective function of PredPCA,

$$\min_{\boldsymbol{\alpha},\boldsymbol{v}} \left\| \boldsymbol{X} - \left(\frac{\boldsymbol{Z}\boldsymbol{\alpha}}{\|\boldsymbol{Z}\boldsymbol{\alpha}\|_2} \right) \boldsymbol{v}^\mathsf{T} \right\|_F^2,$$

where $\boldsymbol{\alpha} \in \mathbb{R}^k$ and $\boldsymbol{v} \in \mathbb{R}^p$. Here the algorithm estimates one PC at a time. The quantity $\boldsymbol{Z\alpha}/\|\boldsymbol{Z\alpha}\|_2$ plays the role of the PC score, while $\boldsymbol{v}$ is the loading. PredPCA directly imposes a spatial structure on the score via $\boldsymbol{Z}$. By doing so, PredPCA essentially aims to recover the best rank-1 approximation of $\boldsymbol{X}$ that has spatial structure embedded within the left singular vectors. Thus the objective function of PredPCA can be rewritten as an L0 problem with spatial constraints. Heuristically, this is similar to what SMC aims to achieve. The fundamental difference is that SMC utilizes a nuclear-norm penalty, while PredPCA resembles our spatial optimization problem with an L0 penalty. When some data are missing, such reformulated version of PredPCA can potentially be solved using hard-thresholding (Mazumder et al., 2010). However, unlike SMC, the lack of convexity means that convergence to an optimal solution and corresponding theoretical properties are not guaranteed.

In its current form, the sequential estimation and the lack of a mechanism to handle missing data are rather unsatisfying. The probabilistic version, ProPrPCA (Vu et al., 2019), provides a better alternative to PredPCA when missing data are present, as imputation and dimension reduction are handled simultaneously. However, ProPrPCA can only estimate one PC at a time, and also requires longer computational time overall.

3.4 Simulations

3.4.1 Simulation setup and evaluation

We reproduce the simulations used in Vu et al. (2019) to have a solid comparison between the existing and our proposed methods. Their performance is evaluated based on the first two PCs, i.e. $q = 2$. The simulations include $p = 15$ pollutant surfaces generated on a dense 100×100 grid. The data are generated based on on three underlying scores, $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, and $\boldsymbol{u}_3$. In terms of spatial predictability, i.e. how well the score can be predicted at new locations, $\boldsymbol{u}_1$ is the highest and $\boldsymbol{u}_2$ is moderate, while $\boldsymbol{u}_3$ is purely noise. There are two scenarios with different level of variance contribution from these scores. In scenario 1, the orders of spatial predictability and variance contribution are the same. All methods are expected to

perform similarly with complete data. In scenario 2, $\boldsymbol{u}_3$ has the highest variance contribution, followed closely by $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$. For this scenario, performance of PCA is anticipated to be poor as PCA will pick up linear combinations of $\boldsymbol{u}_3$ and $\boldsymbol{u}_1$ for the first two PCs. The other methods will identify combinations of $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ instead. We consider various settings, in which the training data is either complete, missing completely at random (MCAR), or missing at random (MAR), where the missing patterns are associated with the generated covariates. Further details are described in Vu et al. (2019) and the previous chapter.

In each of the 1,000 simulations, 400 training locations and 100 test locations are chosen at random. We perform the multi-stage procedure with these dimension reduction techniques, and evaluate them based on the predictive performance in the spatial prediction stage with universal kriging. Due to the stepwise nature of the exposure modeling procedure, evaluating the performance of the four competing methods (PCA, PredPCA, ProPrPCA, and SMC) is complicated because of downstream analysis. Similar to Vu et al. (2019), the simplest way is to compare the predictive performance after spatial prediction at test locations. That is, we determine how much of an agreement between what we find from a spatial prediction model ("predicted" scores) and what we would have got if we projected the unknown test data onto the direction of the loadings estimated from the training data ("known" scores).

Specifically, in notation, we derive the loadings $\boldsymbol{V}_{train}$ based on the training data $\boldsymbol{X}_{train}$, and calculate the PC scores $\boldsymbol{U}_{train}$ by projecting $\boldsymbol{X}_{train}$ onto the column space of $\boldsymbol{V}_{train}$. We then use $\boldsymbol{U}_{train}$ and relevant covariates to predict $\hat{\boldsymbol{U}}_{test}$ at test locations. When training data are complete, the competing methods include PCA, PredPCA, ProPrPCA, and the "fully-observed" version of SMC. With missing data, LRMC is used in the imputation step prior to PCA or PredPCA. The evaluation metric of interest is the prediction R^2 values that reflect the correlations between the columns of the predicted scores $\hat{\boldsymbol{U}}_{test}$ and the known scores $\boldsymbol{U}_{test}$, defined by projecting the test data $\boldsymbol{X}_{test}$ onto the directions of $\boldsymbol{V}_{train}$.

We also conduct an additional set of high-dimensional simulations. In these simulations, we generate 20 pollutants based on four underlying scores with equal variance contribution and non-sparse loadings. The full setups are described in Appendix B.4. Finally, we note

that we only use the spline version of ProPrPCA throughout this chapter.

3.4.2 Results

Table 3.1: The median prediction R^2's across 1,000 simulations for scenario 1. Under missing data scenarios, LRMC is used prior to either PCA or PredPCA.

PC1	Complete	MCAR 35%	MAR
PCA	0.83	0.80	0.61
PredPCA	0.84	0.81	0.63
ProPrPCA	0.84	0.83	0.69
SMC	0.84	0.83	0.74

PC2	Complete	MCAR 35%	MAR
PCA	0.60	0.58	0.67
PredPCA	0.60	0.58	0.68
ProPrPCA	0.60	0.60	0.68
SMC	0.60	0.60	0.63

Table 3.1 shows the median R^2 for scenario 1. As expected, all methods perform equally well when the training data is complete. As discussed in Vu et al. (2019), under MAR scenario, data among the first five pollutants are more likely to be missing at locations with extreme geographic covariate values. This setup effectively lowers the variability contributed by $\boldsymbol{u}_1$. As the predictive methods aim to balance out the trade-off between data representativeness and spatial predictability, linear combinations for PC1 will be obtained with more weight from $\boldsymbol{u}_2$ for PC1, while PC2 will have more weight from $\boldsymbol{u}_1$ than before. This leads to the decreases in median R^2's of PC1 for all methods but slight increases for PC2. Although the overall performance decreases, SMC results in the best median R^2 for PC1 in MAR scenario (0.74), followed by ProPrPCA (0.69). This is obtained in exchange for a better performance for PC2 for ProPrPCA (0.68) compared to SMC (0.63).

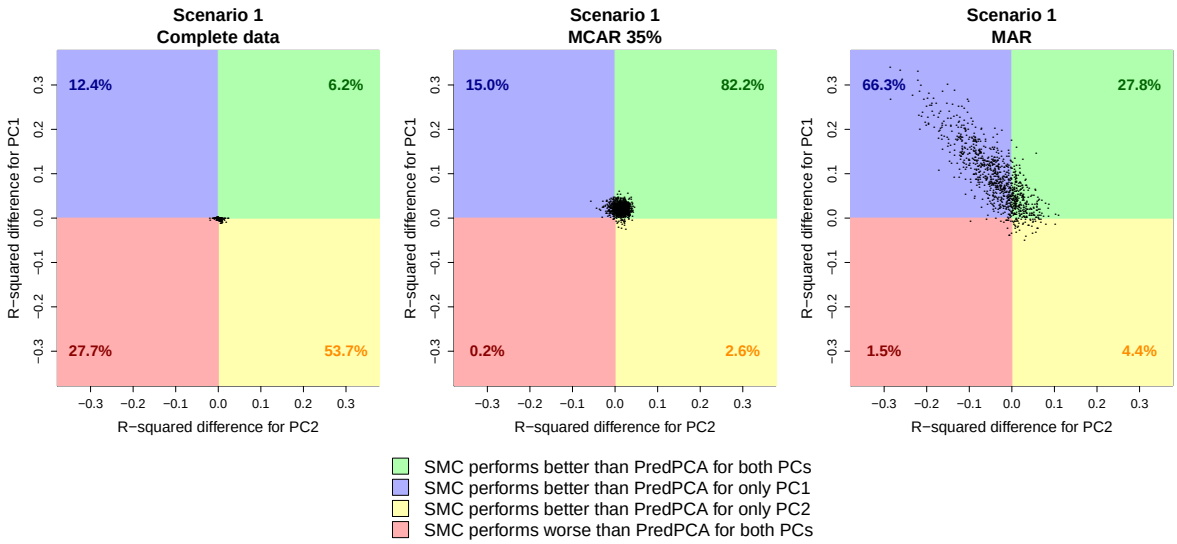

Figure 3.1: Difference in R^2 between SMC and PredPCA for scenario 1. Each dot represents result from one simulation. Percentages indicate the proportion out of 1,000 simulations.

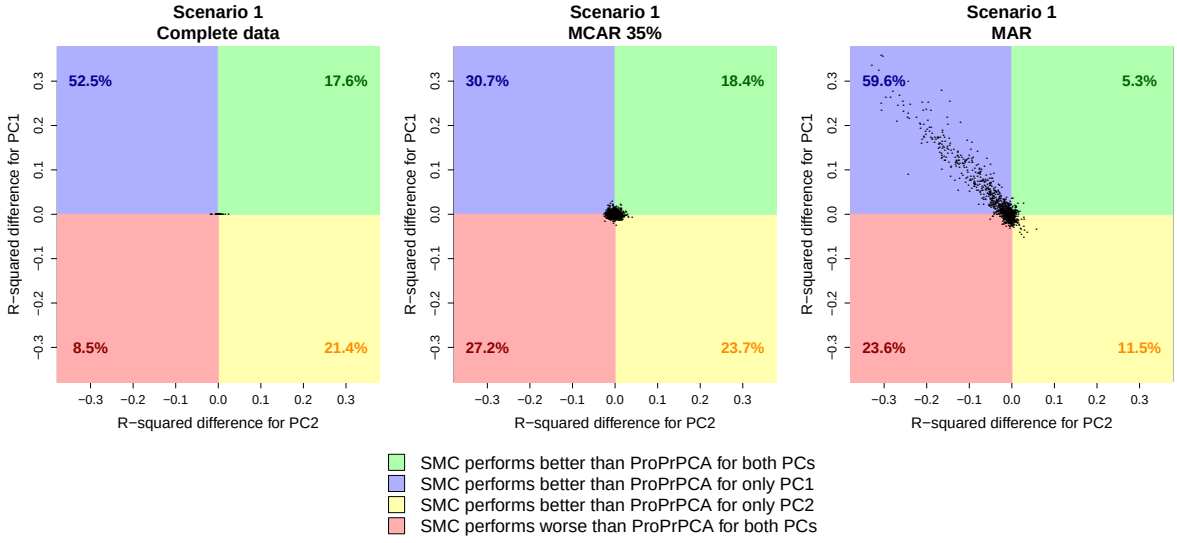

Figure 3.2: Difference in R^2 between SMC and ProPrPCA for scenario 1. Each dot represents result from one simulation. Percentages indicate the proportion out of 1,000 simulations.

The differences between SMC and PredPCA are further examined in Figure 3.1. With

complete data, the discrepancy between the two methods is negligible. Under MCAR 35%, SMC outperforms PredPCA for 82.2% of the 1,000 simulations. The differences are the most prominent under MAR. SMC performs worse than PredPCA for both PCs in only 1.9% of the simulations. This result closely mirrors the comparison between ProPrPCA and PredPCA discussed in Vu et al. (2019).

Figure 3.2 shows additional comparison between SMC and ProPrPCA. The two methods perform equally well throughout all data scenarios. ProPrPCA shows some slight advantage by beating SMC for both PCs in 23.6% of the time (bottom-left quadrant), compared to just 5.3% of the time when SMC is better for both PCs (top-right quadrant). However, the magnitudes of differences in these regions are relatively small. In 59.6% of the simulations, SMC is better in predicting PC1, with the magnitude of difference ranging up to 0.3.

Table 3.2: The median prediction R^2's across 1,000 simulations for scenario 2. Under missing data scenarios, LRMC is used prior to either PCA or PredPCA.

PC1	Complete	MCAR 35%	MAR
PCA	0.01	0.01	0.00
PredPCA	0.81	0.78	0.63
ProPrPCA	0.81	0.80	0.72
SMC	0.81	0.80	0.71

PC2	Complete	MCAR 35%	MAR
PCA	0.78	0.74	0.60
PredPCA	0.56	0.54	0.62
ProPrPCA	0.56	0.56	0.59
SMC	0.56	0.55	0.58

Corresponding results for scenario 2 are shown in Table 3.2, Figures 3.3 and 3.4. As expected, PCA tends to recover linear combinations of $\boldsymbol{u}_3$ and $\boldsymbol{u}_1$ for the first two PCs, as these two underlying scores contribute the most variability to the data. Hence, the predictive performance is very poor for PC1. Other methods are able to identify the underlying scores of interest, and the overall observations are similar to scenario 1.

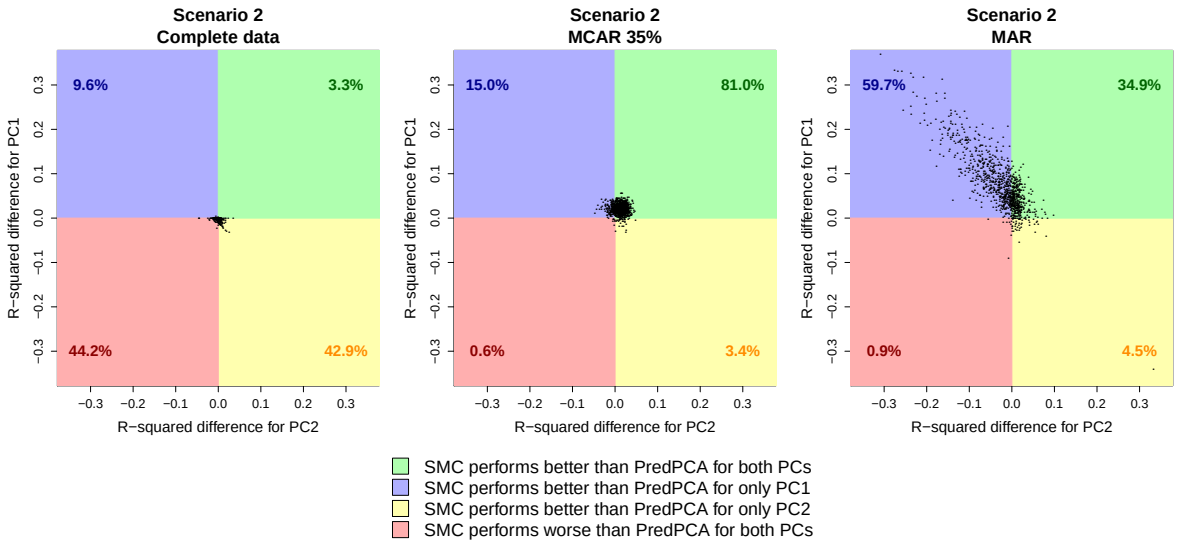

Figure 3.3: Difference in R^2 between SMC and PredPCA for scenario 2. Each dot represents result from one simulation. Percentages indicate the proportion out of 1,000 simulations.

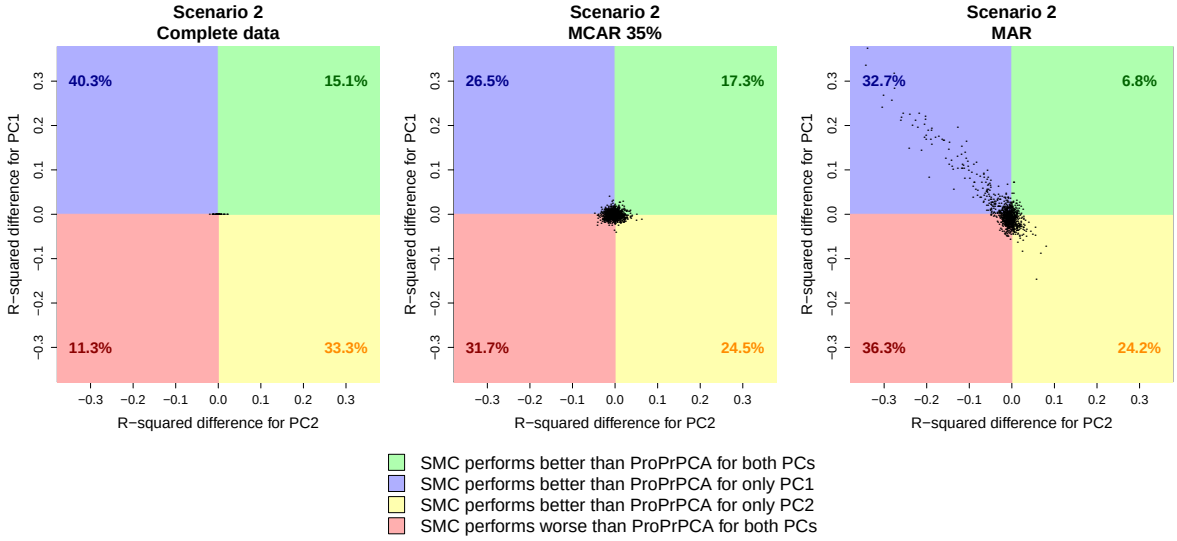

Figure 3.4: Difference in R^2 between SMC and ProPrPCA for scenario 2. Each dot represents result from one simulation. Percentages indicate the proportion out of 1,000 simulations.

Finally, Figure 3.5 compares the computation time between SMC and ProPrPCA. The

burden increases as the training sample size gets larger or when data is missing. Overall, SMC is more efficient as its computation time is small compared to ProPrPCA.

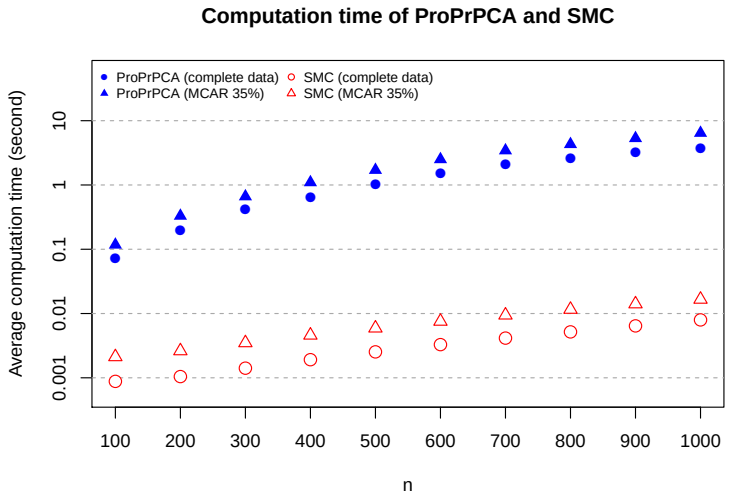

Figure 3.5: Computation time (average over 1,000 simulations under scenario 1) of ProPrPCA and SMC with complete and MCAR 35% missing data by training sample size.

We also conduct an additional set of high-dimensional simulations, with 20 pollutants generated from four underlying scores with equal variance contribution and non-sparse loadings. The observed trends shown in Appendix B.4 are similar to the results above.

3.5 Data application

3.5.1 Data source

Next we apply SMC to the same dataset used in Vu et al. (2019). The data were collected nationally by the Chemical Speciation Network (CSN), a subnetwork of the Air Quality System network of monitors managed by the Environmental Protection Agency. Data are available for 21 components of $PM_{2.5}$: elemental carbon (EC), organic carbon (OC), sulfate ion (SO_4^{2-}), nitrate ion (NO_3^-), aluminum (Al), arsenic (As), bromine (Br), cadmium (Cd), calcium (Ca), chromium (Cr), copper (Cu), iron (Fe), potassium (K), magnesium (MN),

sodium (Na), sulfur (S), silicon (Si), selenium (Se), nickel (Ni), vanadium (V), and zinc (Zn). Geographic covariates are provided through the Exposure Assessment Core Database by the MESA Air team at the University of Washington. Data processing is described with details in Vu et al. (2019).

For consistency with previous literature (Vu et al., 2019; Jandarov et al., 2017; Keller et al., 2017), we use the 2010 CSN data. This dataset consists of 221 CSN sites, only 130 of which have complete data on all 21 components, with the overall missing level being 32.1%. We also look into similar data collected in 2011. This dataset includes 208 CSN sites, only 128 of which have complete data, and the overall amount of missing data is 30.1%.

3.5.2 Methods

For both the 2010 and 2011 CSN data, we first apply the four methods (PCA, PredPCA, ProPrPCA, and SMC) on the sites with complete data on all $PM_{2.5}$ components (the "complete" set). We then apply these methods on the "full" dataset, where LRMC is applied prior to PCA and PredPCA. For all methods, we examine only the first three PCs. The first aim is to evaluate how the pollutant profile varies across dimension reduction techniques. That is, we assess how the estimated loadings and corresponding PC scores change under different methods and whether all sites are utilized.

Next, we aim to evaluate the predictive performance after dimension reduction and spatial prediction. Here we conduct leave-one-site-out cross-validation. Each time, we leave out one site among the set of complete sites as test data, while performing dimension reduction and building spatial prediction model based on the training data. For consistency, we use the same universal kriging model with exponential covariance structure for the spatial prediction step. The training data may consist of either the remaining complete sites (the "complete" training data), or all remaining sites (the "full" training data).

3.5.3 The multi-pollutant profile

For the 2010 CSN data, Figure 3.6 shows the estimated loadings using the complete set (top-left panel) and the full set (top-right panel), as well as the spatial distributions of the corresponding scores for the first PC. Similar to Vu et al. (2019), the first PCs obtained from all methods on complete data are similar across PredPCA, ProPrPCA, and SMC, with highly positive weights on SO_4^{2-} and S. While the maps of PC scores produced by PCA look similar to others, loadings obtained by PCA are fundamentally different than the rest. These have weaker positive weights on SO_4^{2-} and S, and strongly negative weights on additional metal elements, such as Cu, Fe, etc.

Figure 3.7 shows the results for the PC that has strongly positive weights of Na, Ni, and V. With complete data, this corresponds to the third PC obtained by all methods. When all sites are included, this corresponds to the third PC obtained by PCA, but the second PC produced by PredPCA, ProPrPCA, and SMC. ProPrPCA identifies strongly negative loadings on Zn and NO_3^- when using all available data. Thus, while the maps are almost identical, only ProPrPCA successfully highlights some relevant coastal locations in the southeast region with high level of residual oil combustion and marine aerosol (Thurston et al., 2011; Kotchenruther, 2017). Note that the loadings produced by SMC are more similar to those derived by PredPCA.

Figure 3.8 displays the results for the remaining PCs, which have highly positive weights on NO_3^- and Zn. Again, SMC produces loadings that mirrors those obtained by PredPCA. Results of ProPrPCA are strongly positive in Ni, V, and negative in Al, Si, with greater magnitude than both PredPCA and SMC.

Overall, for 2010 CSN data, while the spatial distributions of the scores are similar, the interpretation of these results are scientifically different between PCA and the other three methods. Loading results are substantially different when using only complete sites compared to when using all available sites. SMC and PredPCA results are more similar to each other.

Results for the 2011 CSN data are given in Appendix B.4. For this dataset, the results

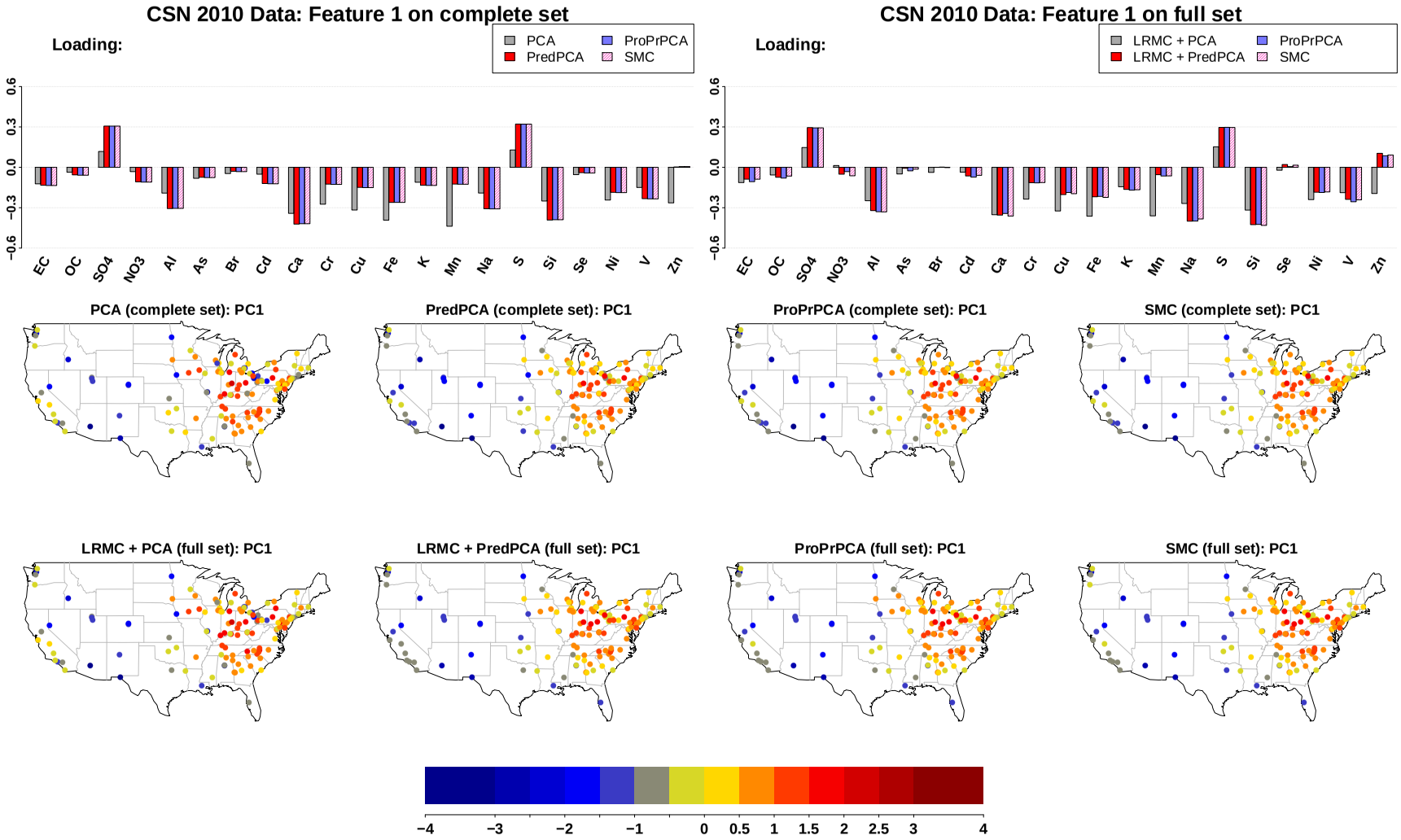

Figure 3.6: Estimated loadings for feature with highly positive weights on SO_4^{2-} and S, and corresponding scores, obtained from different PCA algorithms (PCA, PredPCA, ProPrPCA, and SMC) applied to 2010 CSN data: the complete set (130 sites with complete data) or the full set (all 221 available sites). The top panel of bar plots display the loadings obtained from dimension reduction techniques. The bottom two panels of maps illustrate the scores corresponding to the estimated loadings.

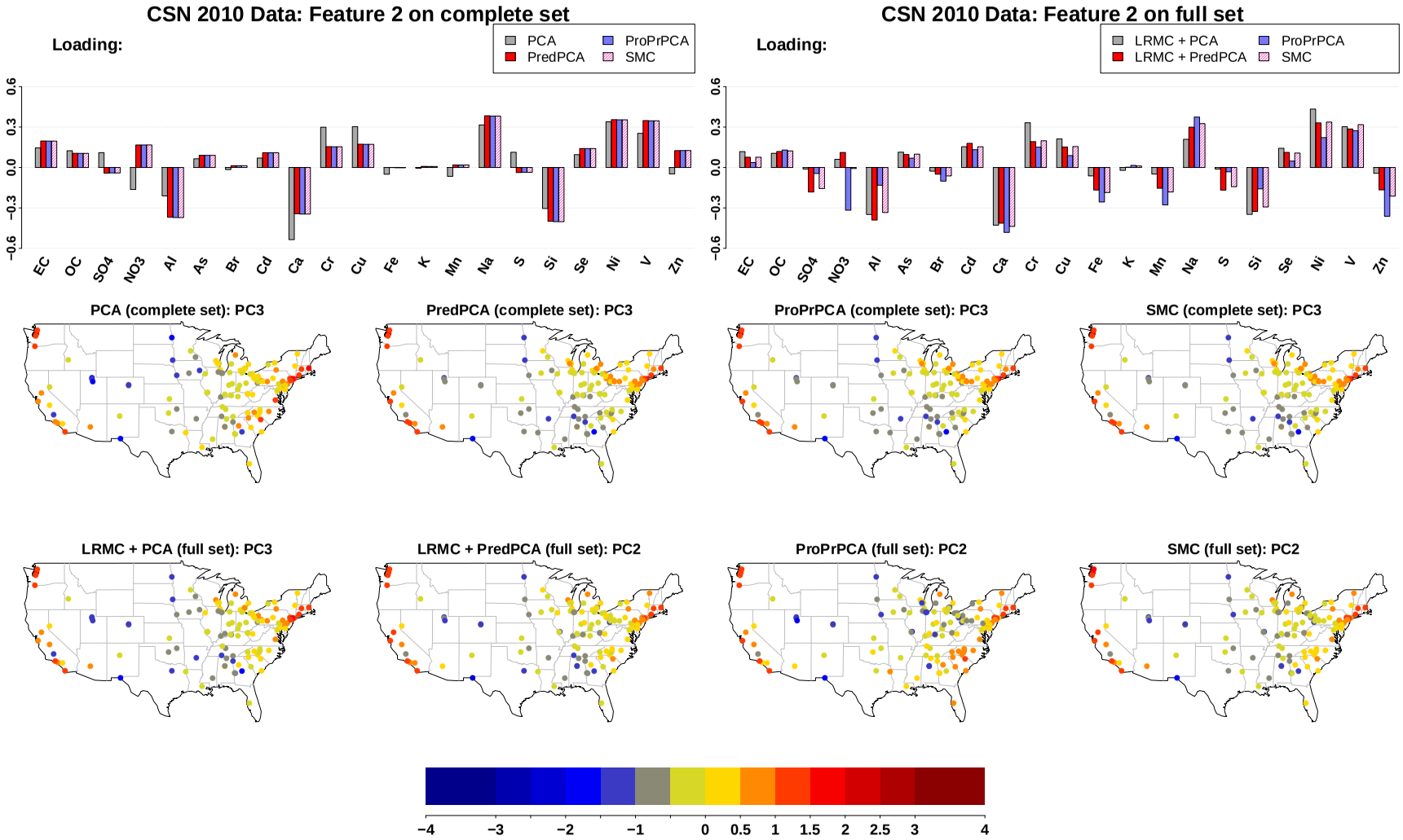

Figure 3.7: Estimated loadings for feature with highly positive weights on Na, Ni, and V, and corresponding scores, obtained from different PCA algorithms (PCA, PredPCA, ProPrPCA, and SMC) applied to 2010 CSN data: the complete set (130 sites with complete data) or the full set (all 221 available sites). The top panel of bar plots display the loadings obtained from dimension reduction techniques. The bottom two panels of maps illustrate the scores corresponding to the estimated loadings.

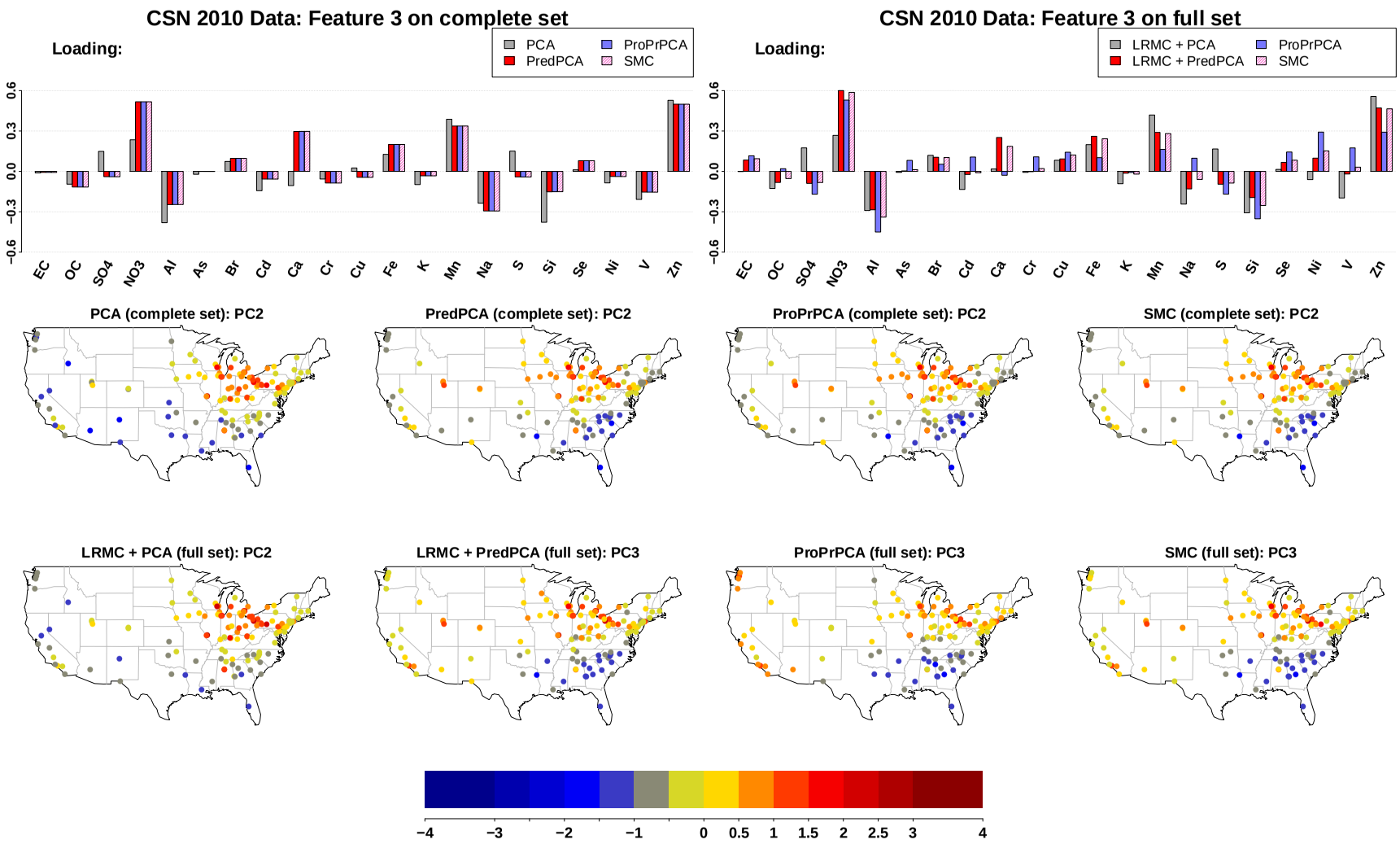

Figure 3.8: Estimated loadings for feature with highly positive weights on NO_3^- and Zn, and corresponding scores, obtained from different PCA algorithms (PCA, PredPCA, ProPrPCA, and SMC) applied to 2010 CSN data: the complete set (130 sites with complete data) or the full set (all 221 available sites). The top panel of bar plots display the loadings obtained from dimension reduction techniques. The bottom two panels of maps illustrate the scores corresponding to the estimated loadings.

are more consistent across PredPCA, ProPrPCA, and SMC. Note that the two datasets have slightly different numbers of sites and amounts of missing data (32.1% for 2010 versus 30.1% for 2011).

3.5.4 Cross-validation results

Finally, we evaluate the predictive performance via cross-validation. The results for 2010 CSN data are shown in Table 3.3. When using only complete sites for training data, PCA has acceptable performance for PC2 and PC3, however, gives poor results for PC1 ($R^2 = 0.24$). Meanwhile, the other three methods give the same results and better R^2 for both PC1 and PC3.

Table 3.3: Prediction R^2's from leave-one-site-out cross-validation on 2010 CSN data, using training data with either complete sites only or all available data. LRMC is used prior to PCA or PredPCA when training set has missing data. Only sites with complete $PM_{2.5}$ component data are used as test data.

Training data with complete sites	PC1	PC2	PC3
PCA	0.24	0.51	0.51
PredPCA	0.52	0.44	0.62
ProPrPCA	0.52	0.44	0.62
SMC	0.52	0.44	0.62

Training data with all available sites	PC1	PC2	PC3
PCA	0.32	0.44	0.52
PredPCA	0.54	0.53	0.45
ProPrPCA	0.57	0.35	0.69
SMC	0.57	0.51	0.48

When training data consist of all remaining sites, there is a slight improvement in PC1, coupled with a decrease in PC2 for PCA. For PredPCA and SMC, there is a slight trade-off in performance between PC2 (increase) and PC3 (decrease). SMC has the same performance as ProPrPCA for PC1, but is more similar to PredPCA overall. ProPrPCA has a substantially higher R^2 for PC3 compared to the rest, but worse performance for PC2. These results may

be due to the switching between PC2 and PC3 observed in the pollutant profile when using all sites compare to including just complete sites. In addition, the loadings estimated by ProPrPCA for these two PCs both have strong components of NO_3^- and Zn, but in opposite directions.

Results for the 2011 CSN data are shown in Table 3.4. The overall performance for 2011 data is better than 2010 across all methods. When using all available sites, the performance of ProPrPCA improves most noticeably in PC2, without a large trade-off in other PCs. While PredPCA has worse performance in both PC1 and PC3, the performance of SMC is closer to ProPrPCA in this dataset.

Table 3.4: Prediction R^2's from leave-one-site-out cross-validation on 2011 CSN data, using training data with either complete sites only or all available data. LRMC is used prior to PCA or PredPCA when training set has missing data. Only sites with complete $PM_{2.5}$ component data are used as test data.

Training data with complete sites	PC1	PC2	PC3
PCA	0.59	0.38	0.66
PredPCA	0.71	0.59	0.56
ProPrPCA	0.71	0.53	0.60
SMC	0.71	0.60	0.58

Training data with all available sites	PC1	PC2	PC3
PCA	0.55	0.44	0.61
PredPCA	0.66	0.65	0.51
ProPrPCA	0.72	0.66	0.57
SMC	0.70	0.65	0.46

Note that for a fair comparison across all methods, we use the same pre-specified spatial information encoded in $\boldsymbol{Z}$ to characterize the spatial variability across all PCs. In addition, across the board we use the same universal kriging model with an exponential covariance structure in the spatial prediction stage. While these choices may simplify the comparisons, using the same $\boldsymbol{Z}$ matrix to model all PCs may not be effective, and the covariance structure might have not been correctly specified in these data applications. Potential solutions, which

are beyond the focus of this chapter, include an adaptive selection of features to be included in $\boldsymbol{Z}$ (Bose et al., 2018), or more sophisticated non-stationary models for spatial prediction instead of universal kriging.

3.6 Discussion

In this chapter, we focus on problems arising in health-pollution cohort studies, in which multi-pollutant data is often spatially misaligned and has a large number of missing observations. The ultimate goal is to develop a dimension reduction technique that is similar to PCA but able to produce PC scores that can be accurately predicted at locations of interest. The scientific motivation includes the ability to characterize the pollutant profile across locations, and to use estimated PC scores as effect modifiers for the health-pollution associations of interest. For example, many studies on fine particulate matter ($PM_{2.5}$) have shown evidence that the associations between health outcomes and $PM_{2.5}$ total mass can be significantly modified by the $PM_{2.5}$ chemical composition (Krall et al., 2013; Zanobetti et al., 2014; Kioumourtzoglou et al., 2015; Wang et al., 2017; Keller et al., 2018).

We formulate a convex optimization problem based on the existing idea of LRMC with nuclear-norm penalization. We show that a closed-form solution exists when the original data is fully observed. In addition, we also derive a proximal algorithm to solve the problem when some elements of the data are missing. In simulations, we evaluate the performance of our proposed SMC algorithm against PCA, PredPCA, and ProPrPCA. SMC outperforms both PCA and PredPCA under various missing data scenarios. The performance of SMC is generally similar to ProPrPCA but it is more computationally efficient.

A slight complication of SMC compared to other methods is that it requires the penalty parameter λ. In our current algorithm, the choice of λ is based on a small grid search to reach the desired rank q of the low-rank approximation. However, the grid search does not have a major impact on computation time, as shown in simulation results. Computation time can also be shortened using warm starts (Mazumder et al., 2010).

Similar to ProPrPCA, SMC can produce PC scores with spatial patterns and impute for

missing data with considerations of external geographic and spatial information, as illustrated in Table 3.5. SMC is also able to estimate all PCs simultaneously, whereas the ProPrPCA model obtains the PCs sequentially. Under SMC, estimated PCs are guaranteed to be orthogonal, and thus considered to be uncorrelated, which is one of the desirable properties of PCA. It is important to note that for ProPrPCA, when data is missing, the parameter estimation and data imputation are separate. The loadings are estimated based only on the observed data. The data is then imputed with consideration of $\boldsymbol{Z}$, and projected onto the directions of the loadings to derive the PC scores. One can practically use different $\boldsymbol{Z}$ matrices for the estimation and imputation procedures. This can potentially be more beneficial and more accurate, particularly when there is reasonable evidence to believe that the missing mechanism only depends on a subset of covariates and spline terms included in $\boldsymbol{Z}$. Meanwhile, imputation and dimension reduction are essentially intertwined in the SMC algorithm, and there is no flexibility in modifying information used for imputing data only. The simulations show that the approach of ProPrPCA consistently produces better results, although SMC follows very closely. The SMC algorithm offers a faster, more compact and elegant alternative to ProPrPCA.

Table 3.5: Evaluation of different approaches with imputation and dimension reduction

	LRMC + PCA	LRMC + PredPCA	ProPrPCA	SMC
Induce spatial patterns in PC scores		✓	✓	✓
Impute data with spatial consideration			✓	✓
Estimate all PCs simultaneously	✓			✓

In its current form proposed by Jandarov et al. (2017), an imputation step is required prior to PredPCA. LRMC is a useful method to fill in the missing data, but it does not account for spatial structure while imputing the data. Using LRMC prior to PredPCA may distort the underlying structure of the data even before dimension reduction, and thus worsen the predictive performance.

In recent literature, LRMC has been employed in various problems involving spatially correlated data. For example, LRMC is used in video denoising (Ji et al., 2010), seismic data reconstruction (Yang et al., 2013), and imaging recovery (Shin et al., 2014). Cabral et al. (2014) tackles the problem of multi-label image classification by extending LRMC with different loss functions to reflect the correct constraints of imaging data. Xie et al. (2017) develops a two-phase matrix-completion-based procedure with spatial and temporal considerations to recover corrupted weather data. These methods are intriguing and work well for the purpose of handling correlated missing data. However, none of these approaches directly impose spatial patterns into the lower-dimensional representation of the data. To the best of our knowledge, no other method has utilize the LRMC framework in such a way that is relevant to spatially-misaligned multi-pollutant data.

While initially focusing on health-pollution cohort studies, our proposed framework can be applicable to other fields where spatial misalignment motivates a separate exposure model. In such cases, the design matrix $\boldsymbol{Z}$ can be modified to incorporate whatever covariates are necessary, with spline terms that represents various structures not limited to just spatial correlations.

Chapter 4

HIGHER-ORDER AND SPATIALLY PREDICTIVE DIMENSION REDUCTION

4.1 Introduction

Spatiotemporal data can be thought of as a three-dimensional array, or a third-order tensor (Kolda and Bader, 2009), as visualized in Figure 4.1. In practice, elements of such tensor can be missing with irregular patterns in both time and space.

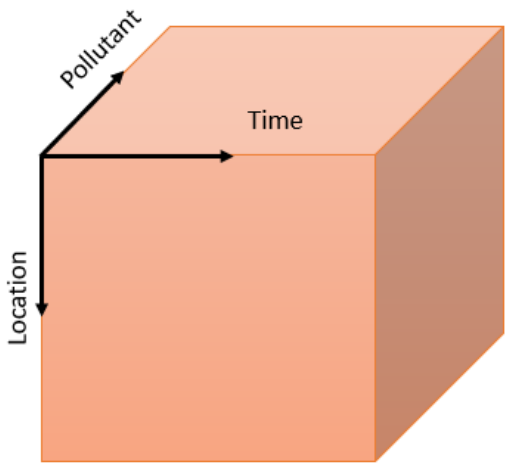

Figure 4.1: Representation of spatiotemporal data as a three-dimensional array, or third-order tensor

Spatiotemporal model for one single pollutant under spatial misalignment has been proposed (Szpiro et al., 2010) with extensive modifications to improve model flexibility as well as computational speed (Sampson et al., 2011; Lindström et al., 2014; Olives et al., 2014). These methods were developed as part of the Multi-Ethnic Study of Atherosclerosis and Air

Pollution, to assign estimates of long-term average air pollution concentrations at unmeasured locations, while taking into account the complex spatio-temporal correlation structure and misalignment of the data. Essentially, these models assume similar structure to the ProPrPCA-Spline model, where, instead of correlations across pollutants, the loadings represent temporal correlations. The probabilistic assumptions with relevant covariates and temporal splines allow these models to handle missing data over time effectively. However, extensions to the multi-pollutant setting has not been explored.

Low-rank tensor completion has recently emerged as a popular technique to recover multi-dimensional arrays of data with irregular missing patterns. Using nuclear norm minimization as a convex constraint while also assuming a low-rank latent structure has been proposed in various tensor completion algorithms (Gandy et al., 2011; Signoretto et al., 2010; Liu et al., 2013). Tensor completion, with various temporal and spatial constraints, has also been applied to recovering data with spatiotemporal structures, such as problems in video completion (Liu et al., 2013; Wang et al., 2014), traffic network (Asif et al., 2013; Zhou et al., 2015), climate data (Bahadori et al., 2014), or internet consumer data (Xie et al., 2016; Ruan et al., 2017). None of these proposed methods attempt to induce spatial structures in the latent space. Thus these approaches are not fully applicable to handle the challenge of spatial misalignment in air pollution studies.

In this chapter, we aim to develop an algorithm that produces a lower-dimensional representation of the original data tensor, while retaining spatial structures in the estimated scores, so that these scores can be easily predicted at unmeasured locations. Ideally, the method should also be able to handle tensors with missing data without requiring a separate imputation step prior to dimension reduction.

In Section 4.2, we introduce notations and basic mathematical properties of tensor algebra. Later in Section 4.3, we introduce existing orthogonal iteration algorithm for higher-order dimension reduction, and then propose an extension to introduce spatial element into the existing algorithm in Section 4.4. Finally, we show the merits of our proposed method via a toy simulation and data application.

4.2 Review of some standard operations in tensor algebra

4.2.1 Notation

In this section, we closely follow the notation used by Kolda and Bader (2009). Tensors are denoted by boldface script letters, e.g. $\boldsymbol{\mathcal{X}}$. The order of a tensor is the number of its dimensions, also known as its ways or modes. In our scenario of interest, the spatial-temporal data tensor is denoted as $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{N \times T \times P}$, where N is the number of monitoring locations, T is the number of time points at which measures are taken, and P is the number of pollutants. Here each element of the tensor is denoted by x_{ntp} where $n = 1, ..., N, t = 1, ..., T$, and $p = 1, ..., P$.

Matrices are denoted by boldface capital letters. Fibers, the higher-order analogue of matrix rows and columns, are denoted by boldface letters. For our spatial-temporal tensor, column, row, and tube fibers are denoted by $\boldsymbol{x}_{:tp}, \boldsymbol{x}_{n:p}$, and $\boldsymbol{x}_{nt:}$.

The following definitions and texts are explained with more details in Kolda and Bader (2009). Here we only look at definitions and propositions that are relevant and important to our understanding of the algorithms in Sections 4.3 and 4.4.

4.2.2 The $n-$mode multiplication

The $n-$mode product defines multiplication of a tensor by a matrix in mode n. For example, consider $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{N \times T \times P}$, $\boldsymbol{A}_1 \in \mathbb{R}^{I \times N}$, $\boldsymbol{A}_2 \in \mathbb{R}^{I \times T}$, and $\boldsymbol{A}_3 \in \mathbb{R}^{I \times P}$. The results of these $n-$mode products are tensors of the following sizes:

$$\boldsymbol{\mathcal{X}} \times_1 \boldsymbol{A}_1 = \boldsymbol{\mathcal{Y}}_1 \in \mathbb{R}^{I \times T \times P}$$
$$\boldsymbol{\mathcal{X}} \times_2 \boldsymbol{A}_2 = \boldsymbol{\mathcal{Y}}_2 \in \mathbb{R}^{N \times I \times P}$$
$$\boldsymbol{\mathcal{X}} \times_3 \boldsymbol{A}_3 = \boldsymbol{\mathcal{Y}}_3 \in \mathbb{R}^{N \times T \times I}$$

We now look at some important properties of the $n-$mode product. Note that these properties are applicable to order higher than three, but for the purpose of this chapter, we

only consider third-order tensors. Let $\mathcal{Y} \in \mathbb{R}^{J_1 \times J_2 \times J_3}$,

(a) Given matrices $\boldsymbol{A} \in \mathbb{R}^{I_m \times J_m}$, $\boldsymbol{B} \in \mathbb{R}^{I_n \times J_n}$ and $m \neq n$,

$$(\mathcal{Y} \times_m \boldsymbol{A}) \times_n \boldsymbol{B} = (\mathcal{Y} \times_n \boldsymbol{B}) \times_m \boldsymbol{A}.$$

(b) Given matrices $\boldsymbol{A} \in \mathbb{R}^{I \times J_n}$ and $\boldsymbol{B} \in \mathbb{R}^{K \times I}$,

$$\mathcal{Y} \times_n \boldsymbol{A} \times_n \boldsymbol{B} = \mathcal{Y} \times_n (\boldsymbol{B}\boldsymbol{A}).$$

(c) If $\boldsymbol{A} \in \mathbb{R}^{I \times J_n}$ is orthonormal, i.e. $\boldsymbol{A}^\mathsf{T}\boldsymbol{A} = \boldsymbol{I}_{J_n}$, then

$$\mathcal{X} = \mathcal{Y} \times_n \boldsymbol{A} \implies \mathcal{Y} = \mathcal{X} \times_n \boldsymbol{A}^\mathsf{T}.$$

4.2.3 The matricization of a tensor

In computations, sometimes it is useful to be able to transform the tensor into a matrix representation. We need to track a few pieces of information: the size of the tensor, the modes that are mapped to the columns of the matrix, the modes that are mapped to the columns of the matrix, and the data themselves.

The matricization of a tensor $\mathcal{X} \in \mathbb{R}^{N \times T \times P}$ is defined as follows. Let the ordered sets $\mathcal{R}$ and $\mathcal{C}$ be a partitioning of the modes $\mathcal{N} = 1, 2, 3$. Denote $I_\mathcal{N} = \{N, T, P\}$ as the size of the tensor. The matricized tensor can then be specified by:

$$\boldsymbol{X}_{(\mathcal{R} \times \mathcal{C}; I_\mathcal{N})} \in \mathbb{R}^{J \times K} \text{ with } J = \prod_{j \in \mathcal{R}} I_j \text{ and} K = \prod_{j \in \mathcal{C}} I_k.$$

That is, the indices in $\mathcal{R}$ are mapped to the rows and the indices in $\mathcal{C}$ are mapped to the columns. An important special case is when $\mathcal{R}$ is a singleton. This means that the fibers of mode n are aligned as the columns of the resulting matrix. This is a special case, known as

the $n-$mode matricization, or mode-n unfolding matrix of the original tensor, as illustrated in Figure 4.2.

$$\boldsymbol{X}_{(1)} = \boldsymbol{X}_{(\mathcal{R}\times\mathcal{C}:I_{\mathcal{N}})} \quad \text{with} \quad \mathcal{R} = \{1\}, \mathcal{C} = \{2,3\},$$
$$\boldsymbol{X}_{(2)} = \boldsymbol{X}_{(\mathcal{R}\times\mathcal{C}:I_{\mathcal{N}})} \quad \text{with} \quad \mathcal{R} = \{2\}, \mathcal{C} = \{1,3\},$$
$$\boldsymbol{X}_{(3)} = \boldsymbol{X}_{(\mathcal{R}\times\mathcal{C}:I_{\mathcal{N}})} \quad \text{with} \quad \mathcal{R} = \{3\}, \mathcal{C} = \{1,2\}.$$

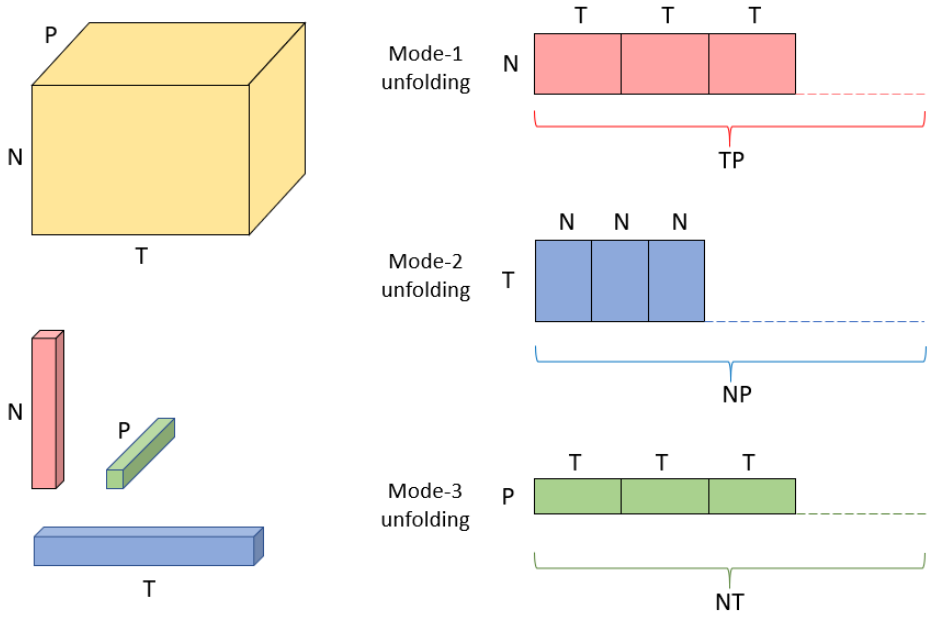

Figure 4.2: Illustration of the mode-n unfolding matrix of a tensor

In addition, a tensor can also be converted into a vector, which is a special case where all modes become row modes:

$$\text{vec}(\boldsymbol{\mathcal{X}}) = \boldsymbol{X}_{(\mathcal{N}\times\emptyset:I_{\mathcal{N}})}$$

Here is a numerical illustration: Let $\boldsymbol{\mathcal{X}}$ be the following $3 \times 4 \times 2$ tensor:

$$\boldsymbol{X}_{::1} = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix}, \quad \boldsymbol{X}_{::2} = \begin{bmatrix} 13 & 16 & 19 & 22 \\ 14 & 17 & 20 & 23 \\ 15 & 18 & 21 & 24 \end{bmatrix}$$

Then one version of matricization can be:

$$\boldsymbol{X}_{(\{3,1\}\times\{2\}:\{3,4,2\})} = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 13 & 16 & 19 & 22 \\ 2 & 5 & 8 & 11 \\ 14 & 17 & 20 & 23 \\ 3 & 6 & 9 & 12 \\ 15 & 18 & 21 & 24 \end{bmatrix}$$

The n-mode unfolding matrices are:

$$\boldsymbol{X}_{(1)} = \boldsymbol{X}_{(\{1\}\times\{2,3\}:\{3,4,2\})} = \begin{bmatrix} 1 & 4 & 7 & 10 & 13 & 16 & 19 & 22 \\ 2 & 5 & 8 & 11 & 14 & 17 & 20 & 23 \\ 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 \end{bmatrix}$$

$$\boldsymbol{X}_{(2)} = \boldsymbol{X}_{(\{2\}\times\{1,3\}:\{3,4,2\})} = \begin{bmatrix} 1 & 2 & 3 & 13 & 14 & 15 \\ 4 & 5 & 6 & 16 & 17 & 18 \\ 7 & 8 & 9 & 19 & 20 & 21 \\ 10 & 11 & 12 & 22 & 23 & 24 \end{bmatrix}$$

$$\boldsymbol{X}_{(3)} = \boldsymbol{X}_{(\{3\}\times\{1,2\}:\{3,4,2\})} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 \end{bmatrix}$$

Here we list some important properties that will be useful for later section. Let $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{J_1 \times J_2 \times J_3}$ and $\mathcal{N} = 1, 2, 3$.

(a) If $\boldsymbol{A} \in \mathbb{R}^{I \times J_n}$. Then

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{Y}} \times_n \boldsymbol{A} \iff \boldsymbol{X}_{(n)} = \boldsymbol{A}\boldsymbol{Y}_{(n)}$$

(b) Let $\boldsymbol{A}_n \in \mathbb{R}^{I_n \times J_n}$ for all $n \in \mathcal{N}$. If $\mathcal{R} = \{r_1, ..., r_L\}$ and $\mathcal{C} = \{c_1, ..., c_M\}$ partition $\mathcal{N}$, then $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{Y}} \times_1 \boldsymbol{A}_1 \times_2 \boldsymbol{A}_2 \times_3 \boldsymbol{A}_3$ if and only if:

$$\boldsymbol{X}_{(\mathcal{R}\times\mathcal{C}:J_{\mathcal{N}})} = (\boldsymbol{A}_{r_L} \otimes ... \otimes \boldsymbol{A}_{r_1})\boldsymbol{Y}_{(\mathcal{R}\times\mathcal{C}:I_{\mathcal{N}})}(\boldsymbol{A}_{c_M} \otimes ... \otimes \boldsymbol{A}_{c_1})^{\mathsf{T}}$$

and, particularly:

$$\begin{aligned}
\boldsymbol{X}_{(1)} &= \boldsymbol{A}_1\boldsymbol{Y}_{(1)}(\boldsymbol{A}_3 \otimes \boldsymbol{A}_2)^{\mathsf{T}} \\
\boldsymbol{X}_{(2)} &= \boldsymbol{A}_2\boldsymbol{Y}_{(2)}(\boldsymbol{A}_3 \otimes \boldsymbol{A}_1)^{\mathsf{T}} \\
\boldsymbol{X}_{(3)} &= \boldsymbol{A}_3\boldsymbol{Y}_{(3)}(\boldsymbol{A}_2 \otimes \boldsymbol{A}_1)^{\mathsf{T}}
\end{aligned}$$

where $\otimes$ denotes the Kronecker product between two matrices.

4.2.4 Norm and inner product of tensor

The norm and inner product are most easily thought of in terms of the vectorized tensor. The inner produce of two tensors of the same size is given by:

$$\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \rangle = \text{vec}(\boldsymbol{\mathcal{X}})^{\mathsf{T}}\text{vec}(\boldsymbol{\mathcal{Y}})$$

The norm of a tensor $\boldsymbol{\mathcal{X}}$ is given by:

$$\|\boldsymbol{\mathcal{X}}\|^2 = \langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{X}} \rangle$$

Here are some useful properties that will be used in later proofs.

(a) Let $n \in \mathcal{N}$. Then $\|\boldsymbol{\mathcal{X}}\| = \|\boldsymbol{X}_{(n)}\|_F = \|\text{vec}(\boldsymbol{\mathcal{X}})\|_2$

(b) $\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{Y}}\|^2 = \|\boldsymbol{\mathcal{X}}\|^2 - 2\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \rangle + \|\boldsymbol{\mathcal{Y}}\|^2$

(c) Let $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times ... \times I_{n-1} \times J \times I_{n+1} \times ... \times I_N}$, $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{I_1 \times ... \times I_{n-1} \times K \times I_{n+1} \times ... \times I_N}$, and $\boldsymbol{A} \in \mathbb{R}^{J \times K}$.

Then $\langle \mathcal{X}, \mathcal{Y} \times_n \boldsymbol{A} \rangle = \langle \mathcal{X} \times_n \boldsymbol{A}^\mathsf{T}, \mathcal{Y} \rangle$

(d) Let $\boldsymbol{Q} \in \mathbb{R}^{J \times I_n}$ be an orthonormal matrix. Then $\|\mathcal{X}\| = \|\mathcal{X} \times_n \boldsymbol{Q}\|$

4.2.5 The Tucker decomposition

The Tucker decomposition of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is given by:

$$\mathcal{X} = [\![\mathcal{G}; \boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3]\!] = \mathcal{G} \times_1 \boldsymbol{A}_1 \times_2 \boldsymbol{A}_2 \times_3 \boldsymbol{A}_3$$

Here $\boldsymbol{A}_n \in \mathbb{R}^{I_n \times J_n}$ and $\mathcal{G} \in \mathbb{R}^{J_1 \times J_2 \times J_3}$. If $\mathcal{G}$ is the same size as $\mathcal{X}$, the Tucker decomposition is simply a change of basis. More often, we are interested in compressing $\mathcal{X}$, thus resulting in a tensor $\mathcal{G}$ that is smaller than $\mathcal{X}$, as illustrated in Figure 4.3. In general, the Tucker decomposition is not unique unless there are other constraints placed on the $\boldsymbol{A}_n$'s or $\mathcal{G}$.

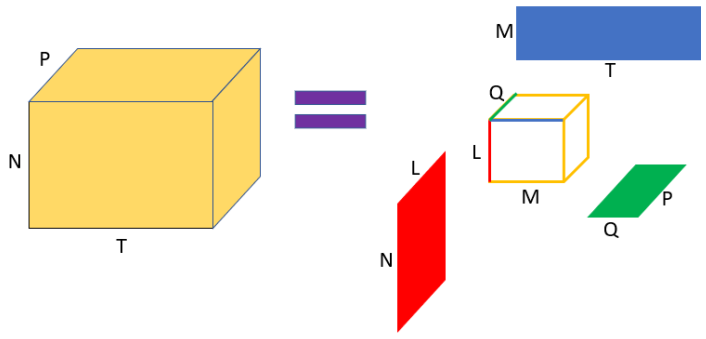

Figure 4.3: Illustration of the Tucker decomposition of a tensor

Some properties of the Tucker operator $[\![.]\!]$, which are essentially the same as some properties of the $n-$mode multiplication:

(a) Given $\boldsymbol{A}_n \in \mathbb{R}^{I_n \times J_n}$ and $\boldsymbol{B}_n \in \mathbb{R}^{K_n \times I_n}$, then

$$[\![[\![\mathcal{Y}; \boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3]\!]; \boldsymbol{B}_1, \boldsymbol{B}_2, \boldsymbol{B}_3]\!] = [\![\mathcal{Y}; \boldsymbol{B}_1\boldsymbol{A}_1, \boldsymbol{B}_2\boldsymbol{A}_2, \boldsymbol{B}_3\boldsymbol{A}_3]\!]$$

(b) Given orthonormal matrices $\boldsymbol{A}_n \in \mathbb{R}^{I_n \times J_n}$:

$$\mathcal{X} = [\![\mathcal{Y}; \boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3]\!] \implies \mathcal{Y} = [\![\mathcal{X}; \boldsymbol{A}_1^\mathsf{T}, \boldsymbol{A}_2^\mathsf{T}, \boldsymbol{A}_3^\mathsf{T}]\!]$$

4.3 The Tucker decomposition with optimal rank approximation

The $n-$rank of a tensor $\mathcal{X}$ is defined as the rank of $\boldsymbol{X}_{(n)}$. In the Tucker decomposition

$$\mathcal{X} = [\![\mathcal{G}; \boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3]\!] = \mathcal{G} \times_1 \boldsymbol{A}_1 \times_2 \boldsymbol{A}_2 \times_3 \boldsymbol{A}_3$$

where $\mathcal{G} \in \mathbb{R}^{J_1 \times J_2 \times J_3}$, if we let J_n be the $n-$rank of $\mathcal{X}$ for each n, then we can always reproduce $\mathcal{X}$ exactly. Otherwise, the decomposition may only produce an approximation of the original tensor.

Given a desired rank of the core tensor $\mathcal{G}$, one can consider the problem of computing a Tucker decomposition with the least amount of error. We can think of this as an SVD in three dimensions. In other words, we would like to solve

$$\begin{aligned} \min_{\mathcal{G}, \boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3} \quad & \left\| \mathcal{X} - [\![\mathcal{G}; \boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3]\!] \right\|^2 \\ \text{s.t.} \quad & \mathcal{G} \in \mathbb{R}^{J_1 \times J_2 \times J_3} \\ & \boldsymbol{A}_n \in \mathbb{R}^{I_n \times J_n} \text{ orthonormal where } n = 1, 2, 3. \end{aligned}$$

It turns out that we can solve this problem easily using an alternating algorithm, in which there is a closed-form solution for each variable when others are fixed. The proofs are briefly laid out in Kolda and Bader (2009) and further explained with more details in Appendix C.1. The proofs utilize some of the properties of tensor algebra and tensor norms mentioned

in the previous section. The algorithm involves the following steps:

The Tucker decomposition algorithm:

Input $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, desired output rank $\{J_1 \times J_2 \times J_3\}$, and l_{max}

Initialize $l = 1$,

$\boldsymbol{A}_1^{(0)} = J_1$ left singular vectors of $\boldsymbol{X}_{(1)}$,

$\boldsymbol{A}_2^{(0)} = J_2$ left singular vectors of $\boldsymbol{X}_{(2)}$,

$\boldsymbol{A}_3^{(0)} = J_3$ left singular vectors of $\boldsymbol{X}_{(3)}$,

while not converged or $l < l_{max}$ **do**

$\boldsymbol{A}_1^{(l+1)} \leftarrow J_1$ left singular vectors of $\boldsymbol{X}_{(1)} \left(\boldsymbol{A}_3^{(l)} \otimes \boldsymbol{A}_2^{(l)} \right)$

$\boldsymbol{A}_2^{(l+1)} \leftarrow J_2$ left singular vectors of $\boldsymbol{X}_{(2)} \left(\boldsymbol{A}_3^{(l)} \otimes \boldsymbol{A}_1^{(l)} \right)$

$\boldsymbol{A}_3^{(l+1)} \leftarrow J_3$ left singular vectors of $\boldsymbol{X}_{(3)} \left(\boldsymbol{A}_2^{(l)} \otimes \boldsymbol{A}_1^{(l)} \right)$

$l \leftarrow l + 1$

end while

Output $\widehat{\boldsymbol{A}}_1 = \boldsymbol{A}_1^{(l)}$, $\widehat{\boldsymbol{A}}_2 = \boldsymbol{A}_2^{(l)}$, $\widehat{\boldsymbol{A}}_3 = \boldsymbol{A}_3^{(l)}$, and $\hat{\mathcal{G}} = \mathcal{X} \times_1 \widehat{\boldsymbol{A}}_1^{\mathsf{T}} \times_2 \widehat{\boldsymbol{A}}_2^{\mathsf{T}} \times_3 \widehat{\boldsymbol{A}}_3^{\mathsf{T}}$

In other words, to solve for $\mathcal{G}$ we only need to perform one tensor multiplication. Meanwhile, to solve for each of the $\boldsymbol{A}_n$'s we need to do SVD on the mode-n unfolding matrix of an approximation using other known values.

4.4 Proposed algorithm: the spatial Tucker decomposition

In our problem with spatial misalignment, we assume that there are some common structures shared across the dimensions of time and pollutant. We also want to impose spatial structures along the location-mode of the original tensor, using a $\boldsymbol{Z}$ matrix of covariates and spline terms

similar to previous chapters. In other words, we would like to solve the following problem:

$$\begin{aligned} \min_{\mathcal{G}, \boldsymbol{V}_t, \boldsymbol{V}_p} \quad & \left\| \mathcal{X} - [\![\mathcal{G}; \boldsymbol{Z}, \boldsymbol{V}_t, \boldsymbol{V}_p]\!] \right\|^2 \\ \text{s.t.} \quad & \mathcal{G} \in \mathbb{R}^{K \times M \times Q}, M < T, Q < P \\ & \boldsymbol{V}_t \in \mathbb{R}^{T \times M} \text{ orthonormal} \\ & \boldsymbol{V}_p \in \mathbb{R}^{P \times Q} \text{ orthonormal} \end{aligned}$$

Here the meaning of $\mathcal{G}$ is no longer the core tensor defined in the Tucker decomposition framework. Rather, the product $\mathcal{G} \times_1 \boldsymbol{Z}$ is the lower-dimensional representation after dimension reduction has been applied to the temporal and pollutant dimensions. Essentially, we are trying to approximate the original data $\mathcal{X}$ by the approximation,

$$\mathcal{X} \approx \mathcal{G} \times_1 \boldsymbol{Z} \times_2 \boldsymbol{V}_t \times_3 \boldsymbol{V}_p$$

where $\boldsymbol{Z}$ is already known. By doing this, we are forcing spatial patterns, described by the covariates and spline terms included in $\boldsymbol{Z}$, across the location-mode (mode-1) of the tensor. $\boldsymbol{V}_t$ and $\boldsymbol{V}_p$ play the roles of temporal and pollutant loadings.

It turns out that we can also derive the solutions using an algorithm similar to the one described in the previous section. Our proofs are shown in Appendix C.2.

Note that here we do not need to get $\hat{\mathcal{G}}$, the core tensor with reduced size in all three dimensions. We are only interested in predicting $\hat{\mathcal{H}} = \hat{\mathcal{G}} \times \boldsymbol{Z}$, the core tensor with reduced size in only the pollutant and time dimensions, at new locations. We can vectorize this approximation tensor and then use universal kriging or other spatial prediction tools to obtain estimates at new locations.

Proposed algorithm: The spatial Tucker decomposition

Input $\mathcal{X} \in \mathbb{R}^{N \times T \times P}$, desired output rank $\{K \times M \times Q\}$, Z, and l_{max}
  Initialize $l = 1$,
  Calculate $\tilde{\mathcal{X}} = \mathcal{X} \times_1 \left(\boldsymbol{Z}(\boldsymbol{Z}^\top \boldsymbol{Z})^{-1}\boldsymbol{Z}^\top\right)$
  $\boldsymbol{V}_t^{(0)} = M$ left singular vectors of $\tilde{\mathcal{X}}_{(2)}$,
  $\boldsymbol{V}_p^{(0)} = Q$ left singular vectors of $\tilde{\mathcal{X}}_{(3)}$,
    while not converged or $l < l_{max}$ **do**
      $\boldsymbol{V}_t^{(l+1)} \leftarrow M$ left singular vectors of $\tilde{\mathcal{X}}_{(2)}\left(\boldsymbol{V}_p^{(l)} \otimes \boldsymbol{I}_n\right)$
      $\boldsymbol{V}_p^{(l+1)} \leftarrow Q$ left singular vectors of $\tilde{\mathcal{X}}_{(3)}\left(\boldsymbol{V}_t^{(l)} \otimes \boldsymbol{I}_n\right)$
      $l \leftarrow l + 1$
    end while
Output $\widehat{\boldsymbol{V}}_t = \boldsymbol{V}_t^{(l)}$, $\widehat{\boldsymbol{V}}_p = \boldsymbol{V}_p^{(l)}$, and tensor with reduced dimensions in temporal and pollutant space $\hat{\mathcal{H}} = \hat{\mathcal{G}} \times \boldsymbol{Z} = \tilde{\mathcal{X}} \times_2 \widehat{\boldsymbol{V}}_t^\top \times_3 \widehat{\boldsymbol{V}}_p^\top$

4.5 Toy simulations

4.5.1 Setups

We conduct two sets of toy simulations to compare the predictive performance between the regular and proposed spatial algorithms. The data generating mechanism follows closely the high-dimensional simulation setups in Chapter 2, with a simple introduction of temporal dimension. We only consider complete data in our simulations.

Specifically, the data $\mathcal{X}$ are simulated on a dense 100×100 grid, with the final dimensions of $10,000 \times 10 \times 15$, i.e. 10,000 locations, 10 time points, and 15 pollutants. The data are generated from an underlying core tensor $\mathcal{H} \in \mathbb{R}^{10,000 \times 1 \times 3}$, a matrix of temporal loadings $\boldsymbol{V}_t \in \mathbb{R}^{10 \times 1}$, and a matrix of pollutant loadings $\boldsymbol{V}_p \in \mathbb{R}^{15 \times 3}$. The full setup is described in Appendix C.3. Similar to Chapter 2, the core tensor $\mathcal{H}$ consists of three scores, where $\boldsymbol{h}_{::1}$ is the most spatially predictable, $\boldsymbol{h}_{::2}$ is moderately predictable, while $\boldsymbol{h}_{::3}$ is not all predictable by generated covariates. The temporal loadings are sine function of time. The pollutant loadings are sparse, such that the pollutant columns $(\boldsymbol{x}_{::1}, \boldsymbol{x}_{::2}, \boldsymbol{x}_{::3}, \boldsymbol{x}_{::4}, \boldsymbol{x}_{::5})$,

$(\boldsymbol{x}_{::6}, \boldsymbol{x}_{::7}, \boldsymbol{x}_{::8}, \boldsymbol{x}_{::9}, \boldsymbol{x}_{::10})$, and $(\boldsymbol{x}_{::11}, \boldsymbol{x}_{::12}, \boldsymbol{x}_{::13}, \boldsymbol{x}_{::14}, \boldsymbol{x}_{::15})$ are generated from $\boldsymbol{h}_{::1}$, $\boldsymbol{h}_{::2}$, and $\boldsymbol{h}_{::3}$, respectively. The sparsity of the pollutant loadings allow us to identify more accurately how the methods perform in terms of finding the correct principal directions.

We also create two scenarios similar to Chapter 2. Scenario 1 has $Var(\boldsymbol{h}_{::1}) = 10$, $Var(\boldsymbol{h}_{::2}) = 7.5$, and $Var(\boldsymbol{h}_{::3}) = 5$, while scenario 2 has $Var(\boldsymbol{h}_{::1}) = 7.5$, $Var(\boldsymbol{h}_{::2}) = 5$, and $Var(\boldsymbol{h}_{::3}) = 10$. That is, in scenario 1, the order of variance contribution is the same as the order of spatial predictability, for which we would then expect both methods to perform well and similarly. In scenario 2, the non-predictable score contributes the largest amount of variance. Thus we expect that our proposed method would identify $\boldsymbol{h}_{::1}$ and $\boldsymbol{h}_{::2}$ as the first two core PCs, while the regular Tucker algorithm would obtain $\boldsymbol{h}_{::3}$ and $\boldsymbol{h}_{::1}$ as the first two PCs instead.

4.5.2 Evaluations

For each scenario, we use 1,000 simulations, in which we randomly choose 400 training and 100 testing locations. We then apply either the regular or the spatial Tucker decomposition algorithms to the training data $\boldsymbol{\mathcal{X}}^{train} \in \mathbb{R}^{400\times10\times15}$ to obtain the temporal loadings $\hat{\boldsymbol{V}}_t^{train} \in \mathbb{R}^{10\times1}$, the pollutant loadings $\hat{\boldsymbol{V}}_p^{train} \in \mathbb{R}^{15\times3}$, and the core tensor $\hat{\boldsymbol{\mathcal{H}}}^{train} \in \mathbb{R}^{400\times1\times3}$ with reduced dimensions in both the temporal and pollutant directions, i.e. $\hat{\boldsymbol{\mathcal{H}}}^{train} = \boldsymbol{\mathcal{X}}^{train} \times_2 (\hat{\boldsymbol{V}}_t^{train})^{\mathsf{T}} \times_3 (\hat{\boldsymbol{V}}_p^{train})^{\mathsf{T}}$. Operating under the assumption that pollutant and temporal correlations have been captured by the loadings, the elements of $\hat{\boldsymbol{\mathcal{H}}}^{train}$ should therefore be spatially correlated across locations only. We can then use $\hat{\boldsymbol{\mathcal{H}}}^{train}$ and relevant covariate information to obtain $\hat{\boldsymbol{\mathcal{H}}}^{test}$, the predicted scores at testing locations, in a universal kriging model with an exponential covariance assumption, which is consistent with the data generating mechanism.

There are several metrics to evaluate the predictive performance of the two methods. For example, we can look at the prediction R^2's of the first two core PCs. That is, we can compare the predicted score, $\hat{\boldsymbol{h}}_{::j}^{test}$ $(j = 1, 2)$, to the known score $\boldsymbol{h}_{::j}^{test}$. The known scores $\boldsymbol{\mathcal{H}}^{test}$ would be the lower-dimensional representation of $\boldsymbol{\mathcal{X}}^{test}$ using $\hat{\boldsymbol{V}}_t^{train}$ and $\hat{\boldsymbol{V}}_p^{train}$, if $\boldsymbol{\mathcal{X}}^{test}$ were

observed. However, rather than focusing on the core PCs, i.e. reduction in both temporal and pollutant spaces, we are more interested in the pollutant PCs, i.e. reduction in only the pollutant space. This is consistent with our prediction target in the previous chapters. That is, how well the known scores $\boldsymbol{\mathcal{U}}^{test} = \boldsymbol{\mathcal{X}}^{test} \times_3 (\hat{\boldsymbol{V}}_p^{train})^{\mathsf{T}} = \boldsymbol{\mathcal{H}}^{test} \times_2 \hat{\boldsymbol{V}}_t^{train}$ agree with the predicted scores $\hat{\boldsymbol{\mathcal{U}}}^{test} = \hat{\boldsymbol{\mathcal{H}}}^{test} \times_2 \hat{\boldsymbol{V}}_t^{train}$. As we are interested in the first two pollutant PCs, we evaluate the prediction R^2 between the elements of $\hat{\boldsymbol{u}}_{::j}^{test}$ and $\boldsymbol{u}_{::j}^{test}$ $(j = 1, 2)$, both are matrices with dimension (100×10). We refer to this as the "spatio-temporal" R^2

In addition, we are also interested in how well the methods can aid in predicting the time series at each location. To do this, we compute the prediction R^2 of $\hat{\boldsymbol{u}}_{i:j}^{test}$ and $\boldsymbol{u}_{i:j}^{test}$ at each location $(i = 1, ..., 100)$ for each pollutant PC $(j = 1, 2)$. We refer to these values as the "temporal" R^2. We then evaluate the distributions of the temporal R^2 for each pollutant PC across the 1,000 simulations.

4.5.3 Results

Table 4.1: Median estimated pollutant loadings across 1,000 simulations of scenario 1

	Regular algorithm		Spatial algorithm	
Pollutant	PC1	PC2	PC1	PC2
1	0.44	0.00	0.45	0.00
2	0.45	0.00	0.45	0.00
3	0.44	0.00	0.44	0.00
4	0.45	0.00	0.45	0.00
5	0.44	0.00	0.44	0.00
6	0.00	0.44	0.00	0.44
7	0.00	0.44	0.00	0.44
8	0.00	0.45	0.00	0.45
9	0.00	0.45	0.00	0.45
10	0.00	0.45	0.00	0.45
11	0.01	-0.01	0.00	0.00
12	0.01	-0.01	0.00	0.00
13	0.01	-0.01	0.00	0.00
14	0.01	-0.01	0.00	0.00
15	0.01	-0.01	0.00	0.00

Table 4.2: Mean (SD) of spatio-temporal prediction R^2 values across 1,000 simulations of scenario 1

	Regular algorithm	Spatial algorithm
Pollutant PC1	0.802 (0.036)	0.802 (0.036)
Pollutant PC2	0.760 (0.043)	0.761 (0.043)

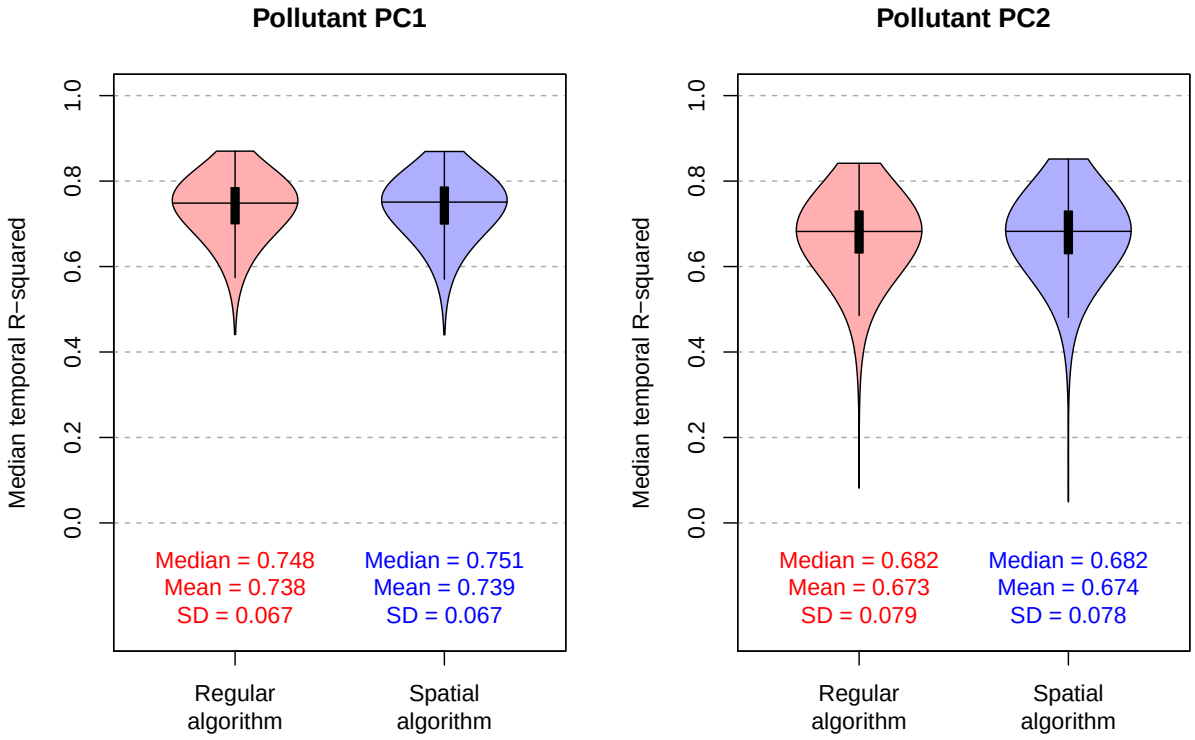

Figure 4.4: Distributions of median temporal R^2 across 1,000 simulations in scenario 1

While not shown in detail, both methods identify similar and accurate values for the temporal loading in both scenarios. Table 4.1 shows the median estimated pollutant loadings across 1,000 simulations of scenario 1. Both methods obtain correct pollutant loadings corresponding to $\boldsymbol{h}_{::1}$ and $\boldsymbol{h}_{::2}$, respectively, as the first two core PCs. As a result, the predictive performance is similar between the methods, as shown in Table 4.2. For each of the 1,000

simulations, we compute the median of the temporal R^2 values across 100 testing locations. Figure 4.4 compares the distributions of these median temporal R^2 values for the two pollutant PCs in scenario 1. The distributions of the two methods are almost indistinguishable in this scenario.

Tables 4.3, 4.4, and Figure 4.5 show the results for scenario 2, in which the non-predictable core PC contributes the highest amount of variability in the data. While the proposed algorithm still identifies $\boldsymbol{h}_{::1}$ and $\boldsymbol{h}_{::2}$ as the first two core PCs, the regular algorithm picks $\boldsymbol{h}_{::3}$ and $\boldsymbol{h}_{::1}$ instead, due to their variance contributions. Because $\boldsymbol{h}_{::3}$ is not predictable at new locations, the prediction R^2 for the first pollutant PC averages to almost zero for the regular Tucker algorithm. Similarly, the corresponding distribution of median temporal R^2 for the first pollutant PC concentrates around zero. Meanwhile, the performance of the spatial algorithm remains relatively the same for both spatio-temporal and temporal R^2.

Table 4.3: Median estimated pollutant loadings across 1,000 simulations of scenario 2

	Regular algorithm		Spatial algorithm	
Pollutant	PC1	PC2	PC1	PC2
1	0.02	0.44	0.44	0.00
2	0.01	0.45	0.45	0.00
3	0.02	0.44	0.44	0.00
4	0.02	0.45	0.45	0.00
5	0.02	0.44	0.44	0.00
6	0.00	0.00	0.00	0.44
7	0.00	0.00	0.00	0.44
8	0.00	0.00	0.00	0.44
9	-0.01	0.00	0.00	0.44
10	0.00	0.00	0.00	0.44
11	0.44	-0.02	0.01	-0.01
12	0.44	-0.02	0.01	-0.01
13	0.45	-0.02	0.01	-0.01
14	0.44	-0.01	0.01	-0.01
15	0.45	-0.02	0.01	-0.01

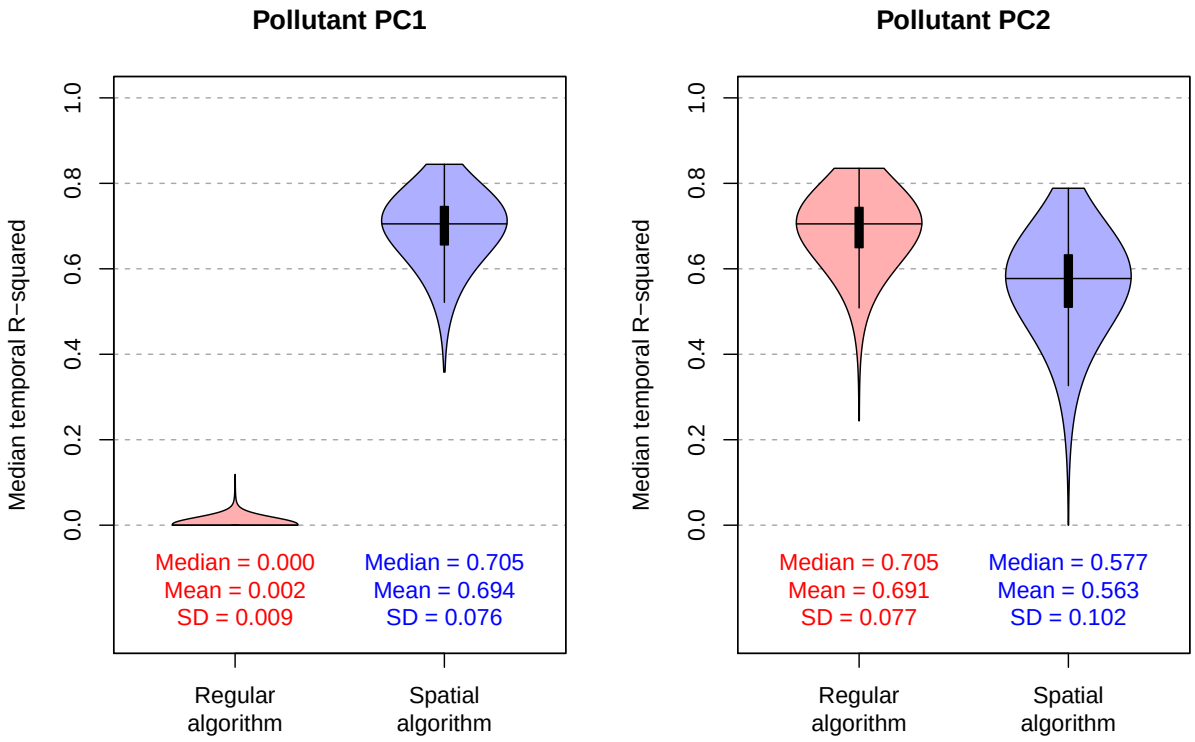

Figure 4.5: Distributions of median temporal R^2 across 1,000 simulations in scenario 2

Table 4.4: Mean (SD) of spatio-temporal prediction R^2 values across 1,000 simulations of scenario 2

	Regular algorithm	Spatial algorithm
Pollutant PC1	0.009 (0.016)	0.780 (0.040)
Pollutant PC2	0.777 (0.043)	0.691 (0.070)

4.6 Data application

4.6.1 Data

To illustrate the merit of our proposed spatial algorithm over the standard method, we use data that are similar to the previous two chapters, but were collected biweekly through the

Chemical Speciation Network of monitors. Data are available for 22 chemical components of $PM_{2.5}$. We only consider sites with complete data for all components throughout the 26 time points. The geographic covariates are provided from the Exposure Assessment Core Database by the MESA Air team at the University of Washington.

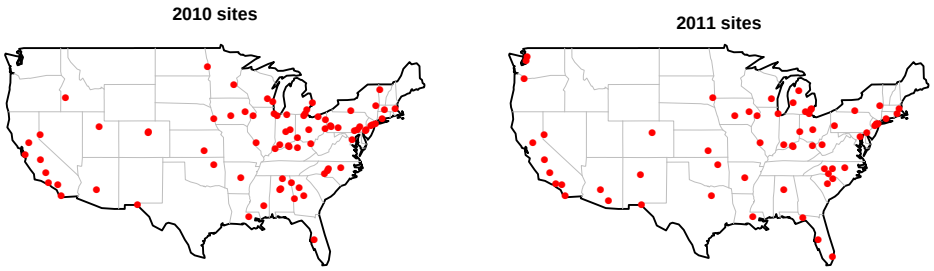

Figure 4.6: Location of sites in 2010 (left) and 2011 (right) that are included in data analysis.

Figure 4.6 displays the locations of sites included in the data analysis for each year. We end up with 78 sites for 2010 and 62 sites for 2011. Note that there are more sites on the East coast in 2010, and more sites on the West coast and Southwest region in 2011. Overall, sites available in 2011 tend to be spread out more. Meanwhile in the 2010 analysis, there are more sites in the Eastern regions, and these sites are more tightly clustered 2010 analysis tend to cluster more in the Eastern regions.

Figures 4.9 and 4.10 illustrate the overall temporal trends across 22 chemical components of $PM_{2.5}$ in 2010 and 2011, respectively. Note that, similar to previous chapters, we convert the mass concentrations to proportions by dividing by the total mass of $PM_{2.5}$, and then log-transform these fractions. The temporal trends vary across different components, with some elements having distinguishing features or being similar to one another, e.g. NO_3^- and Zn, the trio Na, Ni, and V, or SO_4^{2-} and S.

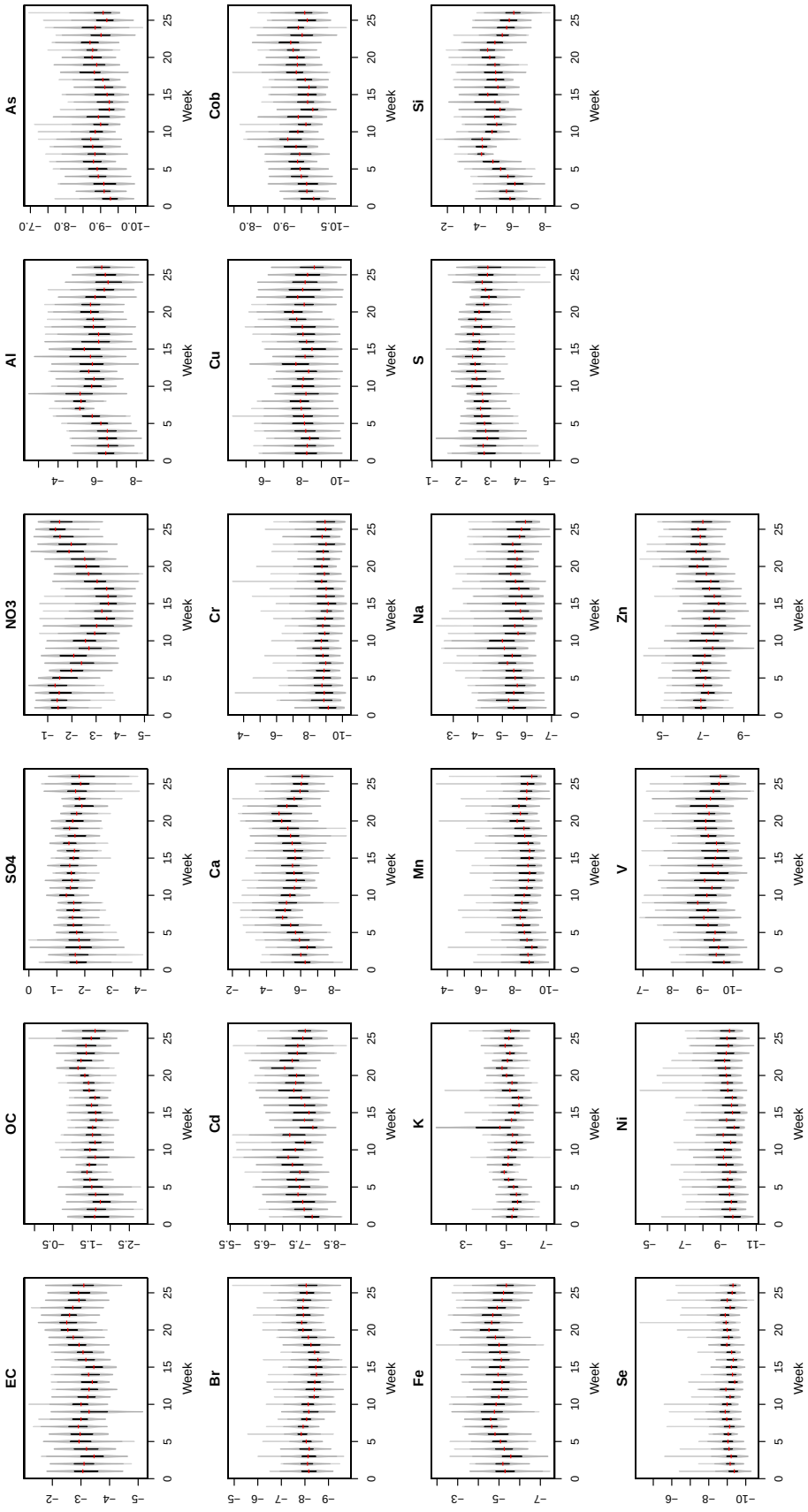

Figure 4.7: Overall temporal trends in 2010 biweekly data. The distribution of the data at each time point is shown by a violin plot in gray, with its median in red.

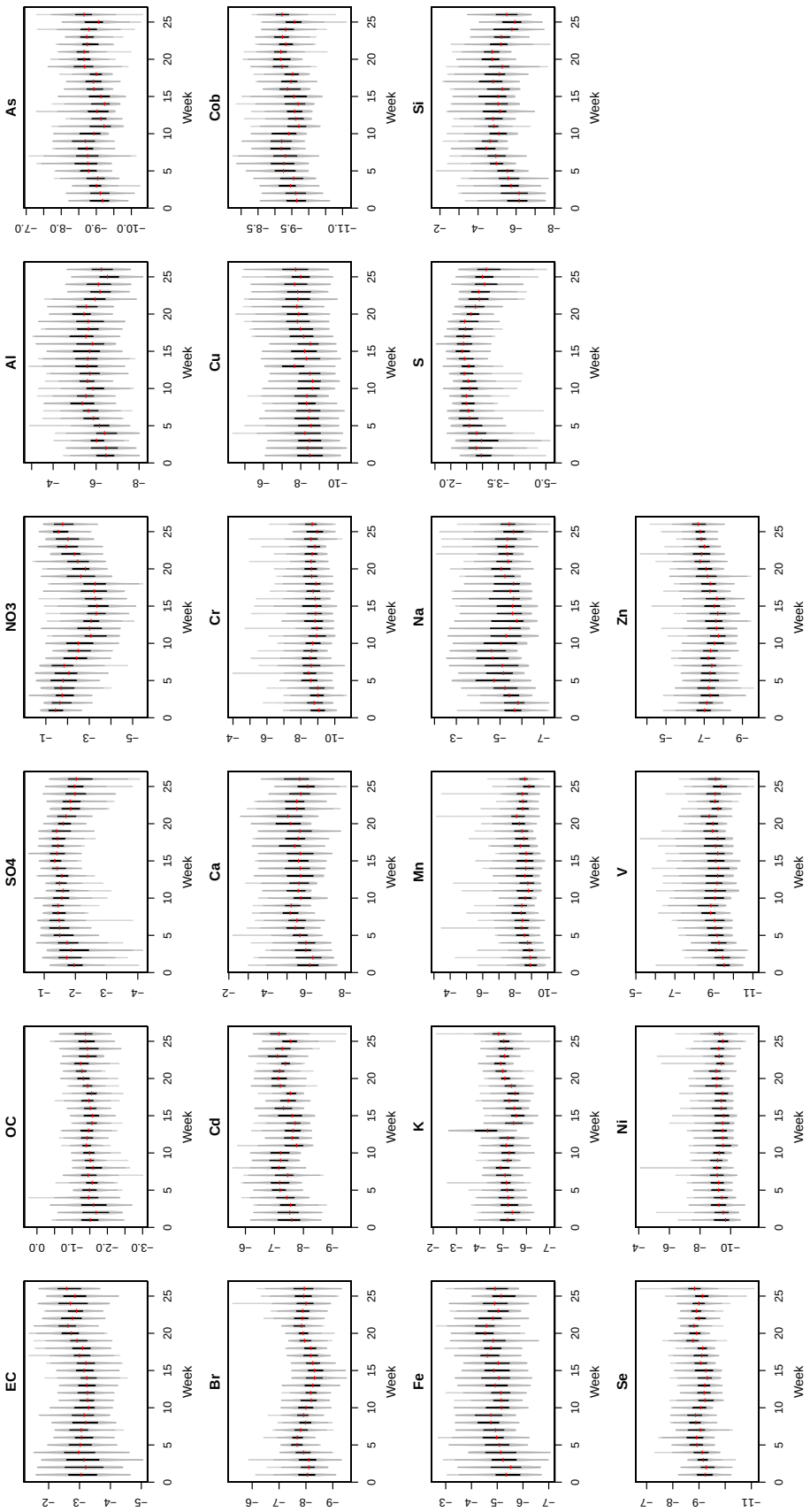

Figure 4.8: Overall temporal trends in 2011 biweekly data. The distribution of the data at each time point is shown by a violin plot in gray, with its median in red.

4.6.2 Methods and evaluations

We follow the same preprocessing procedure for the geographic covariates as described in Chapter 2. In summary, from originally more than 600 GIS covariates, we remove covariates that are missing at analysis sites, those that have the same values for more than 80% of the sites, those that have at least 2% of their values more than five standard deviations away from their sample means, or land-use variables whose maximal values are only 10% among all sites. PCA is used on the remaining geographic covariates, and the first five PCs are used in the spatial prediction stage. These scores, together with thin-plate spline basis functions, are included in the design matrix $\boldsymbol{Z}$ used in our proposed spatial algorithm.

We conduct leave-one-site-out cross-validation to compare the performance between the regular Tucker decomposition and the proposed spatial algorithm. In each round of cross-validation, one site is left out and dimension reduction is performed on the remaining sites. Then universal kriging is used to predict the core PC elements at the left-out location. For our data analysis, we arbitrary pick $M = 3$ and $Q = 3$, i.e. we assume that there are three pollutant loadings and three temporal loadings underlying the original data. As a sensitivity analysis, we use other combinations of M and Q. These results are placed in Appendix C.4.

Similar to the previous section, while the methods produce core tensor with reduced dimensions in both pollutant and temporal spaces, our evaluation is based on the pollutant PCs, i.e. reduction in only the pollutant space. That is, we focus on how well the known scores $\boldsymbol{\mathcal{U}}^{test} = \boldsymbol{\mathcal{X}}^{test} \times_3 (\hat{\boldsymbol{V}}_p^{train})^\mathsf{T} = \boldsymbol{\mathcal{H}}^{test} \times_2 \hat{\boldsymbol{V}}_t^{train}$ agree with the predicted scores $\hat{\boldsymbol{\mathcal{U}}}^{test} = \hat{\boldsymbol{\mathcal{H}}}^{test} \times_2 \hat{\boldsymbol{V}}_t^{train}$. Specifically, we look at the prediction R^2 and also mean squared errors (MSE) between the elements of $\hat{\boldsymbol{u}}_{::j}^{test}$ and $\boldsymbol{u}_{::j}^{test}$ $(j = 1, 2, 3)$, which are both matrices with dimension (78×26) for 2010 data and (62×26) for 2011 data. In addition, we also interested in the temporal R^2 and MSE between $\hat{\boldsymbol{u}}_{i:j}^{test}$ and $\boldsymbol{u}_{i:j}^{test}$ at each location for each pollutant PC.

Finally, we want to examine whether using measurement at finer scale could be beneficial for predicting long-term averages. That is, via cross-validation, we use the predicted time series at each location to calculate its annual average. We then compare these annual averages

with the annual averages based on actual data. We also compare these results using tensor methods with those obtained by applying PCA or ProPrPCA directly onto the actual annual average data.

4.6.3 Results

Table 4.5 shows the overall prediction R^2 and MSE for 2010 data. The spatial algorithm shows improvement in both metrics, most prominently for PC1. However, the overall performance in terms of R^2 is poor for both methods.

Table 4.5: Cross-validation prediction R^2 and MSE for 2010 data with pre-specified $M = 3$ and $Q = 3$.

Prediction R^2	Regular algorithm	Spatial algorithm
Pollutant PC1	0.31	0.37
Pollutant PC2	0.25	0.23
Pollutant PC3	0.33	0.30

Prediction MSE	Regular algorithm	Spatial algorithm
Pollutant PC1	1.96	1.67
Pollutant PC2	0.62	0.70
Pollutant PC3	0.52	0.53

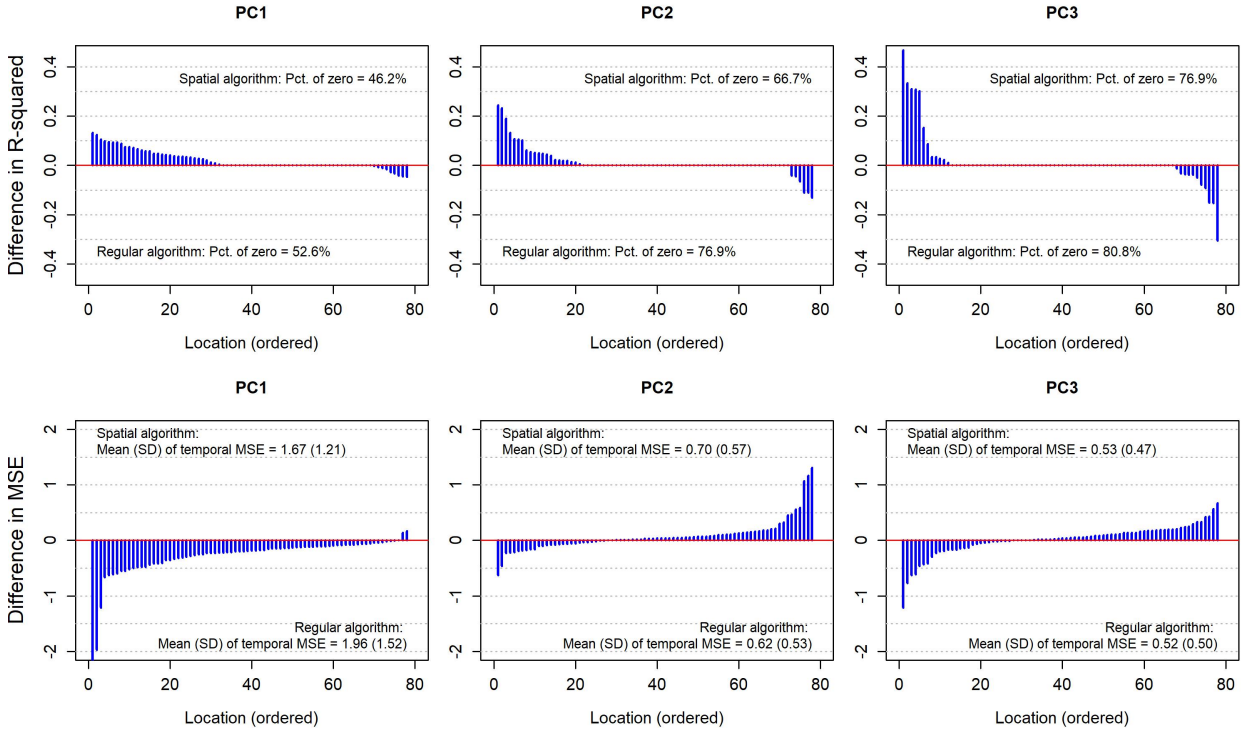

Figure 4.9: Differences in temporal R^2 and MSE for 2010 biweekly data using the regular and spatial Tucker algorithms. Positive values of difference in temporal R^2 and negative values of difference in temporal MSE indicate that the predictive performance of the spatial Tucker algorithm is, respectively, better and worse than the regular version. The locations are ordered in terms of temporal values.

Figure 4.9 shows the distributions of differences in temporal R^2 and MSE. Positive values of difference in temporal R^2 and negative values of difference in temporal MSE indicate that the predictive performance of the spatial Tucker algorithm is better compared to the regular version. The differences in mean temporal values are severe for both methods, with many zeros for prediction R^2. However, when ordered by magnitude, the results show that the spatial algorithm performs slightly better, i.e. lower proportions of zero R^2, and generally better MSE, especially for PC1.

Table 4.6 shows the results for 2010 annual averages. Similar to previous results, while it is not clear for PC2 and PC3, the spatial tensor algorithm has much better performance for

PC1 ($R^2 = 0.55$, MSE = 0.59) compared to the regular tensor algorithm ($R^2 = 0.44$, MSE = 0.74). This result is also better than PCA applied directly to annual averages. Although its performance is not as good as ProPrPCA ($R^2 = 0.61$, MSE = 0.48) for PC1, the spatial tensor algorithm leads to better predictions for PC2 and PC3.

Table 4.6: Cross-validation prediction R^2 and MSE for 2010 annual averages. The regular Tucker and spatial Tucker algorithms (with $M = 3$ and $Q = 3$) are used to predict bi-weekly data at test locations, based on which the annual averages are calculated. PCA and ProPrPCA are applied directly on the annual averages derived at training locations.

Prediction R^2	Regular algorithm	Spatial algorithm	PCA	ProPrPCA
Pollutant PC1	0.44	0.55	0.47	0.61
Pollutant PC2	0.39	0.34	0.23	0.10
Pollutant PC3	0.47	0.43	0.50	0.38

Prediction MSE	Regular algorithm	Spatial algorithm	PCA	ProPrPCA
Pollutant PC1	0.74	0.59	0.70	0.48
Pollutant PC2	0.27	0.27	0.32	0.26
Pollutant PC3	0.27	0.28	0.27	0.40

Table 4.7 shows the results by pollutant PCs for 2011 data. The overall predictive performance for both methods are better than those of 2010. There is a clear advantage of the spatial algorithm compared to the regular algorithm for both PC1 and PC3. Similar trends are also observed for the results of temporal R^2 and MSE as seen in Figure 4.10.

Table 4.7: Cross-validation prediction R^2 and MSE for 2011 data with pre-specified $M = 3$ and $Q = 3$.

Prediction R^2	Regular algorithm	Spatial algorithm
Pollutant PC1	0.41	0.45
Pollutant PC2	0.46	0.47
Pollutant PC3	0.20	0.39

Prediction MSE	Regular algorithm	Spatial algorithm
Pollutant PC1	1.65	1.47
Pollutant PC2	0.64	0.64
Pollutant PC3	0.59	0.39

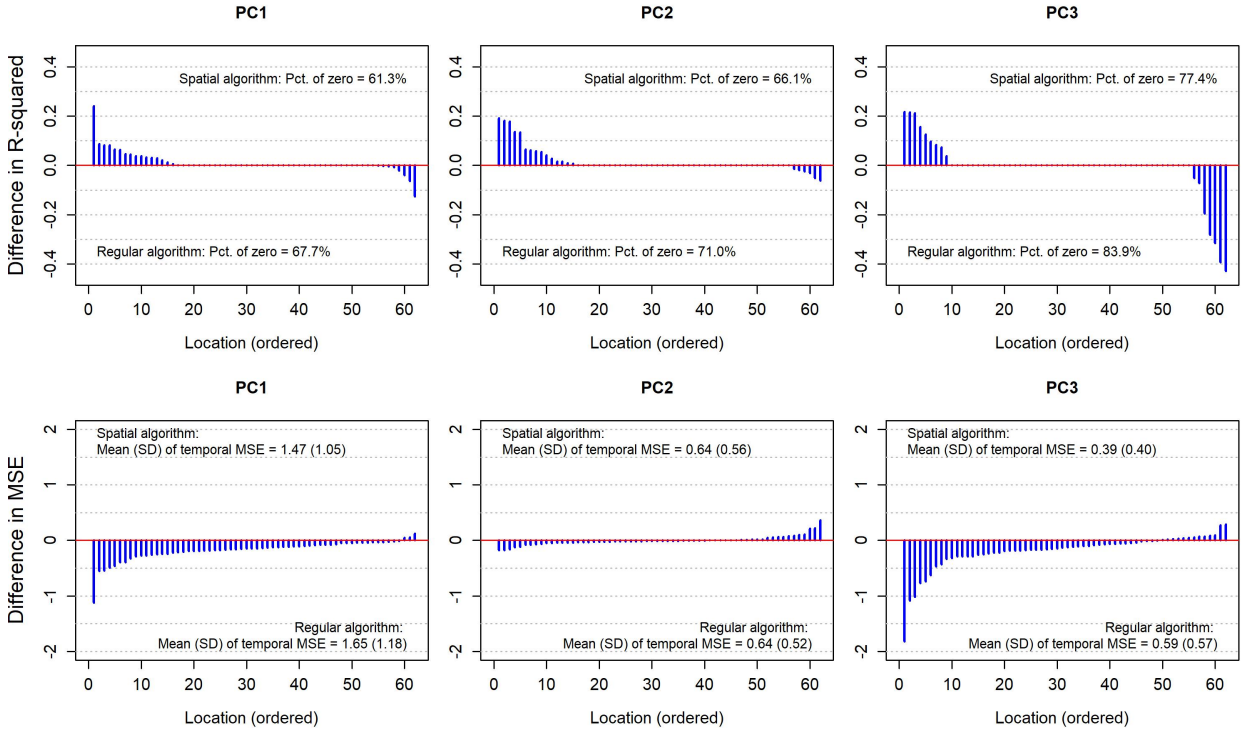

Figure 4.10: Differences in temporal R^2 and MSE for 2011 biweekly data using the regular and spatial Tucker algorithms. Positive values of difference in temporal R^2 and negative values of difference in temporal MSE indicate that the predictive performance of the spatial Tucker algorithm is, respectively, better and worse than the regular version. The locations are ordered in terms of temporal values.

Finally, Table 4.8 shows the results for 2011 annual averages. Here the spatial tensor algorithm produces better performance for PC1 than both regular tensor algorithm and PCA applied to annual data, and not much worse than ProPrPCA. ProPrPCA gives the best performance for all three PCs, but the spatial tensor algorithm follows very closely.

Overall, the results for 2011 are much better than those for 2010. It is possibly due to the locations included in each analysis. While having a smaller number of sites, the 2011 locations are less clustered in one region like 2010 but spread out more. Thus, the 2011 locations are less likely to bias the spatial prediction results of leave-one-site-out cross-validations, using the same universal kriging model that assumes stationary.

Table 4.8: Cross-validation prediction R^2 and MSE for 2011 annual averages. The regular Tucker and spatial Tucker algorithms (with $M = 3$ and $Q = 3$) are used to predict bi-weekly data at test locations, based on which the annual averages are calculated. PCA and ProPrPCA are applied directly on the annual averages derived at training locations.

Prediction R^2	Regular algorithm	Spatial algorithm	PCA	ProPrPCA
Pollutant PC1	0.64	0.69	0.66	0.72
Pollutant PC2	0.65	0.65	0.58	0.64
Pollutant PC3	0.20	0.53	0.44	0.60

Prediction MSE	Regular algorithm	Spatial algorithm	PCA	ProPrPCA
Pollutant PC1	0.55	0.47	0.51	0.42
Pollutant PC2	0.24	0.24	0.28	0.24
Pollutant PC3	0.35	0.20	0.25	0.17

4.7 Discussion

In this chapter, we propose an extension to the Tucker decomposition with optimal rank approximation. The proposed algorithm is suitable for handling misaligned spatiotemporal data, with the ultimate goal of improving the predictive performance of the exposure modeling stage. Similar to proposed methods in previous chapters, our method seeks to identify principal directions suchs that the resulting pollutant subspaces would retain spatial structures and correlations, while assuming separable temporal structures to account for correlations across time points.

We have demonstrated via toy simulations that our proposed algorithm is capable of producing lower-dimensional representation of the data in pollutant space that can be well predicted at new locations. However, our simulation setups are rather limited with sparse pollutant loadings and only one underlying vector of temporal loadings that propagate to 10 time points. More complex and comprehensive simulation studies are necessary to draw further conclusions.

A potential challenge to both the regular Tucker algorithm and our proposed method is in the pre-selection of the dimensions for the pollutant and temporal loadings. Compared to the two-dimensional case where we only need to pre-specify the number of pollutant loadings,

the choice of the temporal dimensions adds on another layer of complexity. In practice, one may choose to handle this via cross-validation. However, choosing among combinations of (Q, M), i.e. the dimensions of the pollutant (Q) and temporal (M) loadings, is not a trivial matter, as it requires double cross-validation. The choices of (Q, M) may also have different implications on later prediction stage.

Unlike the methods proposed in previous chapters, what we have proposed here so far cannot handle data with missing observations effectively. A preprocessing step with imputation is likely required. However, our proposed method for tensor data partially resembles the technique employed by the spatial matrix completion for "fully-observed" data. That is, we also utilize a similar projection onto the column space of $\boldsymbol{Z}$ in the first dimension of the tensor data. Heuristically, this can be classified as a L0-problem in which we penalize the rank of the tensor approximation to the data. Thus, there is a potential to extend the core idea in previous chapter onto the current proposed method to handle missing data efficiently and meaningfully.

Chapter 5

DISCUSSION AND FUTURE WORK

This dissertation has focused on developing statistical methods to improve the overall predictive performance of the exposure modeling procedure for spatially-misaligned multi-pollutant data with substantial amount of missing observations. Particularly for health-pollutant association studies on $PM_{2.5}$ and its components, the goal is to use estimated PC scores at unmeasured study locations as effect modifiers for the main health associations of interest. The scientific motivation arises from existing literature and evidence on the complex nature of multi-pollutant data with spatial misalignment.

We focus our attention onto the first two stages of the exposure modeling process: the imputation and dimension reduction steps. The goal is to develop dimension reduction methods that, similar to PredPCA proposed by Jandarov et al. (2017), can produce lower-dimensional scores that can be well predicted at unmeasured locations, using available geographic covariates and accounting for spatial correlations. In Chapter 2, we build likelihood-based models, in which the probabilistic assumptions allow for model-based imputation. This eliminates the necessity of a separate and disjoint stage of imputation prior to dimension reduction. Further work is necessary to troubleshoot the unstable and poor performance of ProPrPCA-Krige compared to its simpler version, ProPrPCA-Spline. Potential extensions also include the incorporation of ideas proposed by Bose et al. (2018) to adaptively select the features of the design matrix $\boldsymbol{Z}$, or the ability to choose the desired number of PCs appropriately and efficiently.

In Chapter 3, we handle the same problem using a slightly different approach. We reformulate the original problem into a convex optimization problem, SMC, for which we derive a straightforward and computationally efficient algorithm using proximal gradient

descent. While SMC is capable of estimating all PCs simultaneously, the rank of the lower-dimensional representation as well as the features included in $\boldsymbol{Z}$ have to pre-specified, similar to the shortcomings of ProPrPCA.

Finally in Chapter 4, we venture into spatially-misaligned multi-pollutant spatiotemporal data. Using similar ideas as in previous chapters, we extend an existing method in tensor dimension reduction to create a new algorithm that can take into account geographic covariates and spatial information. The proposed algorithm has shown promising results in toy simulations and small data applications. While the algorithm can only handle complete data and still requires pre-specified lower dimensions in both temporal and pollutant spaces, this chapter represents an encouraging start on generalizing the predictive PCA idea to spatiotemporal data. A potential direction for future research is the extension of the nuclear norm penalization in SMC to the tensor setting.

BIBLIOGRAPHY

Achilleos, S., Kioumourtzoglou, M.-A., Wu, C.-D., Schwartz, J. D., Koutrakis, P., and Papatheodorou, S. I. (2017). Acute effects of fine particulate matter constituents on mortality: A systematic review and meta-regression analysis. *Environment International*, 109:89–100.

Asif, M. T., Mitrovic, N., Garg, L., Dauwels, J., and Jaillet, P. (2013). Low-dimensional models for missing data imputation in road networks.

Bahadori, M. T., Yu, Q. R., and Liu, Y. (2014). Fast multivariate spatio-temporal analysis via low rank tensor learning. In *Advances in neural information processing systems*, pages 3491–3499.

Bell, M. L., Dominici, F., Ebisu, K., Zeger, S. L., and Samet, J. M. (2007). Spatial and temporal variation in $PM_{2.5}$ chemical composition in the United States for health effects studies. *Environmental Health Perspectives*, 115(7):989.

Bell, M. L., Ebisu, K., Peng, R. D., Samet, J. M., and Dominici, F. (2009). Hospital admissions and chemical composition of fine particle air pollution. *American Journal of Respiratory and Critical Care Medicine*, 179(12):1115–1120.

Bergen, S., Sheppard, L., Sampson, P. D., Kim, S.-Y., Richards, M., Vedal, S., Kaufman, J. D., and Szpiro, A. A. (2013). A national prediction model for $PM_{2.5}$ component exposures and measurement error–corrected health effect inference. *Environmental Health Perspectives*, 121(9):1017.

Bose, M., Larson, T., and Szpiro, A. A. (2018). Adaptive predictive principal components for modeling multivariate air pollution. *Environmetrics*, 29(8).

Brauer, M., Hoek, G., van Vliet, P., Meliefste, K., Fischer, P., Gehring, U., Heinrich, J., Cyrys, J., Bellander, T., Lewne, M., and Brunekreef, B. (2003). Estimating long-term

average particulate air pollution concentrations: application of traffic indicators and geographic information systems. *Epidemiology*, 14(2):228–239.

Brook, R. D., Franklin, B., Cascio, W., Hong, Y., Howard, G., Lipsett, M., Luepker, R., Mittleman, M., Samet, J., Smith, S. C., and Tager, I. (2004). Air pollution and cardiovascular disease. *Circulation*, 109(21):2655–2671.

Cabral, R., De la Torre, F., Costeira, J. P., and Bernardino, A. (2014). Matrix completion for weakly-supervised multi-label image classification. *IEEE transactions on pattern analysis and machine intelligence*, 37(1):121–135.

Cai, J.-F., Candès, E. J., and Shen, Z. (2010). A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982.

Candès, E. J. and Recht, B. (2009). Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717.

Chan, S. H., Van Hee, V. C., Bergen, S., Szpiro, A. A., DeRoo, L. A., London, S. J., Marshall, J. D., Kaufman, J. D., and Sandler, D. P. (2015). Long-term air pollution exposure and blood pressure in the Sister Study. *Environmental Health Perspectives*, 123(10):951.

Clements, A. L., Fraser, M. P., Upadhyay, N., Herckes, P., Sundblom, M., Lantz, J., and Solomon, P. A. (2017). Source identification of coarse particles in the Desert Southwest, USA using positive matrix ractorization. *Atmospheric Pollution Research*, 8(5):873–884.

Cook, R. D., Li, B., and Chiaromonte, F. (2010). Envelope models for parsimonious and efficient multivariate linear regression. *Statistica Sinica*, pages 927–960.

Coutant, B. W., Engel-Cox, J., and Swinton, K. E. (2003). Compilation of existing studies on source apportionment for $PM_{2.5}$. *Technical report of the Office of Air Quality planning and Standards, Washington, D.C.: USEPA*.

Crouse, D. L., Goldberg, M. S., Ross, N. A., Chen, H., and Labrèche, F. (2010). Postmenopausal breast cancer is associated with exposure to traffic-related air pollution in

Montreal, Canada: a case–control study. *Environmental Health Perspectives*, 118(11):1578.

Dai, L., Zanobetti, A., Koutrakis, P., and Schwartz, J. D. (2014). Associations of fine particulate matter species with mortality in the United States: a multicity time-series analysis. *Environmental Health Perspectives*, 122(8):837.

Dominici, F., Peng, R. D., Barr, C. D., and Bell, M. L. (2010). Protecting human health from air pollution: shifting from a single-pollutant to a multi-pollutant approach. *Epidemiology (Cambridge, Mass.)*, 21(2):187.

Dominici, F., Sheppard, L., and Clyde, M. (2003). Health effects of air pollution: A statistical review. *International Statistical Review*, 71(2):243–276.

Donoho, D. L., Johnstone, I. M., Kerkyacharian, G., and Picard, D. (1995). Wavelet shrinkage: asymptopia? *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(2):301–337.

Franklin, M., Koutrakis, P., and Schwartz, J. (2008). The role of particle composition on the association between $PM_{2.5}$ and mortality. *Epidemiology (Cambridge, Mass.)*, 19(5):680.

Gandy, S., Recht, B., and Yamada, I. (2011). Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 27(2):025010.

Gold, D. R., Litonjua, A., Schwartz, J., Lovett, E., Larson, A., Nearing, B., Allen, G., Verrier, M., Cherry, R., and Verrier, R. (2000). Ambient pollution and heart rate variability. *Circulation*, 101(11):1267–1273.

Hand, J., Schichtel, B., Pitchford, M., Malm, W., and Frank, N. (2012). Seasonal composition of remote and urban fine particulate matter in the United States. *Journal of Geophysical Research: Atmospheres*, 117(D5).

Harman, H. H. (1976). *Modern factor analysis*. University of Chicago Press.

Hogan, J. W. and Tchernis, R. (2004). Bayesian factor analysis for spatially correlated data, with application to summarizing area-level material deprivation from census data. *Journal*

of the American Statistical Association, 99(466):314–324.

Hsu, C.-Y., Chiang, H.-C., Chen, M.-J., Chuang, C.-Y., Tsen, C.-M., Fang, G.-C., Tsai, Y.-I., Chen, N.-T., Lin, T.-Y., Lin, S.-L., and Chen, Y.-C. (2017). Ambient $PM_{2.5}$ in the residential area near industrial complexes: Spatiotemporal variation, source apportionment, and health impact. *Science of the Total Environment*, 590:204–214.

Hunter, D. R. and Lange, K. (2004). A tutorial on mm algorithms. *The American Statistician*, 58(1):30–37.

Jandarov, R. A., Sheppard, L. A., Sampson, P. D., and Szpiro, A. A. (2017). A novel principal component analysis for spatially misaligned multivariate air pollution data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 66(1):3–28.

Ji, H., Liu, C., Shen, Z., and Xu, Y. (2010). Robust video denoising using low rank matrix completion. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1791–1798. IEEE.

Jolliffe, I. T. (1986). Principal component analysis and factor analysis. In *Principal component analysis*, pages 115–128. Springer.

Kammann, E. and Wand, M. P. (2003). Geoadditive models. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 52(1):1–18.

Kaufman, J. D., Adar, S. D., Barr, R. G., Budoff, M., Burke, G. L., Curl, C. L., Daviglus, M. L., Roux, A. V. D., Gassett, A. J., Jacobs, D. R. J., Kronmal, R., Larson, T. V., Navas-Acien, A., Olives, C., Sampson, P. D., Sheppard, L., Siscovick, D. S., Stein, J. H., Szpiro, A. A., and Watson, K. E. (2016). Association between air pollution and coronary artery calcification within six metropolitan areas in the USA (the Multi-Ethnic Study of Atherosclerosis and Air Pollution): a longitudinal cohort study. *The Lancet*, 388(10045):696–704.

Keller, J. P., Drton, M., Larson, T., Kaufman, J. D., Sandler, D. P., and Szpiro, A. A. (2017). Covariate-adaptive clustering of exposures for air pollution epidemiology cohorts.

The Annals of Applied Statistics, 11(1):93.

Keller, J. P., Larson, T. V., Austin, E., Barr, R. G., Sheppard, L., Vedal, S., Kaufman, J. D., and Szpiro, A. A. (2018). Pollutant composition modification of the effect of air pollution on progression of coronary artery calcium: The Multi-Ethnic Study of Atherosclerosis. *Environmental Epidemiology*, 2(3):e024.

Kioumourtzoglou, M.-A., Austin, E., Koutrakis, P., Dominici, F., Schwartz, J., and Zanobetti, A. (2015). $PM_{2.5}$ and survival among older adults: effect modification by particulate composition. *Epidemiology (Cambridge, Mass.)*, 26(3):321.

Kolda, T. G. and Bader, B. W. (2009). Tensor decompositions and applications. *SIAM review*, 51(3):455–500.

Kotchenruther, R. A. (2017). The effects of marine vessel fuel sulfur regulations on ambient $PM_{2.5}$ at coastal and near coastal monitoring sites in the US. *Atmospheric Environment*, 151:52–61.

Krall, J. R., Anderson, G. B., Dominici, F., Bell, M. L., and Peng, R. D. (2013). Short-term exposure to particulate matter constituents and mortality in a national study of US urban communities. *Environmental Health Perspectives*, 121(10):1148.

Künzli, N., Jerrett, M., Mack, W. J., Beckerman, B., LaBree, L., Gilliland, F., Thomas, D., Peters, J., and Hodis, H. N. (2005). Ambient air pollution and atherosclerosis in Los Angeles. *Environmental Health Perspectives*, 113(2):201.

Li, G., Yang, D., Nobel, A. B., and Shen, H. (2016). Supervised singular value decomposition and its asymptotic properties. *Journal of Multivariate Analysis*, 146:7–17.

Lindström, J., Szpiro, A. A., Sampson, P. D., Oron, A. P., Richards, M., Larson, T. V., and Sheppard, L. (2014). A flexible spatio-temporal model for air pollution with spatial and spatio-temporal covariates. *Environmental and ecological statistics*, 21(3):411–433.

Liu, J., Musialski, P., Wonka, P., and Ye, J. (2013). Tensor completion for estimating missing

values in visual data. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):208–220.

Liu, X., Wall, M. M., and Hodges, J. S. (2005). Generalized spatial structural equation models. *Biostatistics*, 6(4):539–557.

Ljungman, P. L., Wilker, E. H., Rice, M. B., Austin, E., Schwartz, J., Gold, D. R., Koutrakis, P., Benjamin, E. J., Vita, J. A., Mitchell, G. F., Vasan, R. S., Hamburg, N. M., and Mittleman, M. A. (2016). The impact of multi-pollutant clusters on the association between fine particulate air pollution and microvascular function. *Epidemiology (Cambridge, Mass.)*, 27(2):194.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

Mazumder, R., Hastie, T., and Tibshirani, R. (2010). Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research*, 11(Aug):2287–2322.

Miller, K. A., Siscovick, D. S., Sheppard, L., Shepherd, K., Sullivan, J. H., Anderson, G. L., and Kaufman, J. D. (2007). Long-term exposure to air pollution and incidence of cardiovascular events in women. *New England Journal of Medicine*, 2007(356):447–458.

Olives, C., Sheppard, L., Lindström, J., Sampson, P. D., Kaufman, J. D., and Szpiro, A. A. (2014). Reduced-rank spatio-temporal modeling of air pollution concentrations in the multi-ethnic study of atherosclerosis and air pollution. *The annals of applied statistics*, 8(4):2509.

Ostro, B., Tobias, A., Querol, X., Alastuey, A., Amato, F., Pey, J., Pérez, N., and Sunyer, J. (2011). The effects of particulate matter sources on daily mortality: a case-crossover study of Barcelona, Spain. *Environmental Health Perspectives*, 119(12):1781.

Paatero, P. and Tapper, U. (1994). Positive matrix factorization: A non-negative factor

model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126.

Pascal, M., Falq, G., Wagner, V., Chatignoux, E., Corso, M., Blanchard, M., Host, S., Pascal, L., and Larrieu, S. (2014). Short-term impacts of particulate matter (PM_{10}, $PM_{10-2.5}$, $PM_{2.5}$) on mortality in nine French cities. *Atmospheric Environment*, 95:175–184.

Pitchford, M. L., Poirot, R. L., Schichtel, B. A., and Malm, W. C. (2009). Characterization of the winter midwestern particulate nitrate bulge. *Journal of the Air & Waste Management Association*, 59(9):1061–1069.

Pope III, C. A., Burnett, R. T., Thun, M. J., Calle, E. E., Krewski, D., Ito, K., and Thurston, G. D. (2002). Lung cancer, cardiopulmonary mortality, and long-term exposure to fine particulate air pollution. *Journal of the American Medical Association*, 287(9):1132–1141.

Rockafellar, R. T. (1976). Monotone operators and the proximal point algorithm. *SIAM journal on control and optimization*, 14(5):877–898.

Ruan, W., Xu, P., Sheng, Q. Z., Falkner, N. J., Li, X., and Zhang, W. E. (2017). Recovering missing values from corrupted spatio-temporal sensory data via robust low-rank tensor completion. In *International Conference on Database Systems for Advanced Applications*, pages 607–622. Springer.

Sampson, P. D., Szpiro, A. A., Sheppard, L., Lindström, J., and Kaufman, J. D. (2011). Pragmatic estimation of a spatio-temporal air quality model with irregular monitoring data. *Atmospheric Environment*, 45(36):6593–6606.

Sarnat, J. A., Marmur, A., Klein, M., Kim, E., Russell, A. G., Sarnat, S. E., Mulholland, J. A., Hopke, P. K., and Tolbert, P. E. (2008). Fine particle sources and cardiorespiratory morbidity: an application of chemical mass balance and factor analytical source-apportionment methods. *Environmental Health Perspectives*, 116(4):459.

Shacklette, H. T. and Boerngen, J. G. (1984). Element concentrations in soils and other surficial materials of the conterminous United States. *Geological Survey Professional Paper*

1270, Washington, D.C.

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.

Shin, P. J., Larson, P. E., Ohliger, M. A., Elad, M., Pauly, J. M., Vigneron, D. B., and Lustig, M. (2014). Calibrationless parallel imaging reconstruction based on structured low-rank matrix completion. *Magnetic resonance in medicine*, 72(4):959–970.

Signoretto, M., De Lathauwer, L., and Suykens, J. A. (2010). Nuclear norms for tensors and their use for convex multilinear estimation. *Submitted to Linear Algebra and Its Applications*, 43.

Szpiro, A. A. and Paciorek, C. J. (2013). Measurement error in two-stage analyses, with application to air pollution epidemiology. *Environmetrics*, 24(8):501–517.

Szpiro, A. A., Paciorek, C. J., and Sheppard, L. (2011). Does more accurate exposure prediction necessarily improve health effect estimates? *Epidemiology (Cambridge, Mass.)*, 22(5):680.

Szpiro, A. A., Sampson, P. D., Sheppard, L., Lumley, T., Adar, S. D., and Kaufman, J. D. (2010). Predicting intra-urban variation in air pollution concentrations with complex spatio-temporal dependencies. *Environmetrics*, 21(6):606–631.

Thurston, G. D., Ito, K., and Lall, R. (2011). A source apportionment of US fine particulate matter air pollution. *Atmospheric Environment*, 45(24):3924–3936.

Tian, L., Zeng, Q., Dong, W., Guo, Q., Wu, Z., Pan, X., Li, G., and Liu, Y. (2017). Addressing the source contribution of $PM_{2.5}$ on mortality: an evaluation study of its impacts on excess mortality in China. *Environmental Research Letters*, 12(10):104016.

Tipping, M. E. and Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622.

Tolbert, P. E., Klein, M., Peel, J. L., Sarnat, S. E., and Sarnat, J. A. (2007). Multipollutant

modeling issues in a study of ambient air quality and emergency department visits in Atlanta. *Journal of Exposure Science and Environmental Epidemiology*, 17(S2):S29.

Tong, D. Q., Dan, M., Wang, T., and Lee, P. (2012). Long-term dust climatology in the western United States reconstructed from routine aerosol ground monitoring. *Atmospheric Chemistry and Physics*, 12(11):5189–5205.

Vu, P. T., Larson, T. V., and Szpiro, A. A. (2019). Probabilistic predictive principal component analysis for spatially-misaligned and high-dimensional air pollution data with missing observations. *arXiv preprint arXiv:1905.00393*.

Wang, F. and Wall, M. M. (2003). Generalized common spatial factor model. *Biostatistics*, 4(4):569–582.

Wang, H., Nie, F., and Huang, H. (2014). Low-rank tensor completion with spatio-temporal consistency. In *AAAI*, pages 2846–2852.

Wang, Y., Shi, L., Lee, M., Liu, P., Di, Q., Zanobetti, A., and Schwartz, J. D. (2017). Long-term exposure to $PM_{2.5}$ and mortality among older adults in the southeastern US. *Epidemiology (Cambridge, Mass.)*, 28(2):207–214.

Xie, K., Ning, X., Wang, X., Xie, D., Cao, J., Xie, G., and Wen, J. (2017). Recover corrupted data in sensor networks: A matrix completion solution. *IEEE Transactions on Mobile Computing*, 16(5):1434–1448.

Xie, K., Wang, L., Wang, X., Xie, G., Wen, J., and Zhang, G. (2016). Accurate recovery of internet traffic data: A tensor completion approach. In *INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, IEEE*, pages 1–9. IEEE.

Yang, Y., Ma, J., and Osher, S. (2013). Seismic data reconstruction via matrix completion. *Inverse Problems and Imaging*, 7(4):1379–1392.

Zanobetti, A., Austin, E., Coull, B. A., Schwartz, J., and Koutrakis, P. (2014). Health effects of multi-pollutant profiles. *Environment International*, 71:13–19.

Zhou, H., Zhang, D., Xie, K., and Chen, Y. (2015). Spatio-temporal tensor completion for imputing missing internet traffic data. In *Computing and Communications Conference (IPCCC), 2015 IEEE 34th International Performance*, pages 1–7. IEEE.

Zhu, J., Eickhoff, J., and Yan, P. (2005). Generalized linear latent variable models for repeated measures of spatially correlated multivariate data. *Biometrics*, 61(3):674–683.

Appendix A

APPENDIX FOR CHAPTER 2

A.1 The ProPrPCA-Krige model and algorithm

A.1.1 The model

The ProPrPCA-Krige assumes that $\boldsymbol{X} = \sum_{l=1}^{q} \left(\boldsymbol{u}_l \boldsymbol{v}_l^{\mathsf{T}} + \boldsymbol{E}_l\right)$ and $\boldsymbol{u}_l = \boldsymbol{R}\boldsymbol{\beta}_l + \boldsymbol{\eta}_l$, where $\boldsymbol{u}_l \in \mathbb{R}^n$, $\boldsymbol{v}_l \in \mathbb{R}^p$, $\boldsymbol{\beta}_l \in \mathbb{R}^k$, $E_{ij} \sim \mathcal{N}(0, \gamma^2)$, and $\boldsymbol{\eta}_l \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}(\boldsymbol{\xi}_l))$. For notation simplification, we are going to ignore the subscript l for the following mathematical derivation. The parameter estimation is the same for all PC. For each PC, the model becomes:

$$\boldsymbol{X} = \boldsymbol{u}\boldsymbol{v}^{\mathsf{T}} + \boldsymbol{E},$$
$$\boldsymbol{u} = \boldsymbol{R}\boldsymbol{\beta} + \boldsymbol{\eta},$$

Denote $\Theta = \{\boldsymbol{v}, \boldsymbol{\beta}, \gamma^2, \boldsymbol{\xi}\}$ as the collection of the model parameters. To solve for Θ, we first rewrite the model in the conventional vectorized version. Denote $\boldsymbol{W} \in \mathbb{R}^N$, for $N = np$, as the vectorized version of $\boldsymbol{X}$, i.e.

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{W}_1 \\ \vdots \\ \boldsymbol{W}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{X}_{1:} \\ \vdots \\ \boldsymbol{X}_{n:} \end{bmatrix},$$

where $\boldsymbol{X}_{i:}$ is the i-th row of $\boldsymbol{X}$. The model assumes that $\boldsymbol{W}_i = \boldsymbol{X}_{i:} = u_i\boldsymbol{v} + \boldsymbol{\epsilon}_i$, for $i = 1, ..., n$. Here $\boldsymbol{v}$ represents the transformation from the latent variable space to the multi-pollutant exposure space, and $\boldsymbol{\epsilon}_i$'s are i.i.d. Gaussian noises distributed with mean zero and variance γ^2. The full model can then be written as $\boldsymbol{W} = \boldsymbol{V}\boldsymbol{u} + \boldsymbol{\epsilon}$, where $\boldsymbol{V} = \boldsymbol{I}_n \otimes \boldsymbol{v}$ and $\otimes$ denotes the Kronecker product. The model also assumes that the latent variables

are normally distributed with a spatial mean model and covariance structure. That is, $\boldsymbol{u} \sim \mathcal{N}(\boldsymbol{R\beta}, \boldsymbol{\Sigma}(\boldsymbol{\xi}))$. In this paper, we assume $\boldsymbol{\Sigma}(\boldsymbol{\xi})$ has an exponential structure with no nugget effect. For identifiability, we assume that $\|\boldsymbol{v}\|_2 = 1$. When every element of $\boldsymbol{X}$ is observed, we have the following hierarchical model:

$$\begin{aligned} \boldsymbol{W} \mid \boldsymbol{u} &\sim \mathcal{N}\left(\boldsymbol{V}\boldsymbol{u}, \gamma^2 \boldsymbol{I}_N\right), \\ \boldsymbol{u} &\sim \mathcal{N}\left(\boldsymbol{R\beta}, \boldsymbol{\Sigma}(\boldsymbol{\xi})\right). \end{aligned}$$

In practice, not all pollutants are measured at every monitoring location. Denote $\boldsymbol{W}_o \in \mathbb{R}^{N_o}$ as the collection of all observed elements of $\boldsymbol{W}$, and $\boldsymbol{W}_m \in \mathbb{R}^{N_m}$ as the collection of all missing entries, where $N_o + N_m = N$. Algebraically, there exists a linear transformation $\boldsymbol{G}$ such that

$$\boldsymbol{GW} = \begin{bmatrix} \boldsymbol{G}_o \\ \boldsymbol{G}_m \end{bmatrix} \boldsymbol{W} = \begin{bmatrix} \boldsymbol{W}_o \\ \boldsymbol{W}_m \end{bmatrix},$$

where $\boldsymbol{G}_o \in \mathbb{R}^{N_o \times N}$ and $\boldsymbol{G}_m \in \mathbb{R}^{N_m \times N}$. Each row and column of $\boldsymbol{G}$ contains exactly one element of one and $(N-1)$ zeros. Thus by construction, $\boldsymbol{G}_o$ and $\boldsymbol{G}_m$ are both full row rank, as well as $\boldsymbol{G}_o\boldsymbol{G}_o^{\mathsf{T}} = \boldsymbol{I}_{N_o}$ and $\boldsymbol{G}_m\boldsymbol{G}_m^{\mathsf{T}} = \boldsymbol{I}_{N_m}$. The hierarchical model for the observed elements become:

$$\begin{aligned} \boldsymbol{W}_o \mid \boldsymbol{u} &\sim \mathcal{N}\left(\boldsymbol{G}_o\boldsymbol{V}\boldsymbol{u}, \gamma^2 \boldsymbol{I}_{N_o}\right), \\ \boldsymbol{u} &\sim \mathcal{N}\left(\boldsymbol{R\beta}, \boldsymbol{\Sigma}(\boldsymbol{\xi})\right). \end{aligned}$$

A.1.2 Estimation of model parameters when monitoring data is complete

Our approach to estimate the model parameters is similar to the EM algorithm employed by Tipping and Bishop (1999). We consider the latent variable $\boldsymbol{u}$ to be the "missing" portion. Thus the "complete" data consists of the observed data $\boldsymbol{W}$, and the latent variable $\boldsymbol{u}$. The goal is then to maximize the joint likelihood of $(\boldsymbol{W}, \boldsymbol{u})$, i.e. $\mathcal{L} = f(\boldsymbol{W}, \boldsymbol{u}) = f(\boldsymbol{W}|\boldsymbol{u})f(\boldsymbol{u})$.

The "complete" log-likelihood, up to a constant, is:

$$\ell_c = -\frac{N}{2}\log\gamma^2 - \frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2\gamma^2}(\boldsymbol{W}-\boldsymbol{V}\boldsymbol{u})^\mathsf{T}(\boldsymbol{W}-\boldsymbol{V}\boldsymbol{u}) - \frac{1}{2}(\boldsymbol{u}-\boldsymbol{R}\boldsymbol{\beta})^\mathsf{T}\boldsymbol{\Sigma}^{-1}(\boldsymbol{u}-\boldsymbol{R}\boldsymbol{\beta}).$$

Because this log-likelihood involves $\boldsymbol{u}$ which is unobserved, in each E step, we find the expectation of ℓ_c with respect to the conditional distribution of $\boldsymbol{u}|\boldsymbol{W}$. We first derive this distribution as follows:

$$\begin{aligned} f(\boldsymbol{u}|\boldsymbol{W}) &\propto \exp\left[-\frac{1}{2\gamma^2}(\boldsymbol{W}-\boldsymbol{V}\boldsymbol{u})^\mathsf{T}(\boldsymbol{W}-\boldsymbol{V}\boldsymbol{u})\right] \times \exp\left[-\frac{1}{2}(\boldsymbol{u}-\boldsymbol{R}\boldsymbol{\beta})^\mathsf{T}\boldsymbol{\Sigma}^{-1}(\boldsymbol{u}-\boldsymbol{R}\boldsymbol{\beta})\right] \\ &\propto \exp\left\{-\frac{1}{2}\left[\boldsymbol{u}^\mathsf{T}\left(\frac{1}{\gamma^2}\boldsymbol{V}^\mathsf{T}\boldsymbol{V}+\boldsymbol{\Sigma}^{-1}\right)\boldsymbol{u} - \boldsymbol{u}^\mathsf{T}\left(\frac{1}{\gamma^2}\boldsymbol{V}^\mathsf{T}\boldsymbol{W}+\boldsymbol{\Sigma}^{-1}\boldsymbol{R}\boldsymbol{\beta}\right)\right.\right. \\ &\qquad\left.\left. - \left(\frac{1}{\gamma^2}\boldsymbol{W}^\mathsf{T}\boldsymbol{V}+\boldsymbol{\beta}^\mathsf{T}\boldsymbol{R}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\right)\boldsymbol{u}\right]\right\}. \end{aligned}$$

Thus the distribution of $\boldsymbol{u}|\boldsymbol{W}$ is $\mathcal{N}(\boldsymbol{M},\boldsymbol{S})$, where:

$$\begin{aligned} \boldsymbol{S} &= \left(\frac{1}{\gamma^2}\boldsymbol{V}^\mathsf{T}\boldsymbol{V}+\boldsymbol{\Sigma}^{-1}\right)^{-1}, \\ \boldsymbol{M} &= \boldsymbol{S}\left(\frac{1}{\gamma^2}\boldsymbol{V}^\mathsf{T}\boldsymbol{W}+\boldsymbol{\Sigma}^{-1}\boldsymbol{R}\boldsymbol{\beta}\right). \end{aligned}$$

We can further simplify these expressions by noticing that $\boldsymbol{V}^\mathsf{T}\boldsymbol{V} = \boldsymbol{I}_n$ and $\boldsymbol{V}^\mathsf{T}\boldsymbol{W} = \boldsymbol{X}\boldsymbol{v}$, using properties of Kronecker products. The conditional covariance and mean become

$$\begin{aligned} \boldsymbol{S} &= \left(\frac{1}{\gamma^2}\boldsymbol{I}_n+\boldsymbol{\Sigma}^{-1}\right)^{-1}, \\ \boldsymbol{M} &= \boldsymbol{S}\left(\frac{1}{\gamma^2}\boldsymbol{X}\boldsymbol{v}+\boldsymbol{\Sigma}^{-1}\boldsymbol{R}\boldsymbol{\beta}\right). \end{aligned}$$

Given a current estimate $\tilde{\Theta}$, the expectation of ℓ_c with respect to $\boldsymbol{u}|\boldsymbol{W}$ is:

$$E\left[\ell_c \mid \boldsymbol{W}, \tilde{\Theta}\right] = -\frac{N}{2}\log\gamma^2 - \frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2\gamma^2}E\left[(\boldsymbol{W}-\boldsymbol{V}\boldsymbol{u})^{\mathsf{T}}(\boldsymbol{W}-\boldsymbol{V}\boldsymbol{u}) \mid \boldsymbol{W}, \tilde{\Theta}\right] - \frac{1}{2}E\left[(\boldsymbol{u}-\boldsymbol{R}\boldsymbol{\beta})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{u}-\boldsymbol{R}\boldsymbol{\beta}) \mid \boldsymbol{W}, \tilde{\Theta}\right] \quad \text{(A.1)}$$

The conditional distribution $\boldsymbol{u}|\boldsymbol{W}, \tilde{\Theta}$ is $\mathcal{N}(\tilde{\boldsymbol{M}}, \tilde{\boldsymbol{S}})$, where $\tilde{\boldsymbol{M}} = \boldsymbol{M}(\tilde{\Theta})$ and $\tilde{\boldsymbol{S}} = \boldsymbol{S}(\tilde{\Theta})$. This implies that

$$\boldsymbol{V}\boldsymbol{u} - \boldsymbol{W} \sim \mathcal{N}\left(\boldsymbol{V}\tilde{\boldsymbol{M}} - \boldsymbol{W}, \boldsymbol{V}\tilde{\boldsymbol{S}}\boldsymbol{V}^{\mathsf{T}}\right),$$
$$\boldsymbol{u} - \boldsymbol{R}\boldsymbol{\beta} \sim \mathcal{N}\left(\tilde{\boldsymbol{M}} - \boldsymbol{R}\boldsymbol{\beta}, \tilde{\boldsymbol{S}}\right).$$

Thus the first expectation term of (A.1) is

$$\begin{aligned} E\left[(\boldsymbol{W}-\boldsymbol{V}\boldsymbol{u})^{\mathsf{T}}(\boldsymbol{W}-\boldsymbol{V}\boldsymbol{u}) \mid \boldsymbol{W}, \tilde{\Theta}\right] &= \mathrm{Tr}\left(\boldsymbol{V}\tilde{\boldsymbol{S}}\boldsymbol{V}^{\mathsf{T}}\right) + (\boldsymbol{V}\tilde{\boldsymbol{M}} - \boldsymbol{W})^{\mathsf{T}}(\boldsymbol{V}\tilde{\boldsymbol{M}} - \boldsymbol{W}) \\ &= \mathrm{Tr}(\tilde{\boldsymbol{S}}) + (\boldsymbol{V}\tilde{\boldsymbol{M}} - \boldsymbol{W})^{\mathsf{T}}(\boldsymbol{V}\tilde{\boldsymbol{M}} - \boldsymbol{W}). \end{aligned}$$

The second expectation term of (A.1) is

$$E\left[(\boldsymbol{u}-\boldsymbol{R}\boldsymbol{\beta})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{u}-\boldsymbol{R}\boldsymbol{\beta}) \mid \boldsymbol{W}, \tilde{\Theta}\right] = \mathrm{Tr}\left(\boldsymbol{\Sigma}^{-1}\tilde{\boldsymbol{S}}\right) + (\tilde{\boldsymbol{M}} - \boldsymbol{R}\boldsymbol{\beta})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\tilde{\boldsymbol{M}} - \boldsymbol{R}\boldsymbol{\beta})$$

Hence (A.1) can be simplified as

$$\begin{aligned} E\left[\ell_c \mid \boldsymbol{W}, \tilde{\Theta}\right] = &-\frac{N}{2}\log\gamma^2 - \frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2\gamma^2}\mathrm{Tr}(\tilde{\boldsymbol{S}}) - \frac{1}{2\gamma^2}(\boldsymbol{V}\tilde{\boldsymbol{M}} - \boldsymbol{W})^{\mathsf{T}}(\boldsymbol{V}\tilde{\boldsymbol{M}} - \boldsymbol{W}) \\ &- \frac{1}{2}\mathrm{Tr}\left(\boldsymbol{\Sigma}^{-1}\tilde{\boldsymbol{S}}\right) - \frac{1}{2}(\tilde{\boldsymbol{M}} - \boldsymbol{R}\boldsymbol{\beta})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\tilde{\boldsymbol{M}} - \boldsymbol{R}\boldsymbol{\beta}). \end{aligned}$$

To solve for $\boldsymbol{v}$, we effectively maximize $\{-\frac{1}{2\gamma^2}(\boldsymbol{V}\tilde{\boldsymbol{M}} - \boldsymbol{W})^{\mathsf{T}}(\boldsymbol{V}\tilde{\boldsymbol{M}} - \boldsymbol{W})\}$, which can be rewrite as follows:

$$
\begin{aligned}
&-\frac{1}{2\gamma^2}(\boldsymbol{V}\tilde{\boldsymbol{M}}-\boldsymbol{W})^{\mathsf{T}}(\boldsymbol{V}\tilde{\boldsymbol{M}}-\boldsymbol{W}) \\
=&-\frac{1}{2\gamma^2}\left\|\boldsymbol{W}-\boldsymbol{V}\tilde{\boldsymbol{M}}\right\|_2^2=-\frac{1}{2\gamma^2}\left\|\begin{bmatrix}\boldsymbol{X}_{1:}\\ \boldsymbol{X}_{2:}\\ \vdots\\ \boldsymbol{X}_{n:}\end{bmatrix}-\begin{bmatrix}\boldsymbol{v} & \boldsymbol{0} & ... & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{v} & ... & \boldsymbol{0}\\ \vdots & \vdots & \ddots & \vdots\\ \boldsymbol{0} & \boldsymbol{0} & ... & \boldsymbol{v}\end{bmatrix}\begin{bmatrix}\tilde{M}_1\\ \tilde{M}_2\\ \vdots\\ \tilde{M}_n\end{bmatrix}\right\|_2^2 \\
=&-\frac{1}{2\gamma^2}\sum_{i=1}^{n}\left\|\boldsymbol{X}_{i.}-\tilde{M}_i\boldsymbol{v}\right\|_2^2=-\frac{1}{2\gamma^2}\sum_{i=1}^{n}\sum_{j=1}^{p}\left(X_{ij}-\tilde{M}_i v_j\right)^2.
\end{aligned}
$$

Differentiate this expression with respect to each v_j, we get

$$\breve{v}_j=\frac{\sum_{i=1}^{n}X_{ij}\tilde{M}_i}{\sum_{i=1}^{n}\tilde{M}_i^2}=\frac{\sum_{i=1}^{n}X_{ij}\tilde{M}_i}{\left\|\tilde{\boldsymbol{M}}\right\|_2^2}.$$

Thus, the solution for $\boldsymbol{v}$ can be written in closed-form as

$$\hat{\boldsymbol{v}}=\frac{\breve{\boldsymbol{v}}}{\|\breve{\boldsymbol{v}}\|_2},\qquad \text{where } \breve{\boldsymbol{v}}=\frac{\boldsymbol{X}^{\mathsf{T}}\tilde{\boldsymbol{M}}}{\left\|\tilde{\boldsymbol{M}}\right\|_2^2}$$

To solve for γ^2, we maximize $\Big\{-\frac{N}{2}\log\gamma^2-\frac{1}{2\gamma^2}\mathrm{Tr}(\tilde{\boldsymbol{S}})-\frac{1}{2\gamma^2}(\boldsymbol{V}\tilde{\boldsymbol{M}}-\boldsymbol{W})^{\mathsf{T}}(\boldsymbol{V}\tilde{\boldsymbol{M}}-\boldsymbol{W})\Big\}$. The closed-form solution for γ^2 is simply

$$\hat{\gamma}^2=\frac{1}{N}\left[\mathrm{Tr}(\tilde{\boldsymbol{S}})+\left\|\boldsymbol{V}\tilde{\boldsymbol{M}}-\boldsymbol{W}\right\|_{2\cdot}^2\right]$$

The solution for $\boldsymbol{\beta}$ by maximizing $\Big\{-\frac{1}{2}(\tilde{\boldsymbol{M}}-\boldsymbol{R}\boldsymbol{\beta})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\tilde{\boldsymbol{M}}-\boldsymbol{R}\boldsymbol{\beta})\Big\}$ is

$$\hat{\boldsymbol{\beta}}=(\boldsymbol{R}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{R})^{-1}\boldsymbol{R}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\tilde{\boldsymbol{M}}.$$

Finally, to solve for $\boldsymbol{\xi}$, we maximize $\Big\{-\frac{1}{2}\log|\boldsymbol{\Sigma}|-\frac{1}{2}\mathrm{Tr}\left(\boldsymbol{\Sigma}^{-1}\tilde{\boldsymbol{S}}\right)-\frac{1}{2}(\tilde{\boldsymbol{M}}-\boldsymbol{R}\boldsymbol{\beta})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\tilde{\boldsymbol{M}}-\boldsymbol{R}\boldsymbol{\beta})\Big\}$ numerically, where $\boldsymbol{\Sigma}$ is a function of $\boldsymbol{\xi}$. In this paper, we adopt the exponential

covariance structure for $\boldsymbol{\Sigma}$. For identifiability, we assume that $\boldsymbol{\Sigma}$ has no nugget effect.

Thus, parameter estimation of ProPrPCA-Krige with complete monitoring data can be summarized as:

Algorithm: ProPrPCA-Krige with complete monitoring data

Input $\boldsymbol{X}$, $\boldsymbol{R}$, q, and t_{max}

for l in $\{1, ..., q\}$ **do**

$\boldsymbol{X}_l \leftarrow \boldsymbol{X}_{l-1} - \hat{\boldsymbol{u}}_{l-1}\hat{\boldsymbol{v}}_{l-1}^{\mathsf{T}}$ where $\boldsymbol{X}_0 = \boldsymbol{X}$, $\hat{\boldsymbol{u}}_0 = \boldsymbol{0}$, and $\hat{\boldsymbol{v}}_0 = \boldsymbol{0}$

Initialize $\boldsymbol{v}_l^{(0)}$, $(\gamma_l^{(0)})^2$, $\boldsymbol{\beta}_l^{(0)}$, $\boldsymbol{\xi}_l^{(0)}$, and $t = 1$

$\boldsymbol{\Sigma}_l^{(0)} \leftarrow \boldsymbol{\Sigma}(\boldsymbol{\xi}_l^{(0)})$

while not converged **or** $t < t_{max}$ **do**

$\tilde{\boldsymbol{S}}_l \leftarrow \left[(\gamma_l^{(t)})^{-2}\boldsymbol{I}_n + (\boldsymbol{\Sigma}_l^{(t)})^{-1}\right]^{-1}$

$\tilde{\boldsymbol{M}}_l \leftarrow \tilde{\boldsymbol{S}}_l \left[(\gamma_l^{(t)})^{-2}\boldsymbol{X}_l\boldsymbol{v}_l^{(t)} + (\boldsymbol{\Sigma}_l^{(t)})^{-1}\boldsymbol{R}\boldsymbol{\beta}_l^{(t)}\right]$

$\tilde{\boldsymbol{v}}_l \leftarrow \boldsymbol{X}_l^{\mathsf{T}}\tilde{\boldsymbol{M}}_l / \left\|\tilde{\boldsymbol{M}}_l\right\|_2^2$

$\boldsymbol{v}_l^{(t+1)} \leftarrow \tilde{\boldsymbol{v}}_l / \|\tilde{\boldsymbol{v}}_l\|_2$

$(\gamma_l^{(t+1)})^2 \leftarrow (np)^{-1}\left[\mathrm{Tr}(\tilde{\boldsymbol{S}}_l) + \left\|(\boldsymbol{I}_n \otimes \boldsymbol{v}_l^{(t+1)})\tilde{\boldsymbol{M}}_l - \mathrm{vec}(\boldsymbol{X}_l)\right\|_2^2\right]$

$\boldsymbol{\xi}_l^{(t+1)} \leftarrow \underset{\boldsymbol{\xi}_l}{\arg\max}\left\{ -\log|\boldsymbol{\Sigma}_l| - \mathrm{Tr}\left(\boldsymbol{\Sigma}_l^{-1}\tilde{\boldsymbol{S}}_l\right) - (\tilde{\boldsymbol{M}}_l - \boldsymbol{R}\boldsymbol{\beta}_l^{(t)})^{\mathsf{T}}\boldsymbol{\Sigma}_l^{-1}(\tilde{\boldsymbol{M}}_l - \boldsymbol{R}\boldsymbol{\beta}_l^{(t)})\right\}$

where $\boldsymbol{\Sigma}_l = \boldsymbol{\Sigma}(\boldsymbol{\xi}_l)$

$\boldsymbol{\beta}_l^{(t+1)} \leftarrow \left(\boldsymbol{R}^{\mathsf{T}}\hat{\boldsymbol{\Sigma}}(\boldsymbol{\xi}_l^{(t+1)})^{-1}\boldsymbol{R}\right)^{-1}\boldsymbol{R}^{\mathsf{T}}\hat{\boldsymbol{\Sigma}}(\boldsymbol{\xi}_l^{(t+1)})^{-1}\tilde{\boldsymbol{M}}_l$

$t \leftarrow t + 1$

end while

$\hat{\boldsymbol{v}}_l \leftarrow \boldsymbol{v}_l^{(t)}$, $\hat{\gamma}_l^2 \leftarrow (\gamma_l^{(t)})^2$, $\hat{\boldsymbol{\beta}}_l \leftarrow \boldsymbol{\beta}_l^{(t)}$, $\hat{\boldsymbol{\xi}}_l \leftarrow \boldsymbol{\xi}_l^{(t)}$

$\hat{\boldsymbol{u}}_l = \boldsymbol{X}_l\hat{\boldsymbol{v}}_l$

end for

Output $\{\hat{\boldsymbol{v}}_1, ..., \hat{\boldsymbol{v}}_q\}$, $\{\hat{\boldsymbol{u}}_1, ..., \hat{\boldsymbol{u}}_q\}$, $\{\hat{\boldsymbol{\beta}}_1, ..., \hat{\boldsymbol{\beta}}_q\}$, $\{\hat{\gamma}_1^2, ..., \hat{\gamma}_q^2\}$, $\{\hat{\boldsymbol{\xi}}_1, ..., \hat{\boldsymbol{\xi}}_q\}$

A.1.3 Parameter estimation and model-based imputation with missing monitoring data

The hierarchical model in the case of missing monitoring data can be written as

$$\boldsymbol{W}_o \mid \boldsymbol{u} \sim \mathcal{N}\left(\boldsymbol{G}_o\boldsymbol{V}\boldsymbol{u}, \gamma^2\boldsymbol{I}_{N_o}\right),$$

$$\boldsymbol{u} \sim \mathcal{N}\left(\boldsymbol{R}\boldsymbol{\beta}, \boldsymbol{\Sigma}(\boldsymbol{\xi})\right).$$

Thus the "complete" log-likelihood becomes

$$\ell_c = -\frac{N_o}{2}\log\gamma^2 - \frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2\gamma^2}(\boldsymbol{W}_o - \boldsymbol{G}_o\boldsymbol{V}\boldsymbol{u})^\mathsf{T}(\boldsymbol{W}_o - \boldsymbol{G}_o\boldsymbol{V}\boldsymbol{u}) - \frac{1}{2}(\boldsymbol{u} - \boldsymbol{R}\boldsymbol{\beta})^\mathsf{T}\boldsymbol{\Sigma}^{-1}(\boldsymbol{u} - \boldsymbol{R}\boldsymbol{\beta}).$$

Similar to the case with complete data, we first derive the conditional distribution of $\boldsymbol{u}|\boldsymbol{W}_o$ as follows

$$\begin{aligned} f(\boldsymbol{u}|\boldsymbol{W}_o) &\propto \exp\left[-\frac{1}{2\gamma^2}(\boldsymbol{W}_o - \boldsymbol{G}_o\boldsymbol{V}\boldsymbol{u})^\mathsf{T}(\boldsymbol{W}_o - \boldsymbol{G}_o\boldsymbol{V}\boldsymbol{u})\right] \times \exp\left[-\frac{1}{2}(\boldsymbol{u} - \boldsymbol{R}\boldsymbol{\beta})^\mathsf{T}\boldsymbol{\Sigma}^{-1}(\boldsymbol{u} - \boldsymbol{R}\boldsymbol{\beta})\right] \\ &\propto \exp\left\{ -\frac{1}{2}\left[\boldsymbol{u}^\mathsf{T}\left(\frac{1}{\gamma^2}\boldsymbol{V}^\mathsf{T}\boldsymbol{G}_o^\mathsf{T}\boldsymbol{G}_o\boldsymbol{V} + \boldsymbol{\Sigma}^{-1}\right)\boldsymbol{u} - \boldsymbol{u}^\mathsf{T}\left(\frac{1}{\gamma^2}\boldsymbol{V}^\mathsf{T}\boldsymbol{G}_o^\mathsf{T}\boldsymbol{W}_o + \boldsymbol{\Sigma}^{-1}\boldsymbol{R}\boldsymbol{\beta}\right)\right.\right. \\ &\qquad \left.\left. - \left(\frac{1}{\gamma^2}\boldsymbol{W}_o^\mathsf{T}\boldsymbol{G}_o\boldsymbol{V} + \boldsymbol{\beta}^\mathsf{T}\boldsymbol{R}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\right)\boldsymbol{u}\right]\right\}. \end{aligned}$$

Thus the distribution of $\boldsymbol{u}|\boldsymbol{W}_o$ is $\mathcal{N}(\boldsymbol{M}, \boldsymbol{S})$, where:

$$\begin{aligned} \boldsymbol{S} &= \left(\frac{1}{\gamma^2}\boldsymbol{V}^\mathsf{T}\boldsymbol{G}_o^\mathsf{T}\boldsymbol{G}_o\boldsymbol{V} + \boldsymbol{\Sigma}^{-1}\right)^{-1}, \\ \boldsymbol{M} &= \boldsymbol{S}\left(\frac{1}{\gamma^2}\boldsymbol{V}^\mathsf{T}\boldsymbol{G}_o^\mathsf{T}\boldsymbol{W}_o + \boldsymbol{\Sigma}^{-1}\boldsymbol{R}\boldsymbol{\beta}\right). \end{aligned}$$

Hence the expectation of ℓ_c with respect to the distribution of $\boldsymbol{u}|\boldsymbol{W}_o, \tilde{\Theta}$ is:

$$\begin{aligned} E\left[\ell_c \mid \boldsymbol{W}_o, \tilde{\Theta}\right] = &-\frac{N_o}{2}\log\gamma^2 - \frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2\gamma^2}\mathrm{Tr}(\boldsymbol{V}^\mathsf{T}\boldsymbol{G}_o^\mathsf{T}\boldsymbol{G}_o^\mathsf{T}\boldsymbol{V}\tilde{\boldsymbol{S}}) \\ &- \frac{1}{2\gamma^2}(\boldsymbol{G}_o\boldsymbol{V}\tilde{\boldsymbol{M}} - \boldsymbol{W}_o)^\mathsf{T}(\boldsymbol{G}_o\boldsymbol{V}\tilde{\boldsymbol{M}} - \boldsymbol{W}_o) \\ &- \frac{1}{2}\mathrm{Tr}\left(\boldsymbol{\Sigma}^{-1}\tilde{\boldsymbol{S}}\right) - \frac{1}{2}(\tilde{\boldsymbol{M}} - \boldsymbol{R}\boldsymbol{\beta})^\mathsf{T}\boldsymbol{\Sigma}^{-1}(\tilde{\boldsymbol{M}} - \boldsymbol{R}\boldsymbol{\beta}) \end{aligned} \tag{A.2}$$

The solutions for $\gamma^2, \boldsymbol{\beta}$, and $\boldsymbol{\xi}$ that maximize (A.2), given current estimates, are relatively

similar to the complete case:

$$\hat{\boldsymbol{\beta}} = \left(\boldsymbol{R}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{R}\right)^{-1}\boldsymbol{R}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\tilde{\boldsymbol{M}},$$
$$\hat{\boldsymbol{\xi}} = \arg\max_{\boldsymbol{\xi}} \left\{ -\frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}\mathrm{Tr}\left(\boldsymbol{\Sigma}^{-1}\tilde{\boldsymbol{S}}\right) - \frac{1}{2}(\tilde{\boldsymbol{M}} - \boldsymbol{R}\boldsymbol{\beta})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\tilde{\boldsymbol{M}} - \boldsymbol{R}\boldsymbol{\beta}) \right\},$$
$$\hat{\gamma}^2 = \frac{1}{N_o}\left[\mathrm{Tr}(\boldsymbol{V}^{\mathsf{T}}\boldsymbol{G}_o^{\mathsf{T}}\boldsymbol{G}_o\boldsymbol{V}\tilde{\boldsymbol{S}}) + \left\|\boldsymbol{G}_o\boldsymbol{V}\tilde{\boldsymbol{M}} - \boldsymbol{W}_o\right\|_2^2\right].$$

To solve for $\boldsymbol{v}$, we maximize a slightly different function,

$$\begin{aligned}
h(\boldsymbol{v}) &= -\frac{1}{2\gamma^2}\mathrm{Tr}(\boldsymbol{V}^{\mathsf{T}}\boldsymbol{G}_o^{\mathsf{T}}\boldsymbol{G}_o^{\mathsf{T}}\boldsymbol{V}\tilde{\boldsymbol{S}}) - \frac{1}{2\gamma^2}(\boldsymbol{G}_o\boldsymbol{V}\tilde{\boldsymbol{M}} - \boldsymbol{W}_o)^{\mathsf{T}}(\boldsymbol{G}_o\boldsymbol{V}\tilde{\boldsymbol{M}} - \boldsymbol{W}_o) \\
&= -\frac{1}{2\gamma^2}\left[\mathrm{Tr}\left(\begin{bmatrix} \sum_{j\in\Omega_{1\cdot}} v_j^2 & 0 & \dots & 0 \\ 0 & \sum_{j\in\Omega_{2\cdot}} v_j^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sum_{j\in\Omega_{n\cdot}} v_j^2 \end{bmatrix}\tilde{\boldsymbol{S}}\right) + \sum_{i=1}^{n}\sum_{j\in\Omega_{i\cdot}}\left(X_{ij} - \tilde{M}_i v_j\right)^2\right] \\
&= -\frac{1}{2\gamma^2}\left[\sum_{i=1}^{n}\sum_{j\in\Omega_{i\cdot}}\tilde{S}_{ii}v_j^2 + \sum_{i=1}^{n}\sum_{j\in\Omega_{i\cdot}}\left(X_{ij} - \tilde{M}_i v_j\right)^2\right] \\
&= -\frac{1}{2\gamma^2}\sum_{i=1}^{n}\sum_{j\in\Omega_{i\cdot}}\left[\tilde{S}_{ii}v_j^2 + \left(X_{ij} - \tilde{M}_i v_j\right)^2\right] \\
&= -\frac{1}{2\gamma^2}\sum_{i=1}^{n}\left[\tilde{S}_{ii}v_j^2 + \left(X_{ij} - \tilde{M}_i v_j\right)^2\right]\mathbf{1}_{[j\in\Omega_{i\cdot}]}.
\end{aligned}$$

Here $\Omega_{i\cdot}$ denotes the set of observed elements across the i-th row of $\boldsymbol{X}$ and $\mathbf{1}_{[.]}$ denotes the indicator function. Taking derivative of $h(\boldsymbol{v})$ with respect to each v_j and setting it equal to zero, we can find the closed-form unscaled solution

$$\breve{v}_j = \frac{\sum_{i=1}^{n} X_{ij}\tilde{M}_i\mathbf{1}_{[j\in\Omega_{i\cdot}]}}{\sum_{i=1}^{n}\left(\tilde{S}_{ii} + \tilde{M}_i^2\right)\mathbf{1}_{[j\in\Omega_{i\cdot}]}}.$$

The final solution for $\boldsymbol{v}$ is then $\hat{\boldsymbol{v}} = \frac{\breve{\boldsymbol{v}}}{\|\breve{\boldsymbol{v}}\|_2}$, where $\breve{\boldsymbol{v}} = (\breve{v}_1, ..., \breve{v}_p)$.

When some elements of the exposure data are missing, parameter estimation for each PC is based only on the observed elements $\boldsymbol{W}_o$. Estimate for PC score can then be made by projecting the model-based imputed exposure data onto the direction of $\boldsymbol{v}$. The joint distribution of $\boldsymbol{W}_o$ and $\boldsymbol{W}_m$ can be written as

$$\begin{bmatrix} \boldsymbol{W}_o \\ \boldsymbol{W}_m \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{m}_o \\ \mathbf{m}_m \end{bmatrix}, \begin{bmatrix} \mathbf{C}_{oo} & \mathbf{C}_{om} \\ \mathbf{C}_{mo} & \mathbf{C}_{mm} \end{bmatrix} \right) = \mathcal{N}(\boldsymbol{M}, \mathbf{C})$$

where $\boldsymbol{M} = \boldsymbol{G}\boldsymbol{V}\boldsymbol{R}\boldsymbol{\beta}$ and $\mathbf{C} = \gamma^2\boldsymbol{G}\boldsymbol{G}^\mathsf{T} + \boldsymbol{G}\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{V}^\mathsf{T}\boldsymbol{G}^\mathsf{T}$. The missing elements, $\boldsymbol{W}_m$, can then be imputed by the conditional mean,

$$E(\boldsymbol{W}_m \mid \boldsymbol{W}_o) = \mathbf{m}_m + \mathbf{C}_{mo}\mathbf{C}_{oo}^{-1}(\boldsymbol{W}_o - \mathbf{m}_o).$$

Thus, the parameter estimation of ProPrPCA-Krige with missing monitoring data can be summarized as:

Algorithm: ProPrPCA-Krige with missing monitoring data

Input $\boldsymbol{X}$, $\boldsymbol{R}$, $\boldsymbol{G}_o$ q, and t_{max}

for l in $\{1, ..., q\}$ **do**

$\boldsymbol{X}_l \leftarrow \boldsymbol{X}_{l-1}^{zero} - \hat{\boldsymbol{u}}_{l-1}^{zero}\hat{\boldsymbol{v}}_{l-1}^{\mathsf{T}}$ where $\boldsymbol{X}_0^{zero} = \boldsymbol{X}$ imputed with zeros, $\hat{\boldsymbol{u}}_0^{zero} = \boldsymbol{0}$, and $\hat{\boldsymbol{v}}_0 = \boldsymbol{0}$

$\boldsymbol{W}_o \leftarrow \boldsymbol{G}_o \text{vec}(\boldsymbol{X}_l)$

Initialize $\boldsymbol{v}_l^{(0)}$, $(\gamma_l^{(0)})^2$, $\boldsymbol{\beta}_l^{(0)}$, $\boldsymbol{\xi}_l^{(0)}$, and $t = 1$

$\boldsymbol{\Sigma}_l^{(0)} \leftarrow \boldsymbol{\Sigma}(\boldsymbol{\xi}_l^{(0)})$

while not converged **or** $t < t_{max}$ **do**

$\tilde{\boldsymbol{S}}_l \leftarrow \left[(\gamma_l^{(t)})^{-2}\boldsymbol{V}_l^{(t)\mathsf{T}}\boldsymbol{G}_o^{\mathsf{T}}\boldsymbol{G}_o\boldsymbol{V}_l^{(t)} + (\boldsymbol{\Sigma}_l^{(t)})^{-1}\right]^{-1}$ where $\boldsymbol{V}_l^{(t)} = \boldsymbol{I}_n \otimes \boldsymbol{v}_l^{(t)}$

$\tilde{\boldsymbol{M}}_l \leftarrow \tilde{\boldsymbol{S}}_l \left[(\gamma_l^{(t)})^{-2}\boldsymbol{V}_l^{(t)\mathsf{T}}\boldsymbol{G}_o^{\mathsf{T}}\boldsymbol{W}_o + (\boldsymbol{\Sigma}_l^{(t)})^{-1}\boldsymbol{R}\boldsymbol{\beta}_l^{(t)}\right]$

$\boldsymbol{v}_l^{(t+1)} \leftarrow \tilde{\boldsymbol{v}}_l / \|\tilde{\boldsymbol{v}}_l\|_2$ where the $j-$th element of $\tilde{\boldsymbol{v}}_l$ (for $j = 1, ..., p$) is calculated as:

$$\frac{\sum_{i=1}^{n}(\boldsymbol{X}_l)_{ij}(\tilde{\boldsymbol{M}}_l)_i \mathbf{1}_{[j\in\Omega_{i.}]}}{\sum_{i=1}^{n}\left((\tilde{\boldsymbol{S}}_l)_{ii} + (\tilde{\boldsymbol{M}}_l)_i^2\right)\mathbf{1}_{[j\in\Omega_{i.}]}}$$

$(\gamma_l^{(t+1)})^2 \leftarrow (N_o)^{-1}\left[\text{Tr}(\boldsymbol{V}_l^{(t+1)\mathsf{T}}\boldsymbol{G}_o^{\mathsf{T}}\boldsymbol{G}_o\boldsymbol{V}_l^{(t+1)}\tilde{\boldsymbol{S}}_l) + \left\|\boldsymbol{V}_l^{(t+1)\mathsf{T}}\boldsymbol{G}_o^{\mathsf{T}}\tilde{\boldsymbol{M}}_l - \boldsymbol{W}_o\right\|_2^2\right]$

where $\boldsymbol{V}_l^{(t+1)} = \boldsymbol{I}_n \otimes \boldsymbol{v}_l^{(t+1)}$

$\boldsymbol{\xi}_l^{(t+1)} \leftarrow \underset{\boldsymbol{\xi}_l}{\arg\max}\{ -\log|\boldsymbol{\Sigma}_l| - \text{Tr}\left(\boldsymbol{\Sigma}_l^{-1}\tilde{\boldsymbol{S}}_l\right) - (\tilde{\boldsymbol{M}}_l - \boldsymbol{R}\boldsymbol{\beta}_l^{(t)})^{\mathsf{T}}\boldsymbol{\Sigma}_l^{-1}(\tilde{\boldsymbol{M}}_l - \boldsymbol{R}\boldsymbol{\beta}_l^{(t)})\}$

where $\boldsymbol{\Sigma}_l = \boldsymbol{\Sigma}(\boldsymbol{\xi}_l)$

$\boldsymbol{\beta}_l^{(t+1)} \leftarrow \left(\boldsymbol{R}^{\mathsf{T}}\hat{\boldsymbol{\Sigma}}(\boldsymbol{\xi}_l^{(t+1)})^{-1}\boldsymbol{R}\right)^{-1}\boldsymbol{R}^{\mathsf{T}}\hat{\boldsymbol{\Sigma}}(\boldsymbol{\xi}_l^{(t+1)})^{-1}\tilde{\boldsymbol{M}}_l$

$t \leftarrow t + 1$

end while

$\hat{\boldsymbol{v}}_l \leftarrow \boldsymbol{v}_l^{(t)}$, $\hat{\gamma}_l^2 \leftarrow (\gamma_l^{(t)})^2$, $\hat{\boldsymbol{\beta}}_l \leftarrow \boldsymbol{\beta}_l^{(t)}$, $\hat{\boldsymbol{\xi}}_l \leftarrow \boldsymbol{\xi}_l^{(t)}$

$\boldsymbol{X}_l^{zero} \leftarrow \boldsymbol{X}_l$ with elements at missing indices replaced with zero

$\boldsymbol{X}_l^{imp} \leftarrow \boldsymbol{X}_l$ with elements at missing indices replaced with conditional means

$\hat{\boldsymbol{u}}_l^{zero} = \boldsymbol{X}_l^{zero}\hat{\boldsymbol{v}}_l$

$\hat{\boldsymbol{u}}_l = \boldsymbol{X}_l^{imp}\hat{\boldsymbol{v}}_l$

end for

Output $\{\hat{\boldsymbol{v}}_1, ..., \hat{\boldsymbol{v}}_q\}$, $\{\hat{\boldsymbol{u}}_1, ..., \hat{\boldsymbol{u}}_q\}$, $\{\hat{\boldsymbol{\beta}}_1, ..., \hat{\boldsymbol{\beta}}_q\}$, $\{\hat{\gamma}_1^2, ..., \hat{\gamma}_q^2\}$, $\{\hat{\boldsymbol{\xi}}_1, ..., \hat{\boldsymbol{\xi}}_q\}$

A.2 The ProPrPCA-Spline model and algorithm

A.2.1 The model

For each PC, the ProPrPCA-Spline algorithm assumes the following model

$$\boldsymbol{X} = \boldsymbol{Z}\boldsymbol{\beta}\boldsymbol{v}^{\mathsf{T}} + \boldsymbol{E}$$

Here $\boldsymbol{Z}$ contains both geographic covariates and the thin-plate spline basis functions. The collection of model parameters Θ now includes $\{\boldsymbol{v}, \boldsymbol{\beta}, \gamma^2\}$. Using the same vectorization established in previous section, the model assumes $\boldsymbol{W}_i = \boldsymbol{X}_{i\cdot} = (\boldsymbol{Z}\boldsymbol{\beta})_i \boldsymbol{v} + \boldsymbol{\epsilon}_i$, for $i = 1, ..., n$. We can then rewrite this model in the vectorized form as

$$\boldsymbol{W} \mid \Theta \sim \mathcal{N}\left(\boldsymbol{V}\boldsymbol{Z}\boldsymbol{\beta}, \gamma^2 \boldsymbol{I}_N\right),$$

where $\boldsymbol{V} = \boldsymbol{I}_n \otimes \boldsymbol{v}$. When there are missing data, the distribution of interest becomes

$$\boldsymbol{W}_o \mid \Theta \sim \mathcal{N}\left(\boldsymbol{G}_o\boldsymbol{V}\boldsymbol{Z}\boldsymbol{\beta}, \gamma^2 \boldsymbol{I}_{N_o}\right).$$

A.2.2 Estimation of model parameters when monitoring data is complete

To solve for the parameters, we maximize the log-likelihood (up to a constant) directly:

$$\ell(\Theta \mid \boldsymbol{W}) = -\frac{1}{N}\log\gamma^2 - \frac{1}{2\gamma^2}\left(\boldsymbol{W} - \boldsymbol{V}\boldsymbol{Z}\boldsymbol{\beta}\right)^{\mathsf{T}}\left(\boldsymbol{W} - \boldsymbol{V}\boldsymbol{Z}\boldsymbol{\beta}\right)$$

To solve for $\boldsymbol{v}$, we effectively maximize the following function:

$$-\frac{1}{2\gamma^2}\left(\boldsymbol{W} - \boldsymbol{V}\boldsymbol{Z}\boldsymbol{\beta}\right)^{\mathsf{T}}\left(\boldsymbol{W} - \boldsymbol{V}\boldsymbol{Z}\boldsymbol{\beta}\right)$$

Denote $\boldsymbol{K} = \boldsymbol{Z}\boldsymbol{\beta} \in \mathbb{R}^{n\times 1}$, we can rewrite this function similarly to the function involved $\boldsymbol{v}$ with complete data for ProPrPCA-Krige. Thus, the solution for $\boldsymbol{v}$ can be written in

closed-form as

$$\hat{\boldsymbol{v}} = \frac{\breve{\boldsymbol{v}}}{\|\breve{\boldsymbol{v}}\|_2}, \qquad \text{where } \breve{\boldsymbol{v}} = \frac{\boldsymbol{X}^\mathsf{T}\boldsymbol{K}}{\|\boldsymbol{K}\|_2^2} = \frac{\boldsymbol{X}^\mathsf{T}\boldsymbol{Z}\boldsymbol{\beta}}{\|\boldsymbol{Z}\boldsymbol{\beta}\|_2^2}.$$

The closed-form solution for $\boldsymbol{\beta}$ is straightforwardly a result of ordinary least squares, $\hat{\boldsymbol{\beta}} = \left[(\boldsymbol{V}\boldsymbol{Z})^\mathsf{T}(\boldsymbol{V}\boldsymbol{Z})\right]^{-1}(\boldsymbol{V}\boldsymbol{Z})^\mathsf{T}\boldsymbol{W}$. This can be further simplified thanks to the constraint on $\boldsymbol{v}$ and noticing that $\boldsymbol{V}^\mathsf{T}\boldsymbol{V} = \boldsymbol{I}_n$. Thus we have, $\hat{\boldsymbol{\beta}} = \left(\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z}\right)^{-1}(\boldsymbol{Z}\otimes\boldsymbol{v})^\mathsf{T}\boldsymbol{W}$. Finally, the solution for γ^2 is simply $\hat{\gamma}^2 = N^{-1}\|\boldsymbol{W} - \boldsymbol{V}\boldsymbol{Z}\boldsymbol{\beta}\|_2^2$. Thus parameter estimation of ProPrPCA-Spline with complete monitoring data can be summarized as:

Algorithm: ProPrPCA-Spline with complete monitoring data

Input $\boldsymbol{X}$, $\boldsymbol{Z}$, q, and t_{max}
  for l in $\{1, ..., q\}$ **do**
    $\boldsymbol{X}_l \leftarrow \boldsymbol{X}_{l-1} - \hat{\boldsymbol{u}}_{l-1}\hat{\boldsymbol{v}}_{l-1}^\mathsf{T}$ where $\boldsymbol{X}_0 = \boldsymbol{X}$, $\hat{\boldsymbol{u}}_0 = \boldsymbol{0}$, and $\hat{\boldsymbol{v}}_0 = \boldsymbol{0}$
    Initialize $\boldsymbol{v}_l^{(0)}$, $(\gamma_l^{(0)})^2$, $\boldsymbol{\beta}_l^{(0)}$, and $t = 1$
      while not converged **or** $t < t_{max}$ **do**
        $\boldsymbol{v}_l^{(t+1)} \leftarrow \tilde{\boldsymbol{v}}_l / \|\tilde{\boldsymbol{v}}_l\|_2$ where $\tilde{\boldsymbol{v}}_l \leftarrow \boldsymbol{X}_l^\mathsf{T}\boldsymbol{Z}\boldsymbol{\beta}_l^{(t)} / \|\boldsymbol{Z}\boldsymbol{\beta}_l^{(t)}\|_2^2$
        $\boldsymbol{\beta}_l^{(t+1)} \leftarrow \left(\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z}\right)^{-1}\left(\boldsymbol{Z}\otimes\boldsymbol{v}_l^{(t+1)}\right)^\mathsf{T}\mathrm{vec}(\boldsymbol{X}_l)$
        $(\gamma_l^{(t+1)})^2 \leftarrow (np)^{-1}\|\mathrm{vec}(\boldsymbol{X}_l) - (\boldsymbol{I}_n\otimes\boldsymbol{v}_l^{(t+1)})\boldsymbol{Z}\boldsymbol{\beta}_l^{(t+1)}\|_2^2$
        $t \leftarrow t + 1$
      end while
    $\hat{\boldsymbol{v}}_l \leftarrow \boldsymbol{v}_l^{(t)}$, $\hat{\gamma}_l^2 \leftarrow (\gamma_l^{(t)})^2$, $\hat{\boldsymbol{\beta}}_l \leftarrow \boldsymbol{\beta}_l^{(t)}$
    $\hat{\boldsymbol{u}}_l = \boldsymbol{X}_l\hat{\boldsymbol{v}}_l$
  end for
Output $\{\hat{\boldsymbol{v}}_1, ..., \hat{\boldsymbol{v}}_q\}$, $\{\hat{\boldsymbol{u}}_1, ..., \hat{\boldsymbol{u}}_q\}$, $\{\hat{\boldsymbol{\beta}}_1, ..., \hat{\boldsymbol{\beta}}_q\}$, $\{\hat{\gamma}_1^2, ..., \hat{\gamma}_q^2\}$

A.2.3 Parameter estimation and model-based imputation with missing monitoring data

The observed log-likelihood with missing monitoring data is

$$\ell(\Theta \mid \boldsymbol{W}_o) = -\frac{1}{N_o}\log\gamma^2 - \frac{1}{2\gamma^2}\left(\boldsymbol{W}_o - \boldsymbol{G}_o\boldsymbol{V}\boldsymbol{Z}\boldsymbol{\beta}\right)^{\mathsf{T}}\left(\boldsymbol{W}_o - \boldsymbol{G}_o\boldsymbol{V}\boldsymbol{Z}\boldsymbol{\beta}\right).$$

The solutions for $\boldsymbol{\beta}$ and γ^2 are trivial and fairly similar to those with complete data. To solve for $\boldsymbol{v}$, we maximize $\{-\frac{1}{2\gamma^2}(\boldsymbol{W}_o - \boldsymbol{G}_o\boldsymbol{V}\boldsymbol{Z}\boldsymbol{\beta})^{\mathsf{T}}(\boldsymbol{W}_o - \boldsymbol{G}_o\boldsymbol{V}\boldsymbol{Z}\boldsymbol{\beta})\}$. We can rewrite the function of $\boldsymbol{v}$ as

$$\begin{aligned}
&-\frac{1}{2\gamma^2}\left(\boldsymbol{W}_o - \boldsymbol{G}_o\boldsymbol{V}\boldsymbol{Z}\boldsymbol{\beta}\right)^{\mathsf{T}}\left(\boldsymbol{W}_o - \boldsymbol{G}_o\boldsymbol{V}\boldsymbol{Z}\boldsymbol{\beta}\right)\\
=&-\frac{1}{2\gamma^2}\left(\boldsymbol{W}_o - \boldsymbol{G}_o\boldsymbol{V}\boldsymbol{K}\right)^{\mathsf{T}}\left(\boldsymbol{W}_o - \boldsymbol{G}_o\boldsymbol{V}\boldsymbol{K}\right)\\
=&\sum_{i=1}^{n}\sum_{j\in\Omega_{i\cdot}}\left(X_{ij} - K_i v_j\right)^2 = \sum_{i=1}^{n}\left(X_{ij} - K_i v_j\right)^2 \mathbf{1}_{[j\in\Omega_{i\cdot}]},
\end{aligned}$$

Taking derivative with respect to each v_j and setting it equal to zero, we can find the closed-form unscaled solution

$$\breve{v} = \frac{\sum_{i=1}^{n} X_{ij}K_i\mathbf{1}_{[j\in\Omega_{i\cdot}]}}{\sum_{i=1}^{n} K_i^2\mathbf{1}_{[j\in\Omega_{i\cdot}]}}.$$

The final solution for $\boldsymbol{v}$ is then $\hat{\boldsymbol{v}} = \frac{\breve{\boldsymbol{v}}}{\|\breve{\boldsymbol{v}}\|_2}$, where $\breve{\boldsymbol{v}} = (\breve{v}_1, ..., \breve{v}_p)$.

When some elements of the exposure data are missing, parameter estimation for each PC is based only on the observed elements $\boldsymbol{W}_o$. Estimate for PC score can then be made by projecting the model-based imputed exposure data onto the direction of $\boldsymbol{v}$. The missing elements, $\boldsymbol{W}_m$, can then be imputed by its estimate $\boldsymbol{G}_m\hat{\boldsymbol{V}}\boldsymbol{Z}\hat{\boldsymbol{\beta}}$. Thus the parameter estimation of ProPrPCA-Spline with missing monitoring data can be summarized as:

Algorithm: ProPrPCA-Spline with missing monitoring data

Input $\boldsymbol{X}$, $\boldsymbol{G}_o$ $\boldsymbol{Z}$, q, and t_{max}

for l in $\{1, ..., q\}$ **do**

$\boldsymbol{X}_l \leftarrow \boldsymbol{X}_{l-1}^{zero} - \hat{\boldsymbol{u}}_{l-1}^{zero}\hat{\boldsymbol{v}}_{l-1}^{\mathsf{T}}$ where $\boldsymbol{X}_0^{zero} = \boldsymbol{X}$ imputed with zeros, $\hat{\boldsymbol{u}}_0^{zero} = \boldsymbol{0}$, and $\hat{\boldsymbol{v}}_0 = \boldsymbol{0}$

$\boldsymbol{W}_o \leftarrow \boldsymbol{G}_o \text{vec}(\boldsymbol{X}_l)$

Initialize $\boldsymbol{v}_l^{(0)}$, $(\gamma_l^{(0)})^2$, $\boldsymbol{\beta}_l^{(0)}$, and $t = 1$

while not converged **or** $t < t_{max}$ **do**

$\boldsymbol{v}_l^{(t+1)} \leftarrow \tilde{\boldsymbol{v}}_l / \|\tilde{\boldsymbol{v}}_l\|_2$ where the $j-$element of $\tilde{\boldsymbol{v}}_l$ for $j = 1, ..., p$ is calculated as:

$$\frac{\sum_{i=1}^{n} (\boldsymbol{X}_l)_{ij} (\boldsymbol{K}_l)_i \boldsymbol{1}_{[j \in \Omega_{i.}]}}{\sum_{i=1}^{n} (\boldsymbol{K}_l)_i^2 \boldsymbol{1}_{[j \in \Omega_{i.}]}}, \quad \text{and} \quad \boldsymbol{K}_l = \boldsymbol{Z}\boldsymbol{\beta}_l^{(t)}$$

$$\boldsymbol{\beta}_l^{(t+1)} \leftarrow \left[\left(\boldsymbol{G}_o \boldsymbol{V}_l^{(t+1)} \boldsymbol{Z} \right)^{\mathsf{T}} \left(\boldsymbol{G}_o \boldsymbol{V}_l^{(t+1)} \boldsymbol{Z} \right) \right]^{-1} \left(\boldsymbol{G}_o \boldsymbol{V}_l^{(t+1)} \boldsymbol{Z} \right)^{\mathsf{T}} \boldsymbol{W}_o$$

where $\boldsymbol{V}_l^{(t+1)} = \boldsymbol{I}_n \otimes \boldsymbol{v}_l^{(t+1)}$

$(\gamma_l^{(t+1)})^2 \leftarrow N_o{}^{-1} \left\| \boldsymbol{W}_o - \boldsymbol{G}_o \boldsymbol{V}_l^{(t+1)} \boldsymbol{Z} \boldsymbol{\beta}_l^{(t+1)} \right\|_2^2$

$t \leftarrow t + 1$

end while

$\hat{\boldsymbol{v}}_l \leftarrow \boldsymbol{v}_l^{(t)}$, $\hat{\gamma}_l^2 \leftarrow (\gamma_l^{(t)})^2$, $\hat{\boldsymbol{\beta}}_l \leftarrow \boldsymbol{\beta}_l^{(t)}$

$\boldsymbol{X}_l^{zero} \leftarrow \boldsymbol{X}_l$ with elements at missing indices replaced with zero

$\boldsymbol{X}_l^{imp} \leftarrow \boldsymbol{X}_l$ with elements at missing indices replaced with conditional means

$\hat{\boldsymbol{u}}_l^{zero} = \boldsymbol{X}_l^{zero}\hat{\boldsymbol{v}}_l$

$\hat{\boldsymbol{u}}_l = \boldsymbol{X}_l^{imp}\hat{\boldsymbol{v}}_l$

end for

Output $\{\hat{\boldsymbol{v}}_1, ..., \hat{\boldsymbol{v}}_q\}$, $\{\hat{\boldsymbol{u}}_1, ..., \hat{\boldsymbol{u}}_q\}$, $\{\hat{\boldsymbol{\beta}}_1, ..., \hat{\boldsymbol{\beta}}_q\}$, $\{\hat{\gamma}_1^2, ..., \hat{\gamma}_q^2\}$

A.3 High-dimensional simulations

A.3.1 Data generating mechanism

To further demonstrate the performance of ProPrPCA, we simulate multi-pollutant exposure surfaces with $p = 15$. We first generate three underlying PC scores on the 100×100 grid ($N = 10,000$), such that

$$\begin{aligned}
&\boldsymbol{u}_j \sim \mathcal{N}\left(\boldsymbol{R}_j \boldsymbol{b}_j, \boldsymbol{S}_j\right), \quad \text{where } j = 1, 2, 3, \\
&\boldsymbol{R}_1 = \begin{bmatrix} \boldsymbol{r}_{1o} & \boldsymbol{r}_{1u} \end{bmatrix}, \quad \text{where } \boldsymbol{r}_{1o}, \boldsymbol{r}_{1u} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{1}), \quad \boldsymbol{b}_1^{\mathsf{T}} = \begin{bmatrix} 5 & 1 \end{bmatrix}, \\
&\boldsymbol{R}_2 = \begin{bmatrix} \boldsymbol{r}_{2o} & \boldsymbol{r}_{2u} \end{bmatrix}, \quad \text{where } \boldsymbol{r}_{2o}, \boldsymbol{r}_{2u} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{1}), \quad \boldsymbol{b}_2^{\mathsf{T}} = \begin{bmatrix} 5 & 2 \end{bmatrix}, \\
&\boldsymbol{R}_3 = \begin{bmatrix} \boldsymbol{r}_{3u} \end{bmatrix}, \quad \text{where } \boldsymbol{r}_{3u} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{1}), \quad \boldsymbol{b}_3 = 1.
\end{aligned}$$

In this setting, $\boldsymbol{r}_{jo}$'s are GIS covariates observed for the model, while $\boldsymbol{r}_{ju}$'s are unobserved covariates, and used primarily to generate the scores themselves. That is, only $\boldsymbol{R} = \begin{bmatrix} \boldsymbol{r}_{1o} & \boldsymbol{r}_{2o} \end{bmatrix}$ is used in the spatial prediction model. Here $\boldsymbol{S}_1$ has exponential structure with no nugget effect, partial sill of 5, and range of 50. Meanwhile, $\boldsymbol{S}_2 = 7.5\boldsymbol{I}_N$ and $\boldsymbol{S}_3 = 2\boldsymbol{I}_N$. This setup is created so that $\boldsymbol{u}_1$ is the most spatially predictable, $\boldsymbol{u}_2$ is moderately predictable in space, and $\boldsymbol{u}_3$ is not spatially predictable. Here spatial predictability refers to how well the quantity can be predicted at new locations using relevant and available covariates.

We then create two scenarios in which we scale the variance of $\boldsymbol{u}_j$'s differently,

$$\begin{aligned}
&\text{Scenario 1: } Var(\boldsymbol{u}_1) = 10, Var(\boldsymbol{u}_2) = 7.5, Var(\boldsymbol{u}_3) = 5, \\
&\text{Scenario 2: } Var(\boldsymbol{u}_1) = 7.5, Var(\boldsymbol{u}_2) = 5, Var(\boldsymbol{u}_3) = 10.
\end{aligned}$$

In both scenarios, the multi-pollutant exposure surface is generated as

$$
\begin{aligned}
&\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V} + \boldsymbol{E}, \ \text{ where } \ E_{ij} \sim \mathcal{N}(0,1) \\
&\boldsymbol{V} = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \boldsymbol{v}_3 \end{bmatrix}, \ \text{ where } \ \boldsymbol{v}_j = \frac{\breve{\boldsymbol{v}}_j}{\left\|\breve{\boldsymbol{v}}_j\right\|_2}, \ \text{ for } \ j = 1,2,3 \\
&\breve{\boldsymbol{v}}_1^{\mathsf{T}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
&\breve{\boldsymbol{v}}_2^{\mathsf{T}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
&\breve{\boldsymbol{v}}_3^{\mathsf{T}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}
\end{aligned}
$$

The use of such sparse loadings is to clearly identify the behavior of each dimension reduction method. Because of the variance contribution setup, in scenario A we expect all three methods to pick $\boldsymbol{u}_1$ as PC1, $\boldsymbol{u}_2$ as PC2, and $\boldsymbol{u}_3$ as PC3. In scenario B, however, we expect TradPCA to pick $\boldsymbol{u}_3$ as PC1, $\boldsymbol{u}_1$ as PC2, and $\boldsymbol{u}_2$ as PC3, as $\boldsymbol{u}_3$ has the largest variance contribution. Meanwhile, PredPCA and ProPrPCA-Spline will still pick $\boldsymbol{u}_1$ as PC1, $\boldsymbol{u}_2$ as PC2.

For these high-dimensional simulations, we consider three MCAR scenarios (30%, 35%, and 40%), and one MAR scenario. In the MAR scenario, we identify training locations with $\boldsymbol{r}_{1o}$ value larger than its sample 60th percentile, and among $\boldsymbol{x}_1$ through $\boldsymbol{x}_5$, 75% of these training locations become missing data. For the rest of the pollutants, from $\boldsymbol{x}_6$ to $\boldsymbol{x}_{15}$, each has 25% of its locations missing completely at random. This setup guarantees a mild spatial pattern in the missing data, as $\boldsymbol{x}_1$ to $\boldsymbol{x}_5$ are generated entirely by $\boldsymbol{u}_1$, which is the most predictable score based on $\boldsymbol{r}_{1o}$.

Appendix B

APPENDIX FOR CHAPTER 3

B.1 Proof of lemma

Lemma. *If* $(\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z})^{-1}$ *exists, then the approximation* $\hat{\boldsymbol{W}} = \boldsymbol{Z}\hat{\boldsymbol{M}}$ *of* $\boldsymbol{X}$*, where* $\hat{\boldsymbol{M}}$ *is the optimal solution for*

$$\min_{\boldsymbol{M}} \left\{ \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{M}\|_F^2 + \lambda\|\boldsymbol{Z}\boldsymbol{M}\|_* \right\},$$

has a closed-form expression,

$$\hat{\boldsymbol{W}} = \tilde{\boldsymbol{U}} S_\lambda(\boldsymbol{D}) \tilde{\boldsymbol{V}}^\mathsf{T},$$

where $\tilde{\boldsymbol{U}}\boldsymbol{D}\tilde{\boldsymbol{V}}^\mathsf{T}$ *is SVD of* $\tilde{\boldsymbol{X}} = \boldsymbol{Z}(\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z})^{-1}\boldsymbol{Z}^\mathsf{T}\boldsymbol{X}$*.*

Proof. First we define the following quantities,

$$\tilde{\boldsymbol{X}} = \boldsymbol{Z}(\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z})^{-1}\boldsymbol{Z}^\mathsf{T}\boldsymbol{X},$$
$$\boldsymbol{X}^\perp = \boldsymbol{X} - \tilde{\boldsymbol{X}} = \boldsymbol{X} - \boldsymbol{Z}(\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z})^{-1}\boldsymbol{Z}^\mathsf{T}\boldsymbol{X}.$$

The convex optimization can then be written as:

$$\min_{\boldsymbol{M}} \left\{ \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{M}\|_F^2 + \lambda\|\boldsymbol{Z}\boldsymbol{M}\|_* \right\} = \min_{\boldsymbol{M}} \left\{ \frac{1}{2}\|\tilde{\boldsymbol{X}} + \boldsymbol{X}^\perp - \boldsymbol{Z}\boldsymbol{M}\|_F^2 + \lambda\|\boldsymbol{Z}\boldsymbol{M}\|_* \right\}$$

The Frobenius-norm term can be rewritten as follows

$$
\begin{aligned}
&\left\|\tilde{\boldsymbol{X}}+\boldsymbol{X}^{\perp}-\boldsymbol{Z}\boldsymbol{M}\right\|_F^2 \\
&=\mathrm{Tr}\left[\left(\tilde{\boldsymbol{X}}+\boldsymbol{X}^{\perp}-\boldsymbol{Z}\boldsymbol{M}\right)\left(\tilde{\boldsymbol{X}}+\boldsymbol{X}^{\perp}-\boldsymbol{Z}\boldsymbol{M}\right)^{\mathsf{T}}\right] \\
&=\mathrm{Tr}\Bigg[\left(\tilde{\boldsymbol{X}}-\boldsymbol{Z}\boldsymbol{M}\right)\left(\tilde{\boldsymbol{X}}-\boldsymbol{Z}\boldsymbol{M}\right)^{\mathsf{T}}+\left(\boldsymbol{X}^{\perp}\right)\left(\boldsymbol{X}^{\perp}\right)^{\mathsf{T}}+\left(\tilde{\boldsymbol{X}}-\boldsymbol{Z}\boldsymbol{M}\right)\left(\boldsymbol{X}^{\perp}\right)^{\mathsf{T}} \\
&\quad+\left(\boldsymbol{X}^{\perp}\right)\left(\tilde{\boldsymbol{X}}-\boldsymbol{Z}\boldsymbol{M}\right)^{\mathsf{T}}\Bigg] \\
&=\left\|\tilde{\boldsymbol{X}}-\boldsymbol{Z}\boldsymbol{M}\right\|_F^2+\left\|\boldsymbol{X}^{\perp}\right\|_F^2+2\mathrm{Tr}\left[\left(\tilde{\boldsymbol{X}}-\boldsymbol{Z}\boldsymbol{M}\right)\left(\boldsymbol{X}^{\perp}\right)^{\mathsf{T}}\right]
\end{aligned}
$$

The remaining trace term can be further simplified:

$$
\begin{aligned}
&\mathrm{Tr}\left[\left(\tilde{\boldsymbol{X}}-\boldsymbol{Z}\boldsymbol{M}\right)\left(\boldsymbol{X}^{\perp}\right)^{\mathsf{T}}\right] \\
&=\mathrm{Tr}\left[\left(\tilde{\boldsymbol{X}}-\boldsymbol{Z}\boldsymbol{M}\right)\left(\boldsymbol{X}-\tilde{\boldsymbol{X}}\right)^{\mathsf{T}}\right] \\
&=\mathrm{Tr}\left[\tilde{\boldsymbol{X}}\boldsymbol{X}^{\mathsf{T}}-\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^{\mathsf{T}}-\boldsymbol{Z}\boldsymbol{M}\boldsymbol{X}^{\mathsf{T}}+\boldsymbol{Z}\boldsymbol{M}\tilde{\boldsymbol{X}}^{\mathsf{T}}\right] \\
&=\mathrm{Tr}\left[\boldsymbol{Z}(\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{Z})^{-1}\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{X}\boldsymbol{X}^{\mathsf{T}}\right]-\mathrm{Tr}\left[\boldsymbol{Z}(\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{Z})^{-1}\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{X}\boldsymbol{X}^{\mathsf{T}}\boldsymbol{Z}(\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{Z})^{-1}\boldsymbol{Z}^{\mathsf{T}}\right] \\
&\quad-\mathrm{Tr}\left(\boldsymbol{Z}\boldsymbol{M}\boldsymbol{X}^{\mathsf{T}}\right)+\mathrm{Tr}\left[\boldsymbol{Z}\boldsymbol{M}\boldsymbol{X}^{\mathsf{T}}\boldsymbol{Z}(\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{Z})^{-1}\boldsymbol{Z}^{\mathsf{T}}\right] \\
&=\mathrm{Tr}\left[\boldsymbol{Z}(\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{Z})^{-1}\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{X}\boldsymbol{X}^{\mathsf{T}}\right]-\mathrm{Tr}\left[\boldsymbol{Z}(\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{Z})^{-1}\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{Z}(\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{Z})^{-1}\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{X}\boldsymbol{X}^{\mathsf{T}}\right] \\
&\quad-\mathrm{Tr}\left(\boldsymbol{Z}\boldsymbol{M}\boldsymbol{X}^{\mathsf{T}}\right)+\mathrm{Tr}\left[\boldsymbol{Z}(\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{Z})^{-1}\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{Z}\boldsymbol{M}\boldsymbol{X}^{\mathsf{T}}\right] \\
&=\mathrm{Tr}\left[\boldsymbol{Z}(\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{Z})^{-1}\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{X}\boldsymbol{X}^{\mathsf{T}}\right]-\mathrm{Tr}\left[\boldsymbol{Z}(\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{Z})^{-1}\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{X}\boldsymbol{X}^{\mathsf{T}}\right]-\mathrm{Tr}\left(\boldsymbol{Z}\boldsymbol{M}\boldsymbol{X}^{\mathsf{T}}\right)+\mathrm{Tr}\left(\boldsymbol{Z}\boldsymbol{M}\boldsymbol{X}^{\mathsf{T}}\right) \\
&=0
\end{aligned}
$$

As a result, the convex optimization problem now becomes

$$\begin{aligned}
&\min_{\boldsymbol{M}} \left\{ \frac{1}{2} \left\| \boldsymbol{X} - \boldsymbol{Z}\boldsymbol{M} \right\|_F^2 + \lambda \left\| \boldsymbol{Z}\boldsymbol{M} \right\|_* \right\} \\
= &\min_{\boldsymbol{M}} \left\{ \frac{1}{2} \left\| \tilde{\boldsymbol{X}} + \boldsymbol{X}^{\perp} - \boldsymbol{Z}\boldsymbol{M} \right\|_F^2 + \lambda \left\| \boldsymbol{Z}\boldsymbol{M} \right\|_* \right\} \\
= &\min_{\boldsymbol{M}} \left\{ \frac{1}{2} \left\| \tilde{\boldsymbol{X}} - \boldsymbol{Z}\boldsymbol{M} \right\|_F^2 + \frac{1}{2} \left\| \boldsymbol{X}^{\perp} \right\|_F^2 + \lambda \left\| \boldsymbol{Z}\boldsymbol{M} \right\|_* \right\} \\
= &\min_{\boldsymbol{M}} \left\{ \frac{1}{2} \left\| \tilde{\boldsymbol{X}} - \boldsymbol{Z}\boldsymbol{M} \right\|_F^2 + \lambda \left\| \boldsymbol{Z}\boldsymbol{M} \right\|_* \right\} \\
= &\min_{\boldsymbol{W}} \left\{ \frac{1}{2} \left\| \tilde{\boldsymbol{X}} - \boldsymbol{W} \right\|_F^2 + \lambda \left\| \boldsymbol{W} \right\|_* \right\} \quad \text{s.t. } \boldsymbol{W} \in \text{col}(\boldsymbol{Z})
\end{aligned}$$

The solution for

$$\min_{\boldsymbol{W}} \left\{ \frac{1}{2} \left\| \tilde{\boldsymbol{X}} - \boldsymbol{W} \right\|_F^2 + \lambda \left\| \boldsymbol{W} \right\|_* \right\}$$

is simply $\hat{\boldsymbol{W}} = \tilde{\boldsymbol{U}} S_{\lambda}(\tilde{\boldsymbol{D}}) \tilde{\boldsymbol{V}}^{\mathsf{T}}$ where $\tilde{\boldsymbol{U}} \tilde{\boldsymbol{D}} \tilde{\boldsymbol{V}}^{\mathsf{T}}$ is the SVD of $\tilde{\boldsymbol{X}}$. Recall that

$$\tilde{\boldsymbol{X}} = \boldsymbol{Z}(\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{Z})^{-1}\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{X} = \boldsymbol{Z}(\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{Z})^{-1}\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{U}_X\boldsymbol{D}_X\boldsymbol{V}_X^{\mathsf{T}} = \left(\boldsymbol{Z}(\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{Z})^{-1}\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{U}_X\right)\boldsymbol{D}_X\boldsymbol{V}_X^{\mathsf{T}}$$

where $\boldsymbol{U}_X\boldsymbol{D}_X\boldsymbol{V}_X^{\mathsf{T}}$ is the SVD of the original data $\boldsymbol{X}$. Thus the solution becomes

$$\hat{\boldsymbol{W}} = \left(\boldsymbol{Z}(\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{Z})^{-1}\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{U}_X\right) S_{\lambda}(\boldsymbol{D}_X)\boldsymbol{V}_X^{\mathsf{T}}$$

and this solution indeed lies in the column space of $\boldsymbol{Z}$. Thus we conclude that the original problem with respect to $\boldsymbol{M}$ is equivalent to the following convex optimization problem

$$\min_{\boldsymbol{W}} \left\{ \frac{1}{2} \left\| \tilde{\boldsymbol{X}} - \boldsymbol{W} \right\|_F^2 + \lambda \left\| \boldsymbol{W} \right\|_* \right\}.$$

That is, the approximation $\hat{\boldsymbol{W}} = \boldsymbol{Z}\hat{\boldsymbol{M}}$ of $\boldsymbol{X}$, where $\hat{\boldsymbol{M}}$ is the optimal solution for the original

problem of interest, has a closed-form expression,

$$\hat{\boldsymbol{W}} = \tilde{\boldsymbol{U}} S_\lambda(\boldsymbol{D}) \tilde{\boldsymbol{V}}^\mathsf{T},$$

where $\tilde{\boldsymbol{U}}\boldsymbol{D}\tilde{\boldsymbol{V}}^\mathsf{T}$ is SVD of $\tilde{\boldsymbol{X}} = \boldsymbol{Z}(\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z})^{-1}\boldsymbol{Z}^\mathsf{T}\boldsymbol{X}$.

B.2 The low-rank matrix completion algorithm

B.2.1 The proximal algorithm

We first revisit the idea of proximal algorithm. The proximal algorithm (Rockafellar, 1976) is an approximation-regularization method in convex optimization. Consider an optimization problem of the form

$$\min_{\boldsymbol{\theta}}\{f(\boldsymbol{\theta}) + h(\boldsymbol{\theta})\} \tag{B.1}$$

where $\boldsymbol{\theta} \in \mathbb{R}^n$ is some parameter of interest, $f : \mathbb{R}^n \mapsto \mathbb{R}$ is a smooth convex loss function, and $h : \mathbb{R}^n \mapsto \mathbb{R}$ is a convex penalization. If $h(.)$ is differentiable, one can solve this problem easily using the majorization-minimization framework (Hunter and Lange, 2004). However, in many situations, $h(.)$ is either not differentiable, or its gradient is hard to compute. This is where the proximal algorithm comes in handy.

Define the *proximal mapping* of a function $h(\boldsymbol{z})$ as follows

$$\text{prox}_c(\boldsymbol{x}) = \arg\min_{\boldsymbol{z}} \left\{ \frac{1}{2c}\left\|\boldsymbol{x} - \boldsymbol{z}\right\|_2^2 + h(\boldsymbol{z}) \right\},$$

then the proximal algorithm for (B.1) involves two steps (Rockafellar, 1976):

- Update: $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta}^{(t)} - c_t \nabla f(\boldsymbol{\theta}^{(t)})$ where $\nabla f(\boldsymbol{\theta}^{(t)})$ is the gradient of $f(.)$ at the current estimate $\boldsymbol{\theta}^{(t)}$

- Solve the proximal problem:

$$\boldsymbol{\theta}^{(t+1)} = \text{prox}_{c_t}(\tilde{\boldsymbol{\theta}}) = \arg\min_{\boldsymbol{\theta}} \left\{ \frac{1}{2c_t}\left\|\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}\right\|_2^2 + h(\boldsymbol{\theta}) \right\}$$

For some particular penalty functions, the proximal problem is usually easier to solve than the original problem, as closed-form solutions exist. In addition, when $\nabla f(.)$ is Lipschitz continuous with constant L, the same step size $c = 1/L$ can be picked for every iteration.

B.2.2 The low-rank matrix completion (LRMC) algorithm by proximal gradient descents

In this section, we revisit the original LRMC problem,

$$\min_{\boldsymbol{W}} \ \frac{1}{2}\left\{\left\|P_\Omega(\boldsymbol{X}) - P_\Omega(\boldsymbol{W})\right\|_F^2 + \lambda\left\|\boldsymbol{W}\right\|_*\right\}$$

where $P_\Omega(.)$ is the projection of a matrix onto its observed entries, i.e. $[P_\Omega(\boldsymbol{X})]_{ij} = X_{ij}$ if $i, j \in \Omega$ and $[P_\Omega(\boldsymbol{X})]_{ij} = 0$ if $i, j \notin \Omega$. It's easy to see that $f(\boldsymbol{W}) = \frac{1}{2}\left\|P_\Omega(\boldsymbol{X}) - P_\Omega(\boldsymbol{W})\right\|_F^2$ is convex and differentiable. In fact,

$$f(\boldsymbol{W}) = \frac{1}{2}\left\|P_\Omega(\boldsymbol{X}) - P_\Omega(\boldsymbol{W})\right\|_F^2 = \begin{cases} \frac{1}{2}(X_{ij} - W_{ij})^2 & \text{if } i, j \in \Omega; \\ 0 & \text{if } i, j \notin \Omega. \end{cases}$$

and thus the gradient of $f(.)$ with respect to $\boldsymbol{W}$ can be easily calculated as

$$\triangledown f(\boldsymbol{W}) = \begin{cases} -(X_{ij} - W_{ij}) & \text{if } i, j \in \Omega; \\ 0 & \text{if } i, j \notin \Omega. \end{cases} = -(P_\Omega(\boldsymbol{X}) - P_\Omega(\boldsymbol{W})) = P_\Omega(\boldsymbol{W}) - P_\Omega(\boldsymbol{X})$$

As a result, the update step of the proximal algorithm at the t-th iteration is:

$$\tilde{\boldsymbol{W}} = \boldsymbol{W}^{(t)} - c_t\left(P_\Omega(\boldsymbol{W}^{(t)}) - P_\Omega(\boldsymbol{X})\right)$$

We then solve the proximal problem,

$$\begin{aligned} &\min_{\boldsymbol{W}}\left\{\frac{1}{2c_t}\left\|\tilde{\boldsymbol{W}} - \boldsymbol{W}\right\|_F^2 + \lambda\|\boldsymbol{W}\|_*\right\} \\ =&\min_{\boldsymbol{W}}\left\{\frac{1}{2}\left\|\tilde{\boldsymbol{W}} - \boldsymbol{W}\right\|_F^2 + \lambda c_t\|\boldsymbol{W}\|_*\right\} \end{aligned}$$

which is exactly the "fully observed" problem that has a closed-form solution by using the soft-thresholding operator, i.e.

$$\hat{\boldsymbol{W}} = \tilde{\boldsymbol{U}} S_{\lambda c_t}(\tilde{\boldsymbol{D}})\tilde{\boldsymbol{V}}^\mathsf{T}$$

where $\tilde{\boldsymbol{U}}\tilde{\boldsymbol{D}}\tilde{\boldsymbol{V}}^{\mathsf{T}}$ is the SVD of $\tilde{\boldsymbol{W}}$

For any $\boldsymbol{W}_1, \boldsymbol{W}_2 \in \mathbb{R}^{n\times p}$, it's easily to see that:

$$\begin{aligned}
&\left\|\nabla f(\boldsymbol{W}_2) - \nabla f(\boldsymbol{W}_1)\right\|_F \\
&= \left\|P_\Omega(\boldsymbol{W}_2) - P_\Omega(\boldsymbol{X}) - (P_\Omega(\boldsymbol{W}_1) - P_\Omega(\boldsymbol{X}))\right\|_F \\
&= \left\|P_\Omega(\boldsymbol{W}_2) - P_\Omega(\boldsymbol{W}_1)\right\|_F \\
&= \sum_{i,j}(W_{2,ij} - W_{1,ij})^2 \mathbb{1}_{[i,j\in\Omega]} \\
&\leq \sum_{i,j}(W_{2,ij} - W_{1,ij})^2 = \left\|\boldsymbol{W}_2 - \boldsymbol{W}_1\right\|_F
\end{aligned}$$

Thus $\nabla f(.)$ is Lipschitz with $L = 1$. The update step can be simplified even further:

$$\begin{aligned}
\tilde{\boldsymbol{W}} &= \boldsymbol{W}^{(t)} - \left(P_\Omega(\boldsymbol{W}^{(t)}) - P_\Omega(\boldsymbol{X})\right) \\
&= P_\Omega(\boldsymbol{X}) + \left(\boldsymbol{W}^{(t)} - P_\Omega(\boldsymbol{W}^{(t)})\right) \\
&= P_\Omega(\boldsymbol{X}) + P_\Omega^\perp(\boldsymbol{W}^{(t)}),
\end{aligned}$$

i.e. we replace the missing entries of the original matrix with the corresponding entries of the solution in the current step. The algorithm can be summarized in the following steps:

Algorithm 1: LRMC adapted from Mazumder et al. (2010)

Input $\boldsymbol{X}$, q, λ, and t_{max}
  Initialize $\boldsymbol{W}^{(0)} = \boldsymbol{0}$, $t = 1$
    while not converged or $t < t_{max}$ **do**
      $\tilde{\boldsymbol{W}}^{(t)} \leftarrow P_\Omega(\boldsymbol{X}) + P_\Omega^\perp(\boldsymbol{W}^{(t)})$, where $P_\Omega^\perp(\boldsymbol{W}^{(t)}) = \boldsymbol{W}^{(t)} - P_\Omega(\boldsymbol{W}^{(t)})$
      $\boldsymbol{W}^{(t+1)} \leftarrow \tilde{\boldsymbol{U}} S_\lambda(\tilde{\boldsymbol{D}})\tilde{\boldsymbol{V}}^{\mathsf{T}}$, where $\tilde{\boldsymbol{U}}\tilde{\boldsymbol{D}}\tilde{\boldsymbol{V}}^{\mathsf{T}}$ is the SVD of $\tilde{\boldsymbol{W}}^{(t)}$
      $t \leftarrow t + 1$
    end while
Output $\hat{\boldsymbol{W}} = \boldsymbol{W}^{(t)}$, $\hat{\boldsymbol{X}} = P_\Omega(\boldsymbol{X}) + P_\Omega^\perp(\hat{\boldsymbol{W}})$

B.3 The spatial matrix completion (SMC) algorithm

When some entries of $\boldsymbol{X}$ are missing, we propose the following problem:

$$\min_{\boldsymbol{M}} \left\{ \frac{1}{2} \left\| P_\Omega(\boldsymbol{X}) - P_\Omega(\boldsymbol{ZM}) \right\|_F^2 + \lambda \left\| \boldsymbol{ZM} \right\|_* \right\}$$

We can rewrite this as:

$$\begin{aligned} & \min_{\boldsymbol{M}} \left\{ \frac{1}{2} \left\| P_\Omega(\boldsymbol{X}) - P_\Omega(\boldsymbol{ZM}) \right\|_F^2 + \lambda \left\| \boldsymbol{ZM} \right\|_* \right\} \\ = & \min_{\boldsymbol{W}} \left\{ \frac{1}{2} \left\| P_\Omega(\boldsymbol{X}) - P_\Omega(\boldsymbol{W}) \right\|_F^2 + \lambda \left\| \boldsymbol{W} \right\|_* \right\} \text{ s.t. } \boldsymbol{W} = \boldsymbol{ZM} \\ = & \min_{\boldsymbol{W},\boldsymbol{M}} \left\{ \frac{1}{2} \left\| P_\Omega(\boldsymbol{X}) - P_\Omega(\boldsymbol{W}) \right\|_F^2 + \lambda \left\| \boldsymbol{W} \right\|_* + \mathbb{I}(\boldsymbol{W} - \boldsymbol{ZM}) \right\} \end{aligned}$$

where

$$\mathbb{I}(\boldsymbol{A}) = \begin{cases} 0 & \text{if } \boldsymbol{A} = \boldsymbol{0}; \\ \infty & \text{if } \boldsymbol{A} \neq \boldsymbol{0}. \end{cases}$$

It is easy to see that $f(\boldsymbol{W}, \boldsymbol{M}) = \frac{1}{2} \left\| P_\Omega(\boldsymbol{X}) - P_\Omega(\boldsymbol{W}) \right\|_F^2$ is convex and differentiable with its gradient (with respect to $\boldsymbol{W}$) being Lipschitz continuous ($L = 1$), and $h(\boldsymbol{W}, \boldsymbol{M}) = \lambda \left\| \boldsymbol{W} \right\|_* + \mathbb{I}(\boldsymbol{W} - \boldsymbol{ZM})$ is also convex.

Thus we can use proximal gradient descent to solve this problem. The update step is the same as in the LRMC algorithm, i.e.

$$\breve{\boldsymbol{W}}^{(t)} = P_\Omega(\boldsymbol{X}) + P_\Omega^\perp(\boldsymbol{W}^{(t)}).$$

In the second step, we solve the following proximal problem:

$$\begin{aligned}
&\min_{\boldsymbol{W},\boldsymbol{M}} \left\{ \frac{1}{2} \left\| \breve{\boldsymbol{W}}^{(t)} - \boldsymbol{W} \right\|_F^2 + \lambda \left\| \boldsymbol{W} \right\|_* + \mathbb{I}(\boldsymbol{W} - \boldsymbol{Z}\boldsymbol{M}) \right\} \\
&= \min_{\boldsymbol{W}} \left\{ \frac{1}{2} \left\| \breve{\boldsymbol{W}}^{(t)} - \boldsymbol{W} \right\|_F^2 + \lambda \left\| \boldsymbol{W} \right\|_* \right\} \text{ s.t. } \boldsymbol{W} = \boldsymbol{Z}\boldsymbol{M} \\
&= \min_{\boldsymbol{M}} \left\{ \frac{1}{2} \left\| \breve{\boldsymbol{W}}^{(t)} - \boldsymbol{Z}\boldsymbol{M} \right\|_F^2 + \lambda \left\| \boldsymbol{Z}\boldsymbol{M} \right\|_* \right\}
\end{aligned}$$

which is exactly the "fully observed" version that we propose earlier, with the complete data now being $\breve{\boldsymbol{W}}^{(t)}$. This proximal problem has a closed-form solution based on $\tilde{\boldsymbol{W}}^{(t)} = \boldsymbol{Z}(\boldsymbol{Z}^\top \boldsymbol{Z})^{-1}\boldsymbol{Z}^\top \breve{\boldsymbol{W}}^{(t)}$. Thus the SMC algorithm can be summarized in the following steps

Algorithm 2: Spatial matrix completion (SMC)

Input $\boldsymbol{X}$, $\boldsymbol{Z}$, q, λ, and t_{max}
  Initialize $\boldsymbol{W}^{(0)} = \boldsymbol{0}$, $t = 1$
    while not converged or $t < t_{max}$ **do**
      $\breve{\boldsymbol{W}}^{(t)} \leftarrow P_\Omega(\boldsymbol{X}) + P_\Omega^\perp(\boldsymbol{W}^{(t)})$
      $\tilde{\boldsymbol{W}}^{(t)} \leftarrow \boldsymbol{Z}(\boldsymbol{Z}^\top \boldsymbol{Z})^{-1}\boldsymbol{Z}^\top \breve{\boldsymbol{W}}^{(t)}$
      $\boldsymbol{W}^{(t+1)} \leftarrow \tilde{\boldsymbol{U}} S_\lambda(\tilde{\boldsymbol{D}}) \tilde{\boldsymbol{V}}^\top$, where $\tilde{\boldsymbol{U}}\tilde{\boldsymbol{D}}\tilde{\boldsymbol{V}}^\top$ is the SVD of $\tilde{\boldsymbol{W}}^{(t)}$
      $t \leftarrow t + 1$
    end while
Output $\hat{\boldsymbol{W}} = \boldsymbol{W}^{(t)}$, $\hat{\boldsymbol{X}} = P_\Omega(\boldsymbol{X}) + P_\Omega^\perp(\hat{\boldsymbol{W}})$

B.4 Additional high-dimensional simulation results

B.4.1 Data generating mechanism

In addition to applying SMC to the same high-dimensional simulations in Vu et al. (2019), we generate a new set of simulations with non-sparse loadings. The multi-pollutant surfaces have $p = 20$ components. We first generate four underlying PC scores on a 100×100 grid ($N = 10,000$) such that

$$\begin{aligned}
&\boldsymbol{u}_j \sim \mathcal{N}\left(\boldsymbol{R}_j \boldsymbol{b}_j, \boldsymbol{S}_j\right), \quad \text{for } j = 1, 2, 3, 4, \\
&\boldsymbol{R}_1 = \begin{bmatrix} \boldsymbol{r}_{1o} & \boldsymbol{r}_{1u} \end{bmatrix}, \quad \text{where } \boldsymbol{r}_{1o}, \boldsymbol{r}_{1u} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{1}), \quad \boldsymbol{b}_1^{\mathsf{T}} = \begin{bmatrix} 5 & 0 \end{bmatrix}, \\
&\boldsymbol{R}_2 = \begin{bmatrix} \boldsymbol{r}_{2o} & \boldsymbol{r}_{2u} \end{bmatrix}, \quad \text{where } \boldsymbol{r}_{2o}, \boldsymbol{r}_{2u} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{1}), \quad \boldsymbol{b}_2^{\mathsf{T}} = \begin{bmatrix} 5 & 1 \end{bmatrix}, \\
&\boldsymbol{R}_3 = \begin{bmatrix} \boldsymbol{r}_{3u} \end{bmatrix}, \quad \text{where } \boldsymbol{r}_{3u} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{1}), \quad \boldsymbol{b}_3 = 1, \\
&\boldsymbol{R}_4 = \begin{bmatrix} \boldsymbol{r}_{4u} \end{bmatrix}, \quad \text{where } \boldsymbol{r}_{4u} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{1}), \quad \boldsymbol{b}_4 = 1.
\end{aligned}$$

In this setting, $\boldsymbol{r}_{jo}$'s are GIS covariates observed for the model. Meanwhile $\boldsymbol{r}_{ju}$'s are unobserved covariates, and used primarily to generate the scores. That is, only $\boldsymbol{R} = \begin{bmatrix} \boldsymbol{r}_{1o} & \boldsymbol{r}_{2o} \end{bmatrix}$ is used in the universal kriging model for spatial prediction. For the variances, $\boldsymbol{S}_1$ has exponential structure with no nugget effect, partial sill of 1, and range of 50. Meanwhile, $\boldsymbol{S}_2 = 3\boldsymbol{I}_N$ and $\boldsymbol{S}_3 = \boldsymbol{S}_4 = 2.5\boldsymbol{I}_N$. This setup is designed such that $\boldsymbol{u}_1$ can be predicted easily at new locations, i.e. the most spatially predictable. On the other hand, $\boldsymbol{u}_2$ is moderately predictable in space, while $\boldsymbol{u}_3$ and $\boldsymbol{u}_4$ are not spatially predictable. Each score contributes an equal variance of 20. Finally, the multi-pollutant exposure surface is generated as

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V} + \boldsymbol{E}, \quad \text{where } E_{ij} \sim \mathcal{N}(0, 5),$$

with the following loadings:

$$\boldsymbol{V} = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \boldsymbol{v}_3 \end{bmatrix}, \text{ where } \boldsymbol{v}_j = \frac{\breve{\boldsymbol{v}}_j}{\left\|\breve{\boldsymbol{v}}_j\right\|_2}, \text{ for } j = 1, 2, 3$$

$$\breve{\boldsymbol{v}}_1^{\mathsf{T}} = \begin{bmatrix} 5 & 5 & 5 & 5 & 5 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \end{bmatrix}$$

$$\breve{\boldsymbol{v}}_2^{\mathsf{T}} = \begin{bmatrix} 2 & 2 & 2 & 2 & 2 & -5 & -5 & -5 & -5 & -5 & 2 & 2 & 2 & 2 & 2 & -2 & -2 & -2 & -2 & -2 \end{bmatrix}$$

$$\breve{\boldsymbol{v}}_3^{\mathsf{T}} = \begin{bmatrix} -2 & -2 & -2 & -2 & -2 & 2 & 2 & 2 & 2 & 2 & 5 & 5 & 5 & 5 & 5 & -2 & -2 & -2 & -2 & -2 \end{bmatrix}$$

$$\breve{\boldsymbol{v}}_4^{\mathsf{T}} = \begin{bmatrix} -2 & -2 & -2 & -2 & -2 & -2 & -2 & -2 & -2 & -2 & 2 & 2 & 2 & 2 & 2 & 5 & 5 & 5 & 5 & 5 \end{bmatrix}$$

In this setting, because of the non-sparse loadings, it is not clear how different methods will behave. We still expect that, with complete data, all predictive methods (PredPCA, ProPrPCA, and SMC) will identify combinations of $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ for the first two PCs.

Similar to the high-dimensional simulations used in Vu et al. (2019), we consider three MCAR scenarios (30%, 35%, and 40%), and one MAR scenario. In the MAR scenario, we identify training locations with $\boldsymbol{r}_{1o}$ value larger than its sample 60th percentile, and among $\boldsymbol{x}_1$ through $\boldsymbol{x}_5$, 75% of these training locations become missing data. For the rest of the pollutants, from $\boldsymbol{x}_6$ to $\boldsymbol{x}_{20}$, each has 25% of its locations missing completely at random. This setup guarantees a mild spatial pattern in the missing data, as $\boldsymbol{x}_1$ to $\boldsymbol{x}_5$ are generated mostly by $\boldsymbol{u}_1$, which is the most predictable score based on $\boldsymbol{r}_{1o}$.

B.4.2 Results

Table B.1 shows the median R^2. As expected, the predictive performance is poor for PCA. Other methods seem to be able to identify the underlying scores of interest. Under MAR scenario, data among the first five pollutants are more likely to be missing at locations with extreme geographic covariate values. This setup leads to the decreases in median R^2's of PC1 for all methods but increases for PC2 in the results of ProPrPCA and SMC. For PredPCA, the median R^2 goes down for PC1 but is not compensated by an improvement in predicting PC2.

Table B.1: The median prediction R^2's across 1,000 simulations for high-dimensional scenario with data generated using non-sparse loadings. Under missing data scenarios, LRMC is used prior to either PCA or PredPCA.

PC1	Complete	MCAR 35%	MAR
PCA	0.36	0.37	0.21
PredPCA	0.75	0.72	0.67
ProPrPCA	0.75	0.74	0.68
SMC	0.75	0.75	0.69

PC2	Complete	MCAR 35%	MAR
PCA	0.35	0.33	0.20
PredPCA	0.69	0.65	0.65
ProPrPCA	0.69	0.68	0.72
SMC	0.69	0.68	0.70

These results are more understandable when we investigate the estimated loadings in these simulations. Figure B.1 shows the median estimated loadings for both PC1 and PC2 when training data are complete. In generally, PCA does not identify the correct underlying loadings of interest. PredPCA produces results that are almost close to the true values. Meanwhile ProPrPCA and SMC both give similar estimates that are most similar to the true values of $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$. While not shown in the figure, the variability of estimates between simulations are the highest in PCA results.

Figure B.2 displays the median estimated loadings under MCAR 35%. It is intriguing to note that, under MCAR 35%, PredPCA gives PC2 estimates that are slightly closer to the true values compared to its estimates under complete data scenario. ProPrPCA and SMC seem to maintain their decent performance for both PCs.

Figure B.3 shows the results under MAR scenario. Note that the "true" values displayed for the PCs are switched in this scenario: $\boldsymbol{v}_2$ for PC1 and $\boldsymbol{v}_1$ for PC2 instead. Under MAR, data among the first five pollutants are more likely to be missing at locations with extreme covariate values. These five pollutants are constituted by a large proportion of $\boldsymbol{u}_1$. Thus, this setting effectively alter the in-sample variance contribution of the original PC scores. It

is most likely that the predictive methods (PredPCA, ProPrPCA, and SMC) now estimate combinations of $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ for PC1 loadings that, on average, resemble like $\boldsymbol{v}_2$ the most. This can be seen in Figure B.3 where none of the methods produce estimates closely match with the true values of $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, but rather somewhere in between. Overall, the performance for PC1 goes down for all methods in Table B.1, but goes up decently in PC2 for ProPrPCA and SMC, due to their estimated loadings.

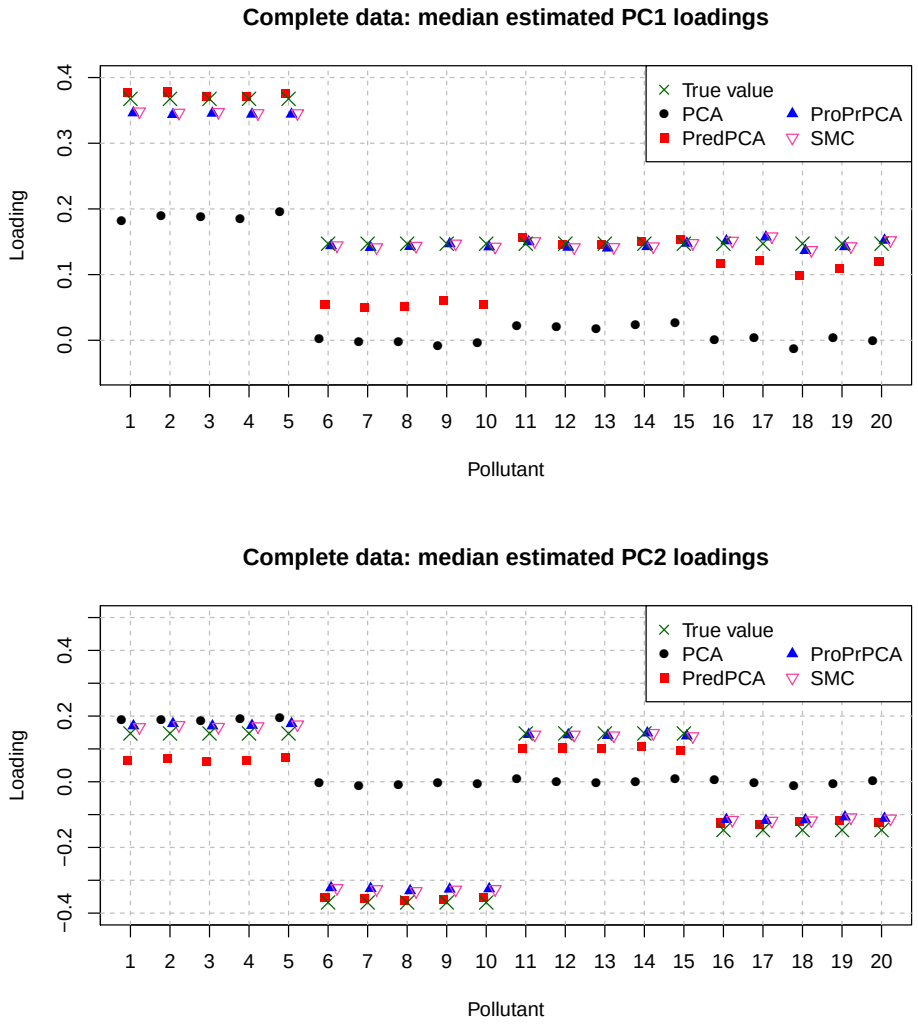

Figure B.1: Median estimated loadings for complete data scenario with data generated using non-sparse loadings. The true values are $\boldsymbol{v}_1$ for PC1 and $\boldsymbol{v}_2$ for PC2.

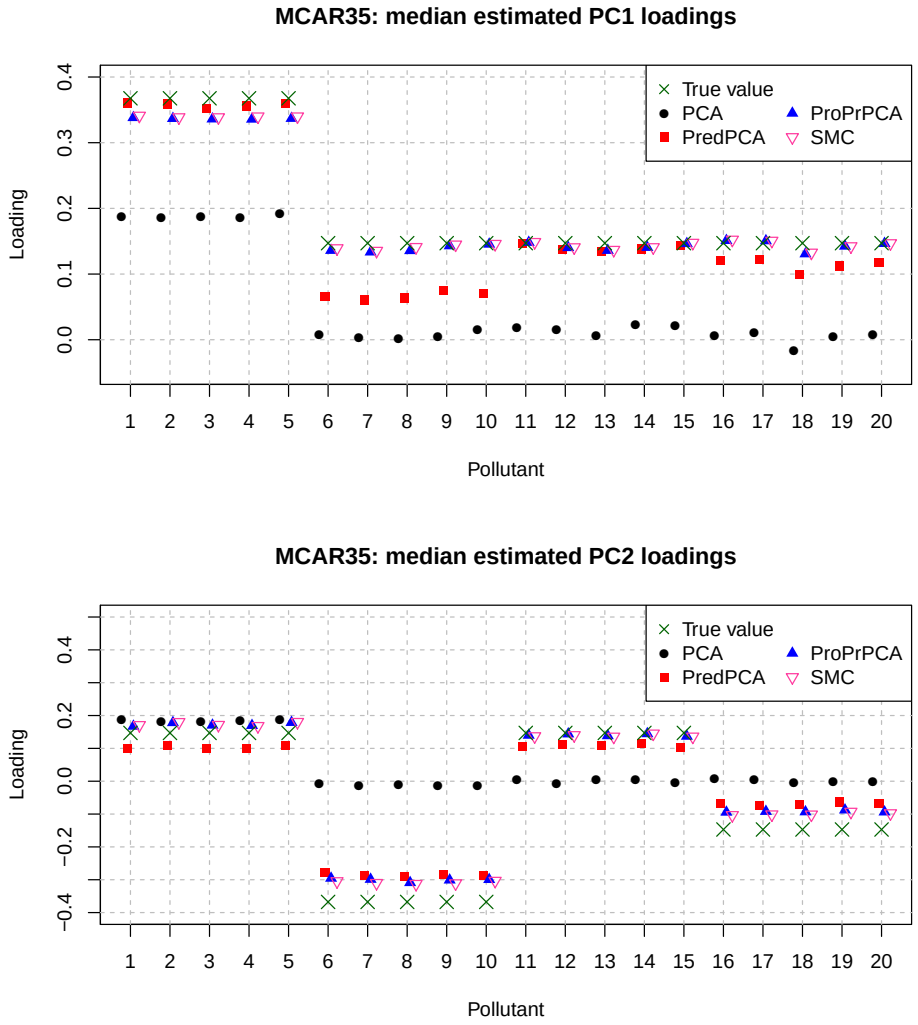

Figure B.2: Median estimated loadings for MCAR 35% scenario with data generated using non-sparse loadings. The true values are $\boldsymbol{v}_1$ for PC1 and $\boldsymbol{v}_2$ for PC2.

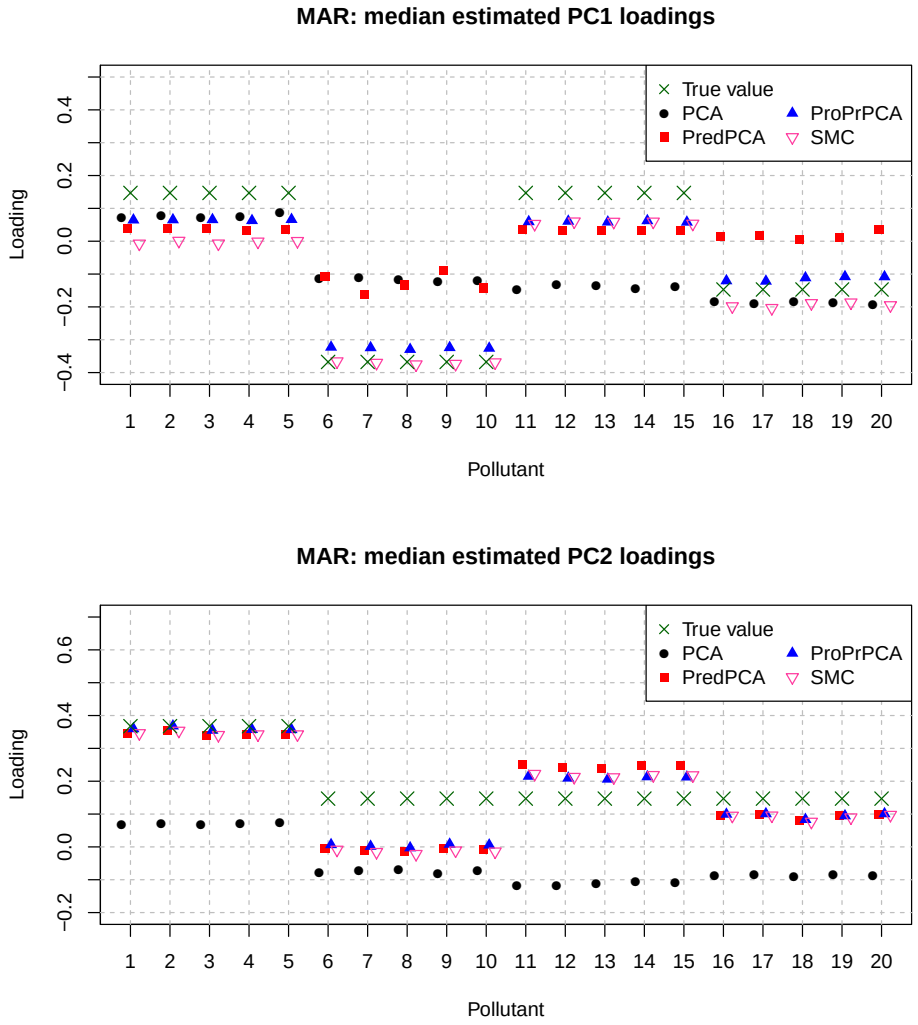

Figure B.3: Median estimated loadings for MAR scenario with data generated using non-sparse loadings. Note the switching in MAR scenario: the true values are $\boldsymbol{v}_2$ for PC1 and $\boldsymbol{v}_1$ for PC2.

The differences between ProPrPCA and PredPCA are further examined in Figure B.4. With complete data, the discrepancy between the two methods is negligible. Under MCAR 35%, ProPrPCA outperforms PredPCA for 67.4% of the 1,000 simulations. The differences are the most prominent under MAR, in which ProPrPCA performs worse than PredPCA for both PCs in only 1.4% of the simulations, and better for both PCs in 60.0% of the time.

The differences between SMC and PredPCA are evaluated in Figure B.4. Again they are very similar with complete data. Under MCAR 35% and MAR, SMC outperforms PredPCA in both PCS for 80.0% and 71.7% of the simulations. These percentages are slightly better than those of ProPrPCA in Figure B.4.

Finally, Figure B.6 shows the differences between SMC and ProPrPCA. The two methods are on par with each other throughout all missing data scenarios.

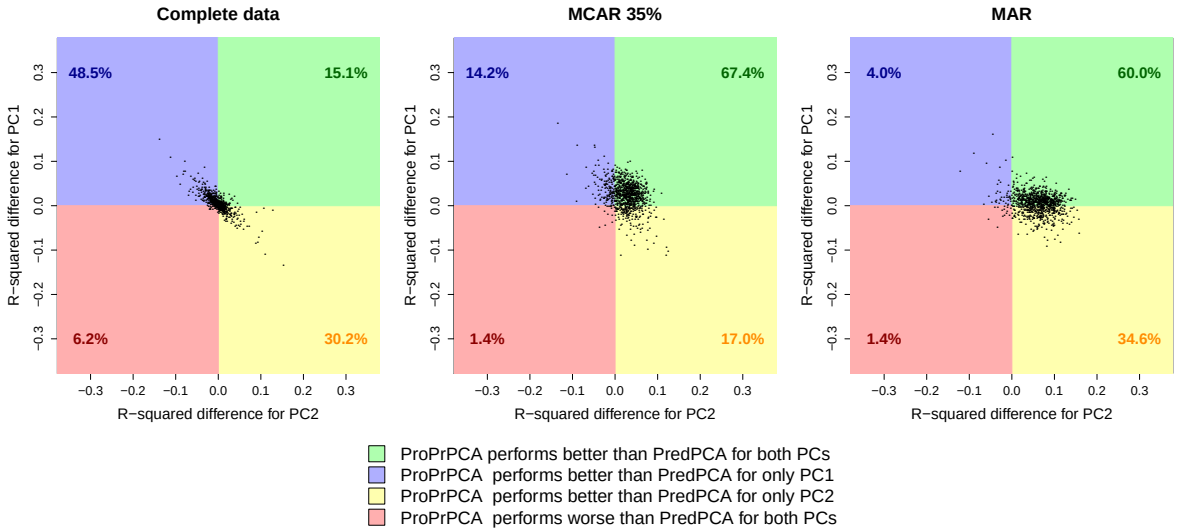

Figure B.4: Difference in R^2 between ProPrPCA and PredPCA for high-dimensional scenario with data generated using non-sparse loadings. Each dot represents result from one simulation. Percentages indicate the proportion out of 1,000 simulations.

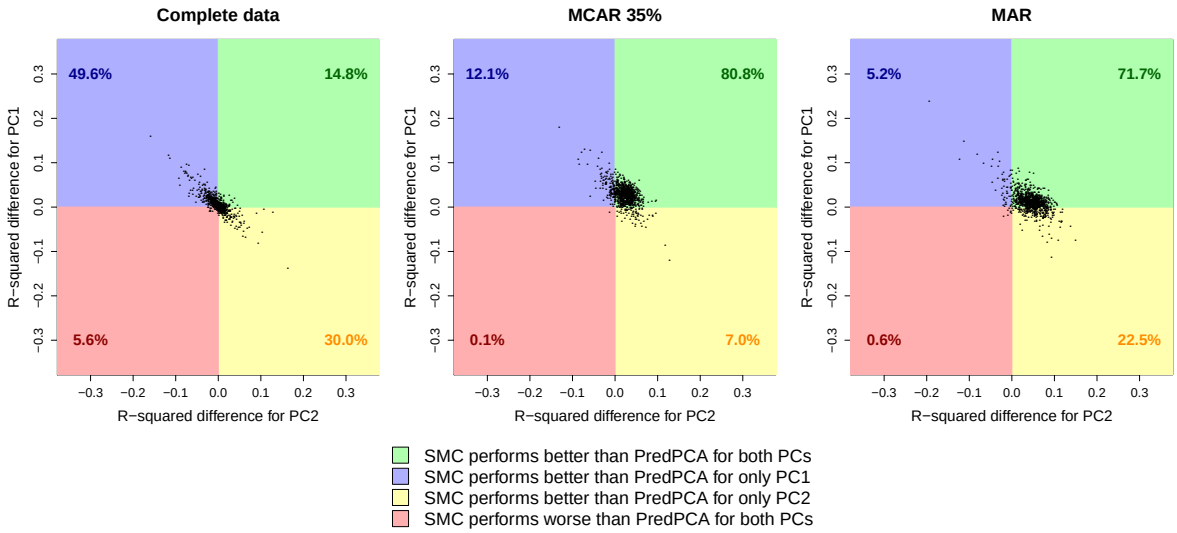

Figure B.5: Difference in R^2 between SMC and PredPCA for high-dimensional scenario with data generated using non-sparse loadings. Each dot represents result from one simulation. Percentages indicate the proportion out of 1,000 simulations.

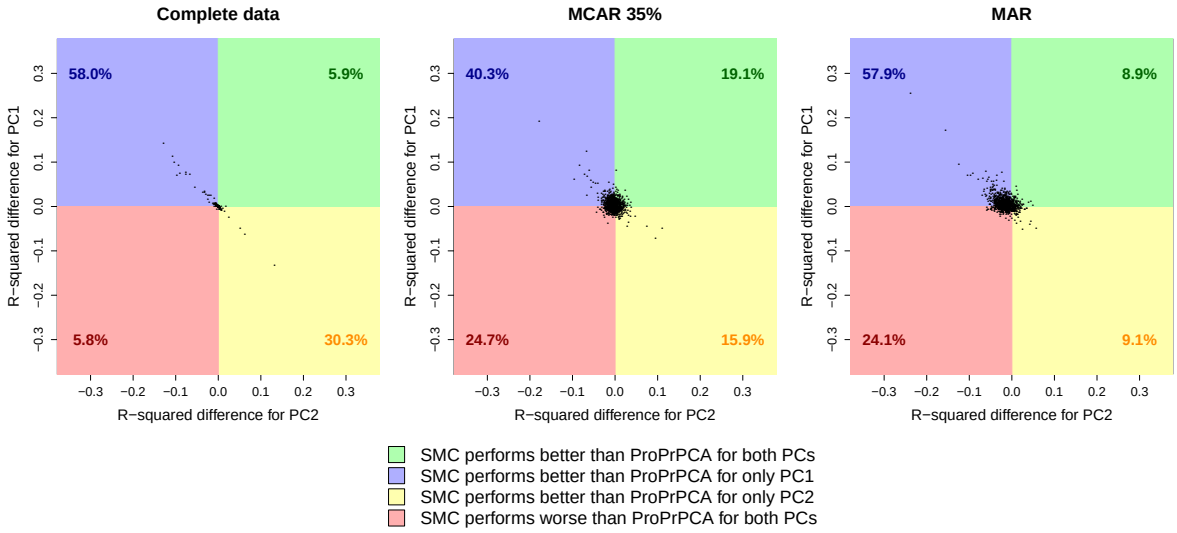

Figure B.6: Difference in R^2 between SMC and ProPrPCA for high-dimensional scenario with data generated using non-sparse loadings. Each dot represents result from one simulation. Percentages indicate the proportion out of 1,000 simulations.

B.5 Additional results for 2011 CSN data

In this section, we present the results of applying the dimension reduction methods to CSN data collected in 2011. This dataset has a total of 208 sites, only 128 of which have complete data for all 21 components. LRMC is used prior to implementing PCA or PredPCA when using the entire dataset.

Figure B.7 shows the estimated loadings and spatial distributions of scores for the feature that is highly positive on SO_4^{2-} and S, which corresponds to PC1 for all combinations of method and dataset. The maps of the PC scores are very similar with each other. However, the loadings obtained by PCA are different those estimated by the other three methods.

Figure B.8 shows the results for feature that has strongly positive weights of Na, Ni, and V. This corresponds to the third PC obtained by PCA, and the second PC obtained by PredPCA, ProPrPCA, and SMC. For this dataset, there is very little difference among the estimated loadings by the predictive methods.

Figure B.9 shows the results for feature that has strongly positive weights of NO_3^- and Zn. While the differences among the loadings estimated by PredPCA, ProPrPCA, and SMC are small, estimates by SMC are more similar to those produced by ProPrPCA, unlike when using 2010 CSN data.

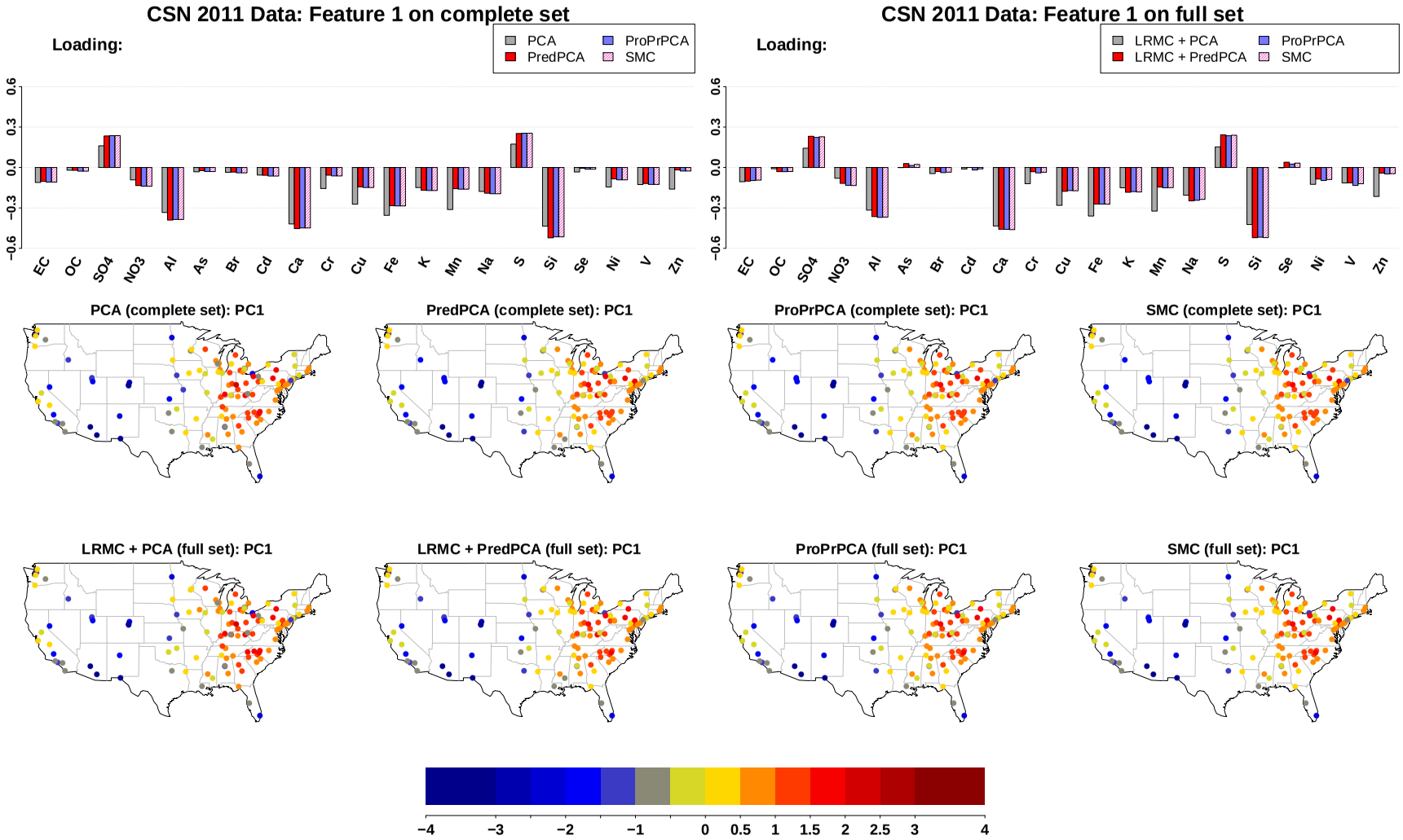

Figure B.7: Estimated loadings for feature with highly positive weights on SO_4^{2-} and S, and corresponding scores, obtained from different PCA algorithms (PCA, PredPCA, ProPrPCA, and SMC) applied to 2011 CSN data: the complete set (128 sites with complete data) or the full set (all 208 available sites). The top panel of bar plots display the loadings obtained from dimension reduction techniques. The bottom two panels of maps illustrate the scores corresponding to the estimated loadings.

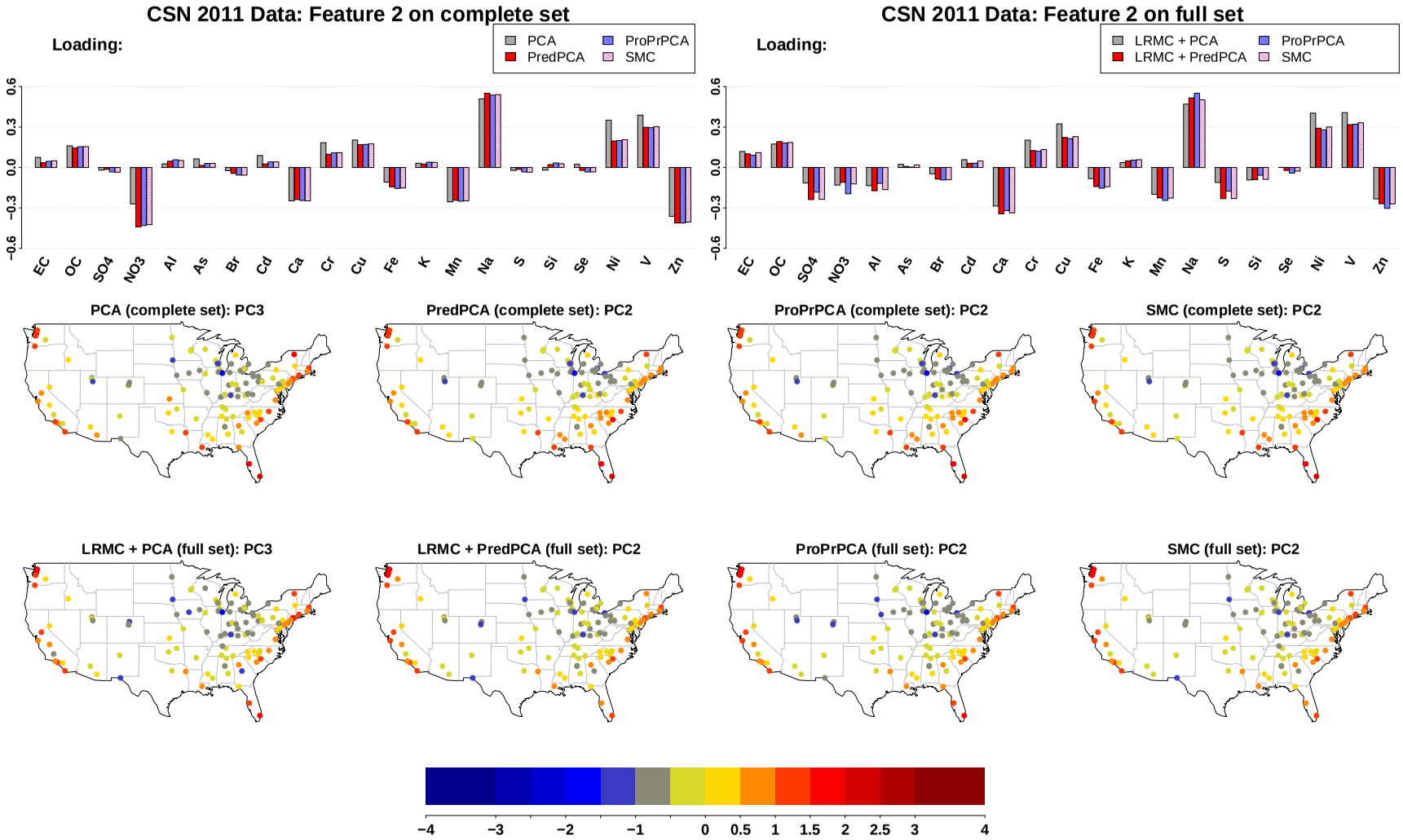

Figure B.8: Estimated loadings for feature with highly positive weights on Na, Ni, and V, and corresponding scores, obtained from different PCA algorithms (PCA, PredPCA, ProPrPCA, and SMC) applied to 2011 CSN data: the complete set (128 sites with complete data) or the full set (all 208 available sites). The top panel of bar plots display the loadings obtained from dimension reduction techniques. The bottom two panels of maps illustrate the scores corresponding to the estimated loadings.

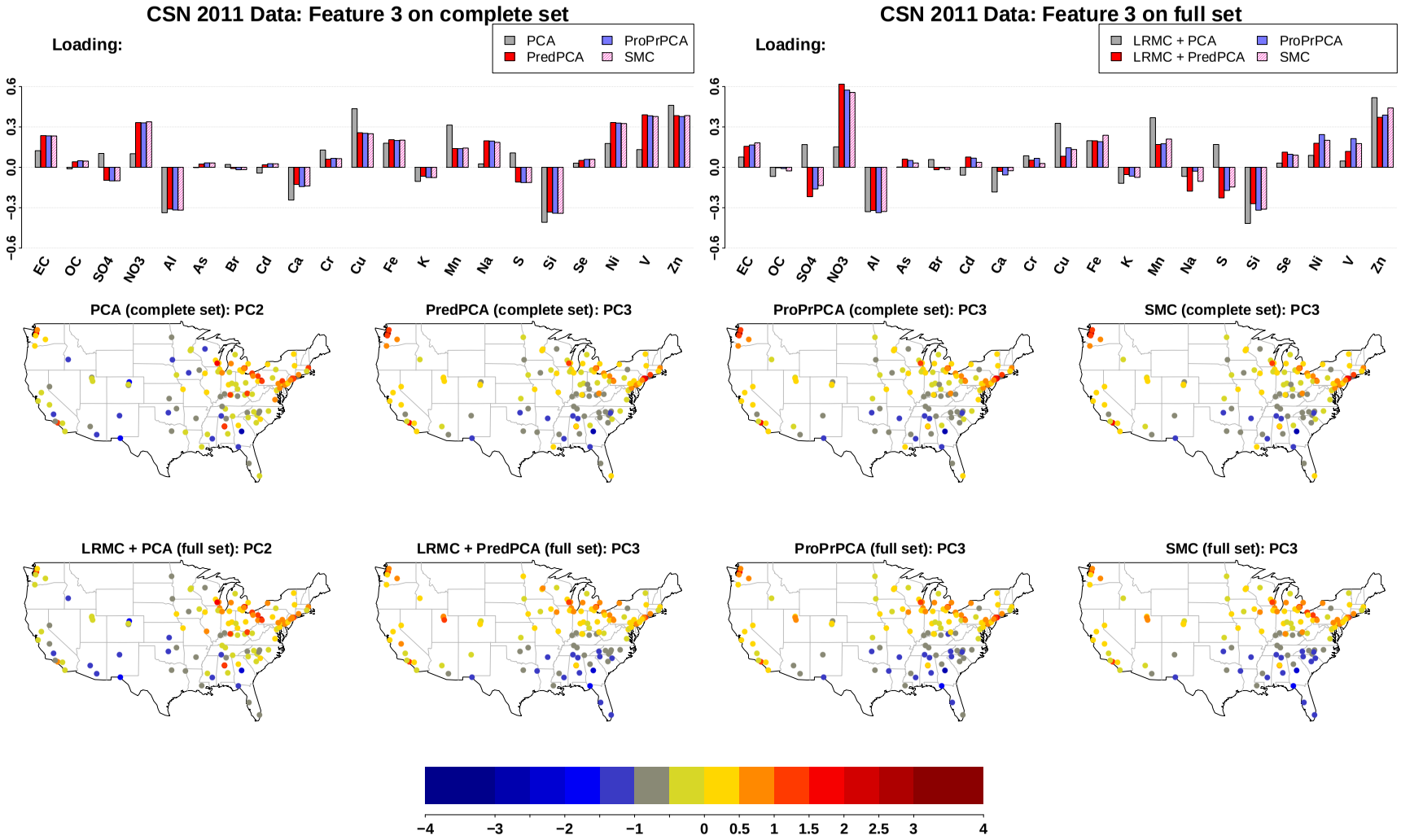

Figure B.9: Estimated loadings for feature with highly positive weights on NO_3^- and Zn, and corresponding scores, obtained from different PCA algorithms (PCA, PredPCA, ProPrPCA, and SMC) applied to 2011 CSN data: the complete set (128 sites with complete data) or the full set (all 208 available sites). The top panel of bar plots display the loadings obtained from dimension reduction techniques. The bottom two panels of maps illustrate the scores corresponding to the estimated loadings.

Appendix C

APPENDIX FOR CHAPTER 4

C.1 Proofs for the higher-order orthogonal iteration for Tucker decomposition

We would like to solve:

$$\begin{aligned} \min_{\mathcal{G},\boldsymbol{A}_1,\boldsymbol{A}_2,\boldsymbol{A}_3} \quad & \left\|\mathcal{X} - [\![\mathcal{G}; \boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3]\!]\right\|^2 \\ \text{s.t.} \quad & \mathcal{G} \in \mathbb{R}^{J_1 \times J_2 \times J_3} \\ & \boldsymbol{A}_n \in \mathbb{R}^{I_n \times J_n} \text{ orthonormal where } n = 1, 2, 3 \end{aligned}$$

We present the same proofs as in Kolda and Bader (2009) with more details.

Solving for $\mathcal{G}$:

When fixing $\boldsymbol{A}_n$'s, we can rearrange the original objective function using property of tensor matricization:

$$\left\|\mathcal{X} - [\![\mathcal{G}; \boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3]\!]\right\|^2 = \left\|\text{vec}(\mathcal{X}) - (\boldsymbol{A}_3 \otimes \boldsymbol{A}_2 \otimes \boldsymbol{A}_1)\,\text{vec}(\mathcal{G})\right\|^2 = \left\|\text{vec}(\mathcal{X}) - \boldsymbol{W}\text{vec}(\mathcal{G})\right\|^2$$

which is simply a least-square problem, where the solution is:

$$\begin{aligned}
\operatorname{vec}(\hat{\mathcal{G}}) &= (\boldsymbol{W}^{\mathsf{T}}\boldsymbol{W})^{-1}\boldsymbol{W}^{\mathsf{T}}\operatorname{vec}(\mathcal{X}) \\
&= \left[(\boldsymbol{A}_3 \otimes \boldsymbol{A}_2 \otimes \boldsymbol{A}_1)^{\mathsf{T}} (\boldsymbol{A}_3 \otimes \boldsymbol{A}_2 \otimes \boldsymbol{A}_1)\right]^{-1} (\boldsymbol{A}_3 \otimes \boldsymbol{A}_2 \otimes \boldsymbol{A}_1)^{\mathsf{T}} \operatorname{vec}(\mathcal{X}) \\
&= \left[\left(\boldsymbol{A}_3^{\mathsf{T}} \otimes \boldsymbol{A}_2^{\mathsf{T}} \otimes \boldsymbol{A}_1^{\mathsf{T}}\right) (\boldsymbol{A}_3 \otimes \boldsymbol{A}_2 \otimes \boldsymbol{A}_1)\right]^{-1} \left(\boldsymbol{A}_3^{\mathsf{T}} \otimes \boldsymbol{A}_2^{\mathsf{T}} \otimes \boldsymbol{A}_1^{\mathsf{T}}\right) \operatorname{vec}(\mathcal{X}) \\
&= \left[\left(\boldsymbol{A}_3^{\mathsf{T}}\boldsymbol{A}_3\right) \otimes \left(\boldsymbol{A}_2^{\mathsf{T}}\boldsymbol{A}_2\right) \otimes \left(\boldsymbol{A}_1^{\mathsf{T}}\boldsymbol{A}_1\right)\right]^{-1} \left(\boldsymbol{A}_3^{\mathsf{T}} \otimes \boldsymbol{A}_2^{\mathsf{T}} \otimes \boldsymbol{A}_1^{\mathsf{T}}\right) \operatorname{vec}(\mathcal{X}) \\
&= \left[\boldsymbol{I}_{J_3} \otimes \boldsymbol{I}_{J_2} \otimes \boldsymbol{I}_{J_1}\right]^{-1} \left(\boldsymbol{A}_3^{\mathsf{T}} \otimes \boldsymbol{A}_2^{\mathsf{T}} \otimes \boldsymbol{A}_1^{\mathsf{T}}\right) \operatorname{vec}(\mathcal{X}) \\
&= \left(\boldsymbol{A}_3^{\mathsf{T}} \otimes \boldsymbol{A}_2^{\mathsf{T}} \otimes \boldsymbol{A}_1^{\mathsf{T}}\right) \operatorname{vec}(\mathcal{X}) \\
\implies \hat{\mathcal{G}} &= \mathcal{X} \times_1 \boldsymbol{A}_1^{\mathsf{T}} \times_2 \boldsymbol{A}_2^{\mathsf{T}} \times \boldsymbol{A}_3^{\mathsf{T}}
\end{aligned} \tag{C.1}$$

Solving for $\boldsymbol{A}_n$'s:

We will use the optimal solution for $\mathcal{G}$ above throughout the remainder of this proof. Using norm properties of tensor, we can rewrite the original problem, given $\hat{\mathcal{G}}$, as

$$\min_{\boldsymbol{A}_1,\boldsymbol{A}_2,\boldsymbol{A}_3} \quad \left\|\mathcal{X} - [\![\hat{\mathcal{G}}; \boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3]\!]\right\|^2 \tag{C.2}$$

$$= \min_{\boldsymbol{A}_1,\boldsymbol{A}_2,\boldsymbol{A}_3} \left\{\|\mathcal{X}\|^2 - 2\left\langle \mathcal{X}, [\![\hat{\mathcal{G}}; \boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3]\!]\right\rangle + \left\|[\![\hat{\mathcal{G}}; \boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3]\!]\right\}$$

$$= \min_{\boldsymbol{A}_1,\boldsymbol{A}_2,\boldsymbol{A}_3} \left\{-2\left\langle \mathcal{X}, [\![\hat{\mathcal{G}}; \boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3]\!]\right\rangle + \left\|[\![\hat{\mathcal{G}}; \boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3]\!]\right\|^2\right\} \tag{C.3}$$

The second term in (C.3) can be written as:

$$\begin{aligned}
&-2\left\langle \mathcal{X}, [\![\hat{\mathcal{G}}; \boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3]\!]\right\rangle \\
&= -2\left\langle \mathcal{X}, \hat{\mathcal{G}} \times_1 \boldsymbol{A}_1 \times_2 \boldsymbol{A}_2 \times_3 \boldsymbol{A}_3\right\rangle \\
&= -2\left\langle \mathcal{X} \times_1 \boldsymbol{A}_1^{\mathsf{T}} \times_2 \boldsymbol{A}_2^{\mathsf{T}} \times_3 \boldsymbol{A}_3^{\mathsf{T}}, \hat{\mathcal{G}}\right\rangle \quad \text{(Property of tensor inner product)} \\
&= -2\left\langle \hat{\mathcal{G}}, \hat{\mathcal{G}}\right\rangle \quad \text{(By result (C.1))} \\
&= -2\left\|\hat{\mathcal{G}}\right\|^2
\end{aligned} \tag{C.4}$$

The third term in (C.3) can be written as:

$$\left\|[\![\hat{\mathcal{G}}; \boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3]\!]\right\|^2 = \left\|\hat{\mathcal{G}} \times_1 \boldsymbol{A}_1 \times_2 \boldsymbol{A}_2 \times_3 \boldsymbol{A}_3\right\|^2 = \left\|\hat{\mathcal{G}}\right\|^2 \tag{C.5}$$

by the orthogonality property of tensor norm. Thus, substituting (C.4) and (C.5) into (C.6), we have:

$$\begin{aligned}
&\min_{\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3} \left\|\mathcal{X} - [\![\hat{\mathcal{G}}; \boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3]\!]\right\|^2 \\
= &\min_{\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3} \left\{-2\left\langle \mathcal{X}, [\![\hat{\mathcal{G}}; \boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3]\!]\right\rangle + \left\|[\![\hat{\mathcal{G}}; \boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3]\!]\right\|^2\right\} \\
= &\min_{\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3} \left\{-2\|\hat{\mathcal{G}}\|^2 + \|\hat{\mathcal{G}}\|^2\right\} \\
= &\max_{\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3} \|\hat{\mathcal{G}}\|^2 \\
= &\max_{\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3} \left\|\mathcal{X} \times_1 \boldsymbol{A}_1^\mathsf{T} \times_2 \boldsymbol{A}_2^\mathsf{T} \times_3 \boldsymbol{A}_3^\mathsf{T}\right\|^2
\end{aligned}$$

Given estimates $\widehat{\boldsymbol{A}}_2$ and $\widehat{\boldsymbol{A}}_3$, consider the problem:

$$\max_{\boldsymbol{A}_1} \left\|\mathcal{X} \times_1 \boldsymbol{A}_1^\mathsf{T} \times_2 \widehat{\boldsymbol{A}}_2^\mathsf{T} \times_3 \widehat{\boldsymbol{A}}_3^\mathsf{T}\right\|^2$$

subject to $\boldsymbol{A}_1 \in \mathbb{R}^{I_1 \times J_1}$, columns of $\boldsymbol{A}_1$ are orthogonal, and $\sum_{i=1}^{I_1} A_{ij}^2 = 1$. We can rewrite the objective function as follows:

$$\begin{aligned}
\left\|\mathcal{X} \times_1 \boldsymbol{A}_1^\mathsf{T} \times_2 \widehat{\boldsymbol{A}}_2^\mathsf{T} \times_3 \widehat{\boldsymbol{A}}_3^\mathsf{T}\right\|^2 &= \left\| \left(\mathcal{X} \times_2 \widehat{\boldsymbol{A}}_2^\mathsf{T} \times_3 \widehat{\boldsymbol{A}}_3^\mathsf{T}\right) \times_1 \boldsymbol{A}_1^\mathsf{T}\right\|^2 \\
&= \left\|\mathcal{W} \times_1 \boldsymbol{A}_1^\mathsf{T}\right\|^2 \text{ where } \mathcal{W} = \mathcal{X} \times_2 \widehat{\boldsymbol{A}}_2^\mathsf{T} \times_3 \widehat{\boldsymbol{A}}_3^\mathsf{T} \\
&= \left\|\boldsymbol{A}_1^\mathsf{T} \boldsymbol{W}_{(1)}\right\|_F^2 \text{ (Property of tensor multiplication)} \\
&= \mathrm{Tr}\left[\left(\boldsymbol{A}_1^\mathsf{T} \boldsymbol{W}_{(1)}\right)\left(\boldsymbol{A}_1^\mathsf{T} \boldsymbol{W}_{(1)}\right)^\mathsf{T}\right] \\
&= \mathrm{Tr}\left[\boldsymbol{A}_1^\mathsf{T} \boldsymbol{W}_{(1)} \boldsymbol{W}_{(1)}^\mathsf{T} \boldsymbol{A}_1\right]
\end{aligned}$$

Denote $\boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\mathsf{T}$ as the singular value decomposition (SVD) of $\boldsymbol{W}_{(1)}$. Here $\boldsymbol{U} \in \mathbb{R}^{I_1 \times J_1}$, $\boldsymbol{D} \in \mathbb{R}^{J_1 \times J_1}$ has singular values on the diagonal and zeros elsewhere, and $\boldsymbol{V} \in \mathbb{R}^{J_2 J_3 \times J_1}$. Columns of $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal and of unit length, i.e. $\sum_{i=1}^{I_1} U_{ij}^2 = 1$. Using the SVD of $\boldsymbol{W}_{(1)}$, we can further rearrange the expression above:

$$
\begin{aligned}
& \operatorname{Tr}\left[\boldsymbol{A}_1^\mathsf{T} \boldsymbol{W}_{(1)} \boldsymbol{W}_{(1)}^\mathsf{T} \boldsymbol{A}_1\right] \\
&= \operatorname{Tr}\left[\boldsymbol{A}_1^\mathsf{T} \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\mathsf{T} \boldsymbol{V}\boldsymbol{D}\boldsymbol{U}^\mathsf{T} \boldsymbol{A}_1\right] \\
&= \operatorname{Tr}\left[\boldsymbol{A}_1^\mathsf{T} \boldsymbol{U}\boldsymbol{D}^2\boldsymbol{U}^\mathsf{T} \boldsymbol{A}_1\right] \\
&= \operatorname{Tr}\left[\boldsymbol{B} \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_{J_1} \end{bmatrix} \boldsymbol{B}^\mathsf{T}\right] \\
&\qquad \text{(Where } \boldsymbol{B} = \boldsymbol{A}_1^\mathsf{T}\boldsymbol{U} \text{ and } \lambda_1, \dots, \lambda_{J_1} \text{ are the eigenvalues of } \boldsymbol{W}_{(1)}\text{)} \\
&= \sum_{j=1}^{J_1} \lambda_j B_{jj}^2 \\
&= \sum_{j=1}^{J_1} \lambda_j \left(\sum_{i=1}^{I_1} A_{ij} U_{ij}\right)^2 \\
&\le \sum_{j=1}^{J_1} \lambda_j \left(\sum_{i=1}^{I_1} A_{ij}^2\right)\left(\sum_{i=1}^{I_1} U_{ij}^2\right) \quad \text{(By Cauchy-Schwarz inequality)} \\
&= \sum_{j=1}^{J_1} \lambda_j
\end{aligned}
$$

As a result, the expression $\left\|\boldsymbol{\mathcal{X}} \times_1 \boldsymbol{A}_1^\mathsf{T} \times_2 \widehat{\boldsymbol{A}}_2^\mathsf{T} \times_3 \widehat{\boldsymbol{A}}_3^\mathsf{T}\right\|^2$ is maximized when $B_{jj} = 1$. This is achieved when $\boldsymbol{A}_1 = \boldsymbol{U}$, i.e. the solution for $\boldsymbol{A}_1$ is the J_1 left singular vectors of the mode-1 unfolding matrix of $\boldsymbol{\mathcal{X}} \times_2 \widehat{\boldsymbol{A}}_2^\mathsf{T} \times_3 \widehat{\boldsymbol{A}}_3^\mathsf{T}$. In computation, we prefer to use matrices instead of high-dimensional arrays. Using properties of tensor multiplication, the mode-1 unfolding matrix of $\boldsymbol{\mathcal{X}} \times_2 \widehat{\boldsymbol{A}}_2^\mathsf{T} \times_3 \widehat{\boldsymbol{A}}_3^\mathsf{T}$ is simply $\boldsymbol{X}_{(1)}\left(\boldsymbol{A}_3^{(l)} \otimes \boldsymbol{A}_2^{(l)}\right)$. Thus the Tucker decomposition

algorithm can be summarized in the following steps:

The Tucker decomposition algorithm:

Input $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, desired output rank $\{J_1 \times J_2 \times J_3\}$, and l_{max}

Initialize $l = 1$,

$\boldsymbol{A}_1^{(0)} = J_1$ left singular vectors of $\boldsymbol{X}_{(1)}$,

$\boldsymbol{A}_2^{(0)} = J_2$ left singular vectors of $\boldsymbol{X}_{(2)}$,

$\boldsymbol{A}_3^{(0)} = J_3$ left singular vectors of $\boldsymbol{X}_{(3)}$,

while not converged or $l < l_{max}$ **do**

$\boldsymbol{A}_1^{(l+1)} \leftarrow J_1$ left singular vectors of $\boldsymbol{X}_{(1)} \left(\boldsymbol{A}_3^{(l)} \otimes \boldsymbol{A}_2^{(l)} \right)$

$\boldsymbol{A}_2^{(l+1)} \leftarrow J_2$ left singular vectors of $\boldsymbol{X}_{(2)} \left(\boldsymbol{A}_3^{(l)} \otimes \boldsymbol{A}_1^{(l)} \right)$

$\boldsymbol{A}_3^{(l+1)} \leftarrow J_3$ left singular vectors of $\boldsymbol{X}_{(3)} \left(\boldsymbol{A}_2^{(l)} \otimes \boldsymbol{A}_1^{(l)} \right)$

$l \leftarrow l + 1$

end while

Output $\widehat{\boldsymbol{A}}_1 = \boldsymbol{A}_1^{(l)}$, $\widehat{\boldsymbol{A}}_2 = \boldsymbol{A}_2^{(l)}$, $\widehat{\boldsymbol{A}}_3 = \boldsymbol{A}_3^{(l)}$, and $\hat{\mathcal{G}} = \mathcal{X} \times_1 \widehat{\boldsymbol{A}}_1^{\mathsf{T}} \times_2 \widehat{\boldsymbol{A}}_2^{\mathsf{T}} \times_3 \widehat{\boldsymbol{A}}_3^{\mathsf{T}}$

C.2 Proofs for our proposed algorithm

We would like to solve the following problem:

$$\begin{aligned} \min_{\mathcal{G}, \boldsymbol{V}_t, \boldsymbol{V}_p} \quad & \left\| \mathcal{X} - [\![\mathcal{G}; \boldsymbol{Z}, \boldsymbol{V}_t, \boldsymbol{V}_p]\!] \right\|^2 \\ \text{s.t.} \quad & \mathcal{G} \in \mathbb{R}^{K \times M \times Q}, M < T, Q < P \\ & \boldsymbol{V}_t \in \mathbb{R}^{T \times M} \text{ orthonormal} \\ & \boldsymbol{V}_p \in \mathbb{R}^{P \times Q} \text{ orthonormal} \end{aligned}$$

Solving for $\mathcal{G}$:

When fixing $\boldsymbol{V}_t$ and $\boldsymbol{V}_p$, we can rearrange the original objective function using property (b) in section 3.2 (matricization of a tensor):

$$\left\| \mathcal{X} - [\![\mathcal{G}; \boldsymbol{Z}, \boldsymbol{V}_t, \boldsymbol{V}_p]\!] \right\|^2 = \left\| \text{vec}(\mathcal{X}) - (\boldsymbol{V}_p \otimes \boldsymbol{V}_t \otimes \boldsymbol{Z}) \, \text{vec}(\mathcal{G}) \right\|^2 = \left\| \text{vec}(\mathcal{X}) - \boldsymbol{W} \text{vec}(\mathcal{G}) \right\|^2$$

where $\boldsymbol{W} = (\boldsymbol{V}_p \otimes \boldsymbol{V}_t \otimes \boldsymbol{Z})$ with the current values of $\boldsymbol{V}_t$ and $\boldsymbol{V}_t$. This is simply a least-square problem where the solution is $(\boldsymbol{W}^\mathsf{T}\boldsymbol{W})^{-1}\boldsymbol{W}^\mathsf{T}\text{vec}(\mathcal{X})$.

The size of $\boldsymbol{W}$ maybe computationally problematic. Given that $\boldsymbol{V}_t$ and $\boldsymbol{V}_p$ are orthonor-

mal, we can simplify the expression a bit further:

$$\begin{aligned}
\boldsymbol{W}^\mathsf{T}\boldsymbol{W} &= (\boldsymbol{V}_p \otimes \boldsymbol{V}_t \otimes \boldsymbol{Z})^\mathsf{T} (\boldsymbol{V}_p \otimes \boldsymbol{V}_t \otimes \boldsymbol{Z}) \\
&= \left(\boldsymbol{V}_p^\mathsf{T} \otimes \boldsymbol{V}_t^\mathsf{T} \otimes \boldsymbol{Z}^\mathsf{T}\right) (\boldsymbol{V}_p \otimes \boldsymbol{V}_t \otimes \boldsymbol{Z}) \\
&= \left((\boldsymbol{V}_p^\mathsf{T}\boldsymbol{V}_p) \otimes (\boldsymbol{V}_t^\mathsf{T}\boldsymbol{V}_t)\right) \otimes (\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z}) \\
&= (\boldsymbol{I}_Q \otimes \boldsymbol{I}_M) \otimes (\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z}) \\
&= \boldsymbol{I}_{QM} \otimes (\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z}) \\
\implies (\boldsymbol{W}^\mathsf{T}\boldsymbol{W})^{-1} &= \boldsymbol{I}_{QM} \otimes (\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z})^{-1} \\
\implies (\boldsymbol{W}^\mathsf{T}\boldsymbol{W})^{-1}\boldsymbol{W}^\mathsf{T} &= \boldsymbol{I}_{QM} \otimes (\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z})^{-1} \left(\boldsymbol{V}_p^\mathsf{T} \otimes \boldsymbol{V}_t^\mathsf{T} \otimes \boldsymbol{Z}^\mathsf{T}\right) \\
&= \boldsymbol{V}_p^\mathsf{T} \otimes \boldsymbol{V}_t^\mathsf{T} \otimes \left((\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z})^{-1}\boldsymbol{Z}^\mathsf{T}\right)
\end{aligned}$$

Using this expression, the most burdensome calculation will be finding the inverse of $(\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z})$, which is of much smaller in size than $\boldsymbol{W}^\mathsf{T}\boldsymbol{W}$.

To sum up, the optimal solution for $\mathcal{G}$, given current estimates of $\boldsymbol{V}_t$ and $\boldsymbol{V}_p$, is:

$$\text{vec}(\hat{\mathcal{G}}) = \left(\boldsymbol{V}_p^\mathsf{T} \otimes \boldsymbol{V}_t^\mathsf{T} \otimes \left((\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z})^{-1}\boldsymbol{Z}^\mathsf{T}\right)\right) \text{vec}(\mathcal{X}),$$

and by the same property of mode-n multiplication, the result can be written as:

$$\hat{\mathcal{G}} = \mathcal{X} \times_1 \left((\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z})^{-1}\boldsymbol{Z}^\mathsf{T}\right) \times_2 \boldsymbol{V}_t^\mathsf{T} \times_3 \boldsymbol{V}_p^\mathsf{T}$$

Solving for $\boldsymbol{V}_t$ and $\boldsymbol{V}_p$:

We will use the optimal solution for $\mathcal{G}$ above throughout the remainder of this proof.

Using norm properties of tensor, we can rewrite the original problem, given $\hat{\mathcal{G}}$, as

$$\begin{aligned} & \min_{\boldsymbol{V}_t, \boldsymbol{V}_p} \left\| \mathcal{X} - [\![\hat{\mathcal{G}}; \boldsymbol{Z}, \boldsymbol{V}_t, \boldsymbol{V}_p]\!] \right\|^2 && \text{(C.6)} \\ = & \min_{\boldsymbol{V}_t, \boldsymbol{V}_p} \left\{ \|\mathcal{X}\|^2 - 2 \left\langle \mathcal{X}, [\![\hat{\mathcal{G}}; \boldsymbol{Z}, \boldsymbol{V}_t, \boldsymbol{V}_p]\!] \right\rangle + \left\| [\![\hat{\mathcal{G}}; \boldsymbol{Z}, \boldsymbol{V}_t, \boldsymbol{V}_p]\!] \right\| \right\} \\ = & \min_{\boldsymbol{V}_t, \boldsymbol{V}_p} \left\{ -2 \left\langle \mathcal{X}, [\![\hat{\mathcal{G}}; \boldsymbol{Z}, \boldsymbol{V}_t, \boldsymbol{V}_p]\!] \right\rangle + \left\| [\![\hat{\mathcal{G}}; \boldsymbol{Z}, \boldsymbol{V}_t, \boldsymbol{V}_p]\!] \right\|^2 \right\} && \text{(C.7)} \end{aligned}$$

From the solution for $\mathcal{G}$, we have:

$$\begin{aligned} & \hat{\mathcal{G}} = \mathcal{X} \times_1 \left((\boldsymbol{Z}^\top \boldsymbol{Z})^{-1} \boldsymbol{Z}^\top \right) \times_2 \boldsymbol{V}_t^\top \times_3 \boldsymbol{V}_p^\top \\ \Longrightarrow & \hat{\mathcal{G}} \times_1 \boldsymbol{I}_K = \left(\mathcal{X} \times_2 \boldsymbol{V}_t^\top \times_3 \boldsymbol{V}_p^\top \right) \times_1 \left((\boldsymbol{Z}^\top \boldsymbol{Z})^{-1} \boldsymbol{Z}^\top \right) \\ \Longrightarrow & \hat{\mathcal{G}} \times_1 \left((\boldsymbol{Z}^\top \boldsymbol{Z})^{-1} \boldsymbol{Z}^\top \boldsymbol{Z} \right) = \left(\mathcal{X} \times_2 \boldsymbol{V}_t^\top \times_3 \boldsymbol{V}_p^\top \right) \times_1 \left((\boldsymbol{Z}^\top \boldsymbol{Z})^{-1} \boldsymbol{Z}^\top \right) \\ \Longrightarrow & \left(\hat{\mathcal{G}} \times_1 \boldsymbol{Z} \right) \times_1 \left((\boldsymbol{Z}^\top \boldsymbol{Z})^{-1} \boldsymbol{Z}^\top \right) = \left(\mathcal{X} \times_2 \boldsymbol{V}_t^\top \times_3 \boldsymbol{V}_p^\top \right) \times_1 \left((\boldsymbol{Z}^\top \boldsymbol{Z})^{-1} \boldsymbol{Z}^\top \right) \\ \Longrightarrow & \hat{\mathcal{G}} \times_1 \boldsymbol{Z} = \mathcal{X} \times_2 \boldsymbol{V}_t^\top \times_3 \boldsymbol{V}_p^\top && \text{(C.8)} \end{aligned}$$

by properties of mode-n multiplications. Thus the second term in (C.7) can be written as:

$$\begin{aligned} & -2 \left\langle \mathcal{X}, [\![\hat{\mathcal{G}}; \boldsymbol{Z}, \boldsymbol{V}_t, \boldsymbol{V}_p]\!] \right\rangle \\ = & -2 \left\langle \mathcal{X}, \hat{\mathcal{G}} \times_1 \boldsymbol{Z} \times_2 \boldsymbol{V}_t \times_3 \boldsymbol{V}_p \right\rangle \\ = & -2 \left\langle \mathcal{X} \times_2 \boldsymbol{V}_t^\top \times_3 \boldsymbol{V}_p^\top, \hat{\mathcal{G}} \times_1 \boldsymbol{Z} \right\rangle \quad \text{(Property of tensor inner product)} \\ = & -2 \left\langle \hat{\mathcal{G}} \times_1 \boldsymbol{Z}, \hat{\mathcal{G}} \times_1 \boldsymbol{Z} \right\rangle \quad \text{(By result (C.8))} \\ = & -2 \left\| \hat{\mathcal{G}} \times_1 \boldsymbol{Z} \right\|^2 && \text{(C.9)} \end{aligned}$$

The third term in (C.7) can be written as:

$$\left\| [\![\hat{\mathcal{G}}; \boldsymbol{Z}, \boldsymbol{V}_t, \boldsymbol{V}_p]\!] \right\|^2 = \left\| \hat{\mathcal{G}} \times_1 \boldsymbol{Z} \times_2 \boldsymbol{V}_t \times_3 \boldsymbol{V}_p \right\|^2 = \left\| \hat{\mathcal{G}} \times_1 \boldsymbol{Z} \right\|^2 \qquad \text{(C.10)}$$

by the orthogonality property of tensor norm. Thus, substituting (C.9) and (C.10) into (C.6), we have:

$$
\begin{aligned}
&\min_{\boldsymbol{V}_t,\boldsymbol{V}_p} \left\|\mathcal{X} - [\![\hat{\mathcal{G}}; \boldsymbol{Z}, \boldsymbol{V}_t, \boldsymbol{V}_p]\!]\right\|^2 \\
=\ &\min_{\boldsymbol{V}_t,\boldsymbol{V}_p} \left\{-2\left\langle \mathcal{X}, [\![\hat{\mathcal{G}}; \boldsymbol{Z}, \boldsymbol{V}_t, \boldsymbol{V}_p]\!]\right\rangle + \left\|[\![\hat{\mathcal{G}}; \boldsymbol{Z}, \boldsymbol{V}_t, \boldsymbol{V}_p]\!]\right\|^2\right\} \\
=\ &\min_{\boldsymbol{V}_t,\boldsymbol{V}_p} \left\{-2\left\|\hat{\mathcal{G}} \times_1 \boldsymbol{Z}\right\|^2 + \left\|\hat{\mathcal{G}} \times_1 \boldsymbol{Z}\right\|^2\right\} \\
=\ &\min_{\boldsymbol{V}_t,\boldsymbol{V}_p} \left\{-\left\|\hat{\mathcal{G}} \times_1 \boldsymbol{Z}\right\|^2\right\} \\
=\ &\max_{\boldsymbol{V}_t,\boldsymbol{V}_p} \left\|\hat{\mathcal{G}} \times_1 \boldsymbol{Z}\right\|^2 \\
=\ &\max_{\boldsymbol{V}_t,\boldsymbol{V}_p} \left\| \left(\mathcal{X} \times_1 \left((\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z})^{-1}\boldsymbol{Z}^\mathsf{T}\right) \times_2 \boldsymbol{V}_t^\mathsf{T} \times_3 \boldsymbol{V}_p^\mathsf{T}\right) \times_1 \boldsymbol{Z}\right\|^2 \\
=\ &\max_{\boldsymbol{V}_t,\boldsymbol{V}_p} \left\|\mathcal{X} \times_1 \left(\boldsymbol{Z}(\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z})^{-1}\boldsymbol{Z}^\mathsf{T}\right) \times_2 \boldsymbol{V}_t^\mathsf{T} \times_3 \boldsymbol{V}_p^\mathsf{T}\right\|^2 \\
=\ &\max_{\boldsymbol{V}_t,\boldsymbol{V}_p} \left\|\tilde{\mathcal{X}} \times_2 \boldsymbol{V}_t^\mathsf{T} \times_3 \boldsymbol{V}_p^\mathsf{T}\right\|^2
\end{aligned}
$$

where $\tilde{\mathcal{X}} = \mathcal{X} \times_1 \left(\boldsymbol{Z}(\boldsymbol{Z}^\mathsf{T}\boldsymbol{Z})^{-1}\boldsymbol{Z}^\mathsf{T}\right)$, which is heuristically a mode-1 projection of $\mathcal{X}$ onto the column space of $\boldsymbol{Z}$.

Given optimal solution $\hat{\boldsymbol{V}}_p$, the solution for $\boldsymbol{V}_t$ will need to satisfy:

$$
\max_{\boldsymbol{V}_t} \left\|\tilde{\mathcal{X}} \times_2 \boldsymbol{V}_t^\mathsf{T} \times_3 \hat{\boldsymbol{V}}_p^\mathsf{T}\right\|^2
$$

Following similar proofs as in the previous section, the solution for $\boldsymbol{V}_t$ is the M left singular vectors of the mode-2 unfolding matrix of $\tilde{\mathcal{X}} \times_3 \hat{\boldsymbol{V}}_p^\mathsf{T}$. Similarly, given optimal solution $\hat{\boldsymbol{V}}_t$, the solution for $\boldsymbol{V}_p$ is the Q left singular vectors of the mode-2 unfolding matrix of $\tilde{\mathcal{X}} \times_2 \hat{\boldsymbol{V}}_t^\mathsf{T}$. Using properties of tensor multiplication, the mode-2 unfolding matrix of $\tilde{\mathcal{X}} \times_2 \hat{\boldsymbol{V}}_t^\mathsf{T}$ is simply $\tilde{\mathcal{X}}_{(2)}\left(\boldsymbol{V}_p^{(l)} \otimes \boldsymbol{I}_n\right)$. Thus the spatial Tucker decomposition algorithm can be summarized in the following steps:

Proposed algorithm: The spatial Tucker decomposition

Input $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{N \times T \times P}$, desired output rank $\{K \times M \times Q\}$, Z, and l_{max}

Initialize $l = 1$,

Calculate $\tilde{\boldsymbol{\mathcal{X}}} = \boldsymbol{\mathcal{X}} \times_1 \left(\boldsymbol{Z}(\boldsymbol{Z}^\intercal \boldsymbol{Z})^{-1}\boldsymbol{Z}^\intercal\right)$

$\boldsymbol{V}_t^{(0)} = M$ left singular vectors of $\tilde{\boldsymbol{\mathcal{X}}}_{(2)}$,

$\boldsymbol{V}_p^{(0)} = Q$ left singular vectors of $\tilde{\boldsymbol{\mathcal{X}}}_{(3)}$,

while not converged or $l < l_{max}$ **do**

$\boldsymbol{V}_t^{(l+1)} \leftarrow M$ left singular vectors of $\tilde{\boldsymbol{\mathcal{X}}}_{(2)} \left(\boldsymbol{V}_p^{(l)} \otimes \boldsymbol{I}_n\right)$

$\boldsymbol{V}_p^{(l+1)} \leftarrow Q$ left singular vectors of $\tilde{\boldsymbol{\mathcal{X}}}_{(3)} \left(\boldsymbol{V}_t^{(l)} \otimes \boldsymbol{I}_n\right)$

$l \leftarrow l + 1$

end while

Output $\widehat{\boldsymbol{V}}_t = \boldsymbol{V}_t^{(l)}$, $\widehat{\boldsymbol{V}}_p = \boldsymbol{V}_p^{(l)}$, and tensor with reduced dimensions in temporal and pollutant space $\hat{\boldsymbol{\mathcal{H}}} = \hat{\boldsymbol{\mathcal{G}}} \times \boldsymbol{Z} = \tilde{\boldsymbol{\mathcal{X}}} \times_2 \widehat{\boldsymbol{V}}_t^\intercal \times_3 \widehat{\boldsymbol{V}}_p^\intercal$

C.3 Setups for toy simulations

To demonstrate the merit of our proposed spatial algorithm, we simulate spatio-temporal multi-pollutant exposure surfaces with $P = 15$. We first generate a core tensor $\boldsymbol{\mathcal{H}} \in \mathbb{R}^{N \times M \times Q}$ on the 100×100 grid ($N = 10,000$), with $M = 1$ and $Q = 3$, such that

$$
\begin{aligned}
&\boldsymbol{h}_{::j} \sim \mathcal{N}\left(\boldsymbol{R}_j \boldsymbol{b}_j, \boldsymbol{S}_j\right), \quad \text{where } j = 1, 2, 3, \\
&\boldsymbol{R}_1 = \begin{bmatrix} \boldsymbol{r}_{1o} & \boldsymbol{r}_{1u} \end{bmatrix}, \quad \text{where } \boldsymbol{r}_{1o}, \boldsymbol{r}_{1u} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{1}), \quad \boldsymbol{b}_1^{\mathsf{T}} = \begin{bmatrix} 5 & 1 \end{bmatrix}, \\
&\boldsymbol{R}_2 = \begin{bmatrix} \boldsymbol{r}_{2o} & \boldsymbol{r}_{2u} \end{bmatrix}, \quad \text{where } \boldsymbol{r}_{2o}, \boldsymbol{r}_{2u} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{1}), \quad \boldsymbol{b}_2^{\mathsf{T}} = \begin{bmatrix} 5 & 1 \end{bmatrix}, \\
&\boldsymbol{R}_3 = \begin{bmatrix} \boldsymbol{r}_{3u} \end{bmatrix}, \quad \text{where } \boldsymbol{r}_{3u} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{1}), \quad \boldsymbol{b}_3 = 1.
\end{aligned}
$$

Similar to previous chapters, $\boldsymbol{r}_{jo}$'s are GIS covariates observed for the model, while $\boldsymbol{r}_{ju}$'s are unobserved covariates, and used primarily to generate the scores. That is, only $\boldsymbol{R} = \begin{bmatrix} \boldsymbol{r}_{1o} & \boldsymbol{r}_{2o} \end{bmatrix}$ is used in the universal kriging model for spatial prediction. Here $\boldsymbol{S}_1$ has exponential structure with no nugget effect, partial sill of 7.5, and range of 50. Meanwhile, $\boldsymbol{S}_2 = 5\boldsymbol{I}_N$ and $\boldsymbol{S}_3 = 2\boldsymbol{I}_N$. This setup is created so that $\boldsymbol{h}_{::1}$ is the most spatially predictable, $\boldsymbol{h}_{::2}$ is moderately predictable in space, and $\boldsymbol{h}_{::3}$ is not spatially predictable.

We then create two scenarios in which we scale the variance of $\boldsymbol{h}_{::j}$'s differently,

$$
\begin{aligned}
&\text{Scenario 1: } Var(\boldsymbol{h}_{::1}) = 10, Var(\boldsymbol{h}_{::2}) = 7.5, Var(\boldsymbol{h}_{::3}) = 5, \\
&\text{Scenario 2: } Var(\boldsymbol{h}_{::1}) = 7.5, Var(\boldsymbol{h}_{::2}) = 5, Var(\boldsymbol{h}_{::3}) = 10.
\end{aligned}
$$

In both scenarios, the spatio-temporal data are generated as

$$
\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{H}} \times_2 \boldsymbol{V}_t \times_3 \boldsymbol{V}_p + \boldsymbol{\mathcal{E}}, \quad \text{where } E_{ijk} \sim \mathcal{N}(0, 1)
$$

with the following loadings

$$\boldsymbol{V}_t = \left[\boldsymbol{v}_{t1}\right] \in \mathbb{R}^{T\times M}, \text{ where } \boldsymbol{v}_{t1} = \frac{\breve{\boldsymbol{v}}_{t1}}{\left\|\breve{\boldsymbol{v}}_{t1}\right\|_2} \text{ and } \breve{\boldsymbol{v}}_{t1} = \{\sin(t)\} \text{ for } t = 1, ..., 10$$

$$\boldsymbol{V}_p = \left[\boldsymbol{v}_{p1} \quad \boldsymbol{v}_{p2} \quad \boldsymbol{v}_{p3}\right] \in \mathbb{R}^{P\times Q}, \text{ where } \boldsymbol{v}_{pj} = \frac{\breve{\boldsymbol{v}}_{pj}}{\left\|\breve{\boldsymbol{v}}_{pj}\right\|_2}, \text{ for } j = 1, 2, 3$$

$$\breve{\boldsymbol{v}}_{p1}^{\mathsf{T}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\breve{\boldsymbol{v}}_{p2}^{\mathsf{T}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\breve{\boldsymbol{v}}_{p3}^{\mathsf{T}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

In actual computation with `python`, the final tensor $\boldsymbol{\mathcal{X}}$ is unfolded in mode-1 by calculating

$$\boldsymbol{X}_{(1)} = \mathbf{H}_{(1)} \left(\boldsymbol{V}_p^{\mathsf{T}} \otimes \boldsymbol{V}_t^{\mathsf{T}}\right)$$

C.4 Sensitivity analysis with different combinations of (M, Q)

In Chapter 4, we show the results of our data analysis with an arbitrary combination ($M = 3, Q = 3$). As a sensitivity analysis, in this section, we use other combinations of M and Q. Figure C.1 shows the overall prediction R^2 from cross-validation for 2010 biweekly data with various combinations of (M, Q), each from 1 to 5. For example, if we assume that there are four pollutant PCs underlying the original data tensor, the cross-validation prediction R^2 are shown on the fourth row of the multi-panel plot. The performance varies based on the assumption of temporal trends (M) in the data. Overall, for this dataset, the choice of M does not have a strong implication on the predictive performance of both methods, except perhaps for later pollutant PCs.

Figures C.2 and C.3 show the distributions of differences in temporal R^2 and MSE values for $(M = 2, Q = 5)$ and $(M = 5, Q = 5)$, respectively. We observe differences with greater magnitudes in later pollutant PCs, such as pollutant PC4 and PC5.

Figures C.4, C.5 and C.6 give the corresponding cross-validation results for the 2011 biweekly results.

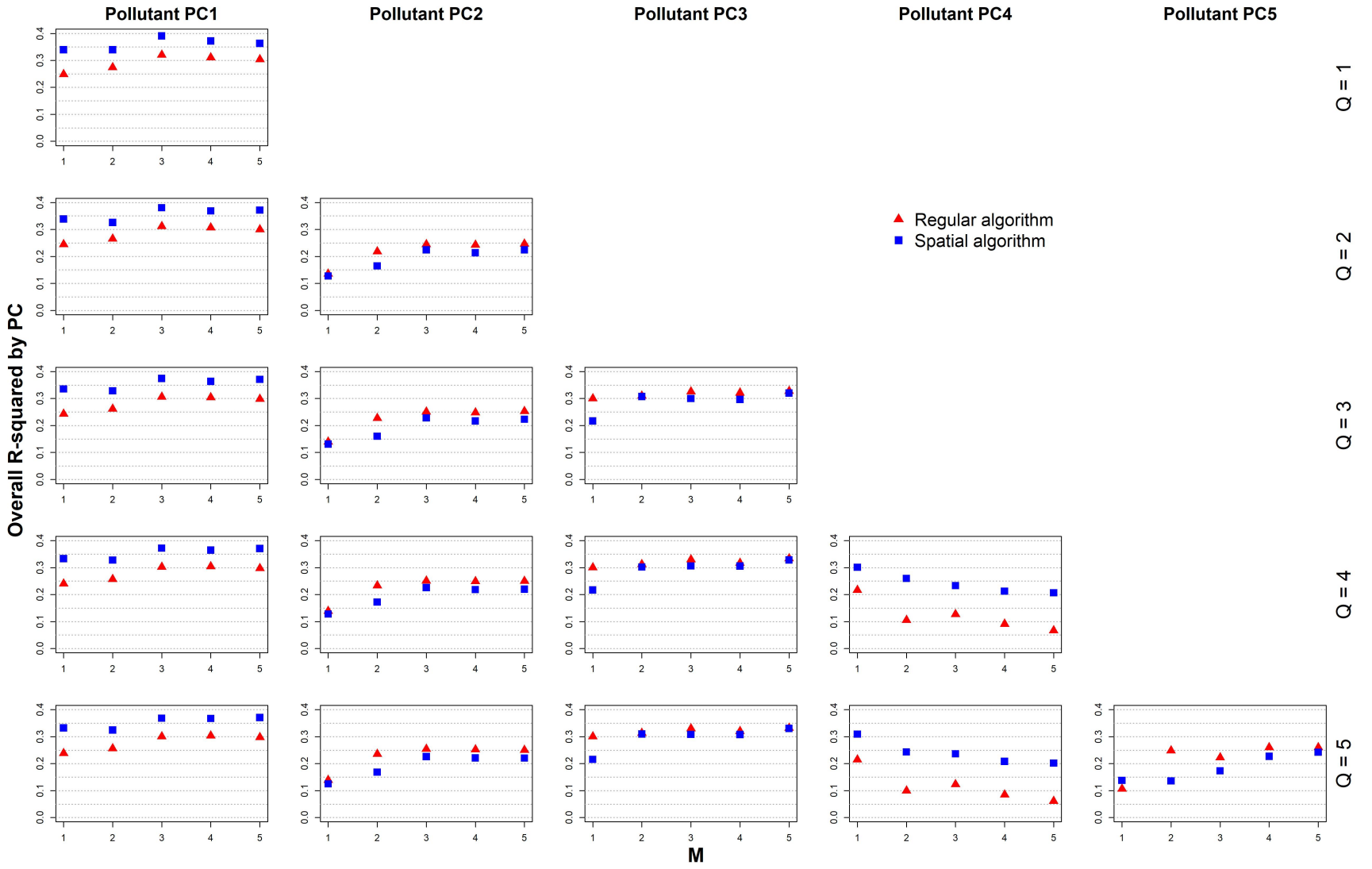

Figure C.1: Cross-validation prediction R^2 by pollutant PC for 2010 data with various combinations of M (the x-axis) and Q (rows of plots).

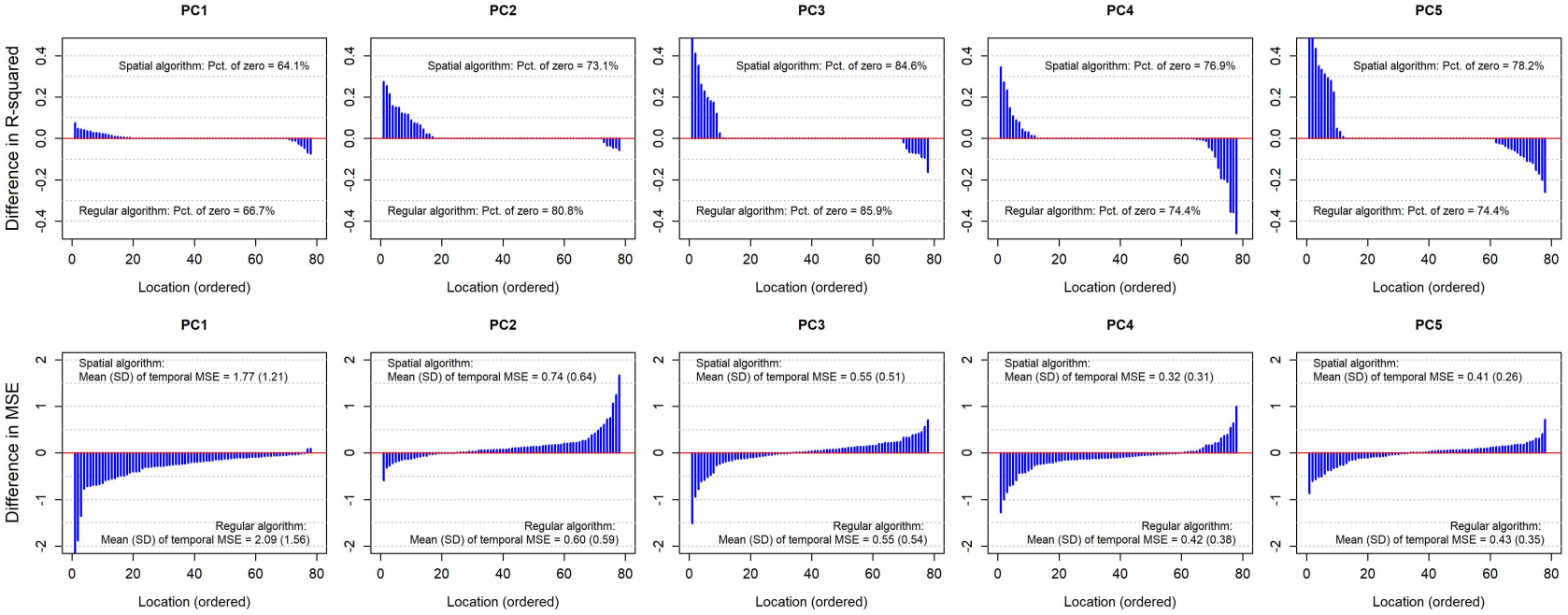

Figure C.2: Differences in temporal R^2 and MSE for 2010 biweekly data using the regular and spatial Tucker algorithms with $(M = 2, Q = 5)$. Positive values of difference in temporal R^2 and negative values of difference in temporal MSE indicate that the predictive performance of the spatial Tucker algorithm is, respectively, better and worse than the regular version. The locations are ordered in terms of temporal values.

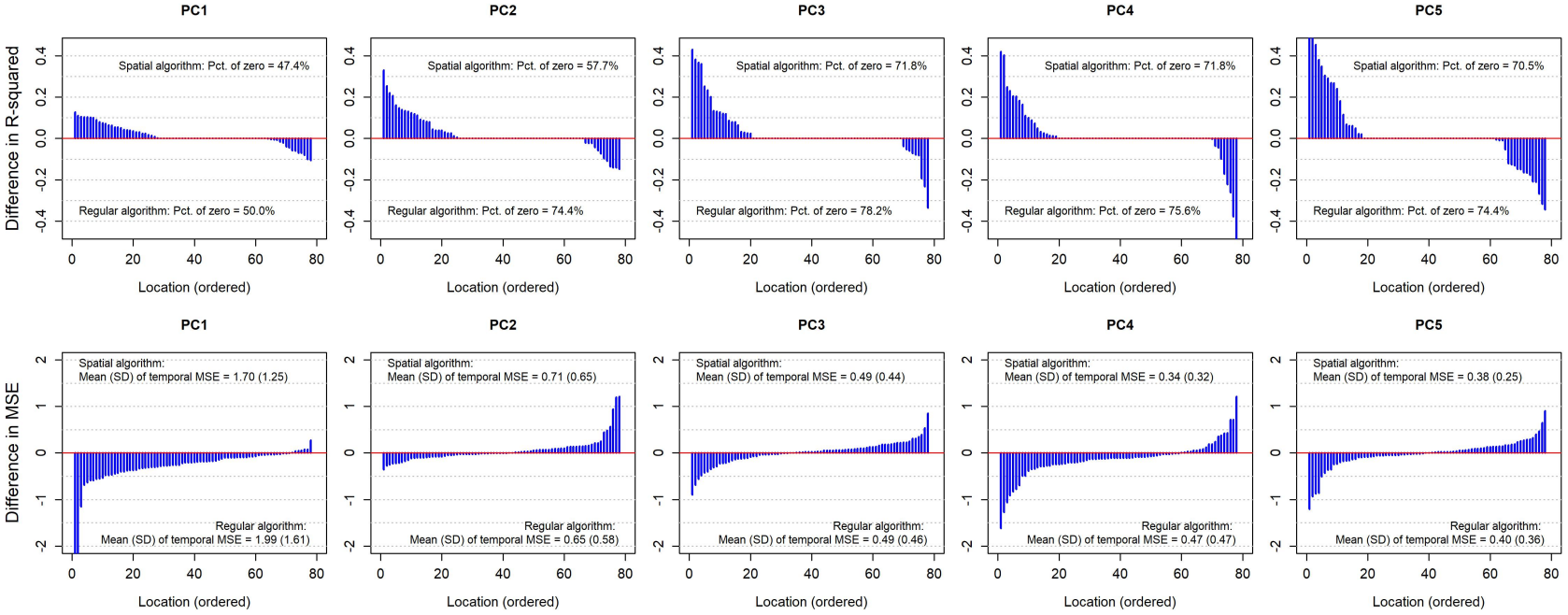

Figure C.3: Differences in temporal R^2 and MSE for 2010 biweekly data using the regular and spatial Tucker algorithms with $(M = 5, Q = 5)$. Positive values of difference in temporal R^2 and negative values of difference in temporal MSE indicate that the predictive performance of the spatial Tucker algorithm is, respectively, better and worse than the regular version. The locations are ordered in terms of temporal values.

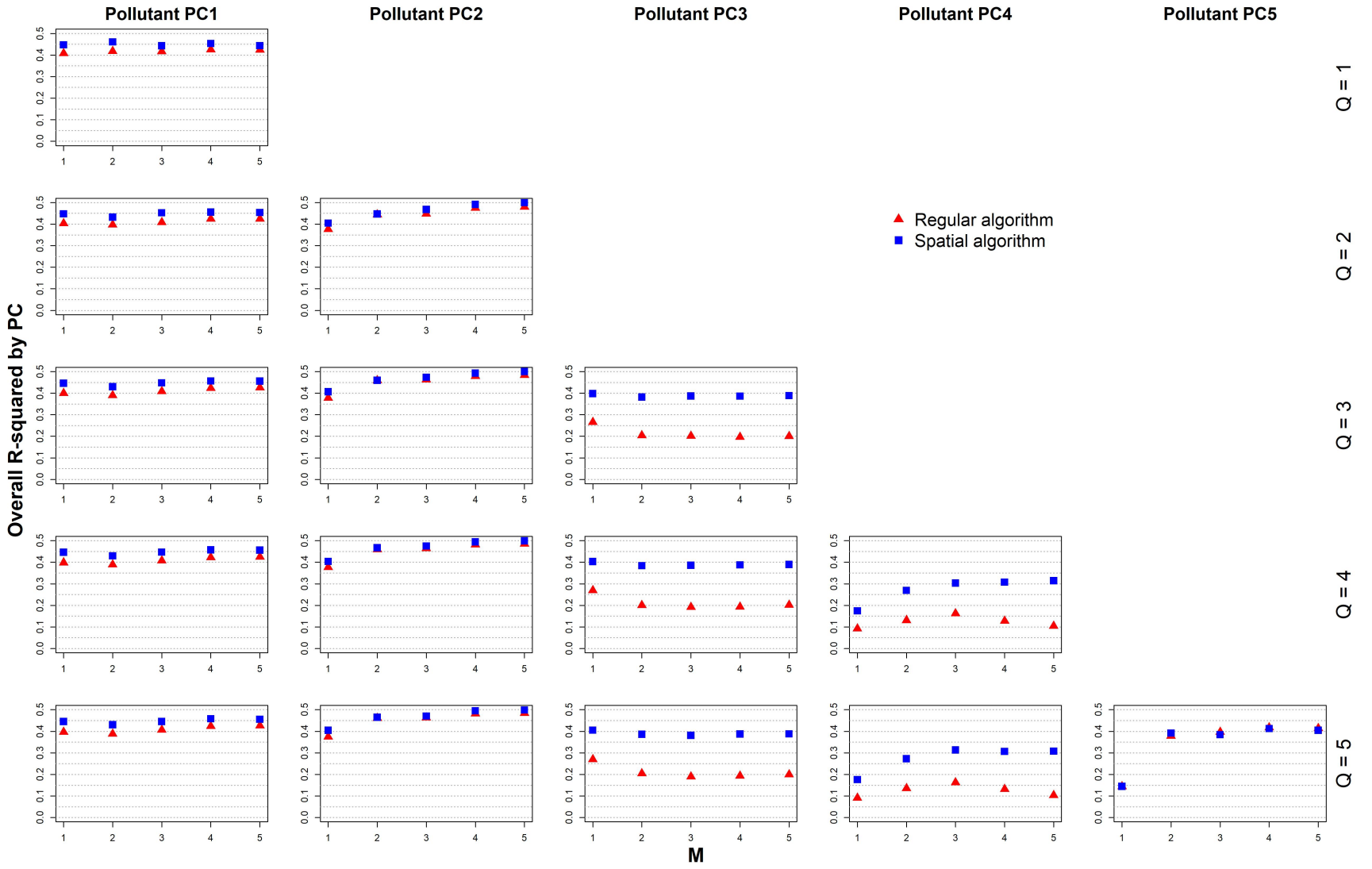

Figure C.4: Cross-validation prediction R^2 by pollutant PC for 2011 data with various combinations of M (the x-axis) and Q (rows of plots).

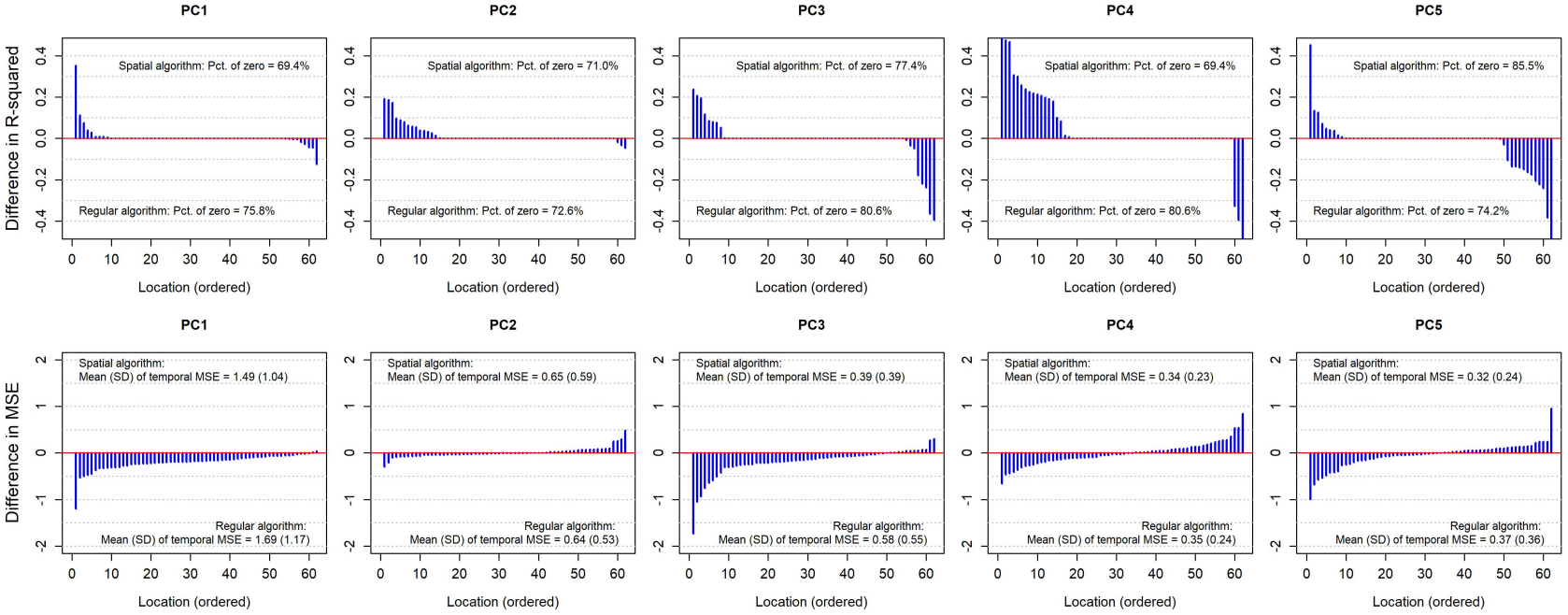

Figure C.5: Differences in temporal R^2 and MSE for 2010 biweekly data using the regular and spatial Tucker algorithms with $(M = 2, Q = 5)$. Positive values of difference in temporal R^2 and negative values of difference in temporal MSE indicate that the predictive performance of the spatial Tucker algorithm is, respectively, better and worse than the regular version. The locations are ordered in terms of temporal values.

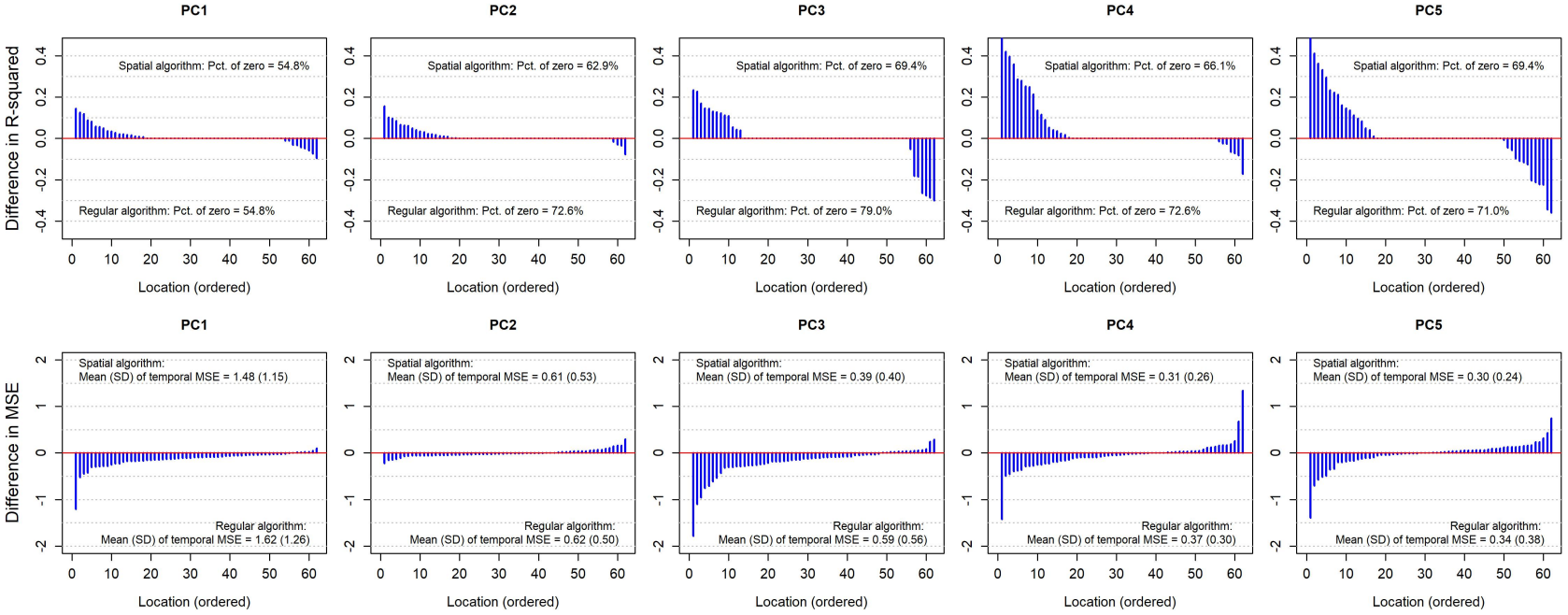

Figure C.6: Differences in temporal R^2 and MSE for 2010 biweekly data using the regular and spatial Tucker algorithms with $(M = 5, Q = 5)$. Positive values of difference in temporal R^2 and negative values of difference in temporal MSE indicate that the predictive performance of the spatial Tucker algorithm is, respectively, better and worse than the regular version. The locations are ordered in terms of temporal values.