Université catholique de Louvain
Institute of Statistics, Biostatistics and Actuarial Sciences

Statistical and Machine Learning Methods for Identifying Clusters of Variables

—

With Applications in *Omics*, Ecology and Psychology

Rebecca Marion

A thesis submitted to the
Université catholique de Louvain
in partial fulfillment of the
requirements for the degree of
Doctor of Sciences

Thesis Committee:

Prof. **Rainer von Sachs**	*Université catholique de Louvain*	Advisor
Prof. **Bernadette Govaerts**	*Université catholique de Louvain*	Advisor
Prof. **Benoît Frénay**	*Université de Namur*	
Prof. **Johannes Lederer**	*Ruhr-Universität Bochum*	
Prof. **Catherine Legrand**	*Université catholique de Louvain*	President
Prof. **Michel Verleysen**	*Université catholique de Louvain*	Secretary

September 2021

All we have to do is decide what to do with the time that is given to us.

GANDALF

Abstract

In many fields, researchers are confronted by datasets whose variables demonstrate grouping patterns. For example, in transcriptomics data, where the variables are gene expression levels, certain groups of genes are involved in the same biological processes, so their expression levels are highly correlated. For complex diseases, such as cancer or heart disease, entire groups of genes are expected to contribute to the development or progression of disease. Thus, identifying these variable groups, or "clusters," can be instrumental in uncovering the mechanisms of disease and developing targeted treatments.

However, in practice, these variable clusters are not known in advance and must be learned from the data. *Clustering* is a data analysis technique used to assign a set of objects to groups, or *clusters*, where similar objects are assigned to the same cluster and dissimilar objects to different clusters. While most work in the literature has focused on the problem of clustering observations (e.g. patients) given a set of variables (e.g. genes), this thesis proposes several statistical and machine learning methods for the problem of *variable clustering*. The objective of the thesis is to propose methods that can improve data analysis in contexts where the ultimate objective is to predict one or more targets (e.g. disease class) and identify clusters of predictor variables (e.g. genes, metabolites) that are most predictive of the target(s).

We explore three problems related to this theme, drawing on applications from the fields of metabolomics, genomics, ecology and psychology. First, we propose AdaCLV, a variable clustering method for pre-processing high-dimensional metabolomics data such that important clusters of variables can be identified with greater precision. Second, we investigate the added value of integrating the target variable (e.g. disease class) into the variable clustering process. We introduce Weighted SOS-NMF, a method that improves variable clustering and variable selection performance by supervising the clustering of variables with the target before a predictive model is fitted. Finally, we examine the case of supervised variable clus-

tering for data with multiple, orthogonal targets. Inspired by a common research problem in ecology and psychology, we propose BIOT, a method for transforming the dimensions of the target matrix so that they can be accurately predicted by small clusters of predictor variables.

Acknowledgements

First of all, I would like to thank my supervisors, Rainer von Sachs and Bernadette Govaerts, for their support and guidance over the past five years. Thanks to your supervision and feedback, I have gained valuable problem-solving reflexes and learned how to "sell" the value of my research, as well as present it more clearly. Thank you for all of our great interactions over the years, at and outside of work. I truly appreciate your reactivity and your concern for my well being as a researcher and a person.

Next, I would like to thank my husband Adrien, who has been an incredible source of encouragement and emotional support throughout this experience. I am very grateful for all of the times that you patiently listened to me talk about my research, brainstormed with me when I got stuck and helped me get back on track when my motivation dwindled. Thank you FE, there is no way I could have completed this thesis without you!

My family and friends were another source of support during the past few years. Thank you Mom and Dad for being with me from a distance via our weekly calls. Thanks also to my brother Joe, whose help with statistics during my Master's program helped me gain the prerequisite skills for taking on this thesis. Finally, I would like to thank my friends LP and Sam, who were always there for me when I needed them and gave excellent advice when times got tough.

In terms of scientific guidance, the members of my thesis jury has given me invaluable feedback. Thank you to Professors Benoît Frénay, Johannes Lederer, Catherine Legrand and Michel Verleysen for your very interesting comments and questions about my research, which pushed me to truly improve the quality of this manuscript.

Finally, I would like to thank my colleagues at the Institute of Statistics, Biostatistics and Actuarial Sciences (ISBA), who made it a joy to come to work every day. First of all, thanks to my current and former office mates —Anna, Mickaël, Os-

waldo, Sophie Mat. and Stefka. It was a pleasure sharing an office with you, and I will truly miss it —even the pigeons! Thanks to my office neighbor on the third floor, Christian, for all our interesting discussions, in particular the one where you helped me warm up to the idea of doing a PhD. I would also like to express my thanks to Maguy, Nadja, Nancy, Sophie Mal. and Tatiana for your support with all things administrative in nature and to the SMCS for the opportunity to gain experience in the world of consulting. Thanks in particular to the members of the SMCS team (Alain, Aurélie, Benjamin C., Catherine, Céline, Nathalie Le., Lieven, Séverine and Vincent B.) for your coaching on consulting projects and for contributing to an excellent working environment. Lastly, thanks to the other doctoral students and post-docs, past and present, who I have had the pleasure of interacting with over the last five years, including Alexandre, Antoine B., Antoine S., Benjamin D., Charles-Guy, Cheikh, Chikêola, Ensiyeh, Emmanuel, Florian, Gilles, Hélène, Hortense, Hugues, John-John, Joris, Keivan, Maïlis, Manon, Michel, Morine, Nathalie Lu., Nathan, Peter, Sandra, Vanessa and Vincent P.

A Note on Vocabulary

Vocabulary

In the machine learning and statistics literature, a variety of words are used to refer to the different elements involved in a data analysis problem. For the sake of clarity, this thesis uses the following vocabulary:

- **observation**: a unit sampled from a population, also known as a *sample*, *individual* or *instance*
- **variable**: a characteristic measured across observations, also known as a *feature*, *factor* or *covariate*
- **target**: a variable that one seeks to predict, also known as a *response* or *dependent variable*
- **predictor**: a variable that is used to predict the target(s), also known as an *explanatory* or *independent variable*
- **latent variable**: an unobserved variable that is constructed as a function of the observed variables, also known as a *latent factor* or *hidden variable*
- **coefficient**: a multiplicative factor relating a predictor variable to a target, also known as a *parameter* or *weight*

Notation and List of Symbols

Notation

Certain conventions are adopted for mathematical notations. Random variables are indicated with upper-case letters in italics (e.g. X). Matrices are denoted with bold-faced capital letters (e.g. $\mathbf{X}$, $\mathbf{\Theta}$), vectors with bold-faced lower-case letters (e.g. $\mathbf{x}$, $\boldsymbol{\theta}$) and scalars with lower-case letters (e.g. x, θ). Row and column vectors of a matrix are indicated using subscripts. For example, if the columns of $\mathbf{X}$ are indexed by the letter j, then the j^{th} column vector is denoted $\mathbf{x}_j$. If the rows are indexed by the letter i, then the i^{th} row vector is denoted $\mathbf{x}_i$. A single element from a matrix is denoted using a lower-case letter with two subscripts, one for the row index and the other for the column index (e.g. x_{ij}).

A list of symbols used throughout this thesis is presented below. Symbols that are specific to a given chapter are presented at the beginning of the chapter.

List of Symbols

$\mathbf{X}$: $n \times p$ matrix of variables to be clustered

i : index for observations $\in \{1, ..., n\}$

j : index for variables $\in \{1, ..., p\}$

k : index for clusters $\in \{1, ..., K\}$

$\mathcal{C}$: partition of K clusters, i.e. $\mathcal{C} := \{C_1, ..., C_K\}$

C_k : set of indices for the objects (observations or variables) in cluster k

n_k : number of observations in cluster k (observation clustering)

p_k : number of variables in cluster k (variable clustering)

$\mathbf{y}$: $n \times 1$ target vector

$\mathbf{U}$: $n \times K$ matrix of latent variables

$\mathbf{V}$: $p \times K$ matrix of cluster membership degrees

$\mathbf{a}$: $K \times 1$ vector of model coefficients

$\mathbf{b}$: $p \times 1$ vector of model coefficients

$\mathbf{B}$: $p \times K$ matrix of model coefficients

$\mathcal{S}$: number of active variables in the true model

$\mathcal{S}^c$: number of inactive variables in the true model

$\text{mean}(\mathbf{x})$: sample mean of $\mathbf{x}$, i.e. $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$

$\text{var}(\mathbf{x})$: sample variance of $\mathbf{x}$, i.e. $\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2$

$\text{sd}(\mathbf{x})$: sample standard-deviation of $\mathbf{x}$, i.e. $\sqrt{\text{var}(\mathbf{x})}$

$\text{cov}(\mathbf{x}_j, \mathbf{x}_\ell)$: sample covariance between $\mathbf{x}_j$ and $\mathbf{x}_\ell$, i.e. $\frac{1}{n-1}\sum_{i=1}^{n}(x_{ij} - \overline{x}_j)(x_{i\ell} - \overline{x}_\ell)$

$\text{cor}(\mathbf{x}_j, \mathbf{x}_\ell)$: sample correlation between $\mathbf{x}_j$ and $\mathbf{x}_\ell$, i.e. $\frac{\text{cov}(\mathbf{x}_j, \mathbf{x}_\ell)}{\text{sd}(\mathbf{x}_j)\text{sd}(\mathbf{x}_\ell)}$

$\text{Var}(X_j)$: theoretical variance of random variable j, i.e. σ_j^2

$\text{Cov}(X_j, X_\ell)$: theoretical covariance between random variables j and ℓ

$\text{Cor}(X_j, X_\ell)$: theoretical correlation between random variables j and ℓ

$\mathbb{I}[x]$: indicator function returning 1 if x is true and 0 otherwise

$(x)_+$: positive truncation function returning x if $x > 0$ and 0 otherwise

$\text{Soft}(x, \delta)$: soft thresholding function, i.e. $\text{Soft}(x, \delta) = \text{sign}(z)(|z| - \delta)_+$

$\text{diag}(\mathbf{x})$: diagonal matrix where $\mathbf{x}$ makes up the diagonal entries

$\text{Tr}(\mathbf{X})$: trace of a square matrix $\mathbf{X}$, i.e. $\sum_{i=1}^{n} x_{ii}$

$\text{rank}(\mathbf{X})$: rank of the matrix $\mathbf{X}$

$||\mathbf{X}||_F^2$: Frobenius norm of $\mathbf{X}$, i.e. $\sum_{i=1}^{n}\sum_{j=1}^{p} x_{ij}^2$

$||\mathbf{x}||_2$: ℓ_2 norm of $\mathbf{x}$, i.e. $\sqrt{\sum_{i=1}^{n} x_i^2}$

$||\mathbf{x}||_1$: ℓ_1 norm of $\mathbf{x}$, i.e. $\sum_{i=1}^{n}|x_i|$

$||\mathbf{x}||_0$: ℓ_0 norm of $\mathbf{x}$, i.e. $\sum_{i=1}^{n}\mathbb{I}[x_i \neq 0]$

$\text{d}_e(\mathbf{x}_j, \mathbf{x}_\ell)$: Euclidean distance between two vectors, i.e. $\sqrt{\sum_{i=1}^{n}(x_{ij} - x_{i\ell})^2}$

Contents

Abstract **2**

Acknowledgements **4**

A Note on Vocabulary **5**

Notation and List of Symbols **8**

1 Introduction **13**
1.1 Objectives of Variable Clustering 14
1.2 Thesis Objectives and Research Questions 15
1.3 Thesis Outline 16

2 Background **19**
2.1 What is a Variable Cluster? 20
2.1.1 Unsupervised Setting 20
2.1.2 Supervised Setting 24
2.2 Observation vs. Variable Clustering 25
2.2.1 Underlying Cluster Structure 25
2.2.2 Clustering Strategy 30
2.2.3 Common Observation Clustering Methods 32
2.3 Clustering and Dimensionality Reduction 34
2.3.1 Clustering as a Matrix Factorization Problem 34
2.3.2 Related Matrix Factorization Methods 36
2.3.3 Advantages of the Matrix Factorization Framework 38

3 Unsupervised Variable Clustering **39**
3.1 Introduction 40
3.1.1 Problem Definition 40

3.1.2 Biomarker Discovery in Metabolomics . . . 40
3.1.3 Contributions . . . 43
3.1.4 Chapter Outline . . . 44
3.2 Proposed Method: AdaCLV . . . 45
3.2.1 Objective . . . 45
3.2.2 Estimation . . . 48
3.2.3 Algorithm and Hyperparameters . . . 49
3.3 Related Work . . . 51
3.3.1 Clustering around Latent Variables (CLV) . . . 51
3.3.2 CLV $K+1$ . . . 53
3.3.3 CLV Sparse LV . . . 56
3.3.4 Sparse Nonnegative Matrix Factorization (SNMF) . . . 56
3.4 Experiments . . . 58
3.4.1 Data . . . 58
3.4.2 Initialization and Hyperparameter Settings . . . 64
3.4.3 Calculation of Ground-Truth Matrices $\mathbf{U}$ and $\mathbf{V}$ . . . 64
3.4.4 Evaluation Criteria . . . 66
3.4.5 Results . . . 68
3.5 Discussion and Conclusions . . . 78
3.6 Numerical Implementation . . . 79
3.7 Appendix . . . 79
3.7.1 CLV Directional Mode . . . 79
3.7.2 CLV $K+1$ Directional Mode . . . 80
3.7.3 CLV Sparse LV Directional Mode . . . 80
3.7.4 Initialization for CLV Local Mode . . . 80
3.7.5 Data Generation for Semi-Artificial Spectra . . . 82
3.7.6 Sensitivity Analysis for the Semi-Artificial Datasets . . . 83

4 Variable Clustering Supervised by a Univariate Target 87
4.1 Introduction . . . 88
4.1.1 Problem Definition . . . 88
4.1.2 Motivation and Related Methods . . . 89
4.1.3 Contributions . . . 90
4.1.4 Chapter Outline . . . 92
4.2 Proposed Method: VC-PCR . . . 92
4.2.1 Sparse Orthogonal Semi-NMF . . . 92
4.2.2 VC-PCR: Step 1 . . . 95
4.2.3 VC-PCR: Step 2 . . . 96
4.2.4 Algorithm and Hyperparameters . . . 98
4.3 Related Work . . . 100
4.3.1 Nonnegative Matrix Factorization Methods . . . 100
4.3.2 Coefficient-Grouping Methods . . . 105
4.3.3 Ensemble Methods and SRR . . . 108

4.3.4 Two-Step Methods . . . 109
4.4 Experiments . . . 111
4.4.1 Methods Compared . . . 111
4.4.2 Simulated Data . . . 112
4.4.3 Evaluation Criteria . . . 113
4.4.4 Hyperparameter Selection . . . 116
4.4.5 Results . . . 117
4.5 Discussion and Conclusions . . . 118
4.6 Numerical Implementation . . . 120
4.7 Appendix . . . 120
4.7.1 Orthonormal Nonnegative Sparse PCA . . . 120
4.7.2 Cluster Elastic Net Reformulation . . . 122

5 Variable Clustering Supervised by Multiple Targets 123
5.1 Introduction . . . 124
5.1.1 Problem Definition . . . 124
5.1.2 Explaining MDS Embedding Dimensions . . . 130
5.1.3 Contributions . . . 131
5.1.4 Chapter Outline . . . 132
5.2 Proposed Methods . . . 132
5.2.1 Best Interpretable Rotation (BIR) . . . 132
5.2.2 Best Interpretable Orthogonal Transformation (BIOT) . . . 133
5.2.3 Selecting the Hyperparameters λ and K . . . 136
5.2.4 Summary . . . 137
5.3 Related Work . . . 137
5.3.1 Eigenvector Partial Least Squares Regression . . . 138
5.3.2 Sparse Reduced Rank Regression (SRRR) . . . 138
5.3.3 Other Methods . . . 139
5.4 Experiments . . . 140
5.4.1 Evaluation Datasets . . . 140
5.4.2 Experimental Protocol . . . 140
5.4.3 Results and Discussion . . . 142
5.5 Case Study: Applying BIOT to Stereotypes . . . 145
5.6 Conclusion . . . 150
5.7 Numerical Implementation . . . 151

6 Conclusion 153

References 157

CHAPTER 1

Introduction

This thesis proposes several statistical and machine learning methods for the problem of *variable clustering*, with applications in metabolomics, genomics, ecology and psychology. *Clustering* is a data analysis technique used to assign a set of objects to K groups or *clusters*, where similar objects are assigned to the same cluster and dissimilar objects are assigned to different clusters. In medicine, clustering patients into different groups (e.g. disease subtypes) can be instrumental to the development of targeted treatments. For example, migraine is a headache disorder for which response to treatment is highly patient-dependent. Given a set of data describing characteristics of patients with chronic migraines (e.g. migraine frequency, age, sex, blood serum measurements, blood pressure, etc.), one may be interested in splitting these patients into groups so that unique treatments may be developed for each group. These groups can be found by clustering the patients based on their clinical and physiological traits.

Clustering is typically performed on a given input matrix containing measurements of variables (e.g. age, blood pressure, etc.) for a set of observations (e.g. patients). In the case of *hard clustering*, each object (e.g. patient) is assigned to only one of the K groups, whereas for *soft clustering*, each object may be assigned to multiple groups to differing degrees. Most often, clustering is used to identify grouping patterns among observations in a dataset, referred to here as *observation clustering*. This is the case for the example described above, where the observations (i.e. migraine patients) are subdivided into groups based on clinical and physiological variables.

Clustering can also be used to identify grouping patterns among variables, rather than observations. For example, for complex diseases, such as cancer and heart disease, multiple groups of genes (e.g. gene pathways) may contribute to the de-

velopment or progression of disease (Ma, Song, and Huang, 2007). These groups can be identified by clustering the variables (i.e. genes) in gene expression datasets. While less common than observation clustering, *variable clustering* has many useful applications.

1.1 Objectives of Variable Clustering

Variable clustering is mainly used to achieve three objectives: identifying grouping structure among variables, dimensionality reduction and variable selection.

Identifying grouping structure among variables

The main objective of variable clustering is to assign p variables to K homogeneous groups, or *clusters*, given the observed data. This makes it possible to uncover possible underlying structure in the dataset, such as sets of variables that are related as well as the inherent dimensionality of the dataset. While clustering is primarily an exploratory analysis, the groups identified can also be used in downstream analysis. For example, some regression and classification methods, such as Group Lasso (Yuan and Lin, 2006), require the input of variable grouping structure. If this structure is not known *a priori*, then variable clustering may be performed in an attempt to learn the structure from the data.

Dimensionality reduction

A secondary objective of variable clustering is to estimate a summary vector for each cluster, such as a mean or median vector (referred to as "centroids" and "medoids" respectively). These summary vectors, also called *latent variables*, can be assembled in a matrix that represents the most important information from the original dataset in a lower-dimensional format. The advantage of using variable clustering to produce this lower-dimensional matrix is that the construction of the variables in the new matrix remains interpretable with respect to the original variables: each new variable represents information from a distinct set of the original variables. Thus, variable clustering is a highly interpretable *dimensionality reduction* technique.

Variable selection

Some variable clustering methods relax the hard clustering constraint so that some variables may be assigned to none of the clusters. In essence, these methods perform *variable selection*, only assigning variables to clusters if they explain a sufficient amount of variability in the data. This approach is particularly appealing in the context of biomarker studies, which seek to identify biological entities whose fluctuation is predictive of a given health outcome (Bingol, Bruschweiler-Li, Li,

Zhang, Xie, and Brüschweiler, 2016). Methods that simultaneously perform variable selection and clustering make it possible to short-list candidate biomarkers and study the relationships between them (i.e. their grouping structure).

1.2 Thesis Objectives and Research Questions

The objective of this thesis is to propose variable clustering methods that can **improve the analysis of datasets whose variables form groups**. In particular, this thesis focuses on data analysis problems where the ultimate objective is to **predict one or more targets** (e.g. disease class) and **select predictor variables** (e.g. genes, metabolites) that are the most predictive of the target(s). We explore three problems related to this theme, proposing solutions in the form of novel statistical and machine learning methods.

First, we address the problem of **pre-processing high-dimensional data** in order to facilitate variable selection in subsequent analyses (e.g. regression or classification). Variable clustering can be particularly useful in this setting because it can reduce the dimensionality of the original dataset and select important variables, while accounting for group structure among variables. The major limitation of existing variable clustering methods is that they tend to overselect variables and to mix important and unimportant variables into the same cluster. **Is it possible to pre-process high-dimensional data via variable clustering such that important clusters of variables may be identified with greater precision?**

Second, we explore the added value of **integrating information from a target variable into the variable clustering process**, in order to simultaneously improve the quality of variable clustering, variable selection and prediction. Several works in the literature have shown the added value of this approach for improving model prediction accuracy, but existing methods tend to overselect variables and/or struggle to recover the true underlying variable clusters. **Is it possible to achieve better variable clustering and variable selection performance by supervising the clustering of variables with the target before fitting a prediction model?**

Finally, we examine the case of **supervised variable clustering for data with multiple, orthogonal targets**. This problem is motivated by an application in ecology and psychology, where a matrix of predictor variables is used to explain a low-dimensional representation, or *embedding*, of another matrix using sparse regression. In contrast to the second research problem, where the target of interest has a pre-defined meaning (e.g. disease class), the meaning of the targets in this setting —i.e. the embedding dimensions —is unknown. Selecting clusters of predictor variables for each dimension via sparse regression makes it possible to assign meaning to these dimensions. However, it may be beneficial to first transform

the embedding, as certain orthogonal transformations of the embedding can improve the prediction, variable selection and variable clustering performance of the regression model, without changing the embedding in any meaningful way. **Is it possible to find an orthogonal transformation of the target matrix such that its dimensions can be accurately predicted by small and distinct clusters of predictor variables?**

1.3 Thesis Outline

In addition to providing a background on the problem of variable clustering, this thesis introduces several variable clustering methods, which were created, implemented and evaluated as a part of the thesis project. These methods fall into two categories: the *unsupervised* methods cluster variables based on information present in the variables themselves, without the aid of a target, and the *supervised* methods use one or more target variables to guide the clustering of variables. Two unsupervised variable clustering methods (AdaCLV and SOS-NMF) are proposed here, as well as three supervised approaches (VC-PCR, BIR and BIOT). The thesis is structured as follows:

- Background on Variable Clustering
  - Chapter 2 presents background information on variable clustering, including its relationship to observation clustering and matrix factorization, as well as definitions of the notion of *variable clusters*.
- Unsupervised Identification of Variable Clusters
  - Chapter 3 presents Adaptive Clustering around Latent Variables (AdaCLV), an unsupervised method for finding variable clusters in spectroscopic data. This method was first introduced in

    Rebecca Marion, Bernadette Govaerts, and Rainer von Sachs (2020). "AdaCLV for Interpretable Variable Clustering and Dimensionality Reduction of Spectroscopic Data". In: **Chemometrics and Intelligent Laboratory Systems** 206, p. 104169.

- Supervised Identification of Variable Clusters
  - Chapter 4 presents Variable Cluster Principal Component Regression (VC-PCR), a prediction method that, given a single target variable, finds clusters of predictive variables. This method depends on an unsupervised variable clustering method, Sparse Orthogonal Semi-Nonnegative Matrix Factorization (SOS-NMF), which is also presented here. These methods will be introduced in

> Rebecca Marion, Johannes Lederer, Bernadette Govaerts, and Rainer von Sachs (2021). "VC-PCR: A Prediction Method based on Supervised Variable Selection and Clustering". In: **To be submitted**.

  - Chapter 5 presents two supervised methods for finding clusters of variables given multiple target variables: Best Interpretable Rotation (BIR) and Best Interpretable Orthogonal Transformation (BIOT). These methods were first introduced in

> Adrien Bibal, Rebecca Marion, and Benoît Frénay (2018). "Finding the Most Interpretable MDS Rotation for Sparse Linear Models based on External Features". In: **Proceedings of the European Symposium on Artificial Neural Networks**. Bruges, Belgium, pp. 537–542,

> Rebecca Marion, Adrien Bibal, and Benoît Frénay (2019). "BIR: A Method for Selecting the Best Interpretable Multidimensional Scaling Rotation using External Variables". In: **Neurocomputing** 342, pp. 83–96

and

> Adrien Bibal, Rebecca Marion, Rainer von Sachs, and Benoît Frénay (2021). "BIOT: Explaining multidimensional nonlinear MDS embeddings using the Best Interpretable Orthogonal Transformation". In: **Neurocomputing** 453, pp. 109–118.

- Conclusion
  - Chapter 6 summarizes the main contributions of this thesis and provides some directions for future work.

CHAPTER 2

Background

This chapter introduces and summarizes key concepts and methodology related to variable clustering. In Section 2.1, the notion of "variable clusters" is clarified, drawing on various definitions from the literature. In Section 2.2, parallels are drawn between observation clustering and variable clustering. Key differences in terms of underlying cluster structure and clustering methodology are outlined, and several popular methods for observation clustering are described. In Section 2.3, parallels are drawn between clustering and the problem of dimensionality reduction. The relationship between clustering and certain matrix factorization methods is explored.

List of Symbols

$\mathbf{X}$: $n \times p$ matrix of centered and scaled variables to be clustered

$\mathbf{Z}$: $n \times p$ nonnegative matrix of variables to be clustered

$\mathbf{y}$: $n \times 1$ target vector (centered and scaled if continuous)

$\mathbf{Y}$: $n \times d$ target matrix (centered and scaled columns if continuous)

$\mathbf{H}$: $p \times K$ matrix of binary cluster memberships

$\mathbf{V}$: $p \times K$ matrix of cluster membership degrees

$\mathbf{M}$: $n \times K$ matrix of centroids

$\mathbf{U}$: $n \times K$ matrix of latent variables

See page 7 for other symbols used in this chapter.

2.1 What is a Variable Cluster?

The true variable clusters in a given datasets are generally unknown. As a result, there is little consensus about how to define them. In the literature, different definitions are used to characterize "variable clusters," depending on the analysis or dataset in question. This section summarizes these different definitions.

Let $\mathbf{X}$ $(n \times p)$ be a matrix containing p variables measured for n observations. Unless otherwise stated, the columns of $\mathbf{X}$ are mean-centered and have a sample variance of 1, i.e. $\text{mean}(\mathbf{x}_j) = 0$, $\text{var}(\mathbf{x}_j) = 1$ and $\mathbf{x}_j^\top \mathbf{x}_j = n-1, \forall j$. We assume that each observation i in $\mathbf{X}$ is drawn from the set of random variables $X_1, ..., X_p$. These random variables can be partitioned into a set of clusters $\mathcal{C} := \{C_1, ..., C_K\}$, where C_k is the set of indices for the variables that belong to cluster k. The objective of variable clustering is to find a partition of the random variables $X_1, ..., X_p$ based on the observed data.

Variable clustering is performed in two main settings, which are represented in the diagram in Figure 2.1. In the diagram, each variable (i.e. column) in the observed matrix $\mathbf{X}$ is assigned to a cluster, where variables in the same cluster share the same color. In the *unsupervised* setting (Figure 2.1(a)), the random variables $X_1, ..., X_p$ are clustered based only on information expressed in $\mathbf{X}$. In the *supervised* setting (Figure 2.1(b)), a target vector $\mathbf{y}$ $(n \times 1)$ or matrix $\mathbf{Y}$ $(n \times d)$ is used to guide the clustering of variables.[1] The definitions of variable clusters for these two settings are generally different.

2.1.1 Unsupervised Setting

In an unsupervised setting, most works in the literature assume that clusters are defined by the "relatedness" of pairs of variables. If two variables j and ℓ are weakly related or entirely unrelated (e.g. a theoretical correlation $\text{Cor}(X_j, X_\ell) \approx 0$), they belong to different clusters, whereas "highly related" variables belong to the same cluster (e.g. high (absolute) correlation). The degree of relatedness between two variables is often quantified via their correlation or covariance (Dhillon, Marcotte, and Roshan, 2003; Vigneau and Qannari, 2003; Vigneau, 2016; Bunea, Giraud, Luo, Royer, Verzelen, et al., 2020).

Some methods for variable clustering in the literature assume that negatively correlated variables belong to different clusters, even if their absolute correlation is large. For example, Bunea, Giraud, Luo, Royer, Verzelen, et al. (2020) assume that the random variables X_j within a given cluster C_k are a function of a common,

[1] In this setting, this problem is sometimes referred to as *supervised clustering*. For French speakers, this terminology could create some confusion, as the term clustering translates to "unsupervised classification" (*classification non supervisée* in French). As a result, *supervised clustering* becomes "supervised unsupervised classification" in French. To be clear, supervised clustering is not a classification problem (the objective is not to assign the observed targets to classes).

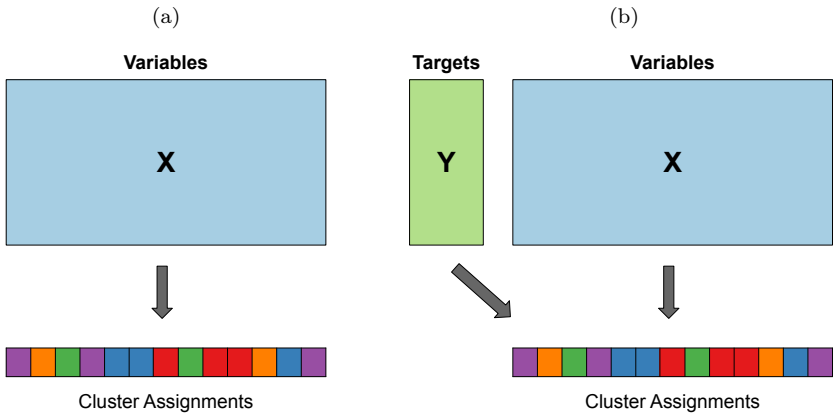


Figure 2.1: **Unsupervised (a) and supervised (b) settings**. Each variable (i.e. column) in the observed matrix $\mathbf{X}$ is assigned to a cluster, where variables in the same cluster share the same color. In the unsupervised setting, no target variables are used to guide the clustering of the variables in $\mathbf{X}$, whereas one or more targets are used in the supervised setting.

random latent variable U_k. Let $X_j \in C_k$ and let E_j be a random error, where $E_j \overset{\text{iid}}{\sim} N(0, \sigma_j^2)$. Then, according to the G-Latent Model (Bunea, Giraud, Luo, Royer, Verzelen, et al., 2020),

$$X_j = U_k + E_j, \tag{2.1}$$

where $U_k \perp\!\!\!\perp E_j$.

In this setting, the random variables within a given cluster have a positive theoretical correlation. Let the random variables X_j and X_ℓ belong to the same cluster k. Then, assuming that $E_j \perp\!\!\!\perp E_\ell$, the theoretical covariance of X_j and X_ℓ is equal to the theoretical variance of U_k:

$$\begin{aligned} \text{Cov}(X_j, X_\ell) &= \text{Cov}(U_k, U_k) + \text{Cov}(U_k, E_j) + \text{Cov}(U_k, E_\ell) + \text{Cov}(E_j, E_\ell) \\ &= \text{Var}(U_k), \end{aligned} \tag{2.2}$$

which is necessarily positive.

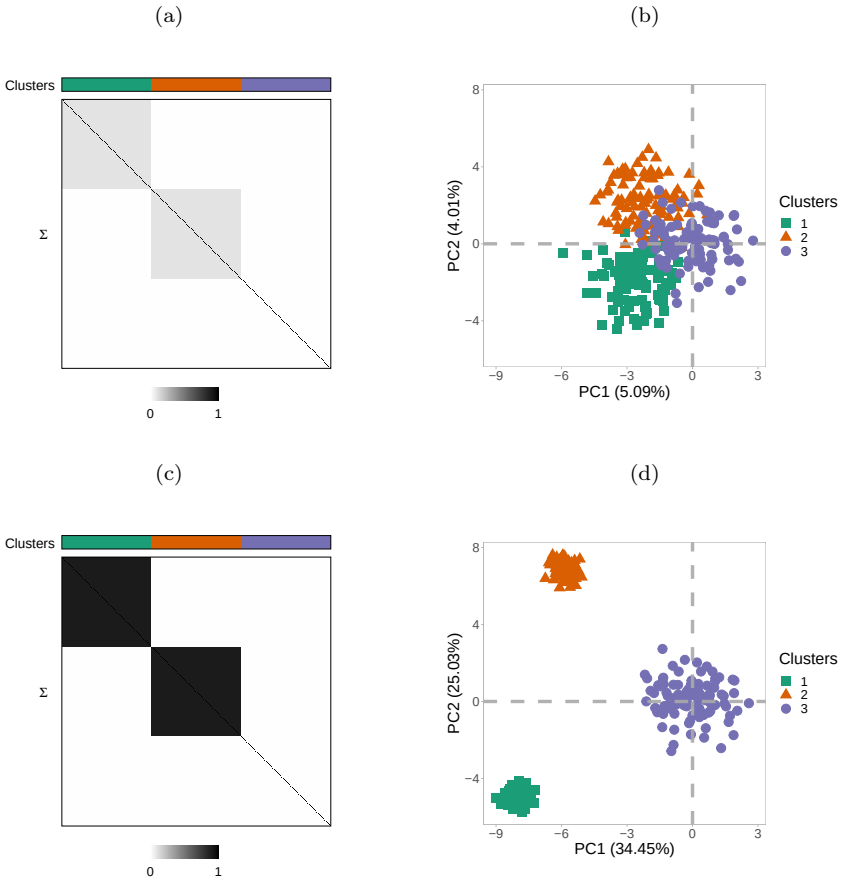


Figure 2.2: **Variable clustering: Effect of within-cluster correlation on cluster separation and density**. Heatmaps of the population covariance matrix Σ (left) and PCA loadings for the first two principal components (right) of simulated iid normal data. The true cluster membership of the variables is indicated above the heatmaps, where each variable along the x-axis of the heatmap is represented with a corresponding colored vertical line in the color bar. Variables depicted in the same color belong to the same cluster. Within-cluster correlation for clusters 1 and 2 is set to 0.1 (top) and 0.9 (bottom).

Their correlation is also positive:

$$\begin{aligned} \text{Cor}(X_j, X_\ell) &= \frac{\text{Cov}(X_j, X_\ell)}{\sqrt{\text{Var}(X_j)}\sqrt{\text{Var}(X_\ell)}} \\ &= \frac{\text{Var}(U_k)}{\sqrt{\text{Var}(U_k) + \text{Var}(E_j)}\sqrt{\text{Var}(U_k) + \text{Var}(E_\ell)}}. \end{aligned} \tag{2.3}$$

Other authors, such as Dhillon, Marcotte, and Roshan (2003), assume that pairs of variables with large absolute correlation are related —and thus belong to the same cluster —even if the sign of the correlation is negative (Dhillon, Marcotte, and Roshan, 2003). Both perspectives (positive and absolute pairwise correlation) are used to define variable clusters in Vigneau and Qannari (2003) and Vigneau (2016), where each perspective is assumed to be appropriate for different data settings.

In the end, the assumptions made about the true underlying variable clusters in a given dataset depend on the context. For consumer preference data, brain imaging data and data representing mixtures (such as Nuclear Magnetic Resonance (NMR) spectra), the variables in a given cluster are generally assumed to have large positive correlation (see Chapter 3 and Vigneau and Qannari (2003), Bunea, Giraud, Luo, Royer, Verzelen, et al. (2020), and Marion, Govaerts, and von Sachs (2020)), whereas for gene expression data, for example, variables in the same cluster are assumed to have large absolute correlation (Dhillon, Marcotte, and Roshan, 2003).

For some authors, some unrelated variables (i.e. uncorrelated variables) may belong to the same underlying cluster. For example, for Vigneau (2016) and Marion, Govaerts, and von Sachs (2020), variables that are uncorrelated with all other variables, referred to as "atypical" or "noise" variables, may be set aside or assigned to a single noise cluster. Somewhat surprisingly, this noise cluster, which is composed of uncorrelated variables, behaves similarly to the other clusters, which are made up of correlated variables. This is demonstrated in Figure 2.2, which depicts PCA loadings plots (right-hand column) for two toy datasets. Both datasets contain three clusters: two clusters (green and orange) whose variables have a within-cluster pairwise correlation of 0.1 (top) or 0.9 (bottom), and a third cluster of variables (purple) that are uncorrelated with all variables. Despite the fact that the variables in the third cluster are uncorrelated, they nonetheless form a cluster. This suggests that positive or negative pairwise correlation between variables is not actually required for them to group together in space. Rather, variables group together if they share the same or a similar covariance pattern with all other variables.

Let Σ be the covariance matrix for $X_1, ..., X_p$. Then, two variables j and ℓ are in the same cluster if the corresponding rows in Σ (excluding the diagonal elements σ_{jj} and $\sigma_{\ell\ell}$) are similar, i.e.

$$(\sigma_{j1}, ..., \sigma_{jj-1}, \sigma_{jj+1}, ..., \sigma_{j\ell-1}, \sigma_{j\ell+1}, ..., \sigma_{jp})^\top \approx (\sigma_{\ell 1}, ..., \sigma_{\ell j-1}, \sigma_{\ell j+1}, ..., \sigma_{\ell \ell-1}, \sigma_{\ell \ell+1}, ..., \sigma_{\ell p})^\top. \tag{2.4}$$

2.1.2 Supervised Setting

In a supervised setting, the clustering of variables is guided by a target vector $\mathbf{y}$ $(n \times 1)$ or matrix $\mathbf{Y}$ $(n \times d)$. Each observation i in the vector $\mathbf{y}$ is drawn from the random variable Y, and each vector $\mathbf{y}_i$ $(d \times 1)$ from $\mathbf{Y}$ is drawn from the set of random variables $Y_1, ..., Y_d$.

For a univariate target in the linear regression framework, we assume that $\mathbf{y}$ can be modeled as a linear function of the matrix $\mathbf{X}$ $(n \times p)$ and a vector of true, unknown regression coefficients $\boldsymbol{b}$ $(p \times 1)$:

$$\mathbf{y} = \mathbf{X}\boldsymbol{b} + \boldsymbol{\varepsilon}, \tag{2.5}$$

where $\boldsymbol{\varepsilon}$ is a vector of errors with elements $\varepsilon_i \overset{\text{iid}}{\sim} N(0, \sigma_\varepsilon^2)$.

Several definitions of variable clusters are used in works from the literature, but few of these works evoke these definitions explicitly. This is because variable clusters are often seen as a characteristic of the dataset —potentially detrimental or beneficial to analysis —rather than as entities to be identified and studied. Indeed, the task of explicitly identifying variable clusters is only addressed in a few works (Sharma, Bondell, and Zhang, 2013; Witten, Shojaie, and Zhang, 2014; Yengo, Jacques, and Biernacki, 2014).

Let $\mathcal{S} := \{j \mid b_j \neq 0\}$ be the set of *active variables* in the true linear model, and let $\mathcal{S}^c := \{j \mid b_j = 0\}$ be the set of *inactive variables*. The general consensus in the literature is that the set of all inactive variables, $\mathcal{S}^c$, constitutes one of the variable clusters in $\mathbf{X}$. However, there is little consensus about how to define clusters within the active set, $\mathcal{S}$. For most works in the literature, the underlying variable clusters in $\mathcal{S}$ are defined based on the correlation structure among the random variables $X_1, ..., X_p$, as well as the vector of coefficients $\boldsymbol{b}$, which characterizes the relationship between $\mathbf{X}$ and $\mathbf{y}$. There are three main definitions of variable clusters for variables in the active set.

Definition 2.1.1 (Cluster membership of variables in the active set)**.** *Two active variables j and ℓ are in the same cluster if $|Cor(X_j, X_\ell)|$ is large and $|b_j| \approx |b_\ell|$.*

Definition 2.1.1 is the perspective taken by most works in the literature. For example, the regression and supervised clustering methods proposed by Bondell and Reich (2008), Daye and Jeng (2009), Kim and Xing (2009), Sharma, Bondell,

and Zhang (2013), and Witten, Shojaie, and Zhang (2014) implicitly depend on this definition. In this setting, the signs of the correlations and coefficients are assumed to be unimportant.

Definition 2.1.2 (Cluster membership of variables in the active set)**.** *Two active variables j and ℓ are in the same cluster if $Cor(X_j, X_\ell)$ is large and $b_j \approx b_\ell$.*

Definition 2.1.2 assumes that negatively correlated variables, as well as variables with opposite-signed coefficients, are not in the same clusters. This is the perspective of authors such as Park, Hastie, and Tibshirani (2007).

Definition 2.1.3 (Cluster membership of variables in the active set)**.** *Two active variables j and ℓ are in the same cluster if $b_j \approx b_\ell$.*

Definition 2.1.3 is proposed in Shen and Huang (2010) and implicitly used by Tutz and Ulbricht (2009) and Yengo, Jacques, and Biernacki (2014) to define variable clusters. Although these authors define clusters only according to the coefficient vector $\boldsymbol{b}$, the methods they propose implicitly depend on the correlation between the random variables $X_1, ..., X_p$. For these authors, this correlation structure is seen as a potential challenge for clustering rather than a defining characteristic of the underlying clusters.

2.2 Observation vs. Variable Clustering

For *observation clustering*, the objects to be clustered are the observations i (i.e. the rows $\mathbf{x}_i$ of $\mathbf{X}$), whereas for *variable clustering*, the objects to be clustered are the variables j (i.e. the columns $\mathbf{x}_j$). This section summarizes the main differences between observation and variable clustering with respect to the underlying cluster structure (Section 2.2.1) and the strategy employed to cluster objects (Section 2.2.2). Each subsection first describes an aspect of observation clustering, then explains the parallels with the variable clustering problem.

2.2.1 Underlying Cluster Structure

Cluster structure in a given dataset is defined by several characteristics, including cluster shape, degree of separation, density and the presence of outlying objects.

Cluster Shape

In general, clusters with a spherical shape are easier to identify than clusters with elongated spherical or non-spherical shapes. For the sake of simplicity, consider the case of normally distributed data.

Observation Clustering

For observation clustering, each cluster of observations is often assumed to follow a different multivariate normal distribution. In other words, for a given cluster k, the observation vectors $\mathbf{x}_i \overset{\text{iid}}{\sim} N_p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, where $\boldsymbol{\mu}_k$ $(p \times 1)$ is the mean vector and $\boldsymbol{\Sigma}_k$ $(p \times p)$ is the covariance matrix of cluster k. Clusters of observations are spherical when all variables have the same variance and are mutually uncorrelated (i.e. $\boldsymbol{\Sigma}_k$ is diagonal). When some variables are correlated for a given cluster, or have unequal variances, that cluster will have an elongated shape.

Variable Clustering

For variable clustering, variable clusters are spherical if the observations i are independent and identically distributed (iid) and the variables j within a cluster have the same mean and variance. This is demonstrated in Figure 2.3, which depicts a common setting in real data: some variables have a block-diagonal covariance structure (clusters 1 and 2) and some variables are uncorrelated with all other variables (cluster 3). Figure 2.3(a) shows the case where variables within the same cluster share the same mean and variance, whereas the other plots (Figure 2.3(b)-(e)) represent cases where either the means or variances are different within a cluster. For clusters of uncorrelated variables (e.g. cluster 3 in purple), a spherical shape can be achieved when the variables in the cluster have different variances (Figure 2.3(d)-(e)), but not when they have different means (Figure 2.3(b)-(c)). For clusters of correlated variables, having disparate means and variances results in an elongated cluster shape, where the elongation increases as the disparity increases from low (left) to high (right).

Given this phenomenon, it may be advantageous to normalize variables to zero mean and unit variance before clustering. In fact, centering the variables in $\mathbf{X}$ is a common practice that is often indispensable. However, in some cases, scaling variables to unit variance can be undesirable. For example, when there is a cluster of low-variance, uncorrelated noise variables, it is better to center variables only. This is demonstrated in Figure 2.4, where cluster 3 is made up of low-variance, uncorrelated variables. In the left-hand plot, all variables are centered only, and in the right-hand plot, they are both centered and scaled. Clearly, the different clusters are easiest to distinguish when the variables are centered only (left-hand plot) and not scaled.

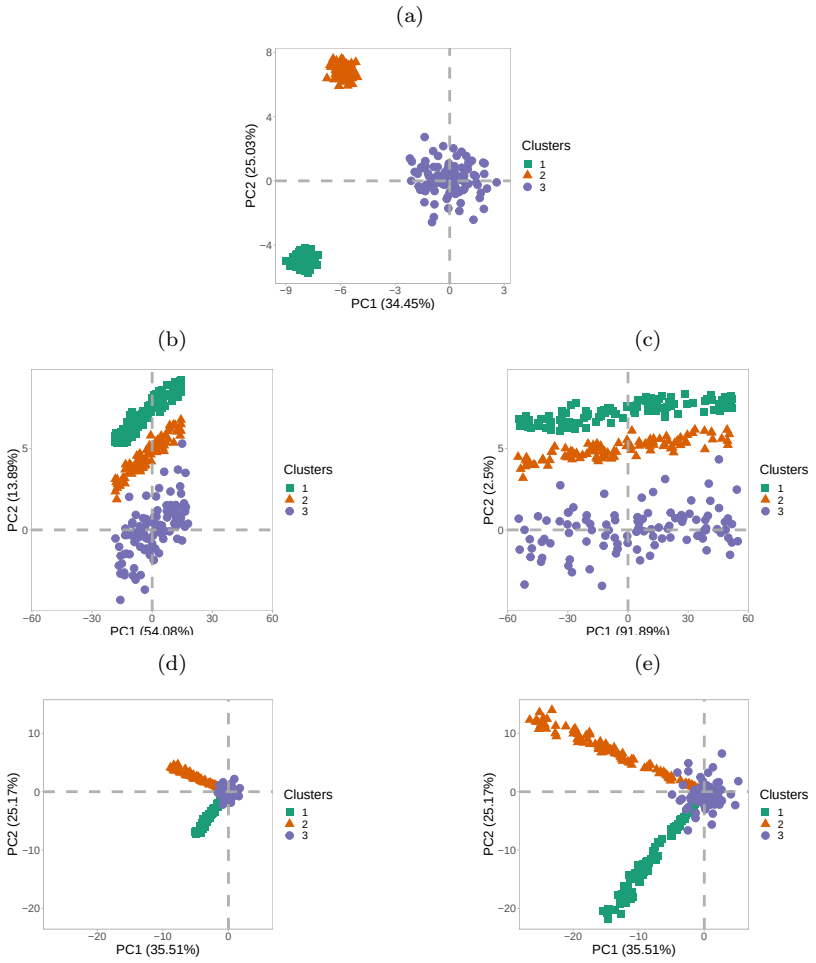


Figure 2.3: **Variable clustering: Effect of variable mean and variance on cluster shape**. Plots of PCA loadings (right) for the first two principal components of iid normal data with three variable clusters (each point is a variable). The variables in clusters 1 and 2 have within-cluster correlations of 0.9 and the variables in cluster 3 are uncorrelated with all other variables. Different settings for within-cluster variable means and variances are shown here: (a) same mean and variance, (b) slightly different means but same variance, (c) highly different means but same variance, (d) slightly different variances but same mean and (e) highly different variances but same mean.

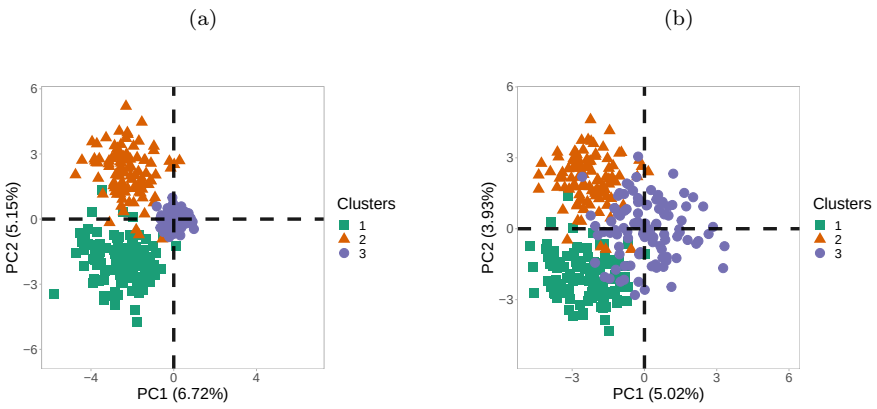


Figure 2.4: **Variable clustering: Effect of normalization**. Plots of PCA loadings for the first two principal components of iid normal data with three variable clusters (each point is a variable). The variables in clusters 1 and 2 have within-cluster correlations of 0.1 and unit variance, and the variables in cluster 3 are uncorrelated with all other variables and have a variance of 0.1. (a) The variables are centered only. (b) The variables are centered and scaled to unit variance.

Cluster Separation and Density

The difficulty of a clustering problem also depends on the degree of separation between clusters and the density of the clusters, with highly compact, well separated clusters being the easiest to identify.

Observation Clustering

For observation clustering with spherical clusters, cluster density depends on within-cluster variance (i.e. the diagonal elements of $\mathbf{\Sigma}_k$). Cluster separation depends on both within-cluster variance and the distance between cluster centers, $\boldsymbol{\mu}_k$.

Variable Clustering

For variable clustering, both cluster separation and density depend on within-cluster correlation. As the correlation of variables within a cluster increases, the compactness of the cluster, as well as its distance from other clusters, increases. This is demonstrated in Figure 2.2, where the within-cluster correlation of clusters 1 and 2 is varied from 0.1 (top plots) to 0.9 (bottom plots). Less cluster overlap and more cluster density is observed in the case where within-cluster correlation is high (bottom plots). Cluster separation also depends on the correlation between variables from different clusters: the greater the correlation, the closer the clusters.

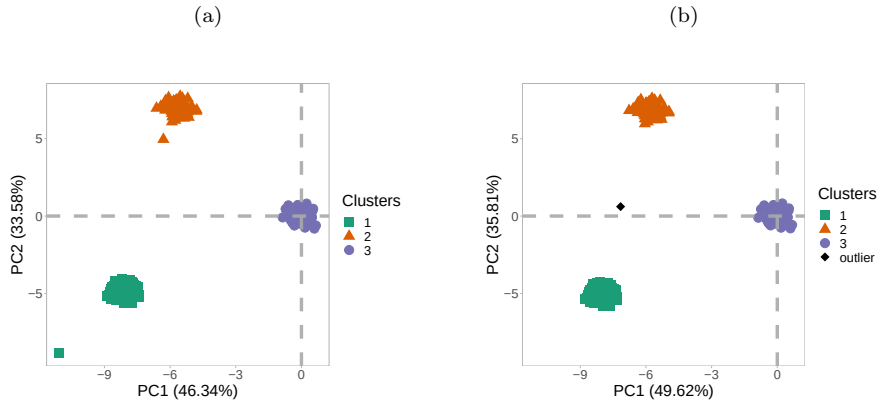


Figure 2.5: **Variable clustering: Outliers**. Plots of PCA loadings for the first two principal components of iid normal data with three main variable clusters (each point is a variable). The variables in clusters 1 and 2 have within-cluster correlations greater than zero, and the variables in cluster 3 are uncorrelated with all other variables. (a) All variables in clusters 1 and 2 have a mutual pairwise correlation of 0.9 except for two outlier variables, which have a lower correlation with all other variables in their cluster. (b) One outlier variable (in black) is correlated with the variables in both clusters 1 and 2.

Outliers

The presence of outliers can make the problem of clustering more challenging.

Observation Clustering

There are two types of outliers in observation clustering problems: (i) observations that belong to one of the clusters (i.e. are generated from one of the cluster distributions) but are positioned in an outlying zone of the cluster and (ii) observations that belong to none of the clusters (i.e. they are not generated from one of the cluster distributions).

Variable Clustering

Both types of outliers can be present in variable clustering problems as well, but they manifest themselves for different reasons. For example, outliers of the first type occur when the variables within a given cluster do not all have the same pairwise correlation. This is demonstrated in the left panel of Figure 2.5. All variables in clusters 1 and 2 have a mutual pairwise correlation of 0.9, with the exception of two outlier variables, which have a lower correlation with all other variables in their cluster. These two variables (one for each cluster) appear in an

outlying zone of their respective clusters. The second type of outlier is present when a given variable is correlated with variables from different clusters. This can be seen in the right panel of Figure 2.5, where a single, outlying variable (colored in black) is correlated with the variables in clusters 1 and 2. For both types of outlier, the outlying variables are correlated with one or more of the other variables.

Noise Cluster

Many datasets contain a set of variables that are uncorrelated with all other variables in the dataset. If the variables in the dataset are mean-centered, these uncorrelated "noise" variables will be grouped together around the origin, forming a "noise cluster." These variables are different from outlying variables because, unlike outlying variables, they are uncorrelated with all variables, and they group together in Euclidean space. Consider the example in Figure 2.6, which depicts a $2 \times p$ column-centered dataset, where p is the number of variables. There are three clusters of variables and two outlying variables. Certain variables are correlated with the other variables in their cluster (clusters 1 and 2 in green and orange), while other variables are uncorrelated with all other variables (cluster 3 in purple). Clearly the outlying variables (in black) do not group together, whereas the uncorrelated "noise" variables (purple cluster) form a cluster around the origin. Such a noise cluster is present whenever a column-centered matrix contains variables that are uncorrelated with all other variables. No equivalent to this noise cluster exists in observation clustering problems.

2.2.2 Clustering Strategy

Observations are generally clustered according to their pairwise (dis-)similarity or their (dis-)similarity with respect to the cluster representatives (e.g. cluster centroids). For most observation clustering methods, dissimilarity is quantified using a distance measure, such as the Euclidean distance. For variable clustering methods, variables are generally clustered according to some measure related to correlation, such as the pairwise Pearson correlation, cosine similarity or covariance.

Nonetheless, cluster analysis based on Pearson correlations is related to cluster analysis based on Euclidean distances. Let $\mathbf{x}_j$ and $\mathbf{x}_\ell$ be two $n \times 1$ vectors, both mean-centered and with a sample variance of 1 (i.e. $\mathbf{x}_j^\top \mathbf{x}_j = n - 1$). The sample estimate for the Pearson correlation between these vectors is given by

$$\text{cor}(\mathbf{x}_j, \mathbf{x}_\ell) = \frac{\mathbf{x}_j^\top \mathbf{x}_\ell}{n-1} = \frac{1}{n-1} \sum_{i=1}^{n} x_{ij} x_{i\ell}. \tag{2.6}$$

The Euclidean distance between these two vectors is

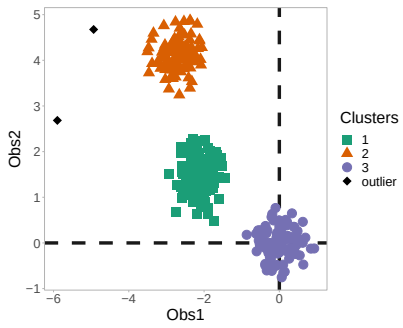


Figure 2.6: **Variable clustering: Uncorrelated variables vs outliers**. Plot of a $2 \times p$ column-centered matrix with p variables. The two axes represent the value of observation 1 and 2 (x- and y-axis) for a given variable. There are three clusters of variables and two outlying variables. Certain variables are correlated with the other variables in their cluster (clusters 1 and 2 in green and orange), while other variables are uncorrelated with all other variables (cluster 3 in purple). Each outlying variable (in black) is correlated with the variables in either the green or the orange cluster, but to a lesser extent than the variables within that cluster. The uncorrelated "noise" variables (purple cluster) form a cluster around the origin.

$$\begin{aligned} \mathrm{d}_e(\mathbf{x}_j, \mathbf{x}_\ell) &= ||\mathbf{x}_j - \mathbf{x}_\ell||_2 \\ &= \sqrt{\mathbf{x}_j^\top \mathbf{x}_j + \mathbf{x}_\ell^\top \mathbf{x}_\ell - 2\mathbf{x}_j^\top \mathbf{x}_\ell} \\ &= \sqrt{(n-1) + (n-1) - 2(n-1)\mathrm{cor}(\mathbf{x}_j, \mathbf{x}_\ell)} \\ &= \sqrt{2(n-1)(1 - \mathrm{cor}(\mathbf{x}_j, \mathbf{x}_\ell))}. \end{aligned} \tag{2.7}$$

Clearly, the Euclidean distance is a monotone function of the Pearson correlation. Thanks to this relationship, it is common practice to use observation clustering methods based on Euclidean distance for the purpose of variable clustering. This is done by centering and scaling the variables in the dataset to unit variance, then applying distance-based observation clustering methods to the transpose of this dataset. For rank-based clustering methods (e.g. hierarchical agglomerative clustering, see Section 2.2.3), using a dissimilarity measure based on Euclidean distances (observation clustering approach) yields the same clusters as when using a measure based on Pearson correlations (variable clustering approach). For other types of clustering methods, however, different results may be achieved for Euclidean distance measures as compared to correlation-based measures.

In some cases, it may be better to cluster variables with respect to pairwise covariance, instead of correlation or distance. Indeed, as discussed in Section 2.2.1, scaling variables to unit variance is not always desirable. However, common distance measures cannot be rewritten as a function of covariance. As a result, distance-based observation clustering methods cannot implicitly cluster according to covariance.

2.2.3 Common Observation Clustering Methods

As mentioned in Section 2.2.2, observation clustering methods are often used to cluster variables, especially methods that use Euclidean distance to characterize the dissimilarity of observations and clusters. Some common methods of this type are described below, although many other methods exist in the literature.

K-means Observation Clustering

K-means observation clustering seeks the partition that minimizes the sum of squared Euclidean distances between observations and the centroid of their assigned cluster. Let the partition $\mathcal{C} := \{C_1, ..., C_K\}$, where $C_k := \{i \mid i \in \text{ cluster } k\}$, and let the matrix of cluster centroids $\mathbf{M} = [\mathbf{m}_1 \ ... \ \mathbf{m}_K]$ $(p \times K)$. The objective function for K-means observation clustering is

K-means Objective Function (Observation Clustering)

$$J(\mathbf{M}, \mathcal{C}) = \sum_{k=1}^{K} \sum_{i \in C_k} ||\mathbf{x}_i - \mathbf{m}_k||_2^2. \tag{2.8}$$

The solution to this problem is found by alternating between optimizing $\mathbf{M}$ for fixed $\mathcal{C}$ and optimizing $\mathcal{C}$ for fixed $\mathbf{M}$. For fixed $\mathcal{C}$, the optimal centroid $\hat{\mathbf{m}}_k$ is the mean of the n_k observations in cluster k:

$$\hat{\mathbf{m}}_k = \frac{1}{n_k} \sum_{i \in C_k} \mathbf{x}_i. \tag{2.9}$$

For fixed $\mathbf{M}$, the optimal cluster $\hat{C}_k$ is the set of all observations for which $\mathbf{m}_k$ is the closest centroid:

$$\hat{C}_k = \{i \mid k = \arg\min_{\ell} ||\mathbf{x}_i - \mathbf{m}_\ell||_2^2\}. \tag{2.10}$$

Hierarchical Agglomerative Clustering

Another popular method for observation clustering is hierarchical agglomerative clustering (HAC). At the first step of the algorithm, each of the observations belongs to a different cluster (i.e. $K = n$). At each successive step, a pair of clusters is merged, generating a hierarchy of partitions. The pair of clusters chosen for merging is the one that minimizes a given dissimilarity-based criterion.

For example, in complete-linkage clustering, the dissimilarity criterion for two clusters C_k and $C_{k'}$ in the maximum distance between observations belonging to different clusters:

Complete-linkage Criterion (Observation Clustering)

$$J(C_k, C_{k'}) = \max_{i \in C_k, \ell \in C_{k'}} \mathrm{d}(\mathbf{x}_i, \mathbf{x}_\ell), \tag{2.11}$$

where d() is a given distance function, such as the Euclidean distance. Single-linkage clustering uses the minimum distance, rather than the maximum distance, as the dissimilarity criterion for two clusters:

Single-linkage Criterion (Observation Clustering)

$$J(C_k, C_{k'}) = \min_{i \in C_k, \ell \in C_{k'}} \mathrm{d}(\mathbf{x}_i, \mathbf{x}_\ell). \tag{2.12}$$

For average-linkage clustering, the dissimilarity criterion is the average distance between observations in the merged cluster $C_k \cup C'_k$:

Average-linkage Criterion (Observation Clustering)

$$J(C_k, C_{k'}) = \frac{1}{n_k \; n_{k'}} \sum_{i \in C_k}^{n_k} \sum_{\ell \in C_{k'}}^{n_{k'}} \mathrm{d}(\mathbf{x}_i, \mathbf{x}_\ell). \tag{2.13}$$

Ward's method (Ward Jr, 1963) quantifies the cost of merging two clusters using a distance-based criterion. Let $\mathbf{m}_{kk'}$ be the mean of the observations in the merged cluster $C_{kk'} = C_k \cup C_{k'}$. The cost of merging C_k and $C_{k'}$ is the increase in sum of squared errors in the resulting cluster:

Ward's Criterion (Observation Clustering)

$$\begin{aligned} J(C_k, C_{k'}) &= \sum_{i \in C_{kk'}}^{n_{kk'}} \mathrm{d}_e^2(\mathbf{x}_i, \mathbf{m}_{kk'}) - \sum_{i \in C_k}^{n_k} \mathrm{d}_e^2(\mathbf{x}_i, \mathbf{m}_k) - \sum_{i \in C_{k'}}^{n_{k'}} \mathrm{d}_e^2(\mathbf{x}_i, \mathbf{m}_{k'}) \\ &= \frac{n_k \; n_{k'}}{n_k + n_{k'}} \mathrm{d}_e^2(\mathbf{m}_k, \mathbf{m}_{k'}), \end{aligned} \tag{2.14}$$

where $\mathrm{d}_e^2()$ is the function for the squared Euclidean distance.

2.3 Clustering and Dimensionality Reduction

This section explores the relationship between clustering and the problem of dimensionality reduction, and in particular, matrix factorization. The following text focuses on variable clustering rather than observation clustering. However, the same analysis could be applied to the problem of observation clustering, where the matrix $\mathbf{X}$ $(n \times p)$ is replaced by its transpose, $\mathbf{X}^\top$.

2.3.1 Clustering as a Matrix Factorization Problem

In most cases, the unsupervised variable clustering problem is formally written as a function of the observed matrix $\mathbf{X}$ $(n \times p)$, a matrix of cluster centroids $\mathbf{M} = [\mathbf{m}_1 \; ... \; \mathbf{m}_K]$ $(n \times K)$ and a variable partition $\mathcal{C} := \{C_1, ..., C_K\}$, where $C_k := \{j \mid j \in \text{ cluster } k\}$. For example, the K-means objective function for variable clustering is

K-means Objective Function (Variable Clustering)

$$J(\mathbf{M}, \mathcal{C}) = \sum_{k=1}^{K} \sum_{j \in C_k} ||\mathbf{x}_j - \mathbf{m}_k||_2^2, \tag{2.15}$$

which is the same as the function in Eq. (2.8), where the indices for observations i are replaced with the indices for variables j. The estimated centroids $\hat{\mathbf{M}}$ and partition $\hat{\mathcal{C}}$ are found by minimizing Eq. (2.15) with respect to $\mathcal{C}$ and $\mathbf{M}$, given $\mathbf{X}$ and a number of clusters K.

When the objective function in Eq. (2.15) is rewritten using matrix notation, it is possible to highlight the relationship between K-means and *matrix factorization* (Bauckhage, 2015). Let $\mathbf{H}$ $(p \times K)$ be a binary cluster membership matrix, where each row j contains exactly one nonzero element, meaning that the columns of $\mathbf{H}$

are orthogonal. Clearly, the variable clusters C_k can be inferred from $\mathbf{H}$: $C_k := \{j \mid h_{jk} = 1\}$ (see Figure 2.7 for an example of how cluster assignments, shown using different colors, can be determined based on the rows of $\mathbf{H}$).

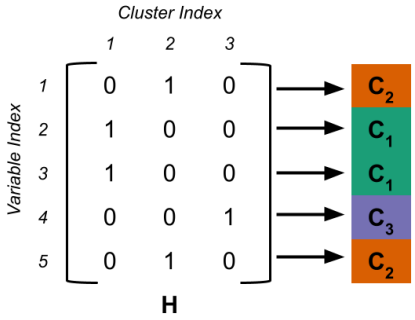


Figure 2.7: **Example of a binary matrix H with orthogonal columns**. The cluster membership of each variable j can be inferred from the corresponding row in $\mathbf{H}$.

Then, the K-means objective function can be rewritten as follows:

K-means Objective Function (Matrix Notation)

$$J(\mathbf{M}, \mathbf{H}) = \sum_{j=1}^{p} ||\mathbf{x}_j - \mathbf{M}\mathbf{h}_j||^2 = ||\mathbf{X} - \mathbf{M}\mathbf{H}^\top||_F^2. \tag{2.16}$$

For fixed $\mathbf{H}$, the solution $\hat{\mathbf{M}}$ that minimizes this problem is

$$\hat{\mathbf{M}} = \mathbf{X}\mathbf{H}(\mathbf{H}^\top\mathbf{H})^{-1}. \tag{2.17}$$

In other words, each centroid $\hat{\mathbf{m}}_k = \mathbf{X}\mathbf{h}_k(\mathbf{h}_k^\top\mathbf{h}_k)^{-1} = \frac{1}{p_k}\sum_{j \in C_k} \mathbf{x}_j$, the mean of the variables in cluster k.

Replacing $\mathbf{M}$ in Eq. (2.16) with $\hat{\mathbf{M}}$, the objective function can be written as a function of $\mathbf{X}$ and $\mathbf{H}$:

$$J(\mathbf{H}) = ||\mathbf{X} - \mathbf{X}\mathbf{H}(\mathbf{H}^\top\mathbf{H})^{-1}\mathbf{H}^\top||_F^2. \tag{2.18}$$

Optimizing this objective function is equivalent to finding the solution

K-means Variable Clustering as a Matrix Factorization Problem

$$\begin{aligned} \hat{\mathbf{H}} = \arg\min_{\mathbf{H}} \; & ||\mathbf{X} - \mathbf{U}\mathbf{V}^\top||_2^2 \\ \text{s.t. } & \mathbf{U} = \mathbf{X}\mathbf{H}(\mathbf{H}^\top\mathbf{H})^{-1/2} \\ & \mathbf{V} = \mathbf{H}(\mathbf{H}^\top\mathbf{H})^{-1/2} \\ & \mathbf{H} \in \{0,1\}^{p\times K} \\ & \mathbf{H} \text{ has orthogonal columns.} \end{aligned} \tag{2.19}$$

2.3.2 Related Matrix Factorization Methods

The reformulation of K-means in Eq. (2.19) highlights the similarities between K-means and the problem of *matrix factorization*. Matrix factorization is a type of dimensionality reduction that seeks to find two matrices $\mathbf{U}$ $(n \times K)$ and $\mathbf{V}$ $(K \times p)$ whose linear combination can reconstruct $\mathbf{X}$ with minimal error. The general objective function for matrix factorization problems is

$$J(\mathbf{U}, \mathbf{V}) = ||\mathbf{X} - \mathbf{U}\mathbf{V}^\top||_2^2, \tag{2.20}$$

which is minimized with respect to $\mathbf{U}$ and $\mathbf{V}$.

The columns of $\mathbf{U}$ are sometimes referred to as *latent variables* because they summarize the latent information in the original dataset $\mathbf{X}$. These latent variables $\mathbf{u}_k$ are the counterpart of the variable centroids $\mathbf{m}_k$ from the clustering framework. The matrix $\mathbf{V}$ is a weight matrix that relates the original variables in $\mathbf{X}$ to the latent variables in $\mathbf{U}$. In certain cases, the weight matrix can bear similarities to the cluster membership matrix $\mathbf{H}$ that is estimated in clustering problems.

Different types of matrix factorizations can be generated by applying different constraints to $\mathbf{U}$ and $\mathbf{V}$ when minimizing Eq. (2.20).

Principal Component Analysis (PCA)

Principal Component Analysis (PCA) constrains $\mathbf{V}$ such that its columns are orthonormal (i.e. $\mathbf{V}^\top\mathbf{V} = \mathbf{I}_K$). Both K-means and PCA impose orthogonality on the columns of $\mathbf{V}$, but for PCA, the elements of $\mathbf{V}$ are not binary, taking real values in the interval $[-1, 1]$. Therefore, PCA can be seen as a relaxation of K-means (Ding and He, 2004). However, no cluster structure can be directly inferred from $\mathbf{V}$. The solution to PCA is

Principle Component Analysis (PCA)

$$(\hat{\mathbf{U}}, \hat{\mathbf{V}}) = \underset{\mathbf{U},\mathbf{V}}{\arg\min} \; ||\mathbf{X} - \mathbf{U}\mathbf{V}^\top||_2^2 \quad \text{s.t. } \mathbf{V}^\top\mathbf{V} = \mathbf{I}_K. \tag{2.21}$$

However, if $\mathbf{V}$ is further constrained to be nonnegative —in addition to having orthonormal columns —only one element in each row j can be nonzero. For such a method, referred to here as Nonnegative PCA (NPCA), cluster structure can be inferred from the elements of $\mathbf{V}$: $C_k := \{j \mid v_{jk} > 0\}$.

Nonnegative Principle Component Analysis (NPCA)

$$(\hat{\mathbf{U}}, \hat{\mathbf{V}}) = \underset{\mathbf{U},\mathbf{V}}{\arg\min} \; ||\mathbf{X} - \mathbf{U}\mathbf{V}^\top||_2^2 \quad \text{s.t. } \mathbf{V}^\top\mathbf{V} = \mathbf{I}_K, \; v_{jk} \geq 0, \; \forall j, k. \tag{2.22}$$

K-means can therefore be seen as a special case of NPCA. The K-means solution $\hat{\mathbf{V}} = \hat{\mathbf{H}}(\hat{\mathbf{H}}^\top\hat{\mathbf{H}})^{-1/2}$ (see Eq. (2.19)) satisfies the original constraints of NPCA for the matrix $\mathbf{V}$: $\hat{\mathbf{V}}$ is nonnegative and $\hat{\mathbf{V}}^\top\hat{\mathbf{V}} = (\hat{\mathbf{H}}^\top\hat{\mathbf{H}})^{-1/2}\hat{\mathbf{H}}^\top\hat{\mathbf{H}}(\hat{\mathbf{H}}^\top\hat{\mathbf{H}})^{-1/2} = \mathbf{I}_K$. However, K-means requires $\mathbf{U}$ and $\mathbf{V}$ to depend on a binary matrix $\mathbf{H}$ with orthogonal columns. Therefore, for K-means, the cluster membership matrix $\hat{\mathbf{V}}$ is not necessarily the same as the solution $\hat{\mathbf{V}}$ for NPCA.

Nonnegative Matrix Factorization (NMF) methods

The relationship between Nonnegative Matrix Factorization (NMF) and K-means is well documented in the literature (Ding, He, and Simon, 2005; Kim and Park, 2008). Like NPCA, one-sided Orthogonal NMF (ONMF) (Ding, Li, Peng, and Park, 2006) constrains $\mathbf{V}$ to be nonnegative and to have orthonormal columns. Thanks to the latter two contraints, the elements of $\mathbf{V}$ encode cluster structure in the same way as above: $C_k := \{j \mid v_{jk} > 0\}$. However, ONMF constrains $\mathbf{U}$ to be nonnegative because the input matrix, $\mathbf{Z}$, is assumed to be nonnegative. This means that ONMF cannot be applied to a matrix with mean-centered columns. The solution to ONMF is

Orthogonal Nonnegative Matrix Factorization (ONMF)

$$
\begin{aligned}
(\hat{\mathbf{U}}, \hat{\mathbf{V}}) = \underset{\mathbf{U},\mathbf{V}}{\arg\min}\ & ||\mathbf{Z} - \mathbf{U}\mathbf{V}^\top||_2^2 \\
\text{s.t. } & \mathbf{V}^\top\mathbf{V} = \mathbf{I}_K \\
& v_{jk} \geq 0,\ \forall j, k \\
& u_{ik} \geq 0,\ \forall i, k.
\end{aligned}
\tag{2.23}
$$

Therefore, K-means is a special case of ONMF for the case where K-means is applied to a nonnegative matrix $\mathbf{Z}$. In such a scenario, the K-means solution $\hat{\mathbf{H}}$ is found by replacing $\mathbf{X}$ with $\mathbf{Z}$ in Eq. (2.19) and solving for $\mathbf{H}$. Note that, because $\mathbf{Z}$ is nonnegative, the solution $\hat{\mathbf{U}} = \mathbf{Z}\hat{\mathbf{H}}(\hat{\mathbf{H}}^\top\hat{\mathbf{H}})^{-1/2}$ is also nonnegative.

2.3.3 Advantages of the Matrix Factorization Framework

Recasting variable clustering as a matrix factorization problem has several advantages. First of all, the literature on matrix factorization is much vaster than the literature on variable clustering, meaning that many methodological and optimization solutions are available in this framework. Furthermore, adding sparsity constraints to matrix factorization problems is very straightforward. This is especially useful in situations where variable clustering is performed in part to select variables: imposing sparsity constraints on the matrix $\mathbf{V}$ makes it possible to remove certain variables from the analysis, as these variables are assigned to none of the clusters. Finally, the cluster memberships v_{jk} are continuous, rather than binary, values. This makes it possible to evaluate the degree to which a variable belongs to its assigned cluster, where higher values indicate a greater degree of membership. Given these potential advantages, this thesis draws parallels between variable clustering and matrix factorization methods wherever it is applicable.

Chapter 3

Unsupervised Variable Clustering

This chapter describes Adaptive Clustering around Latent Variables (AdaCLV), an unsupervised variable clustering method that was originally presented in Marion, Govaerts, and von Sachs (2020). AdaCLV was developed for the purpose of analyzing spectroscopic datasets, such as Nuclear Magnetic Resonance (NMR) spectra. It can be used as a preprocessing step before downstream analyses (e.g. identifying biomarkers), as it performs dimensionality reduction and identifies variables that explain the most variation in the data. AdaCLV is inspired by existing multivariate methods from the Clustering around Latent Variables (CLV) family, but it has several methodological improvements, which are detailed in this chapter. Moreover, experiments involving real and semi-artificial NMR spectra demonstrate that AdaCLV is more robust to hyperparameter choice and has better precision than other competing methods, for all cluster numbers, sample sizes and levels of signal tested, while achieving a comparable level of recall.

Rebecca Marion, Bernadette Govaerts, and Rainer von Sachs (2020). "AdaCLV for Interpretable Variable Clustering and Dimensionality Reduction of Spectroscopic Data". In: **Chemometrics and Intelligent Laboratory Systems** 206, p. 104169

List of Symbols

$\mathbf{X}$: $n \times p$ matrix with centered, but not scaled, columns

$\mathbf{Z}$: $n \times p$ nonnegative matrix

$\mathbf{U}$: $n \times K$ matrix of latent variables

$\mathbf{V}$: $p \times K$ matrix of nonnegative cluster memberships (binary or continuous)

$\mathbf{r}_k$: vector of sample covariances between latent variable $\mathbf{u}_k$ and each variable $\mathbf{x}_j$, i.e. $(\text{cov}(\mathbf{u}_k, \mathbf{x}_1)$ ... $\text{cov}(\mathbf{u}_k, \mathbf{x}_p))^\top$

See page 7 for other symbols used in this chapter.

3.1 Introduction

3.1.1 Problem Definition

Let $\mathbf{X}$ $(n \times p)$ be a data matrix where n is the number of observations and p is the number of variables. In what follows, the columns of $\mathbf{X}$ are assumed to be mean-centered but not scaled. This is a reasonable assumption for the analysis of Nuclear Magnetic Resonance (NMR) spectra, which contain many low-variance noise variables. As explained in Section 2.2.1, scaling variables in such a context can be detrimental to clustering performance. Instead, it is common practice to center the variables only.

Observations are indexed with the letter $i \in \{1, ..., n\}$, variables with the letter $j \in \{1, ..., p\}$ and clusters with the letter $k \in \{1, ..., K\}$, where K is the total number of clusters to be found and $C_k := \{j \in \text{cluster } k\}$ is the set of p_k variables j belonging to a given cluster k.

Given the observed matrix $\mathbf{X}$, we are interested in achieving the three main objectives described in Section 1.1: performing dimensionality reduction and variable selection, as well as identifying grouping structure among variables. The first two objectives are particularly important in the analysis of data from the field of *metabolomics*, where datasets are high-dimensional ($p \gg n$) and only some of the observed variables are related to the object of study (e.g. disease diagnosis or progression). These variables are called *biomarkers* because they are biological markers that are predictive of the health outcome of interest.

3.1.2 Biomarker Discovery in Metabolomics

Metabolomics is a field in which biomarker discovery is of particular interest. In metabolomics, samples of biofluids such as urine or blood are analyzed in search of candidate biomarkers (e.g. metabolites). One popular means of quantifying

metabolite concentrations is through Nuclear Magnetic Resonance (NMR) spectroscopy. NMR spectra are highly reproducible, meaning that most of the variability between spectra represents biological differences between samples (Bingol, Bruschweiler-Li, Li, Zhang, Xie, and Brüschweiler, 2016; Boccard and Rudaz, 2014). Moreover, as NMR spectroscopy can quantify both known and unknown molecules, it is particularly useful for identifying and characterizing new compounds (Markley, Brüschweiler, Edison, Eghbalnia, Powers, Raftery, and Wishart, 2017), making it an important platform for biomarker discovery.

In an NMR spectrum, groups of consecutive variables form peaks. Each molecule detected in the sample is represented by one or more of these peaks, and the height of these peaks is proportional to the concentration of the molecule (Muncey, Jones, De Iorio, and Ebbels, 2010; Markley, Brüschweiler, Edison, Eghbalnia, Powers, Raftery, and Wishart, 2017). Identifying biomarkers boils down to identifying the sets of spectral variables or regions (i.e. variable clusters) that most predict the health outcome of interest. Consider the dataset represented in Figure 3.1, which depicts the mean intensities for NMR spectra of urine. Suppose that fluctuations in two metabolites, citrate and hippurate, are associated with the presence or absence of a hypothetical disease. The variables corresponding to these two metabolites, highlighted in green (citrate) and red (hippurate), are therefore biomarkers for the disease.

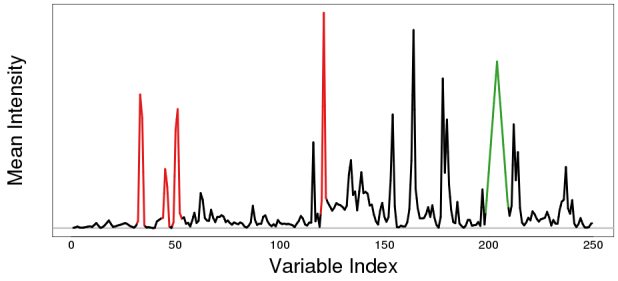


Figure 3.1: **Example spectrum for a hypothetical biomarker study**. Mean intensities for each spectral variable in an NMR dataset representing the urine of rats. If the metabolites citrate (variables highlighted in green) and hippurate (highlighted in red) are associated with a given health outcome, then the variables that represent these metabolites are biomarkers. Methods for biomarker discovery should be able to identify these highlighted variables as potential biomarkers, ideally separating them into two distinct groups.

Despite the potential of NMR data for the discovery of biomarkers, identifying predictive spectral regions is complicated by the complex nature of the data. In practice, these datasets often have many more variables than observations, mak-

ing them very ill conditioned for classical multivariate analyses (Liland, 2011). Moreover, while the underlying dimension of the data is the number of measurable molecules in the mixture (Lindon, Nicholson, Holmes, and Everett, 2000), the actual number of dimensions is much larger, which creates complex correlation structures in the data. Given these challenges, it can be useful to first pre-process NMR data before performing analyses for the identification of biomarkers. One particular pre-processing strategy is to identify the variables that vary the most between samples and to assign these variables to clusters. Afterwards, these clusters can be further analyzed to determine which ones are predictive of the health outcome of interest.

The underlying variable clusters in NMR spectra can be defined according to the "positive correlation" perspective described in Section 2.1. Variables that represent the same molecule (i.e. belong to the same cluster) tend to have a high positive correlation, whereas variables representing different molecules tend to be somewhat negatively correlated. This negative correlation can occur even in the absence of a biological or chemical relationship between the molecules. Indeed, NMR spectra represent mixtures of molecules and thus are inherently compositional: increases in the concentration of one molecule are accompanied by decreases in the concentration of the others. As a result, the latent trends represented in NMR spectra (i.e. fluctuations in molecule concentrations) are inherently non-orthogonal.

Overlapping peaks are very common in NMR spectra (Bingol, Bruschweiler-Li, Li, Zhang, Xie, and Brüschweiler, 2016), resulting in positive between-cluster correlation for variables in overlapping regions of the spectra. The variables in these overlapping regions behave like outliers (see Section 2.2.1 and Figure 2.5), as they are positively correlated with multiple clusters at once. The presence of overlapping peaks, and thus overlapping clusters, motivates the use of clustering methods that can assign variables to multiple clusters, circumventing the outlier problem.

Finally, NMR spectra contain entire regions of uncorrelated, low-variance variables. These variables constitute a *noise cluster* (see Section 2.2.1) that contains no relevant signal. Regions with peaks may also contain low-variance variables, especially if the molecules they represent do not fluctuate between samples. The variables within these low-variance peaks are correlated because they represent the same molecule. However, they do not contain any signal relevant to disease prediction. Thus, in the context of biomarker studies, these low-variance peaks may be treated similarly to the uncorrelated variables (e.g. removed from the analysis). In what follows, both types of low-variance variables (correlated and uncorrelated) are referred to as *irrelevant variables* and irrelevant, uncorrelated variables are referred to as *noise variables*.

3.1.3 Contributions

In this chapter, we propose an unsupervised multivariate method, Adaptive Clustering around Latent Variables (AdaCLV), that pre-processes spectroscopic data by simultaneously performing variable selection, variable clustering and dimensionality reduction: AdaCLV automatically detects the spectral variables that most differentiate observations, assigns them to one or more variable clusters and estimates non-orthogonal summary vectors (or *latent variables*), one for each cluster. This makes it possible to circumvent the dimensionality problem encountered for spectroscopic data, as well as problems arising from the presence of irrelevant variables and complex correlation structure. Indeed, the outputs of AdaCLV (variable cluster assignments and latent variables) capture the most important information from the original dataset but can be more easily analyzed. AdaCLV can thus be a useful preprocessing step in a pipeline for biomarker discovery.

AdaCLV determines cluster memberships by estimating membership degrees that indicate the relative importance of variables within a cluster, making it possible to rank variables. For a given cluster, these membership degrees are shrunk towards zero with a threshold that iteratively adapts to clustering structure in the data. As a result, only the variables that most discriminate observations (i.e. explain the most variation in the dataset) have non-zero cluster membership degrees. In addition to identifying important variables for subsequent analyses, AdaCLV generates a set of latent variables that summarize important information from the original dataset in a lower-dimensional form. These latent variables can be used in downstream analyses, such as regression or classification (see e.g. Bühlmann, Rütimann, Geer, and Zhang (2013)), which can be used to identify potential biomarkers.

AdaCLV is based on the Clustering around Latent Variables (CLV) approach, which was previously developed in Vigneau and Qannari (2003) and Vigneau (2016). Like AdaCLV, CLV (Vigneau and Qannari, 2003) clusters variables into K groups using a covariance-based criterion and estimates K non-orthogonal latent variables. However, CLV clusters are non-overlapping and cluster memberships are binary: a variable either belongs to a cluster or it does not. Thus, these memberships are less informative than AdaCLV memberships, which indicate the relative importance of variables within a cluster and reflect possible cluster overlap. Moreover, when estimating latent variables, CLV does not account for differing degrees of membership within a given cluster, and it is relatively sensitive to the choice of the hyperparameter K, the number of clusters to detect.

One further limitation of CLV is that cluster assignment and latent variable estimation is adversely affected by the presence of uncorrelated, low-intensity variables in noisy regions of the spectra (*noise variables*). Two methods were recently proposed in Vigneau (2016) that attempt to address this issue. CLV $K + 1$ identifies noise variables using a correlation-based threshold at each iteration of its algorithm and removes these variables from the analysis. As a result, noise variables can be

assigned to a "garbage" cluster (cluster $K+1$) and are not used to estimate latent variables. The second method proposed in Vigneau (2016), CLV Sparse LV, also uses a correlation-based threshold to exclude noise variables during the estimation of latent variables. However, CLV Sparse LV remains sensitive to noise for its clustering step, as it assigns each variable, even noise variables, to one of the K clusters. Moreover, for both methods, the filtering of noise variables depends on an extra hyperparameter that must be tuned by the user. As with CLV, these methods depend on binary memberships, cannot find overlapping clusters and are sensitive to the choice of K.

Other existing unsupervised methods also have limitations with respect to AdaCLV. Principal Component Analysis (PCA) is by far the most popular method for linear dimensionality reduction, generating orthogonal latent variables that represent major sources of variance in the data. While it is possible to identify important variables for each of the latent variables, the requirement of orthogonality between latent variables may be too rigid for many natural phenomena, which are not necessarily orthogonal (Liland, 2011). This is particularly true for spectral datasets, where, as explained in Section 3.1.2, the latent phenomena of interest are inherently correlated. As an alternative to PCA, methods built for observation clustering, such as K-means or hierarchical clustering, can be applied to cluster variables. However, like existing CLV approaches, these methods define cluster membership as binary, and they are not adapted for datasets with overlapping clusters.

This chapter summarizes several contributions. First, it presents AdaCLV, an unsupervised method for performing dimensionality reduction and identifying potentially overlapping clusters of relevant variables. In order to better appreciate and understand our novel contribution, we complete the chapter by giving a fairly detailed review of related methods. Finally, a rigorous comparison of AdaCLV with existing CLV methods and baseline approaches (K-means, Ward's hierarchical clustering and Sparse Nonnegative Matrix Factorization) is performed using simulations where the ground-truth is known and performance can be evaluated using numerical criteria. This numerical study demonstrates that AdaCLV estimates target cluster memberships and latent variables better than its competitors. It also shows that AdaCLV, unlike the other methods, is relatively insensitive to the choice of the number K of clusters to detect.

3.1.4 Chapter Outline

AdaCLV is presented in Section 3.2 and related works are summarized in Section 3.3. Numerical experiments and results are described in Section 3.4 and the conclusion and suggestions for future work can be found in Section 3.5.

3.2 Proposed Method: AdaCLV

3.2.1 Objective

AdaCLV is a method for dimensionality reduction and variable clustering inspired by the Clustering around Latent Variables (CLV) approaches introduced in Vigneau and Qannari (2003) and Vigneau (2016) (see Section 3.3). In contrast to existing CLV methods, AdaCLV accounts for differing degrees of cluster membership within a cluster and estimates more informative cluster memberships. It also identifies the variable clusters that most differentiate observations. These variable clusters can be studied in subsequent analyses to identify potential biomarkers.

Figure 3.2 shows an example of how AdaCLV may be used in a pipeline for biomarker discovery. Given a high-dimensional matrix $\mathbf{X}$ of spectra, represented here with a light blue rectangle, AdaCLV can be used to pre-process this matrix (step 1) in order to improve the identification of biomarkers in a subsequent analysis (step 2). AdaCLV does this by identifying clusters of variables that are the most important for explaining variation in the data, which is visualized here by highlighting the columns of $\mathbf{X}$ with different colors. Columns with the same color have been assigned to the same cluster, and the grey columns represent variables that have been identified as irrelevant (i.e. they do not explain much variation in the data). After variable selection and clustering, the dimensionality of the original matrix is reduced, facilitating the downstream analysis for identifying biomarkers (step 2). Moreover, using the cluster assignments in the analysis can improve biomarker selection, as correlation structure can be directly accounted for. For example, the matrix of selected variables ($\tilde{\mathbf{X}}$) and the cluster assignments can be used as inputs to the Group Lasso (Yuan and Lin, 2006), which can identify which of the clusters are most relevant for predicting a given health outcome (see e.g. Bühlmann, Rütimann, Geer, and Zhang (2013)). Once the final set of predictive clusters has been selected, the cluster membership degrees estimated by AdaCLV can be used to weed out potential false positives (variables with low memberships) and to identify the most important variables within each cluster (variables with high memberships).

AdaCLV simultaneously estimates two matrices: $\mathbf{U}$, an $n \times K$ latent representation of the data, and $\mathbf{V}$, a $p \times K$ cluster membership matrix. Each column vector $\mathbf{u}_k$ in $\mathbf{U}$ is a *latent variable* representing the essential information in cluster k, and each column vector $\mathbf{v}_k$ in $\mathbf{V}$ is a cluster membership profile for cluster k. For a given variable j and cluster k, v_{jk} represents a nonnegative membership degree: $v_{jk} \in [0, 1]$, where $v_{jk} = 0$ indicates lack of membership in cluster k. Variable j is assigned to cluster k (i.e. $v_{jk} > 0$) if and only if $\text{cov}(\mathbf{u}_k, \mathbf{x}_j) \geq \tau_k$, where $\tau_k \geq 0$ is a threshold parameter that is updated adaptively at each iteration of the algorithm.

For a given cluster k and fixed τ_k, the criterion to maximize with respect to $\mathbf{u}_k$

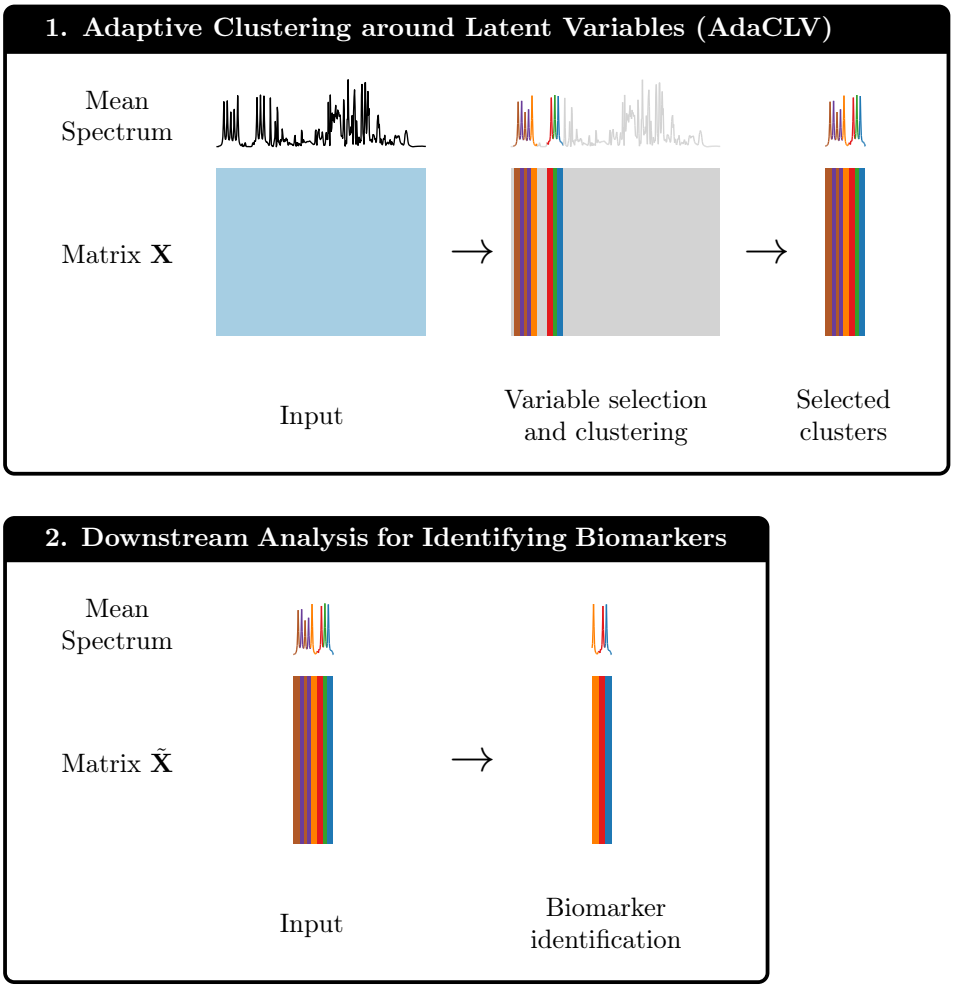


Figure 3.2: **Pipeline for biomarker discovery based on AdaCLV.** Given a matrix $\mathbf{X}$ of spectra, AdaCLV can be used to identify clusters of variables that are the most important for explaining variation in the data (step 1). This is visualized here by highlighting the columns of $\mathbf{X}$ with different colors, where columns (i.e. variables) with the same color have been assigned to the same cluster. The grey columns represent variables that have been identified as irrelevant (i.e. they do not explain much variation in the data). The variables selected by AdaCLV (the matrix $\tilde{\mathbf{X}}$), as well as the cluster assignments, can be used as input to a second analysis, where the variable clusters associated with a given health outcome (i.e. biomarkers) are identified (step 2).

and $\mathbf{v}_k$ is

AdaCLV Objective Function

$$J(\mathbf{u}_k, \mathbf{v}_k) = \sum_{j=1}^{p} v_{jk} \left[\text{cov}(\mathbf{u}_k, \mathbf{x}_j) - \tau_k\right] \tag{3.1}$$

subject to the following constraints:

1. $v_{jk} \geq 0, \forall j$ (nonnegativity)
2. $\mathbf{v}_k^\top \mathbf{v}_k = 1$ (membership profiles have unit length)
3. $\text{var}(\mathbf{u}_k) = 1$ (latent variables have unit variance).

Constraint 1 ensures that the cluster memberships are nonnegative, and it also has the convenient property of encouraging sparseness in the cluster membership vectors $\mathbf{v}_k$. Indeed, v_{jk} will be set to zero if $\text{cov}(\mathbf{u}_k, \mathbf{x}_j) < \tau_k$, which discourages negatively correlated variables from being clustered together. Constraints 2 and 3 are imposed to avoid scale ambiguity. Moreover, constraint 2 ensures that the membership values v_{jk} reflect the relative importance of variable j within cluster k.

AdaCLV cluster memberships are "soft" in the sense that they are non-binary, but they are slightly different than the typical soft memberships estimated for soft or "fuzzy" clustering problems (Nayak, Naik, and Behera, 2015). Indeed, for fuzzy clustering problems, the scale of the rows in $\mathbf{V}$ is fixed ($\sum_{k=1}^{K} v_{jk} = 1, \forall j$), whereas for AdaCLV the scale of the columns is fixed ($\sum_{j=1}^{p} v_{jk}^2 = \mathbf{v}_k^\top \mathbf{v}_k = 1, \forall k$). Thus, fuzzy clustering memberships indicate the degree to which an object belongs to each of the K clusters, but they do not indicate the relative importance of objects within a cluster. Moreover, by design, classical fuzzy clustering methods assign each object to all of the clusters, as there is no mechanism for encouraging membership degrees of zero. This makes it difficult to interpret the cluster memberships for a given object.

In contrast to other dimensionality reduction methods, the columns of the latent variable matrix $\mathbf{U}$ are not required to be orthogonal. This is advantageous for datasets representing mixtures, such as Nuclear Magnetic Resonance spectra, where the true latent phenomena (e.g. molecule concentrations) are inherently non-orthogonal (see Section 3.1.2 for more information).

AdaCLV also does not constrain the cluster membership matrix $\mathbf{V}$ to be orthogonal. This means that if a variable has sufficiently high covariance with multiple latent variables, it can be assigned to multiple clusters. As overlapping peaks are common in NMR spectra (see Section 3.1.2), this property is particularly useful.

Moreover, this non-orthogonality makes AdaCLV more robust to poor initializations. When the initial partition contains too many clusters with respect to the true partition, AdaCLV allows the estimation of redundant clusters (i.e. equivalent cluster membership vectors), thereby avoiding false discoveries.

3.2.2 Estimation

For fixed $\mathbf{v}_k$, the optimal latent variable $\hat{\mathbf{u}}_k$ for the constrained problem above is

$$\hat{\mathbf{u}}_k = \sqrt{n-1}\ \mathbf{X}\mathbf{v}_k / ||\mathbf{X}\mathbf{v}_k||_2. \tag{3.2}$$

For fixed τ_k and $\mathbf{u}_k$, a vector of memberships $\mathbf{v}_k$ can be estimated, where each element is calculated as follows:

$$\hat{v}_{jk} = \frac{(\text{cov}(\mathbf{u}_k, \mathbf{x}_j) - \tau_k)_+}{\sqrt{\sum_{\ell=1}^{p} (\text{cov}(\mathbf{u}_k, \mathbf{x}_\ell) - \tau_k)_+^2}}. \tag{3.3}$$

For fixed $\mathbf{v}_k$ and $\mathbf{u}_k$, the adaptive threshold τ_k is determined using a procedure that takes advantage of covariance properties commonly observed in spectral data. For a given true cluster k, variables in spectral data generally fall into one of three categories: 1) variables that do not contain any discriminating signal (*irrelevant variables*), 2) variables in signal regions that belong to a cluster other than cluster k and 3) variables in signal regions that belong to cluster k. Irrelevant variables (category 1) should not be assigned to any of the clusters or used to estimate latent variables. The covariance of these variables with the true latent variables is generally of small magnitude and distributed around 0. Variables in the second category tend to have negative or small positive covariance with the latent variable for cluster k. This is in part due to the compositional nature of NMR data, which is explained in greater detail in Section 3.1.2. Finally, the third category of variable, in contrast to the other two categories, has relatively high positive covariance with the latent variable for cluster k.

Taken together, these insights suggest that the covariance of variables with a given latent variable k can be used to distinguish the variables $j \in C_k$ (category 3) from variables $j \notin C_k$ (categories 1 and 2). The threshold between these two groups appears somewhere near the maximum covariance observed for the variables in categories 1 and 2 and the minimum covariance observed for category 3. AdaCLV determines this threshold using a two-step procedure. In the first step, irrelevant variables are identified. Given an initial vector $\mathbf{r}_k = (\text{cov}(\mathbf{u}_k, \mathbf{x}_1)\ ...\ \text{cov}(\mathbf{u}_k, \mathbf{x}_p))^\top$ and a user-defined hyperparameter $\eta > 0$, an algorithm for irrelevant variable identification (IVI) removes extreme elements from $\mathbf{r}_k$ until only non-extreme (i.e. irrelevant) variables remain. The algorithm, which is inspired by Golotvin and Williams (2000), proceeds as follows:

1. Calculate the sample mean and standard deviation of $\mathbf{r}_k$: $\overline{r}_k = \text{mean}(\mathbf{r}_k)$ and $\hat{\sigma}_k = \text{sd}(\mathbf{r}_k)$
2. Remove all elements j from $\mathbf{r}_k$ for which $|r_{jk}| > \overline{r}_k + \eta\hat{\sigma}_k$
3. Repeat steps 1 and 2 until no values $|r_{jk}| > \overline{r}_k + \eta\hat{\sigma}_k$

In the second step of the threshold estimation procedure, the output $\mathbf{r}_k$ from the first step is used to calculate τ_k, where

$$\tau_k = (\max_j(r_{jk}))_+. \tag{3.4}$$

Note that the algorithm for IVI assumes that the covariance of irrelevant variables with a given latent variable k is symmetrically distributed. The maximum of this distribution is also assumed to be smaller than $\text{cov}(\mathbf{u}_k, \mathbf{x}_j)$ for most variables $j \in C_k$ and greater than $\text{cov}(\mathbf{u}_k, \mathbf{x}_j)$ for signal variables $j \notin C_k$.

3.2.3 Algorithm and Hyperparameters

The entire AdaCLV algorithm is summarized below in Algorithm 1. At the beginning of the algorithm, $\mathbf{V}$ is initialized based on an initial partition $C_1, ..., C_K$ of K variable clusters, where $v_{jk} = \mathbb{I}[j \in C_k] / \sqrt{p_k}$. Then, at each iteration of the algorithm, the latent variable matrix $\mathbf{U}$, the threshold vector $\boldsymbol{\tau}$ and the membership matrix $\mathbf{V}$ are updated successively.

AdaCLV requires the specification of two user-defined hyperparameters: η and K, the number of clusters in the initial partition. These hyperpamaters may be chosen using cluster validity indices (CVI), which measure the quality of candidate partitions without requiring knowledge of the ground-truth. Entire papers in the clustering literature are devoted to identifying the best CVI for a given problem and clustering algorithm (see Chiang and Mirkin (2010), Arbelaitz, Gurrutxaga, Muguerza, Pérez, and Perona (2013), and Kaczyńska, Marion, and von Sachs (2020)). Indeed, the best choice of CVI depends on many factors: the true number of clusters, cluster shape, size and overlap, the data dimensionality and noise level, and the clustering method used, among others. Choosing an optimal CVI for AdaCLV is therefore beyond the scope of this chapter.

We can nonetheless offer some guidance on a potential default value or range of values for η. Let $\eta_{\max}$ be the smallest value of η such that the entire matrix $\mathbf{V}$ is zero. Based on our experiments, η values in the range $[0.05, 0.65] \times \eta_{\max}$ seem to yield good results in general, with $0.5 \times \eta_{\max}$ being a reasonable choice in all cases tested. The value of $\eta_{\max}$ can be identified when the vectors $\mathbf{r}_k$ are calculated for the first time (i.e. at the first iteration of the AdaCLV algorithm). The algorithm for IVI will label all variables as irrelevant for cluster k if $\max_j(|r_{jk}|) \leq \overline{r}_k + \eta\hat{\sigma}_k$. If this occurs, the vector of memberships $\mathbf{v}_k$ will be set to zero. The entire matrix

Data: A centered matrix $\mathbf{X}$ $(n \times p)$, initial partition $C_1, ..., C_K$, hyperparameter $\eta > 0$

// Initialization
for each cluster k **in** $1, ..., K$ **do**
  for each variable j **in** $1, ..., p$ **do**
    $v_{jk} = \mathbb{I}\left[j \in C_k\right] / \sqrt{p_k}$;
  end
end
// Main Algorithm
while not converged do
  for each cluster k **in** $1, ..., K$ **do**
    // Update latent variable k
    $\mathbf{u}_k = \sqrt{n-1}\, \mathbf{X}\mathbf{v}_k / ||\mathbf{X}\mathbf{v}_k||_2$;
    // Update adaptive threshold k
    Given the hyperparameter η, update τ_k using the algorithm for IVI described above
    // Update membership vector k
    for each variable j **in** $1, ..., p$ **do**
      $v_{jk} = \frac{\left(\text{cov}(\mathbf{u}_k, \mathbf{x}_j) - \tau_k\right)_+}{\sqrt{\sum_{\ell=1}^{p} \left(\text{cov}(\mathbf{u}_k, \mathbf{x}_\ell) - \tau_k\right)_+^2}}$;
    end
  end
end
return Latent representation U and sparse cluster membership matrix V

Algorithm 1: AdaCLV

$\mathbf{V}$ will be set to zero if $\max_j(|r_{jk}|) \leq \overline{r}_k + \eta\hat{\sigma}_k, \forall k$. Thus, the maximum value for η can be defined as follows:

$$\eta_{\max} = \max_k \left[\left(\max_j(|r_{jk}^{(1)}|) - \overline{r}_k^{(1)} \right) / \hat{\sigma}_k^{(1)} \right], \tag{3.5}$$

where $\mathbf{r}_k^{(1)}$ is the input to the algorithm for IVI at the first iteration of AdaCLV, and $\overline{r}_k^{(1)} = \text{mean}(\mathbf{r}_k^{(1)})$ and $\hat{\sigma}_k^{(1)} = \text{sd}(\mathbf{r}_k^{(1)})$.

AdaCLV is relatively insensitive to the choice of K, the number of initial clusters. This is especially true when K is larger than the true number of clusters. Therefore, it is generally better to choose larger values of K, i.e. a number larger than the expected number of clusters. The initial clusters may be identified using another dedicated clustering algorithm, such as the hierarchical clustering method proposed in Vigneau and Qannari (2003). Alternatively, multiple random partitions may be

tested and the solution with the best cluster quality may be selected according to a given CVI.

3.3 Related Work

AdaCLV is closely related to the Clustering around Latent Variables (CLV) family of methods, which are in turn related to K-means variable clustering and certain matrix factorization methods. These related methods are summarized below.

3.3.1 Clustering around Latent Variables (CLV)

Clustering around Latent Variables (CLV) (Vigneau and Qannari, 2003) was introduced for the purpose of data exploration through simultaneous dimensionality reduction and variable clustering. Vigneau and Qannari (2003) proposed two different modes for CLV: local and directional mode. The objective function for the local mode is based on covariances, whereas it is based on squared covariances for the directional mode. This means that when the criterion for the directional mode is maximized, only the magnitude of covariances is considered, not their signs.

For NMR spectra representing mixtures, the directional mode may not always be appropriate. This is because variables are generally positively correlated when they represent the same molecule (i.e. belong to the same cluster) but can be somewhat negatively correlated when they represent different molecules (i.e. belong to different clusters). Thus, the sign of pairwise variable covariances provides useful information about the cluster membership of variables. While the local mode can capitalize on this information, the directional mode cannot.

For the local mode, the matrices $\mathbf{U}$ and $\mathbf{V}$ are found by maximizing the function

CLV Objective Function

$$J(\mathbf{U}, \mathbf{V}) = \sum_{k=1}^{K} \sum_{j=1}^{p} v_{jk} \mathrm{cov}(\mathbf{u}_k, \mathbf{x}_j) = \frac{1}{n-1} \sum_{k=1}^{K} \mathbf{u}_k^\top \mathbf{X} \mathbf{v}_k \tag{3.6}$$

with respect to $\mathbf{U}$ and $\mathbf{V}$, subject to the constraints

1. $v_{jk} \in \{0, 1\}, \forall j, k$ (binary memberships)
2. $\sum_{k=1}^{K} v_{jk} = 1, \forall j$ (each variable assigned to one cluster)
3. $\mathrm{var}(\mathbf{u}_k) = 1, \forall k$ (latent variables have unit variance)

Variable clusters are a function of the membership values and vice versa: $C_k := \{j \mid v_{jk} = 1\}$ and $v_{jk} = \mathbb{I}[j \in C_k]$. Given a binary membership vector $\mathbf{v}_k$, the optimal solution $\hat{\mathbf{u}}_k$ to this constrained problem is the same as for AdaCLV (see Eq. (3.2)). For a given latent component $\mathbf{u}_k$, the function in Eq. (3.6) is maximized when $v_{jk} = \mathbb{I}[k = \arg\max_\ell \text{cov}(\mathbf{u}_\ell, \mathbf{x}_j)]$.

The matrices $\mathbf{U}$ and $\mathbf{V}$ are estimated using alternating optimization. Starting from an initial set of variable clusters $C_1^{(0)}, ..., C_K^{(0)}$ and membership values $v_{jk}^{(0)} = \mathbb{I}\left[j \in C_k^{(0)}\right]$, the following steps are repeated for $t = 1$ until convergence:

1. Calculate $\mathbf{u}_k^{(t)} = \sqrt{n-1}\frac{\mathbf{X}\mathbf{v}_k^{(t-1)}}{||\mathbf{X}\mathbf{v}_k^{(t-1)}||_2}$ for all k

2. Update cluster assignments for all k s.t.
   $C_k^{(t)} := \{j \mid k = \arg\max_\ell \text{cov}(\mathbf{u}_\ell^{(t)}, \mathbf{x}_j)\}$

3. Update memberships for all j and k s.t. $v_{jk}^{(t)} = \mathbb{I}\left[j \in C_k^{(t)}\right]$

At iteration t, the only variables that contribute to the estimation of latent variable $\mathbf{u}_k^{(t)}$ are the variables j for which $k = \arg\max_\ell \text{cov}(\mathbf{u}_\ell^{(t-1)}, \mathbf{x}_j)$ (i.e. $v_{jk}^{(t-1)} = 1$). Thus, each variable contributes to the estimation of only one of the K latent variables (see Table 3.1 for a summary). Vigneau and Qannari (2003)'s proposed initialization strategy is summarized in Section 3.7.4.

Although Vigneau and Qannari (2003) do not study the relationship between K-means variable clustering and CLV (local mode), it can be shown that the two methods are closely related. When K-means is used for variable clustering, the centroid matrix $\mathbf{U}$ and cluster membership matrix $\mathbf{V}$ are estimated as follows:

K-means Variable Clustering

$$\begin{aligned}(\hat{\mathbf{U}}, \hat{\mathbf{V}}) = &\arg\min_{\mathbf{U},\mathbf{V}} \ \frac{1}{2(n-1)}||\mathbf{X} - \mathbf{U}\mathbf{V}^\top||_F^2 \\ &\text{s.t. } \mathbf{V} \in \{0,1\}^{p\times K} \\ &\quad\ \mathbf{V} \text{ has orthogonal columns}\end{aligned} \tag{3.7}$$

For more details on this formulation of the K-means problem, see Section 2.3.1.

The objective function in Eq. (3.7) can be simplified such that

$$
\begin{aligned}
\frac{1}{2(n-1)}||\mathbf{X}-\mathbf{U}\mathbf{V}^\top||_F^2 &= \frac{1}{2(n-1)}\mathrm{Tr}\left(\left[\mathbf{X}-\mathbf{U}\mathbf{V}^\top\right]\left[\mathbf{X}-\mathbf{U}\mathbf{V}^\top\right]^\top\right) \\
&= \frac{1}{2(n-1)}\left(\mathrm{Tr}(\mathbf{X}\mathbf{X}^\top) - 2\mathrm{Tr}(\mathbf{U}^\top\mathbf{X}\mathbf{V}) + \mathrm{Tr}(\mathbf{U}\mathbf{V}^\top\mathbf{V}\mathbf{U}^\top)\right) \\
&= \frac{1}{2(n-1)}\left(c_1 - 2\sum_{k=1}^{K}\mathbf{u}_k^\top\mathbf{X}\mathbf{v}_k + \sum_{k=1}^{K}\underbrace{\mathbf{v}_k^\top\mathbf{v}_k}_{p_k}\ \mathbf{u}_k^\top\mathbf{u}_k\right) \\
&= c_2 - \frac{1}{n-1}\sum_{k=1}^{K}\mathbf{u}_k^\top\mathbf{X}\mathbf{v}_k + \frac{1}{2(n-1)}\sum_{k=1}^{K}p_k\ \mathbf{u}_k^\top\mathbf{u}_k \\
&= c_2 - \underbrace{\sum_{k=1}^{K}\sum_{j=1}^{p}v_{jk}\mathrm{cov}(\mathbf{u}_k,\mathbf{x}_j)}_{\text{CLV objective function}} + \frac{1}{2}\sum_{k=1}^{K}p_k\ \mathrm{var}(\mathbf{u}_k),
\end{aligned}
\tag{3.8}
$$

where c_1 and c_2 are constants that do not depend on $\mathbf{U}$ or $\mathbf{V}$. If the centroids $\mathbf{u}_k$ are constrained to have unit variance for all k, then K-means variable clustering is equivalent to CLV. In this case, the third term in the last row of Eq. (3.8) becomes constant. The only term that depends on $\mathbf{U}$ and $\mathbf{V}$ is the second term, and minimizing this term is equivalent to maximizing Eq. (3.6), the objective function of CLV.

In summary, the solution to CLV can be rewritten using matrix notation as follows:

CLV as a Special Case of K-means Variable Clustering

$$
\begin{aligned}
(\hat{\mathbf{U}},\hat{\mathbf{V}}) = \underset{\mathbf{U},\mathbf{V}}{\arg\min}\ & \frac{1}{2(n-1)}||\mathbf{X}-\mathbf{U}\mathbf{V}^\top||_F^2 \\
\text{s.t. } & \mathbf{V}\in\{0,1\}^{p\times K} \\
& \mathbf{V} \text{ has orthogonal columns} \\
& \mathrm{var}(\mathbf{u}_k) = 1\ \forall k
\end{aligned}
\tag{3.9}
$$

The directional mode of CLV is summarized in Section 3.7.1.

3.3.2 CLV $K+1$

Vigneau (2016) proposed CLV $K+1$ to iteratively filter variables so that variables representing noise ("garbage" variables) do not contribute to the estimation of $\mathbf{U}$ and $\mathbf{V}$. Let $\gamma\in[0,1]$ be a user-defined hyperparameter for filtering variables. For the local mode, CLV $K+1$ maximizes the modified CLV objective function

CLV $K+1$ Objective Function

$$J(\mathbf{U},\mathbf{V}) = \sum_{j=1}^{p} \left\{ \sum_{k=1}^{K} v_{jk}\text{cov}(\mathbf{u}_k, \mathbf{x}_j) + \gamma \left(1 - \sum_{k=1}^{K} v_{jk}\right) \text{sd}(\mathbf{x}_j) \right\} \quad (3.10)$$

with respect to $\mathbf{U}$ and $\mathbf{V}$, subject to the same constraints as CLV, except that constraint 2 ($\sum_{k=1}^{K} v_{jk} = 1, \forall j$) becomes $\sum_{k=1}^{K} v_{jk} \leq 1, \forall j$, to allow for the presence of a garbage cluster.

For a given element j in the sum in Eq. (3.10), only the first or second term inside the curly brackets can be activated at once. The second term is activated if $v_{jk} = 0$ for all $k \in \{1, ..., K\}$, which means that variable j is assigned to cluster $K+1$ (the "garbage" cluster). This solution maximizes the constrained problem when, for all k, $\text{cor}(\mathbf{u}_k, \mathbf{x}_j) < \gamma$. The first term in Eq. (3.10) is only activated if there is a $k \in \{1, ..., K\}$ for which $\text{cor}(\mathbf{u}_k, \mathbf{x}_j) > \gamma$. If this is the case, the solution v_{jk} maximizing the constrained problem is the same as for CLV.

The hyperparameter γ controls the number of variables that are assigned to the "garbage" cluster: as $\gamma \to 1$, more variables satisfy the condition $\text{cor}(\mathbf{u}_k, \mathbf{x}_j) < \gamma$ for all k, so more variables are assigned to the garbage cluster. Note that the threshold γ is the same for each cluster k, which is in contrast to AdaCLV, where the membership threshold is different for each cluster. Moreover, CLV $K+1$ only attempts to remove noise variables, whereas AdaCLV removes all types of irrelevant variables from the analysis.

The CLV $K+1$ algorithm is the same as the CLV algorithm (see Section 3.3.1), except that the cluster assignment step (Step 2) is modified and a third step is added for assigning variables to a garbage cluster: $C_{K+1}^{(t)} := \{j \mid j \notin C_1^{(t)}, ..., C_K^{(t)}\}$. Compared to CLV, fewer variables contribute to the estimation of each latent variable because the conditions for contribution are stricter (see Table 3.1). As a result, some variables are not used to estimate any of the latent variables.

CLV $K+1$ (local mode) is most closely related to the problem of Orthonormal Nonnegative Sparse Principal Component Analysis (Orthonormal NSPCA). For Orthonormal NSPCA, the latent variable matrix $\mathbf{U}$ and weight matrix $\mathbf{V}$ are estimated as follows:

Orthonormal NSPCA

$$
\begin{aligned}
(\hat{\mathbf{U}}, \hat{\mathbf{V}}) = \underset{\mathbf{U},\mathbf{V}}{\arg\min}\ & \frac{1}{2(n-1)}||\mathbf{X} - \mathbf{U}\mathbf{V}^\top||_F^2 + \sum_{k=1}^{K}\sum_{j=1}^{p}\gamma_j|v_{jk}| \\
\text{s.t. } & v_{jk} \geq 0\ \forall j,k \\
& \mathbf{V}^\top\mathbf{V} = \mathbf{I}_K,
\end{aligned}
\tag{3.11}
$$

where $\gamma_j > 0$ is a sparsity hyperparameter for variable j. Due to the orthonormality and nonnegativity constraints on $\mathbf{V}$, each variable j has at most one nonzero element in $\mathbf{v}_j$. This nonzero element allows us to infer cluster assignment: $C_k := \{j \mid v_{jk} > 0\}$. While no works in the literature propose a method for solving the Orthonormal NSPCA problem, two papers propose a solution to a similar problem where exact orthonormality is sacrificed in favor of near-orthonormality (Zass and Shashua, 2007; Sigg and Buhmann, 2008).

Applying slightly different constraints to Orthonormal NSPCA results in the solution to CLV $K+1$. First, if Orthonormal NSPCA is constrained such that the latent variables $\mathbf{u}_k$ have unit variance, the objective function can be simplified using the same strategies seen in Eq. (3.8):

$$
\frac{1}{2(n-1)}||\mathbf{X} - \mathbf{U}\mathbf{V}^\top||_F^2 + \sum_{k=1}^{K}\sum_{j=1}^{p}\gamma_j|v_{jk}| = c - \sum_{k=1}^{K}\sum_{j=1}^{p}\left\{v_{jk}\text{cov}(\mathbf{u}_k, \mathbf{x}_j) - \gamma_j|v_{jk}|\right\},
\tag{3.12}
$$

where c is a constant that does not depend on $\mathbf{U}$ or $\mathbf{V}$.

The weights v_{jk} in Orthonormal NSPCA are non-binary, taking values in the range $[0, 1]$. However, if the matrix $\mathbf{V}$ is constrained to be binary with orthogonal columns, as is the case for CLV $K+1$, Orthonormal NSPCA becomes equivalent to CLV $K+1$. Let $\gamma_j = \gamma\text{sd}(\mathbf{x}_j)$. Then, $v_{jk} = 1$ if and only if $k = \arg\max_\ell\ \text{cov}(\mathbf{u}_\ell, \mathbf{x}_j)$ and

$$
\begin{aligned}
\text{cov}(\mathbf{u}_k, \mathbf{x}_j) - \gamma\text{sd}(\mathbf{x}_j) = \text{sd}(\mathbf{x}_j)(\text{cor}(\mathbf{u}_k, \mathbf{x}_j) - \gamma) > 0 \iff \\
\text{cor}(\mathbf{u}_k, \mathbf{x}_j) > \gamma.
\end{aligned}
\tag{3.13}
$$

These are the same conditions as for CLV $K+1$.

In summary, the solution to CLV $K+1$ can be rewritten as follows:

CLV $K+1$ as a Matrix Factorization Problem

$$
\begin{aligned}
(\hat{\mathbf{U}}, \hat{\mathbf{V}}) = \underset{\mathbf{U},\mathbf{V}}{\arg\min}\ & \frac{1}{2(n-1)}||\mathbf{X} - \mathbf{U}\mathbf{V}^\top||_F^2 + \gamma \sum_{k=1}^{K}\sum_{j=1}^{p} \mathrm{sd}(\mathbf{x}_j)|v_{jk}| \\
\text{s.t. } & \mathbf{V} \in \{0,1\}^{p\times K} \\
& \mathbf{V} \text{ has orthogonal columns} \\
& \mathrm{var}(\mathbf{u}_k) = 1\ \forall k
\end{aligned}
\tag{3.14}
$$

The directional mode of CLV $K+1$ is explained in Section 3.7.2.

3.3.3 CLV Sparse LV

Another method proposed by Vigneau (2016) for addressing noise variables is CLV Sparse LV, which combines elements from CLV and CLV $K+1$ (see Table 3.1). As with CLV $K+1$, CLV Sparse LV modifies the estimation of latent variables $\mathbf{u}_k$ with respect to the original CLV method. However, cluster memberships v_{jk} are estimated using the original CLV approach, meaning that there is no garbage cluster C_{K+1}.

For the estimation of latent variables, a vector of sparse loadings $\mathbf{w}_k$ is introduced so that each latent variable k is estimated using only a subset of the variables in cluster k. Each sparse loading is initialized such that $w_{jk}^{(0)} = \mathbb{I}\left[j \in C_k^{(0)}\right]$. Then, for each iteration t, the latent variables are updated such that $\mathbf{u}_k^{(t)} = \sqrt{n-1}\,\mathbf{X}\mathbf{w}_k^{(t-1)}/||\mathbf{X}\mathbf{w}_k^{(t-1)}||_2$, and the sparse loadings are updated such that

$$
w_{jk}^{(t)} = \mathbb{I}\left[k = \underset{\ell}{\arg\max}\, \mathrm{cov}(\mathbf{u}_\ell^{(t)}, \mathbf{x}_j) \text{ AND } \mathrm{cor}(\mathbf{u}_k^{(t)}, \mathbf{x}_j) > \gamma\right]. \tag{3.15}
$$

Note that the function in Eq. (3.15) is the same as the one used for the memberships $v_{jk}^{(t)}$ in CLV $K+1$. At convergence ($t = T$), the final latent variables are re-calculated using the final sparse loadings $\mathbf{w}_k^{(T)}$. As with CLV $K+1$, the threshold parameter γ is the same for each cluster, allowing less flexibility than AdaCLV.

For details concerning the directional mode of CLV Sparse LV, see Section 3.7.3.

3.3.4 Sparse Nonnegative Matrix Factorization (SNMF)

For all CLV methods outlined above, the matrix of cluster memberships $\mathbf{V}$ is both nonnegative and sparse. Therefore, the problem of variable clustering and dimensionality reduction could also be seen as a sparse nonnegative matrix factorization

	$\mathbf{x}_j$ **contributes to** $\mathbf{u}_k^{(t)}$ **if**	$v_{jk}^{(t)} > 0$ **if**
CLV	$k = \arg\max_\ell \text{cov}(\mathbf{u}_\ell^{(t-1)}, \mathbf{x}_j)$	$k = \arg\max_\ell \text{cov}(\mathbf{u}_\ell^{(t)}, \mathbf{x}_j)$
CLV Sparse LV	$k = \arg\max_\ell \text{cov}(\mathbf{u}_\ell^{(t-1)}, \mathbf{x}_j)$ AND $\text{cor}(\mathbf{u}_k^{(t-1)}, \mathbf{x}_j) > \gamma$	
CLV K + 1		$k = \arg\max_\ell \text{cov}(\mathbf{u}_\ell^{(t)}, \mathbf{x}_j)$ AND $\text{cor}(\mathbf{u}_k^{(t)}, \mathbf{x}_j) > \gamma$
AdaCLV	$\text{cov}(\mathbf{u}_k^{(t-1)}, \mathbf{x}_j) > \tau_k^{(t-1)}$	$\text{cov}(\mathbf{u}_k^{(t)}, \mathbf{x}_j) > \tau_k^{(t)}$

Table 3.1: **Latent variable and cluster membership estimation for AdaCLV and CLV, CLV $K+1$ and CLV Sparse LV (local mode)**. CLV Sparse LV is the same as CLV $K+1$ for latent variable updates (first column) and the same as CLV for cluster membership updates (second column).

(SNMF) problem. For example, for a given nonnegative matrix $\mathbf{Z}$ $(n \times p)$, the SNMF method proposed in Kim and Park (2007) ("SNMF/R") minimizes the following objective function with respect to $\mathbf{U}$ and $\mathbf{V}$:

SNMF Objective Function

$$J(\mathbf{U}, \mathbf{V}) = \frac{1}{2}||\mathbf{Z} - \mathbf{U}\mathbf{V}^\top||_F^2 + \delta \sum_{i=1}^{n} ||\mathbf{u}_i||_2^2 + \beta \sum_{j=1}^{p} ||\mathbf{v}_j||_1^2, \tag{3.16}$$

such that $\mathbf{U}$ and $\mathbf{V}$ are nonnegative. In practice, δ may be fixed to $(\max_{i,j}(z_{ij}))^2$ (Kim and Park, 2007), leaving only two hyperparameters that must be chosen: K, the number of columns in $\mathbf{U}$ and $\mathbf{V}$, and β, which controls the degree of sparsity within the vectors $\mathbf{v}_j$. As β decreases, more elements in $\mathbf{v}_j$ become non-zero, meaning that variable j can be assigned to more clusters.

Starting with an initial matrix $\mathbf{V}^{(0)}$, the matrices $\mathbf{U}$ and $\mathbf{V}$ are found using alternating optimization. As with the CLV methods, the initial matrix $\mathbf{V}^{(0)}$ can be constructed using an initial partition $C_1^{(0)}, ..., C_K^{(0)}$ by setting $v_{jk}^{(0)} = \mathbb{I}\left[j \in C_k^{(0)}\right] / \sqrt{p_k}$ or by generating a random nonnegative matrix and normalizing the columns.

Methods like AdaCLV and CLV $K+1$ encourage global sparsity in $\mathbf{V}$: increasing the hyperparameter η or γ results in clusters with fewer and fewer variables. SNMF, on the other hand, encourages sparsity within the rows of $\mathbf{V}$, thereby controlling the number of clusters to which each variable is assigned. Therefore, unlike AdaCLV and CLV $K+1$, the sparsity penalty for SNMF does not explicitly control the size of the clusters. However, like AdaCLV, SNMF can identify overlapping clusters. Indeed, the weight matrix $\mathbf{V}$ is nonnegative but does not necessarily have orthogonal columns. Thus, each vector $\mathbf{v}_j$ may have more than one nonzero element, thereby assigning variable j to multiple clusters.

3.4 Experiments

Two experiments were performed to compare the performance of the proposed method, AdaCLV, to that of CLV, CLV $K+1$ and CLV Sparse LV, as well as two baseline methods (K-means and Ward's hierarchical clustering) and the sparse non-negative matrix factorization (SNMF) method proposed in Kim and Park (2007) (see Section 3.3).

In the first experiment, these methods were applied to a real NMR dataset with non-overlapping clusters, "CitrateHippurate" (mean spectrum shown in Figure 3.1). The target clusters for this dataset —i.e. the clusters of variables that best explain the variation in this dataset —are highlighted in green and red, representing the metabolites citrate and hippurate respectively. The estimated cluster memberships were assessed visually. In the second experiment, 300 semi-artificial NMR datasets with overlapping clusters were used. For this experiment, the true cluster memberships and latent variables for the target clusters were known by design, so they were compared to the estimated memberships and latent variables for each method.

In both experiments, the true target clusters were defined as highly positively correlated variables that varied significantly between observations. For the semi-artificial dataset, these clusters were simulated, so they were known by design. For the CitrateHippurate dataset, *a priori* knowledge concerning the peak locations of the varying molecules was used to define the true clusters.

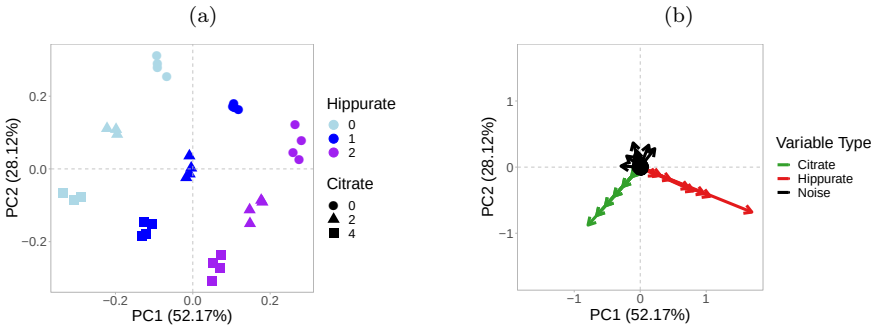
3.4.1 Data

CitrateHippurate Dataset

The CitrateHippurate dataset used here was drawn from a database of 224 ^{1}H-NMR spectra of rat urine, which is described in detail in Rousseau (2011) and Guisset, Martin, and Govaerts (2019). The samples measured for this database were collected from a homogeneous pool of rat urine and then spiked with different concentrations of two molecules: citrate and hippurate. All samples were prepared and frozen on the same day but were only thawed and measured later, at different time points. The spectral FID were pre-processed using classical preprocessing tools available in TopSpin from Bruker[1], and then all spectra were normalized to unit sum.

A subset of $n = 34$ observations from this dataset was retained for the present experiment, referred to here as the "CitrateHippurate" dataset. These observations vary according to three factors: 1) the day of measurement (day 2 or day 3), 2) the concentration of added citrate (0, 2 or 4 mM) and 3) the concentration of added hippurate (0, 1 or 2 mM). For each combination of factor levels, two replicates were

[1] https://www.bruker.com/products/mr/nmr/software/topspin.html



(c)

		Day 2			Day 3		
		Hipp mM			Hipp mM		
		0	**1**	**2**	**0**	**1**	**2**
Cit mM	**0**	2	2	2	2	2	2
	2	2	2	2	1	2	2
	4	2	2	2	1	2	2

Figure 3.3: **PCA and design for the CitrateHippurate dataset**. The PCA scores (a) and loadings (b) plots show that the first two components represent the concentration of the molecules citrate and hippurate. Moreover, most of the variability between observations (80.29%) emanates from differences in these concentrations. The number of observations for each experimental design condition is presented in (c).

performed, resulting in 36 spectra. Two observations from day 3 were detected as outliers and removed, as was done in Guisset, Martin, and Govaerts (2019). Variables in low-signal areas and the water zone (variables $1-99$, $206-354$ and $501-600$), as well as all zero-variance variables, were also removed. Finally, the remaining variables were mean-centered (except when applying SNMF).

Visualizations of the CitrateHippurate dataset can be found in Figure 3.3, as well as a table summarizing the experimental design. The PCA loadings plot (Figure 3.3(b)) for the first two principal components demonstrates that the spectral variables representing citrate (green arrows) and hippurate (red arrows) explain the most variation in the data. Moreover, the scores plot (Figure 3.3(b)) suggests that the first two dimensions of the PCA represent the concentration of hippurate (color scale from light blue to purple indicating concentrations ranging from 0 to 2 mM) and citrate (different symbols indicating concentrations ranging from 0 to 4 mM).

Because citrate and hippurate are the only molecules that vary in concentration between samples, the true number of target variable clusters in this dataset is

defined as $K_{true} = 2$, one cluster for each spiked molecule. The peaks for citrate and hippurate do not overlap each other (see Figure 3.1).

Semi-Artificial Datasets

The data used to generate semi-artificial NMR spectra came from a real ^{1}H-NMR database from the COMET project (Lindon, Nicholson, Holmes, Antti, Bollard, Keun, Beckonert, Ebbels, Reily, Robertson, et al., 2003) containing over 800 spectral FID of urine from healthy control rats. For the present experiment, 400 of these FID were selected, converted to the frequency domain and pre-processed (water suppression, apodization, baseline correction, peak alignment and bucketing), resulting in a spectral dataset, referred to here as the "COMET dataset," with $n = 400$ observations and $p = 500$ variables (see Rousseau (2011) for more details). Finally, zero-variance variables were removed, and then variance stabilizing normalization (Durbin, Hardin, Hawkins, and Rocke, 2002; Parsons, Ludwig, Günther, and Viant, 2007) was performed in order to reduce mean-dependent variance in the spectra. The latter transformation makes the variance in low intensity areas more similar to the variance in high intensity areas, making the spectra more homogeneous.

Semi-artificial spectra with overlapping peaks were generated by adding artificial peaks to spectra from the COMET dataset, which served as a realistic "background." The artificial peaks represented $M = 6$ artificial molecules, and the variables in these peaks constituted the $M = K_{true} = 6$ target clusters to be detected. The background and artificial peaks are depicted in Figure 3.4, where the first panel shows the mean spectrum of the real spectral dataset ($n = 400$), and the second panel shows the mean spectrum of the artificial peaks added to the spectra. The third panel represents the true cluster membership degrees for the artificial molecules, where each colored line represents a cluster membership vector $\mathbf{v}_k$ for a given molecule k. The height of line k for a given variable j is proportional to the degree of membership of variable j to cluster k.

Two factors were varied when generating these artificial datasets: the sample size $n \in \{25, 50, 100, 200, 400\}$ and the proportion of total spectral variability emanating from the artificial peaks (e.g. the target clusters), denoted with the parameter $\theta \in \{0.5, 0.6, 0.7\}$. This latter factor represented the relative amount of relevant signal in the dataset. For each combination of levels of these two factors, 20 datasets were generated, resulting in $5 \times 3 \times 20 = 300$ total datasets. For each dataset, n spectra were sampled randomly without replacement from the set of 400 real COMET spectra. Then, artificial peaks with randomly generated peak heights were added to each sampled COMET spectrum.

The procedure for generating artificial peaks was the following: first, for each spectrum i in the real dataset, $M = 6$ molecule concentrations were uniformly sampled from an M-dimensional simplex. Next, eight peaks (two tuples and four

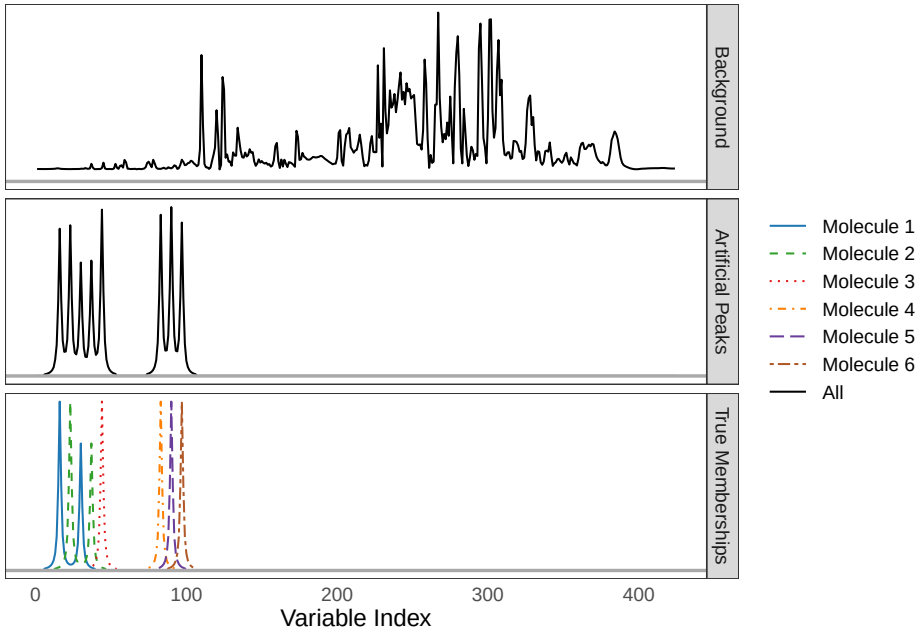


Figure 3.4: **Generation of semi-artificial spectra**. The first panel shows the mean spectrum of a real spectral dataset ($n = 400$) and the second represents the mean spectrum of the artificial peaks added to the real spectra. The third panel indicates the true cluster membership degrees for the artificial clusters, where each true membership vector $\mathbf{v}_k$ is plotted in one of six different colors and line types, and line height is proportional to membership degree.

Dataset Type	θ	n	p	$\mathbf{K_{true}}$	Datasets
CitrateHippurate Spectra	-	34	250	2	1
Semi-Artificial Spectra	$[0.5, 0.7]$	$[25, 400]$	424	6	20

Table 3.2: Characteristics of the datasets used for evaluation. Note that p represents the number of variables after removal of zero-variance variables, and K_{true} is the true number of clusters to detect. For the Semi-Artificial Spectra experiment, 20 datasets were generated for each setting (i.e. each possible combination of the sample sizes $n \in \{25, 50, 100, 200, 400\}$ and levels of signal $\theta \in \{0.5, 0.6, 0.7\}$).

singlets) were generated, with fixed peak width and with peak heights that were proportional to the randomly generated concentrations. The peak centers were evenly spaced in two regions of the spectrum, one region with five peaks and the other with three. A fully artificial spectrum $\mathbf{z}_i^{\text{art}}$ was then calculated by summing over the spectra for each artificial peak: $\mathbf{z}_i^{\text{art}} = \sum_{\ell=1}^{8} \mathbf{z}_{i\ell}^{\text{art}}$, where ℓ is the artificial peak index. More details concerning the generation of the fully artificial spectra can be found in Section 3.7.5.

Each semi-artificial spectrum $\mathbf{z}_i$ was generated by calculating the weighted sum of a real, experimental spectrum ($\mathbf{z}_i^{\text{exp}}$) and an artificial spectrum ($\mathbf{z}_i^{\text{art}}$) containing synthetic peaks (see Figure 3.4):

$$\mathbf{z}_i = \alpha_{\text{exp}} \mathbf{z}_i^{\text{exp}} + \alpha_{\text{art}} \mathbf{z}_i^{\text{art}}, \tag{3.17}$$

where α_{exp} and α_{art} are normalizing constants. These normalizing constants were used to control the proportion θ of the total spectral variability emanating from the artificial peaks (see 3.7.5 for more details). For all θ values tested here, at least half of all inter-sample variability can be attributed to the eight simulated peaks, which represent $M = 6$ artificial molecules. The $K_{true} = 6$ variable clusters to detect are thus the groups of variables in these simulated peaks. Finally, the spectra within each dataset were normalized to unit sum, and then their variables were mean-centered (except when applying SNMF).

PCA scores and loadings plots of an example dataset ($n = 400$ and $\theta = 0.70$) are shown in Figure 3.5. As seen in the loadings plots (Figure 3.5(c) - (d), (g) - (h)), the variables representing the artificial peaks, shown in colored arrows, explain the most variation in the first five (i.e. $M - 1$) principal components (variables representing background noise are shown in black). The scores plots (Figure 3.5(a) - (b), (e) - (f)) reflect the fact that the concentrations of the artificial molecules were uniformly sampled from an M-dimensional simplex. Overall, the first five dimensions of the PCA explain 70.52% of the total variability, which is very close to the expected percentage, $\theta \times 100\% = 70\%$.

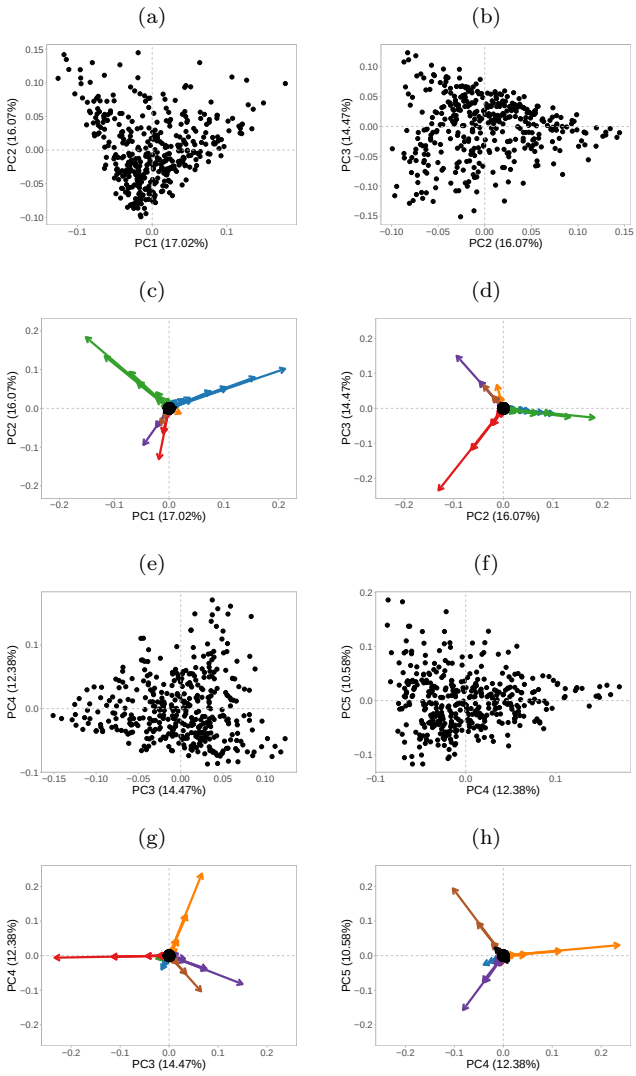


Figure 3.5: **PCA of a semi-artificial dataset**. Scores (a-b, e-f) and loadings (c-d, g-h) plots for one of the semi-artificial datasets with $n = 400$ and $\theta = 0.70$. In the loadings plots, the variables associated with each different artificial molecule are highlighted with different colors, whereas the variables representing background noise are shown in black. The variables in the artificial peaks explain the first five PCA dimensions, which amounts to about 70.52% of the total variability. This is very close to the expected percentage, $\theta \times 100\% = 70\%$.

3.4.2 Initialization and Hyperparameter Settings

With the exception of Ward's hierarchical clustering, all methods studied here require an initial partition of variable clusters. For the CitrateHippurate dataset, initial variable clusters were found using the local mode of the hierarchical algorithm proposed in Vigneau and Qannari (2003) (see Section 3.7.4), for a number of clusters K in the range $[2, 12]$. Based on our experience, this initialization strategy yields better results than random initialization. For the semi-artificial datasets, both random and non-random initializations were used. To assess overall performance and sensitivity to hyperparameters and initialization, 10 random initializations were generated for each value K in the range $[2, 12]$. Examples of typical cluster membership estimates were generated based on non-random initializations in order to demonstrate expected performance in practice. For the example memberships, the same set of non-random initial clusters was used for all datasets that shared the same level of signal θ, regardless of the sample size n. For each level of signal θ, these initial clusters were found based on one of the largest datasets ($n = 400$), using the local mode of the hierarchical algorithm proposed in Vigneau and Qannari (2003).

The distance measure used for K-means and Ward's hierarchical clustering algorithm was the Euclidean distance. For CLV $K + 1$, CLV Sparse LV and AdaCLV, a sequence of hyperparameters γ and η were tested for each dataset and number K. These hyperparameters were proportions $(0.05, 0.1, ..., 0.95)$ of the maximum hyperparameter value, $\gamma_{\max}$ or $\eta_{\max}$. For CLV $K+1$ and CLV Sparse LV, $\gamma_{\max} = 1$ by definition. For AdaCLV, $\eta_{\max}$ was determined based on the first iteration of the algorithm, as explained in Section 3.2.

The hyperparameter $\beta > 0$ in SNMF has no upper bound, making it difficult to determine a range of parameter values that span the most and least sparse solutions possible. Moreover, the publicly available implementation of SNMF in R (found in the `NMF` package (Gaujoux and Seoighe, 2010)) is up to 25 times slower than the other methods studied here, making it infeasible to empirically search for such a range for all 300 datasets, 10 random initializations and 11 cluster numbers K (33,000 total configurations). Therefore, for all experiments β was fixed to 0.01, which is the default value in the `NMF` package.

3.4.3 Calculation of Ground-Truth Matrices U and V

CitrateHippurate Dataset

For the CitrateHippurate dataset, we were interested in detecting the $K_{true} = 2$ molecules that varied significantly between samples. The true latent variable and cluster membership matrices $\mathbf{U}$ $(n \times 2)$ and $\mathbf{V}$ $(p \times 2)$ were not known by design. Rather, these matrices were approximated based on *a priori* knowledge about the location of citrate and hippurate in the spectra. Let $\tilde{C}_1$ be the set of p_1 variable

indexes associated with citrate and $\tilde{C}_2$ the set of p_2 indexes for hippurate. Let $\mathbf{Z}$ $(n \times p)$ represent the data matrix before column centering, $\overline{z}_j$ be the sample mean of variable j in $\mathbf{Z}$ and $\overline{\mathbf{z}}_k$ $(p_k \times 1)$ be the vector containing means $\overline{z}_j$ for all $j \in \tilde{C}_k$. The approximate ground-truth for cluster membership matrix $\mathbf{V}$ was calculated such that

$$v_{jk} = \mathbb{I}\left[j \in \tilde{C}_k\right] \ \overline{z}_j / ||\overline{\mathbf{z}}_k||_2. \tag{3.18}$$

The approximate ground-truth for the latent variable matrix $\mathbf{U}$ was calculated based on the membership matrix $\mathbf{V}$. Let $\mathbf{X}$ be the column-centered data matrix. The true latent variable k was calculated by normalizing the vector $\mathbf{X}\mathbf{v}_k$ $(n \times 1)$:

$$\mathbf{u}_k = \frac{\mathbf{X}\mathbf{v}_k - \text{mean}(\mathbf{X}\mathbf{v}_k)}{\text{sd}(\mathbf{X}\mathbf{v}_k)}. \tag{3.19}$$

The inner product of the matrix of latent variables (i.e. $\mathbf{U}^\top\mathbf{U}/(n-1)$) indicates that the true latent variables are not orthogonal but rather negatively correlated (correlation of -0.33). The cluster membership vectors, on the other hand, are orthogonal because the clusters are non-overlapping.

Semi-Artificial Datasets

For the Semi-Artificial datasets, we were interested in detecting the synthetic peaks, i.e. the $M = K_{true} = 6$ molecules that contributed a proportion θ of total variability between observations. The true spectral profile matrix $\mathbf{V}$ $(p \times 6)$ was defined such that each column k of this matrix was proportional to the mean spectrum for molecule k:

$$\mathbf{v}_k = \frac{\frac{1}{n}\sum_{i=1}^{n} \mathbf{z}_{ik}^{\text{art}}}{||\frac{1}{n}\sum_{i=1}^{n} \mathbf{z}_{ik}^{\text{art}}||_2} \tag{3.20}$$

The true latent variables for these datasets were calculated based on the concentration matrix $\mathbf{C}$ $(n \times 6)$ used to generate the semi-artificial spectra. Each true latent variable $\mathbf{u}_k$ was found by normalizing the corresponding column $\mathbf{c}_k$ $(n \times 1)$ in $\mathbf{C}$ as follows:

$$\mathbf{u}_k = \frac{\mathbf{c}_k - \text{mean}(\mathbf{c}_k)}{\text{sd}(\mathbf{c}_k)}. \tag{3.21}$$

As with the CitrateHippurate dataset, the true latent variables are somewhat negatively correlated (see Table 3.3). However, the true cluster membership vectors for this dataset are not exactly orthogonal (see Table 3.4) because some of the clusters are overlapping.

	1	2	3	4	5	6
1	1.00	-0.26	-0.19	-0.12	-0.22	-0.16
2	-0.26	1.00	-0.23	-0.16	-0.16	-0.15
3	-0.19	-0.23	1.00	-0.24	-0.23	-0.22
4	-0.12	-0.16	-0.24	1.00	-0.25	-0.23
5	-0.22	-0.16	-0.23	-0.25	1.00	-0.17
6	-0.16	-0.15	-0.22	-0.23	-0.17	1.00

Table 3.3: **Inner product of the ground-truth matrix U**: $\mathbf{U}^\top\mathbf{U}/(n-1)$ for a semi-artificial dataset with $\theta = 0.7$.

	1	2	3	4	5	6
1	1.00	0.11	0.00	0.00	0.00	0.00
2	0.11	1.00	0.05	0.00	0.00	0.00
3	0.00	0.05	1.00	0.00	0.00	0.00
4	0.00	0.00	0.00	1.00	0.07	0.00
5	0.00	0.00	0.00	0.07	1.00	0.07
6	0.00	0.00	0.00	0.00	0.07	1.00

Table 3.4: **Inner product of the ground-truth matrix V**: $\mathbf{V}^\top\mathbf{V}$ for a semi-artificial dataset with $\theta = 0.7$.

3.4.4 Evaluation Criteria

Let $\hat{\mathbf{V}}$ be a matrix of estimated cluster memberships and $\hat{\mathbf{U}}$ be a matrix of estimated latent variables. While both of these matrices are outputs of the CLV algorithms, they are not automatically generated for the baseline methods, K-means and Ward's hierarchical clustering. Thus, for the latter two methods, $\hat{\mathbf{U}}$ was defined as the matrix of normalized cluster centroids and $\hat{\mathbf{V}}$ as a binary matrix calculated based on the estimated variable partition $\hat{C}_1, ..., \hat{C}_K$.

For each estimated latent variable $\hat{\mathbf{u}}_k$, a sparse vector $\hat{\mathbf{r}}_k^W$ of within-cluster covariances was also calculated and stored in a $p \times K$ matrix $\hat{\mathbf{R}}^W$, where $\hat{r}_{jk}^W = \text{cov}(\hat{\mathbf{u}}_k, \mathbf{x}_j)\mathbb{I}[\hat{s}_{jk} > 0]$. The matrix $\hat{\mathbf{R}}^W$ summarizes the information contained in $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$ and makes it easier to compare AdaCLV to methods that estimate binary cluster memberships. Indeed, cluster assignments ($v_{jk} > 0$ vs $v_{jk} = 0$), rather than exact membership degrees, are used to calculate this matrix.

Let the matrix $\mathbf{Q}$ represent a ground-truth matrix ($\mathbf{U}$, $\mathbf{V}$ or $\mathbf{R}^W$) and $\hat{\mathbf{Q}}$ represent an estimated matrix ($\hat{\mathbf{U}}$, $\hat{\mathbf{V}}$ or $\hat{\mathbf{R}}^W$). In this experiment, we evaluated the *precision* of each method, i.e. the proportion of information in $\hat{\mathbf{Q}}$ that corresponds to information in $\mathbf{Q}$, as well as the *recall* of each method, i.e. the proportion of

information in $\mathbf{Q}$ that is represented in $\hat{\mathbf{Q}}$. To appreciate the importance of both measures, consider the case where the ground-truth $\mathbf{Q} = [\mathbf{a}\ \mathbf{b}\ \mathbf{c}]$, and there are two estimated matrices, $\hat{\mathbf{Q}}_1 = [\mathbf{a}\ \mathbf{a}\ \mathbf{b}]$ and $\hat{\mathbf{Q}}_2 = [\mathbf{a}\ \mathbf{b}\ \mathbf{c}\ \mathbf{d}]$. The estimated matrix $\hat{\mathbf{Q}}_1$ has perfect precision but is missing the vector $\mathbf{c}$ from the ground-truth matrix (imperfect recall). $\hat{\mathbf{Q}}_2$, on the other hand, has perfect recall but includes a vector $\mathbf{d}$ that does not appear in the ground-truth matrix (imperfect precision).

The precision and recall of an estimated matrix $\hat{\mathbf{Q}}$ was calculated as a function of the Pearson correlation between columns of the estimated and true matrices. Let $\mathbf{q}$ (resp. $\hat{\mathbf{q}}$) be a true (resp. estimated) latent variable, cluster membership or within-cluster covariance vector. A similarity function was defined as follows:

$$\mathrm{Sim}(\mathbf{q}, \hat{\mathbf{q}}) = \left[\mathrm{cor}(\mathbf{q}, \hat{\mathbf{q}}) + 1\right]/2 \in [0, 1] \tag{3.22}$$

This function evaluates to 1 if the two vectors are maximally similar, 0.5 if they are uncorrelated (perpendicular) and 0 if they are perfectly negatively correlated. If $\hat{\mathbf{q}}$ captures all of the relevant information in $\mathbf{q}$, $\mathrm{Sim}(\mathbf{q}, \hat{\mathbf{q}}) = 1$.

For the precision, each non-zero column k of the estimated matrix $\hat{\mathbf{Q}}$ was matched with the most similar column in the true matrix $\mathbf{Q}$, denoted with the index k^*, based on the similarity function in Eq. (3.22). The precision was then calculated as the average similarity for the two matrices:

$$\mathrm{Precision}(\mathbf{Q}, \hat{\mathbf{Q}}) = \frac{1}{K} \sum_{k=1}^{K} \mathrm{Sim}(\mathbf{q}_{k^*}, \hat{\mathbf{q}}_k), \tag{3.23}$$

where K is the total number of non-zero columns in $\hat{\mathbf{Q}}$ and

$$k^* = \arg\max_{\ell} \mathrm{Sim}(\mathbf{q}_\ell, \hat{\mathbf{q}}_k). \tag{3.24}$$

For matrices $\hat{\mathbf{Q}}$ with no non-zero columns, $K = 0$, so the precision is not defined.

For the recall, the similarity function was also used to match columns in the two matrices, except that each column m of the true matrix $\mathbf{Q}$ was matched with the most similar column in the estimated matrix $\hat{\mathbf{Q}}$, denoted with the index m^*. The recall was then calculated as the average similarity for the two matrices:

$$\mathrm{Recall}(\mathbf{Q}, \hat{\mathbf{Q}}) = \frac{1}{M} \sum_{m=1}^{M} \mathrm{Sim}(\mathbf{q}_m, \hat{\mathbf{q}}_{m^*}), \tag{3.25}$$

where

$$m^* = \arg\max_{\ell} \mathrm{Sim}(\mathbf{q}_m, \hat{\mathbf{q}}_\ell). \tag{3.26}$$

The precision and recall evaluate the relevance of $\hat{\mathbf{Q}}$ by measuring two distinct and potentially diverging qualities. The overall quality of $\hat{\mathbf{Q}}$ can be summarized in a single value with the F_1 Score (Jardine and Rijsbergen, 1971), which is a harmonic mean of the precision and recall, with equal weight given to each measure:

$$F_1(\mathbf{Q}, \hat{\mathbf{Q}}) = \left(\frac{\text{Precision}(\mathbf{Q}, \hat{\mathbf{Q}})^{-1} + \text{Recall}(\mathbf{Q}, \hat{\mathbf{Q}})^{-1}}{2} \right)^{-1}. \tag{3.27}$$

This score should be maximized, where the maximum value of 1 indicates perfect precision and recall.

3.4.5 Results

CitrateHippurate Dataset ($K_{true} = 2$)

For the CitrateHippurate dataset, the true latent variable matrix $\mathbf{U}$ and cluster membership matrix $\mathbf{V}$ are not known by design, so they were approximated based on *a priori* knowledge about the location of citrate and hippurate in the spectra (see Section 3.4.3 for more details). The cluster membership estimates are shown for each method in Figure 3.6. For the methods that estimate binary cluster memberships, a vertical bar is shown in color k at position j if $v_{jk} = 1$. For non-binary cluster memberships, a colored line is drawn for each vector $\mathbf{v}_k$, where the line height is proportional to membership degree. For AdaCLV, CLV $K + 1$ and CLV Sparse LV, only results for the optimal hyperparameter γ or η are shown here. For each number K of clusters, the optimal γ or η for these methods was chosen based on the F_1 Score of the matrix of within-cluster covariances, $\hat{\mathbf{R}}^W$. As explained in Section 3.4.4, the matrix $\hat{\mathbf{R}}^W$ encodes information contained in both $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$.

As seen in Figure 3.6, the only methods that provide satisfactory estimates of the true cluster memberships for $K = 2$ are AdaCLV and CLV $K+1$, the methods that account for "garbage" variables. Both methods incorrectly assign some irrelevant variables to signal clusters (i.e. generate false positives). However, the membership degrees for AdaCLV make it possible to interpret these false positives with caution: small v_{jk} indicate low importance or a low likelihood of truly belonging to the cluster. In contrast, the binary memberships estimated by CLV $K + 1$ do not provide this useful information, making misinterpretation of false positives more likely.

CLV Sparse LV, CLV, Ward's Hierarchical clustering and K-means, which do not make use of a garbage cluster, achieve better performance when $K = 3$, as a third cluster can be used to group irrelevant variables. While Ward's Hierarchical clustering and K-means seem to perform better overall than CLV Sparse LV and CLV, all of these methods are still subject to error: Ward's Hierarchical clustering

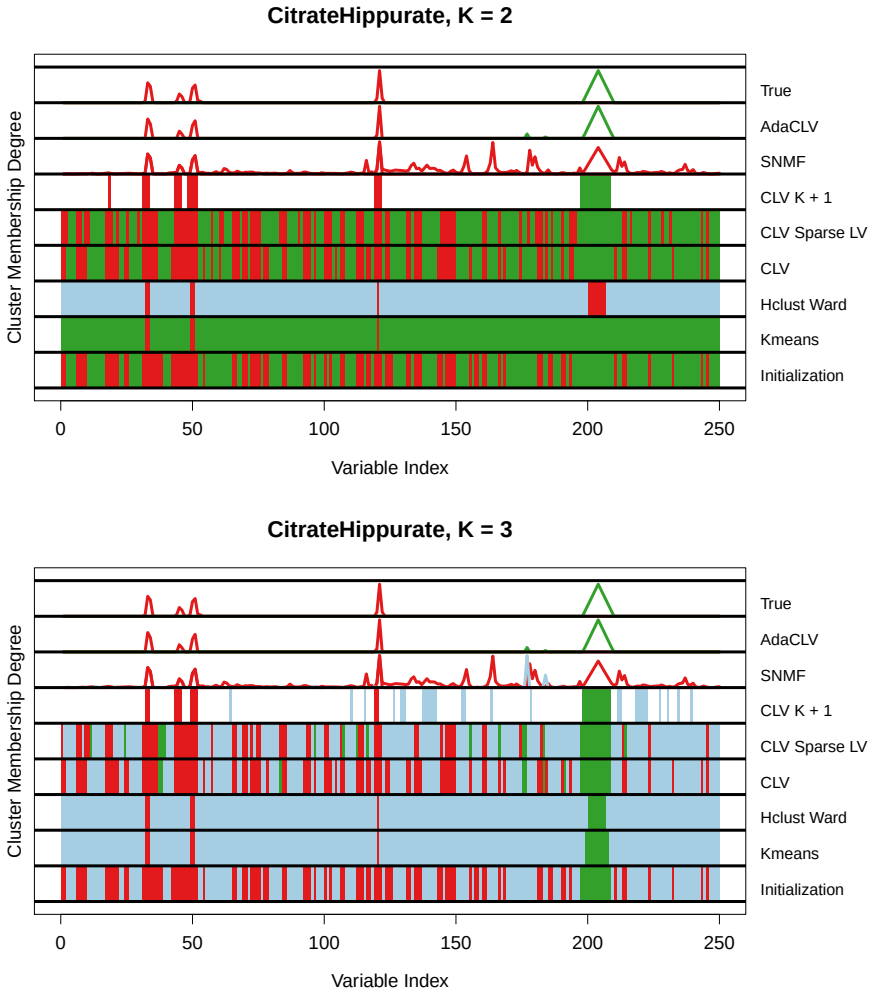


Figure 3.6: **Optimal memberships for the CitrateHippurate dataset and for $K = K_{true} = 2$ and $K = 3$.** For binary memberships v_{jk}, the vertical bar at variable index j is shown in color k if $v_{jk} = 1$. White spaces correspond to variables that are assigned to none of the clusters. For non-binary memberships, each membership vector $\mathbf{v}_k$ is plotted in a different color, where line height is proportional to membership degree.

and K-means miss one of the peaks for hippurate (in red) and CLV Sparse LV and CLV assign many irrelevant variables to the citrate and hippurate clusters (in red and green). For $K = 3$, AdaCLV generates the same cluster memberships as for $K = 2$ (the third cluster is empty), suggesting that AdaCLV is relatively robust to the hyperparameter K. In contrast, CLV $K + 1$ erroneously assigns irrelevant variables to a third cluster when $K = 3$.

Figure 3.7 depicts the cluster membership vectors $\hat{\mathbf{v}}_k$ estimated by SNMF for $K = 2$ and different values of the hyperparameter β. Each colored line represents a different cluster k. For the default hyperparameter $\beta = 0.01$ and an initialization with $K = 2$ clusters, SNMF identifies two clusters, but the variables in the smaller cluster (red line) also belong to the larger cluster (green line). Moreover, SNMF assigns both irrelevant and relevant variables to the larger cluster. For the initialization with $K = 3$, an additional cluster is identified but this cluster only contains irrelevant variables. The weaker performance of SNMF can be explained by the fact that the final number of clusters and the number of variables in each cluster both depend on a single hyperparameter, making this method less flexible. As seen in Figure 3.7, when β decreases from 1 to 0.01, an additional cluster is found (shown in red). As β continues to decrease, this red cluster acquires more and more variables. Moreover, regardless of the value of β, SNMF tends to find at least one cluster (shown in green) to which almost all variables are assigned. This large cluster does not provide much added value in terms of interpretation.

Semi-Artificial Datasets ($K_{true} = 6$)

For the Semi-Artificial spectra, 20 datasets were generated for each sample size ($n \in \{25, 50, 100, 200, 400\}$) and proportion of artificial variability ($\theta \in \{0.5, 0.6, 0.7\}$), so the average performance of the different methods could be calculated and compared. Moreover, the true latent variable matrix $\mathbf{U}$ and cluster membership matrix $\mathbf{V}$ were known by design, making it possible to compare results to the ground-truth with numerical criteria.

As with the CitrateHippurate dataset, the following results are for the optimal hyperparameter settings for each dataset and value K. For all θ values, the ranking of the different methods is similar for all evaluation criteria (results not shown here). All methods generally benefit from higher values of θ (i.e. higher signal), especially for small sample sizes. However, for cluster membership estimation, increasing the amount of signal in the data does not improve the precision or recall of CLV or CLV Sparse LV.

For $\theta = 0.7$, the performance results for latent variable estimation ($\hat{\mathbf{U}}$ vs $\mathbf{U}$) are presented in Figure 3.9, for cluster membership estimation ($\hat{\mathbf{V}}$ vs $\mathbf{V}$) in Figure 3.10 and for within-cluster covariance ($\hat{\mathbf{R}}^W$ vs $\mathbf{R}^W$) in Figure 3.11. For each initialization and sample size n, the precision and recall were summarized by calculating a mean over 20 repetitions. The median precision and recall is plotted with respect

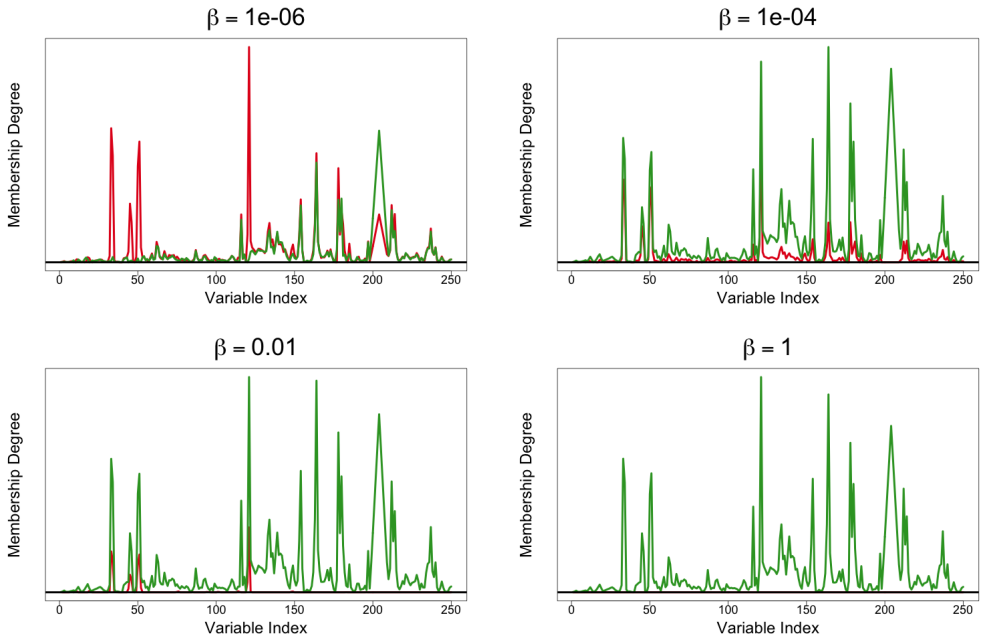


Figure 3.7: **Examples of SNMF memberships for different** β. Cluster membership estimation for $K = K_{true} = 2$.

to K for three of the five sample sizes n, where each method is represented with a different color and line type. The vertical bars represent the interquartile range of performance for 10 different random initializations. The true number of target clusters, $K_{true} = 6$, is indicated with a vertical gray line.

For all K and n tested, AdaCLV has equivalent or better precision than the other methods and comparable recall for latent variable and cluster membership estimation, as well as within-cluster covariance. AdaCLV's recall, while comparable to other methods, is slightly weaker because it tends to under-select variables. Indeed, the algorithm for Irrelevant Variable Identification (IVI, see Section 3.2.2) sets the sparsity threshold τ_k based on the distribution of covariances for irrelevant variables. If this distribution overlaps with the distribution for signal variables, then some signal variables will be removed from the analysis in error. This is demonstrated with a toy example in Figure 3.8. For a given cluster k, a histogram of covariances with the true latent variable $\mathbf{u}_k$ is plotted for irrelevant or "garbage" variables (left) and relevant or "signal" variables (right). The sparsity threshold τ_k is defined based on the maximum covariance for irrelevant variables, which is indicated with a vertical green bar. Certain signal variables have covariances below

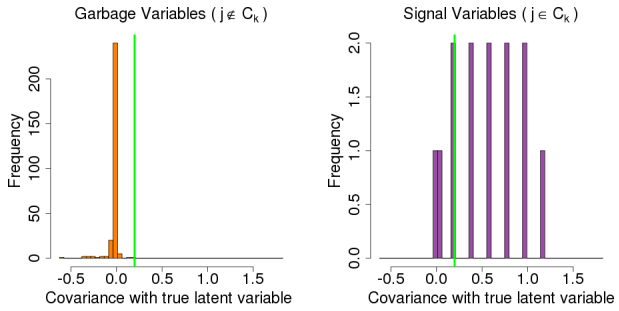


Figure 3.8: **Toy example demonstrating variable under-selection**. For a given cluster k, a histogram of covariances with the true latent variable $\mathbf{u}_k$ is plotted for irrelevant or "garbage" variables (left) and relevant or "signal" variables (right). The sparsity threshold τ_k in AdaCLV's Irrelevant Variable Identification (IVI) algorithm is defined based on the maximum covariance for irrelevant variables, which is indicated with a vertical green bar. Certain signal variables have covariances below this threshold and will thus be removed from the analysis in error.

this threshold and thus will not be selected. However, these variables will generally be the least important of the signal variables, so their loss is less concerning.

For cluster membership estimation, SNMF is the only method with a precision comparable to AdaCLV, but this occurs for $K = K_{true} = 6$ only. For $K > K_{true}$, the precision of SNMF decreases because it adds irrelevant clusters. As was observed for the CitrateHippurate dataset, the worst precision and recall for cluster membership estimation is obtained by CLV and CLV Sparse LV. CLV and Ward's Hierarchical clustering have the worst latent variable estimation performance overall.

Examples of cluster membership estimation for the different methods are shown in Figures 3.12, 3.13 and 3.14 for non-random initializations with $K = 6, 7$ and 10 and $\theta = 0.7$. For each method, the cluster memberships shown here are the ones that best represent the mean performance of the method in terms of within-cluster covariance. AdaCLV estimates memberships that are very close to the true memberships for $K = 6, 7$ and 10, further demonstrating its robustness to the choice of the hyperparameter K. Indeed, when $K > K_{true}$, AdaCLV tends to find redundant clusters. For example, for $n = 400$ and $K = 7$, cluster 3 is found twice. For $K = 10$, cluster 3 is found three times and clusters 4 and 6 are found twice. This redundancy property is advantageous because it means that K can be overestimated without significant negative consequences.

In extreme settings where n is much smaller than p, making it difficult to distin-

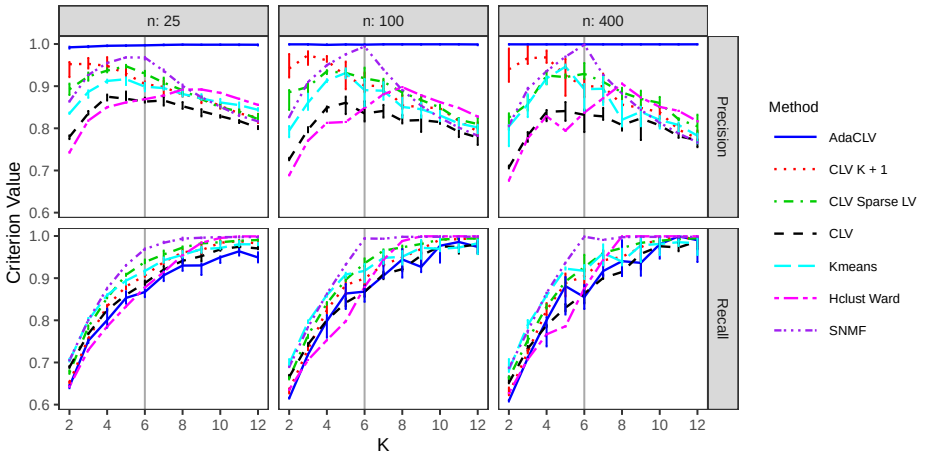


Figure 3.9: **Performance for latent variable estimation**. Precision$(\mathbf{U}, \hat{\mathbf{U}})$ and Recall$(\mathbf{U}, \hat{\mathbf{U}})$ for $\theta = 0.7$ and for different n and K, averaged over 20 repetitions. The vertical bars indicate the interquartile range for 10 random initializations.

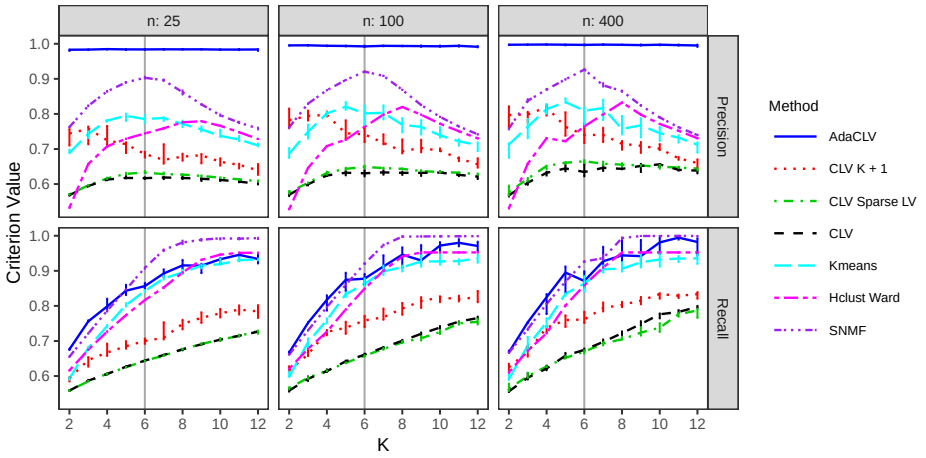


Figure 3.10: **Performance for cluster membership estimation**. Precision$(\mathbf{V}, \hat{\mathbf{V}})$ and Recall$(\mathbf{V}, \hat{\mathbf{V}})$ for $\theta = 0.7$ and for different n and K, averaged over 20 repetitions. The vertical bars indicate the interquartile range for 10 random initializations.

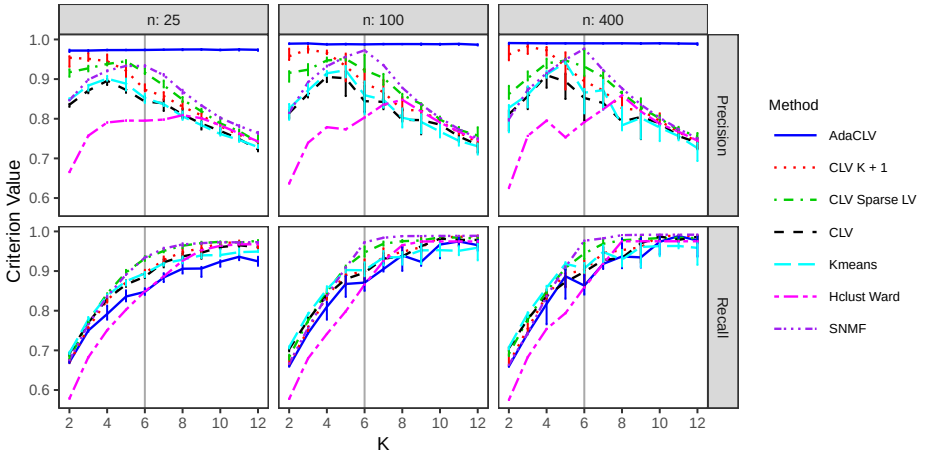


Figure 3.11: **Performance for within-cluster covariance**. Precision($\mathbf{R}^W, \hat{\mathbf{R}}^W$) and Recall($\mathbf{R}^W, \hat{\mathbf{R}}^W$) for $\theta = 0.7$ and for different n and K, averaged over 20 repetitions. The vertical bars indicate the interquartile range for 10 random initializations.

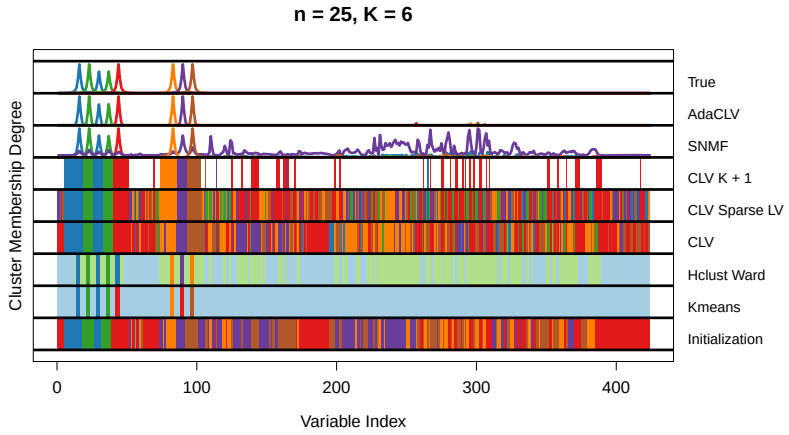
guish irrelevant variables from variables containing signal, all methods assign more irrelevant variables to signal clusters. This effect is much less pronounced for AdaCLV, for which only a few irrelevant variables are assigned to signal clusters. In addition, AdaCLV's low membership degrees for these variables indicate that they are less likely to be true members of these clusters. This is advantageous because it makes it possible to distinguish potential false positives from true positives.

When the sample size n is large, AdaCLV generates fewer false positives and is able to detect more overlap between clusters. CLV $K + 1$, SNMF and K-means also benefit from larger sample sizes, but their estimation of cluster memberships varies considerably with K. Indeed, they only perform comparably to AdaCLV when $n = 400$ and K is correctly chosen ($K = 6$, 7 and 7 respectively). Both SNMF and K-means assign the irrelevant variables to a separate cluster when $K = 7$, but for SNMF, this irrelevant cluster also includes all of the signal variables, making it less easily interpretable. When n increases, the purity of the irrelevant cluster increases for K-means but not for SNMF.

Finally, unlike AdaCLV, SNMF and CLV $K+1$, K-means and Ward's Hierarchical clustering do not assign variables in overlapping peak regions to signal clusters. This suggests that these two methods are less well suited to datasets with overlapping clusters.

For CLV $K+1$, CLV Sparse LV and AdaCLV, the results presented above were for

(a)



(b)

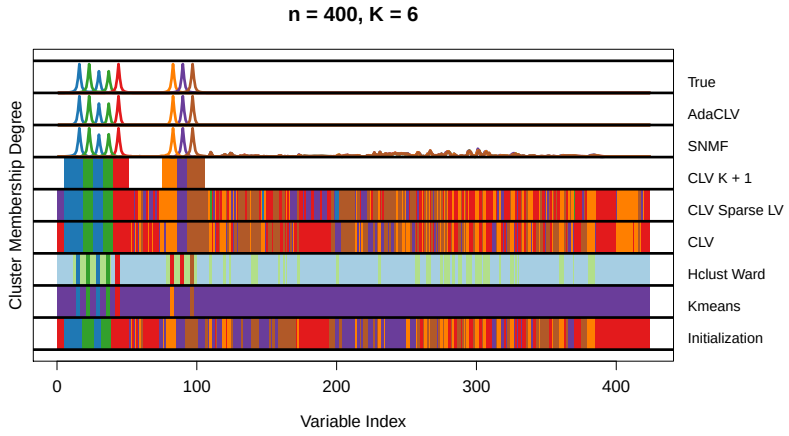
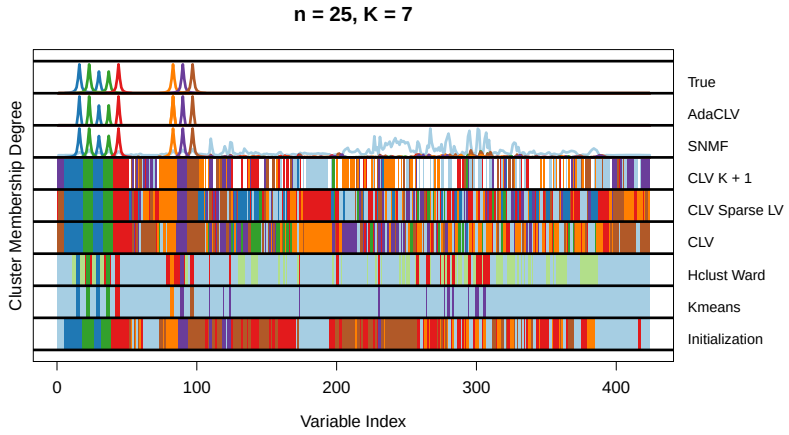


Figure 3.12: **Examples of optimal memberships for semi-artificial datasets with $\theta = 0.70$ and $K = 6$.** Cluster membership estimation for $n = 25$ (3.12a) and $n = 400$ (3.12b). These plots may be interpreted in the same way as for those in Figure 3.6.

(a)



(b)

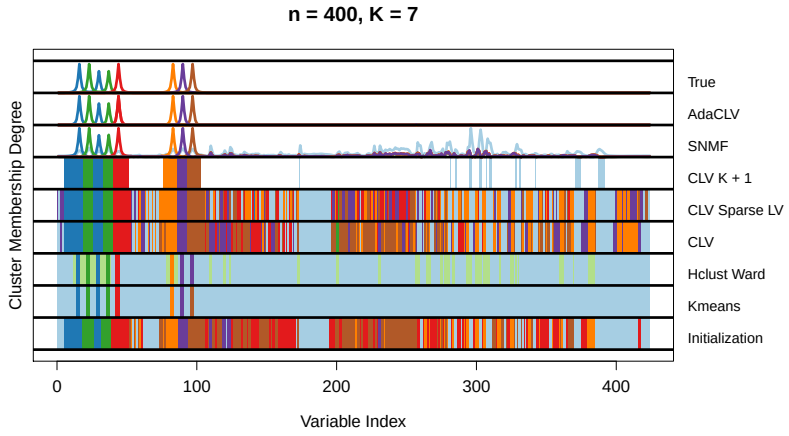
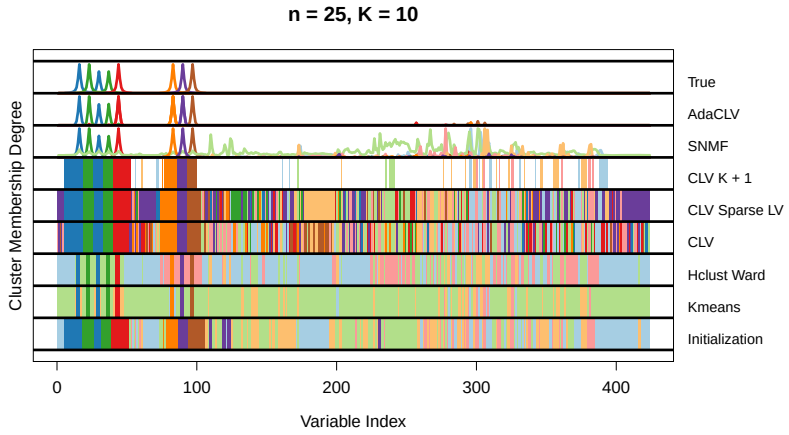


Figure 3.13: **Examples of optimal memberships for semi-artificial datasets with $\theta = 0.70$ and $K = 7$.** Cluster membership estimation for $n = 25$ (3.13a) and $n = 400$ (3.13b). These plots may be interpreted in the same way as for those in Figure 3.6.

(a)



(b)

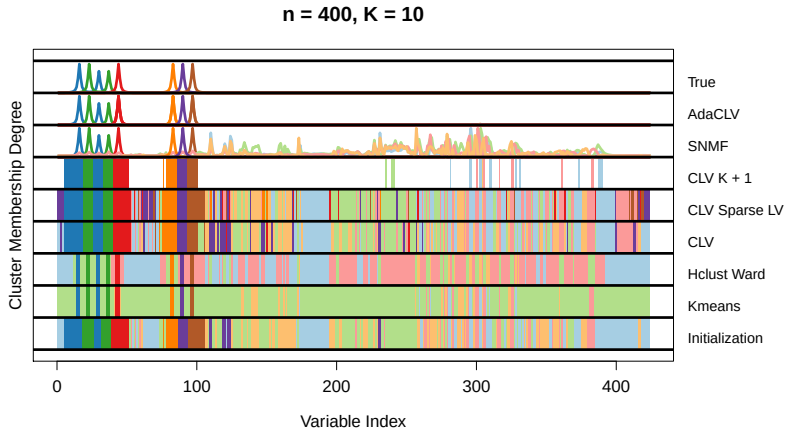


Figure 3.14: **Examples of optimal memberships for semi-artificial datasets with $\theta = 0.70$ and $K = 10$.** Cluster membership estimation for $n = 25$ (3.14a) and $n = 400$ (3.14b). These plots may be interpreted in the same way as for those in Figure 3.6.

the optimal choice of hyperparameters γ and η and for non-random initializations. To explore the effect of initialization and hyperparameter choice on performance, the average F_1 scores for latent variable, cluster membership and within-cluster covariance estimation were calculated for each hyperparameter, initialization and sample size n. The median and interquartile range of these averages were then plotted with respect to the hyperparameters (see the figures in Section 3.7.6, which represent the datasets with $\theta = 0.70$). Overall, AdaCLV is least sensitive to η and initialization for cluster membership estimation. For all n and $K \in [4, 8]$, η values in the range $[0.05, 0.75] \times \eta_{\max}$ yield cluster membership estimates that are better than those for the other two methods. AdaCLV is most sensitive to the choice of η for latent variable estimation, but the other two methods are equally sensitive for this task. Based on these experiments, η values in the range $[0.05, 0.65] \times \eta_{\max}$ generally seem to yield good results. In particular, the value $\eta = 0.5 \times \eta_{\max}$ is a reasonable choice in all cases tested and could be used as a default value.

3.5 Discussion and Conclusions

This chapter presented a new method for simultaneous dimensionality reduction and variable clustering, Adaptive Clustering around Latent Variables (AdaCLV). This method can facilitate the analysis of spectroscopic datasets, such as Nuclear Magnetic Resonance (NMR) spectra, which suffer from high dimensionality and complex correlation structures. Using a data-adaptive threshold, AdaCLV filters out variables that do not vary significantly between observations (i.e. irrelevant variables). For the remaining variables, cluster membership degrees are estimated, making it possible to identify potentially overlapping clusters and to rank the variables within a cluster in order of importance. Latent variables are estimated based on these membership degrees, summarizing the clusters in a low-dimensional representation that can be used for other tasks, such as classification. AdaCLV's ability to adapt to clustering structures present in the data makes it a candidate pre-processing technique to be used in pipelines for the discovery of new biomarkers.

Several numerical experiments were conducted to compare AdaCLV to existing CLV methods (CLV, CLV $K + 1$ and CLV Sparse LV), as well as two baseline clustering methods (K-means and Ward's hierarchical clustering) and Sparse Nonnegative Matrix Factorization (SNMF). For NMR spectra with overlapping and non-overlapping clusters, and for all sample sizes and levels of signal tested, AdaCLV estimated latent variables and cluster memberships with greater or equivalent precision with respect to the other methods. This was the case for all values tested for the hyperparameter K, the number of clusters to detect. In terms of recall, AdaCLV performed comparably to the other methods for all datasets. In contrast to the CLV methods, neither of the baseline clustering methods assigned variables in regions of peak overlap to signal clusters, making these methods less attractive

for datasets with overlapping clusters. While CLV $K+1$, SNMF and K-means were able to estimate fairly accurate cluster memberships for some datasets, this depended on the sample size and the careful selection of hyperparameters. AdaCLV, on the other hand, was able to estimate more interpretable and representative cluster memberships for a variety of hyperparameter choices, sample sizes and levels of signal, suggesting that it is more robust in practice.

AdaCLV is an unsupervised method but it could be integrated into a supervised learning framework, where a categorical variable (e.g. patient status) is predicted (see Chapter 4). As mentioned previously, AdaCLV could be used in a data analysis pipeline involving two steps, where AdaCLV estimates latent variables in the first step, then these latent variables are used to predict a response variable in the second step (see Bühlmann, Rütimann, Geer, and Zhang (2013)). Another two-step approach would be to use the clusters identified by AdaCLV in a Group-Lasso method built for overlapping groups (Jacob, Obozinski, and Vert, 2009). Finally, rather than including AdaCLV in a two-step pipeline, it could be directly integrated into the objective function of a classification method so that the variable clustering would be supervised by the response. Future works could explore these different approaches in order to identify the best strategy in terms of both prediction and clustering quality.

3.6 Numerical Implementation

The experiments from this chapter were programmed in R. Adaptive Clustering around Latent Variables (AdaCLV) was implemented using in-house code that is publicly available on GitHub at https://github.com/rebeccamarion/AdaCLV. Clustering around Latent Variables (CLV), CLV Sparse LV and CLV $K+1$ were performed using the *CLV_kmeans* function from the `ClustVarLV` package. Sparse Nonnegative Matrix Factorization (SNMF) was performed using the function *nmf* from the `NMF` package. Ward's hierarchical clustering and K-means clustering were performed using the functions *hclust* and *kmeans* from the `stats` package. Cluster initializations were generated using the function *CLV* from the `ClustVarLV` package.

3.7 Appendix

3.7.1 CLV Directional Mode

The local mode for CLV was explained in Section 3.3.1. This appendix summarizes the directional mode, where covariances are replaced with squared covariances.

The matrices $\mathbf{U}$ and $\mathbf{V}$ are found by maximizing the function

$$\begin{aligned} J(\mathbf{U}, \mathbf{V}) =& \sum_{k=1}^{K} \sum_{j=1}^{p} v_{jk} \text{cov}^2(\mathbf{u}_k, \mathbf{x}_j) \\ =& \frac{1}{(n-1)^2} \sum_{k=1}^{K} (\mathbf{u}_k^\top \mathbf{X} \mathbf{v}_k)^2 \end{aligned} \tag{3.28}$$

with respect to $\mathbf{U}$ and $\mathbf{V}$, subject to the same constraints as the local mode. The solution for each latent variable $\mathbf{u}_k$ is the first standardized principal component of $\mathbf{X}\text{diag}(\mathbf{v}_k)$, and the optimal cluster k is now defined based on squared covariances: $C_k := \{j \mid k = \arg\max_\ell \text{cov}^2(\mathbf{u}_\ell, \mathbf{x}_j)\}$.

3.7.2 CLV $K+1$ Directional Mode

The local mode for CLV $K+1$ was described in Section 3.3.2. This appendix summarizes the directional mode, where covariances are replaced with squared covariances and $\text{sd}(\mathbf{x}_j)$ and the hyperparameter γ are squared.

The matrices $\mathbf{U}$ and $\mathbf{V}$ are found by maximizing the function

$$J(\mathbf{U}, \mathbf{V}) = \sum_{j=1}^{p} \left\{ \sum_{k=1}^{K} v_{jk} \text{cov}^2(\mathbf{u}_k, \mathbf{x}_j) + \gamma^2 \left(1 - \sum_{k=1}^{K} v_{jk} \right) \text{var}(\mathbf{x}_j) \right\} \tag{3.29}$$

with respect to $\mathbf{U}$ and $\mathbf{V}$, subject to the same constraints as the local mode. The solution for each latent variable $\mathbf{u}_k$ is the same as for CLV (see 3.7.1), and the optimal cluster k is now defined based on squared covariances and correlations: $C_k := \{j \mid k = \arg\max_\ell \text{cov}^2(\mathbf{u}_\ell, \mathbf{x}_j) \text{ AND } \text{cor}^2(\mathbf{u}_k, \mathbf{x}_j) > \gamma^2\}$.

3.7.3 CLV Sparse LV Directional Mode

The local mode for CLV Sparse LV was summarized in Section 3.3.3. For the directional mode and for fixed $\mathbf{U}$, the matrix $\mathbf{V}$ is found by maximizing the original CLV function with respect to $\mathbf{V}$, subject to the same constraints as the local mode. For fixed $\mathbf{V}$, the optimal latent variable $\mathbf{u}_k$ is estimated using a sparse loadings vector $\mathbf{b}_k$, which is calculated according to a soft-thresholding function. For more details about this procedure, see Vigneau (2016).

3.7.4 Initialization for CLV Local Mode

All existing CLV algorithms require the input of an initial partition of variables, which means that a variable clustering algorithm must be run prior to performing CLV. Vigneau and Qannari (2003) proposed a hierarchical agglomerative clustering algorithm for finding an initialization for the local and directional modes of CLV.

At each iteration, this algorithm merges the pair of clusters that incurs the smallest added cost with respect to the CLV objective function.

Let $\mathbf{X}_A$ $(n \times p_A)$ (resp. $\mathbf{X}_B$ $(n \times p_B)$) be the matrix of variables in $\mathbf{X}$ that belong to cluster A (resp. B) and $\overline{\mathbf{x}}_A$ (resp. $\overline{\mathbf{x}}_B$) be an $n \times 1$ vector containing the means of each row in $\mathbf{X}_A$ (resp. $\mathbf{X}_B$). For the local mode of CLV, the added cost of merging two clusters A and B, with respect to the CLV objective function, is

$$\Delta = p_A \text{ sd}(\overline{\mathbf{x}}_A) + p_B \text{ sd}(\overline{\mathbf{x}}_B) - p_{A\cup B} \text{ sd}(\overline{\mathbf{x}}_{A\cup B}). \tag{3.30}$$

Starting from $K = p$ clusters, clusters are merged based on the added cost function Δ until only one cluster remains, resulting in a sequence of partitions with $K = p, ..., 1$ cluster(s). Each value of K is thus associated with a unique partition from this hierarchical sequence of partitions. The user must choose one of these as the initial partition for CLV, i.e. choose the number K of variable clusters that should be detected in the dataset.

This cost function in Eq. (3.30) is motivated as follows. Let $\mathbf{X}_k$ $(n \times p_k)$ be the matrix of variables in $\mathbf{X}$ that belong to cluster k, and let $\mathbf{x}_{jk}$ be column j in $\mathbf{X}_k$. Moreover, let $\overline{\mathbf{x}}_k$ $(n \times 1)$ be a vector containing the means of each row in $\mathbf{X}_k$. For fixed clusters $C_1, ..., C_K$, the local criterion in Eq. (3.6) can be rewritten as follows:

$$\begin{aligned} J(\mathbf{U}) =& \frac{1}{n-1} \sum_{k=1}^{K} \sum_{j=1}^{p_k} \mathbf{u}_k^\top \mathbf{x}_{jk} \\ =& \frac{1}{n-1} \sum_{k=1}^{K} p_k \mathbf{u}_k^\top \left(\frac{1}{p_k} \sum_{j=1}^{p_k} \mathbf{x}_{jk} \right) \\ =& \frac{1}{n-1} \sum_{k=1}^{K} p_k \mathbf{u}_k^\top \overline{\mathbf{x}}_k. \end{aligned} \tag{3.31}$$

For a given k, the unit-variance latent variable $\mathbf{u}_k$ that maximizes this function is

$$\mathbf{u}_k = \sqrt{n-1} \, \frac{\overline{\mathbf{x}}_k}{||\overline{\mathbf{x}}_k||_2}. \tag{3.32}$$

Thus, the maximum of the local criterion can be rewritten as follows:

$$\begin{aligned}\max_{\mathbf{U}} J(\mathbf{U}) =& \frac{1}{n-1}\sum_{k=1}^{K} p_k \left(\sqrt{n-1}\ \frac{\overline{\mathbf{x}}_k}{||\overline{\mathbf{x}}_k||_2}\right)^\top \overline{\mathbf{x}}_k \\ =& \sum_{k=1}^{K} \frac{p_k}{\sqrt{n-1}} ||\overline{\mathbf{x}}_k||_2 \\ =& \sum_{k=1}^{K} p_k \mathrm{sd}(\overline{\mathbf{x}}_k).\end{aligned} \tag{3.33}$$

The added cost of merging two clusters A and B, with respect to this criterion, is then

$$\Delta = p_A\ \mathrm{sd}(\overline{\mathbf{x}}_A) + p_B\ \mathrm{sd}(\overline{\mathbf{x}}_B) - p_{A\cup B}\ \mathrm{sd}(\overline{\mathbf{x}}_{A\cup B}). \tag{3.34}$$

3.7.5 Data Generation for Semi-Artificial Spectra

The procedure for generating the Semi-Artificial datasets was briefly outlined in Section 3.4.1. This appendix provides further details about this procedure.

For each repetition $r : 1, ..., 20$, sample size $n : 25, 50, 100, 200, 400$ and signal level $\theta : 0.5, 0.6, 0.7$, spectral vectors $\mathbf{z}_i^{\mathrm{art}}$ with artificial peaks were generated as follows. $M = 6$ molecule concentrations c_{im} were uniformly sampled from an M-dimensional simplex, then 8 peaks were generated based on these concentrations (2 tuples and 4 singlets). The peak heights were set to

$$\mathbf{h}_i = (c_{i1},\ c_{i2},\ c_{i1} \times 0.75,\ c_{i2} \times 0.75,\ c_{i3},\ c_{i4},\ c_{i5},\ c_{i6}).$$

Peak centers $\nu_{0i\ell}$ were chosen so that evenly-spaced peaks appeared in two regions with five and three peaks respectively. Intensities $z_{ij\ell}^{\mathrm{art}}$ for each peak ℓ, observation i and frequency ν_j were calculated using a Lorentzian function, as was done in Cloarec, Dumas, Trygg, Craig, Barton, Lindon, Nicholson, and Holmes (2005):

$$z_{ij\ell}^{\mathrm{art}} = \frac{h_{i\ell} w^2}{w^2 + 4(\nu_j - \nu_{0i\ell})^2}, \tag{3.35}$$

where w is a fixed peak width. Small intensities ($z_{ij\ell}^{\mathrm{art}} < 0.01 h_{i\ell}$) were then set to zero. The pure spectrum for a given molecule m, $\mathbf{z}_{im}^{\mathrm{art}}$, was calculated as the sum of all peak spectra associated with molecule m:

$$\mathbf{z}_{im}^{\mathrm{art}} = \sum_{\ell \in \mathrm{mol}\ m} \mathbf{z}_{i\ell}^{\mathrm{art}}. \tag{3.36}$$

The pure spectrum $\mathbf{z}_i^{\mathrm{art}}$ was calculated as a sum over all molecules m:

$$\mathbf{z}_i^{\mathrm{art}} = \sum_{m=1}^{M} \mathbf{z}_{im}^{\mathrm{art}}. \tag{3.37}$$

Let $\mathbf{z}_i^{\text{exp}}$ be the experimental spectrum for a given observation i and $\mathbf{z}_i^{\text{art}}$ be a spectrum containing artificial peaks. Each final spectrum $\mathbf{z}_i$ was generated by adding these two spectra,

$$\mathbf{z}_i = \alpha_{\text{exp}}\mathbf{z}_i^{\text{exp}} + \alpha_{\text{art}}\mathbf{z}_i^{\text{art}}, \tag{3.38}$$

where α_{exp} and α_{art} are normalizing constants. These normalizing constants were used to control the proportion θ of the total spectral variability emanating from the artificial peaks. If $\alpha_{\text{exp}} = \sqrt{1-\theta}/\sqrt{\sum_{j=1}^{p}\sigma_{j,\text{exp}}^2}$ and $\alpha_{\text{art}} = \sqrt{\theta}/\sqrt{\sum_{j=1}^{p}\sigma_{j,\text{art}}^2}$, then the total population variance of the final set of spectra, $\sum_{j=1}^{p}\sigma_j^2$, is equal to 1, where a fraction θ of this variance comes from the artificial spectra.

$$\begin{aligned}\sum_{j=1}^{p}\sigma_j^2 &= \alpha_{\text{exp}}^2\sum_{j=1}^{p}\sigma_{j,\text{exp}}^2 &&+ \alpha_{\text{art}}^2\sum_{j=1}^{p}\sigma_{j,\text{art}}^2\\ &= 1-\theta &&+ \theta\\ &= 1\end{aligned} \tag{3.39}$$

In practice, α_{exp} and α_{art} were calculated using sample variances $\hat{\sigma}_{j,\text{exp}}^2$ and $\hat{\sigma}_{j,\text{art}}^2$.

3.7.6 Sensitivity Analysis for the Semi-Artificial Datasets

The overall performance (F_1 Score) of AdaCLV, CLV $K+1$ and CLV Sparse LV for latent variable estimation (Figure 3.15), cluster membership estimation (Figure 3.16) and within-cluster covariance (Figure 3.17) is presented here for $\theta = 0.7$. The mean performance was calculated for each method, initialization and hyperparameter setting (a mean over 20 repetitions). The median and interquartile range of performance for 10 random initializations is plotted with respect to the value of the hyperparameter γ (for CLV $K+1$ and CLV Sparse LV) or η (for AdaCLV). When no clusters are found for a given hyperparameter setting, the corresponding F_1 score is undefined and thus does not appear in the plot.

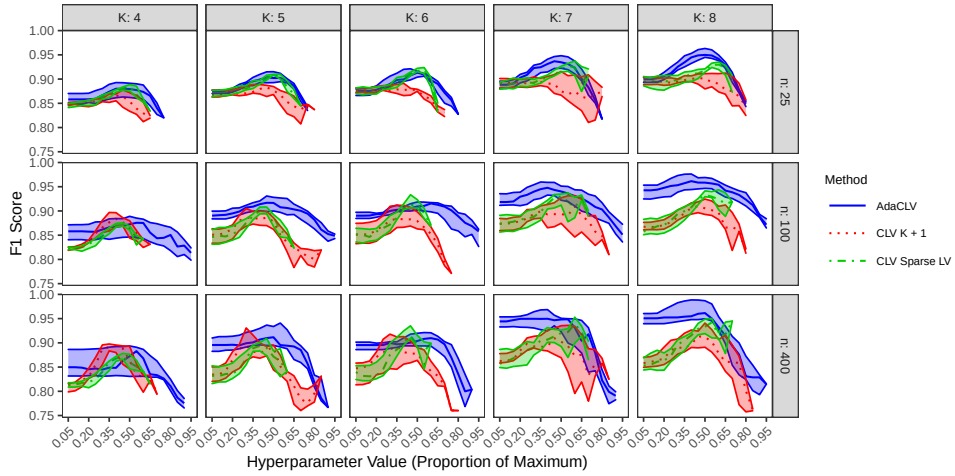


Figure 3.15: **Sensitivity for latent variable estimation**. F_1 score for $\theta = 0.7$ and for different n and hyperparameter values, averaged over 20 repetitions. The interquartile range for 10 random initializations is shaded in color.

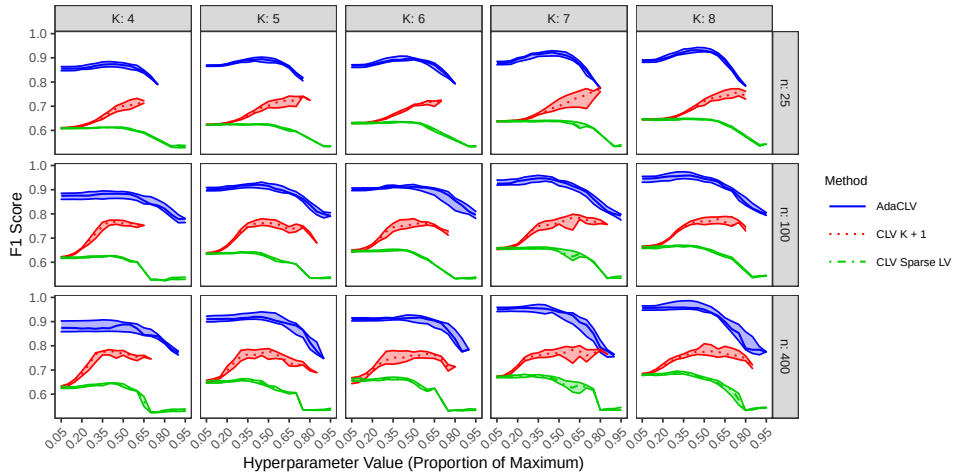


Figure 3.16: **Sensitivity for cluster membership estimation**. F_1 score for $\theta = 0.7$ and for different n and hyperparameter values, averaged over 20 repetitions. The interquartile range for 10 random initializations is shaded in color.

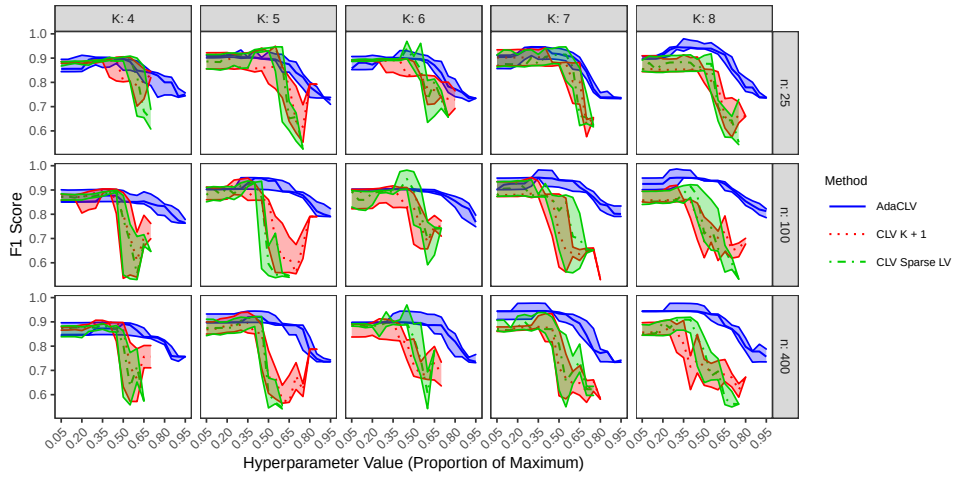


Figure 3.17: **Sensitivity for within-cluster covariance**. F_1 score for $\theta = 0.7$ and for different n and hyperparameter values, averaged over 20 repetitions. The interquartile range for 10 random initializations is shaded in color.

Chapter 4

Variable Clustering Supervised by a Univariate Target

This chapter describes Variable Cluster Principal Component Regression (VC-PCR), a prediction method that accounts for clustering structure among predictors (Marion, Lederer, Govaerts, and von Sachs, 2021) (to be submitted). In addition to predicting a continuous or binary target (e.g. cancer status), VC-PCR clusters and selects variables according to this target. This is particularly useful in the study of complex diseases, for which clusters of variables (e.g. genes) are more likely to contribute to disease than individual variables. As a two-step method involving a supervised variable clustering step, VC-PCR draws on the methodological strengths of two major approaches in the literature: two-step and embedded methods. Experiments with simulated data demonstrate that VC-PCR achieves better model accuracy, clustering performance and variable selection than competitor methods, for all configurations tested.

Rebecca Marion, Johannes Lederer, Bernadette Govaerts, and Rainer von Sachs (2021). "VC-PCR: A Prediction Method based on Supervised Variable Selection and Clustering". In: **To be submitted**

List of Symbols

$\mathbf{X}$: $n \times p$ predictor matrix with centered and scaled columns

$\mathbf{y}$: $n \times 1$ target vector (centered and scaled if continuous)

$\mathbf{U}$: $n \times K$ matrix of latent variables

$\mathbf{M}$: $n \times K$ matrix of latent variables/centroids used for prediction

$\mathbf{V}$: $p \times K$ matrix of cluster membership degrees

$\mathbf{H}$: $p \times K$ matrix of binary cluster memberships

$\mathbf{W}$: $p \times p$ diagonal matrix of weights for supervising the clustering task

π_i : probability that $y_i = 1$

See page 7 for other symbols used in this chapter.

4.1 Introduction

The previous chapter dealt with the problem of unsupervised variable clustering. In this chapter, we assume that a target vector $\mathbf{y}$ $(n \times 1)$ can be used to guide the clustering of variables in the matrix $\mathbf{X}$ $(n \times p)$. We are thus interested in the problem of *supervised variable clustering*. As with unsupervised variable clustering, supervised variable clustering is used to achieve the three main purposes described in Section 1.1: dimensionality reduction, variable selection and the identification of grouping structure. However, in the supervised setting, variable selection depends on the degree of association between predictor variables and the target variable.

The following sections summarize this research problem, our motivations and our contributions.

4.1.1 Problem Definition

In this chapter, we are interested in the problem of predicting a vector $\mathbf{y}$ based on a linear combination of the variables in $\mathbf{X}$. In what follows, we assume that the columns of $\mathbf{X}$ are mean-centered and have a sample variance of 1, i.e. $\text{mean}(\mathbf{x}_j) = 0$, $\text{sd}(\mathbf{x}_j) = 1$ and $\mathbf{x}_j^\top \mathbf{x}_j = n - 1,\ \forall j$. For regression problems, the elements in $\mathbf{y}$ are real, continuous values that have been mean-centered and scaled to unit sample variance. For two-class classification problems, the elements in $\mathbf{y}$ are binary (e.g. $y_i = 1$ if a patient has a given disease and $y_i = 0$ otherwise).

Let $\boldsymbol{b}$ $(p \times 1)$ be the underlying vector of model coefficients relating the variables in $\mathbf{X}$ to the target $\mathbf{y}$. In the context of linear regression, we assume that the target is a linear function of the variables in $\mathbf{X}$ plus an error term:

$$y_i = \mathbf{x}_i^\top \boldsymbol{b} + \varepsilon_i, \tag{4.1}$$

where $\varepsilon_i \overset{\text{iid}}{\sim} N(0, \sigma_\varepsilon^2)$.

For binary classification problems, we assume that $y_i \overset{\text{iid}}{\sim} Bernoulli(\pi_i)$, where

$$\pi_i = P(y_i = 1) = \frac{\exp(\mathbf{x}_i^\top \mathbf{b})}{1 + \exp(\mathbf{x}_i^\top \mathbf{b})}. \tag{4.2}$$

Given $\mathbf{X}$ and $\mathbf{y}$, we are interested in estimating a model that (i) accurately predicts $\mathbf{y}$, (ii) selects only the most relevant predictor variables and (iii) makes it possible to identify predictive variable clusters in $\mathbf{X}$.

4.1.2 Motivation and Related Methods

One field in which this problem arises is *transcriptomics*. In transcriptomics, gene expression datasets are often analyzed to identify the genes (i.e. biomarkers) that predict a given health outcome, such as cancer status or time to death. These datasets contain expression levels for p genes (the variables), measured for n samples (the observations). For complex diseases, it is common for clusters of genes (i.e. gene pathways) to contribute to the disease, rather than individual genes (Ma, Song, and Huang, 2007). Identifying these clusters can make it possible to better understand the mechanisms of disease, as well as develop targeted treatments.

Genes within a given pathway generally share similar functions and have highly correlated expression profiles. For regression and classification problems based on microarray gene expression data, this multicollinearity can greatly increase the variance of model coefficients, thereby decreasing prediction accuracy (Park, Hastie, and Tibshirani, 2007). In addition to this underlying cluster structure, gene expression data is highly dimensional ($p \gg n$), resulting in spurious correlations between relevant and irrelevant variables. In this setting, models tend to select too many variables, resulting in false positives (Daye and Jeng, 2009), i.e. irrelevant variables that are identified as relevant. Indeed, if two variables are correlated but only one of them is causal with respect to the target, classical linear models will estimate nonzero coefficients for both variables. This can lead to the erroneous conclusion that both variables are relevant, which is not the case.

To address these challenges, various methods have been proposed in the literature for estimating a sparse model that accounts for clustering structure among predictor variables. Most of these methods rest on the idea of encouraging similar variables to have similar or identical model coefficients, even if the true coefficients are not identical. This can potentially improve prediction accuracy, especially in

high-dimensional settings (Shen, Huang, and Pan, 2012) and in settings where variables from different clusters are uncorrelated but variables within a cluster are highly correlated (Park, Hastie, and Tibshirani, 2007; Bühlmann, Rütimann, Geer, and Zhang, 2013). This is due to the famous *bias-variance tradeoff*: by introducing bias in the estimation of model coefficients, it is possible to reduce their variance and thus increase prediction accuracy.

Most existing methods fall into one of two categories. *Embedded methods* simultaneously cluster variables and estimate a sparse model, whereas *two-step methods* first cluster variables, then estimate a sparse model based on these clusters. Some authors have shown that the embedded approach results in improved variable selection and clustering (Shen and Huang, 2010). Moreover, by supervising the variable clustering, embedded approaches are less likely to select clusters that are "diluted" with irrelevant variables (Yengo, Jacques, and Biernacki, 2014). On the other hand, two-step methods can actually benefit from high multicollinearity among predictor variables, as long as variables from different clusters are less correlated (Bühlmann, Rütimann, Geer, and Zhang, 2013). Moreover, they are less sensitive to problems related to high dimensionality and are often more computationally efficient. Clearly, both approaches —embedded and two-step —have their own set of advantages and disadvantages.

4.1.3 Contributions

In this chapter, we propose a method called Variable Cluster Principal Component Regression (VC-PCR) that combines the advantages of the embedded and two-step approaches. The first step of VC-PCR consists of supervised clustering, which is performed using a weighted version of a method that we call Sparse Orthogonal Semi-Nonnegative Matrix Factorization (SOS-NMF). As with the embedded approaches, supervising the variable clustering allows VC-PCR to identify homogeneous, predictive clusters. Moreover, irrelevant or "inactive" variables can be removed from the analysis during the clustering step, improving clustering performance and the recovery of the underlying model. In the second step of VC-PCR, a matrix of latent variables is calculated based on the cluster assignments from the first step, and a classical linear model is estimated to relate these latent variables to the target. Similar to existing two-step methods, the two-step approach employed by VC-PCR is more computationally efficient than embedded methods and can be easily applied to different prediction tasks (regression, classification, survival analysis, etc.).

One of the key ingredients to VC-PCR is the supervised clustering in the first step, where SOS-NMF is applied to the variables, which are weighted by their prior importance in the prediction task. SOS-NMF uses a sparsity penalty to remove irrelevant variables from the analysis, assigning them to none of the clusters. For each remaining variable, a continuous cluster membership degree is estimated for

the cluster to which it is assigned. These cluster membership degrees aid in the interpretation of the clusters, as they indicate the relative importance of variables within a cluster. This is useful for researchers involved in biomarker studies, as they can validate potential biomarkers (e.g. the selected variables) in order of their relative importance within their cluster, potentially saving time, energy and financial resources.

Thanks to the sparsity, orthogonality and semi-nonnegativity constraints in SOS-NMF, it is the only Nonnegative Matrix Factorization (NMF) method that can perform both variable clustering and selection on nonnegative matrices and matrices with positive and negative elements. While other related NMF methods exist in the literature (e.g. Ding, Li, and Jordan (2008), Ding, Li, Peng, and Park (2006), Kim and Park (2007), Dai, Liu, Zhu, Kong, Hou, and Yuan (2018), and Zass and Shashua (2007)), none of them simultaneously impose these three constraints. The most closely related method to SOS-NMF is Sparse Orthogonal NMF (SONMF) (Dai, Liu, Zhu, Kong, Hou, and Yuan, 2018), but this method is only applicable to nonnegative matrices, and its objective function is biased in favor of small clusters.

SOS-NMF has some similarities with Adaptive Clustering around Latent Variables (AdaCLV) (Marion, Govaerts, and von Sachs, 2020) and CLV $K+1$ (Vigneau, 2016), which were described in Sections 3.2 and 3.3.2, but it is generally more flexible and informative. For example, CLV $K+1$ only estimates binary clusters memberships, which are less informative than the continuous memberships in SOS-NMF. In addition, the sparsity penalty in CLV $K+1$ is weighted by the standard-deviation of the input variables, thereby attributing equal importance to all variables. SOS-NMF does not assume that each variable should be given equal weight. Rather, the variables can be weighted by their prior importance to encourage better clustering results.

AdaCLV also has certain limitations compared to SOS-NMF. First, AdaCLV depends on distributional assumptions that apply to spectroscopic variables, whereas SOS-NMF makes fewer assumptions about the data. In addition, while the cluster membership matrix in SOS-NMF has orthogonal columns (each variable belongs to at most one cluster), no such constraint is imposed for AdaCLV. As a result, redundant clusters —clusters with almost identical membership vectors —may be found. This is not problematic if AdaCLV is used to visually explore the data, and it makes AdaCLV fairly robust to the misspecification of K, the number of clusters to be found. However, the latent variables estimated based on these clusters cannot be directly used in classical prediction problems due to their near-colinearity. One solution to this problem would be to merge membership vectors whose similarity surpasses a certain threshold. However, based on our experience, this would require the specification of yet another hyperparameter —a similarity threshold —as the best threshold for merging seems to depend on the signal-to-noise ratio in the data.

Our numerical experiments show that the proposed prediction method, VC-PCR, performs better than other competing methods in terms of model error, variable selection and clustering. The model error for VC-PCR is also robust to different types of correlation structure. Correlation (i) between inactive variables and/or (ii) between inactive and active variables does not impact its performance, contrary to other methods.

4.1.4 Chapter Outline

VC-PCR and SOS-NMF are introduced in Section 4.2, and related methods from the literature are presented in Section 4.3. The numerical experiments and results are summarized in Section 4.4, and some conclusions and suggestions for future work can be found in Section 4.5.

4.2 Proposed Method: VC-PCR

Our proposed method, Variable Cluster Principal Component Regression (VC-PCR), consists of two steps. In the first step, variables are clustered, generating a matrix $\hat{\mathbf{V}}$ $(p \times K)$ of cluster membership degrees. Each element $\hat{v}_{jk}$ in $\hat{\mathbf{V}}$ is nonnegative and represents the estimated degree to which variable j belongs to cluster k. In the second step, $\hat{\mathbf{V}}$ is used to calculate a set of latent variables, $\hat{\mathbf{m}}_1, ..., \hat{\mathbf{m}}_K$, one for each variable cluster, and a linear regression or classification model is estimated based on these latent variables.

The first step of VC-PCR (variable clustering) is performed using a method that we call Sparse Orthogonal Semi-Nonnegative Matrix Factorization (SOS-NMF). This method is summarized in Section 4.2.1. The two steps of the VC-PCR algorithm are explained in Sections 4.2.2 and 4.2.3, and the full algorithm and a discussion of its hyperparameters are presented in Section 4.2.4.

4.2.1 Sparse Orthogonal Semi-NMF

Sparse Orthogonal Semi-Nonnegative Matrix Factorization (SOS-NMF) is an unsupervised method that can be used to perform dimensionality reduction, variable selection and variable clustering on both nonnegatives matrices and matrices with positive and negative elements. The objective of SOS-NMF is to decompose $\mathbf{X}$ into the product of two matrices: $\mathbf{U}$ $(n \times K)$, a matrix of K unit-variance latent variables, and $\mathbf{V}$ $(p \times K)$, a sparse, nonnegative matrix with orthogonal columns. Thanks to the orthogonality and nonnegativity of $\mathbf{V}$, each vector $\mathbf{v}_j$ (i.e. row of $\mathbf{V}$) can have at most one nonzero value. As a result, variable clusters can be inferred from $\mathbf{V}$: $C_k := \{j \mid v_{jk} > 0\}$. Moreover, the nonzero elements of $\mathbf{V}$ indicate the degree to which a variable belongs to its assigned cluster, where larger values indicate larger membership degrees.

The solution to our proposed SOS-NMF problem is

Sparse Orthogonal Semi-NMF (SOS-NMF)

$$\begin{aligned} \underset{\mathbf{U},\mathbf{V}}{\arg\min}\ & \frac{1}{2(n-1)}||\mathbf{X}-\mathbf{U}\mathbf{V}^\top||_F^2+\lambda\sum_{j=1}^{p}||\mathbf{v}_j||_1 \\ \text{s.t. } & v_{jk}\geq 0,\ \forall j,k \\ & \text{var}(\mathbf{u}_k)=1,\ \forall k \\ & \mathbf{V} \text{ has orthogonal columns,} \end{aligned} \tag{4.3}$$

where λ is a nonnegative hyperparameter controlling the sparsity in $\mathbf{V}$.

Intuition

The orthogonality and nonnegativity constraints make $\mathbf{V}$ a naturally sparse matrix. The ℓ_1 penalty in Eq. (4.3) encourages $\mathbf{V}$ to be even sparser, with some vectors $\mathbf{v}_j$ being set to zero. As a result, some variables (e.g. noise variables) may be assigned to none of the clusters and removed from the analysis. This is desirable because noise variables make it difficult to correctly cluster variables (see Chapter 3 and Vigneau (2016) and Marion, Govaerts, and von Sachs (2020)).

A given variable j is assigned to cluster k (i.e. $v_{jk}>0$) if its correlation with latent variable k is sufficiently large. As λ increases, the correlation threshold for cluster assignment increases, resulting in the removal of more and more variables from the analysis. Another necessary condition for being assigned to cluster k is having positive correlation with latent variable k. As a result, variables in the same cluster tend to be positively correlated.

Each latent variable vector $\mathbf{u}_k$ is proportional to the weighted sum of the variables j assigned to cluster k, where the weights are the cluster membership degrees v_{jk}. This weighted sum makes it possible to give greater importance to the variables that are the most representative of the cluster.

Optimization

The solution for SOS-NMF can be found by alternating between the optimization of $\mathbf{U}$ for fixed $\mathbf{V}$ and of $\mathbf{V}$ for fixed $\mathbf{U}$. For fixed $\mathbf{V}$, the objective function to be minimized is

$$J(\mathbf{U})=\frac{1}{2(n-1)}||\mathbf{X}-\mathbf{U}\mathbf{V}^\top||_F^2. \tag{4.4}$$

This is a least squares problem, and the solution for an unconstrained $\mathbf{U}$ is $\tilde{\mathbf{U}}=$

$\mathbf{XV}(\mathbf{V}^\top\mathbf{V})^{-1}$. We can account for the unit-variance constraint on the latent variables $\mathbf{u}_k$ by calculating the final solution $\hat{\mathbf{U}}$ as follows:

$$\hat{\mathbf{u}}_k = \tilde{\mathbf{u}}_k / \mathrm{sd}(\tilde{\mathbf{u}}_k). \tag{4.5}$$

Let $C_k := \{j \mid v_{jk} > 0\}$. For fixed $\mathbf{U}$, the objective function from Eq. (4.3) becomes

$$\begin{aligned}
J(\mathbf{V}) =& \frac{1}{2(n-1)}||\mathbf{X} - \mathbf{U}\mathbf{V}^\top||_F^2 + \lambda \sum_{j=1}^{p} ||\mathbf{v}_j||_1 \\
=& \frac{1}{2(n-1)} \sum_{j=1}^{p} \left\{ c_1 - 2\mathbf{v}_j^\top \mathbf{U}^\top \mathbf{x}_j + \mathbf{v}_j^\top \mathbf{U}^\top \mathbf{U} \mathbf{v}_j \right\} + \lambda \sum_{j=1}^{p} ||\mathbf{v}_j||_1 \\
=& c_2 + \sum_{k=1}^{K} \sum_{j \in C_k} \left\{ \frac{1}{2(n-1)} \Big(- 2v_{jk} \mathbf{u}_k^\top \mathbf{x}_j + v_{jk}^2 \underbrace{\mathbf{u}_k^\top \mathbf{u}_k}_{n-1} \Big) + \lambda |v_{jk}| \right\} \\
=& c_2 + \sum_{k=1}^{K} \sum_{j \in C_k} \left\{ - \frac{1}{n-1} v_{jk} \mathbf{u}_k^\top \mathbf{x}_j + \frac{1}{2} v_{jk}^2 + \lambda |v_{jk}| \right\},
\end{aligned} \tag{4.6}$$

where c_1 and c_2 are constants that do not depend on $\mathbf{V}$.

Suppose that, for a given variable j, $j \in C_k$ and thus $v_{jk} > 0$. The solution $\hat{v}_{jk}$ is found by deriving Eq. (4.6) with respect to v_{jk} and setting it to zero.

$$\frac{\partial J(\mathbf{V})}{\partial v_{jk}} = -\frac{1}{n-1} \mathbf{u}_k^\top \mathbf{x}_j + v_{jk} + \lambda = 0. \tag{4.7}$$

Thus, if $j \in C_k$,

$$\begin{aligned}
\hat{v}_{jk} &= \frac{1}{n-1} \mathbf{u}_k^\top \mathbf{x}_j - \lambda \\
&= \mathrm{cov}(\mathbf{u}_k, \mathbf{x}_j) - \lambda \\
&= \underbrace{\mathrm{sd}(\mathbf{u}_k)}_{1} \underbrace{\mathrm{sd}(\mathbf{x}_j)}_{1} \mathrm{cor}(\mathbf{u}_k, \mathbf{x}_j) - \lambda \\
&= \mathrm{cor}(\mathbf{u}_k, \mathbf{x}_j) - \lambda.
\end{aligned} \tag{4.8}$$

The problem then becomes one of determining when $j \in C_k$. Thanks to Eq. (4.8), we can deduce that $\mathrm{cor}(\mathbf{u}_k, \mathbf{x}_j) > \lambda$ is a necessary condition for variable j to belong to cluster k (i.e. $v_{jk} > 0$). In addition to satisfying this condition, another necessary condition is that

$$k = \arg\max_{\ell} \text{cor}(\mathbf{u}_\ell, \mathbf{x}_j). \tag{4.9}$$

Indeed, thanks to the orthogonality and nonnegativity constraints, at most one element of $\mathbf{v}_j$ may be greater than zero. This means that, for a given variable j, the sum over k in Eq. (4.6) is only active for one value k. For $v_{jk} > 0$, the objective function evaluated at the solution $\hat{v}_{jk}$ is

$$\begin{aligned} J(\hat{v}_{jk}) &= -\frac{1}{n-1}\hat{v}_{jk}\mathbf{u}_k^\top \mathbf{x}_j + \frac{1}{2}\hat{v}_{jk}^2 + \lambda \hat{v}_{jk} \\ &= -\hat{v}_{jk}\underbrace{\text{cor}(\mathbf{u}_k, \mathbf{x}_j)}_{\hat{v}_{jk}+\lambda} + \frac{1}{2}\hat{v}_{jk}^2 + \lambda \hat{v}_{jk} \\ &= -\frac{1}{2}\hat{v}_{jk}^2 \end{aligned} \tag{4.10}$$

For a fixed variable j, the index k with the smallest $J(\hat{v}_{jk})$ is the one for which $\text{cor}(\mathbf{u}_k, \mathbf{x}_j)$ is the largest.

4.2.2 VC-PCR: Step 1

For the first step of VC-PCR, a matrix of cluster memberships $\mathbf{V}$ must be estimated. As one of our objectives is to predict a target $\mathbf{y}$, we propose making the clustering step dependent on the target. We do this by integrating information about the target into the SOS-NMF problem using a weight matrix $\mathbf{W}$.

Let $\mathbf{W}$ $(p \times p)$ be a diagonal matrix whose entries w_{jj} encode (i) the prior relationship between variable j and the target $\mathbf{y}$ (e.g. positive vs negative association), as well as (ii) the importance of variable j in the prediction task. For example, the weights could be the coefficients for a Ridge regression model (see Section 4.2.4 for other suggestions). When each variable j is weighted by w_{jj}, the solution to SOS-NMF becomes

Weighted SOS-NMF

$$\begin{aligned} \arg\min_{\mathbf{U},\mathbf{V}} \ & \frac{1}{2(n-1)}||\mathbf{XW} - \mathbf{UV}^\top||_F^2 + \lambda \sum_{j=1}^{p} ||\mathbf{v}_j||_1 \\ \text{s.t. } & v_{jk} \geq 0, \ \forall j, k \\ & \text{var}(\mathbf{u}_k) = 1, \ \forall k \\ & \mathbf{V} \text{ has orthogonal columns,} \end{aligned} \tag{4.11}$$

where λ is a nonnegative hyperparameter controlling the sparsity in $\mathbf{V}$.

Intuition

The weight matrix $\mathbf{W}$ allows the clustering to be supervised by the target $\mathbf{y}$. For example, variables that have a similar importance for the prediction problem (i.e. similar weights in absolute value) are more likely to be clustered together. In addition, variables with smaller weights (in absolute value) are more likely to be removed by the ℓ_1 penalty.

In contrast to the original SOS-NMF problem, variables may be assigned to cluster k even if they are negatively correlated with latent variable k. Suppose there are two variables j and ℓ such that $\text{cor}(\mathbf{u}_k, \mathbf{x}_j) = -\text{cor}(\mathbf{u}_k, \mathbf{x}_\ell) = \rho$ and $w_{jj} = -w_{\ell\ell} = \nu$. If variable j is assigned to cluster k, then variable ℓ will be as well. The higher the absolute correlation between variables, and the more similar their weights (in absolute value), the more likely it is that they will be clustered together.

Optimization

The solution for Weighted SOS-NMF can be found by using $\mathbf{XW}$ as the input to SOS-NMF rather than $\mathbf{X}$. For fixed $\mathbf{V}$, the solution for an unconstrained $\mathbf{U}$ is $\tilde{\mathbf{U}} = \mathbf{XWV}(\mathbf{V}^\top\mathbf{V})^{-1}$. The solution for each unit-variance latent variable $\mathbf{u}_k$ can be calculated by scaling $\tilde{\mathbf{u}}_k$ (see Eq. (4.5)).

For fixed $\mathbf{U}$, a given variable j is assigned to cluster k if two conditions are met:

1. $k = \arg\max_\ell w_{jj}\text{cor}(\mathbf{u}_\ell, \mathbf{x}_j)$ and
2. $w_{jj}\text{cor}(\mathbf{u}_k, \mathbf{x}_j) > \lambda$.

For all $j \in C_k$, the solution $\hat{v}_{jk}$ is given by

$$\begin{aligned}\hat{v}_{jk} &= \frac{1}{n-1}\mathbf{u}_k^\top \mathbf{x}_j w_{jj} - \lambda \\ &= \text{cov}(\mathbf{u}_k, \mathbf{x}_j w_{jj}) - \lambda \\ &= \underbrace{\text{sd}(\mathbf{u}_k)}_{1}\underbrace{\text{sd}(\mathbf{x}_j w_{jj})}_{|w_{jj}|}\underbrace{\text{cor}(\mathbf{u}_k, \mathbf{x}_j w_{jj})}_{\text{sign}(w_{jj})\text{cor}(\mathbf{u}_k,\mathbf{x}_j)} - \lambda \\ &= w_{jj}\text{cor}(\mathbf{u}_k, \mathbf{x}_j) - \lambda.\end{aligned} \tag{4.12}$$

4.2.3 VC-PCR: Step 2

Once the matrix $\mathbf{V}$ of cluster memberships is estimated, the second step is to calculate a latent variable matrix $\hat{\mathbf{M}} = f(\mathbf{X}, \hat{\mathbf{V}})$ and estimate a linear model predicting $\mathbf{y}$ based on $\hat{\mathbf{M}}$. Note that the latent variable matrix $\hat{\mathbf{U}}$ from the first step is not used in the second step. This is because $\hat{\mathbf{U}}$ is a function of $\mathbf{XW}$, a modified version

of $\mathbf{X}$. We are interested in relating the original matrix $\mathbf{X}$ to the target $\mathbf{y}$, therefore, the latent variable matrix $\hat{\mathbf{M}}$ is calculated as follows:

$$\hat{\mathbf{M}} = \mathbf{X}\hat{\mathbf{V}}. \tag{4.13}$$

The prediction problem requires solving

Linear Modeling Problem

$$\arg\min_{\mathbf{a}} \; L(\hat{\mathbf{M}}, \mathbf{y}; \mathbf{a}), \tag{4.14}$$

where $\mathbf{a}$ $(K \times 1)$ is a vector of model coefficients and $L(\hat{\mathbf{M}}, \mathbf{y}; \mathbf{a})$ is a loss function that depends on the modeling task.

For linear regression tasks, the loss function

$$L(\hat{\mathbf{M}}, \mathbf{y}; \mathbf{a}) = ||\mathbf{y} - \hat{\mathbf{M}}\mathbf{a}||_2^2, \tag{4.15}$$

and $\mathbf{a}$ can be found using Ordinary Least Squares (OLS): $\hat{\mathbf{a}} = (\hat{\mathbf{M}}^\top \hat{\mathbf{M}})^{-1}\hat{\mathbf{M}}^\top \mathbf{y}$. Estimating a sparse regression model is not necessary because variable selection is already performed in the first step: variables are removed from the analysis thanks to the sparsity penalty in Eq. (4.11).

For classification tasks with binary y_i, the likelihood-based loss function

$$L(\hat{\mathbf{M}}, \mathbf{y}; \mathbf{a}) = -\sum_{i=1}^{n} y_i \log(\pi_i) + (1 - y_i) \log(1 - \pi_i), \tag{4.16}$$

where

$$\pi_i = P(y_i = 1) = \frac{\exp(\hat{\mathbf{m}}_i^\top \mathbf{a})}{1 + \exp(\hat{\mathbf{m}}_i^\top \mathbf{a})}. \tag{4.17}$$

The model coefficients $\mathbf{a}$ can be found using Logistic Regression (LR). Again, sparse regression is not necessary here because variable selection has already been performed.

For both regression and classification tasks, a vector $\hat{\boldsymbol{b}}$ $(p \times 1)$ of coefficients for the original variables in $\mathbf{X}$ can be calculated as a function of $\hat{\mathbf{V}}$ and $\hat{\mathbf{a}}$. The predicted value $\hat{y}_i$ is a function of $\hat{\mathbf{m}}_i^\top \hat{\mathbf{a}} = \mathbf{x}_i^\top \hat{\mathbf{V}}\hat{\mathbf{a}}$. Therefore, the coefficients $\hat{\boldsymbol{b}}$ relating the original variables to the target can be calculated as $\hat{\boldsymbol{b}} = \hat{\mathbf{V}}\hat{\mathbf{a}}$.

4.2.4 Algorithm and Hyperparameters

This section summarizes the VC-PCR algorithm and offers guidance with respect to the choice of the hyperparameters λ and K, as well as the weight matrix $\mathbf{W}$.

Algorithm

The entire VC-PCR algorithm is summarized below in Algorithm 2. At the beginning of the algorithm, $\mathbf{V}$ is initialized based on an initial partition $C_1, ..., C_K$ of K variable clusters, where $v_{jk} = \mathbb{I}[j \in C_k]$.

Choice of Weights

As mentioned in Section 4.2.2, the entries w_{jj} in $\mathbf{W}$ should indicate (i) the prior relationship between variable j and the target $\mathbf{y}$ (e.g. positive vs negative association) and (ii) the importance of variable j in the prediction task. We propose two potential weighting schemes based on the coefficients of Lasso and Ridge models.

Given $\mathbf{X}$, $\mathbf{y}$ and a hyperparameter δ, a vector of coefficients $\mathbf{w}$ is estimated by solving

Regularized Linear Modeling Problem

$$\arg\min_{\mathbf{w}} \; L(\mathbf{X}, \mathbf{y}; \mathbf{w}) + \text{Penalty}(\mathbf{w}; \delta), \tag{4.18}$$

where $L(\mathbf{X}, \mathbf{y}; \mathbf{w})$ is a loss function and $\text{Penalty}(\mathbf{w}; \delta)$ is a penalty function. For Lasso coefficients, this penalty is

$$\text{Penalty}(\mathbf{w}; \delta) = \delta \sum_{j=1}^{p} |w_j|, \tag{4.19}$$

and for Ridge coefficients, the penalty is

$$\text{Penalty}(\mathbf{w}; \delta) = \delta \sum_{j=1}^{p} w_j^2. \tag{4.20}$$

When Lasso weights are used, prior variable selection is performed, as variables with Lasso coefficients equal to zero are essentially excluded from the variable clustering task. The remaining variables are weighted by a value that reflects their relative importance in the prediction task, as well as the direction of their correlation with

Data: A predictor matrix $\mathbf{X}$ $(n \times p)$ with mean-centered, unit-variance columns, a target vector $\mathbf{y}$ $(n \times 1)$ (binary for classification, centered and scaled for regression), a diagonal weight matrix $\mathbf{W}$ $(p \times p)$, an initial partition $C_1, ..., C_K$ and a hyperparameter $\lambda > 0$

// Initialization
for each cluster k **in** $1, ..., K$ **do**
  for each variable j **in** $1, ..., p$ **do**
    $v_{jk} = \mathbb{I}[j \in C_k]$;
  end
end
// Step 1: Weighted SOS-NMF
while not converged do
  // Update latent variable matrix
  if there are columns of zeroes in V then
    Remove columns of zeroes from $\mathbf{V}$;
  end
  $\tilde{\mathbf{U}} = \mathbf{XWV}(\mathbf{V}^\top\mathbf{V})^{-1}$;
  for each cluster k **in** $1, ..., K$ **do**
    $\mathbf{u}_k = \tilde{\mathbf{u}}_k / \mathrm{sd}(\tilde{\mathbf{u}}_k)$;
  end
  // Update cluster membership matrix
  for each variable j **in** $1, ..., p$ **do**
    $\hat{k} = \arg\max_\ell w_{jj}\mathrm{cor}(\mathbf{u}_\ell, \mathbf{x}_j)$;
    $v_{j\hat{k}} = (w_{jj}\mathrm{cor}(\mathbf{u}_{\hat{k}}, \mathbf{x}_j) - \lambda)_+$;
  end
end
$\hat{\mathbf{V}} = \mathbf{V}$
// Step 2: Linear Modeling
$\hat{\mathbf{M}} = \mathbf{X}\hat{\mathbf{V}}$;
Calculate $\hat{\mathbf{a}}$ by solving Eq. (4.14);
return Sparse cluster membership matrix $\hat{\mathbf{V}}$, latent variable matrix $\hat{\mathbf{M}}$ and model coefficients $\hat{\mathbf{a}}$

Algorithm 2: VC-PCR

the target. Ridge weights encode similar information but do not pre-select variables: in most cases, all variables are given a nonzero weight. However, variables with high absolute correlation are more likely to have similar weights in absolute value. This encourages SOS-NMF to cluster these variables together.

Choice of Hyperparameters

The VC-PCR algorithm depends on three hyperparameters: K, the number of clusters in the initial partitions, δ, the hyperparameter used to estimate the weights in $\mathbf{W}$, and λ, which determines the degree of sparsity in $\mathbf{V}$. In practice, these hyperparameters may be tuned using k-fold cross-validation, using a criterion such as the Mean Squared Error of Prediction (MSEP) for regression problems or the Matthews Correlation Coefficient (MCC) (Matthews, 1975) for classification problems.

For the sparsity hyperparameter λ, the smallest value leading to $\mathbf{V} = \mathbf{0}$ can be determined based on the first iteration of the while loop in Algorithm 2. Let $\mathbf{u}_k^{(1)}$ be the k^{th} latent variable at the first iteration. All elements $v_{jk} = 0$ if $\lambda \geq w_{jj}\text{cor}(\mathbf{u}_k^{(1)}, \mathbf{x}_j)$, $\forall j, k$. Therefore, the largest value of λ to consider is

$$\lambda_{\max} = \max_{j,k} \; w_{jj}\text{cor}(\mathbf{u}_k^{(1)}, \mathbf{x}_j). \tag{4.21}$$

4.3 Related Work

In this section, methods related to the proposed SOS-NMF and VC-PCR methods are presented. In Section 4.3.1, several nonnegative matrix factorization (NMF) methods are summarized and compared to SOS-NMF. In Sections 4.3.2-4.3.4, three different classes of prediction methods are described and compared to VC-PCR: coefficient-grouping methods (Section 4.3.2), ensemble methods (Section 4.3.3) and two-step methods (Section 4.3.4).

4.3.1 Nonnegative Matrix Factorization Methods

SOS-NMF is most closely related to the problem of Nonnegative Matrix Factorization (NMF). While certain methods exist for addressing aspects of the SOS-NMF problem, no method simultaneously addresses all aspects (i.e. sparsity, orthogonality and semi-nonnegativity). This section demonstrates the relationship between SOS-NMF and several NMF methods from the literature. The objective functions and constraints associated with each method are summarized in Table 4.1.

Semi-NMF

Given a matrix $\mathbf{X}$, Semi-NMF (Ding, Li, and Jordan, 2008) solves the problem

Semi-NMF (S-NMF)

$$\begin{aligned} \underset{\mathbf{U},\mathbf{V}}{\arg\min}\ & \frac{1}{2(n-1)}||\mathbf{X}-\mathbf{U}\mathbf{V}^\top||_F^2 \\ & \text{s.t. } v_{jk} \geq 0,\ \forall j,k. \end{aligned} \tag{4.22}$$

As with SOS-NMF, only $\mathbf{V}$ is constrained to be nonnegative (hence the name "semi"-NMF). However, in contrast to SOS-NMF, Semi-NMF does not constrain the columns of $\mathbf{V}$ to be orthogonal or impose any sparsity penalties. As a result, for a given variable j, all of the elements in $\mathbf{v}_j$ may be nonzero, meaning that variable j is essentially assigned to all of the clusters. Thus, Semi-NMF is not a candidate method for variable clustering.

Orthogonal NMF

Given a nonnegative matrix $\mathbf{Z}$, one-sided Orthogonal NMF (Ding, Li, Peng, and Park, 2006) solves the problem

Orthogonal NMF

$$\begin{aligned} \underset{\mathbf{U},\mathbf{V}}{\arg\min}\ & \frac{1}{2(n-1)}||\mathbf{Z}-\mathbf{U}\mathbf{V}^\top||_F^2 \\ & \text{s.t. } v_{jk} \geq 0,\ \forall j,k \\ & \quad\ u_{ik} \geq 0,\ \forall i,k \\ & \quad\ \mathbf{V}^\top\mathbf{V} = \mathbf{I}_K. \end{aligned} \tag{4.23}$$

As with SOS-NMF, the columns of $\mathbf{V}$ are constrained to be orthogonal. Combined with the nonnegativity constraint on $\mathbf{V}$, this makes Orthogonal NMF a candidate method for variable clustering. However, Orthogonal NMF requires the input of a nonnegative matrix $\mathbf{Z}$. Moreover, no sparsity penalty is applied to $\mathbf{V}$, making the method less robust to noise variables.

Sparse NMF

Given a nonnegative matrix $\mathbf{Z}$, the Sparse NMF method proposed in Kim and Park (2007) ("SNMF/R") solves the problem

Sparse NMF (SNMF)

$$\begin{aligned} \underset{\mathbf{U},\mathbf{V}}{\arg\min}\ & \frac{1}{2(n-1)}||\mathbf{Z}-\mathbf{U}\mathbf{V}^\top||_F^2+\delta\sum_{i=1}^{n}||\mathbf{u}_i||_2^2+\lambda\sum_{j=1}^{p}||\mathbf{v}_j||_1^2 \\ \text{s.t. } & v_{jk}\geq 0,\ \forall j,k \\ & u_{ik}\geq 0,\ \forall i,k. \end{aligned} \tag{4.24}$$

As this method does not constrain the columns of $\mathbf{V}$ to be orthogonal, each vector $\mathbf{v}_j$ may contain more than one nonzero element. As a result, each variable j may be assigned to multiple clusters. While the sparsity penalty for the vectors $\mathbf{v}_j$ can encourage some elements of $\mathbf{v}_j$ to be zero, in practice, the clustering of variables can be unsatisfactory (see Chapter 3 and Marion, Govaerts, and von Sachs (2020)).

Sparse Orthogonal NMF

Given a nonnegative matrix $\mathbf{Z}$, Sparse Orthogonal NMF (SONMF) (Dai, Liu, Zhu, Kong, Hou, and Yuan, 2018) solves the problem

Sparse Orthogonal NMF (SONMF)

$$\begin{aligned} \underset{\mathbf{U},\mathbf{V}}{\arg\min}\ & \frac{1}{2(n-1)}||\mathbf{Z}-\mathbf{U}\mathbf{V}^\top||_F^2+\lambda\sum_{j=1}^{p}||\mathbf{v}_j||_1 \\ \text{s.t. } & v_{jk}\geq 0,\ \forall j,k \\ & u_{ik}\geq 0,\ \forall i,k \\ & \mathbf{V}^\top\mathbf{V}=\mathbf{I}_K. \end{aligned} \tag{4.25}$$

In contrast to SOS-NMF, the input $\mathbf{Z}$ to SONMF must be nonnegative, making SONMF less flexible. Moreover, the scale of the columns of $\mathbf{V}$ is constrained, rather than the scale of the columns of $\mathbf{U}$. These scaling differences have implications for the estimation of $\mathbf{V}$. For SOS-NMF, the solution for $\mathbf{v}_j$ can be solved independently for each variable j because the scale of each vector $\mathbf{v}_k$ is not fixed. For SONMF, the solution for v_{jk} depends on all other entries in $\mathbf{V}$ due to the orthonormality constraint. As a result, the solution for each vector $\mathbf{v}_j$ cannot be found independently, and the scale of v_{jk} for variables $j \in C_k$ depends on the size of cluster k. The more variables that are assigned to cluster k, the more likely it is that these variables will have low memberships v_{jk}, to satisfy the constraint $\mathbf{v}_k^\top\mathbf{v}_k = 1$. This unfairly penalizes large clusters, as the sparsity penalty makes small values v_{jk} shrink to zero.

Nonnegative Sparse Principal Component Analysis

SOS-NMF is also related to Nonnegative Sparse Principal Component Analysis (Zass and Shashua, 2007). Its solution is given by

Nonnegative Sparse Principal Component Analysis (NSPCA)

$$\begin{aligned} \arg\max_{\mathbf{V}} \ & \frac{1}{2(n-1)}||\mathbf{XV}||_F^2 - \lambda \sum_{j=1}^{p} ||\mathbf{v}_j||_1 - \gamma||\mathbf{V}^\top\mathbf{V} - \mathbf{I}_K||_F^2 \\ \text{s.t. } & v_{jk} \geq 0, \ \forall j, k, \end{aligned} \tag{4.26}$$

where $\mathbf{V}$ is a sparse, nonnegative loadings matrix and λ and γ are hyperparameters controlling the degree of sparsity and orthonormality of the columns in $\mathbf{V}$. In contrast to SOS-NMF, the columns of $\mathbf{V}$ in NSPCA are only quasi-orthogonal, unless $\gamma = \infty$. This means that variables may be assigned to multiple clusters (i.e. $\mathbf{v}_j$ may have multiple nonzero entries).

For the special case where $\gamma = \infty$, NSPCA only differs from SOS-NMF with respect to the scaling constraints on $\mathbf{U}$ and $\mathbf{V}$: SOS-NMF constrains the scale of $\mathbf{U}$ whereas NSPCA constrains the scale of $\mathbf{V}$. The NSPCA problem for $\gamma = \infty$, referred to here as Orthonormal NSPCA, is

Orthonormal NSPCA

$$\begin{aligned} \arg\min_{\mathbf{U},\mathbf{V}} \ & \frac{1}{2(n-1)}||\mathbf{X} - \mathbf{UV}^\top||_F^2 + \lambda \sum_{j=1}^{p} ||\mathbf{v}_j||_1 \\ \text{s.t. } & v_{jk} \geq 0, \ \forall j, k \\ & \mathbf{V}^\top\mathbf{V} = \mathbf{I}_K, \end{aligned} \tag{4.27}$$

where $\mathbf{U}$ $(n \times K)$ is a matrix of latent variables. The mathematical development leading to this reformulation can be found in the Appendix (Section 4.7.1).

To the best of our knowledge, no optimization strategies have been proposed for solving Orthonormal NSPCA. Moreoever, Orthonormal NSPCA suffers from the same scaling problems as Sparse Orthogonal NMF (SONMF): the scaling constraints make the estimation of $\mathbf{v}_j$ dependent on all of the other variables $\ell \neq j$.

Method	**Objective Function**	**Constraints**
Semi-NMF[1]	$\frac{1}{2(n-1)}\|\|\mathbf{X}-\mathbf{U}\mathbf{V}^\top\|\|_F^2$	$v_{jk} \geq 0,\ \forall j,k$
Orthogonal NMF[2]	$\frac{1}{2(n-1)}\|\|\mathbf{Z}-\mathbf{U}\mathbf{V}^\top\|\|_F^2$	$v_{jk} \geq 0,\ \forall j,k$
		$u_{ik} \geq 0,\ \forall i,k$
		$\mathbf{V}^\top\mathbf{V} = \mathbf{I}_K$
Sparse NMF[3]	$\frac{1}{2(n-1)}\|\|\mathbf{Z}-\mathbf{U}\mathbf{V}^\top\|\|_F^2+\delta\sum_{i=1}^n\|\|\mathbf{u}_i\|\|_2^2+\lambda\sum_{j=1}^p\|\|\mathbf{v}_j\|\|_1^2$	$v_{jk} \geq 0,\ \forall j,k$
		$u_{ik} \geq 0,\ \forall i,k$
Sparse Orthogonal NMF[4]	$\frac{1}{2(n-1)}\|\|\mathbf{Z}-\mathbf{U}\mathbf{V}^\top\|\|_F^2+\lambda\sum_{j=1}^p\|\|\mathbf{v}_j\|\|_1$	$v_{jk} \geq 0,\ \forall j,k$
		$u_{ik} \geq 0,\ \forall i,k$
		$\mathbf{V}^\top\mathbf{V} = \mathbf{I}_K$
Nonnegative Sparse PCA[5]	$\frac{1}{2(n-1)}\|\|\mathbf{X}\mathbf{V}\|\|_F^2-\lambda\sum_{j=1}^p\|\|\mathbf{v}_j\|\|_1-\gamma\|\|\mathbf{V}^\top\mathbf{V}-\mathbf{I}_K\|\|_F^2$	$v_{jk} \geq 0,\ \forall j,k$
Clustering around Latent Variables $K+1$[6]	$\frac{1}{2(n-1)}\|\|\mathbf{X}-\mathbf{U}\mathbf{V}^\top\|\|_F^2+\lambda\sum_{j=1}^p \mathrm{sd}(\mathbf{x}_j)\|\|\mathbf{v}_j\|\|_1$	$\mathbf{V} \in \{0,1\}^{p\times K}$
		$\mathbf{V}$ has orthogonal columns
		$\mathrm{var}(\mathbf{u}_k) = 1$
Sparse Orthogonal Semi-NMF[7]	$\frac{1}{2(n-1)}\|\|\mathbf{X}-\mathbf{U}\mathbf{V}^\top\|\|_F^2+\lambda\sum_{j=1}^p\|\|\mathbf{v}_j\|\|_1$	$v_{jk} \geq 0,\ \forall j,k$
		$\mathbf{V}$ has orthogonal columns
		$\mathrm{var}(\mathbf{u}_k) = 1$

Table 4.1: **Summary of NMF methods.**

Clustering around Latent Variables $K+1$

Although it was not originally presented as a matrix factorization method, Clustering around Latent Variables $K+1$ (CLV $K+1$) (Vigneau, 2016) can be rewritten as a matrix factorization problem that bears certain similarities to SOS-NMF. In Section 3.3.2, it was demonstrated that the solution to the local mode of CLV $K+1$ is

[1] Ding, Li, and Jordan, 2008
[2] Ding, Li, Peng, and Park, 2006
[3] Kim and Park, 2007
[4] Dai, Liu, Zhu, Kong, Hou, and Yuan, 2018
[5] Zass and Shashua, 2007
[6] Vigneau, 2016
[7] Method proposed in this paper

CLV $K+1$

$$\begin{aligned} \underset{\mathbf{U},\mathbf{V}}{\arg\min}\ & \frac{1}{2(n-1)}||\mathbf{X}-\mathbf{U}\mathbf{V}^\top||_F^2+\lambda\sum_{j=1}^{p}\text{sd}(\mathbf{x}_j)||\mathbf{v}_j||_1 \\ \text{s.t. } & \mathbf{V}\in\{0,1\}^{p\times K} \\ & \mathbf{V} \text{ has orthogonal columns} \\ & \text{var}(\mathbf{u}_k)=1\ \forall k. \end{aligned} \tag{4.28}$$

In contrast to SOS-NMF, the cluster memberships v_{jk} are constrained to be binary, making them less informative. In addition, the sparsity penalty is weighted by the standard-deviation of the input variables. As a result, cluster assignments only depend on the correlation between the latent variables and the input variables. This means that modifying the scale of the input variables, as is proposed for the first step of VC-PCR (see Section 4.2.2), does not affect the relative importance of variables during clustering. Therefore, unlike for SOS-NMF, prior weighting cannot be used to supervise the variable clustering.

4.3.2 Coefficient-Grouping Methods

Most of the supervised variable clustering methods in the literature rely on a regression framework, identifying variable clusters by pushing regression coefficients towards each other in (absolute) value. These *coefficient-grouping methods* (see Table 4.2) generally optimize a least-squares criterion with two penalties: a sparsity penalty, which encourages null coefficients, and a grouping penalty, which encourages coefficients for certain variables to take the same (absolute) value. The objective function for these methods usually takes the form

$$J(\boldsymbol{b})=\frac{1}{2}||\mathbf{y}-\mathbf{X}\boldsymbol{b}||_2^2+\text{Penalty}_S(\boldsymbol{b};\delta)+\text{Penalty}_G(\boldsymbol{b};\lambda) \tag{4.29}$$

where $\lambda\geq 0$ and $\delta\geq 0$ are hyperparameters, $\text{Penalty}_S(\boldsymbol{b})$ is a sparsity penalty and $\text{Penalty}_G(\boldsymbol{b})$ is a coefficient-grouping penalty. With the exception of Cluster Elastic Net (CEN) (Witten, Shojaie, and Zhang, 2014), these methods identify variable clusters after the vector of coefficients $\boldsymbol{b}$ is estimated: all variables with the same estimated coefficient (in absolute or real value, depending on the method) are assigned to the same cluster.

Some of the coefficient-grouping methods rely on prior graphs. If the graph is not known in advance, a fully connected graph containing all $p(p-1)/2$ possible edges may be used. However, this strategy can be inefficient, even for relatively small p, as the computational time can increase exponentially with the number of edges (Yang, Yuan, Lai, Shen, Wonka, and Ye, 2012). Non-graph-based methods

that penalize sums over all pairs of variables also suffer from efficiency problems. Thus, computation time is one of the major challenges for coefficient-grouping methods.

Method	HPs	Sparsity Term	Grouping Term	b_j, b_ℓ grouping iff	Grouping Type
Flasso [8]	2	$\delta \sum_{j=1}^{p} \lvert b_j \rvert$	$\lambda \sum_{j=2}^{p} \lvert b_j - b_{j-1} \rvert$	adjacent	NonAbsVal
CLasso [9]	2	$\delta \sum_{j=1}^{p} \lvert b_j \rvert$	$\lambda \sum_{1 \le j < \ell \le p} \lvert b_j - b_\ell \rvert$		NonAbsVal
OSCAR [10]	2	$\delta \sum_{j=1}^{p} \lvert b_j \rvert$	$\lambda \sum_{1 \le j < \ell \le p} \frac{1}{2} \lvert b_j - b_\ell \rvert + \frac{1}{2} \lvert b_j + b_\ell \rvert$		AbsVal
PACS [11]	2+	$\lambda \sum_{j=1}^{p} \omega_j \lvert b_j \rvert$	$\lambda \sum_{1 \le j < \ell \le p} \omega_{j\ell}^{(+)} \lvert b_j - b_\ell \rvert + \sum_{1 \le j < \ell \le p} \omega_{j\ell}^{(-)} \lvert b_j + b_\ell \rvert$		AbsVal
WF [12]	3	$\lambda \sum_{j=1}^{p} \lvert b_j \rvert$	$\lambda \sum_{1 \le j < \ell \le p} \omega_{j\ell} (b_j - \text{sign}(r_{j\ell}) b_\ell)^2$		AbsVal
SSCFS [13]	3+	$\delta \sum_{j=1}^{p} P_\tau(b_j)$	$\lambda \sum_{j,\ell \in \mathcal{E}} P_\tau(\lvert b_j - b_\ell \rvert)$	shared edge	NonAbsVal
GFLasso [14]	2+	$\delta \sum_{j=1}^{p} \lvert b_j \rvert$	$\lambda \sum_{j,\ell \in \mathcal{E}} \lvert b_j - \text{sign}(r_{j\ell}) b_\ell \rvert$	shared edge and $r_{j\ell} > 0$	AbsVal
NCFGS [15]	2+	$\delta \sum_{j=1}^{p} \lvert b_j \rvert$	$\lambda \sum_{j,\ell \in \mathcal{E}} \Big\lvert \lvert b_j \rvert - \lvert b_\ell \rvert \Big\rvert$	shared edge	AbsVal
NCTFGS [16]	3+	$\delta \sum_{j=1}^{p} P_\tau(b_j)$	$\lambda \sum_{j,\ell \in \mathcal{E}} P_\tau\Big(\Big\lvert \lvert b_j \rvert - \lvert b_\ell \rvert \Big\rvert\Big)$	shared edge	AbsVal
CEN [17]	3	$\delta \sum_{j=1}^{p} \lvert b_j \rvert$	$\lambda \sum_{k=1}^{K} \sum_{j \in C_k} \lVert \mathbf{x}_j b_j - \frac{1}{\lvert C_k \rvert} \sum_{\ell \in C_k} \mathbf{x}_\ell b_\ell \rVert_2^2$	$\lvert r_{j\ell} \rvert \to 1$ and $j, \ell \in C_k$	AbsVal

Table 4.2: **Summary of coefficient-grouping methods.** $r_{j\ell} = \text{cor}(\mathbf{x}_j, \mathbf{x}_\ell)$; $\mathcal{E}$ is the set of edges in a given graph; $P_\tau(x) = \min(\frac{x}{\tau}, 1)$; ω_j, $\omega_{j\ell}^{(+)}$ and $\omega_{j\ell}^{(+)}$ are weights that depend on the chosen PACS approach; $\omega_{j\ell} = \frac{\lvert r_{j\ell} \rvert^\gamma}{1 - \lvert r_{j\ell} \rvert}$ for $\gamma > 0$; C_k is the set of variables assigned to cluster k. The grouping type for methods that push coefficients towards each other in absolute value (resp. real value) is "AbsVal" (resp. "NonAbsVal"). For methods involving graphs, the number of hyperparameters (HPs) is presented with a + to indicate that more HPs may be necessary for generating the required graph.

Cluster Elastic Net (CEN)

While in the same spirit as other coefficient-grouping methods, Cluster Elastic Net (CEN) (Witten, Shojaie, and Zhang, 2014) approaches the problem of supervised variable clustering in a unique way. The grouping penalty for Cluster Elastic Net (CEN) contains the objective function for K-means, where the points to cluster are the variables weighted by their respective regression coefficients. At each iteration of the CEN algorithm, pairs of variables in the same cluster have their coefficients

[8] Tibshirani, Saunders, Rosset, Zhu, and Knight, 2005
[9] She et al., 2010
[10] Bondell and Reich, 2008
[11] Sharma, Bondell, and Zhang, 2013
[12] Daye and Jeng, 2009
[13] Shen, Huang, and Pan, 2012
[14] Kim and Xing, 2009
[15] Yang, Yuan, Lai, Shen, Wonka, and Ye, 2012
[16] Yang, Yuan, Lai, Shen, Wonka, and Ye, 2012
[17] Witten, Shojaie, and Zhang, 2014

pushed towards each other in absolute value, especially if they have high absolute correlation.

CEN differs from the other methods with respect to two important aspects. First, the grouping penalty does not require summation over pairs of variables, simplifying the calculations necessary. Second, CEN simultaneously optimizes the regression coefficients and cluster assignments, rather than defining the clusters after estimating the coefficients.

Let the variable partition $\mathcal{C} := \{C_1, ..., C_K\}$. The objective function for CEN is

Cluster Elastic Net (CEN)

$$\begin{aligned} J(\boldsymbol{b}, \mathcal{C}) =& \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{b}||_2^2 + \text{Penalty}_S(\boldsymbol{b}; \delta) + \text{Penalty}_G(\boldsymbol{b}, \mathcal{C}, \mathbf{X}; \lambda) \\ =& \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{b}||_2^2 + \delta \sum_{j=1}^{p} |b_j| + \lambda \sum_{k=1}^{K} \sum_{j \in C_k} \|\mathbf{x}_j b_j - \underbrace{\frac{1}{p_k} \sum_{\ell \in C_k} \mathbf{x}_\ell b_\ell}_{\text{centroid } k}\|_2^2, \end{aligned} \tag{4.30}$$

where p_k is the number of variables in cluster k.

For fixed $\boldsymbol{b}$, the CEN problem reduces to minimizing the grouping penalty function with respect to $\mathcal{C}$. This is equivalent to clustering the coefficient-weighted variables $\mathbf{x}_1 b_1, ..., \mathbf{x}_p b_p$ using K-means. As we will demonstrate below, this weighted K-means problem is closely related to our proposed Weighted SOS-NMF problem. Let $\mathbf{H}$ $(p \times K)$ be a binary matrix with orthogonal columns, and let $C_k := \{j \mid h_{jk} = 1\}$. Given a vector $\boldsymbol{b}$ of regression coefficients, the CEN solution for $C_1, ..., C_K$ can also be found by estimating $\mathbf{H}$ and inferring the clusters from it. The solution for $\mathbf{H}$ is given by

Weighted K-means Variable Clustering

$$\begin{aligned} \hat{\mathbf{H}} = \underset{\mathbf{H}}{\arg\min}\ & ||\mathbf{X}\text{diag}(\boldsymbol{b}) - \mathbf{X}\text{diag}(\boldsymbol{b})\mathbf{H}(\mathbf{H}^\top\mathbf{H})^{-1}\mathbf{H}^\top||_2^2 \\ & \mathbf{H} \in \{0,1\}^{p \times K} \\ & \mathbf{H} \text{ has orthogonal columns,} \end{aligned} \tag{4.31}$$

where $\text{diag}(\boldsymbol{b})$ is a diagonal matrix whose diagonal is composed of the vector $\boldsymbol{b}$ of coefficients. The CEN solution for $C_1, ..., C_K$ can be found by setting $\hat{C}_k := \{j \mid \hat{h}_{jk} = 1\}$, $\forall k$. The proof of this proposition is presented in the Appendix (Section 4.7.2).

Clearly, the problem in Eq. (4.31) is similar to the Weighted SOS-NMF problem, with a few exceptions: (i) the cluster membership matrix $\mathbf{H}$ is binary and (ii) the vector $\boldsymbol{b}$ used to weight the variables in $\mathbf{X}$ is the minimizer of Eq. 4.3.2 for fixed $\mathcal{C}$, not a fixed input. For fixed $\mathcal{C}$ and b_ℓ, $\ell \neq j$ and $j \in C_k$, the minimizer $\hat{b}_j$ of Eq. 4.3.2 is

$$\hat{b}_j = \frac{\text{Soft}(\mathbf{e}_j^\top \mathbf{x}_j + \frac{\lambda}{p_k} \sum_{\ell \in C_k, \ell \neq j} b_\ell \mathbf{x}_j^\top \mathbf{x}_\ell, \delta)}{\mathbf{x}_j^\top \mathbf{x}_j (1 + \frac{\lambda(p_k - 1)}{pk})}, \tag{4.32}$$

where $\mathbf{e}_j = \mathbf{y} - \sum_{\ell \neq j} \mathbf{x}_\ell b_\ell$ and $\text{Soft}(z, \delta) = \text{sign}(z)(|z| - \delta)_+$.

As λ decreases, the vector $\boldsymbol{b}$ becomes sparser. However, less importance is also given to the variable clustering task, resulting in weaker clustering performance. Therefore, unlike Weighted SOS-NMF, CEN generally achieves better clustering performance for less sparse models, regardless of the true underlying model complexity. This makes it difficult for CEN to simultaneously maximize the quality of variable selection and variable clustering.

4.3.3 Ensemble Methods and SRR

Another approach to the supervised variable clustering problem is to simultaneously estimate K models, one for each cluster, then predict the target based on the average coefficient vector across clusters. Split Regularized Regression (SRR) (Christidis, Lakshmanan, Smucler, and Zamar, 2020) takes this "ensemble-method" approach. Rather than estimating a single vector of coefficients, SRR estimates multiple sparse coefficient vectors and encourages these vectors to be as diverse as possible. Although SRR does not explicitly search for variable clusters, they may be inferred from the coefficient vectors: a variable belongs to a given cluster if its coefficient in the corresponding coefficient vector is nonzero. This approach can identify overlapping clusters and does not force coefficients of variables in the same cluster to tend towards the same value.

The objective function for SRR is

Split Regularized Regression (SRR)

$$J(\boldsymbol{b}_1, ..., \boldsymbol{b}_K) = \sum_{k=1}^{K} \left\{ \frac{1}{2n} ||\mathbf{y} - \mathbf{X}\boldsymbol{b}_k||_2^2 + \delta \left[\alpha \sum_{j=1}^{p} |b_{jk}| + (1-\alpha) \sum_{j=1}^{p} b_{jk}^2 \right] + \lambda \sum_{g \neq k} \sum_{j=1}^{p} |b_{jk}||b_{jg}| \right\}. \tag{4.33}$$

The matrix $\mathbf{B} = [\boldsymbol{b}_1 \; ... \; \boldsymbol{b}_K]$ is essentially a cluster membership matrix, where variable j belongs to cluster k if $b_{jk} \neq 0$. Variables can belong to multiple clusters, but "hard" clusters $C_1, ..., C_K$ can also be defined such that $C_k := \{j \mid k = \arg\max_\ell |b_{j\ell}|\}$. Maximal diversity is achieved when $\mathbf{B}$ is orthogonal and thus each variable belongs to a single cluster (i.e. $|b_{jk}||b_{jg}| = 0 \; \forall j, k, g$).

The final vector of regression coefficients used for prediction is an average across all vectors $\boldsymbol{b}_k$:

$$\overline{\boldsymbol{b}} = \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{b}_k. \tag{4.34}$$

Like many of the coefficient-grouping methods, cluster assignment is more of a side effect than a principal objective of SRR. Moreover, the best solutions in terms of prediction error tend to be models with a low degree of sparsity and extensive cluster overlap. As a result, variables are assigned to clusters with less certainty, and the clustering performance is less optimal.

4.3.4 Two-Step Methods

The methods described in Sections 4.3.2 and 4.3.3 are *embedded* methods because model estimation and clustering are performed simultaneously. As a result, the tasks of clustering and modeling are inherently dependent. For two-step methods, these tasks are separated: first, clusters are identified using an unsupervised clustering technique (Step 1), then a model is estimated based on the outputs of the clustering algorithm (Step 2).

One advantage of two-step methods is their computational efficiency compared to embedded methods. It is also easier to adapt them to different types of prediction tasks, such as regression, classification and survival analysis. Indeed, the outputs of the clustering algorithm can simply be used as inputs to existing prediction

algorithms. However, unlike VC-PCR, most two-step methods perform clustering without any supervision by the target. As a result, they cannot distinguish variables that are relevant for prediction from irrelevant ones. This is a significant limitation for these methods, as relevant and irrelevant variables may be mixed into the same cluster, thereby diminishing subsequent prediction performance.

Different unsupervised clustering algorithms have been proposed for the first step of the two-step approach, the most common being K-means (used by e.g. Ma, Song, and Huang (2007)) and hierarchical agglomerative clustering (HAC) (used by e.g. Bühlmann, Rütimann, Geer, and Zhang (2013) and Park, Hastie, and Tibshirani (2007)). A variety of different strategies for the second step have also been proposed. These strategies are summarized below.

Cluster Representative Lasso (CRL)

For Cluster Representative Lasso (CRL) (Bühlmann, Rütimann, Geer, and Zhang, 2013; Park, Hastie, and Tibshirani, 2007), Lasso is used to estimate a sparse model in the second step, where the cluster centroids $\hat{\mathbf{m}}_1, ..., \hat{\mathbf{m}}_K$ from first step are the predictors. The Lasso solution is given by

$$\arg\min_{\mathbf{a}} L(\hat{\mathbf{M}}, \mathbf{y}; \mathbf{a}) + \delta||\mathbf{a}||_1, \tag{4.35}$$

where $L(\hat{\mathbf{M}}, \mathbf{y}; \mathbf{a})$ is a loss function, $\mathbf{a}$ $(K \times 1)$ contains the model coefficients and δ is a nonnegative hyperparameter for the Lasso penalty.

Performing regression or classification based on the K cluster centroids, rather than the p variables in $\mathbf{X}$, can be advantageous because K is typically much smaller than p. This decreases the dimensionality of the problem, making it less prone to the curse of dimensionality.

Cluster Group Lasso (CGL)

For Cluster Group Lasso (CGL) (Bühlmann, Rütimann, Geer, and Zhang, 2013), Group Lasso (Yuan and Lin, 2006) is used in the second step. The variables in $\mathbf{X}$, rather the cluster centroids, are used as predictors, and the clusters $\hat{C}_1, ..., \hat{C}_K$ are used to define the groups used in the grouping penalty. The Group Lasso solution is given by

$$\arg\min_{\mathbf{b}} L(\mathbf{y}, \mathbf{X}, \mathbf{b}) + \lambda \sum_{k=1}^{K} \sqrt{\sum_{j \in \hat{C}_k} b_j^2}, \tag{4.36}$$

where $L(\mathbf{y}, \mathbf{X}, \mathbf{b})$ is a loss function, $\mathbf{b}$ $(p \times 1)$ contains the model coefficients, λ is a nonnegative hyperparameter for the Group Lasso penalty and $\hat{C}_k$ is the k^{th} cluster identified in the first step. For each cluster $\hat{C}_k$, the Group Lasso penalty results in

either the selection of all of the variables in the cluster (i.e. they all have nonzero coefficients) or none of them (i.e. they all have null coefficients).

Supervised Group Lasso (SGLasso)

An alternative to CRL and CGL is to estimate independent Lasso models for each cluster identified in step 1. This strategy, which allows for feature selection within clusters, is used for Supervised Group Lasso (SGLasso) (Ma, Song, and Huang, 2007). SGLasso also performs a third step where these sparse clusters are used to define the groups in Group Lasso (Yuan and Lin, 2006). The Group Lasso step is necessary for selecting the clusters that are important for predicting the response.

4.4 Experiments

Experiments with simulated data are performed to compare the performance of the proposed method, VC-PCR, with related methods from the literature. Performance in terms of model error and complexity, as well as model support and cluster recovery, is evaluated for all methods on simulated datasets. These simulated datasets vary with respect to the correlation structure among predictor variables. For all data configurations and performance criteria, VC-PCR with Ridge weights performs the best of all methods tested.

4.4.1 Methods Compared

Cluster Elastic Net (CEN) (Witten, Shojaie, and Zhang, 2014) is chosen to represent the coefficient-grouping methods because of its computational efficiency compared to the other methods in this category. Split Regularized Regression (SRR) (Christidis, Lakshmanan, Smucler, and Zamar, 2020), the only ensemble method that can be used to cluster variables, is also included. Of the different two-step approaches, Cluster Representative Lasso (CRL) (Bühlmann, Rütimann, Geer, and Zhang, 2013) is tested because it is the most similar to the proposed method. For CRL, three commonly used clustering algorithms are tested: hierarchical agglomerative clustering (HAC) based on Ward's criterion (CRL-hclustCor), HAC based on canonical correlation, as seen in Bühlmann, Rütimann, Geer, and Zhang (2013) (CRL-hclustCC), and K-means clustering (CRL-kmeans). For VC-PCR, three choices of weight matrices $\mathbf{W}$ are tested: a diagonal matrix of Lasso regression coefficients (VC-PCR-Lasso), a diagonal matrix of Ridge regression coefficients (VC-PCR-Ridge) and an identity matrix (VC-PCR-Identity). This last choice serves as a benchmark for assessing the added-value of variable-weighting in VC-PCR.

4.4.2 Simulated Data

The simulated data are generated as follows. For each observation i, a vector $\mathbf{x}_i$ is generated from the distribution $N_p(\mathbf{0}, \mathbf{\Sigma})$ and a model error ε_i is generated from $N(0, \sigma_\varepsilon^2)$, where $\mathbf{\Sigma}$ is covariance matrix for $\mathbf{X}$ and σ_ε^2 is the model noise variance. The response y_i is calculated as $y_i = \mathbf{x}_i^\top \boldsymbol{b} + \varepsilon_i$, where $\boldsymbol{b}$ is the vector of true model coefficients, and p, the number of variables, is fixed to 200.

True Coefficients

The vector of true coefficients is fixed for all experiments, meaning that the set of relevant or "active" variables $\mathcal{S} := \{j \mid b_j \neq 0\}$ is the same for all datasets. The true vector of coefficients is

$$\boldsymbol{b} = (\underbrace{1, ..., 1}_{5}, \underbrace{0, ..., 0}_{5}, \underbrace{-1, ..., -1}_{5}, \underbrace{0, ..., 0}_{5}, \underbrace{1, ..., 1}_{5}, \underbrace{0, ..., 0}_{5}, \underbrace{-1, ..., -1}_{5}, \underbrace{0, ..., 0}_{5}, \underbrace{0, ..., 0}_{160})^\top. \tag{4.37}$$

Thus, four blocks of variables are active in the model.

Covariance Matrix

Various different covariance matrices $\mathbf{\Sigma}$ are used to asses how correlation structure among predictor variables impacts performance. The theoretical variance of all variables is set to one. Active variables are split into R sets $G_1, ..., G_R$ such that for each pair of variables $j, \ell \in \mathcal{S}, j \neq \ell$,

$$\sigma_{j\ell} = \begin{cases} \rho & \exists r \text{ s.t. } j, \ell \in G_r \\ 0 & \exists r \neq r' \text{ s.t. } j \in G_r, \ell \in G_{r'}. \end{cases} \tag{4.38}$$

As a result, variables belonging to different sets G_r and $G_{r'}$ are uncorrelated and variables belonging to the same set have a correlation of ρ.

Previous works in the literature have tested different types of correlation structure for inactive variables, but they have not compared these different types in a single study. For inactive variables (i.e. $b_j = 0$), we test three common configurations:

1. Some inactive variables are correlated (correlation of ρ) with both active variables and other inactive variables (set-up seen in Witten, Shojaie, and Zhang (2014))
2. Some inactive variables are correlated (correlation of ρ) with each other but not with active variables (set-up seen in Shen, Huang, and Pan (2012))

3. Inactive variables are uncorrelated with all other variables (set-up seen in Sharma, Bondell, and Zhang (2013))

These three configurations are summarized in Figure 4.1. For each configuration, horizontal bars are used to indicate the true cluster assignments and coefficients for the $p = 200$ variables, where each variable is represented by a colored vertical line within these bars. The square box below is a heatmap of the covariance matrix, $\mathbf{\Sigma}$. The colors for the coefficients and the covariance heatmap represent continuous values (see the legend), and the colors for cluster assignments are arbitrary, representing categories, i.e. the different clusters $k \in \{1, ..., K\}$. While the true cluster assignments and coefficients are the same for all configurations, the correlation structure, as seen in the heatmaps of the covariance matrices, varies from one configuration to the next.

True Clusters

It is common practice to consider that all inactive variables belong to the same cluster. Moreover, two active variables j and ℓ are assumed to be in the same cluster if $|\text{Cor}(X_j, X_\ell)|$ is large and $|b_j| \approx |b_\ell|$ (see Definition 2.1.1 from Section 2.1). Therefore, in this experimental set-up, there are $K = R + 1$ true variable clusters C_k, one cluster for each set G_r of active variables and one cluster for the set of inactive variables $\mathcal{S}^c$. The true clusters are the same for each configuration, as seen in Figure 4.1.

Variance of the Model Error

For each configuration, the variance of the model error, σ_ε^2, is fixed such that all configurations have the same model signal-to-noise ratio (SNR). The SNR is defined as

$$SNR = \frac{\boldsymbol{b}^\top \mathbf{\Sigma} \boldsymbol{b}}{\sigma_\varepsilon^2}. \tag{4.39}$$

Summary of Fixed Parameters

In addition to the true model coefficients and clusters, several parameters are fixed for all configurations. These parameters are summarized in Table 4.3.

4.4.3 Evaluation Criteria

The performance of each method is evaluated with respect to four different characteristics: the model error, the model complexity, the recovery of the model support and the recovery of the underlying clusters. The model error is quantified with

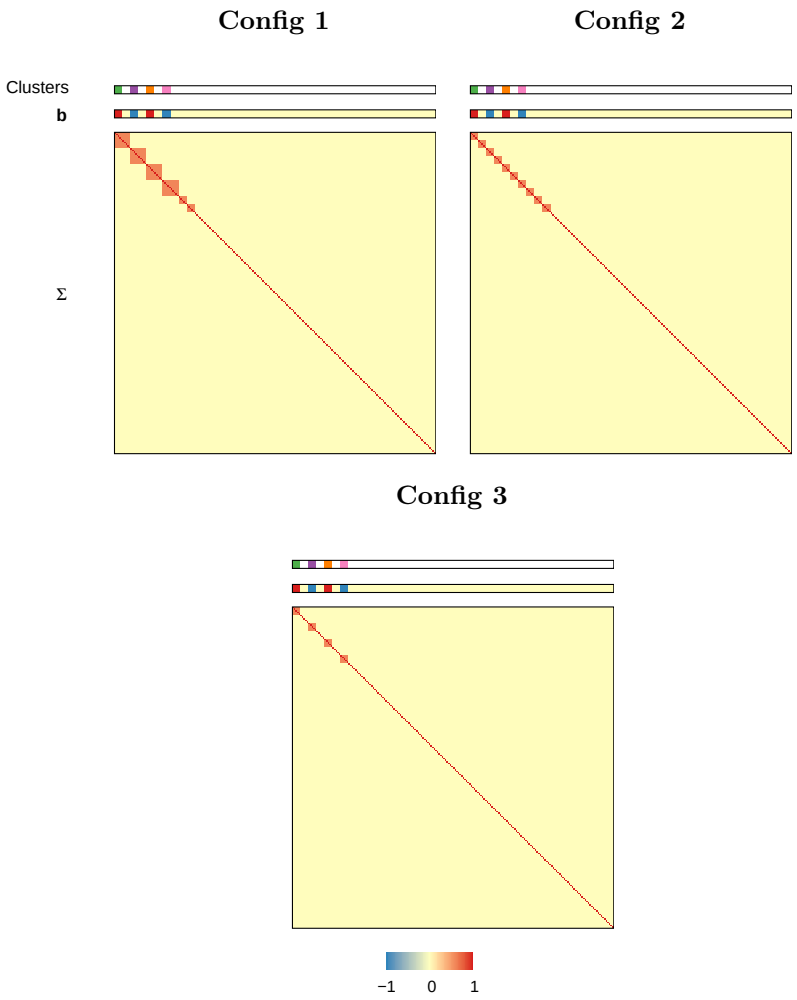


Figure 4.1: **Experimental configurations for the simulation study.** For each configuration, the first and second horizontal bars depict the true cluster assignments and coefficients for the $p = 200$ variables. The square box below is a heatmap of the covariance matrix, $\mathbf{\Sigma}$. The colors for the coefficients and covariance heatmap represent continuous values (see the legend), and the colors for cluster assignments are arbitrary, representing categories, i.e. the different clusters $k \in \{1, ..., K\}$.

Parameter	Value	Meaning
p	200	number of variables
n	50	number of observations
K	5	number of true variable clusters
SNR	10	model signal-to-noise ratio
ρ	0.6	correlation between correlated variables

Table 4.3: **Summary of Fixed Simulation Parameters.**

the Mean Squared Error of Prediction (MSEP). Given a vector $\hat{\mathbf{y}}$ ($n_{test} \times 1$) of predicted response values,

$$MSEP = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} (y_i - \hat{y}_i)^2. \tag{4.40}$$

The model complexity is quantified as the number s of active variables in the model. For methods that estimate one coefficient $\hat{b}_j$ per variable j (e.g. CEN, SRR, etc.),

$$s = ||\hat{\boldsymbol{b}}||_0 = \sum_{j=1}^{p} \mathbb{I}\left[\hat{b}_j \neq 0\right]. \tag{4.41}$$

For methods that estimate one coefficient $\hat{a}_k$ per cluster k (e.g. CRL, etc.),

$$s = \sum_{k=1}^{K} \hat{p}_k \mathbb{I}\left[\hat{a}_k \neq 0\right], \tag{4.42}$$

where $\hat{p}_k$ is the number of variables in $\hat{C}_k$.

The Matthews Correlation Coefficient (MCC) (Matthews, 1975) is used to quantify the recovery of the true model support and underlying clusters. Compared to other existing criteria for evaluating binary classification, MCC is considered to be less biased and is more robust to class imbalance (Chicco, 2017; Chicco and Jurman, 2020; Chicco, Tötsch, and Jurman, 2021). Given the number of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN),

$$MCC = \frac{(TP)(TN) - (FP)(FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \in [-1, 1], \tag{4.43}$$

where a score of 1 represents perfect classification and a score of 0 is equivalent to random classification.

When evaluating model support recovery, variables with nonzero coefficients are assigned to the "positive" class and variables with null coefficients are assigned to the "negative class." Cluster recovery is evaluated for all unique pairs of variables. Pairs of variables assigned to the same cluster are in the "positive class" and all other pairs of variables are in the "negative" class.

4.4.4 Hyperparameter Selection

The methods compared in this study depend on the choice of several hyperparameters: δ, λ and K for VC-PCR and the embedded methods and δ and K for the CRL methods. For SRR, the fourth hyperparameter α is fixed to 1, resulting in a Lasso penalty (see Eq. (4.33)). For the hyperparameters δ and λ, 10 values are tested for each method. For K, whose true value is 5, three values are tested: $K \in \{4, 5, 6\}$. Some methods, such as CRL-kmeans, CEN and VC-PCR, require an initial set of clusters. Five random initializations are tested for these methods, and the resulting performance is subsequently averaged (see Section 4.4.5).

For regression problems, most authors in the literature (e.g. Bondell and Reich (2008), Daye and Jeng (2009), Shen and Huang (2010), Shen, Huang, and Pan (2012), Witten, Shojaie, and Zhang (2014), and Christidis, Lakshmanan, Smucler, and Zamar (2020)) select hyperparameters based on the minimum cross-validated Mean Squared Error of Prediction (MSEP). Some authors (e.g. Park, Hastie, and Tibshirani (2007)) suggest choosing the hyperparameters according to the "one standard error rule," whereby the sparsest model within one-standard error of the minimum cross-validated MSEP is chosen. Other criteria for selecting hyperparameters have also been used, such as the BIC (Sharma, Bondell, and Zhang, 2013). In the following experiments, the minimum cross-validated MSEP is used for hyperparameter selection, as it is the most common approach in the literature.

Nested k-fold cross-validation is used to select hyperparameters and evaluate average out-of-sample performance. This procedure involves k-fold cross-validation at two levels: an inner and an outer loop. The dataset is split into 10 outer folds. Then, for each iteration m of the outer loop, the samples in outer fold m are set aside and 5-fold cross-validation (the nested inner loop) is performed on the remaining samples. The latter samples are normalized before analysis: the target and predictors are centered and scaled to unit variance. The hyperparameters yielding the model with the smallest average MSEP across the inner folds are then selected and a final model is trained using all samples in the inner folds, which are also normalized. The performance of this model is evaluated using the samples in outer fold m, which were not used during model selection or estimation. These samples are normalized using the same parameters used to normalize the samples that trained the model. After iterating through all of the outer folds, the average out-of-sample performance is then calculated.

4.4.5 Results

For each data configuration, the model error, cluster recovery and support recovery are assessed for models with different complexities s. For each method, all but one hyperparameter are tuned using the cross-validation strategy explained in Section 4.4.4, and the average out-of-sample performance is calculated for each value of the remaining hyperparameter. The latter hyperparameter is the one that most impacts model complexity: δ for the CRL and embedded methods and λ for the VC-PCR methods. For methods requiring cluster initializations, performance is further averaged over all random initializations. In Figure 4.2, the average out-of-sample performance is plotted with respect to s, where each point in the lines corresponds to a different value of the fixed hyperparameter (δ or λ). The vertical black line corresponds to the true number of active variables, 20. For VC-PCR-Lasso, the plotted line does not cover the entire domain of model sizes s. This is because VC-PCR-Lasso depends on two sparsity parameters: δ for the Lasso regression used to generate the weight matrix for Weighted SOS-NMF and λ for the sparsity penalty in Weighted SOS-NMF. For the plots depicted in Figure 4.2, only one of these hyperparameters is fixed, while the other is tuned using cross-validation. Clearly, hyperparameters yielding larger models are never selected.

For all of the configurations and performance criteria, VC-PCR-Ridge outperforms the other methods tested. For VC-PCR-Ridge, the best solutions in terms of clustering and support recovery are models with a complexity s close to the true number of active variables. Moreover, unlike the CRL and embedded methods, the model error for VC-PCR-Ridge is robust to the type of configuration. VC-PCR-Lasso demonstrates a similar level of robustness to configuration type, but its performance is slightly inferior to VC-PCR-Ridge. In contrast to these two methods, the model error for VC-PCR-Identity greatly depends on the configuration type. In addition, the clustering and support recovery are much worse. This demonstrates the added-value of using non-uniform weights in Weighted SOS-NMF, the first step of VC-PCR.

The two-step CRL methods, which depend on unsupervised clustering, perform the worst of all methods tested. Among CRL methods, CRL-hclustCor performs best, particularly in configuration 1, where some inactive variables are correlated with both active and inactive variables. Among the embedded methods, SRR generally achieves the best model error, CEN has the best cluster recovery and both methods are comparable in terms of support recovery. The best cluster recovery for CEN is comparable to that of VC-PCR-Lasso, but it requires a more complex model.

For VC-PCR and the embedded methods, the most favorable configuration for cluster and support recovery is configuration 2. This configuration includes more correlated variables than configuration 3, and, in contrast to configuration 1, the inactive variables are not correlated with active variables. This suggests that, for cluster and support recovery, correlation can be beneficial as long as there is little

to no correlation between inactive and active variables. For model error, the VC-PCR and embedded methods perform best in configuration 3, where few variables are correlated. This confirms the idea that high correlation between variables tends to increase estimation variance, and thus decrease model accuracy.

Of the methods tested, CEN, CRL-kmeans and the VC-PCR methods all depend on an initial set of clusters. Compared to the two-step methods (CRL-kmeans and VC-PCR), the prediction and support recovery of CEN is generally less sensitive to initialization (this was observed in a comparative study). The clustering performance of CEN is also relatively robust to initialization, but its sensitivity tends to increase as the model becomes larger (smaller values of the sparsity hyperparameter δ) and/or the number of correlated variables increases. Despite its greater sensitivity to initialization, VC-PCR-Ridge still achieves the best prediction, support recovery and clustering performance overall, even when considering less favorable initializations.

4.5 Discussion and Conclusions

This chapter presented a two-step method for prediction and supervised variable clustering, Variable Cluster Principal Component Regression (VC-PCR). In the first step, VC-PCR identifies variable clusters and performs variable selection via Weighted SOS-NMF. The weights used for Weighted SOS-NMF —which encode the prior relationship between the predictors and the target —make it possible to supervise both the clustering and the variable selection. In the second step, the cluster membership degrees from the first step are used to generate latent variables, and a linear model is estimated to relate the latent variables to the target.

Unlike other two-step methods, the clustering step in VC-PCR is supervised. As a result, the identified clusters tend to be more homogeneous: active and inactive variables are less likely to be mixed together. Compared to embedded methods, VC-PCR is more computationally efficient, and it can be easily applied to a variety of tasks (regression, classification, survival analysis, etc.), not just regression. Therefore, VC-PCR embodies the "best of both worlds", having the strengths of both the two-step and embedded approaches.

Numerical experiments using simulated data were performed to compare VC-PCR to competitors from the literature. For all data configurations, VC-PCR with Ridge regression weights (VC-PCR-Ridge) had the best model error, support recovery and clustering performance. Although conceptually similar to the embedded method Cluster Elastic Net (CEN), VC-PCR had better performance for sparser models. Therefore, both performance and interpretability can be improved by using VC-PCR.

VC-PCR, as well as several other competing methods, depends on an initial set of clusters. Future works could explore initialization strategies that improve per-

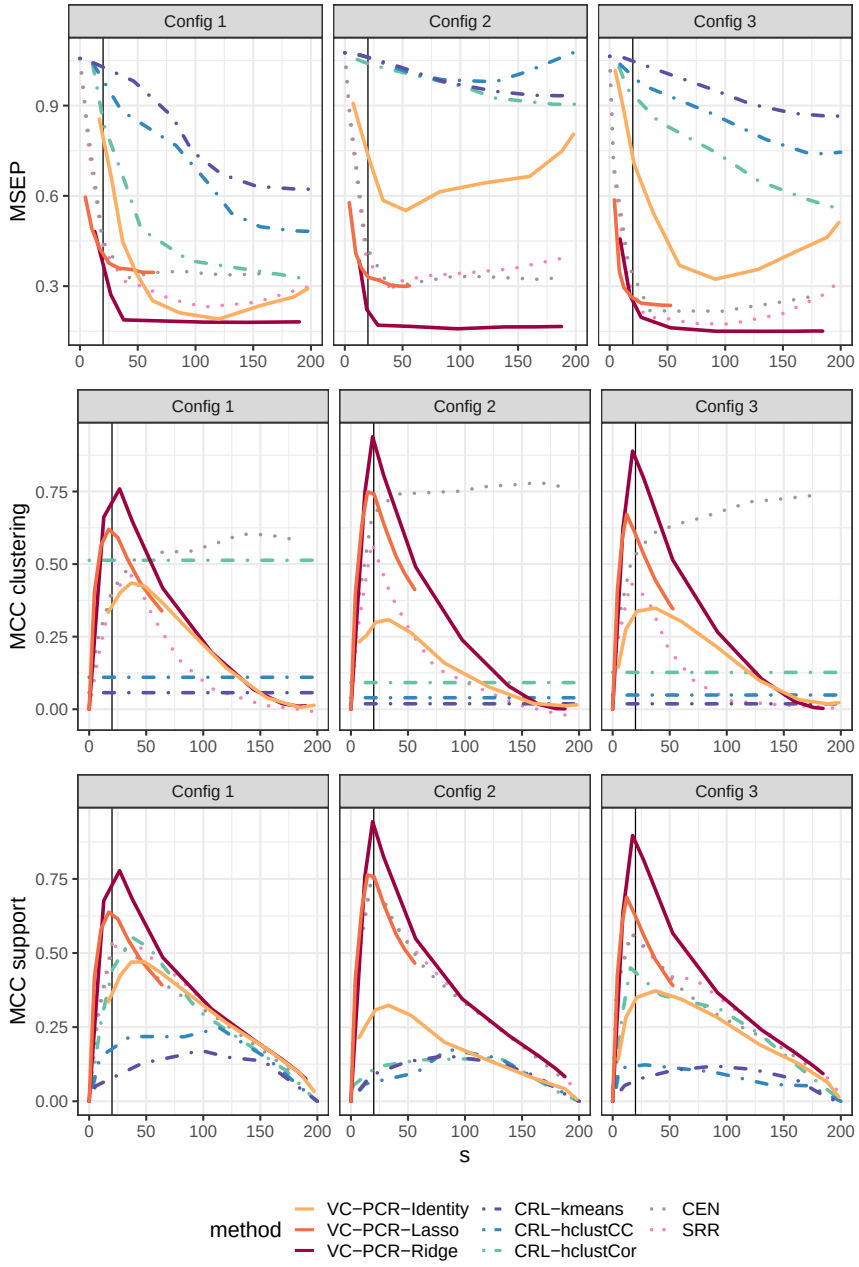


Figure 4.2: **Average out-of-sample performance.** Each point in the plotted lines represents a value of the hyperparameter (HP) δ or λ; all other HPs were tuned using k-fold cross-validation. The vertical line indicates the true number of active variables, 20.

formance with respect to random initializations, potentially drawing on existing research on matrix factorization (e.g. Langville, Meyer, Albright, Cox, and Duling (2006) and Atif, Qazi, and Gillis (2019)). Another avenue for further research is the creation of a measure that, given a target, approximates the number of underlying predictive clusters. This would facilitate the exploration of the hyperparameter space for K, the number of clusters in the initial partition.

4.6 Numerical Implementation

The experiments from this chapter were programmed in R. Weighted SOS-NMF, VC-PCR and SRR were implemented using in-house code. CEN was performed using code provided by the authors. Lasso and Ridge regression were performed using the function *glmnet* from the `glmnet` package. For the CRL methods, Ward's hierarchical clustering and K-means clustering were performed using the functions *hclust* and *kmeans* from the `stats` package. Hierarchical clustering based on canonical correlation was performed using the function *hclustvar* from the `ClustOfVar` package.

Of the methods compared, Cluster Representative Lasso (CRL) with K-means clustering (CRL-kmeans) and hierarchical clustering based on Ward's criterion (CRL-hclustCor) were the fastest methods (faster than VC-PCR by a factor of 16 and 7 respectively). This is likely due to the fact that the *kmeans* function from the `stats` package is partly coded in C and the *hclust* function is partly coded in Fortran. Compared to VC-PCR, Cluster Elastic Net (CEN) was 4 times slower, CRL with hierarchical clustering based on canonical correlation (CRL-hclustCC) was 7 times slower and Split Regularized Regression (SRR) was several 100 times slower.

4.7 Appendix

4.7.1 Orthonormal Nonnegative Sparse PCA

As mentioned in Section 4.3.1, Nonnegative Sparse PCA with $\gamma = \infty$ only differs from SOS-NMF with respect to the scaling constraints on $\mathbf{U}$ and $\mathbf{V}$: SOS-NMF constrains the scale of $\mathbf{U}$ whereas NSPCA constrains the scale of $\mathbf{V}$.

Proof. For $\gamma = \infty$, the NSPCA problem can be rewritten as

Orthonormal NSPCA

$$\begin{aligned} \arg\max_{\mathbf{V}} \ & \frac{1}{2(n-1)}||\mathbf{XV}(\mathbf{V}^\top\mathbf{V})^{-1/2}||_F^2 - \lambda\sum_{j=1}^{p}||\mathbf{v}_j||_1 \\ & \text{s.t. } v_{jk} \geq 0, \ \forall j,k \\ & \mathbf{V}^\top\mathbf{V} = \mathbf{I}_K, \end{aligned} \tag{4.44}$$

where $||\mathbf{XV}(\mathbf{V}^\top\mathbf{V})^{-1/2}||_F^2 = ||\mathbf{XV}||_F^2$ because $\mathbf{V}^\top\mathbf{V} = \mathbf{I}_K$. This can also be rewritten as a minimization problem similar to the one in Eq. (4.6). Note that

$$\begin{aligned} ||\mathbf{X} - \mathbf{XV}(\mathbf{V}^\top\mathbf{V})^{-1}\mathbf{V}^\top||_F^2 &= \text{Tr}\Big(\mathbf{XX}^\top - \mathbf{XV}(\mathbf{V}^\top\mathbf{V})^{-1}\mathbf{V}^\top\mathbf{X}\Big) \\ &= ||\mathbf{X}||_F^2 - ||\mathbf{XV}(\mathbf{V}^\top\mathbf{V})^{-1/2}||_F^2. \end{aligned} \tag{4.45}$$

Therefore, the solution represented in Eq. (4.44) is the same as the solution

$$\begin{aligned} \arg\min_{\mathbf{V}} \ & \frac{1}{2(n-1)}||\mathbf{X} - \mathbf{XV}(\mathbf{V}^\top\mathbf{V})^{-1}\mathbf{V}^\top||_F^2 + \lambda\sum_{j=1}^{p}||\mathbf{v}_j||_1 \\ & \text{s.t. } v_{jk} \geq 0, \ \forall j,k \\ & \mathbf{V}^\top\mathbf{V} = \mathbf{I}_K. \end{aligned} \tag{4.46}$$

Eq. (4.46) can also be rewritten as a matrix factorization problem as follows:

Orthonormal NSPCA (Reformulation)

$$\begin{aligned} \arg\min_{\mathbf{U},\mathbf{V}} \ & \frac{1}{2(n-1)}||\mathbf{X} - \mathbf{UV}^\top||_F^2 + \lambda\sum_{j=1}^{p}||\mathbf{v}_j||_1 \\ & \text{s.t. } v_{jk} \geq 0, \ \forall j,k \\ & \mathbf{V}^\top\mathbf{V} = \mathbf{I}_K, \end{aligned} \tag{4.47}$$

where $\mathbf{U}$ $(n \times K)$ is a matrix of latent variables. For fixed $\mathbf{V}$, the solution $\hat{\mathbf{U}} = \mathbf{XV}(\mathbf{V}^\top\mathbf{V})^{-1}$. Replacing the $\mathbf{U}$ in Eq. (4.47) with $\hat{\mathbf{U}}$ results in the optimization problem represented in Eq. (4.46), as

$$||\mathbf{X} - \hat{\mathbf{U}}\mathbf{V}^\top||_F^2 = ||\mathbf{X} - \mathbf{XV}(\mathbf{V}^\top\mathbf{V})^{-1}\mathbf{V}^\top||_F^2. \tag{4.48}$$

Therefore the solution for $\mathbf{V}$ in Eq. (4.47) is indeed the same as the solution in Eq. (4.46). Clearly, this problem is equivalent to the SOS-NMF problem except for the fact that the scale of the columns of $\mathbf{V}$, rather than $\mathbf{U}$, is fixed. □

4.7.2 Cluster Elastic Net Reformulation

As mentioned in Section 4.3.2, the CEN solution for $C_1, ..., C_K$ can also be found by estimating a binary matrix $\mathbf{H}$ $(p \times K)$ and inferring clusters from it. For a fixed vector $\boldsymbol{b}$ of regression coefficients, the solution for $\mathbf{H}$ is given by

$$\begin{aligned} \hat{\mathbf{H}} = \arg\min_{\mathbf{H}} \; & ||\mathbf{X}\text{diag}(\boldsymbol{b}) - \mathbf{X}\text{diag}(\boldsymbol{b})\mathbf{H}(\mathbf{H}^\top\mathbf{H})^{-1}\mathbf{H}^\top||_2^2 \\ & \mathbf{H} \in \{0,1\}^{p\times K} \\ & \mathbf{H} \text{ has orthogonal columns,} \end{aligned} \tag{4.49}$$

where $\text{diag}(\boldsymbol{b})$ is a diagonal matrix whose diagonal is composed of the vector $\boldsymbol{b}$ of coefficients. The CEN solution for $C_1, ..., C_K$ can be found by setting $\hat{C}_k := \{j \mid \hat{h}_{jk} = 1\}$, $\forall k$.

Proof. As explained in Section 2.3.1, the objective function for K-means variable clustering applied to a matrix $\mathbf{X}$ is

$$J(\mathcal{C}) = \sum_{k=1}^{K} \sum_{j \in C_k} ||\mathbf{x}_j - \frac{1}{p_k} \sum_{j \in C_k} \mathbf{x}_j||_2^2. \tag{4.50}$$

The minimization problem over $\mathcal{C}$ can be rewritten as a minimization over a binary matrix $\mathbf{H}$ with orthogonal columns, where the solution is

$$\begin{aligned} \hat{\mathbf{H}} = \arg\min_{\mathbf{H}} \; & ||\mathbf{X} - \mathbf{X}\mathbf{H}(\mathbf{H}^\top\mathbf{H})^{-1}\mathbf{H}^\top||_2^2 \\ & \mathbf{H} \in \{0,1\}^{p\times K} \\ & \mathbf{H} \text{ has orthogonal columns,} \end{aligned} \tag{4.51}$$

and the estimated clusters $\hat{C}_k$ are inferred from $\hat{\mathbf{H}}$: $\hat{C}_k := \{j \mid \hat{h}_{jk} = 1\}$.

In the CEN grouping penalty, K-means clustering is performed on the variables $\mathbf{x}_j$ weighted by their respective coefficients b_j. This means that the input to the K-means problem is not $\mathbf{X}$ but rather $\mathbf{X}\text{diag}(\boldsymbol{b})$. Replacing $\mathbf{X}$ in Eq. (4.51) with $\mathbf{X}\text{diag}(\boldsymbol{b})$ results in the solution represented in Eq. (4.31).

□

Chapter 5

Variable Clustering Supervised by Multiple Targets

This chapter describes two methods for variable clustering supervised by a multivariate target: Best Interpretable Rotation (BIR), originally presented in Bibal, Marion, and Frénay (2018) and Marion, Bibal, and Frénay (2019), and Best Interpretable Orthogonal Transformation (BIOT), presented in Bibal, Marion, von Sachs, and Frénay (2021). These methods were developed for the purpose of explaining the low-dimensional representations, or *embeddings*, produced by Multidimensional Scaling (MDS), a dimensionality reduction technique. Given a dataset of external variables (i.e. variables that were not used to produce the MDS embedding), both BIR and BIOT identify clusters of variables that explain the embedding dimensions. Compared to related methods, the models estimated by BIOT are generally the sparsest, for a same level of error.

Adrien Bibal, Rebecca Marion, Rainer von Sachs, and Benoît Frénay (2021). "BIOT: Explaining multidimensional nonlinear MDS embeddings using the Best Interpretable Orthogonal Transformation". In: **Neurocomputing** 453, pp. 109–118

Rebecca Marion, Adrien Bibal, and Benoît Frénay (2019). "BIR: A Method for Selecting the Best Interpretable Multidimensional Scaling Rotation using External Variables". In: **Neurocomputing** 342, pp. 83–96

Adrien Bibal, Rebecca Marion, and Benoît Frénay (2018). "Finding the Most Interpretable MDS Rotation for Sparse Linear Models based on External Features". In: **Proceedings of the European Symposium on Artificial Neural Networks**. Bruges, Belgium, pp. 537–542

List of Symbols

$\mathbf{X}$: $n \times p$ predictor matrix with centered and scaled columns

$\mathbf{Y}$: $n \times K$ matrix of unobserved target variables (centered)

k : index for the target variables $\in \{1, ..., K\}$ in the matrix $\mathbf{Y}$ and for the clusters in $\mathbf{X}$

$\mathbf{Z}$: $n \times q$ matrix of variables used to generate $\mathbf{Y}$

$\mathbf{D}$: $n \times n$ dissimilarity matrix for pairs of observations in $\mathbf{Z}$

$\mathbf{R}^\theta$: $K \times 2$ rotation matrix for a given rotation angle θ

$\mathbf{R}$: $K \times K$ orthogonal transformation matrix, i.e. $\mathbf{R}\mathbf{R}^\top = \mathbf{R}^\top\mathbf{R} = \mathbf{I}_K$

See page 7 for other symbols used in this chapter.

5.1 Introduction

In the previous chapter, a single target vector $\mathbf{y}$ $(n \times 1)$ was used to supervise the clustering of variables in the matrix $\mathbf{X}$ $(n \times p)$, and several clusters of variables were associated with a single target. These clusters were identified based on the correlation between the variables as well as their association with the target. In this chapter, a multivariate target matrix $\mathbf{Y}$ $(n \times K)$ is used to supervise the clustering of variables in $\mathbf{X}$. In contrast to the setting in the previous chapter, a single cluster of variables is associated with each individual target $\mathbf{y}_k$, and a variable may potentially belong to multiple clusters. Thus, the notion of "variable clusters" is slightly different. Cluster k contains the variables with nonzero regression coefficients for target k, i.e. $C_k := \{j \mid b_{jk} \neq 0\}$, where b_{jk} is the regression coefficient for variable j and target k. The number of clusters is therefore equal to the number K of targets.

The following sections summarize the research problem addressed in this chapter, as well as our motivation and contributions.

5.1.1 Problem Definition

We are interested in a data analysis problem involving two observed matrices, $\mathbf{Z}$ $(n \times q)$ and $\mathbf{X}$ $(n \times p)$, that contain q and p variables, respectively, measured for the same set of n observations. These datasets are also called *views*, as they contain distinct sets of variables for the same observations (Sun, 2013).

An example of a common data analysis problem involving two views is depicted in the diagram in Figure 5.1. In this example, drawn from the field of ecology, the matrix $\mathbf{Z}$ (shown in light blue) contains abundance levels for q spider species across

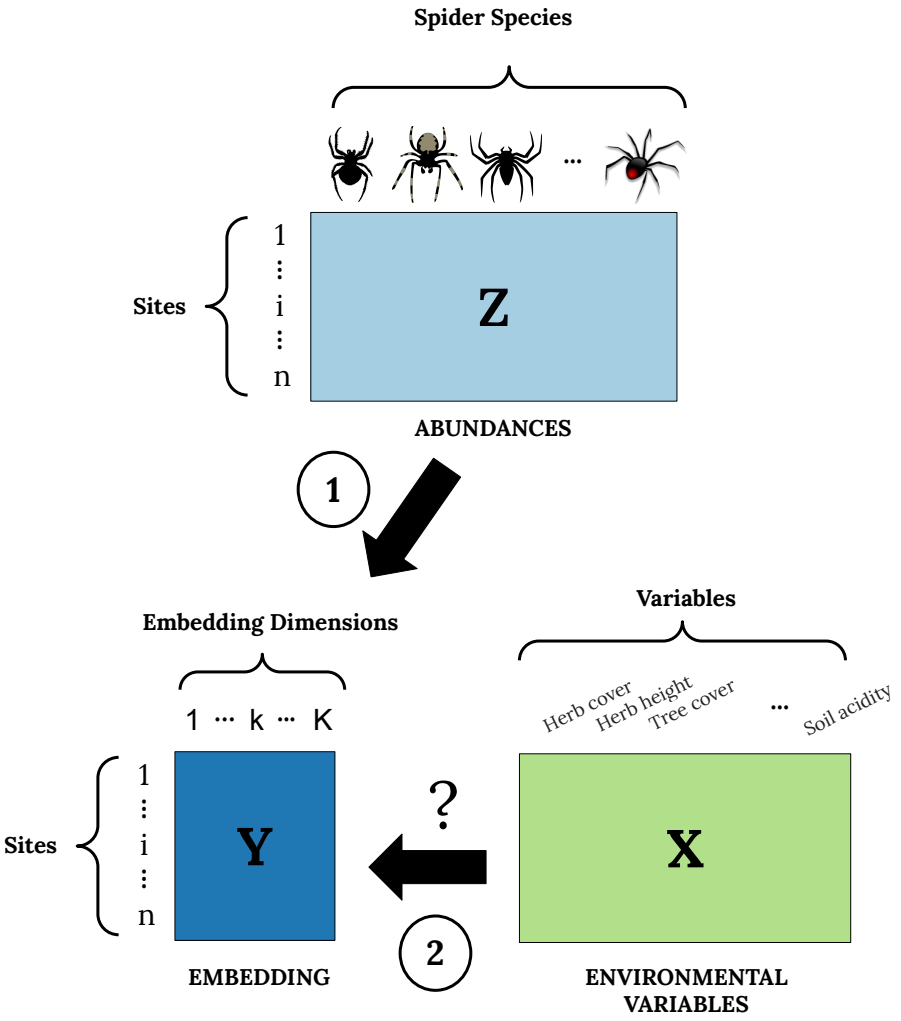


Figure 5.1: **Example of a research problem in ecology.** Abundances of spider species ($\mathbf{Z}$) and environmental variables ($\mathbf{X}$) are measured across n sampling sites. The researcher would like to identify the major patterns in abundance in the spider population as well as explain these patterns with the environmental variables. In a first step (1), a dimensionality reduction technique is applied to $\mathbf{Z}$ to generate a lower-dimensional embedding, $\mathbf{Y}$, whose dimensions characterize orthogonal directions of variation in $\mathbf{Z}$. In a second step (2), a model is built to relate the variables in $\mathbf{X}$ with the dimensions in $\mathbf{Y}$, making it possible to assign meaning to these dimensions.

n sampling sites. The matrix $\mathbf{X}$ (shown in light green) contains a collection of p environmental variables – such as soil acidity and the degree of tree coverage – for the same sites. The objective of the researcher is to use the variables in $\mathbf{X}$ to explain the relationship between observations in $\mathbf{Z}$ (Thioulouse, 2011): which variables in $\mathbf{X}$ can explain the similarity or dissimilarity between certain pairs of observations in $\mathbf{Z}$? Such a research question arises in many fields, including psychology (Jaworska and Chupetlovska-Anastasova, 2009), epidemiology (Lin, Zhu, Knickmeyer, Styner, Gilmore, and Ibrahim, 2012), biology (Lee, Lee, Lee, and Pawitan, 2011) and chemometrics (Varmuza and Filzmoser, 2016).

In practice, the most important information in $\mathbf{Z}$ can be extracted using *dimensionality reduction*. This process is indicated with the numbered arrow in Figure 5.1. $\mathbf{Z}$ is mapped to a lower-dimensional space by applying a dimensionality reduction technique to $\mathbf{Z}$, yielding a matrix $\mathbf{Y}$ ($n \times K$), called an *embedding* (shown in dark blue in Figure 5.1). The dimensions of this embedding characterize K orthogonal directions of variation in $\mathbf{Z}$.

Figure 5.2 depicts an example of an embedding for a matrix $\mathbf{Z}$ containing spider abundances for different sampling sites (data from Smeenk-Enserink and Van Der Aart (1974)). The horizontal and vertical axes represent two major directions of variation in the dataset (left to right and top to bottom in the plot). The meaning of these embedding dimensions is unknown. For example, we do not know why Site7 and Site26 are on opposite ends of the first dimension depicted in Figure 5.2. However, we assume that some of the variables in $\mathbf{X}$ may be predictive of this dimension. For instance, the first dimension may represent soil acidity, where sites on the left of the plot (e.g. Site7) have low soil acidity, whereas sites on the right (e.g. Site26) have high acidity.

We are interested in identifying the variables in $\mathbf{X}$ that best predict the different dimensions of $\mathbf{Y}$, making it possible to assign meaning to these dimensions (the second step depicted in Figure 5.1). In the ecological example from above, each dimension of $\mathbf{Y}$ would be associated with a subset of the environmental variables, making it possible to understand the different mechanisms affecting species abundance. In order to do this, we seek to build an accurate and interpretable linear model that predicts the dimensions of $\mathbf{Y}$ based on the predictor variables in $\mathbf{X}$. For ease of interpretation, we would like to identify the most important predictors, which is a *variable selection* problem. Moreover, we would like to associate a small set, or "cluster", of these selected variables with each dimension of the target data. This is essentially a *supervised variable clustering* problem, as the targets are used to guide the clustering of the variables. Identifying "predictive" variable clusters enhances the interpretability of the model by giving a tangible meaning to the dimensions of $\mathbf{Y}$.

Unlike the columns of $\mathbf{Z}$ and $\mathbf{X}$, the columns (i.e. the dimensions) of $\mathbf{Y}$ do not have a pre-defined meaning. Therefore, rotating $\mathbf{Y}$ does not result in a loss of

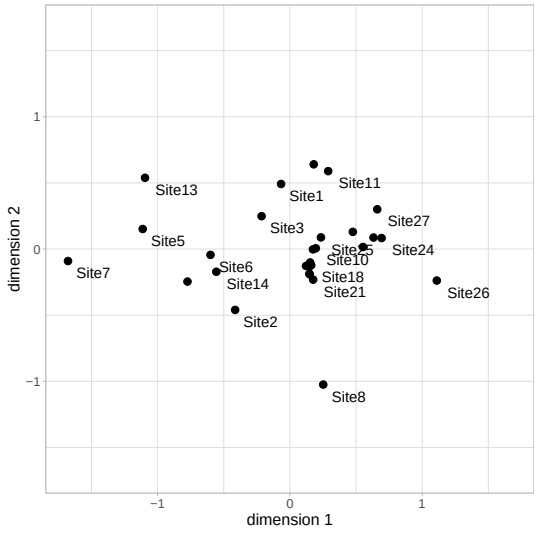


Figure 5.2: **Example of a two-dimensional embedding.** Multidimensional Scaling (MDS) embedding of a matrix $\mathbf{Z}$ of spider abundances at different sampling sites (data from Smeenk-Enserink and Van Der Aart (1974)). The observations (e.g. sites) from the embedding are plotted, where the horizontal and vertical axes correspond to embedding dimensions.

meaning with respect to the columns. In fact, rotating this matrix may better orient the patterns in the data with respect to $\mathbf{X}$, making it easier to estimate an accurate and interpretable (e.g. sparse) model. Orthogonal transformations, like rotations, are particularly appealing because they preserve all pairwise Euclidean distances between observations, thereby preserving all pertinent information from the original embedding.

The following sections describe how orthogonal transformations of $\mathbf{Y}$ impact the model error and interpretability of several regression methods. For the remainder of the chapter, we focus our analysis on the matrices $\mathbf{X}$ (observed) and $\mathbf{Y}$ (unobserved, generated based on $\mathbf{Z}$). The matrix $\mathbf{Y}$ is assumed to be a fixed input.

Ordinary Least Squares Regression (OLS)

For Ordinary Least Squares Regression (OLS), applying an orthogonal transformation to $\mathbf{Y}$ does not change the accuracy of the model, but it can somewhat increase its interpretability. Consider the linear model

$$\mathbf{Y} = \mathbf{X}\mathbf{B}^* + \mathbf{E}^*, \tag{5.1}$$

where $\mathbf{B}^*$ $(p \times K)$ is a matrix of regression coefficients and $\mathbf{E}^*$ $(n \times K)$ is an error

matrix. The model for a transformed matrix $\mathbf{Y}$ is

$$\mathbf{YR} = \mathbf{XB} + \mathbf{E}, \tag{5.2}$$

where $\mathbf{R}$ $(K \times K)$ is an orthogonal transformation matrix, $\mathbf{B}$ $(p \times K)$ is a matrix of regression coefficients for transformed $\mathbf{Y}$ and $\mathbf{E}$ $(n \times K)$ is an error matrix.

Let $\hat{\mathbf{B}}^*$ be the OLS solution for $\mathbf{B}^*$ in Eq. (5.1). For fixed $\mathbf{R}$, the OLS solution for $\mathbf{B}$ in Eq. (5.2) is $\hat{\mathbf{B}} = \hat{\mathbf{B}}^*\mathbf{R}$. The total squared model error for these two OLS solutions is the same for any orthogonal matrix $\mathbf{R}$. Indeed, using the fact that $\mathbf{RR}^\top = \mathbf{I}_K$ and that the trace is invariant under cyclic permutation, we can show that

$$\begin{aligned}
||\mathbf{YR} - \mathbf{X}\hat{\mathbf{B}}||_F^2 &= ||\mathbf{YR} - \mathbf{X}\hat{\mathbf{B}}^*\mathbf{R}||_F^2 \\
&= ||(\mathbf{Y} - \mathbf{X}\hat{\mathbf{B}}^*)\mathbf{R}||_F^2 \\
&= ||\hat{\mathbf{E}}^*\mathbf{R}||_F^2 \\
&= \mathrm{Tr}(\mathbf{R}^\top \hat{\mathbf{E}}^{*\top}\hat{\mathbf{E}}^*\mathbf{R}) \\
&= \mathrm{Tr}(\mathbf{RR}^\top \hat{\mathbf{E}}^{*\top}\hat{\mathbf{E}}^*) \qquad \textit{cyclic permutation} \\
&= \mathrm{Tr}(\hat{\mathbf{E}}^{*\top}\hat{\mathbf{E}}^*) \qquad\qquad\qquad \mathbf{RR}^\top = \mathbf{I}_K \\
&= ||\hat{\mathbf{E}}^*||_F^2 \\
&= ||\mathbf{Y} - \mathbf{X}\hat{\mathbf{B}}^*||_F^2.
\end{aligned} \tag{5.3}$$

However, in terms of model interpretability, transforming $\mathbf{Y}$ may somewhat increase the model sparsity. Consider the example in Figure 5.3(a). A 2D embedding was rotated with different angles θ and OLS coefficients were estimated for each angle. The plot in Figure 5.3(a) shows the coefficient values $\hat{b}_{jk}$ for the different variables j (shown using different colors and line types) for rotation angles $\theta \in [0, 360]$. For certain angles, some of the coefficients $\hat{b}_{jk}$ are equal to 0. This is not the case for the original embedding ($\theta = 0$), where all coefficients are non-zero.

Lasso Regression

Lasso regression may be preferred to OLS as it tends to simultaneously sets many coefficients to zero, resulting in a regression model that is generally less complex. The effect of transformation on the sparsity of Lasso models is even more apparent than for OLS models, and transforming $\mathbf{Y}$ can actually decrease the overall model error. The Lasso solution for a multivariate, transformed $\mathbf{Y}$ is

Lasso Multiple Regression for Transformed Y

$$\arg\min_{\mathbf{B}} ||\mathbf{YR} - \mathbf{XB}||_F^2 + \lambda \sum_{k=1}^{K} ||\mathbf{b}_k||_1. \tag{5.4}$$

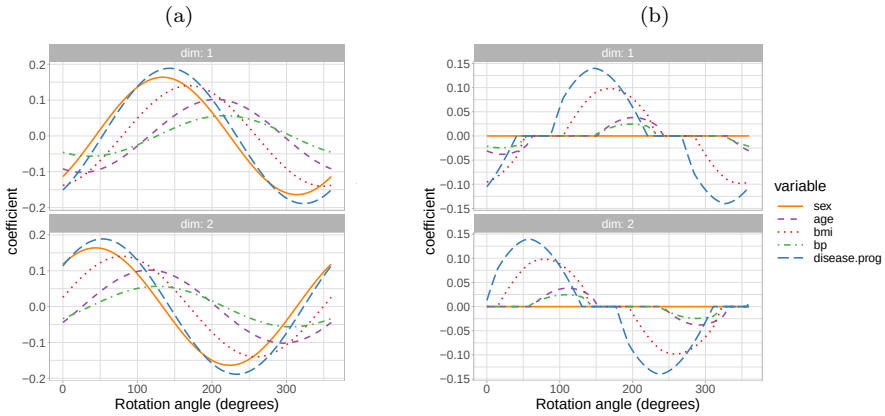


Figure 5.3: **Effect of rotation on OLS and Lasso regression**. Example for the dataset Diabetes from Efron, Hastie, Johnstone, and Tibshirani (2004). The blood serum variables in this dataset are used to generate a 2D embedding $\mathbf{Y}$, which is then rotated with different angles $\theta \in [0, 360]$ degrees. The remaining variables – age, sex, body mass index ("bmi"), blood pressure ("bp") and diabetes progression status ("disease.prog") – are used to predict the dimensions of the rotated embedding. Coefficient estimates for each variable and dimension of $\mathbf{YR}$, the rotated embedding, are shown for (a) OLS and (b) Lasso. The values at the origin represent the setting where the embedding is not rotated.

The ℓ_1 penalty in the second term induces a thresholding effect based on λ, setting all coefficients under this λ-dependent threshold to zero. The effect of different transformations on sparsity can be observed in Figure 5.3(b), where the top panel shows a plot of the number of total nonzero coefficients with respect to the rotation angle. For certain rotation angles, several variables have zero-valued coefficients.

The bottom panel of Figure 5.4 depicts a plot of the total squared prediction error with respect to the rotation angle, where the error fluctuates as the embedding is rotated. Unlike OLS, the total error of Lasso models depends on the choice of the orthogonal transformation of $\mathbf{Y}$. In addition, the rotation angles resulting in the most accurate model (highlighted in gray in the top panel) are not necessarily the ones that yields the sparsest model. Model sparsity is desirable because it generally makes the model more interpretable. Therefore, if Lasso is used to interpret an embedding $\mathbf{Y}$, it may be necessary to strike a balance between minimizing model error and maximizing sparsity. This balance can be achieved by optimizing the transformation applied to $\mathbf{Y}$. In the example in Figure 5.4, this corresponds to finding the rotation angle θ achieving a given level of error and sparsity.

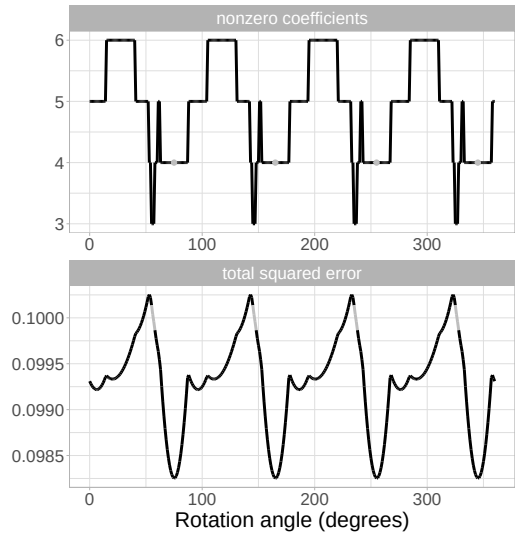


Figure 5.4: **Effect of rotation on Lasso model error and sparsity**. Example for the dataset Diabetes from Efron, Hastie, Johnstone, and Tibshirani (2004). The blood serum variables in this dataset are used to generate a 2D embedding $\mathbf{Y}$, which is then rotated with different angles $\theta \in [0, 360]$ degrees. The remaining variables are used to predict the dimensions of the rotated embedding using Lasso regression. The values at the origin represent the setting where the embedding is not rotated. Line segments highlighted in gray indicate θ values minimizing error (top plot) or maximizing sparsity (bottom plot).

5.1.2 Explaining MDS Embedding Dimensions

As explained in Section 5.1.1, the matrix $\mathbf{Y}$ is generated by performing dimensionality reduction (DR). DR is often used when the high-dimensionality of the original dataset makes it difficult to perform data exploration and/or makes data analysis victim to the curse of dimensionality (Bellman, 1961; Hastie, Tibshirani, and Friedman, 2009). However, some of the most effective DR techniques (i.e. Uniform Manifold Approximation and Projection (UMAP) (McInnes, Healy, and Melville, 2018), t-Distributed Stochastic Neighborhood Embedding (t-SNE) (Maaten and Hinton, 2008), Multidimensional Scaling (MDS) (Kruskal and Wish, 1978), etc.) are nonlinear, which makes the embeddings they generate difficult to understand. One solution to this problem is to use a set of additional variables (i.e. the matrix $\mathbf{X}$) to explain the dimensions of the low-dimensional embedding. These additional variables are also called *external variables* because they were not used to generate the embedding.

When linear models are used to explain the embedding dimensions, these dimen-

sions must be spatially meaningful. In other words, the proximity of observations in the low-dimensional space should reflect the similarity or dissimilarity of observations in the original high-dimensional space. This condition is not necessarily respected for neighborhood-preserving DR methods such as t-SNE or UMAP, which do not preserve small distances in the same way as large distances. These methods can thus generate embedding dimensions that are spatially misleading. On the other hand, methods that seek to preserve pairwise distances between observations, like Multidimensional Scaling (MDS) (Kruskal and Wish, 1978), are good candidates for this explanation approach.

MDS (Kruskal and Wish, 1978) is a very popular nonlinear dimensionality reduction (NLDR) method (Lee and Verleysen, 2007) in this category, especially in fields like psychology and ecology, and it is well-developed in the literature. Let $\mathbf{D}^{\text{HD}}$ $(n \times n)$ be a matrix quantifying the dissimilarity between pairs of observations in $\mathbf{Z}$. The Euclidean distance, for example, may be used to calculate these dissimilarities. Similarly, $\mathbf{D}^{\text{LD}}$ $(n \times n)$ is a matrix of pairwise dissimilarities for the low-dimensional embedding $\mathbf{Y}$. Given a number of dimensions K, the objective of MDS is to find the solution for $\mathbf{Y}$ $(n \times K)$ that minimizes a measure of reconstruction error between these two dissimilarity matrices. For "metric" MDS, one of these measures, Kruskal's stress, is defined as

MDS: Kruskal's Stress

$$\text{Stress} = \sqrt{\frac{\sum_{ii'} \left(d_{ii'}^{\text{HD}} - d_{ii'}^{\text{LD}}\right)^2}{\sum_{ii'} d_{ii'}^{\text{HD}^2}}}, \tag{5.5}$$

where $d_{ii'}^{\text{HD}}$ (resp. $d_{ii'}^{\text{LD}}$) is the dissimilarity between the i-th and i'-th observations in HD (resp. LD). For "nonmetric" MDS, a monotone function is used to transform $d_{ii'}^{\text{HD}}$. As a result, the rank of the HD dissimilarities is compared to the LD dissimilarities.

Because the mapping from high to low dimensions in nonlinear, the orthogonal embedding dimensions generated by MDS do not have an apparent meaning. Moreover, the stress function is invariant to orthogonal transformations of the embedding $\mathbf{Y}$. This means that the orientation of the generated embedding is entirely arbitrary. Therefore, when sparse, linear models are used to explain MDS dimensions, it is necessary to also optimize the orientation of the embedding (i.e. the transformation matrix $\mathbf{R}$, see Section 5.1.1).

5.1.3 Contributions

This chapter describes two methods that we have proposed for explaining MDS embedding dimensions using sparse, linear models. These methods search for an orthogonal transformation matrix $\mathbf{R}$ and sparse coefficient matrix $\mathbf{B}$ such that the

resulting model strikes a balance between sparsity and error. Neither methods require visualization of the transformed embedding $\mathbf{YR}$, as the meaning of its dimensions can be inferred from the coefficient matrix $\mathbf{B}$: the cluster of variables that explains a given dimension k is the set of variables with nonzero coefficients in $\mathbf{b}_k$. The variable clusters identified using our proposed methods tend to overlap very little in practice, lending a distinct meaning to each embedding dimension.

For the first proposed method, Best Interpretable Rotation (BIR), $\mathbf{R}$ is restricted to being a rotation matrix with a single parameter θ, the rotation angle. Therefore, BIR may only be used to explain two-dimensional (2D) embeddings. The second method, Best Interpretable Orthogonal Transformation (BIOT), allows $\mathbf{R}$ to be any orthogonal matrix, and it can be applied to embeddings of any dimension. Thanks to this, embeddings that have a lower reconstruction loss, and are thus more faithful to the original high-dimensional data, can be studied. Moreover, the objective function for BIOT can be optimized without performing an exhaustive exploration of all possible rotation angles, which is an improvement with respect to BIR.

5.1.4 Chapter Outline

BIR and BIOT are presented in Section 5.2 and related works are summarized in Section 5.3. Numerical experiments and a case study are described in Sections 5.4 and 5.5, respectively, and the conclusion and suggestions for future work can be found in Section 5.6.

5.2 Proposed Methods

This section describes two methods that we propose for solving the problem described in Section 5.1.1. The first method, Best Interpretable Rotation (BIR), is built for case where the embedding $\mathbf{Y}$ is two-dimensional ($K = 2$), whereas the second method, Best Interpretable Orthogonal Transformation (BIOT), is an improvement of BIR that can be used for embeddings of any number of dimensions.

5.2.1 Best Interpretable Rotation (BIR)

Best interpretable rotation (BIR) finds rotations of 2D MDS embeddings that can be explained by sparse multiple regression models (Bibal, Marion, and Frénay, 2018; Marion, Bibal, and Frénay, 2019). As seen in Section 5.1.1, the model error and number of non-zero coefficients of Lasso multiple regression models depend on the rotation of the target matrix $\mathbf{Y}$. Let the 2D rotation matrix $\mathbf{R}^\theta$ be defined as

$$\mathbf{R}^{\theta} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}. \tag{5.6}$$

In order to find a rotation balancing model error with interpretability, we propose finding the best rotation angle $\hat{\theta}$ as follows:

Best Interpretable Rotation (BIR)

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{2n} ||\mathbf{Y}\mathbf{R}^{\theta} - \mathbf{X}\mathbf{B}^{\theta}||_F^2 + \lambda \sum_{k=1}^{K} ||\mathbf{b}_k^{\theta}||_0, \tag{5.7}$$

where $K = 2$ and $\mathbf{b}_k^{\theta}$ is the Lasso solution explaining the k^{th} dimension of $\mathbf{Y}$ rotated by $\mathbf{R}^{\theta}$. While the matrix of coefficients $\mathbf{B}^{\theta}$ is the solution to a regression problem with an ℓ_1-norm penalty, BIR's objective function is minimized with respect to a scalar θ, making it feasible to impose an ℓ_0-norm penalty.

Once the optimal rotation angle $\hat{\theta}$ is selected, Lasso regression is applied to the rotated target matrix $\mathbf{Y}\mathbf{R}^{\hat{\theta}}$, yielding a matrix of estimated coefficients $\hat{\mathbf{B}}$. Note that the optimization of $\mathbf{R}^{\theta}$ using BIR involves the ℓ_0 norm (see Equation (5.7)), while the Lasso used to estimate $\mathbf{B}$ involves the ℓ_1 norm (see Eq. (5.4) in Section 5.1.1).

Using BIR to find a rotation angle θ results in better subsequent models than other potential competitors from the literature (Marion, Bibal, and Frénay, 2019), but BIR suffers from two important issues. First, θ is optimized by performing an exhaustive search. In practice, an optimization method for non-convex objective functions is used, such as simulated annealing. The solution for this kind of optimization depends on how long users accept to wait for a solution, as a time-stopping threshold is provided as input. Second, BIR can only find a rotation matrix for 2D MDS embeddings.

5.2.2 Best Interpretable Orthogonal Transformation (BIOT)

A second method we propose, Best Interpretable Orthogonal Transformation (BIOT) (Bibal, Marion, von Sachs, and Frénay, 2021), addresses the potential shortcomings of BIR described above. The overall objective of Best Interpretable Orthogonal Transformation (BIOT) is to explain the dimensions of an embedding $\mathbf{Y}$ $(n \times K)$ using a matrix of external variables $\mathbf{X}$ $(n \times p)$. BIOT does this by finding an orthogonal $K \times K$ matrix $\mathbf{R}$ such that the transformed embedding can be explained by a sparse coefficient matrix $\mathbf{B}$ $(p \times K)$. Given a hyperparameter $\lambda > 0$, the optimization problem for BIOT is

Best Interpretable Orthogonal Transformation (BIOT)

$$\begin{aligned} \arg\min_{\mathbf{R},\mathbf{B}} \frac{1}{2n}||\mathbf{YR}-\mathbf{XB}||_F^2+\lambda\sum_{k=1}^{K}||\mathbf{b}_k||_1 \\ \text{s.t. } \mathbf{R} \text{ is an orthogonal matrix,} \\ \text{i.e. } \mathbf{RR}^\top=\mathbf{R}^\top\mathbf{R}=\mathbf{I}_K. \end{aligned} \tag{5.8}$$

The orthogonality constraint for $\mathbf{R}$ ensures that the transformed embedding $\mathbf{YR}$ retains all meaningful structure from the original embedding: pairwise Euclidean distances and the dimensionality of the embedding are preserved. The Lasso penalty on the columns of $\mathbf{B}$ (i.e. $\sum_{k=1}^{K}||\mathbf{b}_k||_1$) encourages the selection of fewer variables per embedding dimension. As a result, the transformed dimensions can be explained by potentially distinct clusters of variables. The best $\mathbf{R}$ for this problem is the orthogonal transformation that results in the model with the best balance between model error and sparsity, as controlled by the hyperparameter λ.

Optimizing B for Fixed R

Given a fixed embedding orientation $\mathbf{R}$, the optimization of the coefficients $\mathbf{B}$ is a Lasso problem. For a particular embedding dimension k, the optimal coefficient vector is

$$\arg\min_{\mathbf{b}_k} \frac{1}{2n}||\mathbf{Yr}_k-\mathbf{Xb}_k||_2^2+\lambda||\mathbf{b}_k||_1. \tag{5.9}$$

Following cyclic coordinate descent optimization (Hastie, Tibshirani, and Wainwright, 2015), all values of $\mathbf{b}_k$ are fixed, except a certain value b_{jk}, at each iteration. The problem to solve can therefore be rewritten as

$$\arg\min_{b_{jk}} \frac{1}{2n}||\mathbf{e}_{-jk}-\mathbf{x}_j b_{jk}||_2^2+\lambda||\mathbf{b}_{-jk}||_1+\lambda|b_{jk}|, \tag{5.10}$$

where $\mathbf{e}_{-jk}=\mathbf{Yr}_k-\mathbf{X}_{-j}\mathbf{b}_{-jk}$, $\mathbf{X}_{-j}$ is $\mathbf{X}$ without its j-th column $\mathbf{x}_j$ and $\mathbf{b}_{-jk}$ is the coefficient vector $\mathbf{b}_k$ without its j-th value b_{jk}. The optimal value $\hat{b}_{jk}$ can be calculated using soft thresholding (Hastie, Tibshirani, and Wainwright, 2015):

$$\hat{b}_{jk}=\frac{\text{sign}(\mathbf{x}_j^\top\mathbf{e}_{-jk})(|\mathbf{x}_j^\top\mathbf{e}_{-jk}|-n\lambda)_+}{\mathbf{x}_j^\top\mathbf{x}_j}. \tag{5.11}$$

Optimizing R for Fixed B

When $\mathbf{B}$ is found, the next step is to adjust the orientation of the embedding. Since $\mathbf{RR}^\top=\mathbf{I}_K$, for fixed $\mathbf{B}$, Eq. (5.8) can be rewritten as

$$\begin{aligned} &\arg\min_{\mathbf{R}} \frac{1}{2n}||\mathbf{Y}-\mathbf{X}\mathbf{B}\mathbf{R}^\top||_F^2+\lambda\sum_{k=1}^{K}||\mathbf{b}_k||_1 \\ &\text{s.t. } \mathbf{R} \text{ is an orthogonal matrix,} \end{aligned} \tag{5.12}$$

because

$$\begin{aligned} &||\mathbf{Y}\mathbf{R}-\mathbf{X}\mathbf{B}||_F^2 \\ &= \mathrm{Tr}((\mathbf{Y}\mathbf{R}-\mathbf{X}\mathbf{B})^\top(\mathbf{Y}\mathbf{R}-\mathbf{X}\mathbf{B})) \\ &= \mathrm{Tr}(\mathbf{R}^\top\mathbf{Y}^\top\mathbf{Y}\mathbf{R}-\mathbf{R}^\top\mathbf{Y}^\top\mathbf{X}\mathbf{B}-\mathbf{B}^\top\mathbf{X}^\top\mathbf{Y}\mathbf{R}+\mathbf{B}^\top\mathbf{X}^\top\mathbf{X}\mathbf{B}) \\ &= \mathrm{Tr}(\mathbf{Y}^\top\mathbf{Y}-\mathbf{Y}^\top\mathbf{X}\mathbf{B}\mathbf{R}^\top-\mathbf{R}\mathbf{B}^\top\mathbf{X}^\top\mathbf{Y}+\mathbf{R}\mathbf{B}^\top\mathbf{X}^\top\mathbf{X}\mathbf{B}\mathbf{R}^\top) \\ &= \mathrm{Tr}((\mathbf{Y}-\mathbf{X}\mathbf{B}\mathbf{R}^\top)^\top(\mathbf{Y}-\mathbf{X}\mathbf{B}\mathbf{R}^\top)) \\ &= ||\mathbf{Y}-\mathbf{X}\mathbf{B}\mathbf{R}^\top||_F^2, \end{aligned} \tag{5.13}$$

thanks to the cyclic property of the trace and the fact that $\mathbf{R}$ is an orthogonal matrix.

Finding the optimal matrix $\mathbf{R}$ is an orthogonal Procrustes problem (Gower and Dijksterhuis, 2004). Indeed, for fixed $\mathbf{B}$, Eq. (5.12) can be rewritten as

$$\arg\min_{\mathbf{\Psi}}||\mathbf{F}-\mathbf{G}\mathbf{\Psi}||_F^2 \quad \text{s.t.} \quad \mathbf{\Psi}\mathbf{\Psi}^\top=\mathbf{\Psi}^\top\mathbf{\Psi}=\mathbf{I}_K, \tag{5.14}$$

where $\mathbf{F}=\mathbf{Y}/\sqrt{2n}$, $\mathbf{G}=\mathbf{X}\mathbf{B}/\sqrt{2n}$ and $\mathbf{\Psi}=\mathbf{R}^\top$. The matrix $\mathbf{\Psi}$ that minimizes Eq. (5.14) can then be found by decomposing the matrix $\mathbf{L}=\mathbf{G}^\top\mathbf{F}=\frac{1}{2n}(\mathbf{X}\mathbf{B})^\top\mathbf{Y}$ using SVD, such that $\mathbf{L}=\boldsymbol{\mathcal{U}}\boldsymbol{\mathcal{D}}\boldsymbol{\mathcal{V}}^\top$, where $\boldsymbol{\mathcal{U}}$ and $\boldsymbol{\mathcal{V}}$ contain the left- and right-singular vectors of $\mathbf{L}$ and the solution $\hat{\mathbf{\Psi}}=\boldsymbol{\mathcal{U}}\boldsymbol{\mathcal{V}}^\top$ (Golub and Van Loan, 2013). The transformation matrix $\mathbf{R}$ optimizing Eq. (5.12) is thus $\hat{\mathbf{R}}=\hat{\mathbf{\Psi}}^\top=\boldsymbol{\mathcal{V}}\boldsymbol{\mathcal{U}}^\top$.

Optimization Algorithm

Algorithm 3, inspired by Chen and Huang (2012), presents BIOT. It is composed of two repeated steps: 1) optimizing $\mathbf{B}$ given an embedding transformation $\mathbf{R}$ and 2) optimizing $\mathbf{R}$ given regression coefficients $\mathbf{B}$. These steps are repeated until the change of $\mathbf{B}$ from one iteration to another is lower than a predefined threshold.

Algorithm 3 is an Alternating Optimization (AO) algorithm because it alternates between optimizing $\mathbf{B}$ for fixed $\mathbf{R}$ and $\mathbf{R}$ for fixed $\mathbf{B}$. It is difficult to prove that this type of algorithm converges to a global optimum. However, some works have shown that AO algorithms can at least converge to a second-order stationary point under fairly mild assumptions (see e.g. Li, Zhu, and Tang (2019)).

Data: MDS embedding $\mathbf{Y}$ and variable matrix $\mathbf{X}$
Result: Explanation of $\mathbf{Y}$ with sparse coefficients $\mathbf{B}$
$\mathbf{R} = \mathbf{I}_K$;
$\mathbf{B}$ is obtained by solving Eq. (5.9) for each dimension k of $\mathbf{YR}$;
while B changes do
    `// Optimizing R`
    $\mathbf{R}$ is obtained by solving Eq. (5.12);
    `// Optimizing B`
    for each dimension k **of YR do**
        $\mathbf{b}_k$ is obtained by solving Eq. (5.9);
    end
end
return B and R

Algorithm 3: BIOT algorithm, inspired by Chen and Huang (2012).

5.2.3 Selecting the Hyperparameters λ and K

BIR requires the selection of the hyperparameter λ, which represents the relative importance of model sparsity with respect to error. BIOT also requires the selection of λ, as well as an additional hyperparameter K, the number of embedding dimensions to analyze. The first hyperparameter, λ, is common to all Lasso problems and can be set according to the same strategies. For instance, the λ leading to the smallest validation mean squared error (MSE) can be considered. Alternatively, the "one-standard error" rule (Hastie, Tibshirani, and Friedman, 2009) may be used, whereby the largest λ within one-standard deviation of the minimum validation MSE is chosen. This corresponds to a sparser model than for the minimum validation MSE model with a comparable level of error.

While sparsity helps avoid issues like overfitting, it is mainly used, in this work, as a means to obtain interpretable regression models (i.e. models with a reasonable number of non-zero coefficients). Therefore, while the above heuristics can be used to select λ, the final choice remains with the user. In practical settings, it may be interesting to increase the sparsity of regression models, and thus their interpretability, even at the cost of increasing their validation MSE. For the evaluation of BIR and BIOT in this chapter (Section 5.4), however, one of the methods presented in the previous paragraph is used to maintain objectivity.

The number K of embedding dimensions is more similar to the hyperparameters used in unsupervised learning. In classical clustering problems, different numbers of clusters are tested and analyzed, given the knowledge of experts, to see which choice makes the most sense. Similarly, for BIOT, different numbers of dimensions can be tested to observe how the analysis changes. Oftentimes, increasing K results in explanations with more and more nuance, as seen in the example provided in

Section 5.5.

In addition to increasing the granularity of the explanation, increasing the number of dimensions reduces the information loss in the embedding, making it more faithful to the original dataset. It also makes the explanation of each individual dimension easier, as less information must be explained by the external variables. However, these advantages come at the cost of increasing the cognitive load for the user: understanding 10 dimensions simultaneously may be difficult, even if each dimension is explained by only two or three variables. Therefore, some balance must be found between cognitive ease and the level of nuance and faithfulness. Gradually increasing the number of dimensions provides a practical means of evaluating when the number of dimensions K becomes too high for cognitive processing.

5.2.4 Summary

BIR and BIOT were developed in order to address the research problem described in Section 5.1.1, in particular the second step of the data analysis pipeline depicted in Figure 5.1. Given two observed matrices $\mathbf{Z}$ $(n \times q)$ and $\mathbf{X}$ $(n \times p)$, the entire pipeline is as follows:

1. An NLDR technique, such as Multidimensional Scaling (MDS), is applied to $\mathbf{Z}$, independent of $\mathbf{X}$, to generate the embedding $\mathbf{Y}$ $(n \times K)$.

2. BIR or BIOT is applied to the matrices $\mathbf{Y}$ and $\mathbf{X}$:

   (a) An orthogonal transformation of $\mathbf{Y}$ is found such that the transformed dimensions can be accurately explained as a function $\mathbf{X}$ using a sparse regression model.

   (b) The clusters of variables that are selected for a given transformed dimension assign meaning to that dimension: they explain the similarity/dissimilarity between pairs of observations along that dimension.

5.3 Related Work

The MDS objective function preserves Euclidean distances between all pairs of points, which makes it invariant to orthogonal transformations. As such, any orthogonal transformation of a given MDS embedding is an equally valid solution to the MDS problem. This section presents several approaches from the statistics and machine learning literature for orthogonally transforming a data *view* – in this case, an embedding generated by MDS. The methods presented here also estimate a linear model relating the variables in a second data view to the orthogonally transformed view.

5.3.1 Eigenvector Partial Least Squares Regression

Eigenvector Partial Least Squares Regression (PLS-R) involves two steps. In the first step, Eigenvector PLS (Varmuza and Filzmoser, 2016), also known as Bookstein PLS (Abdi, 2010), is applied to the two data views $\mathbf{X}$ and $\mathbf{Y}$. The goal of Eigenvector PLS is to find orthogonal transformation matrices $\mathbf{V}$ and $\mathbf{R}$ such that the covariance between $\mathbf{U} = \mathbf{XV}$ and $\mathbf{T} = \mathbf{YR}$ is maximal. Both $\mathbf{U}$ and $\mathbf{T}$ are of dimension $n \times r$, where $r = \min(K, p)$, p is the number of columns in $\mathbf{X}$ and K is the number of columns in $\mathbf{Y}$.

In the second step of Eigenvector PLS-R, the matrix $\mathbf{T} = \mathbf{YR}$ is regressed onto $\mathbf{U} = \mathbf{XV}$ using ordinary least squares (OLS). The model is defined as

$$\mathbf{T} = \mathbf{UW} + \mathbf{E}, \tag{5.15}$$

where $\mathbf{E}$ is an error term and $\mathbf{W}$ is a matrix of regression coefficients, whose solution is

$$\hat{\mathbf{W}} = (\mathbf{U}^\top \mathbf{U})^{-1} \mathbf{U}^\top \mathbf{T}. \tag{5.16}$$

The orthogonally transformed view $\mathbf{YR}$ can thus be interpreted as a linear combination of the variables in $\mathbf{X}$:

$$\mathbf{YR} = \mathbf{UW} + \mathbf{E} = \mathbf{XB} + \mathbf{E}, \tag{5.17}$$

where $\mathbf{B} = \mathbf{VW}$ is a matrix of regression coefficients describing the linear relationship between each variable in $\mathbf{X}$ and each dimension of $\mathbf{YR}$.

One limitation of Eigenvector PLS-R is that the coefficients linking the two matrices are not sparse. As a result, the model that explains the dimensions of $\mathbf{YR}$ can be difficult to interpret.

5.3.2 Sparse Reduced Rank Regression (SRRR)

For MDS embeddings with two or more dimensions K, sparse reduced rank regression (SRRR) (Chen and Huang, 2012) could potentially be used to regress transformed embedding dimensions on external variables. SRRR was originally introduced as a method for predicting an un-transformed target matrix using a coefficient matrix $\mathbf{C} = \mathbf{BR}^\top$ of fixed rank r. Unlike Eigenvector PLS-R, Sparse Reduced Rank Regression (SRRR) introduces a constraint to encourage $\mathbf{B}$ to be sparse. The original problem presented in Chen and Huang (2012) is to find $\mathbf{R}$ $(K \times r)$ and $\mathbf{B}$ $(p \times r)$ by solving

Sparse Reduced Rank Regression (SRRR)

$$\begin{aligned} \arg\min_{\mathbf{R},\mathbf{B}} \frac{1}{2n}||\mathbf{Y}-\mathbf{X}\mathbf{B}\mathbf{R}^\top||_F^2+\lambda\sum_{j=1}^{p}||\mathbf{b}_j||_2 \\ \text{s.t. } \mathbf{R}^\top\mathbf{R}=\mathbf{I}_r \text{ and } \text{rank}(\mathbf{B}\mathbf{R}^\top)=r, \end{aligned} \tag{5.18}$$

where $\mathbf{b}_j$ is the j^{th} row of $\mathbf{B}$, $\lambda > 0$ and $r \in \{1, ..., \min(p, K)\}$. The second term in Eq. (5.18) is a Group-Lasso penalty that forces the elements of $\mathbf{b}_j$ to be either all zero or non-zero (Yuan and Lin, 2006). As λ increases, more rows of $\mathbf{B}$ are set to zero, meaning that fewer variables are used to explain the target matrix.

The objective function in Eq. (5.18) can be reformulated to show that the matrix $\mathbf{B}$ in SRRR contains the regression coefficients for predicting a transformed target matrix $\mathbf{YR}$. Indeed, thanks to the rotational invariance of the Frobenius norm, the first term in Eq. (5.18) can be rewritten as follows (see Eq. (5.13):

$$\frac{1}{2n}||\mathbf{Y}-\mathbf{X}\mathbf{B}\mathbf{R}^\top||_F^2 = \frac{1}{2n}||\mathbf{Y}\mathbf{R}-\mathbf{X}\mathbf{B}||_F^2. \tag{5.19}$$

For the current application, the meaningful structure from the original embedding $\mathbf{Y}$ must be preserved. Therefore, SRRR is only applicable when its hyperparameter r (the rank of $\mathbf{B}\mathbf{R}^\top$ and the number of columns in $\mathbf{R}$) is fixed to $r = K$, the number of embedding dimensions. The setting $r = K$ is the only one that ensures that $\mathbf{R}$ is an orthogonal matrix and that the transformed embedding $\mathbf{YR}$ retains the same number of dimensions as the original embedding $\mathbf{Y}$.

Despite its potential relevance for the problem at hand, the sparsity constraints in SRRR are less well adapted than the constraints in BIR and BIOT. Indeed, for SRRR, the same cluster of variables would be selected for each transformed embedding dimension, making it difficult to attribute a distinct meaning to each dimension. In contrast, BIR and BIOT make it possible to select potentially distinct clusters of variables for each embedding dimension, providing greater model interpretability.

5.3.3 Other Methods

Other multi-view methods in the literature estimate a sparse multiple regression model, but they do not find an orthogonal transformation of the target matrix. For example, sparse multi-task regression methods (e.g. Multi-Task Lasso (Obozinski, Taskar, and Jordan, 2006), Adaptive Multi-Task Lasso (Lee, Zhu, and Xing, 2010), Robust Variable Selection (Nie, Huang, Cai, and Ding, 2010) and Joint Rank and Row Selection (Bunea, She, Wegkamp, et al., 2012)) find a sparse coefficient matrix but do not transform the target matrix in any way. Sparse latent variable

approaches such as Sparse Canonical Correlation Analysis (SCCA) (Witten, Tibshirani, and Hastie, 2009; Wilms and Croux, 2015; Suo, Minden, Nelson, Tibshirani, and Saunders, 2017; Mai and Zhang, 2019) and Sparse Partial Least Squares Regression (SPLS-R) (Chun and Keleş, 2010) find a transformation of the target matrix, but it is not orthogonal.

5.4 Experiments

This section compares BIR and BIOT with competitors from the literature for MDS embeddings of two or more dimensions.

5.4.1 Evaluation Datasets

Three real-world datasets are drawn from the field of ecology: the Doubs river fish communities dataset (Doubs) (Verneaux, 1973), the Oribatid mites dataset (Mite) (Borcard, Legendre, and Drapeau, 1992; Borcard and Legendre, 1994) and the hunting spider dataset (Spider) (Smeenk-Enserink and Van Der Aart, 1974). Each dataset is made up of two distinct sets of variables, or *views*. The first view, $\mathbf{Z}$, contains abundances of q different species (of fish, mites and spiders, respectively) measured at n different sampling sites. The second view, $\mathbf{X}$, contains p environmental variables measured at the n sites, such as Cartesian coordinates, water pH and altitude. For each dataset, ordinal MDS is applied to the first view ($\mathbf{Z}$) in order to produce several embeddings with a number of dimensions K ranging from 2 to $\min(q, p) - 1$.

The fourth dataset used in our evaluation comes from an experiment in psychology about stereotypes (Stereotypes) (Koch, Imhoff, Dotsch, Unkelbach, and Alves, 2016). In this dataset, the first view, $\mathbf{Z}$ ($n \times n$) contains similarity scores for all possible pairs of n social groups (e.g. students, the homeless and athletes). The second view, $\mathbf{X}$, contains variables that encode stereotypes about these social groups (e.g. degree of smartness, trustworthiness and sincerity). The first view $\mathbf{Z}$ is used to generate several MDS embeddings, as for the other datasets.

For all datasets, the second view $\mathbf{X}$ (normalized) is used to explain the mean-centered embeddings produced by the MDS of view $\mathbf{Z}$. Table 5.1 summarizes the characteristics of the datasets used in the experiments.

5.4.2 Experimental Protocol

Four methods are compared in this study: BIOT, BIR (for 2D embeddings only), SRRR and Eigenvector PLS with Lasso regression (ePLS+Lasso). For ePLS+Lasso, we add a Lasso step to ordinary Eigenvector PLS in order to benchmark BIR and BIOT and to make the results comparable. Eigenvector PLS is used to estimate a

dataset	observations	variables		
		Z	**X**	**total**
Doubs	30	27	13	40
Mite	70	35	16	51
Spider	28	12	15	27
Stereotypes	80	80	31	111

Table 5.1: **Characteristics of the evaluation datasets.**

transformation matrix $\mathbf{R}$, then Lasso regression is performed based on the transformed embedding. A range of 20 values for λ ($[0.0001, 3.5]/\sqrt{p}$ in logarithmic scale) was chosen such that each method produces solutions ranging from entirely sparse to entirely non-sparse. For SRRR, the rank r of the matrix $\mathbf{BR}^\top$ is fixed to the number of embedding dimensions, as explained in Section 5.3.

For each method, embedding and value of λ, 10-fold cross-validation is performed to evaluate the average validation mean squared error (MSE), where

$$\text{MSE} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{n_{\text{fold}}}\sum_{i=1}^{n_{\text{fold}}}(\mathbf{y}_i^\top\hat{\mathbf{r}}_k - \mathbf{x}_i^\top\hat{\mathbf{b}}_k)^2, \tag{5.20}$$

with K being the total number of embedding dimensions, n_{fold} being the total number of observations for which a prediction is made, $\hat{\mathbf{r}}_k$ is column k from the estimated orthogonal matrix $\hat{\mathbf{R}}$ and $\hat{\mathbf{b}}_k$ is column k from the estimated coefficient matrix $\hat{\mathbf{B}}$. Note that the observations i in the formula were not used when estimating $\hat{\mathbf{R}}$ and $\hat{\mathbf{B}}$. The average validation MSEs for each method and embedding are plotted with respect to the average number of non-zero coefficients in $\mathbf{B}$ per dimension, where each point represents a value of λ. The minimum of each plotted curve is the minimum validation MSE.

In order to statistically analyze the results obtained for a given dataset and number of dimensions K, the performance of the methods is compared using nested k-fold cross-validation. Each dataset is first split into 10 outer folds. For each iteration ℓ of an outer loop, the observations in folds $1, ..., \ell-1, \ell+1, ..., 10$ are split into 10 inner folds. The observations in each fold are characterized by both the (external) variables and the target variables, i.e. the MDS coordinates to predict. k-fold cross-validation is performed using the inner folds in order to calculate the average validation MSE for each method and λ. For each method, λ is chosen as the value with the smallest average validation MSE, then the matrices $\hat{\mathbf{R}}$ and $\hat{\mathbf{B}}$ are estimated based on all observations in the inner folds and used to predict the observations in outer fold ℓ. The average test MSE is calculated as the average out-of-sample prediction error (MSE) in the outer loop.

5.4.3 Results and Discussion

In this experiment, BIR, BIOT and the competing methods are applied to embeddings of different numbers of dimensions. Fig. 5.5 shows plots of the average validation MSE with respect to the average number of non-zero coefficients per dimension for 2D, 4D and 6D embeddings (left, middle and right columns respectively). For a given method (represented by a unique color and line type), the plotted line interpolates between the coordinates for each tested λ value, and the minimum validation MSE is highlighted with a colored symbol. For 2D embeddings (Figs. 5.5(a), 5.5(d), 5.5(g) and 5.5(j)), BIOT finds solutions that are generally as sparse or sparser than those of the other methods, for a similar MSE. Indeed, if a horizontal line is drawn in the graphs (representing a fixed MSE value), BIOT is almost always to the left of the other curves.

Similar trends can be observed for embeddings of more than two dimensions, as shown in the second and third columns of Fig. 5.5. Note that BIR is not present in these plots, as it can only be applied to 2D embeddings. Interestingly, the difference between the curves is accentuated as the number of embedding dimensions increases. This can be observed as a shifting pattern in the 4D and 6D embeddings of Stereotypes in Fig. 5.5(k) and Fig. 5.5(l), compared to a similar but less clear pattern in the 2D embedding of Stereotypes in Fig. 5.5(j). This observation is important, as BIOT is designed for use on higher-dimensional embeddings, where reconstruction error (like the stress) is lower.

The comparison of all methods applied to all embeddings is shown in Table 5.2. The last embeddings of Stereotypes ($K > 13$) are omitted, as they have stress levels equivalent to the stress for $K = 13$. The results for each method are shown as a pair of values: average number of non-zero coefficients per dimension and average test MSE. In order to report an error value that was not used for selecting λ, nested 10-fold cross-validation is used (see Section 5.4.2 for more details). On each line, results with the highest sparsity (resp. lowest MSE) are highlighted in bold (resp. italics), as well as all other results that are not significantly different according to a pairwise Wilcoxon signed-rank test ($\alpha = 0.05$). Any results not shown in bold or italics are significantly worse than the best results across the different folds.

As seen in Table 5.2, the best MSE is generally not significantly different for all methods, but the average number of non-zero coefficients often is. Most of the time, BIOT provides solutions with a lower number of variables per dimension, while having a test MSE similar to its competitors. The average number of variables used to explain a dimension generally decreases as the number of dimensions K increases. This can be explained by the fact that each new embedding dimension adds less information than previous ones (the stress decreases less). Therefore, fewer and fewer variables are needed to explain each additional dimension.

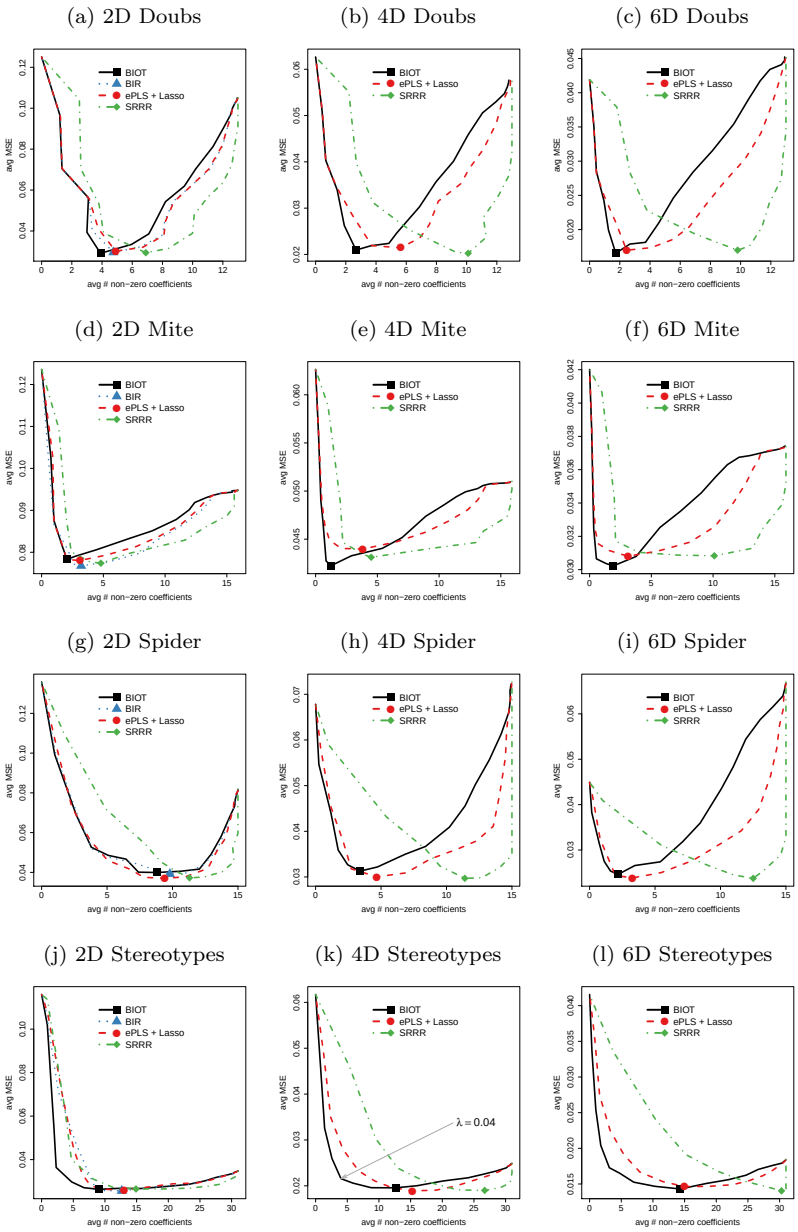


Figure 5.5: **Performance of methods for several λ values**. The average validation MSE is plotted against the average number of non-zero coefficients per dimension. The minimum validation MSE is highlighted for each method with a colored symbol.

	K	stress	BIR	BIOT	ePLS	SRRR
Do	2	0.070	**6.7**, *0.096*	**6.3**, *0.092*	**7.0**, *0.094*	7.0, *0.098*
	3	0.038		**4.8**, *0.060*	**5.4**, *0.061*	5.4, *0.063*
	4	0.026		**4.8**, *0.046*	**5.3**, 0.056	5.3, *0.049*
	5	0.018		**3.9**, *0.041*	**3.4**, *0.047*	3.4, *0.040*
	6	0.013		**2.9**, *0.037*	**2.9**, *0.036*	2.9, *0.038*
	7	0.012		**1.8**, *0.034*	1.8, *0.032*	1.8, *0.037*
	8	0.008		**1.5**, *0.029*	1.5, *0.030*	1.5, *0.033*
	9	0.006		**1.3**, *0.026*	1.3, *0.028*	1.3, *0.031*
	10	0.005		**1.0**, *0.022*	1.0, 0.026	1.0, 0.030
	11	0.004		**1.1**, *0.023*	1.1, *0.024*	1.1, 0.028
	12	0.003		**1.0**, *0.022*	1.0, *0.022*	1.0, 0.026
Mi	2	0.144	1.0, *0.156*	**3.0**, *0.161*	**3.4**, *0.160*	3.4, *0.159*
	3	0.112		**1.7**, *0.104*	1.7, *0.107*	1.7, *0.105*
	4	0.091		**1.2**, *0.084*	1.2, 0.090	1.2, *0.088*
	5	0.077		**1.2**, *0.072*	1.2, 0.075	1.2, *0.073*
	6	0.065		**1.8**, *0.061*	1.8, *0.063*	1.8, 0.063
	7	0.057		**1.5**, *0.053*	1.5, *0.054*	1.5, 0.055
	8	0.049		**1.5**, *0.048*	1.5, *0.048*	1.5, *0.049*
	9	0.044		**1.2**, *0.043*	1.2, *0.044*	1.2, *0.044*
	10	0.040		**1.3**, *0.040*	1.3, *0.040*	1.3, *0.039*
	11	0.036		**1.0**, *0.036*	1.0, *0.037*	1.0, *0.036*
	12	0.032		**1.1**, *0.033*	1.1, *0.034*	1.1, *0.033*
	13	0.029		**0.9**, *0.031*	0.9, *0.031*	0.9, *0.031*
Sp	2	0.089	0.9, *0.085*	**8.5**, *0.081*	8.5, *0.077*	8.5, *0.075*
	3	0.055		**4.0**, *0.071*	4.0, *0.069*	4.0, *0.080*
	4	0.037		**4.3**, *0.065*	4.3, *0.062*	4.3, *0.063*
	5	0.025		**3.3**, *0.060*	**3.6**, *0.055*	3.6, *0.059*
	6	0.019		**2.4**, *0.053*	2.4, *0.048*	2.4, *0.048*
	7	0.016		**1.9**, *0.044*	1.9, *0.043*	1.9, *0.042*
	8	0.012		**1.6**, *0.039*	1.6, *0.039*	1.6, *0.040*
	9	0.007		**1.5**, *0.036*	1.5, *0.036*	1.5, *0.036*
	10	0.004		**1.4**, *0.033*	1.4, *0.033*	1.4, *0.033*
	11	0.001		**1.4**, *0.031*	1.4, *0.030*	1.4, *0.030*
St	2	0.291	**11.3**, *0.053*	**10.8**, 0.058	**10.4**, *0.054*	10.4, *0.060*
	3	0.207		**12.7**, 0.036	**11.7**, *0.033*	11.7, *0.033*
	4	0.169		**9.7**, *0.040*	9.7, *0.039*	9.7, *0.039*
	5	0.146		**10.2**, 0.035	10.2, *0.034*	10.2, *0.032*
	6	0.134		**9.7**, 0.030	9.7, *0.030*	9.7, *0.029*
	7	0.127		**6.5**, *0.028*	6.5, 0.028	6.5, *0.027*
	8	0.122		**7.2**, *0.025*	7.2, *0.025*	7.2, *0.025*
	9	0.120		**8.0**, *0.022*	8.0, *0.023*	8.0, *0.022*
	10	0.118		**7.0**, 0.022	7.0, 0.022	7.0, *0.021*
	11	0.116		**6.4**, *0.020*	6.4, *0.020*	6.4, *0.020*
	12	0.116		**6.5**, *0.018*	6.5, 0.019	6.5, *0.018*
	13	0.115		**5.0**, 0.018	5.0, 0.018	5.0, *0.017*

Table 5.2: **Results for four datasets: Doubs (Do), Mite (Mi), Spider (Sp) and Stereotypes (St)**. Each result is a pair (average number of non-zero coefficients per dimension, average test MSE) corresponding to the average across the 10 outer folds of the nested 10-fold cross-validation.

5.5 Case Study: Applying BIOT to Stereotypes

The Stereotypes dataset was collected in order to study how people (in the US) implicitly assign stereotypes to social groups. In a first experiment, participants gave scores for the similarity between social groups, such as celebrities, students and criminals (dataset $\mathbf{Z}$). In a second experiment, participants scored these social groups with respect to stereotypes, such as wealthy, altruistic and skillful (external variables $\mathbf{X}$). The goal was then to see how these stereotypes could explain perceived similarities between social groups.

Let us consider that a researcher has generated three embeddings $\mathbf{Y}$ of the Stereotypes dataset $\mathbf{Z}$ using MDS (embeddings with $K = 3$, $K = 4$ and $K = 5$ dimensions) and now would like to use Lasso regression models to explain the dimensions of these embeddings. In this scenario, the researcher initially assumes that transforming the embeddings is not necessary (i.e. $\mathbf{R} = \mathbf{I}_K$). For Lasso, it is common practice to choose a λ resulting in the sparsest, most interpretable model possible while maintaining a low MSE. In order to remain objective, λ is chosen as follows. For λ with the smallest average validation MSE, a 95% confidence interval is calculated. Then, the largest λ value with an average validation MSE within this confidence interval is selected. The regression coefficients for this approach are shown in Table 5.3. It can be seen that the number of external variables used to explain each of the embedding dimensions is large, and that the same external variable is sometimes used to explain several dimensions at once (e.g. for $K = 3$, wealthy is used to explain the first and the third dimensions with a large coefficient).

In order to generate results that are more interpretable, the researcher then applies BIOT (i.e. $\mathbf{R}$ is optimized). The same approach is used to select the value of the hyperparameter λ (the chosen value is highlighted in Fig. 5.5(k)) and regression coefficients are estimated for each embedding (see Table 5.4). In contrast to the initial approach (Lasso without embedding transformation), BIOT selects fewer external variables per dimension, and each dimension is explained by a distinct cluster of variables. For the first embedding ($K = 3$, first column of Table 5.4), BIOT explains the three dimensions with the stereotypes wealthy, traditional/conventional and not smart. For the 4D embedding (second column of Table 5.4), BIOT provides an explanation of the fourth dimension by roughly separating traditional and conventional into two dimensions. Finally, for the 5D embedding (third column of Table 5.4), BIOT explains the new fifth dimension as a political dimension through the conservative-liberal stereotype.

Analyzing the coefficients estimated by BIOT makes it possible to identify the meaning of the embedding dimensions without requiring visualization. However, it is also possible to visualize the results if desired. An example of a visualization of BIOT's explanations is provided for the embedding with three dimensions.

In Figure 5.6, the orthogonally transformed embedding is plotted for pairs of dimensions, where the points represent observations (i.e. social groups). To avoid misinterpretation, observations that are well represented in two dimensions, with respect to the full three dimensions, are highlighted with darker shading and poorly represented points with lighter shading. An observation is well represented in the 2D plot if its individual stress is low, where the stress for observation i is defined as follows:

$$\text{Stress}_i = \sqrt{\frac{\sum_{\ell \neq i} \left(d_{i\ell}^{\text{HD}} - d_{i\ell}^{\text{LD}}\right)^2}{\sum_{\ell \neq i} d_{i\ell}^{\text{HD}^2}}}. \tag{5.21}$$

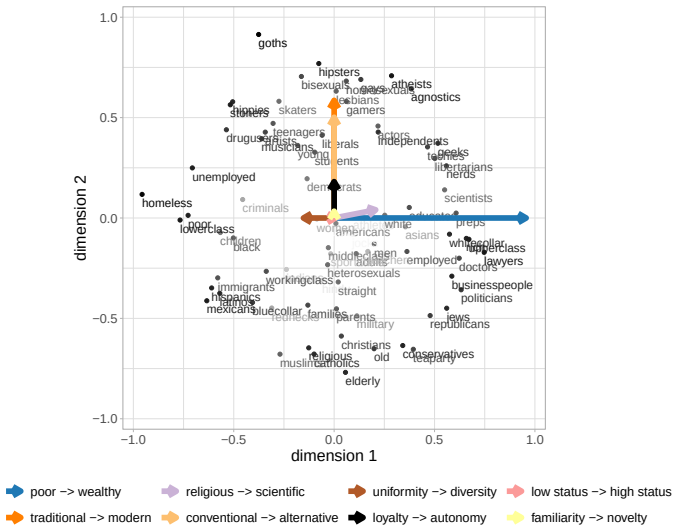
Colored arrows represent the variables (i.e. stereotypes) selected by BIOT to explain the embedding dimensions. The coordinates of the arrow heads are proportional to the coefficient values for the plotted dimensions. With the exception of the religious-scientific stereotype, each arrow runs parallel to one of the plotted dimensions, meaning that each selected variable explains only one of the dimensions.

For the first dimension of the MDS, the poor-wealthy stereotype is the most important for distinguishing social groups such as the homeless, the poor and the lower class (on the left in Figure 5.6(a) and (b)) from white collar people, the upper class, doctors and lawyers (on the right). For the second dimension, the BIOT model suggests that hipsters and goths are perceived as modern, alternative and autonomous, whereas religious people, Catholics and the elderly are perceived as the opposite: traditional, conventional and loyal. Finally, the third dimension represents the contrast between groups like teachers, Buddhists and Hindus (top of Figure 5.6(b) and (c)) and groups like jocks, athletes and sport fans (bottom), the groups at the top being perceived as altruistic, feminine, unassertive and smart, and the groups at the bottom being perceived as egoistic, masculine, competitive and not smart.

The advantage of analyzing more than two dimensions is that higher-dimensional embeddings have less reconstruction error, which is quantified by Kruskal's stress (see Eq. (5.5) in Section 5.1.2). Moreover, with BIOT, it is possible to observe the way in which low-dimensional embeddings approximate patterns from higher-dimensional embeddings. For example, when changing from 4D to 3D (0.169 to 0.207 in stress), BIOT associates the traditional/religious and conventional stereotypes with a single dimension, rather than two. This combination may explain the increase in stress in the 3D embedding, as two orthogonal trends are approximated by a single trend.

By adding dimensions, it is also possible to identify trends that are not apparent in lower-dimensional embeddings. While the original study did not identify

(a)



(b)

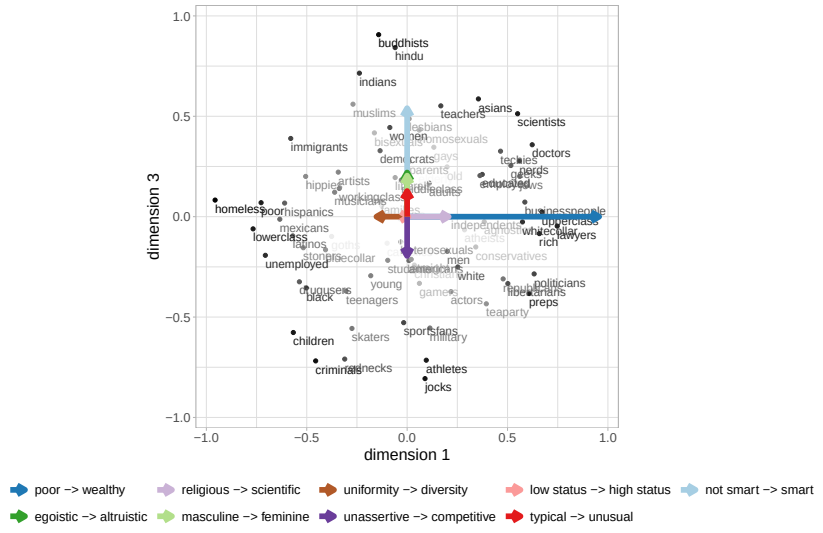


Figure 5.6: **BIOT explanations for the MDS embedding with three dimensions.** Observations (i.e. social groups) are plotted for pairs of embedding dimensions. The shading of points and labels ranges from white to black, where black represents the least stress (i.e. observation-level reconstruction error with respect to the full, three-dimensional embedding). The colored arrows represent the coefficients of variables selected by BIOT for the plotted dimensions, where arrow length is proportional to the coefficient value.

(c)

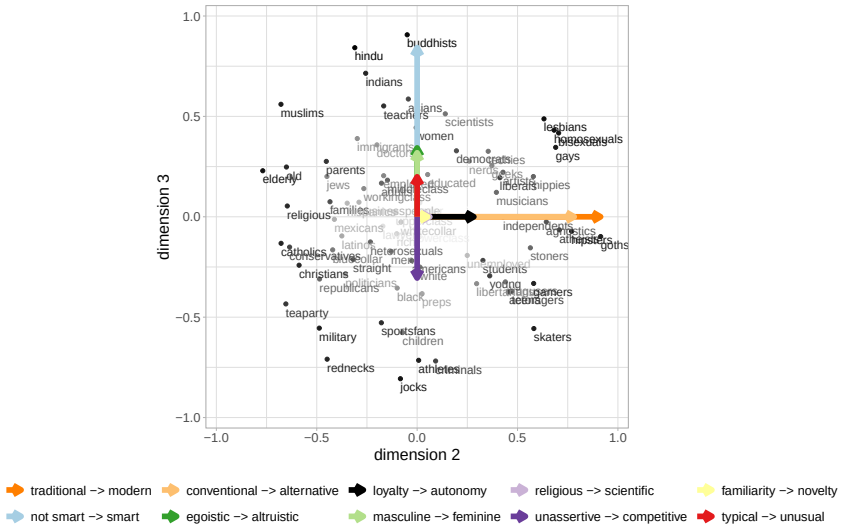


Figure 5.6: **BIOT explanations for the MDS embedding with three dimensions.** Observations (i.e. social groups) are plotted for pairs of embedding dimensions. The shading of points and labels ranges from white to black, where black represents the least stress (i.e. observation-level reconstruction error with respect to the full, three-dimensional embedding). The colored arrows represent the coefficients of variables selected by BIOT for the plotted dimensions, where arrow length is proportional to the coefficient value.

smartness as a relevant stereotype, the 3D analysis with BIOT identifies it as important for explaining a third dimension in the data. Indeed, social groups such as criminals and red necks are identified as egoistic, masculine and not smart by the participants. At the same time, the two other dimensions explained by BIOT correspond to the findings in the original paper, where MDS embeddings were explained by two quasi-orthogonal trends: the socio-economical status (represented here by wealthy) and the type of beliefs (represented here by the stereotypes conventional and traditional) (Koch, Imhoff, Dotsch, Unkelbach, and Alves, 2016).

This case study shows how insightful it is to use BIOT to analyze MDS embeddings with more than two dimensions. Given BIOT's sparsity and MSE performance, increasing the number of dimensions (and therefore reducing the stress) does not make the new embedding much more difficult to understand. Indeed, each new dimension is explained by small, generally disjoint sets of external variables. The results of this case study were presented to the main investigator of the original study (Koch, Imhoff, Dotsch, Unkelbach, and Alves, 2016), who found them coherent with current theory in psychology, while providing interesting insights.

$K = 3$	$K = 4$	$K = 5$
wealthy (0.14)	**wealthy** (0.14)	**wealthy** (0.13)
conservative (0.09)	conservative (0.08)	conservative (0.08)
conventional (0.07)	conventional (0.1)	conventional (0.06)
safety (0.07)	safety (0.06)	safety (0.05)
not smart (-0.03)	not smart (-0.01)	not smart (-0.03)
uniformity (0.02)	uniformity (0.01)	uniformity (0.02)
traditional (0.02)		traditional (0.02)
confident (0.01)	confident (0.01)	powerful (0.02)
loyalty (0.01)	loyalty (0.01)	communal (0.02)
competitive (-0.08)	competitive (-0.03)	competitive (-0.06)
masculine (-0.03)	masculine (-0.06)	masculine (-0.04)
communal (0.02)	communal (0.01)	
religious (0.02)	religious (0.01)	religious (0.04)
typical (-0.02)	typical (-0.01)	typical (-0.03)
egoistic (-0.1)	**egoistic** (-0.11)	**egoistic** (-0.09)
not smart (-0.1)	not smart (-0.06)	**not smart** (-0.1)
		uniformity (-0.01)
religious (-0.2)	**religious** (-0.2)	**religious** (-0.17)
wealthy (0.13)	wealthy (0.1)	wealthy (0.12)
familiarity (-0.06)	familiarity (-0.04)	familiarity (-0.05)
masculine (-0.05)	masculine (-0.05)	masculine (-0.05)
traditional (-0.02)	traditional (-0.02)	traditional (-0.05)
conventional (-0.01)		promotion (0.01)
intolerant (-0.01)		
loyalty (-0.01)		
	comfort (0.06)	**comfort** (0.1)
	friendly (-0.06)	friendly (-0.03)
	conventional (-0.04)	conventional (-0.05)
	loyalty (-0.02)	loyalty (-0.04)
		cold (0.02)
		untrustworthy (0.02)
		powerful (0.02)
		typical (-0.02)
		competitive (-0.01)
		threatening (0.01)
		change (0.06)
		communal (0.06)
		competitive (0.04)
		masculine (0.02)
		conservative (-0.01)
		friendly (0.01)
		powerful (0.01)
		promotion (0.01)

Table 5.3: **Lasso coefficients for three un-transformed embeddings of the Stereotypes dataset.** The embedding dimensions are in rows, and the Lasso coefficients for the transformed dimensions are in parentheses. The most important variables for each dimension are in bold.

$K = 3$	$K = 4$	$K = 5$
wealthy (0.28)	**wealthy** (0.26)	**wealthy** (0.22)
religious (-0.06)		powerful (0.05)
uniformity (0.05)	uniformity (0.06)	uniformity (0.01)
traditional (0.17)	traditional (0.04)	**traditional** (0.08)
religious (0.01)	**religious** (0.15)	**religious** (0.09)
	comfort (0.04)	comfort (0.04)
	prevention (0.02)	prevention (0.04)
conventional (0.15)	**conventional** (0.22)	**conventional** (0.14)
loyalty (0.05)	loyalty (0.07)	loyalty (0.01)
familiarity (0.01)		communal (0.01)
		friendly (0.01)
not smart (0.16)	**not smart** (0.13)	**not smart** (0.16)
egoistic (0.07)	egoistic (0.05)	egoistic (0.01)
masculine (0.06)	masculine (0.09)	masculine (0.04)
competitive (0.06)		competitive (0.05)
typical (0.04)	typical (0.03)	typical (0.03)
	intolerant (0.02)	
	familiarity (0.01)	
		conservative (0.14)
		masculine (0.03)
		preservation (0.03)

Table 5.4: **BIOT coefficients for three embeddings of the Stereotypes dataset.** The embedding dimensions are in rows, and the BIOT coefficients for the transformed dimensions are in parentheses. The most important variables for each dimension are in bold.

5.6 Conclusion

In this chapter, we proposed two methods, BIR and BIOT, that make it possible to explain MDS embeddings. While BIR can be used to explain 2D embeddings only, BIOT can be applied to embeddings of any number of dimensions.

BIR and BIOT were evaluated on datasets corresponding to real-world problems. We demonstrated that BIOT outperforms its competitors with respect to the interpretability of solutions. The analysis of MSE-sparsity curves revealed that, for the same level of MSE, BIOT provides models that are sparser, and thus easier to interpret. In order to demonstrate BIOT's ease of use, a case study based on a dataset from a psychological experiment on stereotypes was presented.

Currently, BIOT selects a single cluster of variables for each embedding dimension. In future work, a grouping penalty could be added to BIOT in order to identify multiple variable clusters for each embedding dimension. By grouping variables in a meaningful way, even models with many selected variables could be easily interpreted. This would be advantageous, for example, for datasets where mostly non-sparse models have the best test MSE.

5.7 Numerical Implementation

The experiments from this chapter were programmed in R. BIR and BIOT were implemented using in-house code that is publicly available on GitHub at https://github.com/AdrienBibal/BIR and https://github.com/rebeccamarion/BIOT. Sparse Reduced Rank Regression (SRRR) was performed using the function *srrr* from the `rrpack` package, Eigenvector PLS (ePLS) using the *pls_eigen* function from the `chemometrics` package, Lasso regression using the function *glmnet* from the `glmnet` package and Multidimensional Scaling (MDS) using the function *smacofSym* from the `smacof` package.

CHAPTER 6

Conclusion

Even the very wise cannot see all ends.

GANDALF

While *observation clustering* has received much attention in the scientific literature, *variable clustering* is a subject that deserves more exploration, as it can play an important role in the data analysis pipeline. First and foremost, variable clustering can be used to **identify grouping structure among variables**, which makes it possible to better understand the data. In the field of *transcriptomics*, for example, the identification of clusters of variables (i.e. gene pathways) is important for better understanding the function of genes and the mechanisms of disease. Second, variable clustering can be used to **reduce the dimensionality of the original dataset**, mapping the data to a lower-dimensional space in an easily interpretable manner. Indeed, a summary vector (e.g. centroid, medoid, latent variable, etc.) is calculated for each cluster of variables, representing the latent information contained in the cluster of variables that it represents. Finally, certain variable clustering methods can be used to **perform variable selection**. These methods, which allow some variables to be assigned to none of the clusters, identify the variables that explain the most variability in the data. This is helpful for better understanding the data, as well as for guiding future research. In biomarker studies, for instance, the variables selected via variable clustering can be further studied for the development of targeted treatments for a given disease.

In addition to establishing an in-depth background on the problem of variable clustering (see Chapter 2), this thesis presented a series of methods developed for tackling this problem. In Chapter 3, the problem of **unsupervised variable clustering** was addressed in the context of *metabolomics*. Adaptive Clustering around

Latent Variables (AdaCLV) was proposed for the purpose of clustering variables in spectroscopic data, such as Nuclear Magnetic Resonance (NMR) spectra. AdaCLV performs both variable selection and dimensionality reduction, and it can identify overlapping clusters of variables. Compared to related methods in the literature, AdaCLV was shown to be more robust to hyperparameter choice and have better clustering precision for a comparable level of recall.

Chapter 4 dealt with the problem of **supervised variable clustering**, where both variable selection and the identification of clusters are guided by a **univariate target**. Two methods were proposed in this chapter: Sparse Orthogonal Semi-Nonnegative Matrix Factorization (SOS-NMF), an unsupervised variable clustering method, and Variable Cluster Principal Component Regression (VC-PCR), which uses a supervised version of SOS-NMF to cluster variables, then predicts the target based on these clusters. By integrating supervision into its clustering step, VC-PCR is able to identify homogeneous clusters of variables that are relevant for the prediction task. Moreover, as a two-step method, VC-PCR is computationally efficient and can be easily applied to a variety of prediction tasks (i.e. regression, classification, survival analysis, etc.). Compared to other methods, VC-PCR was shown to have better prediction, variable selection and clustering performance.

Finally, Chapter 5 tackled the problem of **supervised variable clustering** when **multiple targets** are available. In this chapter, variable clustering was primarily used to give meaning to the targets, rather than to understand the predictors. Two methods were proposed for this purpose: Best Interpretable Rotation (BIR), which is built for cases where the target matrix has two dimensions, and Best Interpretable Orthogonal Transformation (BIOT), which can be used for multivariate target matrices with any number of dimensions. These methods can be used to interpret the orthogonal dimensions of embeddings generated by Multidimensional Scaling (MDS), a popular nonlinear dimensionality reduction technique in ecology and psychology. Numerical experiments demonstrated that, for a same level of model sparsity, BIR and BIOT generally estimate models with less prediction error.

A recurring theme throughout this thesis was the **use of sparsity penalties to remove irrelevant variables** from the clustering process. By assigning these variables to none of the clusters, the estimation of the cluster summary vectors and/or the prediction of a target can be improved. This strategy is particularly useful for variable clustering —as opposed to observation clustering —as most datasets contain at least some irrelevant variables.

Another important theme, and a contribution of this thesis, was the demonstration of the **relationship between variable clustering and matrix factorization methods**. This thesis is the first text to have demonstrated the link between certain variable clustering methods (e.g. Clustering around Latent Variables (CLV) and CLV $K+1$) and matrix factorization methods (e.g. K-means and Orthonormal Sparse Nonnegative PCA, see Chapter 3). It also demonstrated the similarities and

differences between SOS-NMF, one of our proposed variable clustering methods, and other important methods in the Nonnegative Matrix Factorization (NMF) framework (see Chapter 4). Finally, it proposed a reformulation of the objective function for Cluster Elastic Net (CEN), a supervised variable clustering method, re-framing it as a penalized matrix factorization problem (Chapter 4).

The many connections made between different methods, as well as the clarification of their relationship to matrix factorization problems, will make it easier for future researchers to build on existing work. Indeed, reformulating variable clustering as a matrix factorization problem has several significant benefits. Adding sparsity and other constraints to matrix factorization problems is well studied in the literature, and many solutions for optimization and initialization already exist. Inspiration can be taken from this existing body of work to develop other variable clustering methods or improve existing ones. Another advantage of the matrix factorization framework is that cluster membership degrees, rather than binary memberships, can be readily estimated. These membership degrees improve the interpretation of the clusters, as they indicate the relative importance of variables within a cluster. They also indicate the degree of confidence with which a given variable is assigned to a given cluster, making it possible to filter out possible false positives. Clustering variables via matrix factorization is therefore one of the most promising directions for future research.

While this thesis explored a variety of topics related to variable clustering, some research questions remain open to future work. For example, there is currently no research on **how to select hyperparameters that simultaneously optimize prediction and variable clustering**. One solution to this problem would be to propose selection criteria that quantify both prediction and variable clustering quality using a single, global measure, then to perform cross-validation to select the hyperparameters that maximize or minimize this measure. The quality of prediction and variable clustering could be measured individually, then combined using an approach similar to the F_1 score, which uses a harmonic mean to aggregate two potentially opposing quality measures. The development of such a global measure could draw on a vast literature on criteria for prediction (see e.g. Chen and Chen (2008), Kim, Kwon, and Choi (2012), Sun, Wang, and Fang (2013), and Jansen (2014)), as well as criteria for observation clustering quality (i.e. *cluster validity indices*, see e.g. Chiang and Mirkin (2010), Arbelaitz, Gurrutxaga, Muguerza, Pérez, and Perona (2013), and Kaczyńska, Marion, and von Sachs (2020)).

Another approach to this issue would be to modify existing supervised variable clustering methods such that prediction performance is more closely related to variable clustering performance. This would make it possible to use a single measure of quality (prediction or clustering) to optimize the hyperparameters with respect to both objectives. However, for certain methods, such as Cluster Elastic Net (CEN, see Chapter 4), models with low prediction error can be achieved regardless of the quality of the variable clustering. As a result, selecting hyperparameters according

to prediction error results in clusters of arbitrary quality. Rather than developing a global selection criterion, as proposed above, it may be better to modify the objective function of the method so that low prediction error can only be achieved for models with high cluster quality.

To conclude on a more general note, this thesis project has made my supervisors, collaborators and I better appreciate the difficulty of a research problem that at first appeared straightforward. Throughout this project, we were confronted by a series of challenges: datasets with exceedingly complicated correlation structure, algorithms that failed to converge, hyperparameters that were difficult to tune and results that were difficult to evaluate due to the lack of an available ground-truth (e.g. the true clusters), among other things. All in all, these challenges demonstrate the richness of this research area, and the potential for other researchers to continue related work in this field.

References

Abdi, Hervé (2010). "Partial least squares regression and projection on latent structure regression (PLS Regression)". In: **Wiley Interdisciplinary Reviews: Computational Statistics** 2.1, pp. 97–106.

Arbelaitz, Olatz, Ibai Gurrutxaga, Javier Muguerza, Jesús M Pérez, and Inigo Perona (2013). "An extensive comparative study of cluster validity indices". In: **Pattern Recognition** 46.1, pp. 243–256.

Atif, Syed Muhammad, Sameer Qazi, and Nicolas Gillis (2019). "Improved SVD-based initialization for nonnegative matrix factorization using low-rank correction". In: **Pattern Recognition Letters** 122, pp. 53–59.

Bauckhage, Christian (2015). "K-means clustering is matrix factorization". In: **arXiv preprint arXiv:1512.07548**.

Bellman, Richard E (1961). **Adaptive control processes: a guided tour**. Princeton university press.

Bibal, Adrien, Rebecca Marion, and Benoît Frénay (2018). "Finding the Most Interpretable MDS Rotation for Sparse Linear Models based on External Features". In: **Proceedings of the European Symposium on Artificial Neural Networks**. Bruges, Belgium, pp. 537–542.

Bibal, Adrien, Rebecca Marion, Rainer von Sachs, and Benoît Frénay (2021). "BIOT: Explaining multidimensional nonlinear MDS embeddings using the Best Interpretable Orthogonal Transformation". In: **Neurocomputing** 453, pp. 109–118.

Bingol, Kerem, Lei Bruschweiler-Li, Dawei Li, Bo Zhang, Mouzhe Xie, and Rafael Brüschweiler (2016). "Emerging new strategies for successful metabolite identification in metabolomics". In: **Bioanalysis** 8.6, pp. 557–573.

Boccard, Julien and Serge Rudaz (2014). "Harnessing the complexity of metabolomic data with chemometrics". In: **Journal of Chemometrics** 28.1, pp. 1–9.

Bondell, Howard D and Brian J Reich (2008). "Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with OSCAR". In: **Biometrics** 64.1, pp. 115–123.

Borcard, Daniel and Pierre Legendre (1994). "Environmental control and spatial structure in ecological communities: an example using oribatid mites (Acari, Oribatei)". In: **Environmental and Ecological Statistics** 1.1, pp. 37–61.

Borcard, Daniel, Pierre Legendre, and Pierre Drapeau (1992). "Partialling out the spatial component of ecological variation". In: **Ecology** 73.3, pp. 1045–1055.

Bühlmann, Peter, Philipp Rütimann, Sara van de Geer, and Cun-Hui Zhang (2013). "Correlated variables in regression: clustering and sparse estimation". In: **Journal of Statistical Planning and Inference** 143.11, pp. 1835–1858.

Bunea, Florentina, Christophe Giraud, Xi Luo, Martin Royer, Nicolas Verzelen, et al. (2020). "Model assisted variable clustering: minimax-optimal recovery and algorithms". In: **The Annals of Statistics** 48.1, pp. 111–137.

Bunea, Florentina, Yiyuan She, Marten H Wegkamp, et al. (2012). "Joint variable and rank selection for parsimonious estimation of high-dimensional matrices". In: **The Annals of Statistics** 40.5, pp. 2359–2388.

Chen, Jiahua and Zehua Chen (2008). "Extended Bayesian information criteria for model selection with large model spaces". In: **Biometrika** 95.3, pp. 759–771.

Chen, Lisha and Jianhua Z Huang (2012). "Sparse reduced-rank regression for simultaneous dimension reduction and variable selection". In: **Journal of the American Statistical Association** 107.500, pp. 1533–1545.

Chiang, M. and B. Mirkin (2010). "Intelligent choice of the number of clusters in k-means clustering: an experimental study with different cluster spreads". In: **Journal of Classification** 27.1, pp. 3–40.

Chicco, Davide (2017). "Ten quick tips for machine learning in computational biology". In: **BioData mining** 10.1, pp. 1–17.

Chicco, Davide and Giuseppe Jurman (2020). "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation". In: **BMC genomics** 21.1, pp. 1–13.

Chicco, Davide, Niklas Tötsch, and Giuseppe Jurman (2021). "The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation". In: **BioData mining** 14.1, pp. 1–22.

Christidis, Anthony-Alexander, Laks Lakshmanan, Ezequiel Smucler, and Ruben Zamar (2020). "Split Regularized Regression". In: **Technometrics** 62.3, pp. 330–338.

Chun, Hyonho and Sündüz Keleş (2010). "Sparse partial least squares regression for simultaneous dimension reduction and variable selection". In: **Journal of the Royal Statistical Society: Series B (Statistical Methodology)** 72.1, pp. 3–25.

Cloarec, Olivier, Marc E Dumas, Johan Trygg, Andrew Craig, Richard H Barton, John C Lindon, Jeremy K Nicholson, and Elaine Holmes (2005). "Evaluation of the orthogonal projection on latent structure model limitations caused by chemical shift variability and improved visualization of biomarker changes in

1H NMR spectroscopic metabonomic studies". In: **Analytical chemistry** 77.2, pp. 517–526.

Dai, Ling-Yun, Jin-Xing Liu, Rong Zhu, Xiang-Zhen Kong, Mi-Xiao Hou, and Sha-Sha Yuan (2018). "Sparse Orthogonal Nonnegative Matrix Factorization for Identifying Differentially Expressed Genes and Clustering Tumor Samples". In: **2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)**. IEEE, pp. 1332–1337.

Daye, Z John and X Jessie Jeng (2009). "Shrinkage and model selection with correlated variables via weighted fusion". In: **Computational Statistics & Data Analysis** 53.4, pp. 1284–1298.

Dhillon, Inderjit S, Edward M Marcotte, and Usman Roshan (2003). "Diametrical clustering for identifying anti-correlated gene clusters". In: **Bioinformatics** 19.13, pp. 1612–1619.

Ding, Chris and Xiaofeng He (2004). "K-means clustering via principal component analysis". In: **Proceedings of the twenty-first international conference on Machine learning**, p. 29.

Ding, Chris, Xiaofeng He, and Horst D Simon (2005). "On the equivalence of nonnegative matrix factorization and spectral clustering". In: **Proceedings of the 2005 SIAM international conference on data mining**. SIAM, pp. 606–610.

Ding, Chris, Tao Li, Wei Peng, and Haesun Park (2006). "Orthogonal nonnegative matrix t-factorizations for clustering". In: **Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining**, pp. 126–135.

Ding, Chris HQ, Tao Li, and Michael I Jordan (2008). "Convex and semi-nonnegative matrix factorizations". In: **IEEE transactions on pattern analysis and machine intelligence** 32.1, pp. 45–55.

Durbin, Blythe P, Johanna S Hardin, Douglas M Hawkins, and David M Rocke (2002). "A variance-stabilizing transformation for gene-expression microarray data". In: **Bioinformatics** 18.suppl_1, S105–S110.

Efron, B., T. Hastie, I. Johnstone, and R. Tibshirani (2004). "Least angle regression". In: **The Annals of Statistics** 32.2, pp. 407–499.

Gaujoux, Renaud and Cathal Seoighe (2010). "A flexible R package for nonnegative matrix factorization". In: **BMC Bioinformatics** 11.1, p. 367. ISSN: 1471-2105. DOI: 10.1186/1471-2105-11-367. URL: http://www.biomedcentral.com/1471-2105/11/367.

Golotvin, Sergey and Antony Williams (2000). "Improved baseline recognition and modeling of FT NMR spectra". In: **Journal of Magnetic Resonance** 146.1, pp. 122–125.

Golub, Gene H and Charles F Van Loan (2013). **Matrix Computations**. Johns Hopkins University Press.

Gower, John C and Garmt B Dijksterhuis (2004). **Procrustes Problems**. Oxford University Press.

Guisset, Séverine, Manon Martin, and Bernadette Govaerts (2019). "Comparison of PARAFASCA, AComDim, and AMOPLS approaches in the multivariate GLM modelling of multi-factorial designs". In: **Chemometrics and Intelligent Laboratory Systems** 184, pp. 44–63.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). **The Elements of Statistical Learning: Data Mining, Inference, and Prediction**. Springer-Verlag New York.

Hastie, Trevor, Robert Tibshirani, and Martin Wainwright (2015). **Statistical Learning with Sparsity: the Lasso and Generalizations**. Chapman and Hall/CRC.

Jacob, Laurent, Guillaume Obozinski, and Jean-Philippe Vert (2009). "Group lasso with overlap and graph lasso". In: **Proceedings of the 26th annual international conference on machine learning**, pp. 433–440.

Jansen, Maarten (2014). "Information criteria for variable selection under sparsity". In: **Biometrika** 101.1, pp. 37–55.

Jardine, Nick and Cornelis Joost van Rijsbergen (1971). "The use of hierarchic clustering in information retrieval". In: **Information storage and retrieval** 7.5, pp. 217–240.

Jaworska, Natalia and Angelina Chupetlovska-Anastasova (2009). "A review of multidimensional scaling (MDS) and its utility in various psychological domains". In: **Tutorials in Quantitative Methods for Psychology** 5.1, pp. 1–10.

Kaczyńska, Sara, Rebecca Marion, and Rainer von Sachs (2020). "Comparison of cluster validity indices and decision rules for different degrees of cluster separation". In: **ESANN**.

Kim, Hyunsoo and Haesun Park (2007). "Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis". In: **Bioinformatics** 23.12, pp. 1495–1502.

Kim, Jingu and Haesun Park (2008). **Sparse nonnegative matrix factorization for clustering**. Tech. rep. Georgia Institute of Technology.

Kim, Seyoung and Eric P Xing (2009). "Statistical estimation of correlated genome associations to a quantitative trait network". In: **PLoS Genet** 5.8, e1000587.

Kim, Yongdai, Sunghoon Kwon, and Hosik Choi (2012). "Consistent model selection criteria on high dimensions". In: **The Journal of Machine Learning Research** 13, pp. 1037–1057.

Koch, A., R. Imhoff, R. Dotsch, C. Unkelbach, and H. Alves (2016). "The ABC of stereotypes about groups: Agency/socioeconomic success, conservative–progressive beliefs, and communion". In: **Journal of Personality and Social Psychology** 110.5, pp. 675–709.

Kruskal, Joseph B and Myron Wish (1978). **Multidimensional Scaling**. Sage.

Langville, Amy N, Carl D Meyer, Russell Albright, James Cox, and David Duling (2006). "Initializations for the nonnegative matrix factorization". In: **Proceedings of the twelfth ACM SIGKDD international conference on knowledge discovery and data mining**. Citeseer, pp. 23–26.

Lee, John A and Michel Verleysen (2007). **Nonlinear Dimensionality Reduction**. Springer.

Lee, Seunghak, Jun Zhu, and Eric P Xing (2010). "Adaptive multi-task lasso: with application to eQTL detection". In: **Advances in neural information processing systems**, pp. 1306–1314.

Lee, Woojoo, Donghwan Lee, Youngjo Lee, and Yudi Pawitan (2011). "Sparse canonical covariance analysis for high-throughput data". In: **Statistical Applications in Genetics and Molecular Biology** 10.1, pp. 1–24.

Li, Qiuwei, Zhihui Zhu, and Gongguo Tang (2019). "Alternating minimizations converge to second-order optimal solutions". In: **International Conference on Machine Learning**. PMLR, pp. 3935–3943.

Liland, Kristian Hovde (2011). "Multivariate methods in metabolomics–from preprocessing to dimension reduction and statistical analysis". In: **TrAC Trends in Analytical Chemistry** 30.6, pp. 827–841.

Lin, Ja-an, Hongtu Zhu, Rebecca Knickmeyer, Martin Styner, John Gilmore, and Joseph G Ibrahim (2012). "Projection regression models for multivariate imaging phenotype". In: **Genetic Epidemiology** 36.6, pp. 631–641.

Lindon, John C, Jeremy K Nicholson, Elaine Holmes, Henrik Antti, Mary E Bollard, Hector Keun, Olaf Beckonert, Timothy M Ebbels, Michael D Reily, Donald Robertson, et al. (2003). "Contemporary issues in toxicology the role of metabonomics in toxicology and its evaluation by the COMET project". In: **Toxicology and applied pharmacology** 187.3, pp. 137–146.

Lindon, John C, Jeremy K Nicholson, Elaine Holmes, and Jeremy R Everett (2000). "Metabonomics: metabolic processes studied by NMR spectroscopy of biofluids". In: **Concepts in Magnetic Resonance: An Educational Journal** 12.5, pp. 289–320.

Ma, Shuangge, Xiao Song, and Jian Huang (2007). "Supervised group Lasso with applications to microarray data analysis". In: **BMC bioinformatics** 8.1, pp. 1–17.

Maaten, Laurens van der and Geoffrey Hinton (2008). "Visualizing data using t-SNE". In: **Journal of Machine Learning Research** 9.Nov, pp. 2579–2605.

Mai, Qing and Xin Zhang (2019). "An iterative penalized least squares approach to sparse canonical correlation analysis". In: **Biometrics**.

Marion, Rebecca, Adrien Bibal, and Benoît Frénay (2019). "BIR: A Method for Selecting the Best Interpretable Multidimensional Scaling Rotation using External Variables". In: **Neurocomputing** 342, pp. 83–96.

Marion, Rebecca, Bernadette Govaerts, and Rainer von Sachs (2020). "AdaCLV for Interpretable Variable Clustering and Dimensionality Reduction of Spectroscopic Data". In: **Chemometrics and Intelligent Laboratory Systems** 206, p. 104169.

Marion, Rebecca, Johannes Lederer, Bernadette Govaerts, and Rainer von Sachs (2021). "VC-PCR: A Prediction Method based on Supervised Variable Selection and Clustering". In: **To be submitted**.

Markley, John L, Rafael Brüschweiler, Arthur S Edison, Hamid R Eghbalnia, Robert Powers, Daniel Raftery, and David S Wishart (2017). "The future of NMR-based metabolomics". In: **Current opinion in biotechnology** 43, pp. 34–40.

Matthews, Brian W (1975). "Comparison of the predicted and observed secondary structure of T4 phage lysozyme". In: **Biochimica et Biophysica Acta (BBA)-Protein Structure** 405.2, pp. 442–451.

McInnes, Leland, John Healy, and James Melville (2018). "Umap: Uniform manifold approximation and projection for dimension reduction". In: **arXiv preprint arXiv:1802.03426**.

Muncey, Harriet J, Rebecca Jones, Maria De Iorio, and Timothy MD Ebbels (2010). "MetAssimulo: simulation of realistic NMR metabolic profiles". In: **BMC bioinformatics** 11.1, p. 496.

Nayak, Janmenjoy, Bighnaraj Naik, and HS Behera (2015). "Fuzzy C-means (FCM) clustering algorithm: a decade review from 2000 to 2014". In: **Computational intelligence in data mining-volume 2**. Springer, pp. 133–149.

Nie, Feiping, Heng Huang, Xiao Cai, and Chris H Ding (2010). "Efficient and robust feature selection via joint $\ell 2$, 1-norms minimization". In: **Advances in neural information processing systems**, pp. 1813–1821.

Obozinski, Guillaume, Ben Taskar, and Michael Jordan (2006). "Multi-task feature selection". In: **Statistics Department, UC Berkeley, Tech. Rep** 2.2.2.

Park, Mee Young, Trevor Hastie, and Robert Tibshirani (2007). "Averaged gene expressions for regression". In: **Biostatistics** 8.2, pp. 212–227.

Parsons, Helen M, Christian Ludwig, Ulrich L Günther, and Mark R Viant (2007). "Improved classification accuracy in 1-and 2-dimensional NMR metabolomics data using the variance stabilising generalised logarithm transformation". In: **BMC bioinformatics** 8.1, p. 234.

Rousseau, Réjane (2011). "Statistical contribution to the analysis of metabonomics data in 1H NMR spectroscopy". In: **Université catholique de Louvain**.

Sharma, Dhruv B, Howard D Bondell, and Hao Helen Zhang (2013). "Consistent group identification and variable selection in regression with correlated predictors". In: **Journal of Computational and Graphical Statistics** 22.2, pp. 319–340.

She, Yiyuan et al. (2010). "Sparse regression with exact clustering". In: **Electronic Journal of Statistics** 4, pp. 1055–1096.

Shen, Xiaotong and Hsin-Cheng Huang (2010). "Grouping pursuit through a regularization solution surface". In: **Journal of the American Statistical Association** 105.490, pp. 727–739.

Shen, Xiaotong, Hsin-Cheng Huang, and Wei Pan (2012). "Simultaneous supervised clustering and feature selection over a graph". In: **Biometrika** 99.4, pp. 899–914.

Sigg, Christian D and Joachim M Buhmann (2008). "Expectation-maximization for sparse and non-negative PCA". In: **Proceedings of the 25th international conference on Machine learning**, pp. 960–967.

Smeenk-Enserink, Nellie and PJM Van Der Aart (1974). "Correlations between distributions of hunting spiders (Lycosidae, Ctenidae) and environmental characteristics in a dune area". In: **Netherlands Journal of Zoology** 25.1, pp. 1–45.

Sun, Shiliang (2013). "A survey of multi-view machine learning". In: **Neural Computing and Applications** 23.7-8, pp. 2031–2038.

Sun, Wei, Junhui Wang, and Yixin Fang (2013). "Consistent selection of tuning parameters via variable selection stability". In: **The Journal of Machine Learning Research** 14.1, pp. 3419–3440.

Suo, Xiaotong, Victor Minden, Bradley Nelson, Robert Tibshirani, and Michael Saunders (2017). "Sparse canonical correlation analysis". In: **arXiv preprint arXiv:1705.10865**.

Thioulouse, Jean (2011). "Simultaneous analysis of a sequence of paired ecological tables: A comparison of several methods". In: **The Annals of Applied Statistics**, pp. 2300–2325.

Tibshirani, Robert, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight (2005). "Sparsity and smoothness via the fused lasso". In: **Journal of the Royal Statistical Society: Series B (Statistical Methodology)** 67.1, pp. 91–108.

Tutz, Gerhard and Jan Ulbricht (2009). "Penalized regression with correlation-based penalty". In: **Statistics and Computing** 19.3, pp. 239–253.

Varmuza, Kurt and Peter Filzmoser (2016). **Introduction to Multivariate Statistical Analysis in Chemometrics**. CRC press.

Verneaux, J (1973). "Cours d'eau de Franche-Comté (Massif du Jura). Recherches écologiques sur le réseau hydrographique du Doubs. Essai de biotypologie." PhD thesis. Université de Besançon.

Vigneau, Evelyne (2016). "Dimensionality reduction by clustering of variables while setting aside atypical variables". In: **Electronic Journal of Applied Statistical Analysis** 9.1, pp. 134–153.

Vigneau, Evelyne and EM Qannari (2003). "Clustering of variables around latent components". In: **Communications in Statistics-Simulation and Computation** 32.4, pp. 1131–1150.

Ward Jr, Joe H (1963). "Hierarchical grouping to optimize an objective function". In: **Journal of the American statistical association** 58.301, pp. 236–244.

Wilms, Ines and Christophe Croux (2015). "Sparse canonical correlation analysis from a predictive point of view". In: **Biometrical Journal** 57.5, pp. 834–851.

Witten, Daniela M, Ali Shojaie, and Fan Zhang (2014). "The cluster elastic net for high-dimensional regression with unknown variable grouping". In: **Technometrics** 56.1, pp. 112–122.

Witten, Daniela M, Robert Tibshirani, and Trevor Hastie (2009). "A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis". In: **Biostatistics** 10.3, pp. 515–534.

Yang, Sen, Lei Yuan, Ying-Cheng Lai, Xiaotong Shen, Peter Wonka, and Jieping Ye (2012). "Feature grouping and selection over an undirected graph". In: **Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining**, pp. 922–930.

Yengo, Loïc, Julien Jacques, and Christophe Biernacki (2014). "Variable clustering in high dimensional linear regression models". In: **Journal de la Société Française de Statistique** 155.2, pp. 38–56.

Yuan, Ming and Yi Lin (2006). "Model selection and estimation in regression with grouped variables". In: **Journal of the Royal Statistical Society: Series B (Statistical Methodology)** 68.1, pp. 49–67.

Zass, Ron and Amnon Shashua (2007). "Nonnegative sparse PCA". In: **Advances in neural information processing systems**. Citeseer, pp. 1561–1568.