POLITECHNIKA WROCŁAWSKA

Wydział Matematyki

ROZPRAWA DOKTORSKA

mgr. Piotr Jakub Sobczyk

Identifying low-dimensional structures through model selection in high-dimensional data

Promotor
dr hab. Małgorzata Bogdan

Wrocław 2019

Acknowledgements

First of all, I would like to thank my supervisor Professor Malgorzata Bogdan.
Without her ideas, comments and enthusiasm this thesis would not be even half as interesting. I am very grateful for the time she spent on our talks. Her inexhaustible energy and positive attitude made it possible for me to collaborate with researchers all around the globe. It made time of my PhD studies worthwhile and I will always remember that.

There are a lot of people who shaped in some way this thesis. I would like to thank them all.
Julie Josse. For her expertise in the field of dimensionality reduction and for her comments that taught me creating better statistical software.
Michal Burdukiewicz. Though my work with him is not part of this thesis, his enthusiasm inspired me and drove me to learn bioinformatics.
Sangkyun Lee. For hours spend together on trying out hundreds of formulations and algorithms for graphical SLOPE.
Emmanuel Candes. For his suggestion to construct ADMM algorithm for solving graphical SLOPE.
Piotr Graczyk. For his inspiration and advice on the theorems on PESEL.
Andrzej Dabrowski. Who first kindled in me the spark of interest for statistics.
I would also like to thank other people I have worked with through these years: Damian Brzyski, Michal Kos, Malgorzata Kotulska, Pawel Mackiewicz, Stefan Rodinger, Chiara Sabbati, Mateusz Staniak, Stanislaw Sulicki, Stanislaw Wilczynski.

It was a pleasure.

To my parents and grandparents.
I owe them my craving for truth and knowledge.
Without those I would have never begun this thesis.
And to Agata. Without her love I would have never finished it.

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

People are generating and collecting more and more data[Baiju]. Increase is not only in the number of observations like page views on internet website, comments in social media or patients in genetics study. At the same time we make more and more measurements which lead to increase in data dimensionality. Those extra variables could be additional characteristics of user web session, longer sequences of words (n-grams) or more Single Nucleotide Polimorphisms (SNP) from genome. This vast amount of data poses a number of challenges. Data is abundant, but not always carefully pre-selected. As digital storage becomes cheaper and cheaper, there is not too much thought on how data will be actually processed and modeled. An example of such a case could be tracking data on websites. Every action users make is recorded and stored, but reasonable processing of this data remains a challenge. This problem also stems out from the fact that sometimes we honestly do not know what can be found in the data, and we plan on collecting as much as possible and then do data mining. Another example could be genomic data, in which we collect information about expression of thousands of genes and genetic variations of hundreds of thousands of nucleotides (SNP) in DNA, and hope to find the ones that are connected with certain phenotype or medical condition. There is a big pressure on transforming data into information and knowledge. This is non trivial and a number of pitfalls await explorers.

When the number of variables p is greater than the number of observations n, then most of the classical statistical models become non-identifiable. Let us use as an example linear model with $n \times p$ matrix $\mathbf{X}$. In such case $\mathbf{X}^T\mathbf{X}$ becomes non-invertible, which leads to infinite number of least-squares solutions. For people without mathematical background this does not sound dangerous. However, even when $n \approx p$, we have a problem of high variance of estimators, which makes it difficult to draw conclusions from the data. This problem exists for new estimation methods in machine learning, although it is sometimes 'hidden' and cannot be as easily quantified as in the example of linear regression. However in many cases it is reasonable to assume that despite the fact that data is high dimensional, its intrinsic dimension is much smaller. This creates an opportunity. Reduction of data dimensionality might lead to more accurate estimators

The real world examples presented in this thesis come from the field of genetics. One of the important problems of modern medical studies is the identification of genetic pathways. Pathways, known also as gene regulatory networks, are groups of molecular regulators that govern gene expressions. As a result, multiple genes act together, because they have the same cause. It could be some DNA sequence or some protein. The identification of such pathways is a step to understand underlying biological processes. There are multiple applications of that knowledge, one of which is predicting genetic based diseases or expected outcome of the therapy. One would collect the data (on a gene expression level) from both healthy and diseased individuals. Ultimate goal would be to distinguish these two groups based on genetic information. As number of genes in humans is around 22 thousands, the number of variables can be much larger than the number of individuals $p >> n$. Because of that, classical statistical

methods do not work as expected. Motivation behind the methods presented in this thesis is to identify those genes that 'work together'.

One obstacle that we encounter in such a setting is interpretation of statistical model. This is an important issue for getting useful, human understandable knowledge from the data. An example question could be about the effect of given variable on outcome. Let us consider sign of coefficients in the linear model. If coefficients of two variables are of different signs, then one might expect them to have contradictory effect on the outcome. However this intuition is not justifiable when variance is high. This poses problem for data analysis made for scientific research, for which interpretation of the model is crucial. But there are some even more gruesome threats. Estimator with high variability has a tendency for overfitting. This manifests heavily when p is very large. In such a case, great performance on training data is usually not matched by on test data. The problem with model identification is sometimes discarded by 'practioners', however poor prediction is a very pragmatic problem. Its roots is sometimes overfitting which can be addressed by previous dimensionality reduction. This is what makes this topic a very practical one.

From the analysis of these challenges in the modern data analysis, it becomes obvious that there is a need for the methods for data dimensionality reduction. The justification for this reduction and focusing only on sparse models, was well put by [Hastie et al., 2009] who coined phrase 'bet on sparsity'. If the true model is sparse, then models that assume sparsity have a good shot at finding it. If true model is dense, then no method would perform well unless $n >> p$. This idea that was thought of as a justification for ℓ_1 penalty in LASSO describes well the principal behind most of the ideas and research presented in this thesis.

In this dissertation I focus on the problem of unsupervised dimensionality reduction. Data we consider does not include any labels. Our goal is to infer its structure and based on that knowledge reduce dimension of the ambient space. It could be further used to model some outcome variable, but this is not covered in this thesis. Our focus is on two different data models.

First one is associated with graphical models [Lauritzen, 1996]. We think of variables as vertices in the graph. Some of them are connected by edges. Edge between two vertices denotes dependency, which can be defined in multiple ways. The goal of the estimation is to identify all of the edges with no or little false positives. This can be viewed as a feature selection problem. Starting from the complete graph we simplify its structure by discarding some edges.

Second kind of model is the one used in subspace clustering research. It is not explicitly stated, but one can find it in papers on computer vision, e.g. Elhamifar and Vidal [2009], Vidal and Favaro [2014], McWilliams and Montana [2014], Agarwal and Mustafa [2004] and it is more rigorously treated in statistical literature e.g.[Soltanolkotabi et al., 2013], [Tipping and Bishop, 1999b]. In this model we assume that data is generated by a small number of latent variables. Groups of variables lie in the same subspace. This kind of assumption on data is significantly different than the one underlying graphical model. Since latent variables are not observed, we cannot use feature selection and should instead transform data $\mathbf{X}$ into a smaller dimension matrix that consists of latent variables.

Going back to our genetics data motivation. Graph model is proper for the data, when there are no latent variables. We would assume that genes are in some way dependent and that dependency should be reflected in the structure of the graph. Ideally, separate connected components should correspond to different pathways.
Second model assumes that true regulators are not necessarily present in the data. Those are

unobserved, latent variables. In this case our goal would be to identify genes that are in the same gene network and to quantify effect of (latent) regulatory elements.

In chapter 2 we introduce all the necessary mathematical and statistical background. We start with basic theory and two algorithms for solving convex optimization problems. We then discuss model selection in linear models focusing on penalized regressions with different kinds of penalty functions. Subsequently, we introduce fixed and probabilistic models for principal component analysis. After that model and methods for subspace clustering are discussed. Finally a brief introduction to Gaussian graphical models is provided.

In chapter 3 we consider the classical tool for exploratory data analysis called Principal Component Analysis. We introduce a new criterion PEnalized SEmi-integrated Likelihood (*PESEL*) for a dimensionality reduction by selecting some number of principal components. Our approach stems from Bayesian approach, more specifically maximum a posteriori (MAP) rule. To minimize number of prior assumptions we use Laplace method for integrals to get closed formula for posterior probability. For this criterion we provide a detailed derivation. We also proof that under some mild conditions, PESEL is consistent meaning that when $n \to \infty$ number of estimated principal components is almost surely equal to the true number of principal components. Similar probabilistic model was considered before by [Tipping and Bishop, 1999a] and [Minka, 2000]. We unified that approach with a case when $p > n$, which is of the main interest in this thesis. We argue that one should use appropriate version based on data dimensions. Furthermore PESEL proves to be more robust than state-of-the-art methods [Choi et al., 2014], [Josse and Husson, 2012], [Minka, 2000, eq. 76] for estimating number of signal related Principal Components. At the same time, because its computation burden is equivalent to performing PCA, PESEL is much faster than full Bayesian methods. Extensive simulation study is presented, as well as real data example that proves usefulness of the method. This criterion can be used by researchers as a method for denoising the data. Number of first principal components should be retained as a signal while the rest is truncated as noise. PESEL was implemented in R package that is available on github `https://github.com/psobczyk/pesel`.

In chapter 4 we introduce new method for subspace clustering called *Multiple Latent Components Clustering (MLCC)*. Just like *PESEL*, method stems from Bayesian perspective. Specific prior distributions are assumed and their interpretation is discussed. Formula for maximum a posteriori is then approximated using Laplace method for integrals. Because our estimation is based on ℓ_0 norm, finding optimal variables clustering is intractable problem. Because of that we introduce an efficient heuristic algorithm that combines classical k-medoids algorithm with Bayesian model selection. There were several attempts for using k-means for subspace clustering (see [Agarwal and Mustafa, 2004] and [Chavent et al., 2012]). However they lacked mathematical rigor and experienced limitations in comparing subspaces of different dimensions. Because *MLCC* is computationally exhaustive, a number of heuristics are demonstrated that significantly speed it up. To minimize chance of being stuck in local maximum, we run algorithm multiple times and provide some insight on reasonable warm start. Simulations study reassures us, that using heuristics does not significantly deteriorate methods performance. The is introduced and its efficacy is demonstrated, again in extensive simulation study against methods found in the recent literature. Varclust was implemented in R package that is available on github `https://github.com/psobczyk/varclust`.

In chapter 5 we introduce a new method *graphical SLOPE* for estimating precision matrix in the problem of Gaussian-Markov random field. Method is an extension of the popular glasso ([Banerjee et al., 2008a] and [Friedman et al., 2008b]). Instead of regularizing likelihood with ℓ_1 penalty, we use sorted ℓ_1. This extension was inspired by the method SLOPE [Bogdan et al.,

2015a] for sparse linear regression, where, under certain conditions and for specific penalty, false discovery rate (FDR) is controlled. *Graphical SLOPE* proves to have much higher power at the cost of treating few zero entries in precision matrix as non-zeros, i.e. introducing small number of false discoveries. In various simulation scenarios we show that *Graphical SLOPE* can in fact control FDR. We introduce a choice of λ parameters that, under certain conditions, controls FWER, just like glasso does. We prove that our method is guaranteed to have higher power. Graphical SLOPE and ADMM algorithm for solving SLOPE were efficiently implemented in R package that is available on github `https://github.com/psobczyk/gslope`.

2 Mathematical introduction

Overview

In this chapter we shall introduce notation and the most important notions, techniques and methods that are going to be used throughout this thesis.

We start this chapter with definitions of optimization problem. We then focus on convex optimization and introduce two techniques that are used to solve special types of convex problems. This theory is useful, as numerous statistical methods are defined by being optimal in some sense, usually being a minimizer of a given function which often can be treated as convex optimization problem. Then we state some definitions regarding multiple testing and maximum likelihood. This leads to regularization and variable selection methods. As an example we selected linear regression which is well described in a literature. Next topic is dimensionality reduction. We first define PCA and do overview of techniques used to estimate number of principal components. Then we define a problem of subspace clustering and give some background behind two methods we shall compare to in a simulation study. Finally we move to Gaussian graphical models, make a brief introduction and state interpretation of the precision matrix in multivariate Gaussian distribution.

2.1 Convex optimization

In this section we use notation from [Boyd and Vandenberghe, 2004].

2.1.1 Optimization

Let $f_0, f_1, ..., f_k$ and $h_1, ..., h_s$ be functions defined on $\mathbb{R}^p$. We consider an optimization problem of the form

$$\begin{array}{ll} \underset{x}{\text{minimize}} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0,\ i = 1, \ldots, k \\ & h_i(x) = 0,\ i = 1, \ldots, s. \end{array} \tag{2.1}$$

We say that $x \in \mathbb{R}^p$ is the optimization variable and the function $f_0 : \mathbb{R}^p \to \mathbb{R}$ is the objective function or cost function. The inequalities $f_i(x) \leq 0$ are called inequality constraints, and the equations $h_i(x) = 0$ are called the equality constraints. If there are no constraints (i.e., $k = s = 0$), the problem (2.1) is said to be unconstrained.

Definition 2.1.1 *It is stated that b is feasible when it satisfies the constrains* $\forall_{i=1,...,k} f_i(x) \leq 0$ *and* $\forall_{i=1,...,s} h_i i(x) = 0$. *We say that b is strictly feasible, if it is feasible and additionally for* $i = 1, \ldots, k$ *it holds* $f_i(x) < 0$. *Vector* x^* *is solution to (3.3), if* x^* *is feasible and for all feasible* $x \in \mathbb{R}^p$ *we have* $f_0(x) \geq f_0(x^*)$. *The set of all solutions is denoted by* B^*. *The problem is said to have a unique solution, if* B^* *is a singleton.*

Definition 2.1.2 *Suppose that set B^* is nonempty. Then, for all vectors from B^*, the objective function takes the same value, which will be denoted by f_0^* and will be called the optimal value of f_0. If B^* is empty, then we define $f_0^* := -\infty$.*

Definition 2.1.3 *We call function $L : \mathbb{R}^p \times \mathbb{R}^k \times \mathbb{R}^s \to \mathbb{R}$ Lagrangian associated with the problem (2.1), if L is of form:*

$$L(x, \nu, \mu) = f_0(x) + \sum_{i=1}^{k} \nu_i f_i(x) + \sum_{i=1}^{s} \mu_i h_i(x) \tag{2.2}$$

Variables $\mu = (\mu_1, \ldots, \mu_k)$ and $\nu = (\nu_1, \ldots, \nu_k)$ are called dual variables associated with the problem (2.1). We say that μ_i is the Lagrange multiplier associated with the i^{th} equality constraint and ν_i is Lagrange multiplier associated with the i^{th} inequality constraint.

Exploiting symmetry, in Lagrangian we can swap primal and dual variables obtaining dual function and dual optimization problem.

Definition 2.1.4 *We define Lagrange dual function $g : \mathbb{R}^k \times \mathbb{R}^s \to \mathbb{R} \cup \{-\infty\}$ as*

$$g(\nu, \mu) = \inf_x L(x, \nu, \mu) \tag{2.3}$$

Definition 2.1.5 (Dual problem)
Taking function (2.3) as an objective, we define Lagrange dual problem as

$$\begin{aligned} &\underset{\nu,\mu}{maximize} && g(\nu, \mu) \\ &subject\ to && \nu \succeq 0. \end{aligned} \tag{2.4}$$

The pair (μ^*, ν^*), being solution to (2.4), is referred to as the dual solution. We will denote by f^* and g^* the optimal values for primal and dual problem respectively. It always occurs $g^* \leq f^*$, which is called weak duality.

Definition 2.1.6 (Strong duality)
Let f^* and g^* the optimal values for primal and dual problem respectively. Then value $f^* - g^*$ is known as the duality gap. We say that strong duality holds for optimization problem if duality gap is equal to zero.

2.1.2 Convexity

We say that the function $f : \mathbb{R}^p \to \mathbb{R}$ is convex if for every $x, \tilde{x} \in \mathbb{R}^p$ and every $\alpha \in [0, 1]$,

$$f(\alpha x + (1 - \alpha)\tilde{x}) \leq \alpha f(x) + (1 - \alpha) f(\tilde{x}) \tag{2.5}$$

We say that f is strictly convex if inequality 2.5 is strict for all $x \neq \tilde{x}$ and $\alpha \in (0, 1)$.

Lemma 2.1.7 (Operations preserving convexity) *The following comes from chapter 3 of from [Boyd and Vandenberghe, 2004]. Let f_1, f_2, f_m be convex functions. Then:*

- *non-negative weighted sum $f := w_1 f_1 + \cdots + w_k f_k$ is convex*

- *composition with an affine mapping. Suppose* $f : \mathbb{R}^n \to \mathbb{R}$. $\mathbf{A} \in \mathbb{R}^{n\times m}$, $b \in \mathbb{R}^n$ *and define* $g : \mathbb{R}^m \to \mathbb{R}$ *by* $g(x) = f(\mathbf{A}x + b)$. *Then* g *is convex*
- *pointwise maximum* $f(x) := \max_i f_i(x)$ *is convex*

Definition 2.1.8 *Convex optimization problem is an optimization problem (2.1) in which objective function and inequality constraint functions* $f_0, f_1, ..., f_k$ *are convex and* $h_1, ..., h_s$ *are affine.*

Proposition 2.1.9 *Suppose that the objective function in convex optimization problem is strictly convex. If a solution exists, then this solution is unique.*

Proof
Suppose this proposition is false.

Assume that the optimal set, $B*$, has more than one point and let $x*, \tilde{x*} \in B*$ be two distinct solutions. Then we have optimal value $f* = f(x*) = f(\tilde{x*})$. For any $\alpha_0 \in (0, 1)$, we construct $x_0 = \alpha x* + (1-\alpha)\tilde{x*}$. From strict convexity we get that $f(x_0) < \alpha f(x) + (1-\alpha)f(\tilde{x}) = \alpha f* + (1-\alpha)f* = f*$. Which contradicts that $x*, \tilde{x*} \in B*$. Thus the proposition is true. ∎

Definition 2.1.10 (Slater's condition)
We say that the optimization problem

$$\begin{array}{ll} \underset{x}{\text{minimize}} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0,\ i = 1, \ldots, k \\ & Ax = b,\ i = 1, \ldots, s. \end{array} \tag{2.6}$$

satisfies Slater's condition if there exists x_0 *that is strictly feasible i.e.*

$$f_i(x_0) < 0,\ i = 1, \ldots, k \text{ and } Ax_0 = b$$

The following theorem holds (see [Boyd and Vandenberghe, 2004]).

Theorem 2.1.11 (Slater's theorem) *If optimization problem is convex and it satisfies Slater's condition then strong duality holds.*

2.1.3 Proximal algorithm

Consider unconstrained optimization problem of the form

$$\underset{x}{\text{minimize}} \quad g(x) + h(x), \tag{2.7}$$

where g is convex and differentiable and h is convex. If there exists efficient algorithm for computing proximal operator function for h

$$prox_{th}(y) := \arg\min_x \left\{ \frac{1}{2t}\|y - x\|_2^2 \ + h(x) \right\} \tag{2.8}$$

for each $y \in \mathbb{R}^p$ and $t > 0$, then the problem (2.7) can be solved using the following proximal gradient algorithm.

Suppose that in k step $x^{(k)}$ is the current guess. Then, guess $x^{(k+1)}$ is given by

$$x^{(k+1)} = \arg\min_x \left\{ g(x^{(k)}) + \langle \nabla g(x^{(k)}), x - x^{(k)} \rangle + \frac{1}{2t}\|x - x^{(k)}\|_2^2 + h(x) \right\} \tag{2.9}$$

(2.9) resembles our initial objective as first two terms are Taylor expansion of g, while third one ensures that new estimate lies in proximity of the old one. It can be further reformulated to:

$$x^{(k+1)} = \arg\min_x \left\{ \frac{1}{2t}\|x^{(k)} - x - t\nabla g(x^{(k)})\|_2^2 + h(x) \right\}$$

Proof of convergence of algorithm 1 can be found in [Beck and Teboulle, 2009]. There are however methods, that can speed up proximal gradient method even further.

Algorithm 1 Proximal gradient algorithm

Set x_0. $k = 0$
while convergence criterion is not satisfied **do**
  $x^{k+1} := prox_{t_k h}(x_k - t_k \nabla g(x_k))$
end while

2.1.4 Fast proximal gradient method (FISTA)

Consider a convex optimization problem with composite objective

$$\underset{x}{\text{minimize}} \; g(x) + h(x) \tag{2.10}$$

where g is differentiable and h has inexpensive prox operator.

[Beck and Teboulle, 2009] noted that by choosing intermediate update point in algorithm 1 one can speed up convergence. They showed couple of examples including solving linear model with ℓ_1 penalty (LASSO).
The following algorithm can be used to solve (2.10):

Algorithm 2 General form of Accelerated proximal gradient algorithm

Choose $x_0 = y_0$. $\theta_0 = 1$
for $k = 0, 1, 2, \ldots$ **do**
  $x^{k+1} := prox_{th}(y_k - t_k \nabla g(y_k)$
  $\theta^{k+1} := \dfrac{1 + \sqrt{1 + 4/\theta_{k-1}^2}}{2}$
  $y^{k+1} := x^{k+1} + \theta_{k+1}[\frac{1}{\theta_k} - 1](x_{k+1} - x_k)$
end for

The proof of convergence of this algorithm can be found in [Beck and Teboulle, 2009]. As we do not use this method, only compare to implementation based on it, we omit exact formulation of the theorem.

2.1.5 Alternating direction method of multipliers (ADMM)

In this subsection we follow [Boyd et al., 2011].

Consider a problem with the separable objective

$$\begin{array}{ll} \underset{x}{\text{minimize}} & f(x) + g(z) \\ \text{subject to} & Ax + Bz = c \end{array} \tag{2.11}$$

where $x \in \mathbb{R}^n$ and $z \in \mathbb{R}^m$ are variables and $A \in \mathbb{R}^{p \times n}, B \in \mathbb{R}^{p \times m}, c \in \mathbb{R}^p$ are given.

For such a problem we form an augmented Lagrangian with the penalty parameter μ

$$\mathcal{L}_\mu(x, z, y) = f(x) + g(z) + \langle y, Ax + Bz - c \rangle + \frac{\mu}{2}\|Ax + Bz - c\|_F^2$$

Algorithm sequentially optimizes over x and z, thus name alternating direction, and dual variable y update using step size μ.

Algorithm 3 General form of ADMM algorithm

$y_0 \leftarrow \tilde{y}$. $z_0 \leftarrow \tilde{z}$, $k \leftarrow 1$, $\mu \leftarrow \mu_0 > 0$
while convergence criterion is not satisfied **do**
  $x^{k+1} := \arg\min_x \mathcal{L}(x, z^k, y^k)$
  $z^{k+1} := \arg\min_z \mathcal{L}(x^{k+1}, z, y^k)$
  $y^{k+1} := y^k + \mu(Ax^{k+1} + Bz^{k+1} - c)$
end while

Scaled form

The augmented Lagrangian can be rewritten by combining linear and quadratic terms. Let us define residual as $r = Ax + Bz - c$.

$$y^T r + \frac{\mu}{2}\|r\|_2^2 = \frac{\mu}{2}\|r + \frac{1}{\mu}y\|_2^2 - \frac{1}{2\mu}\|y\|^2 = \frac{\mu}{2}\|r + u\|_2^2 - \frac{\mu}{2}\|u\|$$

where $u = \frac{1}{\mu}y$ is scaled dual variable. Updates in ADMM take then the following form

$$\begin{aligned} x^{k+1} &:= \arg\min x f(x) + \frac{\mu}{2}\|Ax + Bz^k - c + u^k\|_2^2 \\ z^{k+1} &:= \arg\min z g(z) + \frac{\mu}{2}\|Ax^{k+1} + Bz - c + u^k\|_2^2 \\ u^{k+1} &:= u^k + Ax^{k+1} + Bz^{k+1} - c \end{aligned}$$

Convergence

In chapter 3 of [Boyd et al., 2011] conditions for convergence and suggested stopping criteria of the algorithm 3 are given.

Theorem 2.1.12 (Convergence for ADMM)
If

- *extended-real-valued function $f(\cdot)$ and $g(\cdot)$ are closed, proper and convex*

- *unaugmented Lagrangian L_0 has a saddle point, that is there exists, not necessarily unique, $(x*, z*, y*)$ such that for every (x, y, z) the following holds:*

$$\mathcal{L}(x*, z*, y) \leq \mathcal{L}(x*, z*, y*) \leq \mathcal{L}(x, z, y*)$$

then ADMM iterates converge in the following sense

- *Residual convergence, $r^k = x^k - y^k \to 0$ as $k \to \infty$. This condition means that iterates approach feasibility*
- *Objective convergence to optimal value $f(x^k) + g(z^k) \to p* = f(x*) + g(z*)$*
- *Dual variable convergence $y^k \to y*$. Here $y*$ is dual optimal point.*

[Boyd et al., 2011] also show optimality conditions from which they derive suggested stopping criterion.

Definition 2.1.13 (Stopping criterion for ADMM)
Let us denote $r^k = x^k - y^k$ as primal feasibility and $s^k = \mu A^T B(z^{k+1} - z^k)$ as dual feasibility. Then stopping criterion is

$$|r^k|_2 \leq \epsilon^{primal} \quad and \quad |s^k|_2 \leq \epsilon^{dual}$$

where $\epsilon^{primal} > 0$ and $\epsilon^{dual} > 0$ are feasibility tolerance for primal and dual feasibility.

2.2 Probability results

Theorem 2.2.1 (Law of iterated logarithm) *Let $\{Y_n\}$ be independent, identically distributed random variables with means zero and unit variances. Let $S_n = Y_1 + \cdots + Y_n$. Then almost surely:*

$$\limsup_{n\to\infty} \frac{S_n}{\sqrt{2n \log\log n}} = 1$$

Corollary 2.2.2 *Under assumptions of theorem 2.2.1*

$$\forall_{C>1} \text{ almost surely } \exists_{n_0} \forall_{n \geq n_0} \quad \frac{S_n}{n} \leq \frac{C\sqrt{2 \log\log n}}{\sqrt{n}}$$

There are multiple generalizations of the law of iterated logarithm, most of which are focused on the violation of independence. For us the following results is of more interest:

Theorem 2.2.3 (Theorem 7.2 from Petrov and Petrov [1995]) *Let $\{X_n\}_{n\geq 1}$ be independent random variables sequence with $\mathbb{E}X_n = 0$, $\mathbb{E}X_n^2 = \sigma_n^2 < \infty$. Let $B_n = \sum_{i=1}^n \sigma_i^2$, $S_n = \sum_{i=1}^n X_i$ and $\delta_n = \sup_x |\mathbb{P}(S_n < x\sqrt{B_n}) - \Phi(x)|$ where $\Phi(x)$ is a standard normal distribution function. If*

1. *$B_n \to \infty$, as $n \to \infty$*
2. *$\frac{B_{n+1}}{B_n} \to 1$ as $n \to \infty$*
3. *For some $\gamma > 0$ $\delta_n = O(\log {B_n}^{-1-\gamma})$*

hold, then

$$\limsup_{n\to\infty} \frac{S_n}{\sqrt{2B_n \log\log B_n}} = 1 \text{ a.s.}$$

∎

2.3 Variable selection

One of the crucial challenges in the modern data analysis is increasing dimensionality of the data. Its abundance causes computational and theoretical problems. Making statistical inference about thousands of variables requires careful handling because of the problem of multiple testing. A number of methods were introduced to deal with that and we shall give a brief overview of those.

2.3.1 Correction for multiple testing

In statistical testing a classic p-value thresholds are often used, most common one being 0.05. It means that test is allowed to reject null hypothesis 5 out of 100 times when it really holds. Note that when we have 20 statistical tests then the expected number of rejected tests, when all null hypothesis are true, is 1. So we expect to make a mistake.
In case of multiple hypothesis testing we use the following notation:

- m is the total number hypotheses tested
- m_0 is the number of true null hypotheses
- $m_1 = m - m_0$ is the number of true alternative hypotheses
- V is the number of false positives (Type I error) (also called "false discoveries")
- S is the number of true positives (also called "true discoveries")
- T is the number of false negatives (Type II error)
- U is the number of true negatives
- $R = V + S$ is the number of rejected null hypotheses (also called "discoveries", either true or false)

Interaction between above quantities are visualized in table 2.1.

Table 2.1: Multiple hypothesis testing summary

	Declared significant	Declared non-significant	Total
True null hypothesis	V	U	m_0
False null hypothesis	S	T	$m_1 = m - m_0$
Total	R	$m - R$	m

There are multiple ways to measure the error made by multiple testing procedures. We shall focus on three of them. Before that we introduce one function that will make future notation easier.

Definition 2.3.1 (False discovery proportion) *False discovery proportion (FDP, also denoted by Q) in the problem of multiple hypothesis testing is the ratio of false discoveries to all discoveries*

$$Q = \begin{cases} \frac{V}{R} & \text{if } R > 0 \\ 0 & \text{if } R = 0 \end{cases}$$

First measure is a conservative one. It refers to the probability of making at least one mistake of false discovery.

Definition 2.3.2 (Familiy wise error rate) *Familiy wise error rate in the problem of multiple hypothesis testing is the probability of making one or more false discoveries*

$$\mathbb{P}(V > 0)$$

Second is a weaker one. We require FWER to be controlled *only* under global null, that is all null hypothesis are true.

Definition 2.3.3 (Weak FWER) *Weak Familiy wise error rate in the problem of multiple hypothesis testing is the probability of making one or more false discoveries when all null hypothesis are true.*

$$\mathbb{P}(V > 0 | m_1 = 0)$$

Third was introduced in the seminal paper in 1995 by Benjamini and Hochberg [1995]. Rather than concentrating on zero-one thinking (making - not making a false discovery), it looks at the fraction on false discoveries to all discoveries made by a procedure.

Definition 2.3.4 (False discovery rate) *False discovery rate in the problem of multiple hypothesis testing is the expected ratio of false discoveries to all discoveries*

$$FDR = \mathbb{E}(Q)$$

When we perform multiple testing with hundreds of thousands of tests at once we expect thousands of them to be rejected. Such problem is common in the field of genetics, i.e. when marginal tests are performed on phenotype vs one SNP. In the reminder of this thesis we shall refer to making a type 1 error as a false discovery. The question is can we limit number of false discoveries. A widely used correction that guarantees FWER control was proposed by Bonferroni [1936].

Definition 2.3.5 (Bonferroni correction for multiple testing)
Let $H_1, \ldots, H_m$ and $p_1, \ldots, p_m$ corresponding p-values. Bonferroni correction rejects individual hypothesis if their p values are smaller than $p_i \leq \frac{\alpha}{m}$

Lemma 2.3.6 *Bonferroni correction 2.3.5 controls FWER at level α*

Proof

$$\text{FWER} = \mathbb{P}\left\{\cup_{i=1}^{m_0}\left(p_i \leq \frac{\alpha}{m}\right)\right\} \leq \sum^{m_0} \mathbb{P}\left(p_i \leq \frac{\alpha}{m}\right) = m_0 \frac{\alpha}{m} \leq \alpha$$

∎

There is less conservative procedure that holds, like Bonferroni correction, without any additional assumptions.

Definition 2.3.7 (Holm-Bonferroni correction for multiple testing)
Let $H_1, \ldots, H_m$ and $p_1, \ldots, p_m$ corresponding p-values. We sort p values getting $p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(m)}$. Let k be the minimal index for which $p_{(k)} > \frac{\alpha}{m+1-k}$. Then we reject hypothesis $H_{(1)} \ldots H_{(k-1)}$ and do not reject $H_{(k)} \ldots H_{(m)}$

In paper Holm [1979] it is proved that:

Lemma 2.3.8 *Holm-Bonferroni correction 2.3.7 controls FWER at level α.*

Note, that Holm's procedure does not compare all sorted p-values to adjusted thresholds. It starts with first, smallest p-value and increases index step by step until it finds index that violates the inequality. Contrary, one might find the first index that satisfies inequality starting from largest p-value.

Definition 2.3.9 (Hochberg step-up correction for multiple testing)
Let $H_1, \ldots, H_m$ and $p_1, \ldots, p_m$ corresponding p-values. We sort p values getting $p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(m)}$. Let k be the first index in order $m, m-1, \ldots$ for which $p_{(k)} \leq \frac{\alpha}{m+1-k}$. Then we reject hypothesis $H_{(1)} \ldots H_{(k)}$ and do not reject $H_{(k+1)} \ldots H_{(m)}$

The following lemma holds for procedure 2.3.9 (for proof see [Hochberg, 1988]).

Lemma 2.3.10 *If test statistics $H_1, \ldots, H_m$ are independent then Hochberg step-up procedure controls FWER.*

This results can be further generalized for various types of positive dependency (see [Sarkar, 1998]).
We mention one more correction for multiple testing that, under certain assumptions, controls FDR at a given level.

Definition 2.3.11 (Benjamini-Hochberg correction for multiple testing)
Let $H_1, \ldots, H_m$ and $p_1, \ldots, p_m$ corresponding p-values. We sort p values getting $p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(m)}$. Let k be the largest index k for which $p_{(k)} \leq \frac{k}{m}\alpha$. Then we reject hypothesis $H_{(1)} \ldots H_{(k)}$ and do not reject $H_{(k+1)} \ldots H_{(m)}$

Theorem 2.3.12 (FDR control for independent test statistics) *Benjamini and Hochberg [1995] proved that under assumption that test statistics $H_1, \ldots, H_m$ are independent, correction 2.3.11 controls FDR.*

Recall that a set D is called increasing if $x \in D$ and $y \geq x$, imply that $y \in D$.

Definition 2.3.13 (Positive regression dependency on each one from a subset (PRDS))
We say that random vector X has property PRDS on indices subset I_0 if For any increasing set D, and for each $i \in I_0$

$$\mathbb{P}(X \in D | X_i = x) \quad \text{is nondecreasing in } x.$$

Theorem 2.3.14 *If the joint distribution of the test statistics is PRDS on the subset of test statistics corresponding to true null hypotheses, the Benjamini Hochberg procedure controls the FDR at level less than or equal to $\frac{m_0}{m}\alpha$.*

Proof can be found in [Benjamini and Yekutieli, 2001].

2.3.2 Maximum likelihood

One of the classical statistical approaches is maximum likelihood. Assuming that data is drawn from the model M_Θ of unknown parameters Θ according to density function

$$\mathbf{y} = f_{M_\Theta}(\mathbf{X}) \tag{2.12}$$

we estimate it, by maximizing density function over Θ, and which is then called likelihood function

$$L_{\mathbf{X}}(\Theta) = f_{M_\Theta}(\mathbf{X}) \tag{2.13}$$

We say that $\hat{\Theta}$ is maximum likelihood estimator if it maximizes

$$\hat{\Theta} = \arg\max_{\Theta} L_{\mathbf{X}}(\Theta)$$

Maximum likelihood estimators are well described in the literature [van der Vaart, 2000]. They have a lot of interesting properties. First of all they are consistent, meaning that if data is in fact drawn from the assumed distribution with parameter Θ_0 then, under mild regularity conditions (see [van der Vaart, 2000])

$$\hat{\Theta} \underset{n\to\infty}{\to} \Theta_0$$

Moreover, under mild regularity conditions the asymptotic distribution of $\hat{\Theta}$ is normal with covariance matrix being inverse of Fisher Information Matrix.
Despite all these properties and features that made ML very useful technique in classical statistics, it proves to be inadequate to handle modern data analysis challenges. Specifically, ML are not unique when the number of parameters p is larger than the number of observations n. Also, in case when p is smaller but comparable to n, the variance of ML tends to be very large, which leads to very poor behavior of related testing procedures for significance of individual parameters in the model.
The tools of modern statistics modify MLE by including the penalty function, which stabilizes the variance of resulting estimators. Particularly interesting are penalties which automatically lead to eliminating some of the model parameters (i.e. make them equal to zero) and thus lead to sparse models. Observe that since maximum likelihood estimators are normally distributed, with probability 1 estimators are non-zero. If one builds a model based on thousands of variables, then it is unreasonable to assume that all of them give significant input. Therefore a lot of them should not be included in the model. In [Hastie et al., 2009] a term *bet on sparsity* was coined. It basically says, that when dealing with high dimensional data on of the following is true: either only a handful of variables are significant (therefore, solution is sparse) or solution is not sparse in which case, we any kind of modeling is doomed to fail.

2.3.3 Bias variance trade-off

One of the most important concepts in the modern statistics is bias variance trade-off.

Assume that we want to predict $\mathbf{y}$ based on $\mathbf{X}$ and that

$$\mathbf{y} = f(\mathbf{X}) + \epsilon \tag{2.14}$$

where f is some function and ϵ is independent normally distributed irreducible error. Let us denote an estimator of $f(\mathbf{X})$ by $f(\hat{\mathbf{X}})$. Then expected squared prediction error can be decomposed into:

$$\begin{aligned}\mathbb{E}[(\mathbf{y}-f(\hat{\mathbf{X}}))^2] &= \mathbb{E}[(f(\mathbf{X})+\epsilon-f(\hat{\mathbf{X}}))^2] \\ &= \mathbb{E}[\Big((f(\mathbf{X})-\mathbb{E}f(\hat{\mathbf{X}}))+(\mathbb{E}f(\hat{\mathbf{X}})-f(\hat{\mathbf{X}}))+\epsilon)\Big)^2] \\ &\overset{\text{independace and orthogonality}}{=} \mathbb{E}[(f(\mathbf{X})-\mathbb{E}f(\hat{\mathbf{X}}))^2]+\mathbb{E}[(\mathbb{E}(f(\hat{\mathbf{X}})-f(\hat{\mathbf{X}}))^2]+\mathbb{E}\epsilon^2 \\ &= \underbrace{f(\mathbf{X})-\mathbb{E}f(\hat{\mathbf{X}}))^2}_{\text{Squared bias}}+\underbrace{\mathbb{E}[(\mathbb{E}(f(\hat{\mathbf{X}})-f(\hat{\mathbf{X}}))^2]}_{\text{Estimator variance}}+\underbrace{\mathbb{E}\epsilon^2}_{\text{Irreducible Error}} \qquad (2.15)\end{aligned}$$

In the context of linear regression. The Gauss-Markov estimator minimizes prediction error among all the estimator that are unbiased. However, one might define various biased estimators with much smaller variance, and therefore with lower total expected prediction error. One of the ways of introducing bias is introducing additional penalty on the level of complication of function f. This technique is called regularization, and we shall cover it in the following subsection.

2.3.4 Laplace Method for Integrals

In this section we shall cover in detail so called Laplace approximation. First we cover simplified version, then full scale and we end with multivariate version. We follow [de Bruijn, 1970].

Theorem 2.3.15 *Let us consider integral*

$$F(t)=\int_{-\infty}^{\infty}e^{th(x)}dx$$

with h and F satisfying following assumptions

(a) $h(x)$ has global maximum in 0 with $h(0)=0$

(b) $\exists_{b,c>0}$ such that $h(x)\leq -b$ for $|x|>c$

(c) $F(t)$ should converge for some value of t. For simplicity let us assume that $\int e^{h(x)}dx<\infty$

(d) $h'(x)$ and $h''(x)$ exist in some neighborhood of 0, and $h''(x)<0$ in that neighborhood

Then

$$\frac{F(t)}{\sqrt{2\pi}(-th''(0))^{-1/2}}\underset{t\to\infty}{\to}1 \qquad (2.16)$$

Proof
First let us observe that from (a) and (d) we have that $h'(0)=0$. Furthermore we can actually omit most of our integration range when t is large.

$$\forall\delta>0\ \ \exists\nu(\delta)>0 \text{ such that } h(x)\leq-\nu(\delta) \text{ for } x\in\mathbb{R}\setminus(-\delta,\delta) \qquad (2.17)$$

For $\delta > c$ statement (2.17) is true because of (b). For $\delta < c$ observe that because h is continuous on (δ, c) it achieves its maximum and this maximum obviously equals less than 0.

$$\int_\delta^\infty e^{th(x)}dx = \int_\delta^\infty e^{(t-1)h(x)}e^{h(x)}dx < e^{-(t-1)\nu(\delta)}\int_\delta^\infty e^{h(x)}dx$$

Therefore

$$\int_{\mathbb{R}\setminus(-\delta,\delta)} e^{th(x)}dx < 2e^{-(t-1)\nu(\delta)}\int_{-\infty}^\infty e^{h(x)}dx \tag{2.18}$$

So $\int_{\mathbb{R}\setminus(-\delta,\delta)} e^{th(x)}dx = O(e^{-t\alpha})$ where $\alpha = \nu(\delta)$ depends on δ, but not on t.
Next we shall estimate the remaining integral.
Consider $\phi(x) = h(x) - \frac{1}{2}x^2h''(0)$. Then $\phi(0) = \phi'(0) = \phi''(0) = 0$. Because of that $\frac{\phi'(x)-\phi'(0)}{x} \xrightarrow{x\to 0} 0$. So $\phi'(x) = o(x)$ $(x \to 0)$. From mean value theorem we get $\phi(x) - \phi(0) = x\phi'(\theta x)$ for some $0 < \theta < 1$. Therefore

$$\phi(x) = xo(\theta x) = o(x^2)\ (x \to 0) \tag{2.19}$$

Therefore, for a given ϵ such that $0 < 3\epsilon < |h''(0)|$, we can determine $\delta > 0$ such that

$$|h(x) - \frac{1}{2}x^2h''(0)| \le \epsilon x^2\ ,\ \ (-\delta \le x \le \delta) \tag{2.20}$$

Consequently, we can bound integral from below and from above

$$\int_{-\delta}^\delta e^{1/2tx^2(h''(0)-2\epsilon)}dx < \int_{-\delta}^\delta e^{th(x)}dx < \int_{-\delta}^\delta e^{1/2tx^2(h''(0)+2\epsilon)}dx \tag{2.21}$$

All three above integrals differ from corresponding integrals over all axis by an amount $O(e^{-t\alpha})$. For the central we get this from (2.18). For the left-hand side observe that:

$$\begin{aligned}\int_\delta^\infty e^{1/2tx^2(h''(0)-2\epsilon)}dx &= \int_\delta^\infty e^{(t-1)1/2x^2(h''(0)-2\epsilon)}e^{1/2x^2(h''(0)-2\epsilon)}dx \\ &< e^{-(t-1)1/2\delta^2(h''(0)-2\epsilon)}\int_\delta^\infty e^{1/2x^2(h''(0)-2\epsilon)}dx = O(e^{-t\alpha})\end{aligned}$$

The rightmost integral is then simple gamma function or if you prefer normal pdf without scaling factor. Thus combining everything we got so far,

$$\begin{aligned}\int_{-\infty}^\infty e^{th(x)}dx &< \sqrt{\pi}(-\frac{1}{2}t(h''(0)+2\epsilon))^{-1/2} + O(e^{-t\alpha}) \\ &< \sqrt{2\pi}(-h''(0)-2\epsilon)^{-1/2}t^{-1/2} + O(e^{-t\alpha}) \\ &< \sqrt{2\pi}(-h''(0)-3\epsilon)^{-1/2}t^{-1/2}\end{aligned} \tag{2.22}$$

The last inequality holds for sufficiently large t. Because ϵ can be set arbitrarily small, and because we can similarly bound integral from below, we get

$$\frac{\int_{-\infty}^\infty e^{th(x)}dx}{\sqrt{2\pi}(-th''(0))^{-1/2}} \underset{t\to\infty}{\to} 1 \tag{2.23}$$

∎

Corollary 2.3.16
We've assumed that global maximum of function h is in 0 with $h(0) = 0$. We can easily extend theorem 2.3.15. Suppose we have function $g(x)$ with maximum in $\hat{x}$. and $g''(x) < 0$ in neighborhood of $\hat{x}$ Then by substitution $h(x - \hat{x}) = g(x) - g(\hat{x})$ we get:

$$\begin{aligned}\int_{-\infty}^{\infty} e^{tg(x)}dx &= \int_{-\infty}^{\infty} e^{t(g(\hat{x})+h(x-\hat{x}))}dx = \int_{-\infty}^{\infty} e^{tg(\hat{x})} \int_{-\infty}^{\infty} e^{th(x-\hat{x})}dx \\ &\approx e^{tg(\hat{x})}\sqrt{2\pi}(-tg''(\hat{x}))^{-1/2}\end{aligned}$$

Integral of a posteriori

Laplace approximation can be used to compute posterior probabilities (see [Ghosh et al., 2007].

Theorem 2.3.17 *Let*

$$F(t) = \int_{-\infty}^{\infty} g(x)e^{th(x)}dx$$

Under assumptions from theorem 2.3.15 and

- *g is twice differentiable around 0*
- *g is bounded $\sup_{\mathbb{R}} g(x) = G < \infty$*
- *g is nonzero in 0 $g(0) > 0$*

the following holds:

$$\frac{F(t)}{\sqrt{2\pi}g(0)(-h''(0))^{-1/2}t^{-1/2}} \underset{t\to\infty}{\to} 1 \tag{2.24}$$

Proof
The bound is probably not required here. What we really want is, as in (2.18)

$$\int_{\mathbb{R}\setminus(-\delta,\delta)} g(x)e^{th(x)}dx = O(e^{-t\nu(\delta)})$$

We follow proof of theorem 2.3.15.

$$\int_{\delta}^{\infty} g(x)e^{th(x)}dx = \int_{\delta}^{\infty} e^{(t-1)h(x)}g(x)e^{h(x)}dx < e^{-(t-1)\nu(\delta)}G\int_{\delta}^{\infty} e^{h(x)}dx \tag{2.25}$$

For the second part of integral we use Taylor series of order two $g(x) = g(0) + xg'(0) + \frac{x^2}{2}g''(0) + o(x^2)$

$$\int_{-\delta}^{\delta} g(x)e^{th(x)}dx < \int_{-\delta}^{\delta} \left(g(0) + xg'(0) + \frac{x^2}{2}g''(0) + o(x^2)\right) e^{1/2tx^2(h''(0)-2\epsilon)}dx$$

We can choose ϵ so that the following holds:

$$\begin{aligned}\int_{-\delta}^{\delta} &\left(g(0) + xg'(0) + \frac{x^2}{2}g''(0) + o(x^2)\right) e^{1/2tx^2(h''(0)-2\epsilon)}dx \\ &< \int_{-\delta}^{\delta} \left(g(0) + xg'(0) + x^2(\frac{g''(0)}{2} + \epsilon)\right) e^{1/2tx^2(h''(0)-2\epsilon)}dx \end{aligned} \tag{2.26}$$

We can split the above into three integrals
$\int_{-\delta}^{\delta} xg'(0)e^{1/2tx^2(h''(0)-2\epsilon)}dx = 0$ because of symmetry. To get the other two integrals, we need to use substitution, that yields gamma integrals.

$$\int_{-\delta}^{\delta} x^2(\frac{g''(0)}{2}+\epsilon)e^{\frac{1}{2}tx^2(h''(0)-2\epsilon)}dx = (\frac{g''(0)}{2}+\epsilon)\int_{-\delta}^{\delta} x^2 e^{\frac{1}{2}tx^2(h''(0)-2\epsilon)}$$

Here we can simply used the properties of normal pdf again.

$$\int_{-\delta}^{\delta} x^2 e^{\frac{1}{2}tx^2(h''(0)-2\epsilon)} = \sqrt{(}2\pi)\sigma\cdot\sigma^2 = \sqrt{(}2\pi)(-\frac{1}{2}t(h''(0)-2\epsilon))^{-3/2}$$

Alternative is using substitution $y = -\frac{1}{2}t(h''(0)-2\epsilon)x^2$. So $dy = -t(h''(0)-2\epsilon)xdx$ and

$$\frac{dy}{-t(h''(0)-2\epsilon)}\sqrt{\frac{-t(h''(0)-2\epsilon)}{y}} = dx$$

$$\begin{aligned}
&(\frac{g''(0)}{2}+\epsilon)\int_{-\delta}^{\delta} x^2 e^{\frac{1}{2}tx^2(h''(0)-2\epsilon)} \\
&= (\frac{g''(0)}{2}+\epsilon)\int_0^{\delta^2 t(-h''(0)+2\epsilon)} \frac{y^2}{-\frac{1}{2}t(h''(0)-2\epsilon)}e^y\sqrt{\frac{-\frac{1}{2}t(h''(0)-2\epsilon)}{y}}\frac{1}{-\frac{1}{2}t(h''(0)-2\epsilon)}dy \\
&= t^{-3/2}(\frac{g''(0)}{2}+\epsilon)(-\frac{1}{2}(h''(0)-2\epsilon))^{-3/2}\int_0^{\delta^2 t(-h''(0)+2\epsilon)} y^{3/2}e^y dy \\
&< t^{-5/2}(\frac{g''(0)}{2}+\epsilon)(-\frac{1}{2}(h''(0)-2\epsilon))^{-5/2}\int_0^{\infty} y^{3/2}e^y dy \\
&= t^{-3/2}(\frac{g''(0)}{2}+\epsilon)(-\frac{1}{2}(h''(0)-2\epsilon))^{-3/2}\Gamma(\frac{3}{2}) \\
&= O(t^{-\frac{3}{2}})
\end{aligned}$$

Combing this with results from section 2.3.4 we get:

$$\begin{aligned}
\int_{-\infty}^{\infty} g(x)e^{th(x)}dx &= \int_{-\infty}^{\delta} g(x)e^{th(x)}dx + \int_{\delta}^{\infty} g(x)e^{th(x)}dx + \int_{-\delta}^{\delta} g(x)e^{th(x)}dx \\
&= e^{-(t-1)\nu(\delta)}G\int_{\delta}^{\infty} e^{h(x)}dx + e^{-(t-1)\nu(\delta)}G\int_{-\infty}^{\delta} e^{h(x)}dx + \int_{-\delta}^{\delta} g(x)e^{th(x)}dx \\
&< e^{-(t-1)\nu(\delta)}G\int_{-\infty}^{\infty} e^{h(x)}dx + \int_{-\delta}^{\delta} g(x)e^{th(x)}dx \\
&= O(e^{-t\alpha}) + \int_{-\delta}^{\delta} g(x)e^{th(x)}dx \\
&= \sqrt{2\pi}g(0)(-h''(0)-2\epsilon)^{-1/2}t^{-1/2} + O(e^{-t\alpha}) + O(t^{-\frac{3}{2}})
\end{aligned}$$

For sufficiently large t, because ϵ was chosen arbitrary

$$\frac{\int_{-\infty}^{\infty} g(x)e^{th(x)}dx}{\sqrt{2\pi}g(0)(-h''(0))^{-1/2}t^{-1/2}} \to 1$$

∎

Multivariate Laplace approximation

As in every source it is said that multivariate case is *the same as univariate with obvious alterations* we aim to write it down in rigorous way.

Theorem 2.3.18 *Let us define a function*

$$F(t) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} e^{th(x_1,\ldots,x_n)} dx_1 \ldots dx_n \tag{2.27}$$

$$= \int_{\mathbb{R}^n} e^{th(\mathbf{x})} d\mathbf{x} \tag{2.28}$$

We assume the following:

(a) $h(\mathbf{x}) = h(x_1, \ldots, h_n)$ *has global, unique maximum in 0, with* $h(0, \ldots, 0) = 0$

(b) $\frac{\partial h(\mathbf{x})}{\partial x_i}$ *and* $\frac{\partial^2 h(\mathbf{x})}{\partial x_i \partial x_j}$ *exist and are continuous in some neighborhood of 0*

(c) $F(t)$ *should converge for some value of t. For simplicity let us assume that it holds for* $t = 1$*:* $\int e^{h(\mathbf{x})} dx < \infty$

(d) $h(x_1, \ldots, h_n) = -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{i,j} x_i x_j + o(x_1^2 + \cdots + x_n^2)$ *when* $x_1^2 + \cdots + x_n^2 \to 0$

(e) We require $A = (a_{i,j})_{i,j=1}^{n}$ *to be positive definite*

(f) $\exists_{b,c>0}$ *such that* $h(x) \leq -b$ *for* $\|x\| > c$

Then

$$\frac{F(t)}{\sqrt{2\pi}^{\frac{k}{2}} t^{-\frac{k}{2}} |A|^{-1/2}} \underset{t\to\infty}{\to} 1 \tag{2.29}$$

Proof First observe that because h has unique maximum we can separate h from 0 outside any arbitrary neighbourhood of 0.

$$\forall \delta > 0 \;\; \exists \nu(\delta) \text{ such that } h(\mathbf{x}) \leq -\nu(\delta) \text{ for } x \in \mathbb{R}^n \setminus \mathcal{B}(\delta) \tag{2.30}$$

By $\mathcal{B}(\delta)$ we mean an open ball centered in the origin with radius of δ. Argument is the same as in proof of theorem 2.3.15 and uses ((f)).
Therefore, using ((c))

$$\begin{aligned} \int_{\mathbb{R}^n \setminus \mathcal{B}(\delta)} e^{th(\mathbf{x})} dx &= \int_{\mathbb{R}^n \setminus \mathcal{B}(\delta)} e^{(t-1)h(\mathbf{x})} e^{h(\mathbf{x})} dx \\ &< \int_{\mathbb{R}^n \setminus \mathcal{B}(\delta)} e^{-(t-1)\nu(\delta)} e^{h(\mathbf{x})} dx \\ &< e^{-(t-1)\nu(\delta)} \int_{\mathbb{R}^n} e^{h(x)} dx = O(e^{-t\alpha}) \end{aligned} \tag{2.31}$$

From (e) all minors are positive, so we can choose ϵ such that $3\epsilon < \min_i a_{ii}$, and subsequently we have that, for $x \in \mathcal{B}(\delta)$, the following holds:

$$|h(\mathbf{x}) + \frac{1}{2} \mathbf{x}^T A \mathbf{x}| \leq \epsilon \|\mathbf{x}\|^2 \,, \quad x \in \mathcal{B}(\delta) \tag{2.32}$$

Subsequently, for any $\epsilon > 0$, we get

$$\int_{\mathcal{B}(\delta)} e^{-\frac{1}{2}t\cdot\mathbf{x}^T(A+\epsilon\mathbf{I})\mathbf{x}}d\mathbf{x} < \int_{\mathcal{B}(\delta)} e^{th(\mathbf{x})}d\mathbf{x} < \int_{\mathcal{B}(\delta)} e^{-\frac{1}{2}t\cdot(\mathbf{x}^TA\mathbf{x}-\epsilon\mathbf{x}^T\mathbf{x})}d\mathbf{x} < \int_{\mathcal{B}(\delta)} e^{-\frac{1}{2}t\cdot\mathbf{x}^T(A-\epsilon\mathbf{I})\mathbf{x}}d\mathbf{x} \tag{2.33}$$

To compute this integral observe that it is, up to scaling, probability density function of multivariate normal distribution $(2\pi)^{\frac{k}{2}}|\Sigma|^{-\frac{1}{2}}e^{-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)}$. Using (2.31) we get

$$\int_{\mathbb{R}^n} e^{\frac{1}{2}t\cdot\mathbf{x}^T(A-\epsilon\mathbf{I})\mathbf{x}}d\mathbf{x} = (2\pi)^{\frac{n}{2}}|t(A-\epsilon\mathbf{I})|^{-1/2} + O(e^{-t\alpha}) = (2\pi)^{\frac{n}{2}}t^{-\frac{n}{2}}|A-\epsilon\mathbf{I}|^{-1/2} + O(e^{-t\alpha}) \tag{2.34}$$

Thanks to the fact that k does not depend on t and that ϵ is arbitrarily small, for large t:

$$\frac{\int_{\mathbb{R}^n} e^{th(\mathbf{x})}d\mathbf{x}}{(2\pi)^{\frac{n}{2}}t^{-\frac{n}{2}}|A|^{-1/2}} \to 1 \tag{2.35}$$

Similarly one can get the same asymptotic lower bound. ∎

2.3.5 Regularization

For simplicity in the remain of this chapter we shall consider the linear model. Please note that this methodology is not limited to it, but it is much easier to follow.

$$\mathbf{y} = \mathbf{X}\beta + \epsilon \tag{2.36}$$

where $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ is a vector of observations, $\mathbf{X} \in M_{n\times p}$ is known design matrix, β is unknown vector of parameters that we wish to estimate and ϵ is vector of errors. We assume errors independent, normally distributed $\epsilon_i \sim N(0, \sigma^2)$.

Observe that log-likelihood function for model (2.36) is the following

$$-\frac{n}{2}\log(2\pi) + \frac{1}{2}\log\sigma^2 + \frac{(y-\mathbf{X}\beta)^T(y-\mathbf{X}\beta)}{2\sigma^2}$$

Which brings the following MLE by computing derivative and setting it to zero.

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^Ty$$
$$\hat{\sigma^2} = \frac{1}{n}(y-\mathbf{X}\beta)^T(y-\mathbf{X}\beta)$$

As said before, MLE for this problem has good statistical properties as n tends to infinity. However things get ugly when $n \approx p$. When $p > n$ then there are infinitely many solutions to (2.36), as it involves computing pseudo-inverse of non-invertible matrix $\mathbf{X}^T\mathbf{X}$.

In their paper [Hoerl and Kennard, 1970] noticed that it suffices to add a small number to the diagonal of $\mathbf{X}^T\mathbf{X}$ to make it invertible and thus having unique and stable solution. Interestingly this corresponds to two different formulation of the problem. One from optimization perspective and the other one Bayesian.

One can add additional term to the log-likelihood and minimize so called penalized log-likelihood. We shall omit here σ^2 term, because optimization obviously can be split into two independent parts. So for the sake of simplicity, we assume that σ^2 is fixed.

$$\hat{\beta}_{\text{rigde}} = \arg\min_{\beta} \underbrace{\frac{(y - \mathbf{X}\beta)^T(y - \mathbf{X}\beta)}{2\sigma^2}}_{\text{loglikelihood}} + \underbrace{\lambda\beta^T\beta}_{\text{penalty}} \tag{2.37}$$

Which is basically a quadratic optimization problem with closed solution.

$$\hat{\beta}_{\text{rigde}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T y \tag{2.38}$$

This is equivalent to constrained formulation

$$\begin{aligned} \underset{x,z}{\text{minimize}} \quad & \frac{(y - \mathbf{X}\beta)^T(y - \mathbf{X}\beta)}{2\sigma^2} \\ \text{subject to} \quad & \beta^T\beta \leq t \end{aligned} \tag{2.39}$$

For some value t. There is a direct correspondence between λ and t, but not in a closed form.

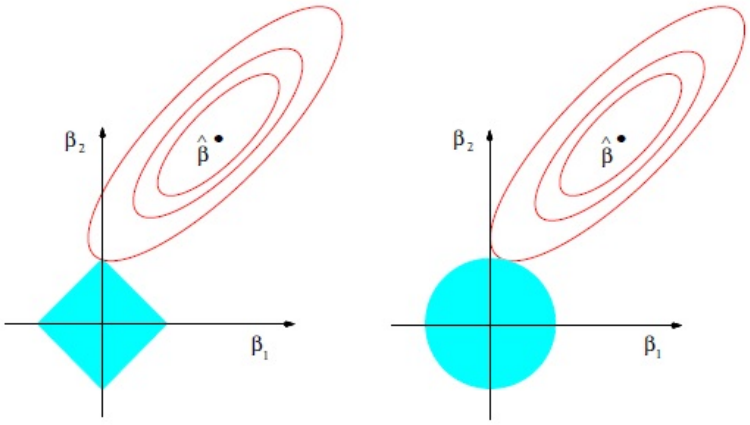


FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions $|\beta_1| + |\beta_2| \leq t$ and $\beta_1^2 + \beta_2^2 \leq t^2$, respectively, while the red ellipses are the contours of the least squares error function.*

Figure 2.1: Visualization of difference between ℓ_1 and ℓ_2 penalties. From book [Hastie et al., 2009]

Observe that ridge regression is a biased method for optimization. However, its variance is smaller than the one of MLE, and thus, for some set of values of the tuning parameter, its mean squared error is smaller. There are still some limitations of ridge. Figure 2.1 is probably a single most influential visualization in modern statistics. It gives a good intuition, that all elements of ridge solution will be nonzero with probability 1. So even if one includes dummy variable into the model, it will still be part of a model (though likely with small coefficient). In the next subsections we shall look into the methods that overcome this limitations.

2.3.6 Akaike Information Criterion (AIC)

An alternative approach, Akaike Information Criterion, was proposed by [Akaike, 1975]. To the likelihood function the following term was added

$$\hat{\beta}_{\text{AIC}} = \arg\min_{\beta} \underbrace{\frac{(y - \mathbf{X}\beta)^T(y - \mathbf{X}\beta)}{2\sigma^2}}_{\text{loglikelihood}} + \underbrace{\|\beta\|_0}_{\text{penalty}} \tag{2.40}$$

Observe that because penalty term is l_0 norm, in optimal β some entries will be 0 i.e. they will not be included in the model. AIC asymptotically with n minimizes Kullback–Leibler divergence between the true distribution generating y and its estimator [Akaike, 1975]. So in some way it is optimized with respect to the prediction error.

2.3.7 Bayesian Information Criterion (BIC)

One obvious modification of AIC is to alter penalty term. One such approach is famous Bayesian Information Criterion (BIC) proposed in 70ties by Schwarz [1978].

$$\hat{\beta}_{\text{BIC}} = \arg\min_{\beta} \underbrace{\frac{(y - \mathbf{X}\beta)^T(y - \mathbf{X}\beta)}{2\sigma^2}}_{\text{loglikelihood}} + \underbrace{\frac{\ln(n)}{2}\|\beta\|_0}_{\text{penalty}} \tag{2.41}$$

This formula is in fact a result of rigorous Bayesian reasoning and we shall cover it in details. Say we have a number of competing models $\{M_j\}_{j=1,\ldots,}$ and data $Y = (Y_1, \ldots, Y_n)$. Then how probable is the model given the data? We follow Bayes rule:

$$P(M_j|Y) \propto P(Y|M_j)P(M_j),$$

Where $P(M_j)$ is prior probability for the model. We select model that maximizes the above. Equivalently we may maximizes logs

$$\log P(M_j|Y) \propto logP(Y|M_j) + \log P(M_j),$$

The first term on the right hand side can be further expanded

$$P(Y|M_j) = \int_{\Theta_j} p(Y|M_j, \theta_j)\pi_j(\theta_j)d\theta_j = \int_{\Theta_j} L(\theta_j)\pi_j(\theta_j)d\theta_j,$$

where Θ_j is parameter space for model M_j, and L is likelihood function. For example, in normal distribution $\Theta = (\sigma^2, \mu) = ((0, \infty) \times \mathbb{R})$
Unfortunately in many cases that last integral is too difficult to compute, so we are left with approximation. BIC is result of using Laplace approximation (see section 2.3.4), in which the whole integral is approximated by integration in the neighborhood of the maximum likelihood estimator $\hat{\theta}$.

$$\log \int_{\Theta_j} L(\theta_j)\pi_j(\theta_j)d\theta_j \approx l(\hat{\theta_j}) - \frac{d_j}{2}\log n,$$

where $l = \log L$, d_j is the dimension of parameter space of M_j.

Finally we pick model that maximizes

$$\arg\max_j \; l(\hat{\theta_j}) - \frac{d_j}{2}\log n + \log P(M_j) \tag{2.42}$$

Usually uniform prior is assumed for all the models and value of BIC criterion for model M_j is given by::

$$BIC_{M_j} := l(\hat{\theta}_j) - \frac{d_j}{2} \log n$$

2.3.8 modified BIC

While both AIC and BIC work well when number of competing models is small, they have however a major drawback: there is no closed form for solution for it. This is a big difference compared to the case of ridge regression. Even worse, there is no tractable algorithm for finding solution as it, in theory, requires fitting all the competing models. In our regressions example, this means all possible variants of zero entries of β, which grows exponentially with the number of parameters 2^p. In practice greedy algorithms are used to get approximately optimal model. One starts with either empty or full model, and at each step we add or remove variable, such that the improvement in AIC/BIC is maximized. The procedure stops if no improvement of the criterion can be obtained by adding/removing a single variable. This strategy is sometimes extended a bit not to fall into local minimum, but in general, one follows this greedy approach.

The main non-computational and practical problem with AIC and BIC is that they do not include the penalty for multiple testing. Therefore, when p is large they have a strong tendency to overestimate the number of true regressors (see [Bogdan et al., 2008]). This also has inferior influence on the prediction properties. Here it is important to note that while AIC gives unbiased estimator of prediction error for any given model, the estimator of prediction error for the model selected from a huge set of possible models by minimizing AIC will be strongly biased downward. The reason is that the distribution of the minimum of many variables is shifted to the left with respect to the distribution of any of these variables (unless they are all perfectly correlated).

mBIC

In [Bogdan et al., 2004] an interesting extension of BIC was proposed that stem from Bayesian perspective for sparse model selection found in the problem of finding quantitative trait loci (QTL). It turns out, that when dealing with sparse model, that is when number of variables in true model $\hat{p}$ is small compared to the large number of all available variables p, BIC tends to overestimate number $\hat{p}$. It means that a lot of noise is admitted to the model. The reason for that phenomenon is described in details in [Bogdan et al., 2004], but the main rationale is that, in the context of sparse regression model, number of model of size k, that is having k non-zero coefficients grows exponentially, namely $\binom{p}{k}$. Therefore, just by chance, larger models are more likely to overfit to the data by admitting noise. Solution to this problem is selecting different prior in 2.42 i.e. the one that takes into account number of variables. More specifically, following solution proposed by [George and McCulloch, 1993], [Bogdan et al., 2004] choose binomial distribution with fixed hyperparameter for the mean. The connection between this hyperparameter and the expected number of variables in the model is later established as well as the bound for the type I error (choosing non-null model, when all variables are noise).

Although this extension of BIC enhanced analysis of QTL, it still had some practical drawbacks. One most prominent is that method is, like regular BIC, intractable and requires using some kind of heuristics on which model we want to compare.

2.3.9 Least absolute shrinkage and selection operator (LASSO)

In the section 2.1 we discussed basic ideas in the topic of Convex Optimization. In contrast to many other techniques in optimization it's main feature is that it allows to solve a considerable variety of optimization problems using similar techniques. These solutions are exact, which makes it different from heuristics like greedy algorithms, genetic algorithms etc. Although some of them require significantly more computational power than problems for which simple derivative can be computed, the advent of stronger and cheaper computers in the 1990 drove more interest in Convex Optimization.

The best known example of using convex optimization techniques in statistics is by far LASSO. In his seminal paper [Tibshirani, 1996] proposed convex relaxation of l_0 penalty term in BIC.

Definition 2.3.19 *For given value vector* $\mathbf{y}$ *and plan matrix* $\mathbf{X}$ *in the regression problem, we call lasso estimator* β_{lasso} *of vector* β *a solution to the following optimization problem:*

$$\begin{aligned} &\underset{\beta}{maximize} && \frac{(y-\mathbf{X}\beta)^T(y-\mathbf{X}\beta)}{2\sigma^2} \\ &subject\ to && \sum |\beta_j| < t \end{aligned} \tag{2.43}$$

for some positive number t*.*

Transforming problem (2.43) through the Lagrange dual form one can obtain formulation that is similar to regularization methods introduced in previous sections.

Lemma 2.3.20 *Lasso problem can be reformulated as unconstrained form*

$$\underset{\beta}{minimize} \quad \frac{(y-\mathbf{X}\beta)^T(y-\mathbf{X}\beta)}{2\sigma^2} + \lambda_0 \sum |\beta_j| \tag{2.44}$$

where there is one to one correspondence between λ_0 *and* t*, for which there is however no closed form.*

Comparing (2.41) with (2.44) one can see that the difference is only in the penalty term. Lasso problem is convex relaxation of discrete optimization problem BIC (caused by using discrete l_0 norm). This resemblance causes Lasso to have multiple properties. First of all, for certain conditions and the right choice of λ_0 it is asymptotically consistent [Zhao and Yu, 2006]. Lasso, like BIC, also produces sparse solutions. The rationale behind it can best understood by analyzing figure (**??**).

Recall that ridge regression estimator can be viewed as Bayesian estimator for linear model in which normal prior on vector $\beta \sim \mathcal{N}(0, \lambda\mathbf{I})$ is assumed. It turns out that similar formulation is possible for Lasso.

Lemma 2.3.21 *Assume independent Laplace prior on every element* $\beta_i \sim Laplace(\nu)$ *in the classic regression model (2.36). Then mode of the posterior distribution Bayesian is equivalent to lasso.*

Proof
Let us write down the model from the lemma

$$\begin{aligned} &\mathbf{y} \sim \mathbf{X}\beta + \epsilon \\ &\text{where} \\ &\epsilon \sim \mathcal{N}(0, \sigma^2\mathbf{I}) \\ &\beta_i \sim \text{Laplace}(\nu) \end{aligned} \tag{2.45}$$

Then joint log-likelihood that we want to maximize is:

$$-\frac{n}{2}\log(2\pi) + \frac{1}{2}\log\sigma^2 - \frac{(y-\mathbf{X}\beta)^T(y-\mathbf{X}\beta)}{2\sigma^2} + p\log(2\nu) - \nu\sum|\beta_j| \tag{2.46}$$

When we exclude terms that do not depend on β we get:

$$-\frac{(y-\mathbf{X}\beta)^T(y-\mathbf{X}\beta)}{2\sigma^2} - \nu\sum|\beta_j| \tag{2.47}$$

Which is exactly the unconstrained Lasso problem. ∎

2.3.10 Sorted L-One Penalized Estimation (SLOPE)

Lasso is not the only convex relaxation of BIC. One frequently used is combination of l_1 and l_2 penalty terms called elastic net. We shall however focus on a one that involves completely new norm called sorted l_1 norm, J_λ.

Definition 2.3.22 *Function $J_\lambda : \mathbb{R}^p \to \mathbb{R}$ for nonnegative, nonincreasing sequence $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p \geq 0$ is defined as*

$$J_\lambda(\beta) = \sum_{i=1}^{p} \lambda_i |\beta_{(i)}| \tag{2.48}$$

where $\beta_{(i)}$ is the i-th biggest element in vector β in terms of absolute value.

Proposition 2.3.23 *J_λ is a norm*

Proof First let us prove that J_λ satisfies triangle inequality

$$\begin{aligned} J_\lambda(\beta_1+\beta_2) &= \sum_{i=1}^{p} \lambda_i |\beta_1+\beta_2|_{(i)} \\ &\leq \sum_{i*=1}^{p} \lambda_i |\beta_1|_{(i*)} + |\beta_2|_{(i*)} \\ &= \sum_{i=1}^{p} \lambda_i |\beta_{1(i*)}| + \sum_{i=1}^{p} \lambda_i |\beta_{2(i*)}| \\ &\leq \sum_{i=1}^{p} \lambda_i |\beta_{1(i)}| + \sum_{i=1}^{p} \lambda_i |\beta_{2(i)}| \end{aligned}$$

By $i*$ we mean the order of the vector $|\beta_1+\beta_2|$. The last inequality comes from the fact that if there are two non negative, decreasing sequences a and b then: $\sum_{i=1}^{p} a_i b_i \geq \sum_{i=1}^{p} a_i b_{\pi(i)}$ for every permutation π.
Absolutely homogeneity and point-separation come straight from the definition. ∎

J_λ is called sorted l_1 norm (SL1, a.k.a. OWL, ordered weighted ℓ_1). It will be a crucial element of Graphical SLOPE method we shall introduce in the subsequent chapter.

Proposition 2.3.24 *The dual norm to J_λ is given by $J_\lambda^D(x) := \max\left\{\frac{|x|_{(1)}}{\lambda_1}, \ldots, \frac{\sum_{i=1}^p |x|_{(i)}}{\sum_{i=1}^p \lambda_i}\right\}$*

Definition 2.3.25 *SLOPE [Bogdan et al., 2015a] is the solution to the following convex optimization problem*

$$\underset{\beta}{minimize} \quad (y - \mathbf{X}\beta)^T(y - \mathbf{X}\beta) + \sigma J_\lambda(\beta_j) \tag{2.49}$$

Definition 2.3.26 *By Benjamini-Hochberg λ sequence we denote*

$$\lambda_{BH}(i) := \Phi^{-1}(1 - \alpha_i), \;\; \alpha_i = \alpha\frac{i}{2p},$$

Theorem 2.3.27 (FDR control) *In the linear model with orthogonal design $\mathbf{X}$ and $\epsilon \sim \mathcal{N}(0, \sigma^2\mathbf{I}_n)$, the procedure*
rejecting hypotheses for which $\beta_j \neq 0$ has an FDR obeying

$$FDR = E\left[\frac{V}{R \vee 1}\right] \leq \alpha\frac{p_0}{p}.$$

In [Bogdan et al., 2015a] a λ sequence is shown that leads to the control of FDR under assumption of orthogonality of $\mathbf{X}$.
An algorithm for solving this problem was also introduced in the paper. Current implementation in R language (CRAN packge *SLOPE*) is based on FISTA method. This implementation included efficient algorithm for computing prox function (2.8). In section 5 we introduce an alternative implementation based on ADMM and show its superior computational properties for some types of design matrices $\mathbf{X}$.

2.4 Dimension reduction

2.4.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) [Pearson, 1901] is a dimensionality reduction technique that is widely used in practice. Its main application is in exploratory data analysis, where data is projected onto a small number of orthogonal directions (usually two dimensions). Thanks to the PCA researcher can get intuition on the data structure conjecture a hypothesis, that would be further investigated. Subsequent principal components refer to directions that explain less and less variability of the data. Therefore it seems natural to assume that some first components in PCA are signal, represent the values of interest, while remaining components as noise that could be discarded. In exploratory data analysis, it is often redundant to have a tool for precise choice of the number of non-noise components. However, there are other applications of PCA where precise distuingish between signal and noise is important. For example in projective clustering (see e.g. [Agarwal and Mustafa, 2004]), where data is clustered along various linear subspaces, an incorrect estimation of subspaces dimensions may lead to the choice of wrong number of clusters and incorrect segmentation. Another example is an important problem of missing values in PCA, where inaccurate estimation of the number of components may lead to overfitting (see [Josse et al., 2009], [Ilin et al., 2010], [Josse et al., 2011]).

2.4.2 Selecting the number of principal components

In Jolliffe [2002] three types of methods for choosing the number of factors are distinguished. First are ad-hoc rules such as a scree test [D'agostino and Russell, 2005] or a rule of thumb that chooses the smallest number of factors which jointly explain e.g. 90% of variance of data. Although these methods are usually fast and easy to implement, they are difficult to use in automatic way, since in high-dimensional data it is common that few first components explain a lot of variance even if data is entirely random [Husson et al., 2010].
Methods of the second type include techniques that consider the problem in a more systematic way, but do not rely on any probabilistic assumptions. Those are for example bootstrap and permutation methods (see [Jackson, 1993]) or cross-validation (see [Owen and Perry, 2009], [Josse and Husson, 2012]).
Finally there exists a group of methods based on specific probabilistic models. In the next chapter we shall focus on unifying a couple of different approaches.
To formally define our model, let us start with reformulating (2.55) as a model for either rows or columns of the matrix $\mathbf{X}$. We shall use notation $x_{i\bullet}$ and $x_{\bullet j}$ respectively.

$$\begin{aligned}\mathbf{x}_{i\bullet} - \boldsymbol{\mu}_{i\bullet} &= \sum_{l=1}^{k} t_{i,l}\,\mathbf{w}_{\bullet l} + \epsilon_{i\bullet} = \mathbf{t}_{i\bullet}\,\mathbf{W}^T + \epsilon_{i\bullet}, \quad i = 1,\dots,n, \quad \epsilon_{1\bullet},\dots,\epsilon_{n\bullet} \text{ are i.i.d. } \mathcal{N}(0,\mathbf{I}_p),\\ \mathbf{x}_{\bullet j} - \boldsymbol{\mu}_{\bullet j} &= \sum_{l=1}^{k} w_{j,l}\,\mathbf{t}_{\bullet l} + \epsilon_{\bullet j} = \mathbf{T}\,w_{j\bullet}^T + \epsilon_{\bullet j}, \quad j = 1,\dots,p, \quad \epsilon_{\bullet 1},\dots,\epsilon_{\bullet p} \text{ are i.i.d. } \mathcal{N}(0,\mathbf{I}_n).\end{aligned} \tag{2.50}$$

Let us focus here on the Bayesian estimation of number of components k. Idea is to, given data, select model that yields highest probability. This approach is called maximum *a posteriori* (MAP) rule.

For the simplicity of computations, we maximize the logarithm of the posterior probability. In the most general form it is of the form:

$$\begin{aligned}\log(P(k|\mathbf{X})) &= \log(P(\mathbf{X}|k)) + \log(P(k)) + C(\mathbf{X}) \\ &= \log\left(\int_\Theta p(\mathbf{X}|\theta)\,\pi_k(\theta)\,d\theta\right) + \log(P(k)) + C(\mathbf{X}),\end{aligned} \tag{2.51}$$

where $P(k)$ is the prior distribution of k concentrated on the set $\{1,\dots,\min(n,p)\}$, $\pi_k(\theta)$ is a prior distribution on the model parameters given k, and $C(\mathbf{X})$ is a scaling factor that does not depend on k.
In terms of model (2.55), $\theta = (\boldsymbol{\mu}, \mathbf{W}, \mathbf{T}, \sigma) \in \Theta$, and $P(\mathbf{X}|k)$ takes the following form:

$$\log(P(\mathbf{X}|k)) = \log \int_\Theta p(\mathbf{X}|\boldsymbol{\mu}, \mathbf{W}, \mathbf{T}, \sigma)\,\pi_M(\boldsymbol{\mu}, \mathbf{W}, \mathbf{T}, \sigma)\,d\boldsymbol{\mu}\,d\mathbf{W}\,d\mathbf{T}\,d\sigma. \tag{2.52}$$

2.4.3 Probabilistic model for PCA

Let $\mathbf{X} = [x_{ij}]_{n\times p}$ be high-dimensional data, where n is the number of observations and p is the number of variables. Consider the fixed effect model for PCA:

$$\mathbf{X} - \boldsymbol{\mu}_{n\times p} = \mathbf{M} + E, \tag{2.53}$$

where $\boldsymbol{\mu}$ is of rank one, $\mathbf{M}$ is assumed to be of low rank $k \leq \min(n, p)$
and $E = [\epsilon_{i,j}]_{n\times p}$ is a matrix of i.i.d. errors, $\epsilon_{i,j} \sim \mathcal{N}(0, \sigma^2)$.
Equivalently, we may use the singular value decomposition (SVD) of $\mathbf{M}$ and write (2.53) as

$$\mathbf{X} - \boldsymbol{\mu}_{n\times p} = \mathbf{PLQ}^T + E, \tag{2.54}$$

where $\mathbf{P}^T\mathbf{P} = \mathbf{I}_{k\times k}$, $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}_{k\times k}$ and $\mathbf{L}$ is a diagonal matrix with the singular values of $\mathbf{M}$, or use the fixed factor analysis representation as in Caussinus [1986]:

$$\mathbf{X} - \boldsymbol{\mu}_{n\times p} = \mathbf{TW}^T + E, \tag{2.55}$$

where $\mathbf{T} = [t_{i,l}]_{n\times k}$ is a matrix whose columns contain factors spanning the data, and $\mathbf{W} = [w_{i,l}]_{p\times k}$ is a matrix of coefficients.
Given the number of components k, maximum likelihood estimators for the parameters in model (2.53) are obtained by performing SVD of $\mathbf{X} - \boldsymbol{\mu}$ truncated at the order k (see for example [Caussinus, 1986], [Allen et al., 2014]).

2.4.4 Probabilistic PCA - maximum likelihood estimator

In [Tipping and Bishop, 1999a] a fixed effect model was considered that can be seen as a special case of (2.55) with:

$$\begin{aligned}\mathbf{V} &= \sigma^2\mathbf{I}_d\\ \mathbf{w} &\sim \mathcal{N}(0, \mathbf{I}_k)\end{aligned}$$

and $\boldsymbol{\mu}_{n\times p}$ having all rows equal μ. Then we can write for every row of $\mathbf{X}$

$$p(\mathbf{x}_i|\mathbf{T}, \mathbf{m}, v) \sim \mathcal{N}(\mu, \mathbf{TT}^T + \sigma^2\mathbf{I})$$

Theorem 2.4.1 *Maximum likelihood estimators for the model* (2.4.4) *are given by:*

$$\begin{aligned}\hat{\mu} &= \frac{1}{n}\mathbf{x}_i\\ \hat{\mathbf{T}} &= \mathbf{U}(\Lambda_k - v\mathbf{I}_k)^{1/2}\mathbf{R}\\ \hat{\sigma^2} &= \frac{\sum_{j=k+1}^{d}\lambda_j}{d-k}\end{aligned}$$

Proof Here we follow mostly [Tipping and Bishop, 1999a]. Some parts missing from this paper are treated with more details.
From our model for one row (2.4.4) of whole dataset (consisting of n variables).

$$\begin{aligned}p(\mathbf{X}|\mathbf{T}, \mathbf{m}, \sigma^2) &= \Pi_i p(\mathbf{x}_i|\mathbf{T}, \mathbf{m}, \sigma^2)\\ &= (2\pi)^{-nd/2}|\mathbf{TT}^T + \sigma^2\mathbf{I}|^{-n/2}\exp(-\frac{1}{2}tr((\mathbf{TT}^T + \sigma^2\mathbf{I})^{-1}\mathbf{S}))\\ \mathbf{S} &= \sum_i (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^T\end{aligned} \tag{2.56}$$

Obviously,

$$\hat{\mu} = \frac{1}{n}\mathbf{x}_i \tag{2.57}$$

From [Tipping and Bishop, 1999a] we have formula for maximum-likelihood estimator of $\mathbf{T}$.

$$\hat{\mathbf{T}} = \mathbf{U}(\Lambda_k - \sigma^2\mathbf{I}_k)^{1/2}\mathbf{R} \tag{2.58}$$

where $\mathbf{U}$ contains k top eigenvectors of vS/n matrix, Λ_k are corresponding eigenvalues and $\mathbf{R}$ is an arbitrary rotation matrix.
To derive this formula we analyze logarithm of 3.6.

$$-nd/2\log(2\pi) - \frac{n}{2}\log|\mathbf{T}\mathbf{T}^T + \sigma^2\mathbf{I}| - \frac{1}{2}\operatorname{Tr}((\mathbf{T}\mathbf{T}^T + \sigma^2\mathbf{I})^{-1}\mathbf{S}) \tag{2.59}$$

To compute derivative of (2.59) with respect to H we use rules described in [Minka, December 2000]. To make derivation more clear we use notation $C = \mathbf{T}\mathbf{T}^T + \sigma^2\mathbf{I}$. Observe that both C and S are symmetric matrices.

$$\begin{aligned}
\frac{\partial \log|C|}{\partial H} &= \operatorname{Tr}(C^{-1}\frac{\partial C}{\partial H}) \\
&= \operatorname{Tr}(C^{-1}\frac{\partial HH^T}{\partial H}) \\
&= \operatorname{Tr}(C^{-1}(\frac{\partial H}{\partial H}H^T + H(\frac{\partial H}{\partial H})^T)) \\
&= \operatorname{Tr}(H^TC^{-1}\frac{\partial H}{\partial H}) + \operatorname{Tr}(C^{-1}H\frac{\partial H^T}{\partial H}) \\
&= 2H^TC^{-1}
\end{aligned} \tag{2.60}$$

Here we have derivative of other part of log-likelihood in slightly different (abbreviated) notation.

$$\begin{aligned}
d\operatorname{Tr}(C^{-1}\mathbf{S}) &= \operatorname{Tr}(dC^{-1}\mathbf{S}) = -\operatorname{Tr}(C^{-1}dCC^{-1}\mathbf{S}) \\
&= -\operatorname{Tr}(C^{-1}d(HH^T)C^{-1}\mathbf{S}) \\
&= -\operatorname{Tr}(C^{-1}(dHH^T + HdH^T)C^{-1}\mathbf{S}) \\
&= -\operatorname{Tr}(H^TC^{-1}\mathbf{S}C^{-1}dH) - \operatorname{Tr}(C^{-1}\mathbf{S}C^{-1}HdH^T) \\
&= -H^TC^{-1}\mathbf{S}C^{-1} - H^TC^{-1}\mathbf{S}C^{-1} \\
&= -2H^TC^{-1}\mathbf{S}C^{-1}
\end{aligned} \tag{2.61}$$

From that we get a condition for stationary points

$$n(C^{-1}\mathbf{T} - C^{-1}\mathbf{S}C^{-1}\mathbf{T}) = 0 \tag{2.62}$$

So we get

$$\mathbf{T} = \mathbf{S}C^{-1}\mathbf{T} = 0 \tag{2.63}$$

Here one needs to write $\mathbf{T}$ in terms of its SVD decomposition as shown in [Tipping and Bishop, 1999a]. Briefly, a stationary point is any approximation by k eigenvalues. From likelihood we get that taking actually largest eigenvalues maximizes likelihood. We can even compute the difference between the global maximum and second local maximum. This is actually important for Laplace approximation.

Now we move to ML estimate for v.

$$p(\mathbf{X}|\mathbf{H}=\hat{\mathbf{T}},\mu=\hat{\mu},\sigma^2)=(2\pi)^{-nd/2}|\hat{\mathbf{T}}\hat{\mathbf{T}}^T+\sigma^2\mathbf{I}|^{-n/2}\exp(-\frac{1}{2}\operatorname{Tr}((\hat{\mathbf{T}}\hat{\mathbf{T}}^T+\sigma^2\mathbf{I})^{-1}\mathbf{S})) \tag{2.64}$$

Let us consider first

$$\begin{aligned}\hat{\mathbf{T}}\hat{\mathbf{T}}^T+\sigma^2\mathbf{I}&=\mathbf{U}(\Lambda_k-\sigma^2\mathbf{I}_k)^{1/2}\mathbf{R}(\mathbf{U}(\Lambda_k-\sigma^2\mathbf{I}_k)^{1/2}\mathbf{R})^T+\sigma^2\mathbf{I}=\\&=\mathbf{U}(\Lambda_k-\sigma^2\mathbf{I}_k)^{1/2}\mathbf{R}\mathbf{R}^T(\Lambda_k-\sigma^2(\mathbf{I}_k)^{1/2}))^T\mathbf{U}^T+\sigma^2\mathbf{I}=\\&=\mathbf{U}(\Lambda_k-\sigma^2\mathbf{I}_k)\mathbf{U}^T+\sigma^2\mathbf{I}_d\end{aligned} \tag{2.65}$$

In 3.5 we use the fact that $\mathbf{R}$ is orthogonal square matrix.
To compute determinant we use Sylvester's determinant theorem.

$$\begin{aligned}|\mathbf{U}(\Lambda_k-\sigma^2\mathbf{I}_k)\mathbf{U}^T+\sigma^2\mathbf{I}_d|&=|\sigma^2\mathbf{I}_d||\mathbf{I}_k+(\Lambda_k-\sigma^2\mathbf{I}_k)\mathbf{U}^T\frac{1}{\sigma^2}\mathbf{I}_d\mathbf{U}|\\&=(\sigma^2)^d|\mathbf{I}_k+(\Lambda_k-\sigma^2\mathbf{I}_k)\frac{1}{\sigma^2}\mathbf{U}^T\mathbf{U}|\\&=(\sigma^2)^d|\mathbf{I}_k+(\Lambda_k/\sigma^2-\mathbf{I}_k)\mathbf{I}_k| \qquad (2.66)\\&=(\sigma^2)^d|\Lambda_k/\sigma^2|\\&=(\sigma^2)^d\Pi_{j=1}^k\lambda_j(\sigma^2)^{-k}\\&=\Pi_{j=1}^k\lambda_j(\sigma^2)^{d-k} \qquad (2.67)\end{aligned}$$

In 2.66 we use the fact that $\mathbf{U}$ is orthogonal matrix.
Similarly we consider

$$\begin{aligned}\operatorname{Tr}((\hat{\mathbf{T}}\hat{\mathbf{T}}^T+\sigma^2\mathbf{I})^{-1}\mathbf{S})&=\operatorname{Tr}((\mathbf{U}(\Lambda_k-\sigma^2\mathbf{I}_k)^{1/2}\mathbf{R}(\mathbf{U}(\Lambda_k-\sigma^2\mathbf{I}_k)^{1/2}\mathbf{R})^T+\sigma^2\mathbf{I})^{-1}\mathbf{S})\\&=\operatorname{Tr}((\mathbf{U}(\Lambda_k-\sigma^2\mathbf{I}_k)\mathbf{U}^T+\sigma^2\mathbf{I}_d)^{-1}\mathbf{S})\end{aligned} \tag{2.68}$$

Now suppose $\mathbf{S}=n\mathbf{U}_d\Lambda\mathbf{R}_d$ and A - square diagonal matrix with upper left block equal $(\Lambda_k-v\mathbf{I}_k)$

$$\begin{aligned}&\operatorname{Tr}((\mathbf{U}(\Lambda_k-\sigma^2\mathbf{I}_k)\mathbf{U}^T+\sigma^2\mathbf{I}_d)^{-1}\mathbf{S})\\&\qquad=\operatorname{Tr}((\mathbf{U}_dA\mathbf{U}_d^T+\mathbf{U}_d\sigma^2\mathbf{I}_d\mathbf{U}_d^T)^{-1}n\mathbf{U}_d\Lambda\mathbf{R}_d)\\&\qquad=\operatorname{Tr}((\mathbf{U}_dB\mathbf{U}_d^T)^{-1}n\mathbf{U}_d\Lambda\mathbf{R}_d)\\&\qquad=n\operatorname{Tr}((\mathbf{U}_d^{-1}B^{-1}\mathbf{U}_d^{-1}\mathbf{U}_d\Lambda\mathbf{R}_d)\\&\qquad=n\operatorname{Tr}((\mathbf{U}_dB^{-1}\Lambda\mathbf{R}_d)\end{aligned} \tag{2.69}$$

Observe that since diagonal of $B^{-1}\Lambda$ are eigenvalues of matrix $\mathbf{U}_dB^{-1}\Lambda\mathbf{R}_d$ then trace equals

$$n\cdot(k+\sum_{j=k+1}^{d}\frac{\lambda_j}{\sigma^2}) \tag{2.70}$$

This gives us a formula for a likelihood

$$\begin{aligned}p(\mathbf{X}|\mathbf{H}=\hat{\mathbf{T}},\mu=\hat{\mu},\sigma^2)&=(2\pi)^{-nd/2}\left(\Pi_{j=1}^k\lambda_j(\sigma^2)^{d-k}\right)^{-n/2}\exp(-\frac{1}{2}n\cdot(k+\sum_{j=k+1}^{d}\frac{\lambda_j}{\sigma^2}))\\&=(2\pi)^{-nd/2}\left(\Pi_{j=1}^k\lambda_j\right)^{-n/2}(\sigma^2)^{-n(d-k)/2}\exp(-\frac{nk}{2})\exp(\frac{n}{2\sigma^2}\sum_{j=k+1}^{d}\lambda_j)\end{aligned} \tag{2.71}$$

From this we shall compute ml estimator for noise v. Taking, again, log-likelihood

$$-nd/2\log(2\pi) - \frac{n}{2}\sum_{j=1}^{k}\log(\lambda_j) - \frac{n(d-k)}{2}\log(\sigma^2) - \frac{nk}{2} - \frac{n}{2\sigma^2}\sum_{j=k+1}^{d}\lambda_j \tag{2.72}$$

Taking derivative wrt v yields

$$\begin{aligned}
&(-\frac{n(d-k)}{2}\log(\sigma^2) - \frac{n}{2\sigma^2}\sum_{j=k+1}^{d}\lambda_j)' = 0\\
&-\frac{(d-k)}{\sigma^2} + \frac{\sum_{j=k+1}^{d}\lambda_j}{(\sigma^2)^2} = 0\\
&\sum_{j=k+1}^{d}\lambda_j = \sigma^2(d-k)\\
&\hat{\sigma^2} = \frac{\sum_{j=k+1}^{d}\lambda_j}{d-k}
\end{aligned} \tag{2.73}$$

∎

In the paper [Tipping and Bishop, 1999a] no method for selecting number of PCs. Several criteria including BIC, were proposed [Minka, 2000].

2.4.5 Full Bayesian approach

There exist several Bayesian methods for estimating the number of principal components in the PCA model. One of them was proposed in Bishop [1999a], who used the following priors in model (2.55):

$$\begin{aligned}
\mathbf{t}_{i\bullet} &\sim \mathcal{N}(0, \mathbf{I}), & \alpha_j &\sim \Gamma(a_\alpha, b_\alpha),\\
\mathbf{w}_{j\bullet} &\sim \mathcal{N}(0, \frac{1}{\alpha_j}\mathbf{I}), & \frac{1}{\sigma^2} &\sim \Gamma(c_\sigma, d_\sigma),
\end{aligned} \tag{2.74}$$

where $a_\alpha, b_\alpha, c_\sigma, d_\sigma$ are model hyperparameters. The rows of $\boldsymbol{\mu}$ were estimated by $\bar{\mathbf{x}} = (\overline{\mathbf{x}}_{\cdot 1}, \ldots, \overline{\mathbf{x}}_{\cdot p})$, where $\overline{\mathbf{x}}_{\cdot j} = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$.

Bishop [1999a] introduces non-discrete "model selection" for PCA, by means of continuous parameters, that control the variability of the columns of $\mathbf{W}$. More specifically, a large value of α_j effectively "switches off" $\mathbf{w}_{j\bullet}$. Bishop [1999a] proposes three computational methods for marginalizing over the posterior on $\mathbf{W}$, including, among others, Markov Chain Monte Carlo. In a follow up paper, Bishop [1999b] recommends the variational approach, which proves to be the most efficient. This idea was further pursued by Ilin et al. [2010], who propose a fast algorithm for variational Bayesian PCA (VBPCA), which however does not enable direct estimation of the number of PCs. In fact, Ilin et al. [2010] note, in the context of missing values, that the quality of the reconstructed matrix depends on estimating the number of PCs beforehand, and they suggest using methods based on the Laplace approximation from Minka [2000] for this task. The Variational Bayes approach was also used in Nakajima et al. [2015], who propose a numerical algorithm for estimating the number of PCs. This algorithm is designed for the asymptotic regime when p and n go to infinity at the same rate and under this setup it turns out

to be suboptimal compared to the competitive method of Hoyle [2008], based on an extended version of the Laplace approximation.
Another Bayesian approach was proposed by Hoff [2007], who considered the representation (2.54) with $\boldsymbol{\mu} = 0$ and the following priors imposed on the components of SVD:

$$\begin{aligned} \mathbf{P} &\sim uniform(\mathcal{S}_{k\times n}), & \ell &\sim \mathcal{N}(l_0, v_0^2), \\ \mathbf{Q} &\sim uniform(\mathcal{S}_{k\times p}), & \psi &\sim \Gamma(\eta_0/2, \eta_0\tau_0^2/2), \\ \epsilon_{i\bullet} &\sim \mathcal{N}(0, 1/\phi), & \phi &\sim \Gamma(\nu_0/2, \nu_0\sigma_0^2/2), \\ l_{i,i} &\sim \mathcal{N}(\ell, 1/\psi), \end{aligned} \tag{2.75}$$

where $uniform(\mathcal{S}_{k\times n})$ denotes the uniform distribution on the Stiefel manifold of orthogonal matrices [Chikuse, 2003], $l_{i,i}$ are elements of the diagonal matrix L and (l_0, v_0^2), (η_0, τ_0), (ν_0, σ_0) are hyper-parameters. To estimate the number of principal components, Hoff [2007] considers the model with $k = p$ and uses the prior on $l_{i,i}$ specified in (2.75) as a continuous component in the spike and slab prior, with a positive mass at 0. The posterior distributions of the parameters are computed by MCMC. The software provided by Hoff [2007] requires $n \geq p$; however, because of the symmetry in the model (2.75), when $p > n$ one may transpose the data and then use the method. Due to the complexity of MCMC, implementation is rather slow and does not scale very well. Because even for moderately sized matrices (*i.e.* 1000×100) generating a Markov chain of length 1000 takes more than two hours, we decided not to include this method in the simulation study.

2.5 Subspace clustering

Subspace clustering problem comes from the field of computer vision. Such data sets contains huge number of variables. Since computer visions models operate on limited number of parameters related to appearance, geometry and dynamics of a scene, many researchers developed methods for finding a low-dimensional representation of a high-dimensional data set. Since in this case data naturally comes from multiple subspaces (motion different objects in a movie). This need resulted in multiple papers with methods for performing subspace clustering.

We define subspace clustering following [Vidal, 2011]. Let $\mathbf{X} \in \mathbb{R}^{n\times p}$ be our data set. We assume that it is drawn from a union of K affine subspaces $S_1, \dots S_K$ of unknown dimensions $k_1, \dots k_K$ respectively.

$$S_i = \{\mathbf{x} \in \mathbb{R}^p \,:\, \mathbf{x} = \mu_i + U_i\mathbf{y},\ \mathbf{y} \in \mathbb{R}^{k_i}\}$$

The goal of subspace clustering is finding number of subspaces K, their dimensions $\{k_i\}$, points μ_i, subspace bases U_i and segmentation of data points. We shall sometimes refer to subspaces as clusters (they 'bind' variables).

An example of subspace clustering with $p = 3$ and $K = 4$ can be seen in figure 2.2.

There are multiple methods for solving this problem, their formulation does not necessarily include clearly defined probabilistic model. Significant group is based on spectral clustering [Elhamifar and Vidal, 2009], [Vidal and Favaro, 2014], [Liu et al., 2013]. Others are iterative methods that aim at optimizing some goodness of fit criterion [Agarwal and Mustafa, 2004] [Timmerman et al., 2013] [Chavent et al., 2012] [Vigneau and Qannari, 2003]. There are however several papers that define statistical model. Among them MPPCA by [Tipping and Bishop, 1999a] (mentioned in section 2.4.4) and [Soltanolkotabi et al., 2013].

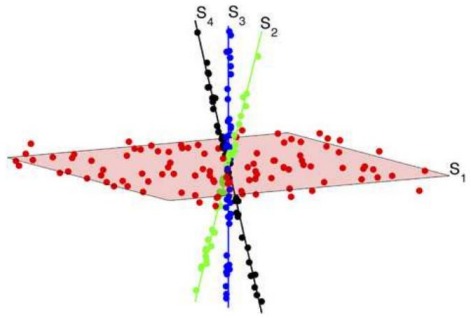


Figure 2.2: Example of data in subspace clustering problem. From paper [Soltanolkotabi et al., 2013]

2.5.1 Methods based on spectral clustering

In the simulation study we shall compare our new method to two based on spectral clustering (Luxburg [2007]). These methods exploit the notion of a similarity matrix. Let $X = (x_1, \ldots, x_p)$ be the data set. Clustering is not done directly on data X. In general, spectral clustering methods consist of several steps.

- We define similarity matrix $A = (a_{ij})_{i,j=1}^{n}$, where a_{ij} stands for the similarity between x_i and x_j. This could be for example based on correlations.
- Then based on A, we construct graph $G = (V - \text{vertices}, E - \text{edges})$ with vertices corresponding to columns of X. We also construct adjacency matrix W, which gives weights to the edges of the graph.
- Clustering is performed on this adjacency matrix. We look for a partition of a graph such that the edges within one group have large weights and edges between two groups have small weights.

  We define D as a diagonal matrix such that $D_{ii} = \sum_{j=1}^{p} w_{ij}$

  We define Laplacian matrix L as some function of D and W (for example check Ng et al. [2001])
- Now we store the k (number of clusters) eigenvectors of L, corresponding to k largest eigenvalues, in $U_k = (u_1, \ldots, u_k)$ as columns. In the simplest version of the algorithm rows of matrix U_k: $r_1, \ldots, r_n$ are grouped into clusters $C_1, \ldots, C_k$ and desired clustering is $A_1, \ldots, A_k$ where $A_i = \{j | r_j \in C_i\}$. For details check Luxburg [2007] and Ng et al. [2001].

Sparse subspace clustering (SSC) from Elhamifar and Vidal [2009] uses the spectral clustering method from Ng et al. [2001]. It is based on the assumption that data comes from the union of subspaces. The idea is that such data is self expressive. This means that for every point in the data set $Y = (y_1, \ldots y_n)$, there exists sparse vector c_i, such that $y_i = Yc_i + z_i$, where $Z = (z_1, \ldots, z_n)$ denotes the noise matrix. The main observation (in noiseless case) is that there exists such a sparse solution c_i that data points corresponding to its non-zero coordinates lie in the same subspace as y_i. So the first part of SSC is formulated as the optimization problem

$$\text{Find } \min_{C,Z} \left(||C||_1 + \frac{\lambda}{2} ||Z||_F^2 \right) \text{ s.t } Y = YC + Z \text{ and } C_{ii} = 0 \text{ for } i = 1, \ldots, n$$

where λ is a tuning parameter and $||\ ||_F$ is the Frobenius norm (square root of the sum of squared eigenvalues). After this step spectral clustering is performed using adjacency matrix $W = abs(C) + abs(C^T)$ where $abs(C)$ denotes the result of taking element-wise the absolute value.

Low-rank subspace clustering (LRSC) is a similar method but we formulate the optimization problem differently.

$$\text{Find } \min_{A,C,Z} \left(||C||_* + \frac{\tau}{2}||A - AC||_F^2 + \frac{\lambda}{2}||Z||_F^2 \right) \text{ s.t } Y = A + Z \text{ and } C = C^T$$

which is a convex relaxation of

$$\text{Find } \min_{A,C,Z} \left(||C||_* + \frac{\lambda}{2}||Z||_F^2 \right) \text{ s.t } Y = A + Z,\ A = AC \text{ and } C = C^T$$

where $||\ ||_*$ denotes the nuclear norm (the sum of the eigenvalues). Such formulation causes the matrix C (which is later used in spectral clustering as adjacency) to be low rank and allows this problem to have closed solution which is shown in Vidal and Favaro [2014]. Due to this property, LRSC is not computationally complex.

2.6 Gaussian Graphical Models

Graphical model is a probabilistic model represented as a graph in which random variables are vertices V and their dependency structure is given by edges E. There are many ways to measure this dependency, the most straightforward being correlation, but also partial correlation or conditional independence.

The topic of graphical models is very wide, and we shall focus only on one specific type of model called Gauss-Markov Random Field and one measure of dependency we shall focus on partial correlation.

Definition 2.6.1 *Let X and Y be random variables and Z random vector. We define partial correlation between X and Y given Z by:*

$$\rho(X, Y|Z) := \rho(\epsilon_X, \epsilon_Y), \tag{2.76}$$

where $\epsilon_X = X - \pi_Z(X)$ and $\epsilon_Y = Y - \pi_Z(Y)$ are residuals from linear regression on the linear subspace spanned by Z.

We want to estimate partial correlation matrix R

Lemma 2.6.2 *Partial correlation matrix is given by*

$$R_{i,j} := \frac{-\Omega_{i,j}}{\sqrt{\Omega_{i,i}\Omega_{j,j}}}, \tag{2.77}$$

where $\Omega = \Sigma^{-1}$ is precision matrix, inverse covariance matrix.

We are covering just Gaussian graphical models, which is a very special case. In particular the following theorem holds (for proof see e.g. [Lauritzen, 1996].

Theorem 2.6.3 *Let $\mathbf{X} = (X_1, \dots X_p) \sim \mathcal{N}(\mu, \Sigma)$ and $\Theta^{-1} = \Sigma$. Then X_i and X_j and conditionally independent if and only if $\Theta_{i,j} = 0$*

From the theorem above, we know that for Gaussian graphical model, no correlation means independence.

2.6.1 Likelihood function

Let vertices of the graph $\mathbf{x} = (x_1, \ldots, x_p)$ follow zero-mean multivariate Gaussian with covariance matrix Σ, $\mathbf{x} \sim \mathcal{N}(\mu, \Sigma)$. Then, edges are given by non-zero elements of inverse covariance matrix $\Theta = Sigma^{-1}$. We assume non-degenerate case when Σ is invertible matrix.

For data $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_n$ log-likelihood $l(\mathbf{X}, \Theta)$ takes the form

$$\begin{aligned} l(\mathbf{X}, \Theta) &= \frac{1}{n} \sum_{i=1}^{n} \log l_\Theta(x_i) \\ &= \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} \log \det[\Theta/2\pi] - \frac{1}{2} x_i^T \Theta x_i \\ &= \frac{1}{2n} \sum_{i=1}^{n} \log \det \Theta - n \log 2\pi - x_i^T \Theta x_i \\ &= \frac{1}{2} \log \det \Theta - \frac{n}{2} \log 2\pi - \frac{1}{2n} \sum_{i=1}^{n} \operatorname{Tr}\left(-x_i^T x_i \Theta\right) \\ &= \frac{1}{2} \log \det \Theta - \frac{n}{2} \log 2\pi - \frac{1}{2n} \sum_{i=1}^{n} - \operatorname{Tr}\left(x_i x_i^T \Theta\right) \\ &= \frac{1}{2} \log \det \Theta - \frac{n}{2} \log 2\pi - \frac{1}{2} \operatorname{Tr}\left(\mathbf{S}\Theta\right), \end{aligned} \tag{2.78}$$

where $\mathbf{S}$ is sample covariance matrix given by $\frac{1}{n} \sum_{i=1}^{n} x_i x_i^T$.

Up to a constant, log-likelihood function is equal to

$$l(\mathbf{X}, \Theta) = \log \det \Theta - \operatorname{Tr}\left(\mathbf{S}\Theta\right). \tag{2.79}$$

Considering above as optimization problem, obserce that objective function (**??**) is strictly concave (see **?**) so maximum must be unique, and defines the precision matrix MLE $\widehat{\Theta}_{ML}$. Because for this problem general regularity conditions hold, MLE converges to the true parameter Θ as n goes to infinity. This leads to the idea, that finding Gaussian graphical model could consist on thresholding on entries of $\widehat{\Theta}_{ML}$.

In fact, the most straightforward way to test whether $r_{i,j} = 0$ is using plug-in estimators.

$$\hat{r}_{ij} = \frac{-\hat{\Omega_{i,j}}}{\sqrt{\hat{\Omega_{i,i}}\hat{\Omega_{j,j}}}}, \tag{2.80}$$

where $\hat{\Omega} = \mathbf{S}^{-1}$, inverse of sample covariance matrix.

It can be done using either bootstrap or using normal approximation.

$$Z_{i,j} = \frac{1}{2} \log\left(\frac{1 + r_{i,j}}{1 - r_{i,j}}\right) \sim \mathcal{N}\left(\theta_{i,j}, \frac{1}{n - p - 5}\right) \tag{2.81}$$

However, especially in practice, the number of nodes p may be comparable to, or larger than, the sample size n. It is obvious that when number of variables is higher then the number of samples, sample covariance matrix becomes not invertible, MLE estimator is no longer unique and no testing based on (2.81) is possible. Furthermore, variance of estimators is very high when p and n are comparable. Therefore idea is to add some kind or regularization.

The most basic idea is to adjust $\mathbf{S}$ matrix in $\hat{\Sigma}$ by shrinking off-diagonal elements of $\mathbf{S}$ and thus making it invertible. Specifically ,

$$\hat{\Omega} = ((1-\epsilon)\mathbf{S} + \epsilon D)^{-1},$$

where $D_{ii} = S_{ii}$.

One can choose ϵ so that estimator risk is minimized [Ledoit and Wolf, 2004]. Testing is done using bootstrap. Please note the idea behind this approach is very similar to ridge regression in which we can in fact also use bootstrap to test significance of variables. In the chapter on Graphical SLOPE we shall give description of other method that uses l_1 penalty for the sparse estimation of the matrix Θ rather that testing each entry of precision matrix separately.

2.6.2 Connectivity components

Definition 2.6.4 *We say that $C_k \subset V$ is a connectivity component of node x_k if it contains all vertices connected to x_k by some chain of edges*

Note that when vertex x_j is not in the component C_k, then variables x_j and x_k are independent.

3 Estimating number of Principal Components

Overview

In this chapter we introduce a novel approach to choosing number of principal components. PEnalized SEmi-integrated Likelihood (PESEL) [Sobczyk et al., 2017a] can be seen as a kind of compromise between fixed effect and full Bayesian models. It is an extension of papers described in section 2.4.2. We follow MAP rule (2.51) but instead of assuming specific prior distributions and performing exact calculation like in Hoff [2007], we approximate the integral in (2.52) using Laplace approximation (2.3.4). The rational behind this approach is that full Bayesian methods require exact computation of posterior, which results in heavy computations and need making very specific assumptions about prior distributions. We on the other hand minimize number of made assumptions to minimum, which makes our method robust. In section 3.2 we prove that PESEL is consistent, under few assumptions stated in that section. Furthermore, we created an efficient implementation in **R** package *pesel* [Sobczyk et al., 2018]. We extensively tested properties and performance of PESEL in the simulations study. It proved to be a robust methods compared to other classical and recent approaches. This chapter is based on the paper Sobczyk et al. [2017a].

3.1 PEnalized SEmi-integrated Likelihood (PESEL)

When using representation (2.55), the integrated likelihood in the fixed effect model takes the form:

$$\begin{aligned}\log P(\mathbf{X}|k) &= \log \int_{\Theta} P(\mathbf{X}|\boldsymbol{\mu}, \mathbf{W}, \mathbf{T}, \sigma)\, \pi_k(\boldsymbol{\mu}, \mathbf{T}, \mathbf{W}, \sigma)\, d\boldsymbol{\mu}\, d\mathbf{T}\, d\mathbf{W}\, d\sigma \\ &= \log \int_{\Theta} \Pi_{i=1}^{n} \phi(x_{i\bullet}; \boldsymbol{\mu}_{i\bullet} + \mathbf{t}_{i\cdot}\mathbf{W}^T, \sigma^2\mathbf{I}_p)\, \pi_k(\boldsymbol{\mu}, \mathbf{T}, \mathbf{W}, \sigma)\, d\boldsymbol{\mu}\, d\mathbf{T}\, d\mathbf{W}\, d\sigma, \end{aligned} \tag{3.1}$$

where $\phi(x; \mathbf{m}, \boldsymbol{\Sigma})$ is the probability density function of the normal distribution with mean $\mathbf{m}$ and covariance matrix $\boldsymbol{\Sigma}$. It is invalid to apply Laplace approximation directly to the integral in (3.1), as the number of parameters in this model is proportional to both the number of observations n and the number of variables p. This violates the assumption in the Laplace approximation that the dimension of the parameter space is constant. Thus, to perform such an approximation one should reduce the dimensionality, for example by integrating out the prior on either $\mathbf{T}$ or $\mathbf{W}$. This choice is determined by asymptotics. For $p \to \infty$, we need to integrate out $\mathbf{W}$ because its number of parameters grows linearly in p. Similarly, for $n \to \infty$, $\mathbf{T}$ needs to be integrated out. After integrating out one of the priors, we can apply the Laplace approximation for the resulting semi-integrated likelihood. This yields a new Bayesian criterion

for estimating the dimension of the model, which we call PEnalized SEmi-integrated Likelihood (PESEL).

In the introduction chapter we defined general MAP rule (2.51). In this chapter, for the simplicity of calculations, we assume that prior distribution on models $P(k)$ is uniform *i.e.* it does not influence model selection. Please note, that *PESEL* can be readily used with any prior distribution as the crux of the method lies within approximating the other part of posterior probability.

3.1.1 PESEL for p fixed and $n \to \infty$

If we work in an asymptotic regime where $n \to \infty$, then to apply the Laplace approximation we need to integrate out $\mathbf{T}$ from (3.1) according to the formula:

$$\log P(\mathbf{X}|k) = \log \int SIL(\mathbf{X}|\boldsymbol{\mu}, \mathbf{W}, \mathbf{T}, \sigma)\pi(\mu, \mathbf{W}, \sigma)\, d\boldsymbol{\mu}\, d\mathbf{W}\, d\sigma,$$

where $SIL(\mathbf{X}|\boldsymbol{\mu}, \mathbf{W}, \mathbf{T}, \sigma) := \int P(\mathbf{X}|\boldsymbol{\mu}, \mathbf{W}, \mathbf{T}, \sigma)\, \pi(\mathbf{T})\, d\mathbf{T}$ is a semi-integrated likelihood function. In the above $\boldsymbol{\mu} = \boldsymbol{\mu}_{i\bullet} = [\mu_1, \mu_2, \ldots, \mu_p]$ are the rows of $\boldsymbol{\mu}$ from equation (2.55).

We propose using two forms of PESEL, based on specific prior distributions on the rows of $\mathbf{T}$. Firstly, we use the prior $\mathbf{t}_{i,\bullet} \sim \mathcal{N}(0, \mathbf{I}_k)$, which gives the Probabilistic Principal Component Analysis (PPCA) model of Tipping and Bishop [1999a]. (a random-effects version of our fixed-effects model (2.55)). In this case $\mathbf{t}_{i\bullet}\, \mathbf{W}^T \sim \mathcal{N}(0, \mathbf{W}\mathbf{W}^T)$. Therefore our semi-integrated likelihood is reduced to the likelihood in PPCA, under which $\mathbf{x}_{1\bullet}, \ldots, \mathbf{x}_{n\bullet}$ are independent and

$$\mathbf{x}_{i\bullet} = \boldsymbol{\mu} + \mathbf{t}_{i\bullet}\mathbf{W}^T + \epsilon_{i\bullet} \sim \mathcal{N}(\boldsymbol{\mu}; \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}_p). \tag{3.2}$$

The second approach is to consider a prior $\mathbf{t}_{i\bullet} \sim \frac{1}{\beta}\mathcal{N}(0, \mathbf{I}_k)$ with the additional restriction $\mathbf{W}^T\mathbf{W} = \mathbf{I}_k$. This constraint makes all the singular values in PCA *homogeneous* (we will refer to this by the abbreviation *homo*). In other words, all the PCA factors are equally weighted, *i.e.* none of the directions dominate the data. This distinguishes it from the previous prior, which allows for *heterogeneous* (abbreviated to *hetero*) singular values. The resulting *homogeneous* distribution for $\mathbf{t}_{i\bullet}$ was discussed in [Rajan and Rayner, 1997]. With this prior $\mathbf{t}_{i\bullet}$, $\mathbf{W}^T \sim \mathcal{N}(0, \frac{1}{\beta}\mathbf{W}\mathbf{W}^T)$, and the semi-integrated likelihood function for the rows of $\mathbf{X}$ corresponds to $\mathbf{x}_{1\bullet}, \ldots, \mathbf{x}_{n\bullet}$ being independent and

$$\mathbf{x}_{i\bullet} \sim \mathcal{N}(\boldsymbol{\mu}; \frac{1}{\beta}\mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}_p). \tag{3.3}$$

Let us now focus on the semi-integrated likelihood specified in formula (3.2), which yields

$$\begin{aligned}\log P(\mathbf{X}|k) &= \log \int SIL(\mathbf{X}|\mu, \mathbf{W}, \sigma)\, \pi(\boldsymbol{\mu}, \mathbf{W}, \sigma)\ d\mu\, d\mathbf{W}\, d\sigma \\ &= \log \int \Pi_{i=1}^n \phi(\mathbf{x}_{i\cdot} - \boldsymbol{\mu}; \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}_p)\, \pi(\mu, \mathbf{W}, \sigma)\, d\mu\, d\mathbf{W}\, d\sigma. \end{aligned} \tag{3.4}$$

Now, assuming that $p << n$ and provided that $\pi(\boldsymbol{\mu}, \mathbf{W}, \sigma)$ satisfies standard regularity conditions, it is possible to apply the Laplace approximation to the integral in (3.4).

To calculate number of free parameters in (3.4) let us decompose the matrix $\mathbf{W}$, as in (3.9):

$$\begin{aligned}\mathbf{W} &= \mathbf{U}(\mathbf{L} - \sigma^2\mathbf{I}_k)^{1/2}\mathbf{R},\\ \mathbf{U}^T\mathbf{U} &= \mathbf{I}_k,\\ \mathbf{R}^T\mathbf{R} &= \mathbf{I}_k.\end{aligned}$$

This implies the following equality:

$$\begin{aligned}\mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}_p &= \mathbf{U}(\Lambda_k - \sigma^2\mathbf{I}_k)^{1/2}\mathbf{R}(\mathbf{U}(\Lambda_k - \sigma^2\mathbf{I}_k)^{1/2}\mathbf{R})^T + \sigma^2\mathbf{I}_p =\\ &= \mathbf{U}(\Lambda_k - \sigma^2\mathbf{I}_k)^{1/2}\mathbf{R}\mathbf{R}^T(\Lambda_k - \sigma^2(\mathbf{I}_k)^{1/2}))^T\mathbf{U}^T + \sigma^2\mathbf{I}_p = \\ &= \mathbf{U}(\Lambda_k - \sigma^2\mathbf{I}_k)\mathbf{U}^T + \sigma^2\mathbf{I}_p.\end{aligned} \tag{3.5}$$

In the above derivation we use the fact that $\mathbf{R}$ is an orthogonal square matrix.
Using (3.5) we can write the likelihood in (3.4) for the whole data as:

$$\begin{aligned}p(\mathbf{X}|\mu, \mathbf{W}, \sigma^2) &= \Pi_{i=1}^n p(\mathbf{x}_{i,\cdot}|\mu, \mathbf{W}, \sigma^2)\\ &= (2\pi)^{-pn/2}|\mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}_p|^{-n/2}\exp\left[-\frac{1}{2}tr((\mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}_p)^{-1}\mathbf{S})\right]\\ &= (2\pi)^{-pn/2}|\mathbf{U}(\Lambda_k - \sigma^2\mathbf{I}_k)\mathbf{U}^T + \sigma^2\mathbf{I}_p|^{-n/2}\exp\left[-\frac{1}{2}tr\left((\mathbf{U}(\Lambda_k - \sigma^2\mathbf{I}_k)\mathbf{U}^T + \sigma^2\mathbf{I}_n)^{-1}\mathbf{S}\right)\right].\end{aligned} \tag{3.6}$$

We assume that all parameters are *a priori* independent. Then, since $\mathbf{R}$ is not part of the likelihood in (3.6), it can be integrated out. Thus, the integral in (3.4) is reduced to

$$\int p(\mathbf{X}|\mu, \mathbf{W}, \sigma^2)d\mu\ d\mathbf{W}\ d\sigma^2 = \int p(\mathbf{X}|\mu, \mathbf{U}, \mathbf{L}, \mathbf{R}, \sigma^2)d\mu\ d\mathbf{U}\ d\mathbf{L}\ d\mathbf{R}\ d\sigma^2 = \int p(\mathbf{X}|\mu, \mathbf{U}, \mathbf{L}, \sigma^2)d\mu\ d\mathbf{U}\ d\mathbf{L}\ d\sigma \tag{3.7}$$

$\mathbf{U}$ can be described with $pk - \frac{k(k+1)}{2}$ parameters – this is a dimension of the $p \times k$ Steifel manifold [James, 1954]. $\mathbf{L}$ has k parameters, μ has p parameters and σ is one parameter.

$$\log P(\mathbf{X}|k) \approx \log SIL(\mathbf{X}|\hat{\boldsymbol{\mu}}, \hat{\mathbf{W}}, \hat{\sigma}) - \frac{1}{2}K\log n, \tag{3.8}$$

where $K = \dfrac{pk - \frac{k(k+1)}{2} + k + p + 1}{2}$ is the number of free parameters in the integral in (3.4).
From Tipping and Bishop [1999a], we get the parameter values that maximize the semi-integrated likelihood SIL:

$$\hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_{i\bullet}, \tag{3.9}$$

$$\hat{\mathbf{W}} = \mathbf{U}(\boldsymbol{\Lambda_k} - \hat{\sigma}^2\mathbf{I}_k)^{1/2}\mathbf{R}, \tag{3.10}$$

$$\hat{\sigma}^2 = \frac{\sum_{j=k+1}^p \lambda_j}{p-k}\ ,$$

where the orthogonal matrix $\mathbf{U}$ contains the first k eigenvectors of the sample covariance matrix $\boldsymbol{\Sigma} = \frac{\mathbf{S}}{n}$ with $\mathbf{S} = \sum_{i=1}^n \left(\mathbf{x}_{i\bullet} - \hat{\boldsymbol{\mu}}\right)^T(\mathbf{x}_{i\bullet} - \hat{\boldsymbol{\mu}})$, $\boldsymbol{\Lambda_k} = \begin{pmatrix}\lambda_1 & \dots & 0\\ \vdots & \lambda_l & \vdots\\ 0 & \dots & \lambda_k\end{pmatrix}$ contains the corresponding eigenvalues, and $\mathbf{R}$ is a rotation matrix.

After plugging in the ML estimates (3.9) to the semi-integrated likelihood, we obtain (see also [Minka, 2000]):

$$SIL(\mathbf{X}|\hat{\boldsymbol{\mu}}, \hat{\mathbf{W}}, \hat{\sigma}^2) = (2\pi)^{-pn/2} \left(\Pi_{j=1}^{k} \lambda_j\right)^{-n/2} (\hat{\sigma}^2)^{-n(p-k)/2} \exp(-\frac{pn}{2}). \tag{3.11}$$

Thus, (3.8) gives the Penalized SEmi-integrated Likelihood criterion (PESEL):

$$PESEL_n^{hetero}(k) = -\frac{pn}{2}\log 2\pi - \frac{n}{2}\sum_{j=1}^{k}\log\lambda_j - \frac{n(p-k)}{2}\log(\hat{\sigma}^2) - \frac{pn}{2} - \log(n)\frac{pk - \frac{k(k+1)}{2} + k + p + 1}{2} \tag{3.12}$$

Remark
$PESEL_n^{hetero}$ coincides with BIC for PPCA, as proposed by Minka [2000]. The major difference is that Minka [2000] developed this criterion using a specific prior distribution on $\mathbf{W}$ and noise σ^2, while we show that the approximation is valid for any regular prior on these parameters. Minka [2000] also suggests a second criterion called the Laplace evidence, which depends on the selected prior distribution on $\mathbf{W}$. This idea was further developed by Hoyle [2008], who added additional terms in the approximation, which make it possible to deal with the situation of p increasing proportionally to n. However, the drawback of this approach is that it is highly dependent on the prior on $\mathbf{W}$ and does not solve the problem when $n = const$ and $p \to \infty$, which is the main focus of this article and which is solved by the $PESEL_p$ criterion introduced in the next section.

Now, consider the semi-integrated likelihood in (3.3). As before, we can compute the parameters that maximize the semi-integrated likelihood:

$$\begin{aligned}
\hat{\boldsymbol{\mu}} &= \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_{i\bullet}, \\
\hat{\mathbf{W}} &= \text{first } k \text{ eigenvectors of the covariance matrix}, \\
\hat{\sigma}^2 &= \frac{\sum_{j=k+1}^{p}\lambda_j}{n-k}, \\
\hat{\beta} &= \frac{\sum_{j=1}^{k}\lambda_j}{k} - \hat{\sigma}^2.
\end{aligned} \tag{3.13}$$

(3.14)

Then PESEL is of the form:

$$\begin{aligned}
PESEL_n^{homo} = -\frac{pn}{2}\log 2\pi - \frac{nk}{2}\log\left(\frac{\sum_{j=1}^{k}\lambda_j}{k}\right) - \frac{n(p-k)}{2}\log(\hat{\sigma}^2) \\
-\frac{pn}{2} - \log(n)\frac{pk - \frac{k(k+1)}{2} + p + 1 + 1}{2}.
\end{aligned} \tag{3.15}$$

Remark
As for $PESEL_n^{homo}$, it uses the prior and marginal likelihood from [Rajan and Rayner, 1997]. However, Rajan and Rayner [1997] did not penalize the likelihood according to the number of parameters. Thus their criterion tends to significantly overestimate the number of components, which was confirmed in simulations.

Let us provide some insight into the difference between the two priors and criteria presented in this Section. Observe that in (3.12) there is a term with the sum of logarithms of the first k eigenvalues $\sum_{j=1}^{k} \log \lambda_j$. As in model (3.3), $\mathbf{W}$ is assumed to be orthonormal, all of the k largest eigenvalues have to be equal, and their estimate is $\frac{\sum_{j=1}^{k} \lambda_j}{k}$. Thus, in the corresponding term in (3.15), the sum of the logarithms of the k largest eigenvalues is $k \log \left(\frac{\sum_{j=1}^{k} \lambda_j}{k} \right)$. This observation is yet another justification for referring to formula (3.12) as a *heterogeneous* PESEL and to formula (3.15) as a *homogeneous* PESEL. The other difference is in the penalty term. Due to the assumption of equal eigenvalues in *homogeneous* PESEL, the number of free parameters related to the estimation of eigenvalues is equal to 1, while in the *heterogeneous* PESEL this number is equal to k; since we need to estimate k distinct eigenvalues.

Now, let us return to the choice of prior for k. Observe from (2.51) that

$$\log P(k|X) \approx PESEL(k) + \log P(k) + C(X). \tag{3.16}$$

Thus, in the case when prior distribution $P(k)$ is uniform, maximization of $PESEL$ corresponds to maximization of the posterior probability of k. According to equation (3.16), supplementing $PESEL$ by $\log P(k)$ allows maximization of the approximated posterior probability for any selection of $P(k)$. Moreover, $PESEL$ can be used to approximate the posterior probability for k using the formula

$$P(k|X) \approx \frac{e^{PESEL(k)} P(k)}{\sum_{k=1}^{min(n,p)} e^{PESEL(k)} P(k)} \ ,$$

which allows us to evaluate the uncertainty related to the specific choice of k, as illustrated in Section 3.3.5.

3.1.2 PESEL for n fixed and $p \to \infty$

The asymptotics for $p \to \infty$ and $n = const$, which is of great interest in many applications, has, as far as we know, never been properly discussed. In this setting we need to integrate out $\mathbf{W}$ from (3.1). Then it becomes possible to apply the Laplace approximation. Consider the fixed-effects model expressed in terms of the columns of matrix $\mathbf{X}$ (see 2.50),:

$$\mathbf{x}_{\bullet j} \sim \mathcal{N}(\boldsymbol{\mu} + \mathbf{T}\mathbf{w}_{j\bullet}^T, \sigma^2 \mathbf{I}_n),$$

where $\boldsymbol{\mu} = \boldsymbol{\mu}_{\bullet j} = [\mu_1, \mu_2, \ldots, \mu_n]^T$ satisfies equation (2.55).

Analogously to the previous section, we propose using one of two priors on the rows of the loadings matrix $\mathbf{W}$. The difference between these priors was previously described in Section 3.1.1.

$\mathbf{w}_{j\cdot} \sim \mathcal{N}(0, \mathbf{I}_k)$, which yields:

$$\mathbf{x}_{\bullet j} \sim \mathcal{N}(\boldsymbol{\mu}; \mathbf{T}\mathbf{T}^T + \sigma^2 \mathbf{I}_n). \tag{3.17}$$

$\mathbf{w}_{j\cdot} \sim \frac{1}{\beta}\mathcal{N}(0, \mathbf{I}_k)$ with the constraint that $\mathbf{T}^T\mathbf{T} = \mathbf{I}_k$, which yields:

$$\mathbf{x}_{\bullet j} \sim \mathcal{N}(\boldsymbol{\mu}; \frac{1}{\beta}\mathbf{T}\mathbf{T}^T + \sigma^2 \mathbf{I}_n). \tag{3.18}$$

For both these priors, the marginal distributions for variables $\mathbf{x}_{\bullet j}$ are independent with the covariance matrix depending only on factors $\mathbf{T}$. The related model with random loadings $\mathbf{W}$ and fixed factors $\mathbf{T}$ is in fact interpretable and intuitive. This is the case because when p is much larger than n, we may model our variables being randomly selected from a set of linear combinations of a small number $k \leq n$ of fixed factors. Now, observe that the probabilistic models (3.2) and (3.17) are equivalent up to the transposition of the data $\mathbf{X}$. To see this, consider transposition of the model (2.55) $\mathbf{X}^T - \boldsymbol{\mu}^T = \mathbf{W}\mathbf{T}^T + E^T$. Now, the equivalence follows directly from the symmetry of the prior distributions for the rows of $\mathbf{T}$ and $\mathbf{W}$. The simulation results that we present in Section 3.3.3 confirm that depending on the relationship between n and p, one should choose the model designed for either p or $n \to \infty$.
In case of the first prior (3.17) $PESEL_p$ takes the form:

$$\begin{aligned} PESEL_p^{hetero} =& \log p(\mathbf{X}|\hat{\boldsymbol{\mu}}, \hat{\mathbf{T}}, \hat{\sigma}^2) - \log(p)\, \frac{nk - \frac{k(k+1)}{2} + k + n + 1}{2} \\ =& -\frac{pn}{2}\log(2\pi) - \frac{p}{2}\sum_{j=1}^{k}\log\lambda_j - \frac{p(n-k)}{2}\log(\hat{\sigma}^2) - \frac{pn}{2} \\ & - \log(p)\frac{nk - \frac{k(k+1)}{2} + k + n + 1}{2}, \end{aligned} \tag{3.19}$$

where

$$\begin{aligned} \hat{\boldsymbol{\mu}} &= (\hat{\mu}_1, \ldots, \hat{\mu}_n) = \frac{1}{p}\sum_{j=1}^{p}\mathbf{x}_{\bullet j}, \\ \hat{\mathbf{T}} &= \mathbf{U}(\boldsymbol{\Lambda}_{\boldsymbol{k}} - \hat{\sigma}^2\mathbf{I}_k)^{1/2}\mathbf{R}, \\ \hat{\sigma}^2 &= \frac{\sum_{j=k+1}^{p}\lambda_j}{p-k}\ , \end{aligned}$$

where the orthogonal matrix $\mathbf{U}$ contains the first k eigenvectors of the sample covariance matrix of $\mathbf{X}^T$, $\boldsymbol{\Sigma}_{\boldsymbol{p}} = \frac{\mathbf{S}}{p}$ with $\mathbf{S}_p = \sum_{j=1}^{p}\left(\mathbf{x}_{\bullet j} - \hat{\boldsymbol{\mu}}\right)^T(\mathbf{x}_{\bullet j} - \hat{\boldsymbol{\mu}})$, $\boldsymbol{\Lambda}_{\boldsymbol{k}} = \begin{pmatrix} \lambda_1 & \ldots & 0 \\ \vdots & \lambda_l & \vdots \\ 0 & \ldots & \lambda_k \end{pmatrix}$ contains the corresponding eigenvalues, and $\mathbf{R}$ is a rotation matrix.

In the case of the second prior distribution (3.18), PESEL has the following form:

$$\begin{aligned} PESEL_p^{homo} =& \log p(\mathbf{X}|\hat{\boldsymbol{\mu}}, \hat{\mathbf{T}}, \hat{\sigma}^2, \hat{\beta}) - \log(p)\frac{nk - \frac{k(k+1)}{2} + n + 1 + 1}{2} \\ =& -\frac{pn}{2}\log(2\pi) - \frac{pk}{2}\log\left(\frac{\sum_{j=1}^{k}\lambda_j}{k}\right) - \frac{p(n-k)}{2}\log(\hat{\sigma}^2) - \frac{pn}{2} \\ & - \log(p)\frac{nk - \frac{k(k+1)}{2} + n + 2}{2}\ , \end{aligned} \tag{3.20}$$

where

$$\hat{\beta} = \frac{\sum_{j=1}^{k}\lambda_j}{k} - \hat{\sigma}^2.$$

Remark To evaluate PESEL, the eigenvalues of the covariance matrix need to be computed. To reduce the computational burden, when $n >> p$ we diagonalize the covariance matrix $\mathbf{X}^T\mathbf{X}$ of size $p \times p$, whereas when $n << p$ we diagonalize the inner-product matrix $\mathbf{X}\mathbf{X}^T$ of size $n \times n$. Since we work in regimes where one mode is usually much larger than the other, this procedure is well suited to reduce the computational cost, which is $O(\min(n,p)^3)$. Alternatively, SVD can be used, which is $O(npk)$ and it is not a burden when one of the dimensions is relatively small. In the case of super large matrices, one can still efficiently calculate eigenvalues using random projections (see Witten and Candès [2013]), which paves the way also applying our technique in this "super-large" setup.

3.2 Consistency of PESEL

In this section we shall prove the consistency of PESEL method. We are considering the case when the number of variables p is fixed and the number of observations $n \to \infty$. Because of the transposition argument in the Section 3.1.2, the following result also holds when $p \to \infty$ and n is fixed.
Assume that the data $\mathbf{X}$ comes from normal distribution according to the model (2.53):

$$\mathbf{X}_{n\times p} - \boldsymbol{\mu}_{n\times p} = \mathbf{M}_{n\times p} + E_{n\times p}, \tag{3.21}$$

where

- for each $n \in \mathcal{N}$, matrices $\mathbf{M}_{n\times p} = \mathbf{M}(n)$ and $\boldsymbol{\mu}_{n\times p}$ are deterministic
- $\boldsymbol{\mu}_{n\times p}$ is rank-one matrix in which all rows are identical, i.e. it represents average variable effect.
- $\mathbf{M}_{n\times p}$ is a centered $\sum_{i=1}^{n} M_{i,j} = 0$
- elements of matrix $\mathbf{M}_{n\times p}$ are bounded, $\sup_{n,i\in(1,\ldots,n),j\in(1,\ldots,n)} M_{i,j} < \infty$
- $\mathbf{M}_{n\times p}$ is a low rank matrix $k_0 = \operatorname{rank}(\mathbf{M}_{n\times p})$

$$\forall_n \ \frac{\mathbf{M}_{n\times p}^T \mathbf{M}_{n\times p}}{n} = \mathbf{U}\mathbf{D}_{p\times p}\mathbf{U}^T \tag{3.22}$$

where

$$\mathbf{D}_{p\times p} = \begin{pmatrix} diag[\gamma_i]_{i=1}^{k_0} & 0 \\ 0 & diag[0] \end{pmatrix}$$

with $\gamma_i > 0$ and matrix $\mathbf{U}$ is fixed $p \times p$ matrix. consists of eigenvectors of matrix $\mathbf{M}_{n\times p}^T \mathbf{M}_{n\times p}$

- the noise matrix $E_{n\times p}$ consists of i.i.d. terms $e_{ij} \sim N(0, \sigma^2)$

Theorem 3.2.1 (Consistency Theorem)
Let $\hat{k}_0(n)$ be the PESEL$_n$ estimator of the number of PCA factors for the data matrix $\mathbf{X}_{n\times p}$ drawn according to the probabilistic model (3.21).

Then, for p fixed,

$$\mathbb{P}(\exists_{n_0} \forall_{n>n_0} \ \hat{k}_0(n) = k_0) = 1$$

The idea of the proof is the following. As value of PESEL criterion $\text{PESEL}_n(\mathbf{X}, k)$ for a given number of principal components k depends **only** on a sample covariance matrix $\mathbf{S} = \frac{(\mathbf{X}-\bar{\mathbf{X}})^T(\mathbf{X}-\bar{\mathbf{X}})}{n}$ and not on the data $\mathbf{X}$ directly, we shall treat it as a function of sample covariance matrix. Firstly, we prove that if we substitute sample covariance matrix with its expected value

$$\Sigma_n = \mathbb{E}(\mathbf{S}) = \frac{\mathbf{M}^T_{n\times p}\mathbf{M}_{n\times p}}{n} + diag[\sigma^2]$$

then, as $n \to \infty$, PESEL criterion $\text{PESEL}_n(\Sigma_n, k)$ is consistent. This part comprises two steps: showing that PESEL is increasing for $k < k_0$ and showing that is is decreasing for $k > k_0$. Second element of the proof, is quantifying the difference between Λ_n and $\mathbf{S}$. We focus on the maximum discrepancy between eigenvalues of these two matrices. We prove it to be bounded by the matrix norm of their difference, which goes to 0 at the pace $\frac{\sqrt{\ln\ln n}}{\sqrt{n}}$as n grows to infinity because of the law of iterated logarithm. Finally, we show that for large enough n, such a small perturbation of eigenvalues of Σ_n does not lead to a violation of consistency, thus proving Theorem 3.2.1.

Lemma 3.2.2 *Let Σ_n be expected value of sample covariance matrix of $\mathbf{X}$ and Λ_n be the diagonal matrix of sorted eigenvalues of Σ_n Let $\hat{k}_0(n)$ be the PESEL$_n$ estimator of the number of PCA factors based on the matrix Λ_n, that is the maximizer of the formula for PESEL$_n$:*

$$F(\Sigma_n, n, k) = \frac{n}{2}\cdot \underbrace{-\left[\sum_{j=1}^{k}\ln(\lambda_j) + (p-k)\ln\left(\frac{1}{p-k}\sum_{j=k+1}^{p}\lambda_j\right) + p\ln(2\pi) + p\right]}_{M_{\Sigma_n}(k)} \underbrace{-\ln(n)\frac{pk - \frac{k(k+1)}{2} + k + p + 1}{2}}_{P_{\Sigma_n}(n,k)} \tag{3.23}$$

Then, for p fixed,

$$\lim_{n\to\infty}\hat{k}_0(n) = k_0.$$

Lemma 3.2.3 *PESEL function $F(\Sigma_n, n, k)$ is decreasing in k for $k \geq k_0$*

Proof

For the clarity, we shall use notation $M(k) = M_{\Sigma_n}(k)$ and $P(n,k) = P_{\Sigma_n}(n,k)$. Let λ_i denote eigenvalues of matrix Λ_n. $\lambda_i = \gamma_i + \sigma^2$ because from (3.21):

$$\frac{\mathbf{M}^T_{n\times p}\mathbf{M}_{n\times p}}{n} + \sigma^2\mathbf{I} = \mathbf{U}\mathbf{D}_{p\times p}\mathbf{U}^T + \sigma^2\mathbf{U}\mathbf{I}\mathbf{U}^T = \mathbf{U}\left(\mathbf{D}_{p\times p} + \sigma^2\mathbf{I}\right)\mathbf{U}^T$$

We combine two observations.

1. **When k increases, the penalty**

$$P(n,k) = \ln(n)\frac{pk - \frac{k(k+1)}{2} + k + p + 1}{2}$$

increases. Indeed, we compute $P(n,k+1)-P(n,k)=\frac{\ln n}{2}(p-k)>0$
2.**The non-penalty part of PESEL formula** (3.23)

$$M(k)=-\left[\sum_{j=1}^{k}\ln(\lambda_j)+(p-k)\ln\left(\frac{1}{p-k}\sum_{j=k+1}^{p}\lambda_j\right)+p\ln(2\pi)+p\right]$$

is constant for $k\geq k_0$.
For the clarity, we omit terms in $M(k)$ that do not include k. Observe that λ_l is constant for $l\geq k_0+1$. Denote $\tilde{\lambda}=\lambda_{k_0+1}$. For $k\geq k_0$ we have

$$\begin{aligned}
-M(k)&=\sum_{j=1}^{k}\ln(\lambda_j)+(p-k)\ln\left(\frac{1}{p-k}\sum_{j=k+1}^{p}\lambda_j\right)\\
&=\sum_{j=1}^{k_0}\ln(\lambda_j)+(k-k_0)\ln\tilde{\lambda}+(p-k)\ln\left(\frac{1}{p-k}\sum_{j=k+1}^{p}\tilde{\lambda}\right)\\
&=\sum_{j=1}^{k_0}\ln(\lambda_j)+(k-k_0)\ln\tilde{\lambda}+(p-k)\ln\tilde{\lambda}\\
&=\sum_{j=1}^{k_0}\ln(\lambda_j)+(p-k_0)\ln\tilde{\lambda}
\end{aligned}$$

which is independent of k.

∎

Lemma 3.2.4 *PESEL function $F(\Sigma_n,n,k)$ is increasing in k for $k<k_0$*

Proof
We first prove that $M(k)$ is increasing for $k<k_0$. The idea is to use concavity of logarithm function.
For the simpler notation, let us consider $M(k)-M(k+1)$ thus getting rid of a minus sign.

$$\begin{aligned}
M(k)-M(k+1)&=\left[\ln\lambda_{k+1}+(p-k-1)\ln\frac{\sum_{k+2}^{p}\lambda_j}{p-k-1}-(p-k)\ln\frac{\sum_{k+1}^{p}\lambda_j}{p-k}\right]\\
&=\ln\lambda_{k+1}-\ln\frac{\sum_{k+2}^{p}\lambda_j}{p-k-1}+(p-k)\left[\ln\frac{\sum_{k+2}^{p}\lambda_j}{p-k-1}-\ln\frac{\sum_{k+1}^{p}\lambda_j}{p-k}\right]
\end{aligned}$$

Let us now denote, $a=\lambda_{k+1}$ and $b=\frac{\sum_{k+2}^{p}\lambda_j}{p-k-1}$. Then the above becomes:

$$\ln a-\ln b+(p-k)\left[\ln b-\ln\frac{b(p-k-1)+a}{p-k}\right]$$

Let $a=b+\epsilon$, $\epsilon>0$.

$$\ln(b+\epsilon) - \ln b + (p-k)\left[\ln b - \ln(b+\frac{\epsilon}{p-k})\right] \stackrel{?}{<} 0$$

$$\ln(b+\epsilon) - \ln b \stackrel{?}{<} (p-k)\left[\ln(b+\frac{\epsilon}{p-k}) - \ln b\right]$$

$$\frac{\ln(b+\epsilon) - \ln b}{p-k} \stackrel{?}{<} \ln(b+\frac{\epsilon}{p-k}) - \ln b$$

$$\frac{\ln(b+\epsilon)}{p-k} + (1-\frac{1}{p-k})\ln b \stackrel{?}{<} \ln(b+\frac{\epsilon}{p-k})$$

Which is concavity condition for $x_1 = b$, $x_2 = b+\epsilon$, $\theta = \frac{1}{p-k}$.
However, the crux of the proof of consistency is making sure, that increase in M(k) is larger than $\frac{(p-k)\log n}{n}$ (increase in penalty). Let us further approximate the difference $M(k) - M(k+1)$ using Taylor expansion and bounding by the value of second derivative. This should be larger than $\frac{\ln n}{n}$.
We will use notation as above. And exploit concavity of ln function. We use Taylor expansion at point x_0

$$f(x) = f(x_0) + f'(x_0)(x-x_0) + \frac{f''(x^\star)}{2}(x-x_0)^2,$$

where $x^\star \in (x, x_0)$.
Let $x_0 = \theta x_1 + (1-\theta)x_2$ and take $x = x_1$

$$f(x_1) = f(x_0) + f'(x_0)(1-\theta)(x_1-x_2) + \frac{f''(x_1^\star)}{2}(1-\theta)^2(x_1-x_2)^2 \tag{3.24}$$

Similarly take $x = x_2$

$$f(x_2) = f(x_0) + f'(x_0)\theta(x_2-x_1) + \frac{f''(x_2^\star)}{2}\theta^2(x_2-x_1)^2 \tag{3.25}$$

Now let us multiple (3.24) by θ and (3.25) by $1-\theta$ and sum them up.

$$\begin{aligned}
\theta f(x_1) + (1-\theta)f(x_2) =& \theta\left[f(x_0) + f'(x_0)(1-\theta)(x_1-x_2) + \frac{f''(x_1^\star)}{2}(1-\theta)^2(x_1-x_2)^2\right] + \\
&+ (1-\theta)\left[f(x_0) + f'(x_0)\theta(x_2-x_1) + \frac{f''(x_2^\star)}{2}\theta^2(x_2-x_1)^2\right] \\
=& \theta f(x_0) + f'(x_0)\theta(1-\theta)(x_1-x_2) + \theta\frac{f''(x_1^\star)}{2}(1-\theta)^2(x_1-x_2)^2 + \\
&+ (1-\theta)f(x_0) + f'(x_0)(1-\theta)\theta(x_2-x_1) + (1-\theta)\frac{f''(x_2^\star)}{2}\theta^2(x_2-x_1)^2 \\
=& f(x_0) + \theta\frac{f''(x_1^\star)}{2}(1-\theta)^2(x_1-x_2)^2 + (1-\theta)\frac{f''(x_2^\star)}{2}\theta^2(x_2-x_1)^2 \\
=& f(x_0) + \theta(1-\theta)(x_2-x_1)^2\left[\frac{f''(x_1^\star)}{2}(1-\theta) + \frac{f''(x_2^\star)}{2}\theta\right]
\end{aligned}$$

In our case $f''(x) = -\frac{1}{x^2}$, which means that $\frac{f''(x_i^\star)}{2} < \frac{f''(x_2)}{2}$ because $x_1^\star \in (x_1, x_0) < x_2$ and $x_2^\star \in (x_0, x_2) < x_2$. This gives us:

$$\begin{aligned}\theta f(x_1) + (1-\theta)f(x_2) - f(x_0) &= \theta(1-\theta)(x_2-x_1)^2\left[\frac{f''(x_1^\star)}{2}(1-\theta) + \frac{f''(x_2^\star)}{2}\theta\right] \\ &< \theta(1-\theta)(x_2-x_1)^2\left[\frac{f''(x_2)}{2}(1-\theta) + \frac{f''(x_2)}{2}\theta\right] \\ &= \theta(1-\theta)(x_2-x_1)^2\frac{f''(x_2)}{2}\end{aligned} \tag{3.26}$$

Going back with our notation to $M(k)$,

$$\begin{aligned}x_1 = b &= \frac{\sum_{k+2}^{p}\lambda_j}{p-k-1}, \\ x_2 = b+\epsilon &= a = \lambda_{k+1}, \\ \theta &= 1 - \frac{1}{p-k}\end{aligned}$$

we have

$$\theta f(x_1) + (1-\theta)f(x_2) - f(x_0) = \frac{1}{p-k}[M(k) - M(k+1)]$$

and inequality (3.26) becomes:

$$\begin{aligned}&\theta f(x_1) + (1-\theta)f(x_2) - f(x_0) = \\ &\left(1-\frac{1}{p-k}\right)\ln\left(\frac{\sum_{k+2}^{p}\lambda_j}{p-k-1}\right) + \frac{1}{p-k}\ln(\lambda_{k+1}) - \ln\left((1-\frac{1}{p-k})\frac{\sum_{k+2}^{p}\lambda_j}{p-k-1} + \frac{1}{p-k}\lambda_{k+1}\right) \\ &< \frac{1}{p-k}\left(1-\frac{1}{p-k}\right)\left(\lambda_{k+1} - \frac{\sum_{k+2}^{p}\lambda_j}{p-k-1}\right)^2\frac{-1}{2\lambda_{k+1}^2}\end{aligned}$$

Now we multiple both sides by $p-k$.

$$\begin{aligned}&(p-k-1)\ln\left(\frac{\sum_{k+2}^{p}\lambda_j}{p-k-1}\right) + \ln(\lambda_{k+1}) - (p-k)\ln\left((1-\frac{1}{p-k})\frac{\sum_{k+2}^{p}\lambda_j}{p-k-1} + \frac{1}{p-k}\lambda_{k+1}\right) \\ &< -\left(1-\frac{1}{p-k}\right)\left(\lambda_{k+1} - \frac{\sum_{k+2}^{p}\lambda_j}{p-k-1}\right)^2\frac{1}{2\lambda_{k+1}^2}\end{aligned}$$

So,

$$\begin{aligned}M(k+1) - M(k) &> \left(1-\frac{1}{p-k}\right)\left(\lambda_{k+1} - \frac{\sum_{k+2}^{p}\lambda_j}{p-k-1}\right)^2\frac{1}{2\lambda_{k+1}^2} \\ &= \frac{p-k-1}{p-k}\left(\lambda_{k+1} - \frac{\sum_{k+2}^{p}\lambda_j}{p-k-1}\right)^2\frac{1}{2\lambda_{k+1}^2} \\ &\overset{\text{because } k+1\le k_0}{>} \frac{p-k_0-1}{p-k_0}\left(\lambda_{k+1} - \frac{\sum_{k+2}^{p}\lambda_j}{p-k-1}\right)^2\frac{1}{2\lambda_1^2} \\ &= \frac{p-k_0-1}{p-k_0}c_k(\gamma)^2\frac{1}{2(\gamma_1+\sigma^2)^2} \\ &> \frac{p-k_0-1}{p-k_0}\min_{k<k_0}c_k(\gamma)^2\frac{1}{2(\gamma_1+\sigma^2)^2} = C > 0\end{aligned} \tag{3.27}$$

where $c_k(\gamma) = \gamma_{k+1} - \frac{\sum_{i=k+2}^{k_0}\gamma_i}{p-k-1} > 0$ depends on the eigenvalues of matrix vM γ_i. In the inequality (3.27), we used the fact that $\frac{p-x-1}{p-x}$ is decreasing in x, inequality $\lambda_{k+1} \leq \lambda_1$ and the fact that

$$\left(\lambda_{k+1} - \frac{\sum_{i=k+2}^{p}\lambda_j}{p-k-1}\right) = \gamma_{k+1} + \sigma^2 - \frac{\sum_{i=k+2}^{p}\gamma_i}{p-k-1} - \sigma^2 = c_k(\gamma).$$

Thus we have constant C, which is independent of k and n.
To conclude the proof for expected value of sample covariance matrix, let us observe, that the increase in $\frac{n}{2}M(k)$ is larger than the increase in penalty term $P(n,k+1) - P(n,k) = \frac{\ln n}{2}(p-k) > 0$ when $n \to \infty$,
In fact

$$\frac{n}{2}[M(k+1) - M(k)] \geq \frac{n}{2}C >> \frac{\ln n}{2}(p-k) = P(n,k+1) - P(n,k).$$

This implies that the PESEL estimator function $F(n,k) = \frac{n}{2}M(k) - P(n,k)$ is strictly increasing for $k \leq k_0$ and large enough n.

∎

Proof of lemma 3.2.2
Since, for large enough n, PESEL function $F(n,k)$ is decreasing in k for $k \geq k_0$ and $F(n,k)$ is increasing in k for $k < k_0$ then

$$\lim_{n\to\infty} \arg\max_k F(n,k) = \lim_{n\to\infty} \hat{k}_0(n) = k_0$$

∎

To prove the theorem for sample covariance matrix, we are going to need a bound on difference between eigenvalues of sample covariance and covariance matrices.

Lemma 3.2.5 *There exists $C' > 0$ such that*

$$\text{almost surely } \exists n_0\ \forall n \geq n_0 \quad \|\lambda(\mathbf{S}) - \lambda(\Sigma)\|_\infty \leq C'\frac{\sqrt{2\ln\ln n}}{\sqrt{n}}, \tag{3.28}$$

where $\mathbf{S}$ is sample covariance matrix for data drawn according to model (3.21), Σ is its expected value and function $\lambda(\cdot)$ returns sequence of eigenvalues.

Proof
Observe that

$$\left\|\frac{(\mathbf{X}-\overline{\mathbf{X}})^T(\mathbf{X}-\overline{\mathbf{X}})}{n} - \Sigma\right\|_\infty = \max_{1\leq i,j\leq p}\|\frac{1}{n}(X_{\cdot,i} - \overline{X_{\cdot,i}})^T(X_{\cdot,j} - \overline{X_{\cdot,j}}) - \Sigma_{i,j}\|$$

Because of norm equivalence, we shall now consider ℓ_2 norm and add some constant multiplier. Each element $\frac{1}{n}(X_{\cdot,i} - \overline{X_{\cdot,i}})^T(X_{\cdot,j} - \overline{X_{\cdot,j}}) - \Sigma_{i,j}$ we can rewrite in the following way:

$$\begin{aligned}
&\frac{1}{n}(X_{\cdot,i} - \overline{X_{\cdot,i}})^T(X_{\cdot,j} - \overline{X_{\cdot,j}}) - \Sigma_{i,j} = \\
&\quad = \frac{1}{n}(X_{\cdot,i} - \mu_i)^T(X_{\cdot,j} - \overline{X_{\cdot,j}}) + \frac{1}{n}(\mu_i - \overline{X_{\cdot,i}})^T(X_{\cdot,j} - \overline{X_{\cdot,j}}) - \Sigma_{i,j} \\
&\quad = \frac{1}{n}(X_{\cdot,i} - \mu_i)^T(X_{\cdot,j} - \mu_j) + \frac{1}{n}(X_{\cdot,i} - \mu_i)^T(\mu_j - \overline{X_{\cdot,j}}) \\
&\qquad + \frac{1}{n}(\mu_i - \overline{X_{\cdot,i}})^T(X_{\cdot,j} - \mu_j) + \frac{1}{n}(\mu_i - \overline{X_{\cdot,i}})^T(\mu_j - \overline{X_{\cdot,j}}) - \Sigma_{i,j}
\end{aligned}$$

Taking norm over above and using triangle inequality we get:

$$\begin{aligned}&\left\|\frac{1}{n}(X_{\cdot,i}-\overline{X_{\cdot,i}})^T(X_{\cdot,j}-\overline{X_{\cdot,j}})-\Sigma_{i,j}\right\|< \\ &<\left\|\frac{1}{n}(X_{\cdot,i}-\mu_i)^T(X_{\cdot,j}-\mu_j)-\Sigma_{i,j}\right\|+\left\|\frac{1}{n}(X_{\cdot,i}-\mu_i)^T(\mu_j-\overline{X_{\cdot,j}})\right\| \\ &+\left\|\frac{1}{n}(\mu_i-\overline{X_{\cdot,i}})^T(X_{\cdot,j}-\mu_j)\right\|+\left\|\frac{1}{n}(\mu_i-\overline{X_{\cdot,i}})^T(\mu_j-\overline{X_{\cdot,j}})\right\|\end{aligned} \tag{3.29}$$

Recall that

$$\Sigma_{i,j}=\begin{cases}\dfrac{M_i^TM_j}{n} & \text{if } i\neq j\\ \dfrac{M_i^TM_j}{n}+\sigma^2 & \text{if } i=j\end{cases}$$

Without loss of generality, let us consider first case. Let us now consider first element:

$$\begin{aligned}&\frac{1}{n}(X_{\cdot,i}-\mu_i)^T(X_{\cdot,j}-\mu_j)-\Sigma_{i,j}= \\ =&\frac{1}{n}(X_{\cdot,i}-\mu_i)^T(X_{\cdot,j}-\mu_j)-\frac{M_i^TM_j}{n} \\ =&\frac{1}{n}(X_{\cdot,i}-\mu_i-M_i)^T(X_{\cdot,j}-\mu_j-M_j) \\ &+\frac{1}{n}(X_{\cdot,i}-\mu_i)^TM_j-\frac{M_i^TM_j}{n} \\ &+\frac{1}{n}M_i^T(X_{\cdot,j}-\mu_j)-\frac{M_i^TM_j}{n}\end{aligned}$$

Again we apply norm and triangle inequality and get:

$$\begin{aligned}&\left\|\frac{1}{n}(X_{\cdot,i}-\mu_i)^T(X_{\cdot,j}-\mu_j)-\Sigma_{i,j}\right\|< \\ &\left\|\frac{1}{n}(X_{\cdot,i}-\mu_i-M_i)^T(X_{\cdot,j}-\mu_j-M_j)\right\| \\ &+\left\|\frac{1}{n}(X_{\cdot,i}-\mu_i)^TM_j-\frac{M_i^TM_j}{n}\right\| \\ &+\left\|\frac{1}{n}M_i^T(X_{\cdot,j}-\mu_j)-\frac{M_i^TM_j}{n}\right\|= \\ =&\left\|\frac{1}{n}(X_{\cdot,i}-\mu_i-M_i)^T(X_{\cdot,j}-\mu_j-M_j)\right\| \\ &+\left\|\frac{1}{n}(X_{\cdot,i}-\mu_i-M_i)^TM_j\right\| \\ &+\left\|\frac{1}{n}M_i^T(X_{\cdot,j}-\mu_j-M_j)\right\|\end{aligned} \tag{3.30}$$

Putting above results together we get:

$$\begin{aligned}
&\left\|\frac{1}{n}(X_{\cdot,i}-\overline{X_{\cdot,i}})^T(X_{\cdot,j}-\overline{X_{\cdot,j}})-\Sigma_{i,j}\right\| < \\
&\left\|\frac{1}{n}(X_{\cdot,i}-\mu_i-M_i)^T(X_{\cdot,j}-\mu_j-M_j)\right\| \\
&+\left\|\frac{1}{n}(X_{\cdot,i}-\mu_i-M_i)^TM_j\right\| \\
&+\left\|\frac{1}{n}M_i^T(X_{\cdot,j}-\mu_j-M_j)\right\| \\
&+\left\|\frac{1}{n}(X_{\cdot,i}-\mu_i)^T(\mu_j-\overline{X_{\cdot,j}})\right\| \\
&+\left\|\frac{1}{n}(\mu_i-\overline{X_{\cdot,i}})^T(X_{\cdot,j}-\mu_j)\right\|+\left\|\frac{1}{n}(\mu_i-\overline{X_{\cdot,i}})^T(\mu_j-\overline{X_{\cdot,j}})\right\|
\end{aligned} \tag{3.31}$$

Observe that $X_{l,i}\sim\mathcal{N}(\mu_i+M_{l,i},\sigma^2)$. So,

- $\overline{X_{\cdot,i}}\sim\frac{1}{n}\mathcal{N}(n\mu_i+\sum_l M_{l,j},n\sigma^2)=\mathcal{N}(\mu_i,\frac{\sigma^2}{n})$ because $\mathbf{M}$ is centered
- $X_{\cdot,i}-\mu_i-M_{\cdot,i}\sim\mathcal{N}(0,\sigma^2\mathbf{I})$
- $X_{\cdot,i}-\mu_i\sim\mathcal{N}(M_{\cdot,i},\sigma^2\mathbf{I})$

We shall now bound norms in (3.31).
Let us consider:

$$\left\|\frac{1}{n}(X_{\cdot,i}-\mu_i)^T(\mu_j-\overline{X_{\cdot,j}}\mathbb{1})\right\| \le \overbrace{\left\|\frac{1}{\sqrt{n}}(X_{\cdot,i}-\mu_i)\right\|}^{v_i}\overbrace{\left\|\frac{1}{\sqrt{n}}(\mu_j-\overline{X_{\cdot,j}}\mathbb{1})\right\|}^{w_j}$$

where inequality comes from Cauchy-Schwartz and $w_j=(\mu_j-\overline{X_{\cdot,j}}\mathbb{1})=\frac{1}{n}\sum_l E_{l,j}.$ are i.i.d. $\mathcal{N}(0,\sigma^2)$, so we can apply law of iterated logarithm. For some constant $C>\sigma^2$:

$$\text{almost surely } \exists n_C\ \forall n\ge n_C\quad \frac{1}{n}\sum_l E_{l,j}<C\frac{\sqrt{\ln\ln n}}{\sqrt{n}}$$

Second term v_i we can expanded using the fact that $X_{\cdot,i}-\mu_i=M_{\cdot,i}+E_{\cdot,i}$

$$v_j^2=\frac{1}{n}(M_{\cdot,i}+E_{\cdot,i})^TM_{\cdot,i}+E_{\cdot,i}=\frac{1}{n}\left(M_{\cdot,i}^TM_{\cdot,i}+2M_{\cdot,i}^TE_{\cdot,i}+E_{\cdot,i}^TE_{\cdot,i}\right)$$

Let us denote $\tilde{M}=\frac{M^TM}{n}$. From (3.22) $\tilde{M}_{i,i}=\frac{M_{\cdot,i}^TM_{\cdot,i}}{n}$ is constant. Furthermore $\frac{1}{n}E_{\cdot,i}^TE_{\cdot,i}=\frac{1}{n}\sum_l E_{l,i}^2$ converges a.s. to $\mathbb{E}E_{l,i}^2=\sigma^2$ from law of large numbers. To prove right pace of convergence of term $M_{\cdot,i}^TE_{\cdot,i}$ we are going need a generalized version of Law of iterated algorithm (Theorem 2.2.3). Its assumtions are trivially met for random variables

$$M_{l,i}E_{l,i}\sim\mathcal{N}(0,M_{l,i}^2\sigma^2)$$

as they are Gaussian and $\frac{B_{n+1}}{B_n} = \frac{n+1}{n} \to 1$, where B_n is defined as $B_n = \sum_l M_{l,i}^2\sigma^2 = M_{\cdot,i}^T M_{\cdot,i}\sigma^2 = n\tilde{M}_{i,i}\sigma^2$.
Thus the following holds

$$limsup_{n\to\infty} \sum_{l=1} M_{l,i}^T E_{l,i} = \sqrt{2B_n \log\log B_n} \quad a.s.$$

where

To conclude, v_j goes to some constant a.s. Therefore $v_j w_j$ can be bounded for some constant C' by:

$$\text{almost surely } \exists n_{C'}\ \forall n \geq n_{C'} \quad v_j w_j \leq C' \frac{\sqrt{\ln\ln n}}{\sqrt{n}}$$

Similarly let us consider

$$\left\| \frac{1}{n}(\mu_i - \overline{X_{\cdot,i}})^T(\mu_j - \overline{X_{\cdot,j}}) \right\| \leq \left\| \frac{1}{\sqrt{n}}(\mu_i - \overline{X_{\cdot,i}}) \right\| \left\| \frac{1}{\sqrt{n}}(\mu_j - \overline{X_{\cdot,j}}) \right\|$$

Those two parts are w_i and w_j. Therefore:

$$\text{almost surely } \exists n_{C''}\ \forall n \geq n_{C''} \quad w_i w_j \leq C'' \frac{\ln\ln n}{n}$$

Finally let us consider:

$$\begin{aligned} \frac{1}{n}(X_{\cdot,i} - \mu_i - M_i)^T M_j &= \frac{1}{n}\sum_{l=1}^{n}(X_{l,i} - \mu_i - M_{l,i})M_{l,j} \\ &= \frac{1}{n}\sum_{l=1}^{n} E_{l,j}M_{l,j} \end{aligned}$$

This exactly term we bounded using generalized Law of iterated logarithm.
For the final term $\left\|\frac{1}{n}(X_{\cdot,i} - \mu_i - M_i)^T(X_{\cdot,j} - \mu_j - M_j)\right\|$ we can also apply Law of Iterated Logarithm. For every (i,j) when $n \to \infty$:

$$\text{almost surely } \exists n_{i,j}\ \forall n \geq n_{i,j} \quad \|\frac{1}{n}(X_{\cdot,i} - \mu_i - M_i)^T(X_{\cdot,j} - \mu_j - M_j)\| \leq C_{i,j}^0 \frac{\sqrt{2\ln\ln n}}{\sqrt{n}} \quad (3.32)$$

Because we bounded all the elements in 3.31 by terms that tend to 0 at pace $\frac{\sqrt{2\ln\ln n}}{\sqrt{n}}$ we get that

$$\text{almost surely } \exists n_{i,j}\ \forall n \geq n_{i,j} \quad \|\frac{1}{n}\frac{1}{n}(X_{\cdot,i} - \overline{X_{\cdot,i}})^T(X_{\cdot,j} - \overline{X_{\cdot,j}}) - \Sigma_{i,j}\| \leq C_{i,j}\frac{\sqrt{2\ln\ln n}}{\sqrt{n}}$$

Because we take maximum over finite number of p^2 elements, we also get a bound for the norm almost surely with $\tilde{n_0} = \max_{i,j} n_{i,j}$, $\tilde{C} = \max_{i,j} C_{i,j}$

$$\text{almost surely } \exists \tilde{n_0}\ \forall n \geq \tilde{n_0} \quad \|\frac{1}{n}X^T X - \Sigma\| \leq \tilde{C}\frac{\sqrt{2\ln\ln n}}{\sqrt{n}} \quad (3.33)$$

Inequality (3.28) holds because (3.33) holds and, by [Bai and Silverstein] Theorem A.46(A.7.3), when A, B are symmetric

$$\max_k |\lambda_k(A) - \lambda_k(B)| \leq \|A - B\|,$$

where function $\lambda_k(\cdot)$ denotes k^{th} eigenvalue.

∎

Having consistency for true covariance matrix and bound on perturbation of eigenvalues when dealing with sample covariance matrix, we are ready to prove the main theorem.

Proof of Theorem 3.2.1

Recall notation from lemma 3.2.2 where $F(n,k)$ denotes the value of PESEL criterion for k principal components and n observations, $M(k)$ denotes the value of main part of PESEL criterion while $P(n,k)$ denotes the penalty term.

Because of lemma 3.2.5 eigenvalues of sample covariance matrix $\mathbf{S}$ are approximately equal to:

$$\lambda(Cov(X)) = \lambda(\mathbf{M}^T_{n\times p}\mathbf{M}_{n\times p} + \sigma^2\mathbf{I}) = (\gamma_1 + \sigma^2, \ldots, \gamma_{k_0} + \sigma^2, \sigma^2, \ldots, \sigma^2),$$

where $\gamma_1, \ldots, \gamma_{k_0}$ are the eigenvalues of $\mathbf{M}^T_{n\times p}\mathbf{M}_{n\times p}$ like in the proof of lemma (3.2.2).
If those were equal i.e. $\lambda(\mathbf{S}) = \lambda(\Sigma)$, then lemma 3.2.2 holds. We will use the main ideas from the proof of lemma 3.2.2 to solve the general case $\lambda(\mathbf{S}) \neq \lambda(\Sigma)$.
Let $\epsilon_n = \max_i |\lambda_i(\mathbf{S}) - \lambda_i(\Sigma)|$. From lemma 3.2.5 we have $\lim_n \epsilon_n = 0$ almost surely, so for almost all samplings, there exists n_0 such that if $n \geq n_0$,

$$\epsilon_n < \sigma^2 \text{ and } \epsilon_n < \frac{1}{4}\min_{k\leq k_0-1} c_k(\gamma), \tag{3.34}$$

where $c_k(\gamma) = \gamma_{k+1} - \frac{\sum_{k+2}^p \gamma_i}{p-k-1} > 0$ for $k \leq k_0 - 1$.
We study the sequence $M(k)$. For the simplicity, from now we use notation for $\lambda_j = \lambda_j(\mathbf{S})$.

Case $k \leq k_0 - 1$.
We use the formula (3.27) from the proof of lemma 3.2.2:

$$M(k+1) - M(k) > \frac{p-k_0-1}{p-k_0}\left(\lambda_{k+1} - \frac{\sum_{k+2}^p \lambda_j}{p-k-1}\right)^2 \frac{1}{2\lambda_1^2}$$

We have $\lambda_i \in [\gamma_i + \sigma^2 - \epsilon_n, \gamma_i + \sigma^2 + \epsilon_n]$, so

$$\left(\lambda_{k+1} - \frac{\sum_{k+2}^p \lambda_j}{p-k-1}\right) \geq \gamma_{k+1} + \sigma^2 - \epsilon_n - \frac{\sum_{k+2}^p(\gamma_i + \sigma^2 + \epsilon_n)}{p-k-1} = c_k(\gamma) - 2\epsilon_n \geq \min_{k\leq k_0-1} c_k(\gamma) - 2\epsilon_n > 0$$

Putting above two together we get that, almost surely, there exists n_0 such that for all $n \geq n_0$ and $k = 1, \ldots, k_0 - 1$,

$$M(k+1) - M(k) > \frac{p-k_0-1}{p-k_0}(\min_{k\leq k_0-1} c_k(\gamma) - 2\epsilon_n)^2 \frac{1}{2(\gamma_1 + \sigma^2 + \epsilon_n)^2} > \frac{C'}{2}\min_{k\leq k_0-1} c_k(\gamma) > C > 0$$

where C, C' are constants independent of k and n. It follows that for large enough n

$$\frac{n}{2}[M(k+1) - M(k)] \geq \frac{n}{2}C >> \frac{\ln n}{2}(p-k) = P(n,k+1) - P(n,k).$$

This implies that the PESEL estimator function $F(n,k) = \frac{n}{2}M(k) - P(n,k)$ is strictly increasing for $k \leq k_0$.

Case $k \geq k_0$. Recall that this case was very simple for the deterministic matrix case, as $M(k)$ was constant for $k \geq k_0$.

By Lemma 3.2.5 we have that, for almost all samplings, there exists n_0 such that if $n \geq n_0$,

$$\epsilon_n \leq C\frac{\sqrt{2\ln\ln n}}{\sqrt{n}} \text{ and } \epsilon_n < \frac{1}{2}\sigma^2 \tag{3.35}$$

We apply the formula (3.26) and as before, we set

$$\begin{aligned} x_1 = b &= \frac{\sum_{k+2}^{p}\lambda_j}{p-k-1} \\ x_2 = a &= \lambda_{k+1} \\ \theta &= 1 - \frac{1}{p-k} \end{aligned}$$

which makes left hand side of (3.26) equal:

$$\theta f(x_1) + (1-\theta)f(x_2) - f(x_0) = \frac{1}{p-k}[M(k) - M(k+1)]$$

Multiplying by -1 and substituting values in (3.26) yields:

$$\begin{aligned} M(k+1) - M(k) &\leq \left(1 - \frac{1}{p-k}\right)\left(\lambda_{k+1} - \frac{\sum_{k+2}^{p}\lambda_j}{p-k-1}\right)^2 \frac{1}{2b^2} \leq (\lambda_{k+1} - b)^2\frac{1}{2b^2} \\ &\leq (|\lambda_{k+1} - \sigma^2| + |\sigma^2 - b|)^2\frac{1}{2b^2} \\ &\leq (|\lambda_{k+1} - \sigma^2| + \frac{\sum_{k+2}^{p}|\sigma^2 - \lambda_j|}{p-k-1})^2\frac{1}{2b^2} \\ &\leq 4\epsilon_n^2\frac{1}{2(\sigma^2-\epsilon_n)^2} \leq C^2\frac{2\ln\ln n}{n}\frac{4}{2\sigma^4} = C'\frac{\ln\ln n}{n} \end{aligned}$$

and consequently

$$\frac{n}{2}\left[M(k+1) - M(k)\right] \leq C''\ln\ln n$$

Recall that PESEL criterion equals $F(n,k) = \frac{n}{2}M(k) - P(n,k)$. So The increase of $\frac{n}{2}M(k)$ is smaller than the rate $\ln\ln n$, while the increase of penalty is $P(n,k+1) - P(n,k) = \frac{\ln n}{2}(p-k)$ is of rate $\ln n$. Consequently, there exists n_1 that for $n > n_1$, the PESEL estimator function is strictly decreasing for $k \geq k_0$ with probability 1

.

We saw in the first part of the proof that the PESEL estimator function $F(n,k)$ is strictly increasing for $k \leq k_0$, for n big enough. It implies that with probability 1, $\exists n_2$ such that for $n > n_2$ $\hat{k}_0(n) = k_0$. ∎

3.3 Simulation study

We tested the performance of various methods of model selection by comparing the distributions of the inferred dimensionality for data drawn from a known model. Firstly, we aimed to verify how different heterogeneous and homogeneous PESELs are in practice. Secondly, we ask how crucial is the assumption of particular asymptotics, *i.e.* how much better can we do by using $PESEL_p$ when the number of variables exceeds the number of observations. Thirdly, we focused on how robust PESEL is in comparison to state-of-the-art approaches.

3.3.1 Methods

We present results of simulations of seven methods for the estimation of the number of PCs. Three of them have already been described in this paper:

- *Heterogeneous* PESEL for $n >> p$, $PESEL_n^{hetero}$ defined in formula (3.12) and equivalent to BIC for the PPCA model proposed by Minka [2000].
- *Heterogeneous* PESEL for $p >> n$, $PESEL_p^{hetero}$ defined in formula (3.19).
- *Homogeneous* PESEL for $p >> n$, $PESEL_p^{homo}$ defined in formula (3.20).

We compare these three criteria to four state-of-the-art methods:

- Laplace evidence [Minka, 2000, eq. 76], which can be viewed as an extension of $PESEL_n^{hetero}$, as it contains more terms from the Laplace approximation. Since Minka [2000] used a specific non-informative prior distribution on the elements of SVD decomposition of the matrix $\mathbf{W}$ and the variance of the noise σ^2, Laplace evidence depends on that choice and is less general than $PESEL$.
- Generalized Cross-Validation [Josse and Husson, 2012], which, according to the simulation study presented in [Josse and Husson, 2012], performs very well in comparison to many other up-to-date methods for estimating the number of principal components. We used the implementation from the R package FactoMineR [Husson et al., 2014].
- CSV [Choi et al., 2014], which is an exact distribution-based method for testing a hypothesis about the number of principal components. We used our own implementation in MATLAB, since the authors did not provide the code for CSV. In the simulation study, we experienced numerical difficulties with computing the multidimensional integrals that are part of the test statistic. This was observed for a moderate increase in either the number of variables or the signal to noise ratio (defined thereafter). CSV provides an exact test for the number of principal components when the variance of the noise σ^2 is known. In the case when σ^2 needs to be estimated, CSV no longer guarantees control of the type I error. To compare CSV with other methods which do not require knowledge of σ, we followed the suggestion made by Choi et al. [2014] and estimated σ^2 by cross-validation using the softImpute R package [Hastie and Mazumder, 2015].
- The method proposed in [Passemier et al., 2015], which uses the random matrix theory to estimate the variance of the noise. This enhanced estimator is then applied to choose the number of principal components using Stein's unbiased risk estimator (SURE) or the determination criterion of Bai and Ng [2002]. This method is developed in the asymptotic

setting where both n and p diverge to infinity, and $n/p \to \gamma > 0$. The implementation of this method is available on the author's webpage. In the results of the simulations we shall refer to this method as Passemier. We use the version of Passemier based on the determination criterion, since the software for SURE requires $n > p$, .

Apart from both versions of $PESEL_p$, all of the methods are based on decomposition of the standard covariance matrix, which implicitly assumes a model with independent rows and centers the data by subtracting the column means.

3.3.2 Simulations

In the simulations, we compared performance of analyzed methods for various numbers of variables in the data set, varying from 50 to 2000, the number of observations equal to 50, 100 or 2000, and the signal to noise ratios (SNR) in the range [0.25; 8]. By **SNR** we mean the ratio between the l_2 norm of the columns of the signal matrix $\mathbf{M}$ and the variance of the noise. In the simulations, we standardized the columns of the signal matrix $\mathbf{M}$ to have a zero mean and a unit l_2 norm, and so the SNR is given by:

$$\text{SNR} = \frac{1}{\sigma^2},$$

where σ^2 is the variance of the noise (as in (2.53)). Naturally, when the number of variables grows, the combined signal from all the variables is relatively stronger, since all these variables are spanned by the same few factors and combined information allows more accurate estimation of the number of factors. Therefore, we expect that the performance of any statistical method should become more accurate when p increases. This intuition is backed up by the simulation results.

We studied the following scenarios:

Scenario 1. In the first scenario we verified how different the criteria $PESEL_n^{hetero}$ (3.12) and $PESEL_n^{homo}$ (3.15) are in practice. In the first scheme we set all the non-zero singular values equal to each other:

Algorithm 4 Simulation scheme for a signal matrix with equal singular values

Input: Number of observations n, number of variables p, number of PCs k, SNR

1: Each entry of the matrix $\mathbf{M}$ is drawn from the standard normal distribution, $m_{ij} \sim N(0,1)$.

2: **while** all singular values in the normalized matrix $\mathbf{M}$ are not equal to each other **do**

3: Perform SVD of the matrix $\mathbf{M} = \mathbf{ULV}^T$.

4: Set all of the first k singular values from $\mathbf{L}$ equal to their mean and the rest of the singular values to 0.

$$\tilde{l_i} := \frac{1}{k}\sum_{j=1}^{k} l_j, \quad i = 1, \ldots, k,$$
$$\tilde{\mathbf{U}} := \mathbf{U}[\cdot, 1:k]$$
$$\tilde{\mathbf{V}} := \mathbf{V}[\cdot, 1:k],$$

where l_j is the j-th element on the diagonal of $\mathbf{L}$.

5: Set $\mathbf{M} := \tilde{\mathbf{U}}\tilde{\mathbf{L}}\tilde{\mathbf{V}}^T$.

6: Standardize $\mathbf{M}$ so that each column has a zero mean and a unit l_2 norm.

7: **end while**

8: $x_{i,j} := m_{i,j} + \mathcal{N}(0, \frac{1}{SNR})$

The reason for the **while** loop is that after standardization the eigenvalues might no longer be equal. Therefore, we need several steps to obtain the matrix $\mathbf{M}$ which has all eigenvalues equal and at the same time it has standardized columns.

Scenario 2. The second scheme is analogous, but this time we make the non-zero singular values decrease exponentially:

Algorithm 5 Simulation scheme for a signal matrix with exponentially decreasing singular values

Input: Number of observations n, number of variables p, number of PCs k, SNR

1: Each entry of the matrix $\mathbf{M}$ is drawn from the standard normal distribution, $m_{ij} \sim N(0,1)$

2: Perform SVD of the matrix $\mathbf{M} = \mathbf{ULV}^T$.

3: Set all of the singular values of order greater than k to 0, and the largest k to:

$$\tilde{l_i} := C2^{-i}, \quad i = 1, \ldots, k$$
$$\tilde{\mathbf{U}} := \mathbf{U}[\cdot, 1:k],$$
$$\tilde{\mathbf{V}} := \mathbf{V}[\cdot, 1:k],$$

where l_j is the j-th element on the diagonal of $\mathbf{L}$ and $C = (\sum_{j=1}^{k} l_j)/(\sum_{i=1}^{k} 2^{-i})$ is a normalizing constant.

4: Set $\mathbf{M} := \tilde{U}\tilde{L}\tilde{V}^T$

5: Standardize $\mathbf{M}$ so that each column has a zero mean and a unit l_2 norm.

6: $x_{i,j} := m_{i,j} + \mathcal{N}(0, \frac{1}{SNR})$

Scenario 3. The data are generated according to the fixed effect probabilistic model (2.55). Both scores $\mathbf{T}$ and coefficients $\mathbf{W}$ are drawn once from the multivariate normal distribution: $\mathbf{t}_{i\bullet} \sim \mathcal{N}(0, \mathbf{I})$, $\mathbf{w}_{j\bullet} \sim \mathcal{N}(0, \mathbf{I})$. The signal matrix is calculated as $\tilde{\mathbf{M}} := \mathbf{T}\mathbf{W}^T$ and standardized so that each column has a zero mean and a unit l_2 norm. In each iteration of the experiment, a random noise is added to the signal matrix $\mathbf{M}$:

$$x_{i,j} = m_{i,j} + \epsilon_{ij} \quad i = 1, \ldots, n \ , \ j = 1, \ldots, p \tag{3.36}$$
$$\epsilon_{ij} \sim \mathcal{N}(0, \frac{1}{SNR}).$$

Scenario 4. The data are generated as in Scenario 3. However, noise is drawn from the rescaled Student distribution with three degrees of freedom, $\epsilon_{ij} \sim \frac{1}{SNR}\sqrt{\frac{1}{3}}\, t(3)$.

Scenario 5. The data are generated as in Scenario 3. However, noise is drawn from the rescaled log-normal distribution with parameters $\mu = 2$ and $\sigma^2 = 1.2$, $\epsilon_{ij} \sim \frac{1}{SNR}\frac{1}{\sqrt{(e^{\sigma^2}-1)e^{2\mu+\sigma^2}}}\, ln\mathcal{N}(\mu, \sigma^2)$.

Scenario 6. The data are generated as in Scenario 3. However, a number of surplus noisy variables $z_i \sim \mathcal{N}(0, \mathbf{I})$, $i = 1, \ldots, p/2$ is added to the data. $\mathbf{X}_{n\times\frac{3}{2}p} = \left[\mathbf{M}_{n\times p} + E_{n\times p} \,|\, \mathbf{Z}_{n\times p/2}\right]$. An example of when such a violation of our assumptions could occur is when PCA is used in an iterative procedure for clustering variables. It might happen that some of the variables are falsely classified, yet we would still like to recover the true dimensionality of a given cluster.

We replicated each simulation scenario 100 times to get a reliable comparison between the methods.

3.3.3 Results

In the following sections, we present only some selected, representative, simulation results. The true number of principal components is $k = 5$. The results for the number of components equal to 2 or 10 were similar, and therefore are not reported in this paper. We also simulated the data from a random effects model. Factors and coefficients were drawn from normal, heavy-tailed (student), skewed (exponential) or uniform distributions. The qualitative conclusions were also consistent with the simulation results presented in this paper.

When using $k = 5$ we reduced the computational burden of the simulation study by restricting our methods to search only for the dimensions from the set $\{k_{min} = 1, \ldots, k_{max} = 10\}$.

Comparison between $PESEL^{homo}$ and $PESEL^{hetero}$

It can be seen in Figure 3.1 that the difference in performance between the two PESEL criteria backs up the remarks made in Section 3.1.1. $PESEL^{homo}$, which assumes the equality of singular values, performs consistently better when the data are simulated in accordance with this assumption. Conversely, when singular values are substantially different from each other, $PESEL_n^{hetero}$ gets has an edge and the difference between the methods is larger. Since in our simulation study these two criteria performed comparably, in the remainder of the paper we report only results for $PESEL^{hetero}$. Results for $PESEL^{homo}$ are nevertheless available in a supplementary file.

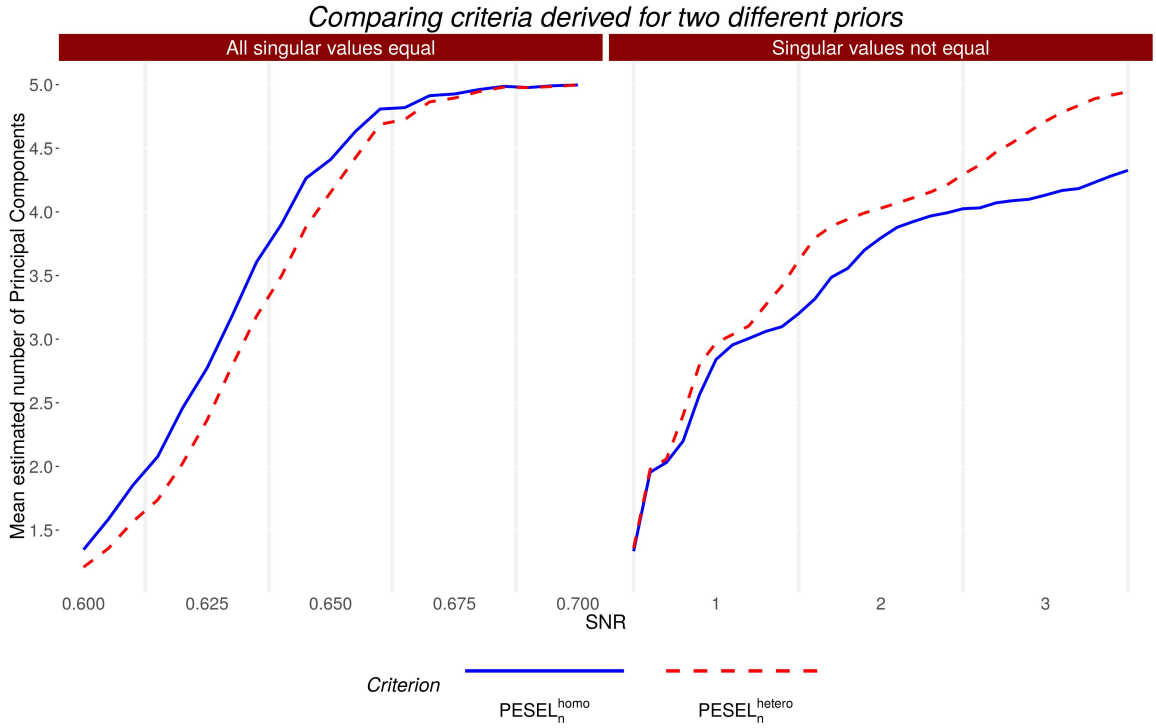


Figure 3.1: Comparison of performance for $PESEL_n^{hetero}$ (3.12) and $PESEL_n^{homo}$ (3.15). Data are simulated using scenarios 1 (left) and 2 (right). The number of variables is 50, number of observations is 100. The true number of PCs is 5. When the singular values are equal, then *homogenous* PESEL has an edge over *heterogenous* PESEL and vice versa.

Impact of the number of the ratio n/p

Figure 3.2 illustrates the performance of various methods when the number of observations n is either very large or very small compared to the number of variables p. The results are not surprising, as the methods that assume asymptotics in n work better when n is large and *vice versa*. Note that probabilistic methods outperform GCV when the $\frac{p}{n}$ ratio is in accordance with their underlying asymptotics. In particular, $PESEL_p$ is superior to all the other approaches when $p >> n$. In the case of $n >> p$, we observe superior performance of the criterion of [Minka, 2000] based on an extended version of Laplace approximation.

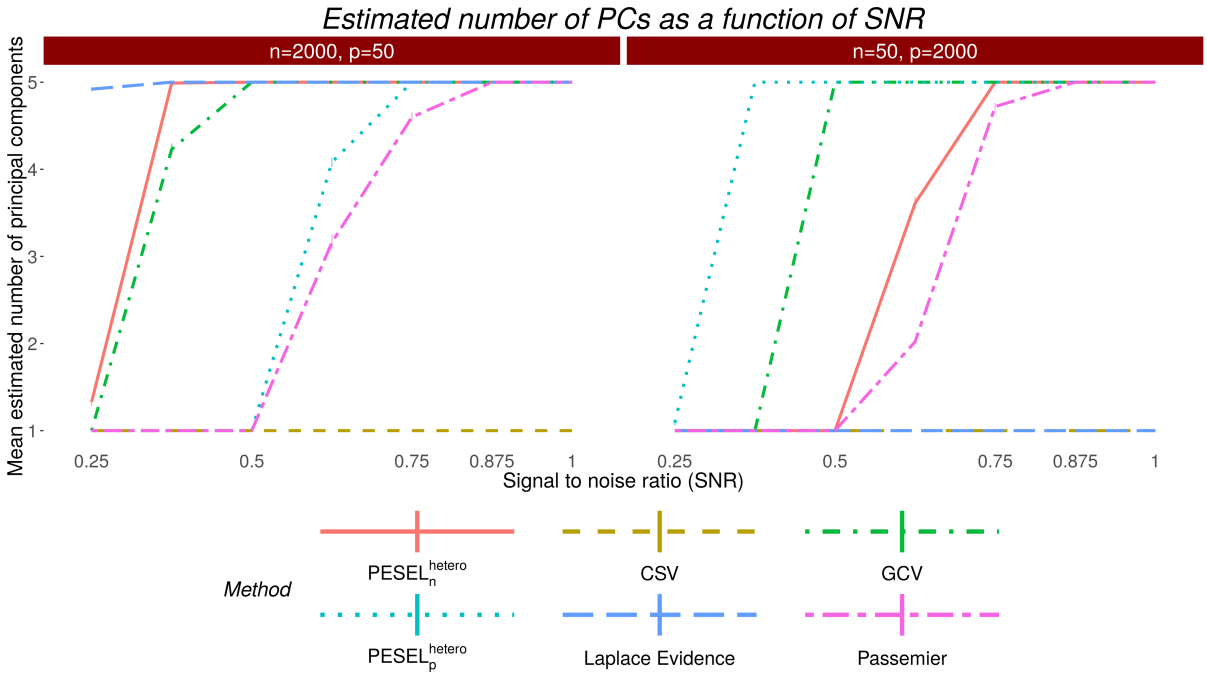


Figure 3.2: Data generated according to Scenario 2. The true number of components is 5. The results are for $n = 2000$, $p = 50$ and $n = 50$, $p = 2000$.

Figure 3.3 illustrates the situation when the data are drawn according to Scenario 3. On the right panel, we retrieve the same concusions as the ones given previously. The left panel corresponds to a case where the number of variables and observations are more balanced. In the case of $p \sim n$ ($n = 100$, $p = 150$), both $PESEL_n^{hetero}$ and $PESEL_p^{hetero}$ methods are outperformed by GCV. This is not surprising, because neither is designed for such cases. The GCV method, although not always the best, often provides fair results in many settings. Passemier's discriminant criterion is inferior to both PESELs. Laplace evidence performs poorly when the number of variables is large compared to the number of observations. CSV works well with weak signals and a small number of variables. However, when either grows, it encounters the numerical problems described in Section 3.3.1.

Robustness

As mentioned in the introduction, the main motivation for testing robustness is when PCA is used as an auxiliary technique. In such a case it might have to deal with data with excessive

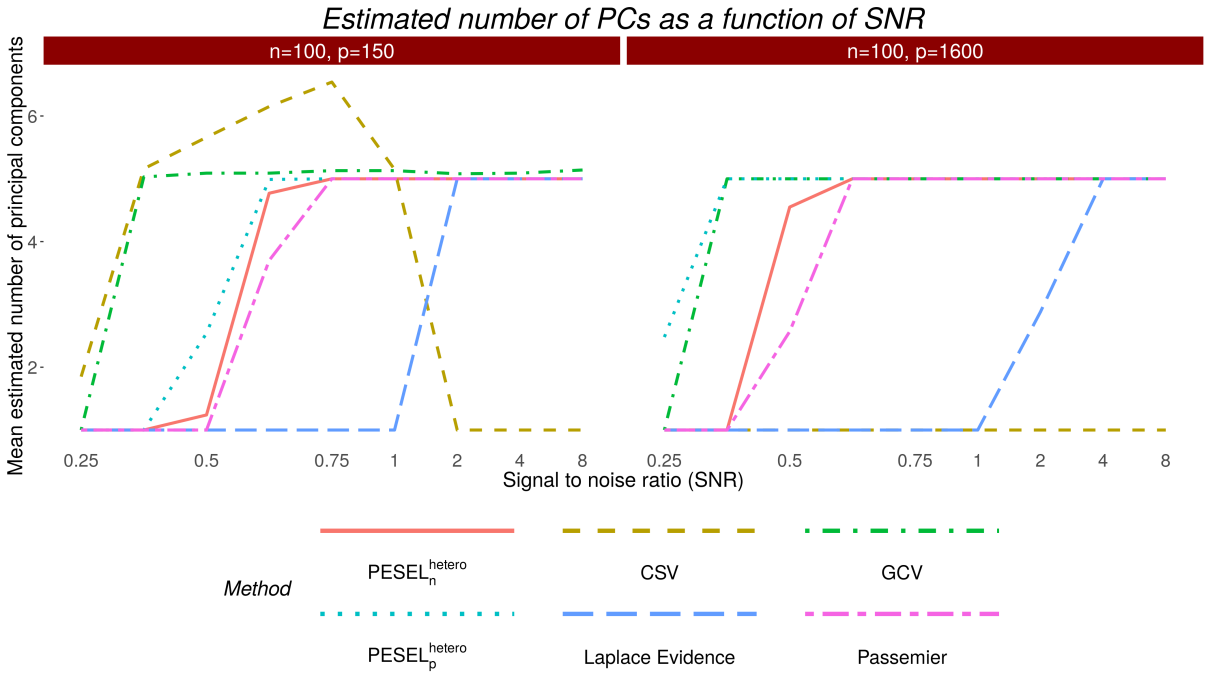


Figure 3.3: Data generated according to Scenario 3. The true number of components is 5. The numbers of variables are 150 and 1600, respectively. The number of observations is constant and equal to 100.

noise. We report results for three kinds of violations of the assumed probabilistic model, previously described in Section 3.3.2. Specifically, Figures 3.4, 3.5, 3.6 and 3.7 illustrate the performance of various methods for the data generated according to Scenarios 4, 5 and 6 (twice), respectively.

For the clarity of plots, in Figures 3.4-3.7 we selected only 4 methods for a detailed comparison. The graphs illustrate the frequency of selection of various numbers of principal components as a function of the signal to noise ratio for two different dimensions of the data set: $n = 100,\ p = 150$ and $n = 100,\ p = 800$.

We observe that all the represented methods deal quite well with the symmetric Student noise (Scenario 4). Here GCV performs very well when the signal is weak or moderate. This however comes at the price of overestimation the number of components when the signal is very strong. Passemier performs opposite. It substantially underestimates the number of principal components when the signal is weak or moderate and performs well when the signal is very strong. $PESEL_p^{hetero}$ takes the place between these two approaches. It slightly underestimates the number of principal components when the signal is weak and does not overestimate when the signal is strong.

Scenario 5, with stronly skewed log-normal noise, turns out to be much more difficult. When $n = 100$ and $p = 150$, GCV substantially overestimates the number of principal components, even for relatively weak signals. Passemier is rather unstable. It overestimates the number of principal components when the signal is weak and substantially underestimates when the signal is strong. Compared to these two methods $PESEL$ performs surprisingly well. It accurately estimates the number of PCs when the signal is weak and moderate and only slightly overestimates when the signal is strong. When p in increased to 800, the performance of GCV

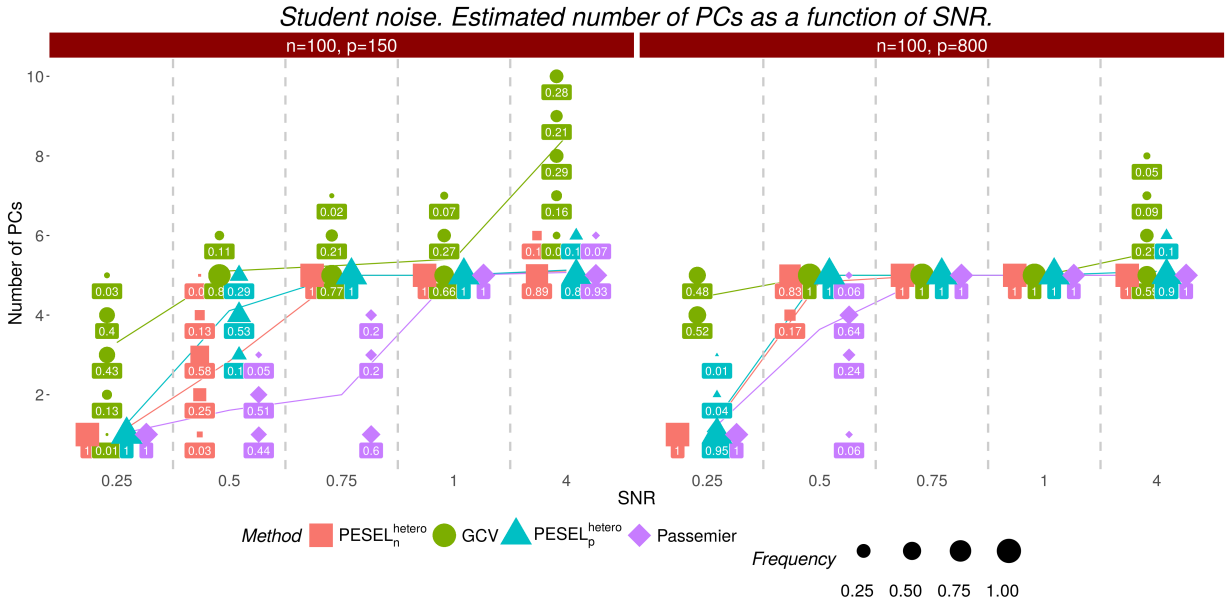


Figure 3.4: Data drawn according to Scenario 4 (noise from a Student distribution). The true number of components is 5. The size of each symbol is proportional to the particular frequency of a result. The lines represent the mean of the estimated numbers of Principal Components. The numbers of variables are 150 and 800. The number of observations is constant and equal to 100.

substantially improves, while Passemier deteriorates completely and is not able to pick any signal.

Scenario 6, with additional noisy variables, yields results similar to Scenario 4. When p is comparable to n, PESEL is inferior to GCV when signal is weak or moderate, but does not overestimate the number of principal components when either number of variables or SNR grows. Passemier is systematically inferior to PESEL.

As for the methods not included in the plots, $PESEL_p^{homo}$ behaves comparably to $PESEL_p^{hetero}$, but is slightly more robust agains the deviations from the assumptions of the probabilistic model when the signal is strong (results in the supplementary file). Laplace evidence proves to be least robust, as it has a tendency to underestimate the number of PCs when a probabilistic model is violated. Laplace evidence is also highly dependent on the assumed asymptotics, *i.e.* $n >> p$. For CSV, when the signal becomes stronger or the number of variables increases, it is increasingly difficult to compute any of the multidimensional integrals this method requires. As a result, we did not manage to use this method to estimate the number of PCs under our simulation scenarios.

3.3.4 Summary of simulation results

All in all, the performance of $PESEL_p$ is competitive with up-to-date methods. Note that GCV is a serious competitor for data simulated according to Scenario 3. However, when p is much larger than n, which is our main focus, $PESEL_p$ is better. Similar conclusions are drawn from a robustness study, despite the fact that $PESEL_p$ was derived under specific probabilistic assumptions. Specifically, when the number of variables is moderate and the signal is strong PESEL is less prone to overfitting than GCV.

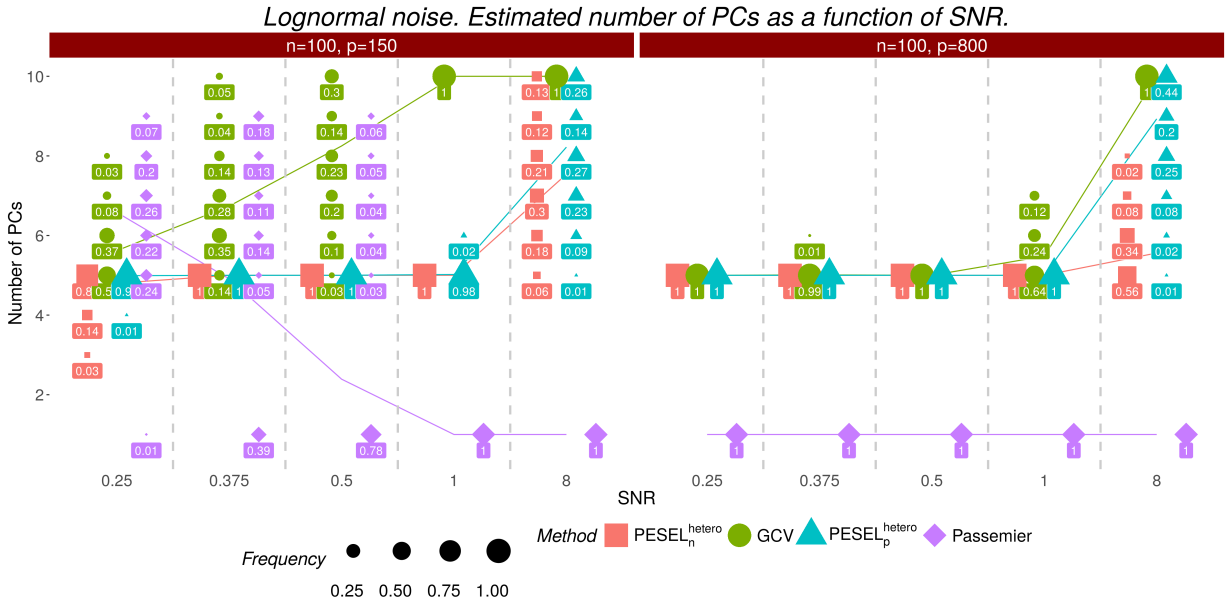


Figure 3.5: Data drawn according to Scenario 5 (noise from the log-normal distribution) with parameters $\mu = 2$, $\sigma^2 = 1.2$. The true number of components is 5. The size of each symbol is proportional to the frequency of the particular result. The lines represent the mean of the estimated numbers of Principal Components. The numbers of variables are 150 and 800. The number of observations is constant and equal to 100.

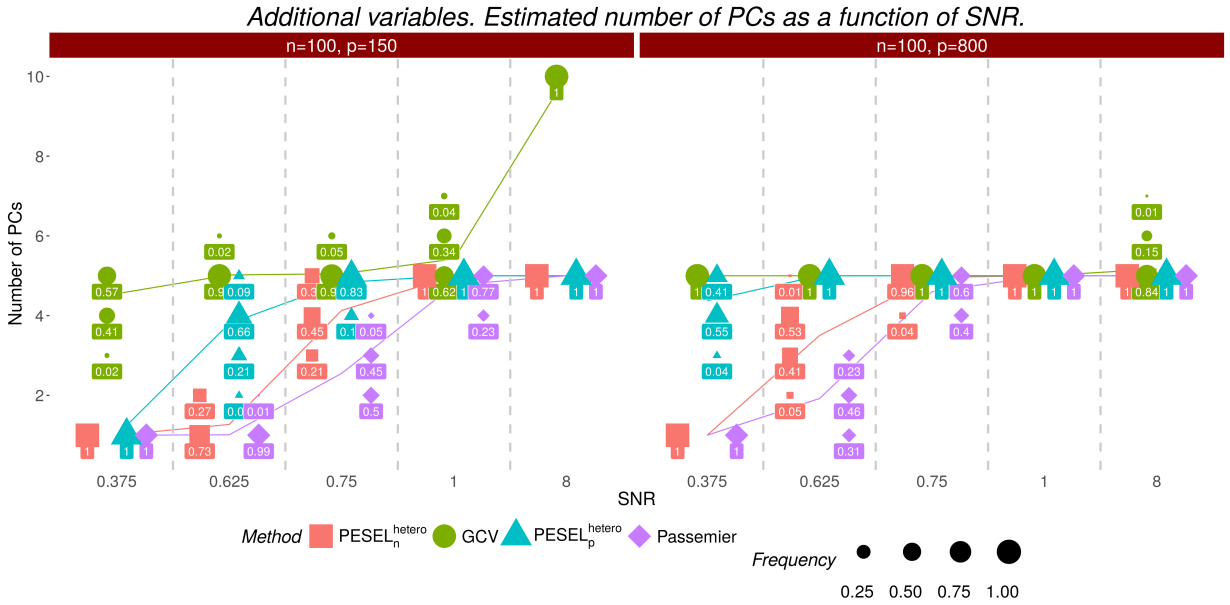


Figure 3.6: Data drawn according to the Scenario 6 (surplus noisy variables). The true number of components is 5. The size of each symbol is proportional to the frequency of the particular result. The lines represent the mean of the estimated numbers of Principal Components. The numbers of variables are 150 and 800. The number of observations is constant and equal to 100.

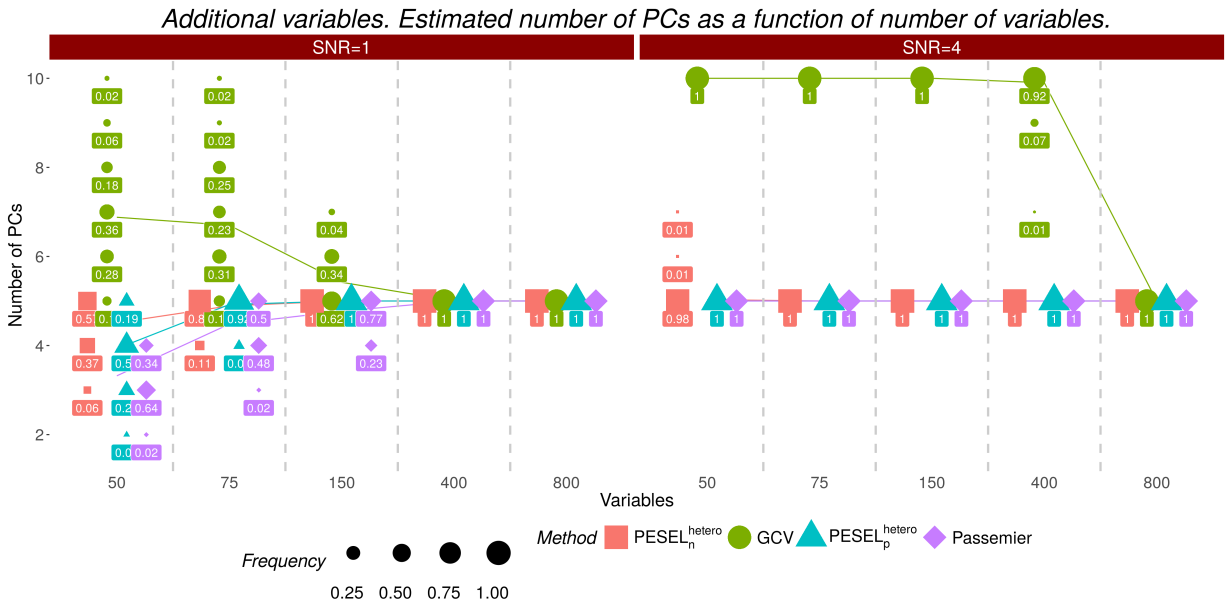


Figure 3.7: Data drawn according to Scenario 6 (surplus noisy variables). The true number of components is 5. The size of each symbol is proportional to the frequency of the particular result. The lines represent the mean of the estimated numbers of Principal Components. The signal to noise ratios take values 1 and 4. The number of variables varies from 50 to 800. The number of observations is constant and equal to 100.

3.3.5 Data analysis

We illustrate our method based on the "mice" data which come from the Genetics Department of the University of Agronomy Agrocampus in France. This is an experiment with 40 mice of 2 genotypes (wild, PPARα-deficient). In the field of molecular biology, peroxisome proliferator-activated receptors (PPARs) are a group of nuclear receptor proteins that function as transcription factors regulating the expression of genes. PPARs play essential roles in the regulation of cellular differentiation, development, and metabolism (carbohydrate, lipid, and protein) in higher organisms. PPARs are expressed in the liver, kidney, heart, muscle, adipose tissue, and others. The mice were subject to 5 diets: dha (a diet rich in fatty acids of the Omega 3 family and particularly docosahexaenoic acid [DHA], based on fish oil), efad (Essential Fatty Acid Deficient: a diet based on saturated fatty acids only, made from hydrogenated coconut oil), lin (a diet rich in Omega 3, made from linseed oil), ref (a regime with seven times more Omega 6 than Omega 3), tsol (a diet rich in Omega 6, based on sunflower oil). At the end of the diet periods, the genes were analysed using DNA chips, and the expression of 120 genes was read for all the mice. The aim of the study is to see whether genes are expressed differently depending on the level of stress.

We used GCV and $PESEL_p^{hetero}$ to identify the number of principal components in gene expression data. Figure 3.8 provides both the PESEL criterion and the associated posterior probabilities calculated with equation (3.16) and using the non-informative, uniform prior on k. The second plot is very informative and suggests that apart from the dimension 5, for which PESEL obtains a maximum value, the dimensions 6 and 7 are also very likely.

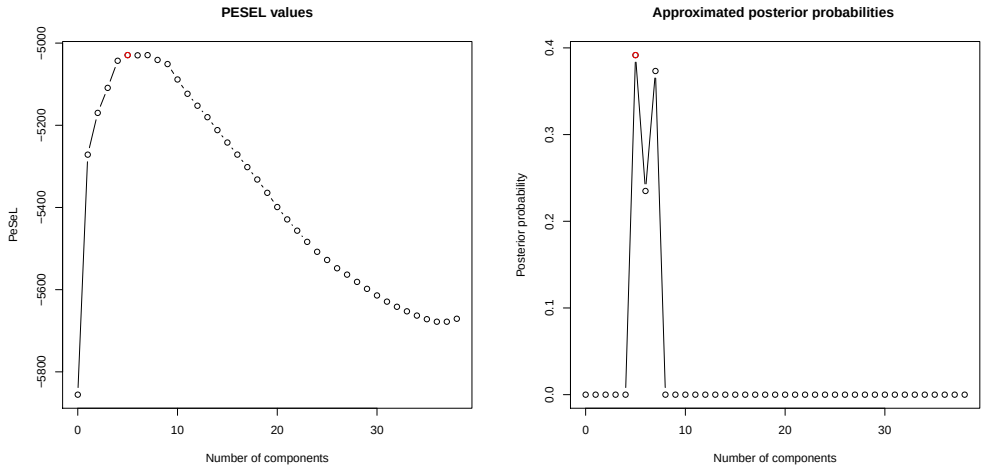


Figure 3.8: Left: Values of $PESEL_p^{hetero}$ for various k, right: approximated posterior probabilities of the number of principal components.

GCV returns 12 dimensions which seems less likely, since our mice are differentiated only by 2 genotypes and 5 diets and we may expect at most 9 dimensions to represent the between-class variability.

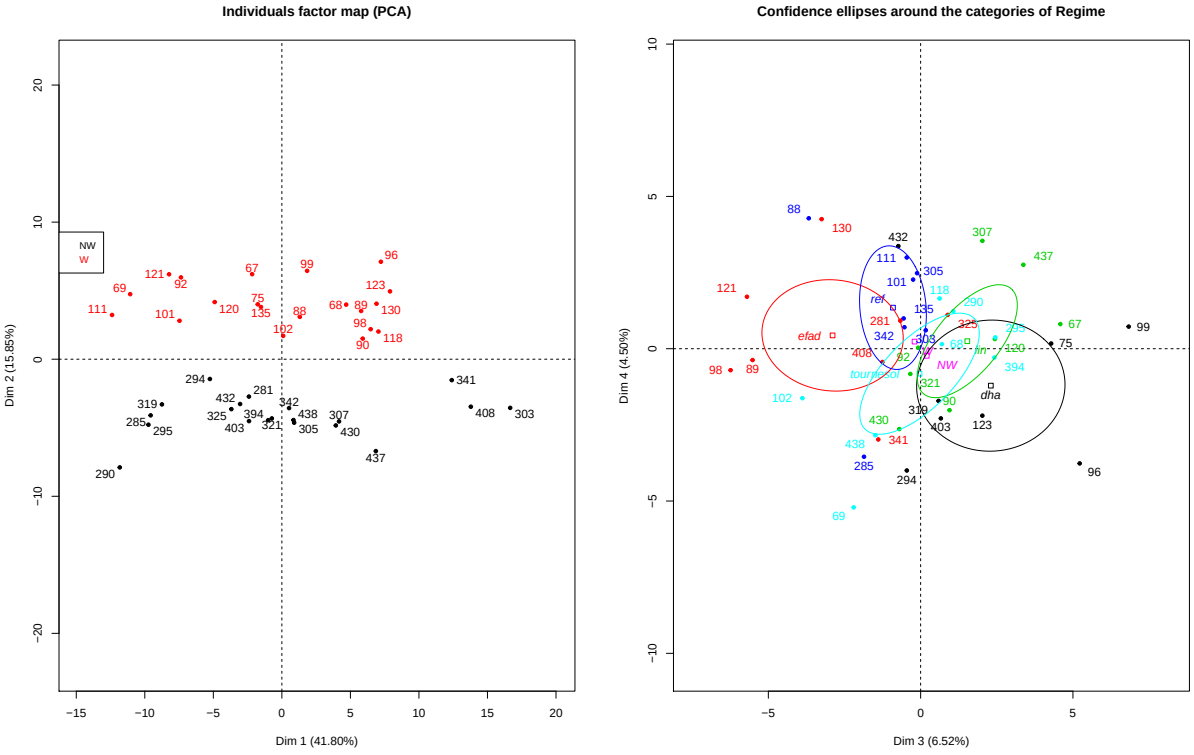


Figure 3.9: Left: scores of the mice on dimensions 1 and 2, mice are coloured according to their genotype - right: dimensions 3 and 4; mice are coloured according to their diet with associated confidence ellipses.

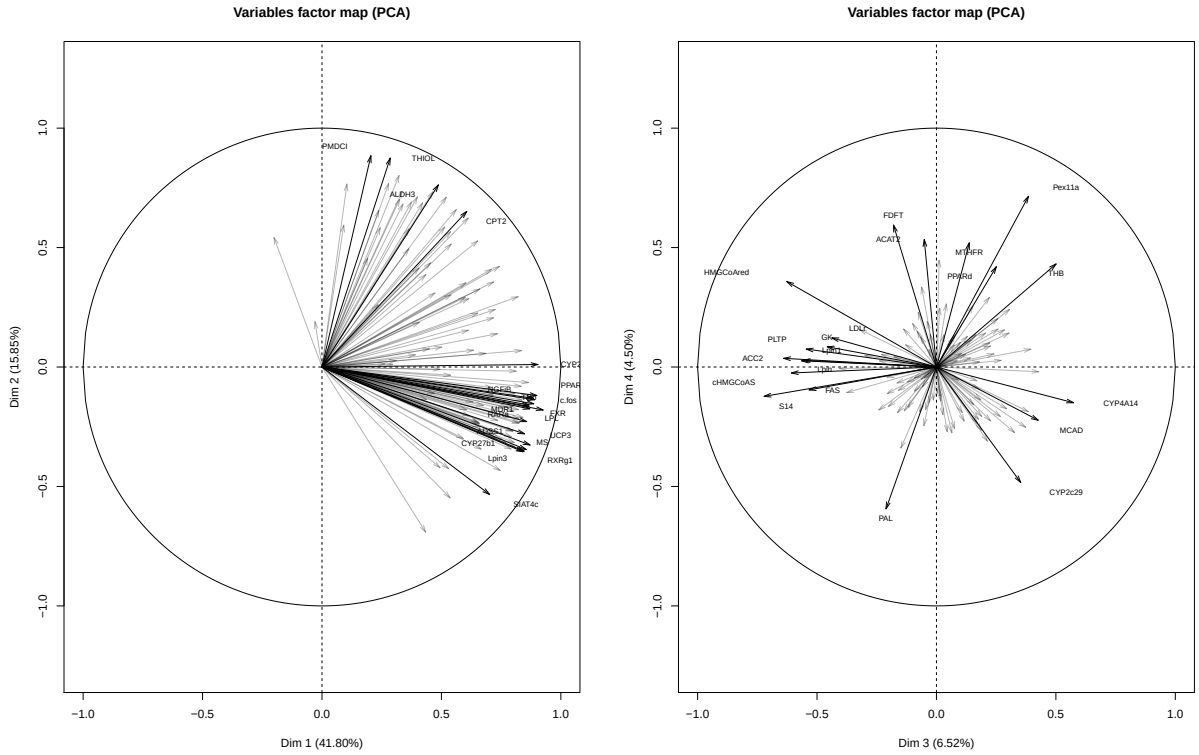


Figure 3.10: Left: correlation between the genes and dimensions 1 and 2. Right: correlation between the genes and dimensions 3 and 4.

Left panel of Figure 3.9 represents the scores on the first two dimensions of the PCA with the mice colored according to their genotype. The figure shows that the second dimension of variability differentiates genotypes W and NW. In the right panel of Figure 3.9, the observations (mice) are projected on dimensions 3 and 4, and each diet is represented at the barycenter of the mice that take this diet. This figure highlights that dimension 3 differentiates mice with the regime efad from mice with the regime dha. Figure 3.10 represents the loadings, *i.e.* the correlations between the variables (the genes) and the principal components. The rules of interpretation are the following: mice with large coordinates on a given principal component have high expressions of the genes highly positively correlated with that dimension and low expressions of the genes highly negatively correlated with that dimension.

In Table 3.1 we report the results of some other methods, which we considered in the simulation study. It turns out that Passemier almost agrees with PESEL and returns 6 PCs, while CSV fails completely (which may be due to the sub-optimal implementation of numerical integration). Compared to other methods PESEL is extremely fast to compute. Specifically, it is 10 times quicker than GCV and 20 times quicker than Passemier.

Table 3.1: Elapsed time of the analysis of the mice data-set on the 8-core computer with Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz

Method	Time	Number of PCs
PESEL	4.5 ms	5
GCV	47.5 ms	12
Passemier	85 ms	6
CSV	2200 ms (R) + 76 ms (MATLAB)	0

4 Dimensionality reduction via variables clustering

Overview

In this chapter we propose a new method for subspace clustering problem: Multiple Latent Components Clustering (MLCC), which is a major extension of an algorithm described in [Chavent et al., 2012]. We formulate a probabilistic model and starting from 'maximum a posteriori' rule, we derive a modified Bayesian Information Criterion (mBIC). Our method enables clustering of multivariate data under the assumption that variables are grouped in low-rank subspaces. We created an efficient heuristic algorithm for identifying models with large values of mBIC criterion. It is based on k-medoids algorithm, where a) each cluster is a subspace b) medoid (representative, center) of a cluster is a set of principal components c) similarity measure between variable and the "representative" is given by the classical Bayesian Information Criterion in the respective multiple regression model. The final number of clusters and their dimensions are selected based on mBIC. The proposed method is implemented in **R** package *varclust* [Sobczyk et al., 2017b]. The simulation study confirms good properties of our method as compared to existing, state-of-the-art approaches. The algorithm, initial versions of the package and the simulation study were reported at the conference "Statystyka Matematyczna" [Sobczyk et al., 2014]. All results are reported in the forthcoming preprint Sobczyk et al. [2019].

4.1 Probabilistic model for subspace clustering

Let us denote the number of clusters by K and let us assume that after some permutations of columns $\mathbf{X}$ can be represented as

$$\mathbf{X} = \mathbf{M} + E \ ,$$

where

$$\mathbf{M} = F_{n\times k} C_{k\times p} \ ,$$

with $F = [F_1, \ldots, F_K]$, $F_i \in M_{n\times k_i}$, $\sum_i k_i = k$ and the matrix C where in each column of C the nonzero elements can occur only in rows corresponding to variables spanning a given cluster. Clearly, such a representation is not unique. However, our main goal is to find permutation and representation of subspaces, such that k is minimized. Putting above in a less formal way, we assume that the data comes from the union of subspaces. This means that the variables (columns of the data matrix) can be divided into clusters, each of which consists of variables from one of the subspaces. We work under an assumption that subspaces are low dimensional. Therefore every variable in a single cluster can be expressed as a linear combination of small number of factors (common for every variable in this cluster) plus some noise and a constant vector (mean).

To formalize these statements, let us consider data coming from just one cluster. We denote $X_i = (x^i_1, \ldots, x^i_{p_i})$ as set of the variables in i^{th} cluster. $x^i_j \in \mathbb{R}^n$ denotes j^{th} variable and

n is, as always, the number of individuals. Let $F_i \in \mathbb{R}^{n \times k_i}$ stand for the matrix of factors. Let us denote by $c_1, \ldots, c_{k_i}$ vectors of coefficients corresponding to variables $x^i_1, \ldots, x^i_{p_i}$ where $c^i_j = (c^i_{j1}, \ldots, c^i_{jk_i})^T$. Then, the matrix X_i containing variables from i^{th} cluster that comes from the above model, has the form

$$X_i = F_i C_i + \mu_i + \varepsilon$$
$$\varepsilon \sim N(0, \sigma_i^2 I_n)$$

For each individual subspaces we assume the same probabilistic model as in PESEL. Depending on whether $n > p_i$ or $p_i < n$ we assume normal prior for C_i or F_i (see section 3.1 for details) and μ is matrix with either identical rows or columns.
Model above is assumed for every cluster. Therefore for the entire data $\mathbf{X}$ we specify model

$$M^K = \{K, \{k_1, \ldots, k_K\}, \{D_i\}, \{\theta_i\}\},$$

where

- K - number of clusters (low dimensional subspaces that data come from)
- $k_1, \ldots, k_K$ - dimensions (number of factors) of each cluster,
- segmentation $\{D_i\}$ - segmentation of variables into matrices X_i – clusters, subsets of columns of $\mathbf{X}$
- set of parameters for the i-th cluster: θ_i $\{F_i, C_i \in M_{k_i \times p_i}, \sigma_i^2, \mu_i\}$. C_i are nonzero elements in the rows of matrix of coefficients C corresponding to F_i. We have separate sets for each cluster $i = 1, \ldots, K$

Please take a careful look at the notation in this section. $\mathbf{M}$ represents the signal in the data, while M is a model. Let us denote by M_i the model (set of variables and parameters) for i^{th} cluster. $f(\theta_i)$ is likelihood function in the i^{th} cluster. We denote space of parameters for i^{th} cluster by Θ_i and for whole model by Θ. One representative of the latter is $\theta = (\theta_1, \ldots, \theta_K)$.
We want to select model that maximizes a posteriori probability

$$\ln\left(P(M|\mathbf{X})\right) = -\ln P(\mathbf{X}) + \ln P(M) + \ln\left(P(\mathbf{X}|M)\right)$$

$P(\mathbf{X})$ does not depend on the model. $P(M)$ is prior distribution on the model. We shall focus on it in the next section. For now, let us consider last term in the sum above. We assume that prior distributions for all clusters are independent which yields:

$$\begin{aligned}
\ln\left(P(\mathbf{X}|M)\right) &= \ln\left(\int_\Theta f(X|\theta) f(\theta|M) d\theta\right) \\
&= \ln\left(\int_\Theta \prod_{i=1}^K f(X_i|\theta_i) f(\theta_i|M_i) d\theta\right) \\
&= \ln\left(\prod_{i=1}^K \int_{\Theta_i} f(X_i|\theta_i) f(\theta_i|M_i) d\theta_i\right) \\
&= \sum_{i=1}^K \ln\left(\int_{\Theta_i} f(X_i|\theta_i) f(\theta_i|M_i) d\theta_i\right) = \sum_{i=1}^K \ln\left(P(X_i|M_i)\right) \qquad (4.1)
\end{aligned}$$

Approximation of individual terms $\ln\left(P(X_i|M_i)\right)$ is provided by PESEL, described in details in Chapter 3.

4.1.1 Choosing prior distribution

In the classic BIC prior distribution is set to be non-informative (see e.g. [Neath and Cavanaugh, 2012]). Which means that $\ln P(M^K)$ has no effect on estimation process. However, because of all possible variables segmentations, when the number of clusters K increases, number of models grows exponentially K^p. Due to this fact, BIC tends to overestimate number of clusters. Similar phenomenon was previously described for the problem of regression in [Bogdan et al., 2004]. To mitigate this effect we must use an informative prior that would lead to each number of clusters being equally probable.
Let us consider how many different partitions we have for a given model. Each of p variables can be assigned to any of K clusters, so we have approximately K^p different partitions. Since we want each number of clusters to be a priori equally probable, it brings us to the following formula

$$\begin{aligned} P(M^K) &\sim \frac{1}{K^p} \\ \log(P(M^K)) &\sim -p\log(K) \end{aligned} \tag{4.2}$$

Moreover, we need to take into account the dimensions of clusters. Assume that maximal dimensions of single subspace is d. Then for fixed segmentation we have d different models for every cluster. This gives total of d^K different possible dimensions of the clusters. Thus we need to adjust 4.2 and prior distribution is the following:

$$\begin{aligned} P(M^K) &= \frac{1}{K^p}\frac{1}{d^K} \\ \ln(P(M^K)) &= -p\ln(K) - K\ln(d) \end{aligned} \tag{4.3}$$

Combining (4.3) and we get

$$\begin{aligned} \log(P(M^K|X) &\approx \log(P(M^K)) + \log(P(X|M^K)) \\ &= \log(P(M^K)) + \sum_{i=1}^{K} \ln\left(P(X_i|M_i^K)\right) \\ &\sum_{i=1}^{K} \ln\left(P(X_i|M_i^K)\right) - p\ln(K) - K\ln(d) \end{aligned} \tag{4.4}$$

Approximation to formula for $P(X_i|M_i^K)$ is provided by PESEL and depends on whether $n > p_i$ (see Section 3.1.1). It is given either by (3.19) or (3.12).

4.2 Heuristic algorithm to reduce computational burden

In the previous section w argued that finding a model that maximizes mBIC (4.4) is a viable approach for solving subspace clustering problem. However, due to the huge number of competing models, computing mBIC for each of them becomes intractable even for moderate p and d. Therefore, in Sobczyk et al. [2016] we propose heuristic algorithm for finding optimal value of criterion (4.4). This algorithm is based on a specific implementation of K-means algorithm

where subspaces (clusters) centers are vectors that span those subspaces. We compute them by taking a subset of principal components. Their number i.e. dimension of cluster is not fixed, but estimated using PESEL. In each iteration variables are assigned to the cluster closest to it. We measure that based on regular BIC - penalized residual sum of squares.
K-means algorithm is known to get stuck at the local maxima. Performance strongly depends on the choice of initial subspaces. To reduce the chance that the local minimum of mBIC is obtained instead of the global one, algorithm is run many times, with different (not necessarily random) initializations of cluster centers.

Algorithm 6 Multiple Latent Components Clustering

Input: n - number of individuals, p - number of variables, $X_{n\times p} = (x_1, \ldots, x_p)$ - data set, d - maximal subspace dimension, N - number of runs of the algorithm
Scale X to have columns with mean 0 and unit variance
for $i \in \{1, \ldots, N\}$ **do**

1. Initialize clusters' centers
2. Until convergence or maximal number of iterations is reached repeat
   a) For every variable x_j and every cluster factors $F_{j'}$ fit a linear regression model without intercept $lm(x_j \sim F_{j'})$ and store BIC value as $BIC_{jj'}$
   b) Assign each variable x_j to the cluster M_q where
   $$q = \underset{j' \in \{1,\ldots,K\}}{\arg\max}\ BIC_{jj'}$$
   c) For every cluster M_i use PESEL to estimate its dimensionality k_i with an upper bound of d. Use PCA to compute the first k_i principal components and store them in F_i
3. Store mBIC for computed model

end for
Choose the model with the highest value of mBIC and return the model (segmentation, mBIC, factors) as the result.

The first step of our algorithm (initialization), in section (4.3) we will compare two approaches - random initialization and initialization by the result of SSC. What is significant MLCC can also be run when the number of clusters is not known. In that case the algorithm is run for different number of clusters and the best model is chosen using mBIC. It is crucial because none of the methods mentioned in the next section can be directly used to automatically choose the number of clusters.
MLCC can be viewed as an extended version of ClustOfVar (COV, Chavent et al. [2012]). This algorithm also exploits K-means method. Initial clusters centers are chosen randomly from the data. Unlike in MLCC the center of a cluster is always one variable. The similarity measure is squared Pearson correlation coefficient. In each iterative step of the algorithm, every variable is assigned to the most similar cluster using mentioned measure. Then for every cluster, PCA is performed to find the first principal component and make it a new cluster center. What is interesting this method can be used both on quantitative and qualitative variables. The downside is that no rationale for selecting number of clusters was introduced.

4.3 Simulation study

In this section, we present the result of simulation studies, in which we compare MLCC with other methods of clustering. To measure the quality of the procedures we use introduce 3 measure of clustering effectiveness. We also compare execution time.

4.3.1 Clustering methods

In our simulations we compare the following methods:

1. MLCC with random initialization (MLCC)
2. MLCC with initialized with the result of Sparse Subspace Clustering (MLCC$_{aSSC}$)
3. Sparse Subspace Clustering (SSC, Elhamifar and Vidal [2009])
4. Low Rank Subspace Clustering (LRSC, Vidal and Favaro [2014])
5. ClustOfVar (COV, Chavent et al. [2012])

In the first method we use random initialization which is a default setting in our software. This means that we sample without replacement K variables from the data set and assign each of them to $F_1, \ldots, F_K$ respectively. This procedure is done multiple times to avoid local minimum. In the second method we take advantage of the possibility to provide the initial segmentation before the start of the algorithm. It accelerates the procedure, because thanks to such a hot start, we no longer need to run the algorithm many times. We initialize the centers by performing the step 2 c) of the Algorithm 6 using given segmentation. In our simulation study, we use the segmentation returned by SSC. Third and fourth methods are based on spectral clustering and are described in detail in section 2.5.1. ClustOfVar is briefly described in the previous section.

4.3.2 Synthetic data generation

To generate synthetic data to compare the methods from the previous section we use two data generation methods. We shall refer to them later as modes. In the first mode, factors spanning the subspaces in the first mode are shared between subspaces. In the second mode subspaces are independent which is in accordance with the assume probabilistic model in section 4.1.

Here we present exact algorithms used for data generation:

Algorithm 7 Data generation with shared factors

Input: n - number of individuals, SNR - signal to noise ratio, K - number of clusters, p - number of variables, d - maximal dimension of a subspace
Number of factors $m \leftarrow K\frac{d}{2}$
Factors $F = (f_1, \ldots, f_m)$ are generated independently from the multivariate standard normal distribution and then F is scaled to have columns with mean equal to 0 and standard deviation 1
Draw subspaces' dimension $d_1, \ldots d_K$ uniformly from $\{1, \ldots, d\}$
for $i = 1, \ldots, K$ **do**
  Draw i-th subspaces basis as sample of size d_i uniformly from columns of F as F_i
  Draw matrix of coefficients C_i from $U(0.1, 1) \cdot sgn(U(-1, 1))$
  Variables in the i-th subspace are $X_i \leftarrow F_i C_i$
end for
Scale matrix $X = (X_1, \ldots, X_K)$ to have columns with unit variance
return $X + Z$ where $Z \sim N(0, \frac{1}{SNR} I_n)$

Algorithm 8 Data generation with independent subspaces

Input: n - number of individuals, SNR - signal to noise ratio, K - number of clusters, p - number of variables, d - maximal dimension of a subspace
Draw subspaces' dimension $d_1, \ldots d_K$ uniformly from $\{1, \ldots, d\}$
for $i = 1, \ldots, K$ **do**
  Draw i-th subspaces basis F_i as sample of size d_i from multivariate standard normal distribution
  Draw matrix of coefficients C_i from $U(0.1, 1) \cdot sgn(U(-1, 1))$
  Variables in i-th subspace are $X_i \leftarrow F_i C_i$
end for
Scale matrix $X = (X_1, \ldots, X_K)$ to have columns with unit variance
return $X + Z$ where $Z \sim N(0, \frac{1}{SNR} I_n)$

4.3.3 Measures of effectiveness

To compare clustering produced by clustering methods we use three measures of effectiveness.

1. Adjusted Rand Index - one of the most popular measures of clustering effectiveness. Let A, B be the partitions that we compare (one of them should be true partition). Let a, b, c, d denote respectively the number of pairs of points from data set that are in the same cluster both in A and B, that are in the same cluster in A but in different clusters in B, that are in the same cluster in B but in different clusters in A and that are in the different clusters both in A and B. Note that the total number of pairs is $\binom{p}{2}$. Then

$$ARI = \frac{\binom{p}{2}(a+d) - [(a+b)(a+c)+(b+d)(c+d)]}{\binom{p}{2}^2 - [(a+b)(a+c)+(b+d)(c+d)]}$$

   The maximum value of ARI is 1 and when we assume that every clustering is equally probable its expected value is 0. For details check Hubert and Arabie [1985].

The following two measures are taken from Sołtys [2010]. Let $X = (x_1, \dots x_p)$ be the data set, A be a partition into clusters $A_1, \dots A_n$ (true partition) and B be a partition into clusters $B_1, \dots, B_m$.

2. Integration - for the cluster A_j it is given by formula

$$Int(A_j) = \frac{max_{k=1,\dots,m} \#\{i \in \{1, \dots p\} : x_i \in A_j \wedge x_i \in B_k\}}{\#A_j}$$

Cluster B_k for which the maximum is reached is called integrating cluster of A_j. Integration can be interpreted as the percentage of data points from given cluster of true partition are in the same cluster in partition B. For the whole clustering

$$Int(A, B) = \frac{1}{n} \sum_{j=1}^{n} Int(A_j)$$

3. Acontamination - for cluster A_j it is given by formula

$$Acont(A_j) = \frac{\#\{i \in \{1, \dots p\} : x_i \in A_j \wedge x_i \in B_k\}}{\#B_k}$$

where B_k is integrating cluster for A_j. Idea of acontamination is complementary to integration. It can be interpreted as the percentage of the data in the integrating cluster B_k are from A_j. For the whole clustering

$$Acont(A, B) = \frac{1}{n} \sum_{j=1}^{n} Acont(A_j)$$

Note that the bigger ARI, integration and acontamination are, the better is the clustering. For all three indices the maximal value is 1.

4.3.4 Simulation results

In the following sections we present the results of our simulation. In order to make the results reliable, for a given set of parameters, we generate the data (using algorithms (7) and (8)) 100 times. We plot multiple boxplots to present not only mean performance but also the variability of clustering for different methods. *dimension* stands for maximum dimension in the data generation algorithms. By default the number of runs (random initializations) is set to $n_{init} = 30$ and the maximal number of iterations within the kmeans loop is set to $n_{iter} = 30$. These parameters proved to be a good trade-off between speed and accuracy. Other parameters used to generate the data are written below the plots. To distinguish between two different algorithms for generating data we use label *mode*, which takes the value *shared*, if the subspaces may share the factors, and the value *not_shared* otherwise.

Generation method

In this section we compare performance for one specific choice of parameters and two different methods for generating data.

(a) factors not shared (b) shared factors

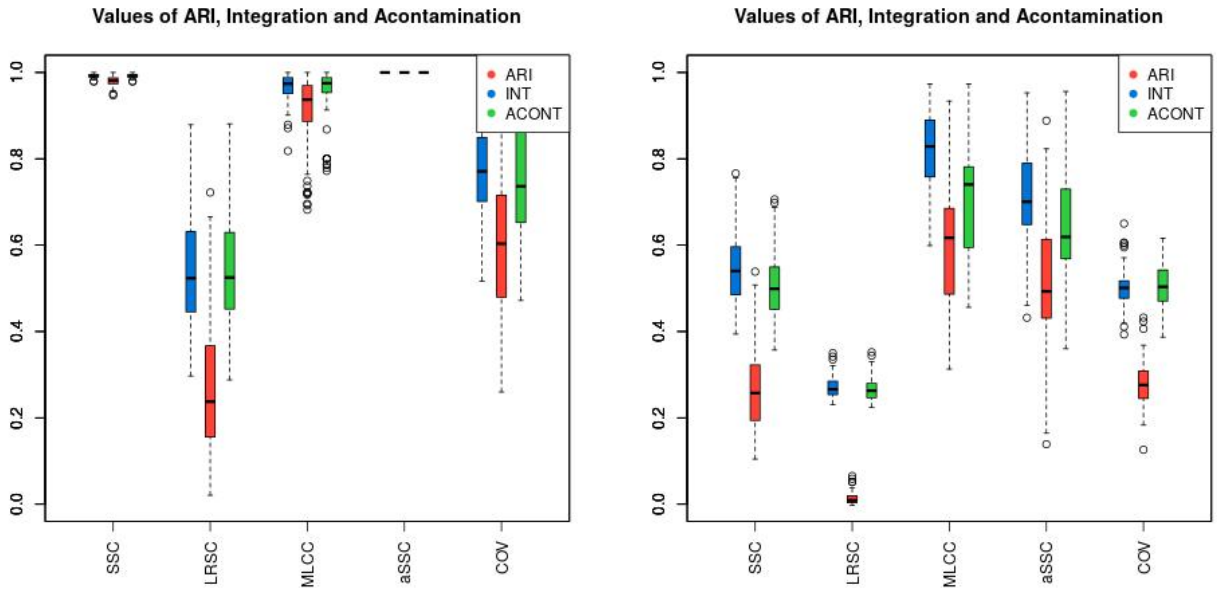


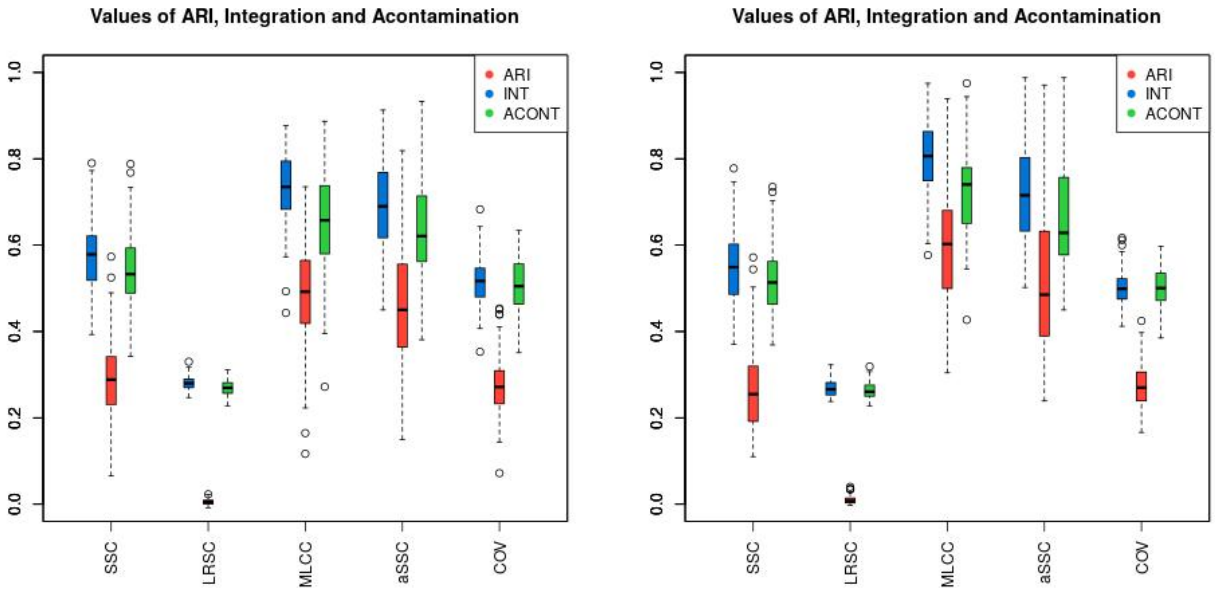
Figure 4.1: Comparison with respect to the data generation method. Simulation parameters: $repets = 100,\ n = 100,\ p = 800,\ K = 5,\ d = 3,\ SNR = 1$.

When the factors are not shared, SSC and MLCC provide almost perfect clustering, see figure 4.1. Observe that clustering subspaces with shared factors is much more difficult. All the methods give worse results in that case. However, MLCC and MLCC$_{aSSC}$ outperform all the other methods and provide quite good clustering in opposite to SSC, LRSC and COV. The reason for that is the mathematical formulation of SSC and LRSC - they assume that the subspaces are independent which means that they do not have common factors in their bases. It seems that they are not robust to the violation of this assumption.

Number of variables

In this section we compare the change in performance of methods with respect to the number of variables in the dataset (figure 4.2).

(a) $p = 300$ (b) $p = 600$



(c) $p = 800$ (d) $p = 1500$

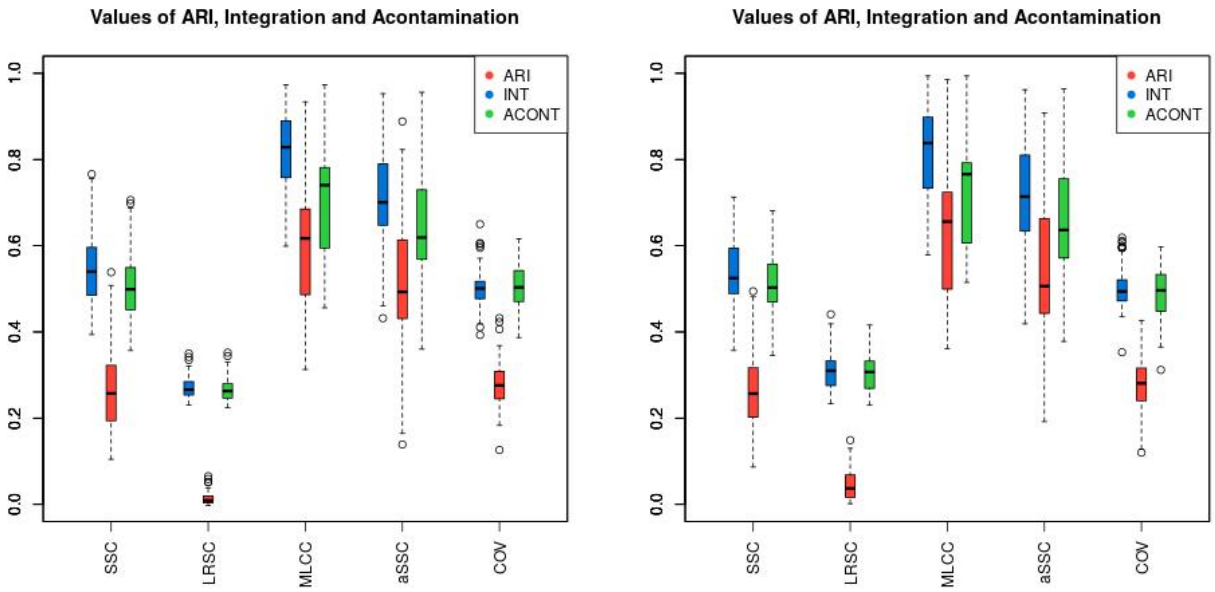


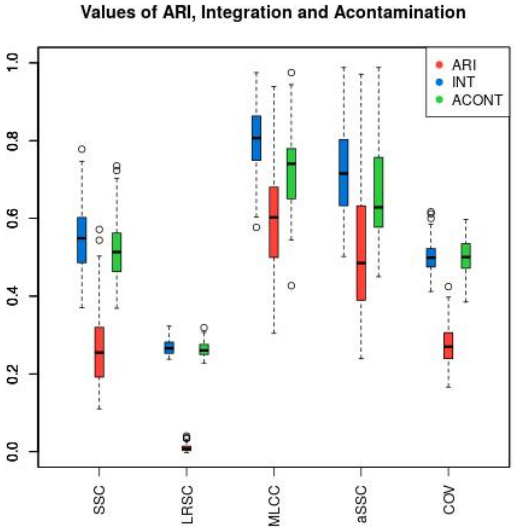
Figure 4.2: Comparison with respect to the number of variables. Simulation parameters: $repets = 100,\ n = 100,\ K = 5,\ d = 3,\ SNR = 1,\ mode : shared.$

When the number of variables increases, MLCC tends to produce better clustering. This is an expected effect because when the number of clusters and subspace dimension stay the same the total information about the cluster structure grows with every additional variable. Therefore also PESEL from (3.19) gives a better approximation of the cluster dimensionality and the task of finding the real model becomes easier. Note however, that for COV, LRSC, SSC this does not hold as the results are nearly identical.

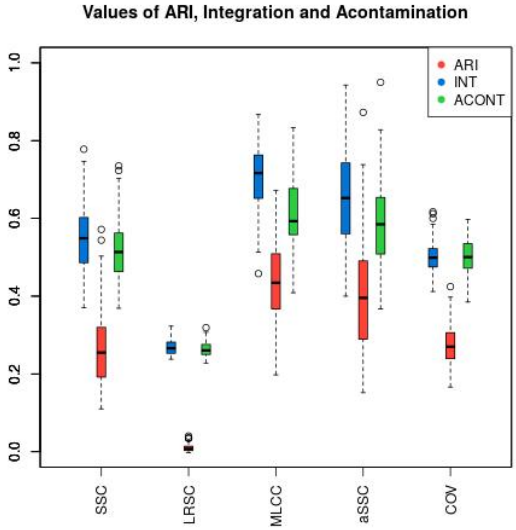
Maximal dimension of subspace

We also check what happens when the number of parameters in the model of MLCC increases. This could lead to overfitting as clusters are possibly more complex. Below, in the left column, we compare the methods with respect to the maximal dimension of a subspace ($dim = 3, 5, 7$). In the 'real world' clustering problems it is however uncommon to know in advance the maximal dimension of the subspaces. Therefore, in the right column, we check the performance of MLCC and MLCC$_{aSSC}$ when let maximal subspace dimension to be twice as large as true one.

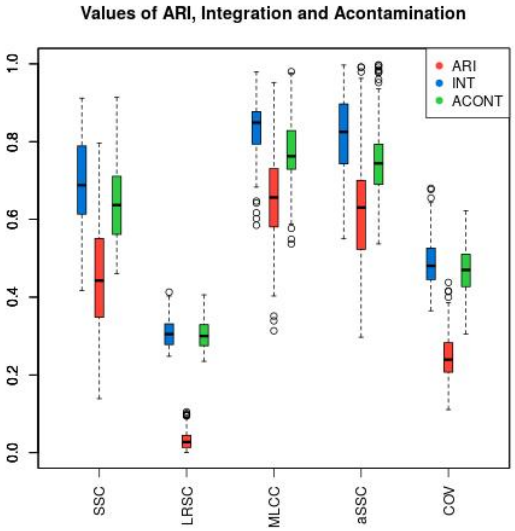
(a) $d = 3$



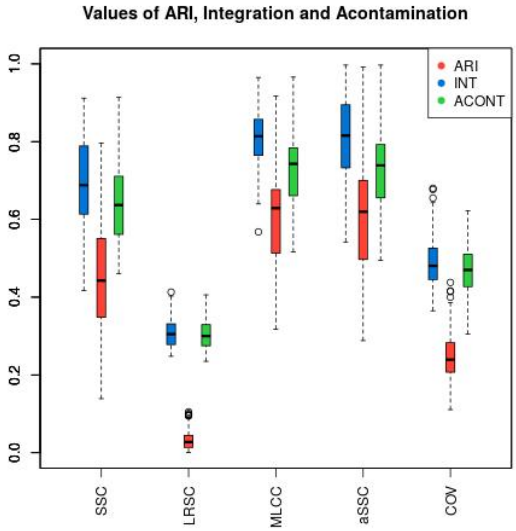
(b) $d = 3$



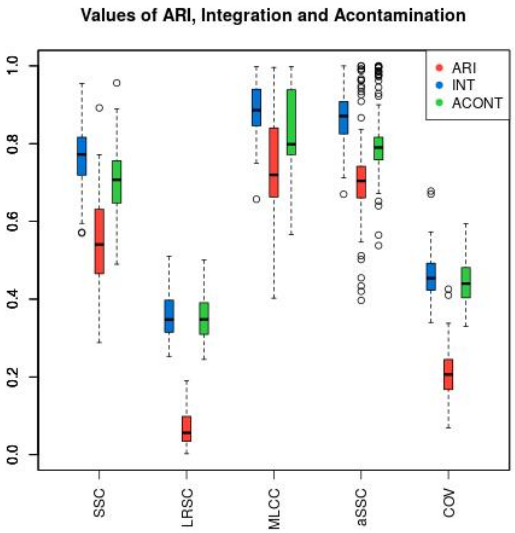
(c) $d = 5$



(d) $d = 5$



(e) $d = 7$



(f) $d = 7$

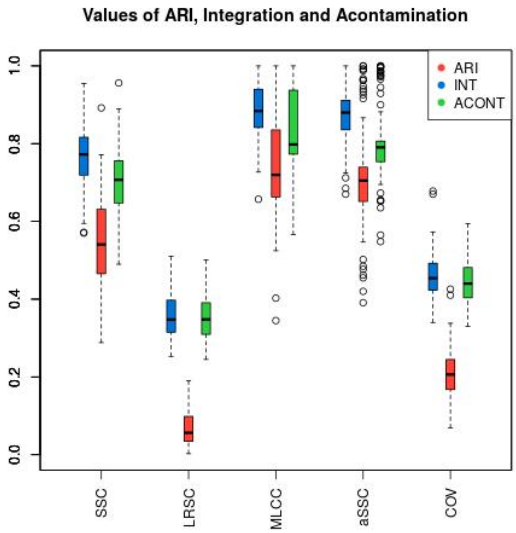


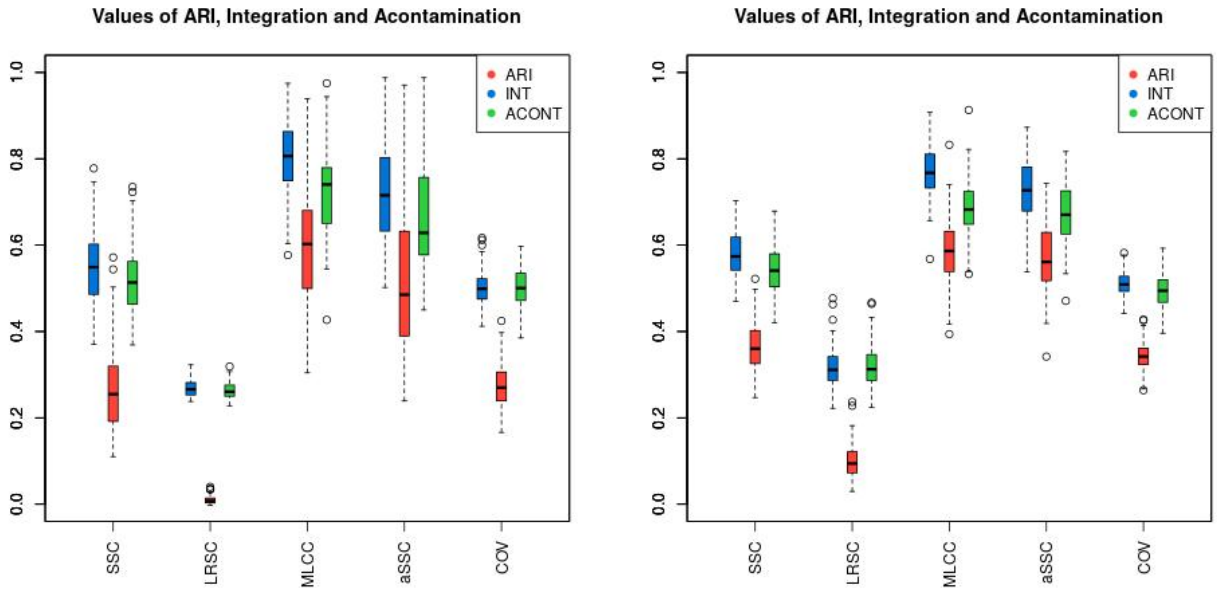
Figure 4.3: Comparison with respect to the number of variables. Simulation parameters: $repets = 100$, $n = 100$, $p = 600$, $K = 5$, $SNR = 1$, $mode : shared$. In the left column the maximal dimension passed to MLCC was equal to d, in the right we passed $2d$.

Looking at the first column, we can see that the effectiveness of MLCC grows slightly when the maximal dimension increases. However, this effect is not as noticeable as in SSC. This effect may seem to be unexpected for MLCC but variables from subspaces of higher dimensions are easier to distinguish because their bases have more not shared components. In the second column, the effectivenesses of our methods are very similar to the first column except for $dimension = 3$, where the difference is not negligible. Nonetheless, these results indicate that thanks to PESEL, MLCC performs well in terms of estimating the dimensions of the subspaces.

Number of clusters

The number of the parameters in the model for MLCC also grows significantly with the number of clusters in the data set (figure 4.4).

(a) $K = 5$ (b) $K = 10$



(c) $K = 15$ (d) $K = 20$

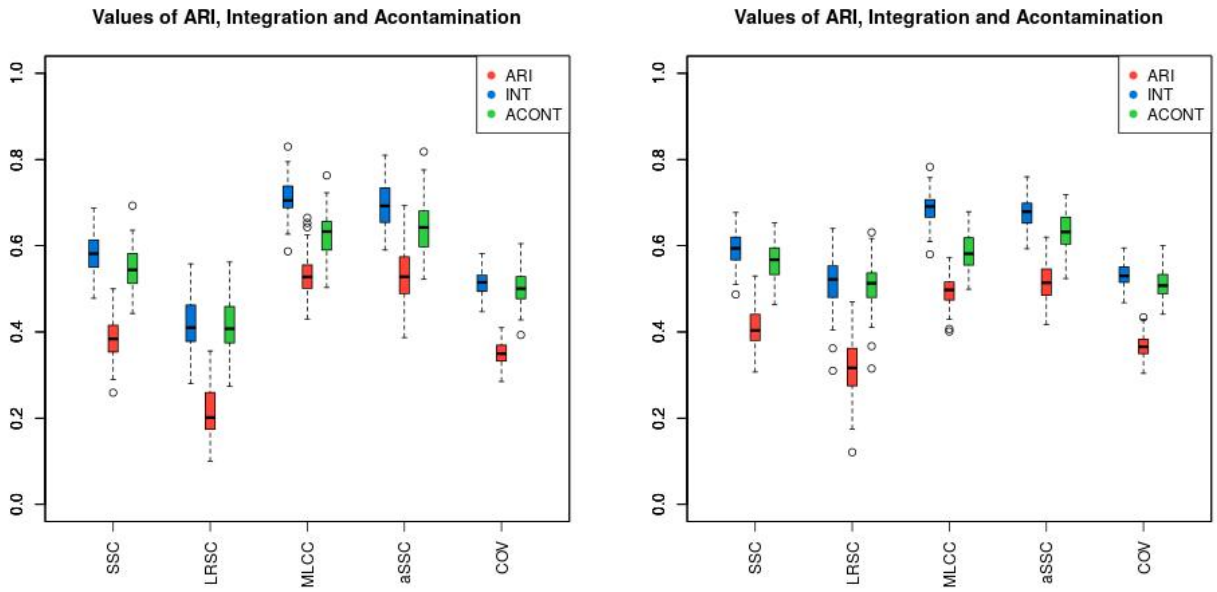


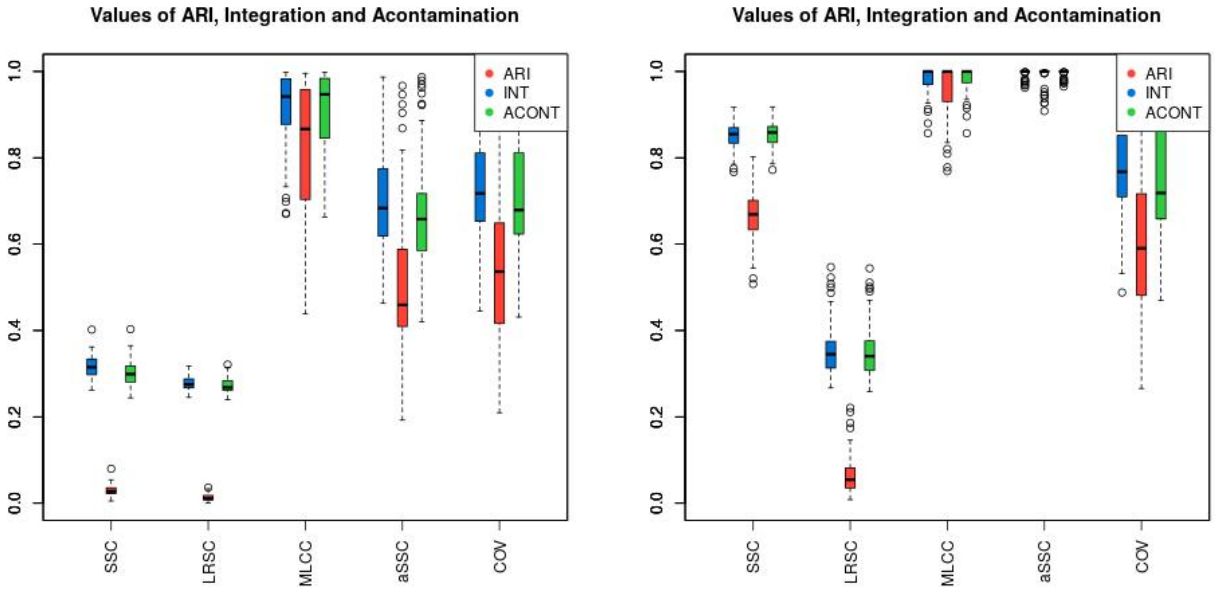
Figure 4.4: Comparison with respect to the number of clusters. Simulation parameters: $repets = 100,\ n = 100,\ p = 600,\ d = 3,\ SNR = 1,\ mode : not\ shared.$

We can see that for MLCC the effectiveness of the clustering diminishes when the number of clusters grows. The reason is the increased number of parameters in our model to estimate. The opposite effect holds for LRSC, SSC and COV, although it is not as noticeable as for our methods.

Signal to noise ratio

One of the most important features of the data set is signal to noise ratio (SNR). Of course, the problem of clustering is much more difficult when SNR is small because the corruption caused by noise dominates the data. However, it is not uncommon in practice to find data where $SNR < 1$ (figure 4.5).

(a) $SNR = 0.5$ (b) $SNR = 0.75$



(c) $SNR = 1$ (d) $SNR = 2$

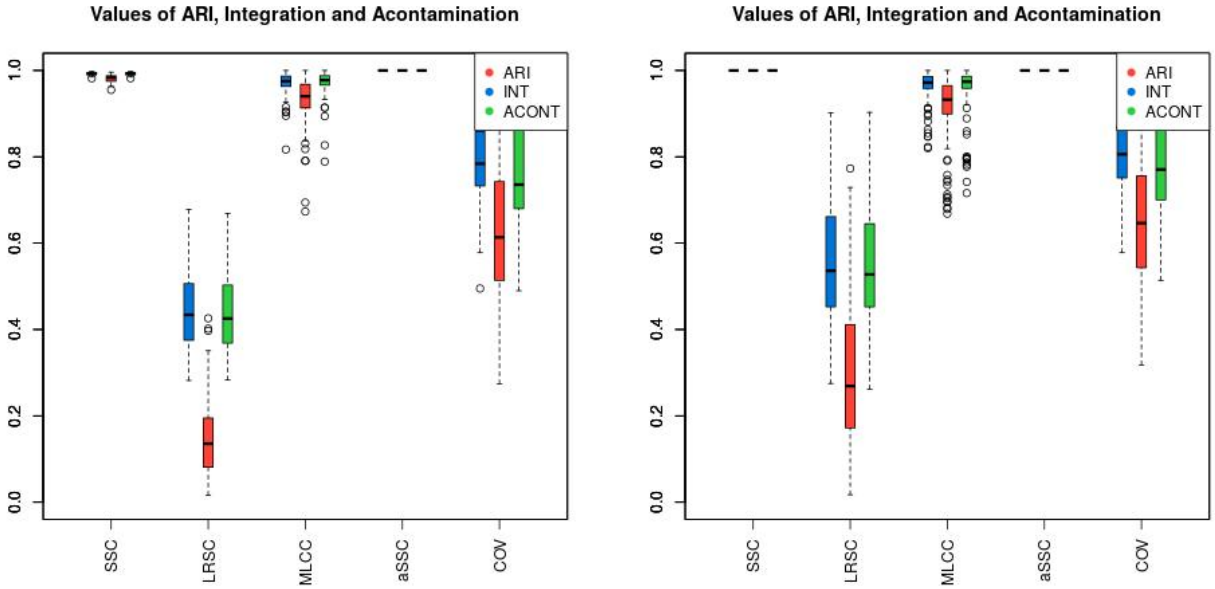


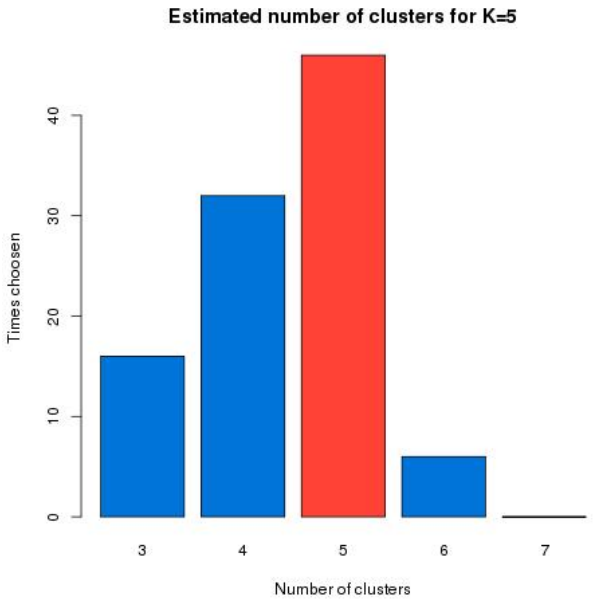
Figure 4.5: Comparison with respect to the signal to noise ratio. Simulation parameters: $repets = 100,\ n = 100,\ p = 600,\ K = 5,\ d = 3,\ mode : not\ shared.$

For $SNR = 0.5$, MLCC produces a very good clustering. In contrary, SSC and LRSC perform poorly. All methods give better results when SNR increases, however for SSC this effect is the most noticeable. For $SNR \geq 1$, SSC produces perfect or almost perfect clustering while MLCC performs slightly worse.

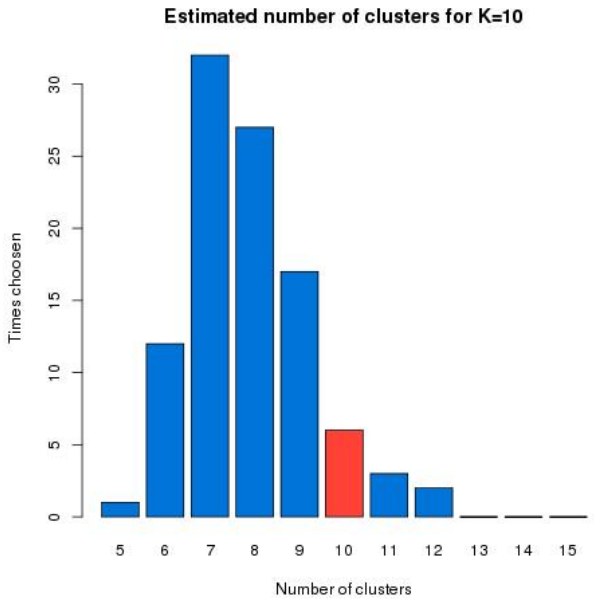
Estimation of the number of clusters

Despite a large number of existing methods used in variable clustering, there are not many tools for automatic detection of the number of clusters in the problem of clustering of variables. Thanks to mBIC defined in Section 3, MLCC can be used in such scenario. W generate the data set with given parameters 100 times and check how often each number of clusters from range $\left[K - \frac{K}{2}, K + \frac{K}{2}\right]$ was chosen (figure 4.6)

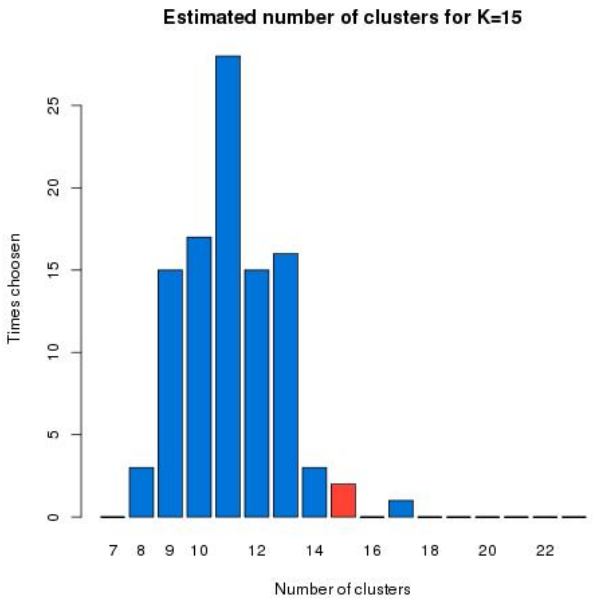
(a) $K = 5$



(b) $K = 10$



(c) $K = 15$



(d) $K = 20$

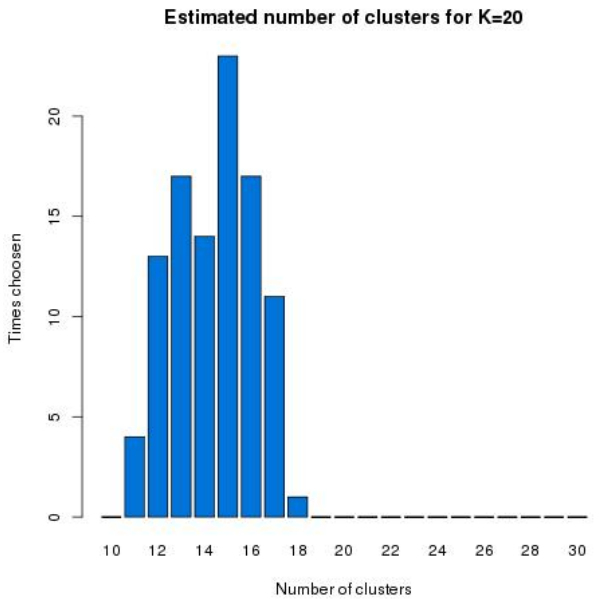


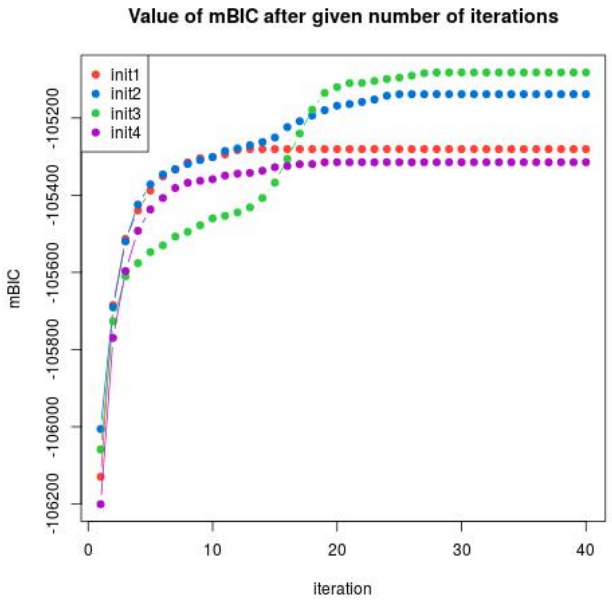
Figure 4.6: Estimation of the number of clusters. Simulation parameters: $repets = 100$, $n = 100$, $p = 600$, $d = 3$, $SNR = 1$ $mode : not\ shared$.

We see that for $K = 5$ the correct number of clusters was chosen most times. However, when the number of clusters increases, the clustering task becomes more difficult, the number of parameters in the model grows and MLCC tends to underestimate the number of clusters.

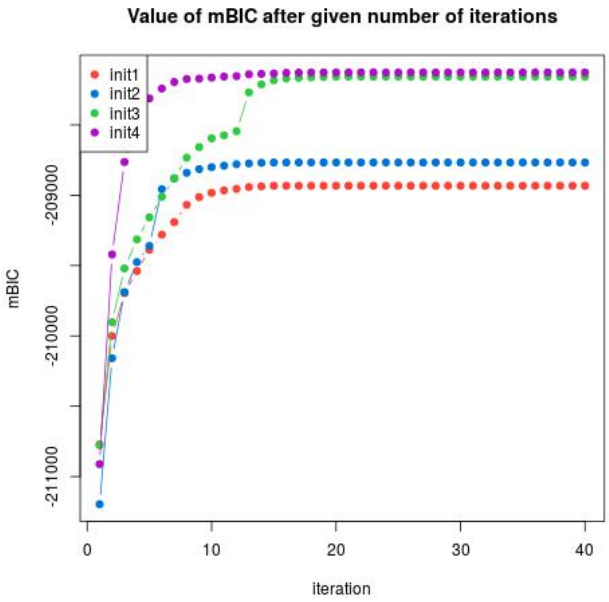
4.3.5 Convergence speed analysis

In this section we check how mBIC converges with following iteration of the Kmeans loop for four different initializations (figure 4.7).

(a) $p = 750$



(b) $p = 1500$



(c) $p = 3000$

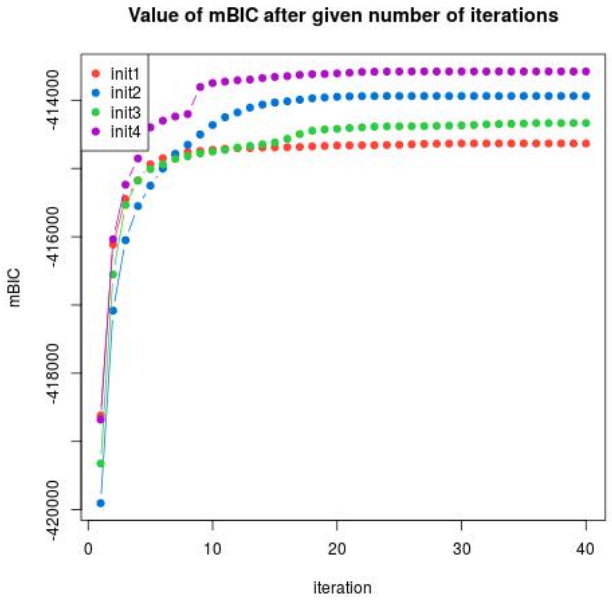


Figure 4.7: mBIC with respect to the number of iteration for 4 different initializations. Simulation parameters: $n = 100,\ K = 5\ d = 3,\ SNR = 1\ mode : shared.$

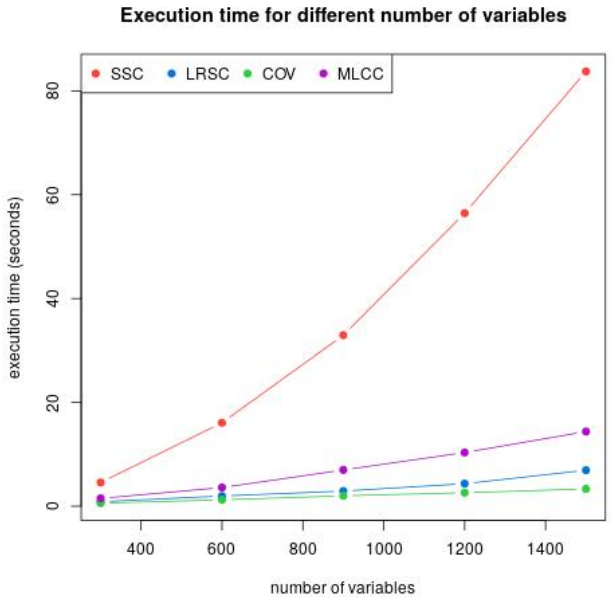
We can see that the convergence of mBIC criterion is quite fast: in most cases it needed no more than 20 iterations of the Kmeans loop. We can also notice that the size of the dataset (in this case the number of variables) doesn't have big influence on the number of iterations

till convergence. However the results in figure 4.7 show that multiple random initializations in our algorithm are required to get satisfying results - the value of mBIC criterion varies a lot between different initializations.

Execution time

In this section we compare the execution times of compared methods. They were obtained on the machine with *Intel(R) Core(TM) i7-4790 CPU 3.60GHz, 8 GB RAM*. The results are in figure 4.8. For the left plot $K = 5$ and for the right one $p = 600$.

(a) With respect to the number of variables



(b) With respect to the number of clusters

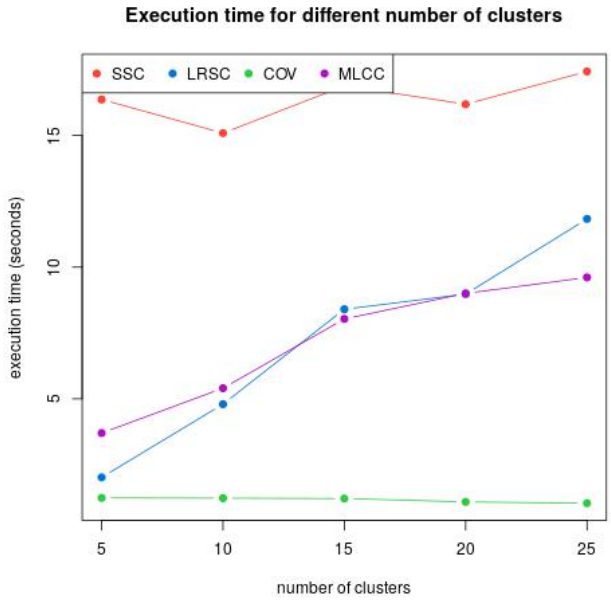


Figure 4.8: Comparison of the execution time of the methods with respect to p and K. Simulation parameters: $repets = 100,\ n = 100,\ d = 3,\ SNR = 1\ mode : shared$.

On the plots for both MLCC and COV we used only one random initialization. Therefore we can see that for $n_{init} = 30$ the execution time of MLCC will be proportionally larger. However, thanks to parallel implementation in Sobczyk et al. [2016], MLCC execution time is proportional to $\frac{n_{init}}{n_{cores}}$ where n_{cores} denotes the number of cores used (parameter given by the user). Nonetheless, it MLCC is the most computationally complex of these methods. On the other hand we can see that COV and SSC do not take longer for bigger number of clusters when the opposite holds for MLCC and LRSC. What is more when the number of variables increases, the execution time of SSC grows much more rapidly than time of one run of MLCC. Therefore, for bigger datasets it is possible to test more random initializations of MLCC in the same time as computation of SSC. Furthermore, running MLCC with segmentation returned by SSC (enhancing the clustering) is not much more time consuming than SSC itself.

4.3.6 Summary of simulation results

The simulation results proved that usefulness of MLCC as a method for variable clustering. Unlike other approaches, MLCC performs well even in the data with significant noise. It also

proves its ability to recognize subspaces which have some common factors. This is an extremely valuable feature having in mind applying it to identify genetic pathways. MLCC seems also to be quite resistant to the increase of the maximal dimension of a subspace. To sum up, in every setting of the parameters used in our simulation, MLCC outperformed LRSC and COV and did better or as well as SSC. Furthermore, it can be used to detect the number of clusters in the data set. Although it slightly underestimates number of clusters, simulation results suggests that it is reasonably accurate.

The main drawback of MLCC is its computational complexity. Therefore, to reduce the execution time one can provide custom hot initialization as in MLCC$_{aSSC}$. This strategy in all cases provided better results than SSC. It suggests that our algorithm can also be used to enhance the clustering results of the other methods. This computational cost is related to the choice of the parameters n_{init} or n_{iter}. Unfortunately, when data size increases, in order to get acceptable clustering we have to increase at least one of these two values. However, it is worth mentioning that in case of parameters used in out tests $n_{init} = 30$ and the maximal number of iterations equal to 30 on a machine with 8 cores the execution time of MLCC is comparable with execution time of SSC. So the problem is also with the complexity of the task, not only the limits of the algorithm.

5 Graphical Slope

Overview

In this chapter we introduce a new approach to estimating sparse precision matrix called graphical SLOPE (gSLOPE). We define it as a convex optimization problem where objective is a regularized likelihood function. Penalty term is based on ordered ℓ_1 norm (2.48). We introduce two strategies for choosing sequence of λs. One is based on Holm correction for multiple testing, other on Benjamini-Hochberg (B-H). We prove that for our specific choice of λs, under certain assumptions on true precision matrix, we control block-FWER in a strong sense. We propose algorithm for solving graphical SLOPE using ADMM. We prove its convergence. We further construct new algorithm for solving regular SLOPE problem using ADMM. We show in simulation study that it is superior to current FISTA implementation when columns of design matrix are highly correlated or they vary significantly in variance. Finally we perform extensive simulation study in which we compare gSLOPE to glasso. gSLOPE with B-H λs proves to control FDR in various settings (especially for block diagonal matrix). Proving such a property is a next step of research which is however not included in this thesis.

Estimation of the sparse inverse covariance (a.k.a. precision) matrix of multivariate Gaussian variables has been studied quite actively in recent years, as it provides a practical tool to understand statistical relations of variables in complex data in forms of a simple undirected graph, which often reveals meaningful interactions of genes, users, news articles, operational parts of a human driver and so on. We consider problem of estimating precision matrix, inverse covariance matrix, Θ from a sample drawn from normal distribution with unknown covariance matrix Σ. This is equivalent to estimating structure of graphical model. Non-zero elements in precision matrix correspond to the vertices that share an edge. We assume that only few elements of Θ are nonzero which makes it reasonable to use regularization in estimation process.

5.1 Regularization

The idea of regularization is simple. As we assume sparsity of a graph, we want to impose penalty on the number of edges. This can be done mathematically formalized, analogously to AIC and BIC, by using ℓ_0-based penalty.

$$R_0(X) = \sum_{i \neq j} \mathbf{I}[x_{ij} \neq 0].$$

where $X \in M_{ptimesp}$. Adding it to the likelihood we obtain the following optimization problem

$$\widehat{\Theta} \in \underset{\substack{\Theta \in \mathbf{S}_+^p \\ \rho_0(\Theta) \leq k}}{\arg\max} \{\log\det\Theta - \operatorname{Tr}(\mathbf{S}\Theta)\} \tag{5.1}$$

Just as in regression that was described in the section (2.3.7) ℓ_0-based constraint defines a discrete, nonconvex problem. The problem is intractable, as finding the optimal Θ requires

checking all $\binom{\binom{p}{2}}{k}$ possible subsets of $k \in \{1, \ldots, p\}$ edges. One might of course mimic heuristic, greedy algorithms used for 'solving' BIC. This has a downside, that we are never sure if a solution found is optimal. Since number of parameters grows quadratic with respect to p, even this simplified process is computationally complex.

5.1.1 Graphical lasso

Natural relaxation of ℓ_0 constraint are ℓ_1 based methods like graphical lasso [Banerjee et al., 2008a] and [Friedman et al., 2008b]. This method is formulated as a solution to the unconstrained, convex optimization problem:

$$\hat{\Sigma}^{-1} = \arg\min_{X \succ 0} \; -\log det(X) + \mathrm{Tr}\,(\mathbf{S}X) + \lambda \|X\|_1 \tag{5.2}$$

Different variations of problem (5.2) have been addressed quite extensively in machine learning, e.g. d'Aspremont et al. [2008]; Banerjee et al. [2008b]; Friedman et al. [2008a]; Oztoprak et al. [2012]; Rolfs et al. [2012]; Hsieh et al. [2011, 2012, 2013]; Mazumder and Hastie [2012]; Treister and Turek [2014] just to name a few. More theoretical results on the statistical quality of the estimate have been appearing, e.g. Meinshausen and Bühlmann [2006, 2010]; Yuan and Lin [2007]; Banerjee et al. [2008b]; Rothman et al. [2008]; Lam and Fan [2009]; Raskutti et al. [2009]; Yuan [2010], as new properties become available for the closely related ℓ_1-penalized regression in vector spaces. From these we shall look a bit more closely on two results, that are of most interest for us.

Definition 5.1.1 (FWER for connected components) *Let graph structure be comprised of several disjoint connected components. C_k is true connectivity component of k-th vertex, while $\tilde{C}_k$ is connectivity component of k-th vertex obtained by graph structure estimator. Then by family wise error rate for connected components we denote the probability*

$$P(\exists k \, \tilde{C}_k \nsubseteq C_k) \leq \alpha$$

Let us note that graph having more than one connected components is equivalent to the true precision matrix Θ being block diagonal. In such a case [Banerjee et al., 2008a] suggested a choice of λ, so that probability of falsely joining any two blocks is controlled. More specifically they showed:

Theorem 5.1.2 (from [Banerjee et al., 2008a]) *For any $0 < \alpha < 1$, if we perform glasso using*

$$\lambda(\alpha) = \max_{i>j} \tilde{\sigma}_i \tilde{\sigma}_j \frac{t_{n-2}(\alpha/p^2)}{\sqrt{n-2+t^2_{n-2}(\alpha/p^2)}}$$

where $t_{n-2}(\alpha)$ denotes $1-\alpha$ quantile from Student's distribution with $n-2$ degrees of freedom. Then FWER for connected components is controlled:

$$P(\exists k \, C^{\lambda}_k \nsubseteq C_k) \leq \alpha$$

where C^{λ}_k is connectivity component of glasso estimator with parameter λ.

Another property of interest is consistency, by which we mean recognizing true support of Θ i.e. discovering all the edges in graphical model (with no false positives) when number of observations grows to infinity. Proof of the consistency of glasso for a specific kind of λ can be found in [Ravikumar et al., 2008]. Let us also note, that there is a proof of consistency for a similar procedure by [Meinshausen and Bühlmann, 2006]. This method comprises fitting lasso multiple times. In each fit, one variable is treated as response, while all the others form design matrix. Zero coefficient in regression fit is equivalent to conditional independence, no edge between variables. This way, this method can be used to estimate structure of graphical model. In both of these proofs λs are asymptotically (in p and n) equivalent, and they are both of rate $\frac{\log p}{\sqrt{n}}$.

5.2 Ordered L1 relaxation

In this section method glasso [Banerjee et al., 2008a] is generalized by using sorted ℓ_1 (2.48), which exhibits attractive properties such as false discovery rate (FDR) control Bogdan et al. [2015b]; Brzyski et al. [2015] and clustering of similar coefficients Bondell and Reich [2008]; Figueiredo and Nowak [2016]. We show the strategy for choosing regularization parameters λ, so that we control probability of falsely joining connected components in the graph, while getting higher power, than [Banerjee et al., 2008a].

Definition 5.2.1 *For multivariate data $\mathbf{X} \in M_{n\times p}$, we define graphical SLOPE estimator of precision matrix $\widehat{\Theta}_{gslope}$ as a solution to the following optimization problem*

$$\widehat{\Theta}_{gslope} = \underset{X\succ 0}{\arg\max}\ \log det(X) - \mathrm{Tr}\,(\mathbf{S}X) - J_\lambda(X) \tag{5.3}$$

where $\mathbf{S}$ is sample covariance matrix and J_λ is element wise sorted ℓ_1 norm.

Just like (5.2), (5.3) is convex optimization problem which can be efficiently solved. We show an algorithm based on ADMM is section 5.3.3. The name graphical SLOPE (gslope) refers to [Bogdan et al., 2015b] which was an inspiration for changing standard ℓ_1 norm to its sorted version.

5.2.1 Dual problem

To prove the properties of (5.2.1) we must first consider its dual problem.

Lemma 5.2.2 *Dual problem to the graphical SLOPE* (5.2.1) *has the following form*

$$\max_{J_\lambda^D(W-\mathbf{S})\leq 1} \log det(W) \tag{5.4}$$

Proof Let us start by rewriting SL1 norm in terms of its dual norm, which we shall denote by J_λ^D. Putting standard formula into (5.3)

$$J_\lambda(X) = \max_{J_\lambda^D(U)\leq 1} \mathrm{Tr}\,(UX)$$

yields

$$\max_{X \succ 0} \log det(X) - \mathrm{Tr}\left(\mathbf{S}X\right) - \max_{J_\lambda^D(U) \leq 1} \mathrm{Tr}\left(UX\right).$$

Using the fact that trace is an additive function we get

$$\max_{X \succ 0} \min_{J_\lambda^D(U) \leq 1} \log det(X) - \mathrm{Tr}\left(X(U + \mathbf{S})\right)$$

For (5.2.1) strong duality holds (because problem is convex and Slater's condition is satisfied). Therefore min and max can be exchanged in the above.

$$\min_{J_\lambda^D(U) \leq 1} \max_{X \succ 0} \log det(X) - \mathrm{Tr}\left(X(U + \mathbf{S})\right)$$

There is a closed formula for the solution of inner maximization. We compute gradient (for matrix function gradients and more see [Minka, December 2000]) and set it to zero. This yields

$$\begin{aligned} 0 &= d(\log det(X) - \mathrm{Tr}\left(X(U + \mathbf{S})\right)) \\ &= d \log det(X) - d\,\mathrm{Tr}\left(X(U + \mathbf{S})\right) \\ &= \mathrm{Tr}\left(X^{-1}dX\right) - \mathrm{Tr}\left((U + \mathbf{S})dX\right) \\ &= \mathrm{Tr}\left((X^{-1} - (U + \mathbf{S}))dX\right) \end{aligned}$$

This is zero only when $X^{-1} - (U + \mathbf{S})$. For which $\mathrm{Tr}\left(X(U + \mathbf{S})\right) = \mathrm{Tr}\left((U + \mathbf{S})^{-1}(U + \mathbf{S})\right) = p$ and finally we get dual problem:

$$\min_{J_\lambda^D(U) \leq 1} - \log det(U + \mathbf{S}) - p$$

For the sake of notation let us rewrite $\mathbf{W} := U + \mathbf{S}$

$$\max_{J_\lambda^D(\mathbf{W} - \mathbf{S}) \leq 1} \log det(\mathbf{W})$$

∎

Dual problem (5.4) has an insightful interpretation. We want to maximize $\log det$ of a matrix W, with a constraint that $\mathbf{W}$ cannot be a perturbation on sample covariance matrix $\mathbb{S}$ larger than 1 in norm J_λ^D. When we solve graphical SLOPE (5.2.1), $\mathbf{W}$ is our estimate of covariance matrix.

5.2.2 FWER for connected components by graphical SLOPE

Suppose that Σ is block diagonal matrix.

$$\Sigma = \begin{pmatrix} \Sigma_1 & 0 & \cdots & 0 \\ 0 & \Sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \Sigma_l \end{pmatrix} \tag{5.5}$$

which is, as we noted before, equivalent to graph having l disjoint connected components. [Banerjee et al., 2008a] show how to choose λ so we can control the FWER for connected

components i.e. probability of falsely 'joining' these blocks, estimating non-zero conditional covariance outside of blocks.
We will show how to improve result by [Banerjee et al., 2008a] and also construct a λ sequence for gslope that is guaranteed to have higher power and controls FWER in a weak sense.

Definition 5.2.3 (Holm λ sequence for gslope) *For any given $0 < \alpha < 1$ we define Holm λ sequence for gslope as*

$$\lambda_{2k} = \lambda_{2k-1} = \frac{\exp \frac{2\Phi^{-1}(\frac{\alpha}{2\cdot(m+1-k)})}{\sqrt{n-3}} - 1}{1 + \exp \frac{2\Phi^{-1}(\frac{\alpha}{2\cdot(m+1-k)})}{\sqrt{n-3}}},$$

for $k = 1, \ldots, m$, $m = \frac{p(p-1)}{2}$.

λ elements come in pairs, which corresponds to correlation matrix being symmetric. First let us prove:

Theorem 5.2.4 *Applying Holm λs 5.2.3 to step down procedure on off-diagonal elements of sample correlation matrix $\mathbf{S}$ controls FWER asymptotically.*

$$FWER_{Holm\ \lambda} \leq \alpha \text{ as } n \to \infty$$

Proof Under hypothesis of zero correlation we have asymptotically with $n \to \infty$ (see [Kendall et al., 1987]):

$$\frac{1}{2} \log \left(\frac{1 + S_{i,j}}{1 - S_{i,j}} \right) \sqrt{n-3} \sim \mathcal{N}(0, 1)$$

From lemma 2.3.8, we know that step down procedure with critical values corresponding to p-values $\frac{\alpha}{m+1-k}$, $k = 1, \ldots, m$ controls FWER.

Since correlation matrix is symmetric, we have just $m = \frac{p(p-1)}{2}$ hypothesis to test. We get the following equality for the asymptotic critical values:

$$\begin{aligned}
|\frac{1}{2} \log \left(\frac{1 + \lambda_k}{1 - \lambda_k} \right) \sqrt{n-3}| &= \Phi^{-1} \left(\frac{\alpha}{2 \cdot (m+1-k)} \right) \\
\log \left(\frac{1 + \lambda_k}{1 - \lambda_k} \right) &= \frac{2\Phi^{-1}(\frac{\alpha}{2\cdot(m+1-k)})}{\sqrt{n-3}} \\
\frac{1 + \lambda_k}{1 - \lambda_k} &= \exp \left\{ \frac{2\Phi^{-1}(\frac{\alpha}{2\cdot(m+1-k)})}{\sqrt{n-3}} \right\} \\
1 + \lambda_k &= (1 - \lambda_k) \cdot \exp \frac{2\Phi^{-1}(\frac{\alpha}{2\cdot(m+1-k)})}{\sqrt{n-3}} \\
\lambda_k &= \frac{\exp \frac{2\Phi^{-1}(\frac{\alpha}{2\cdot(m+1-k)})}{\sqrt{n-3}} - 1}{1 + \exp \frac{2\Phi^{-1}(\frac{\alpha}{2\cdot(m+1-k)})}{\sqrt{n-3}}}
\end{aligned} \tag{5.6}$$

∎

Theorem 5.2.5 *Let Σ be given block diagonal correlation matrix and $\alpha \in (0,1)$. We assume further the dependency structure between elements of the sample covariance matrix is such that the Hochberg's step-up procedure controls FWER (see for example [Sarkar, 1998]). Solution to graphical SLOPE problem 5.3 for the choice of λ sequence according to* (5.2.3) *asymptotically controls FWER for connected components on level α. Furthermore, all elements of λ are smaller than the value specified for analogous result for glasso by [Banerjee et al., 2008a].*

The idea of the proof is the following. We consider dual problem instead of primal one. Because duality gap is zero, inverse of matrix $\mathbf{W}$ is maximizer of original problem. Note that if precision matrix Θ has block structure than so does correlation matrix Σ. In the first lemma, we show that with high probability estimated correlation matrix $\mathbf{W}$ is feasible in our dual optimization problem. We use Holms multiple testing correction for that. In the second lemma we show that it is beneficial for the goal function to set off-block diagonal elements to zero, as it leads to increase of determinant of the matrix $\mathbf{W}$. Hence, with high probability, solution is in agreement with block structure of true correlation matrix Σ. Please note, that the graphical SLOPE estimator might be more sparse than true precision matrix. For example when λs $\to \infty$ then solution is diagonal matrix, which has FWER equal to 0.

Lemma 5.2.6 *We use assumptions from Theorem (5.2.5). With probability $1-\alpha$, there exists a matrix W with all off-block elements equal to 0 that satisfies feasibility condition $J^D_\lambda(W-\mathbb{S}) \leq 1$.*

Proof

From the form of dual norm to sorted ℓ_1 we know that If $\forall_k |W-\mathbb{S}|_{(k)} \leq \lambda_k$ then

$$J^D_\lambda(W-\mathbb{S}) = \max\left\{\frac{|W-\mathbb{S}|_{(1)}}{\lambda_1}, \ldots, \frac{\sum_{k=1}^p |W-\mathbb{S}|_{(k)}}{\sum_{k=1}^p \lambda_k}\right\} \leq 1$$

and as a consequence, treating the following as random events

$$\{(^D_\lambda(W-\mathbb{S}) \leq 1\} \supseteq \{\forall_{k=1,\ldots,p^2} \quad |W-\mathbb{S}|_{(k)} \leq \lambda_k\} \tag{5.7}$$

So by setting λs that satisfy right hand side of (5.7) with given probability we get a lower bound on probability that W is feasible.
For correlation estimate W which recovers true block structure, we have $W_{i,j} = 0$ for i and j in separate blocks. We want to choose λs to that for all such (i,j) $|\mathbb{S}_{i,j}|$ is smaller than corresponding λ. In worst case scenario this has to hold for all $m := \binom{p}{2}$ different entries in sample correlation matrix $\mathbb{S}$. We exclude diagonal elements which are obviously non-zero, and use symmetry of $\mathbb{S}$.
So, we need $|S_{i,j}|_{(k)} \leq \lambda_k$.

Now, we test it. If we assume that we scaled our data, we have correlation matrix $\mathbf{S}$ equal to correlation matrix. Given that, under hypothesis of zero correlation we have asymptotically with $n \to \infty$ (see [Kendall et al., 1987]):

$$\frac{1}{2}\log\left(\frac{1+S_{i,j}}{1-S_{i,j}}\right)\sqrt{n-3} \sim \mathcal{N}(0,1)$$

Alternatively, one might use different limit distribution

$$\sqrt{n-2}\frac{S_{i,j}}{\sqrt{1-S_{i,j}^2}} \sim t(n-2)$$

where $t(n-2)$ is Student distribution with $n-2$ degrees of freedom.
This transformation is monotone in $S_{i,j}$. What's more, taking absolute value on $S_{i,j}$ is equivalent to taking absolute value over whole term.

$$\frac{1}{2}\log\left(\frac{1+|S_{i,j}|}{1-|S_{i,j}|}\right)\sqrt{n-3} \leq \frac{1}{2}\log\left(\frac{1+\lambda_k}{1-\lambda_k}\right)\sqrt{n-3} \tag{5.8}$$

We shall apply Holm-Bonferroni correction for multiple testing. So we want

$$P(Z \geq \frac{1}{2}\log\left(\frac{1+\lambda_k}{1-\lambda_k}\right)\sqrt{n-3}) = \frac{\alpha}{2\cdot(m+1-k)} \tag{5.9}$$

where Z is r.v. following standard normal distribution. Term two in nominator comes from taking absolute value.
Solving for λ_k yields,

$$\begin{aligned}
\frac{1}{2}\log\left(\frac{1+\lambda_k}{1-\lambda_k}\right)\sqrt{n-3} &= \Phi^{-1}\left(\frac{\alpha}{2\cdot(m+1-k)}\right) \\
\log\left(\frac{1+\lambda_k}{1-\lambda_k}\right) &= \frac{2\Phi^{-1}(\frac{\alpha}{2\cdot(m+1-k)})}{\sqrt{n-3}} \\
\frac{1+\lambda_k}{1-\lambda_k} &= \exp\left\{\frac{2\Phi^{-1}(\frac{\alpha}{2\cdot(m+1-k)})}{\sqrt{n-3}}\right\} \\
1+\lambda_k &= (1-\lambda_k)\cdot\exp\frac{2\Phi^{-1}(\frac{\alpha}{2\cdot(m+1-k)})}{\sqrt{n-3}} \\
\lambda_k &= \frac{\exp\frac{2\Phi^{-1}(\frac{\alpha}{2\cdot(m+1-k)})}{\sqrt{n-3}}-1}{1+\exp\frac{2\Phi^{-1}(\frac{\alpha}{2\cdot(m+1-k)})}{\sqrt{n-3}}}
\end{aligned} \tag{5.10}$$

We assumed that Σ is such, that test statistics satisfy condition for Hochberg's step up procedure to control FWER. Hochberg's step-up procedure rejects all hypothesis with p-values smaller than the minimal index k satisfying $p_{(k)} \leq \Phi^{-1}\left(\frac{\alpha}{2\cdot(m+1-k)}\right)$. Therefore it rejects more hypothesis than our procedure does, so our FWER is smaller than α.
If data is not scaled, then one can multiple λ sequence by $\max_{i>j}\tilde{\sigma}_i\tilde{\sigma}_j$, which guarantees that (5.8) also when $S_{i,j}$ is covariance rather than correlation.

∎

We can get λs more similar to Banerjee if we use Student approximation rather than normal approximation for entries of correlation matrix.

We've showed that choosing λs properly, results in true block structure matrix feasible in our optimization problem with high probability. Now, we show that, if true Θ is block diagonal, then objective function grows when off-block-diagonal elements are set to 0.

Lemma 5.2.7 *Assume that matrix with true block-diagonal structure in problem* (5.4) *is feasible. Then objective function in problem* (5.4) *is maximized for such a block-diagonal matrix* W*.*

Proof

Observe that

$$\frac{\partial \log |W|}{\partial W} = W^{-1} \tag{5.11}$$

Note that when W is block-diagonal, then also W^{-1} is block-diagonal. So, the gradient at all off diagonal elements is equal to zero. Because such W is invertible and feasible and because problem (5.4) is convex, optimal solution has to have a block structure. ∎

Proof of Theorem (5.2.5) From lemma (5.2.6), with high probability, block-diagonal matrix is feasible. By second lemma it also maximizes objective function, so it is a solution to the dual problem. Because for gSLOPE strong duality holds, optimal values of primal and dual are the same (see [Boyd and Vandenberghe, 2004]) and inverse of matrix W is solution to optimization problem (5.3). ∎

Corollary 5.2.8 *Theorem 5.2.5 has an important consequence. Because all elements of* λ *sequence where smaller than in [Banerjee et al., 2008a], penalty term with those* λ*s in graphical smaller is more liberal than glasso. Therefore graphical SLOPE finds at least as many edges as glasso.*

Corollary 5.2.9 *Theorem 5.2.5 can be rewritten to find smaller* λ *than the one in [Banerjee et al., 2008a].*

$$\lambda(\alpha) = \max_{i>j} \tilde{\sigma}_i \tilde{\sigma}_j \frac{\exp \frac{2\Phi^{-1}(\frac{\alpha}{2\cdot m})}{\sqrt{n-3}} - 1}{1 + \exp \frac{2\Phi^{-1}(\frac{\alpha}{2\cdot m})}{\sqrt{n-3}}}, \tag{5.12}$$

where $m = \dfrac{p(p-1)}{2}$ *or*

$$\lambda(\alpha) = \max_{i>j} \tilde{\sigma}_i \tilde{\sigma}_j \frac{t_{n-2}(\alpha/(2\cdot m))}{\sqrt{n-2+t^2_{n-2}(\alpha/(2\cdot m))}}$$

depending on which approximate distribution for the entries in precision matrix we use. Because Bonferroni correction is one step procedure, for such λ *strong FWER is controlled. We shall use this* λ *in simulation study.*

Corollary 5.2.10 *Examples of when step-up procedure controls FWER can be found for example in [Sarkar, 1998]. In particular it does for positive equicorrelated multivariate normal distributions [Steck and Owen, 1962]. This type of dependency in multivariate normal is covered in our simulation study.*

5.3 Solving optimization problem

In the previous section we focused on theoretical properties of graphical SLOPE for specific λ sequence. Yet, for the method to be useful in practice, an efficient method for computing it is required.

5.3.1 ADMM for SLOPE

We used alternative direction method of multipliers (ADMM) to construct new algorithm for solving SLOPE (formulation and more details here (2.49)). Current implementation using FISTA proves to have poor efficiency when, for example, columns of design matrix $\mathbf{X}$ are highly correlated.
To derive algorithm, we first reformulate (2.49) to the form that resembles more the standard one for ADMM (2.11).

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \frac{1}{2}(b-\mathbf{A}x)^T(b-\mathbf{A}x) + \sigma J_\lambda(y_j) \\ \text{subject to} \quad & x = y \end{aligned} \tag{5.13}$$

Then augmented Lagrangian with penalty parameter μ is

$$\mathcal{L}_\mu(x,y,z) = \frac{1}{2}(b-\mathbf{A}x)^T(b-\mathbf{A}x) + \sigma J_\lambda(y) + \mu\langle x-y, z\rangle + \frac{\mu}{2}\|x-y\|_F^2 \tag{5.14}$$

Optimization takes place in turns, with respect to x and y.

$$\begin{aligned} x_k &= \text{argmin}_x \mathcal{L}_\mu(x, y_{k-1}, z_{k-1}) \\ &= \text{argmin}_x \frac{1}{2}\|b-\mathbf{A}x\|^2 + \mu < z_{k-1}, x > + \frac{\mu}{2}\|x-y_{k-1}\|^2 \end{aligned}$$

Setting derivative to zero we get

$$0 = \nabla_x \mathcal{L}_\mu(x, y_{k-1}, z_{k-1}) = -\mathbf{A}^T(b-\mathbf{A}x) + \mu z_{k-1} + \mu(x-y_{k-1})$$

Equivalently:

$$\begin{aligned} \mathbf{A}^T(b-\mathbf{A}x) &= \mu z_{k-1} + \mu(x-y_{k-1}) \\ \mathbf{A}^Tb - \mathbf{A}^T\mathbf{A}x) &= \mu(z_{k-1}-y_{k-1}) + \mu x \\ -\mu x - \mathbf{A}^T\mathbf{A}x) &= \mu(z_{k-1}-y_{k-1}) - \mathbf{A}^Tb \\ (\mathbf{A}^T\mathbf{A}+\mu\mathbf{I})x &= \mu(y_{k-1}-z_{k-1}) + \mathbf{A}^Tb \\ x &= (\mathbf{A}^T\mathbf{A}+\mu\mathbf{I})^{-1}\mu(y_{k-1}-z_{k-1}) + \mathbf{A}^Tb \end{aligned}$$

For y it is easier:

$$\begin{aligned} y_k &= \text{argmin}_y \mathcal{L}_\mu(x_k, y, z_{k-1}) \\ &= \text{argmin}_y \sigma J_\lambda(y) - \mu\langle y, z_{k-1}\rangle + \frac{\mu}{2}\|x_k-y\|_F^2 \\ &= \text{argmin}_y \sigma J_\lambda(y) + \frac{\mu}{2}\|x_k-(y+z_{k-1})\|_F^2 \end{aligned}$$

This can efficiently solved using prox function defined by [Bogdan et al., 2015a].

$$y_k = \text{prox}_{\sigma\lambda/\mu}(x_k + z_{k-1})$$

Algorithm

In this section we present detailed description of algorithms for solving SLOPE using ADMM.

Algorithm 9 ADMM for SLOPE

$y_0 \leftarrow \tilde{y}$. $z_0 \leftarrow \tilde{z}$, $k \leftarrow 1$, $\mu \leftarrow \mu_0 > 0$, {algorithm initialization}
while convergence criterion is not satisfied **do**
  $x_{k+1} = (\mathbf{A}^T\mathbf{A} + \mu\mathbf{I})^{-1}\mu(y_k - z_k) + \mathbf{A}^T b$
  $y_{k+1} = \text{prox}_{\sigma\lambda/\mu}(x_{k+1} + z_k)$
  $z_{k+1} := z_k + \mu(x_{k+1} - y_{k+1})$
end while

Comparing SLOPE FISTA vs ADMM

In this section we compare two implementations of SLOPE. First is FISTA-based present in R package SLOPE. Second is ADMM written also in R.
Number of variables is 400, number of observations is 100. Design matrix is generated from multivariate normal distribution with covariance matrix having diagonal elements equal 1 and off-diagonal elements all equal to ρ. Big ρ corresponds to highly correlated columns in design matrix. There are 5 variables with non-zero coefficient that are randomly selected. Response vector is selected according to linear model with error drawn from normal distribution with $\sigma^2 = 1$.

Table 5.1: SLOPE FISTA vs ADMM. Highly correlated columns. Time comparison, ale times in seconds

ρ	Mean FISTA	Standard deviation FISTA	Mean ADMM	Standard deviation ADMM
0.20	0.53	0.43	0.12	0.00
0.50	1.82	1.04	0.12	0.01
0.80	1.88	1.72	0.11	0.02

ADMM implementation proves to be much more stable. Execution time for FISTA implementation depends on covariance structure.
We also performed experiment in which we compare performance, when variance of columns in data matrix $\mathbf{X}$ differs significantly. This is a typical setup expected if SLOPE is be used in adaptive way.
Again ADMM implementation proves to be much more stable. Execution time for FISTA implementation significantly depends on the difference in variance for columns of design matrix.

5.3.2 ADMM for gSLOPE

The solution to (5.2.1) can be found using alternative direction method of multipliers (ADMM). Details about this method can be found in chapter with mathematical introduction.

We reformulate 5.3 to the form that resembles more the standard form for ADMM (2.11).

Table 5.2: SLOPE FISTA vs ADMM. Columns with significant differences in variance. Time comparison, ale times in seconds

Max diff in variance	Mean FISTA	Sd FISTA	Mean ADMM	Sd ADMM
10^1	0.13	0.03	0.10	0.01
10^2	0.19	0.05	0.12	0.02
10^3	0.30	0.10	0.12	0.02
10^4	0.65	0.37	0.13	0.01

$$\begin{aligned} \underset{X}{\text{minimize}} \quad & -\log det(X) + \text{Tr}\left((\mathbf{S}X) + \mathcal{I}_{X\succ 0}(X) + J_\lambda(Y)\right. \\ \text{subject to} \quad & X = Y \end{aligned} \tag{5.15}$$

Then augmented Lagrangian with penalty parameter μ is

$$\mathcal{L}_\mu(X, Y, Z) = \log det(X) + \text{Tr}\left(\mathbf{S}X\right) + \mathcal{I}_{X\succ 0}(X) + J_\lambda(Y) + \mu\langle X - Y, Z\rangle + \frac{\mu}{2}\|X - Y\|_F^2 \tag{5.16}$$

Now, optimization takes place in turns, with respect to X and Y.

$$\begin{aligned} X_k &= \text{argmin}_X \mathcal{L}_\mu(X, Y_{k-1}, Z_{k-1}) \\ &= \text{argmin}_X - \text{logdet}X + < X, \mathbf{S} > + \|Y_{k-1}\|_{J_\lambda} + \mu < Z_{k-1}, X - Y_{k-1} > + \frac{\mu}{2}\|X - Y_{k-1}\|_F^2 \\ &= \text{argmin}_X - \text{logdet}X + < X, \mathbf{S} > + \mu < Z_{k-1}, X > + \frac{\mu}{2}\|X - Y_{k-1}\|_F^2 \\ &= \text{argmin}_X - \text{logdet}X + \frac{\mu}{2}\|X + (Z_{k-1} + \frac{1}{\mu}\mathbf{S} - Y_{k-1})\|_F^2 \end{aligned}$$

Let us denote $\tilde{\mathbf{S}}_{k-1} = Z_{k-1} + \frac{1}{\mu}\mathbf{S} - Y_{k-1}$. This can be viewed as approximated covariance matrix, that is why we use this notation.

Let us take derivative of augmented logarithm with respect to X and look for $X*$ that sets it 0.

$$\nabla_X \mathcal{L}_\mu(X, Y_{k-1}, Z_{k-1}) = -X^{-1} + \mu X - \mu\tilde{\mathbf{S}}_{k-1}$$

We do eigenvalue decomposition of $tilde\mathbf{S}_{k-1}$

$$\tilde{\mathbf{S}}_{k-1} = U\Lambda U^T, \;\; \Lambda = diag(\lambda_i)$$

Observe that

$$X^* = U diag\left(\frac{1}{2}(\lambda_i + \sqrt{\lambda_i^2 + \frac{4}{\mu}})\right) U^T$$

Solves our gradient equation:

$$
\begin{aligned}
\nabla_X \mathcal{L}_\mu(X, Y_{k-1}, Z_{k-1}) &= -X^{-1} + \mu X - \mu \tilde{\mathbf{S}}_{k-1} \\
&= -Udiag\left(\frac{1}{\frac{1}{2}(\lambda_i + \sqrt{\lambda_i^2 + \frac{4}{\mu}})}\right) U^T + Udiag\left(\mu \frac{1}{2}(\lambda_i + \sqrt{\lambda_i^2 + \frac{4}{\mu}})\right) U^T - \mu U \Lambda U^T \\
&= Udiag\left(-\frac{1}{\frac{1}{2}(\lambda_i + \sqrt{\lambda_i^2 + \frac{4}{\mu}})} + \mu \frac{1}{2}(\lambda_i + \sqrt{\lambda_i^2 + \frac{4}{\mu}}) - \mu\lambda_i\right) U^T \\
&= Udiag\left(-2\frac{\lambda_i - \sqrt{\lambda_i^2 + \frac{4}{\mu}}}{(\lambda_i - \sqrt{\lambda_i^2 + \frac{4}{\mu}})(\lambda_i + \sqrt{\lambda_i^2 + \frac{4}{\mu}})} - \mu\frac{1}{2}\lambda_i + \frac{1}{2}\mu\sqrt{\lambda_i^2 + \frac{4}{\mu}})\right) U^T \\
&= Udiag\left(-2\frac{\lambda_i - \sqrt{\lambda_i^2 + \frac{4}{\mu}}}{\lambda_i^2 - \lambda_i^2 - \frac{4}{\mu}} - \mu\frac{1}{2}\lambda_i + \frac{1}{2}\mu\sqrt{\lambda_i^2 + \frac{4}{\mu}})\right) U^T \\
&= Udiag\left(\frac{1}{2}\mu\left(\lambda_i - \sqrt{\lambda_i^2 + \frac{4}{\mu}}\right) - \mu\frac{1}{2}\lambda_i + \frac{1}{2}\mu\sqrt{\lambda_i^2 + \frac{4}{\mu}})\right) U^T \\
&= 0
\end{aligned}
$$

Optimization with respect to Y is somewhat less complicated.

$$
\begin{aligned}
Y_k &= \text{argmin}_Y L_\mu(X_k, Y, Z_{k-1}) \\
&= \text{argmin}_Y - \text{logdet} X_k + < X_k, C > + \|Y\|_{J_\lambda} + \mu < Z_{k-1}, X_k - Y > + \frac{\mu}{2}\|X_k - Y\|_F^2 \\
&= \text{argmin}_Y \|Y\|_{J_\lambda} + \frac{\mu}{2}\|Y - (X_k + Z_{k-1})\|_F^2
\end{aligned}
$$

This means that we can update Y_k using prox function from [Bogdan et al., 2015a] just like we did in case of SLOPE.

$$Y_k = \text{prox}_{\lambda/\mu}(X_k + Z_{k-1})$$

5.3.3 Algorithm

In this section we present detailed description of algorithms for solving graphical SLOPE using ADMM.

Algorithm 10 ADMM for gslope - non-scaled version

$Y_0 \leftarrow \tilde{Y}$. $Z_0 \leftarrow \tilde{Z}$, $k \leftarrow 1$, $\mu \leftarrow \mu_0 > 0$, {algorithm initialization}
while convergence criterion is not satisfied **do**
  Perform eigenvalue decomposition of matrix $Z_k + \frac{1}{\mu}\mathbf{S} - Y_k = Udiag(\lambda_i)U^T$
  $X_{k+1} := Udiag\left(\frac{1}{2}(\lambda_i + \sqrt{\lambda_i^2 + \frac{4}{\mu}})\right) U^T$
  $Y_{k+1} := \text{prox}_{\lambda/\mu}(X_{k+1} + Z_k)$
  $Z_{k+1} := Z_k + \mu(X_{k+1} - Y_{k+1})$
end while

or in a scaled form using $V = \frac{1}{\mu}Z$

Algorithm 11 Simulation scheme for a signal matrix with equal singular values

$U_0 \leftarrow \tilde{U}$. $Z_0 \leftarrow \tilde{Z}$, $k \leftarrow 1$, $\mu \leftarrow \mu_0 > 0$, {algorithm initialization}
while convergence criterion is not satisfied **do**
  Perform eigenvalue decomposition of matrix $\mu V_k + \frac{1}{\mu}\mathbf{S} - Y_k = Udiag(\lambda_i)U^T$
  $X_{k+1} := Udiag\left(\frac{1}{2}(\lambda_i + \sqrt{\lambda_i^2 + \frac{4}{\mu}})\right)U^T$
  $Y_{k+1} := \text{prox}_{\lambda/\mu}(X_{k+1} + \mu V_k)$
  $U_{k+1} := U^k + X_{k+1} - Y_{k+1}$
end while

As proved by [Boyd et al., 2011], sufficient convergence criterion is that primal and dual feasibility is met.

Theorem 5.3.1 *Algorithm (10) converges to the solution of the problem (5.2.1)*

Proof
We shall check that conditions for convergence from [Boyd et al., 2011] are met. Firstly, notice that both functions that are part of goal $-\log det(X) + trace(SX) + \mathcal{I}_{X\succ 0}(X)$ and $J_\lambda(Y)$ are obviously convex, closed and proper.
Remaining part is showing that unaugmented Lagrangian has a saddle point

$$\log det(X) - \text{Tr}\left(\mathbf{S}X\right) - \max_{J_\lambda^D(U)\leq 1} \text{Tr}\left(UX\right)$$

As assumptions for the Saddle Point Theorem (see [Boyd and Vandenberghe, 2004]) are strong duality and Slater's condition, we know that Lagrangian has saddle point, as we previously proved that these two are satisfied.

∎

5.4 Simulations

In this section, we present the result of simulation studies, in which we compare Graphical SLOPE with other methods of graph estimation. To measure the quality of the procedures we use their effectiveness and execution time. We also present time comparison for two implementations of SLOPE.

5.4.1 Methods

In our simulations we compare following methods:

1. Graphical Lasso with λ from [Banerjee et al., 2008a]
2. Graphical Lasso with enhanced choice of λ
3. Graphical SLOPE with λ sequence inspired by Holm correction for multiple testing
4. Graphical SLOPE with λ sequence inspired by Benjamini-Hochberg correction for multiple testing

5.4.2 Simulation scenarios

Scenario 1. In the first scenario we evaluated performance in block-diagonal precision matrix. Set of variables is divided into a number of blocks. Elements from separate blocks are conditionally independent. Within each block all partial correlations are nonzero. We tested different number of variables, observations, value of nonzero partial correlations and α levels. Example of such graph can be seen in figure (5.1).

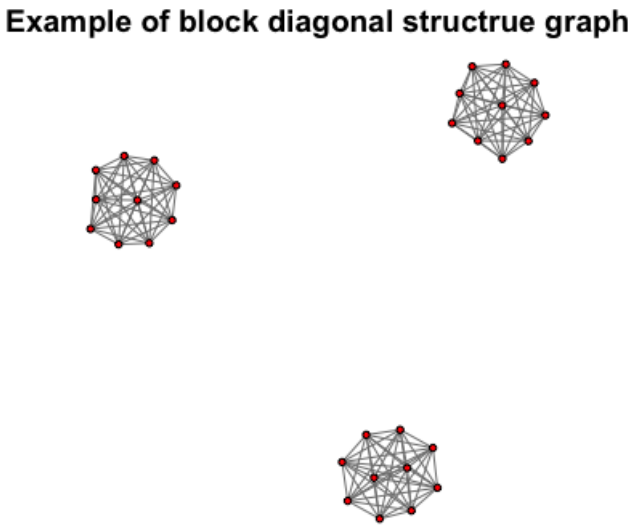


Figure 5.1: Example of block diagonal graph structure

Algorithm 12 Simulation scheme for a block-diagonal precision matrix

Input: Number of observations n, number of variables p, α, value of off-diagonal entry in precision matrix ρ and block size b

1: Create diagonal matrix Θ
2: For each block, set every entry of the precision matrix Θ within this block to ρ
3: Generate data using multivariate normal distribution with mean equal zero vector, and covariance matrix equal $\Sigma = \Theta^{-1}$.

Scenario 2. Second scenario is for hub structure precision matrix. Set of all variables is divided into subgroups of size 10. Groups are partially independent. Within groups, one variable is connected to all the others. Rest are partially independent. Example of such graph can be seen in figure (5.2).

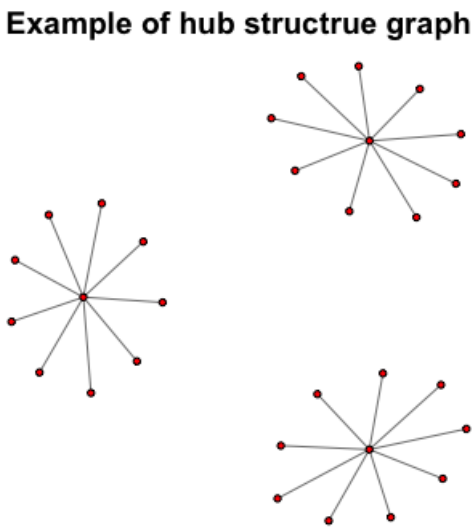


Figure 5.2: Example of hub graph structure

Algorithm 13 Simulation scheme for a hub precision matrix

Input: Number of observations n, number of variables p, α, value of off-diagonal entry in precision matrix ρ and size of hub is 10.

1: Create diagonal matrix Θ
2: For each hub, choose one variable. Set every entry of the precision matrix Θ, that is associated with this variable, within this hub to ρ
3: Generate data using multivariate normal distribution with mean equal zero vector, and precision matrix equal $\Sigma = \Theta^{-1}$.

Scenario 3. Third scenario banded structure. All but first and last variables have nonzero partial correlation with two other variables, its predecessor and successor. First and last variable are 'connected' to just one variable. Example of such graph can be seen in figure (5.3).

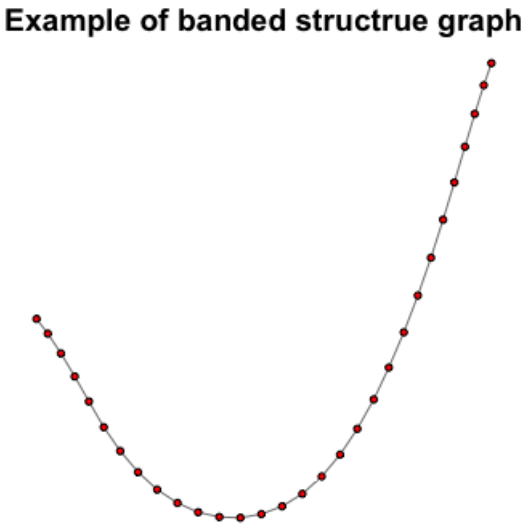


Figure 5.3: Example of banded graph structure

Algorithm 14 Simulation scheme for a banded precision matrix

Input: Number of observations n, number of variables p, α, value of off-diagonal entry in precision matrix ρ and size of hub is 10.

Create diagonal matrix Θ
for $i \in \{1, \ldots, p\}$ **do**
  if $i > 0$ **then**
    $\theta_{i-1,i} = \theta_{i,i-1} := \rho$
  end if
  if $i < p$ **then**
    $\theta_{i,i+1} = \theta_{i+1,i} := \rho$
  end if
end for
Generate data using multivariate normal distribution with mean equal zero vector, and precision matrix equal $\Sigma = \Theta^{-1}$.

5.4.3 Estimation of performance metrics

To compare graph estimation we use three measures. Let us use the following notation E is the set of all edges, D is the set of edges discovered by some procedure. E_i and D_i denotes set of true and discovered edges for i^{th} variable

1. power, fraction of edges in the graph discovered by method

$$\text{power} := \frac{|E \cap D|}{|D|} \tag{5.17}$$

2. false discovery rate (FDR)

$$\text{FDR} := \frac{|D \setminus E|}{|D|}, \tag{5.18}$$

fraction of wrong discoveries to all discoveries

3. local false discovery rate (local FDR)

$$\text{local FDR} := \frac{\sum_{i=1}^{p} \frac{Di \setminus Ei|}{|Di|}}{p}, \tag{5.19}$$

average fraction of wrong discoveries to all discoveries per one variable

4. family-wise error rate (FWER)

$$\text{FWER} := |D \setminus E| > 0 \tag{5.20}$$

In the following sections, we report average value of these metrics calculated in each simulation repetition.

5.4.4 Block diagonal matrix

In this simulation, data is drawn 1000 times according to the algorithm (12).
We compared performance for a grid of parameters

1. numbers of variables in the data set, varying from 60 to 200
2. number of observations in the data set, varying from 100 to 800
3. value of nonzero elements in precision matrix $\rho \in [0.2, 0.7]$
4. nominal FDR/FWER level in range $\alpha \in (0.05, 0.2)$

Full simulations results are available online. Here we show and comment on selected results.

When the number of observations grows, then the signal is relatively stronger, and all methods have higher power. Recall that λs from Banerjee et al. [2008a] are getting smaller when n grows. Therefore we expect that power of any statistical method to grow with n. This intuition is backed up by the simulation results, see e.g. (5.8). Because glasso is consistent under certain assumptions, one might expect that FDR should go to zero as n grows. In fact this is what we observe. What is more interesting, in multiple scenarios, FDR is controlled for gSLOPE with BH sequence. We do not provide any theoretical justification for this fact, but it seems, that we λ sequence is selected in a proper way. The bigger is the value on nonzero elements in precision matrix, the higher power is achieved by all of the methods.

5.4.5 Hub structure matrix

In this simulations, data is drawn 100 times according to the algorithm (13).
We compared performance for a grid of parameters

1. numbers of variables in the data set, varying from 30 to 200

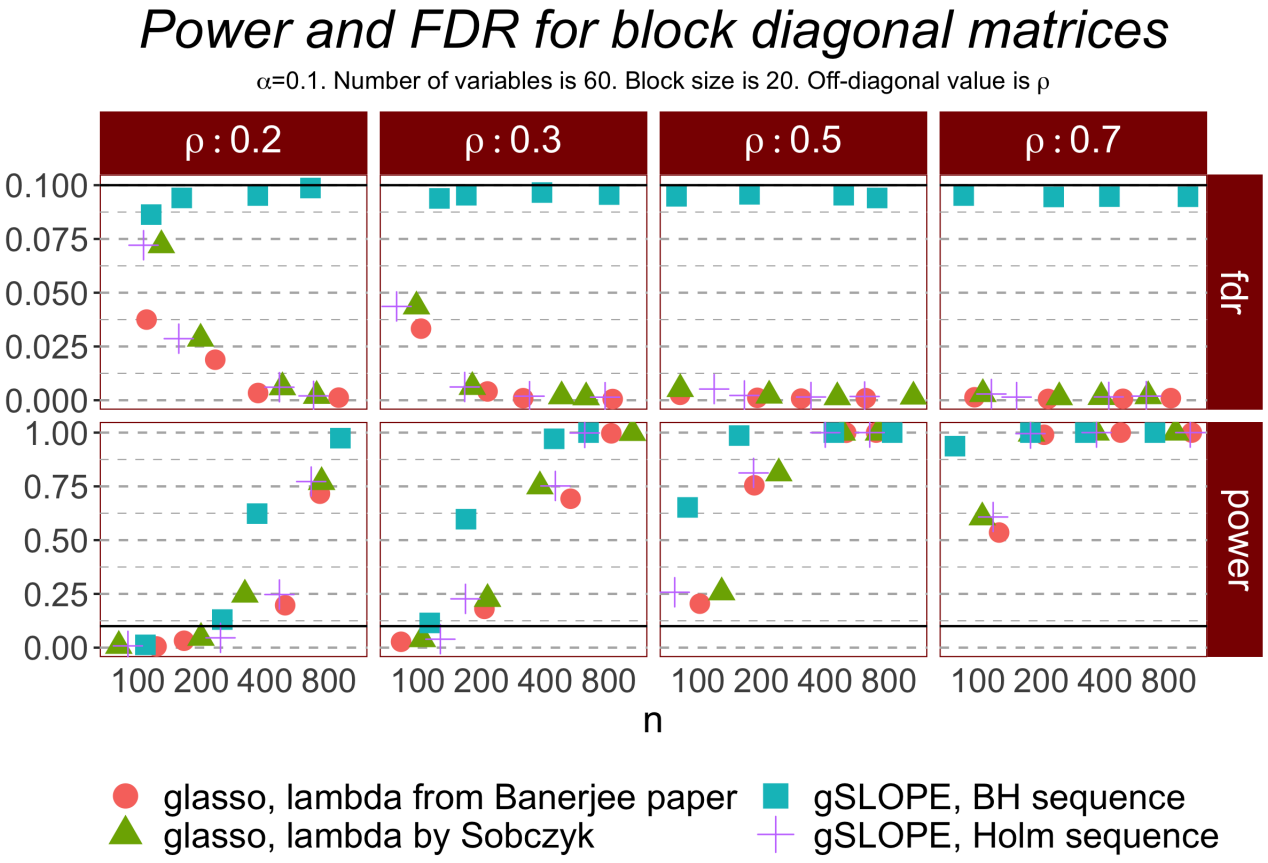


Figure 5.4: Graphical SLOPE. Block diagonal matrix. p is small (60), α is average - 0.1

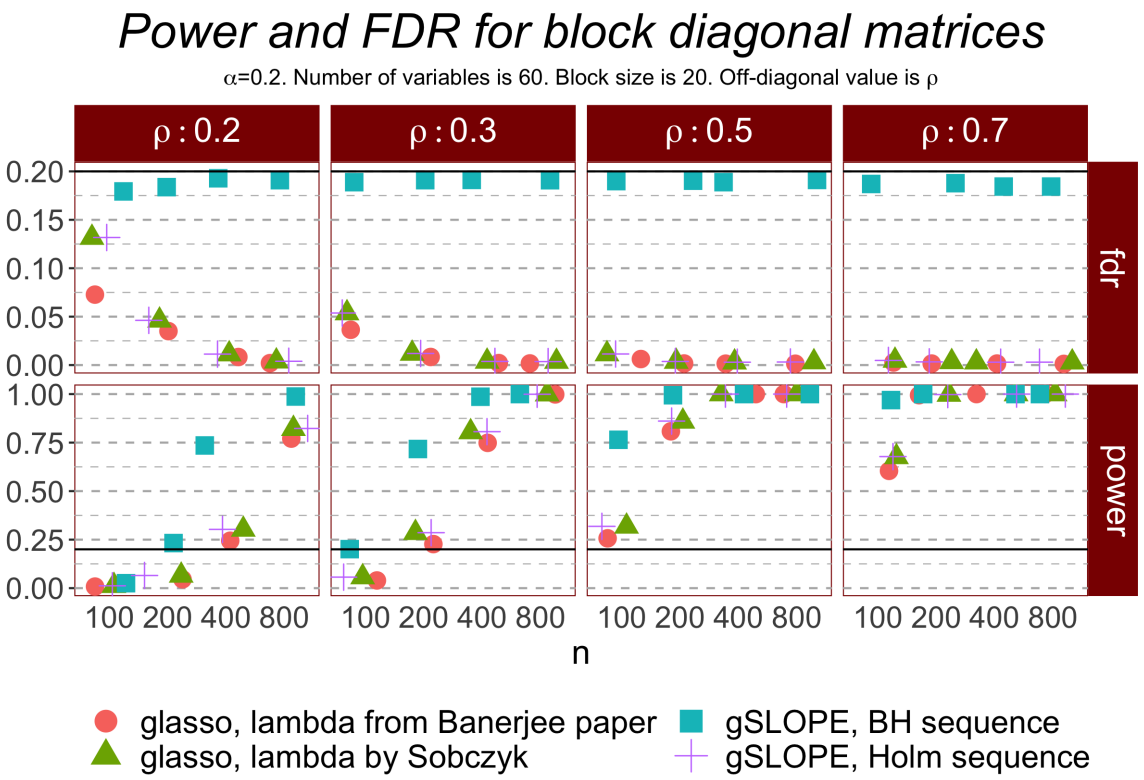


Figure 5.5: Graphical SLOPE. Block diagonal matrix. p is small (60), α is big - 0.2

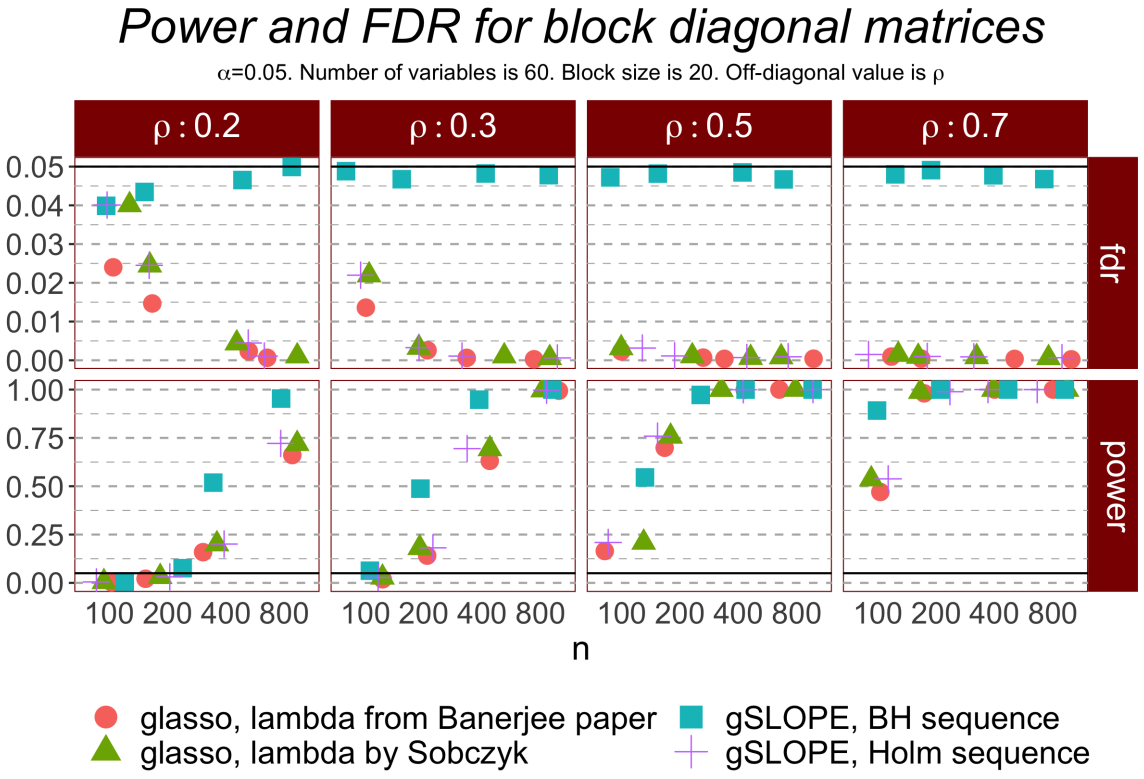


Figure 5.6: Graphical SLOPE. Block diagonal matrix. p is small (60), α is also small - 0.05

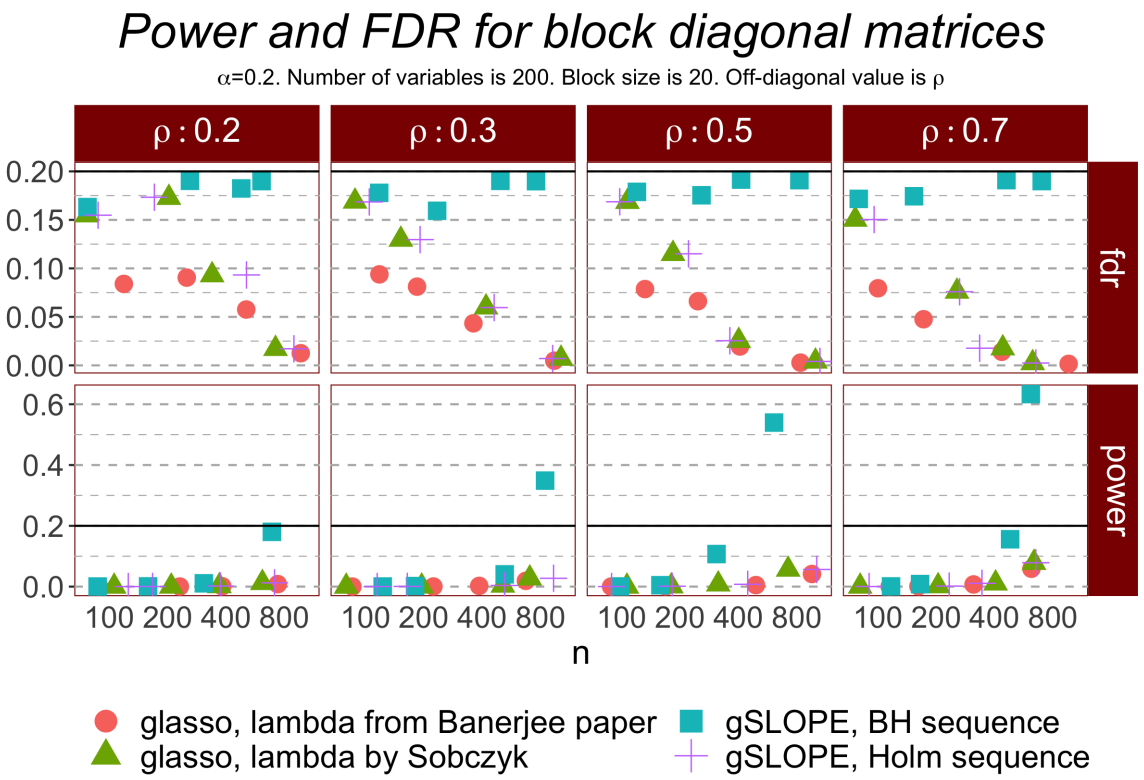


Figure 5.7: Graphical SLOPE. Block diagonal matrix. p is large (200), α is also high - 0.2

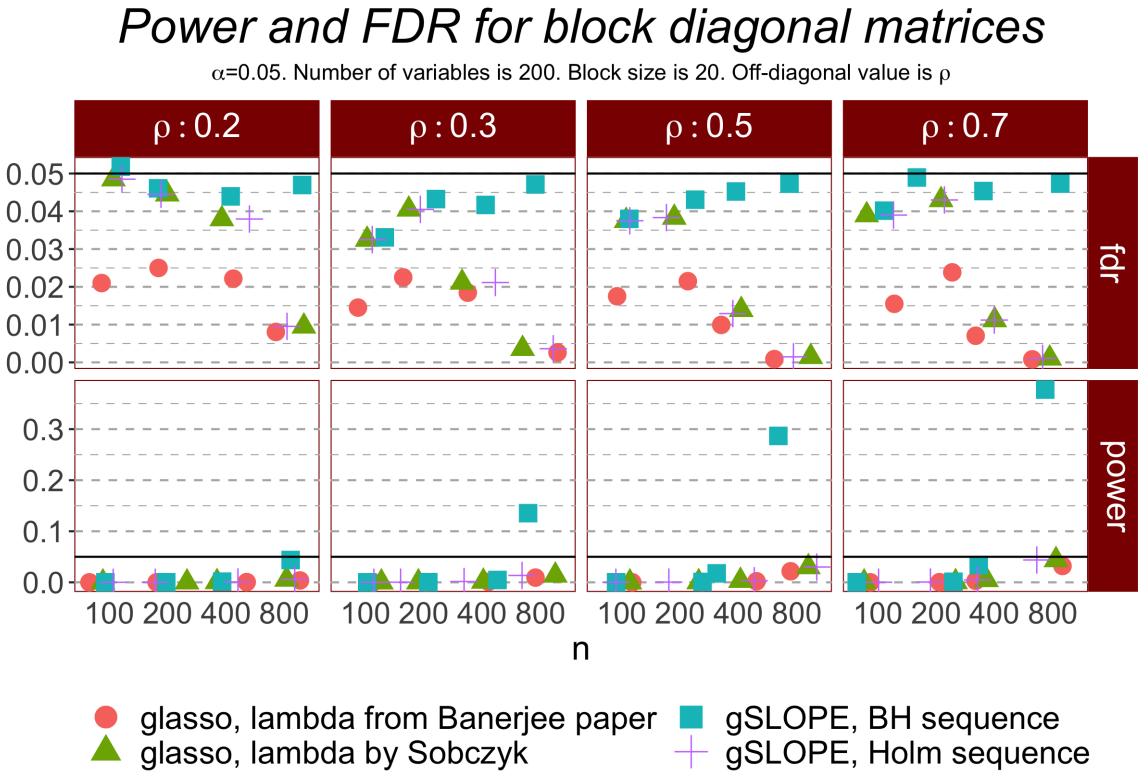


Figure 5.8: Graphical SLOPE. Block diagonal matrix. p is large (200), α is small - 0.05

2. number of observations in the data set, varying from 50 to 800
3. value of nonzero elements in precision matrix $\rho \in [0.1, 0.5]$
4. nominal FDR/FWER level in range $\alpha \in (0.01, 0.1)$

Full simulations results are available online. Here we show and comment on selected results. When the number of observations grows, then the signal is relatively stronger, and all methods have higher power. Again, as in block diagonal case, recall that λs from Banerjee et al. [2008a] are getting smaller when n grows. Therefore we expect that power of any statistical method to grow with n. This intuition is backed up by the simulation results, see e.g. (**??**). As expected by the choice of λ sequence for gSLOPE, it has higher power than glasso. This difference can be substantial, even 2 times higher.
In the case of hub structure neither FWER nor FDR are controlled. This is because this particular simulation scenario violates assumption for glasso consistency. Interestingly, when precision matrix entries ρ are smaller than 0.2, FDR seems to be controlled. This is again an insightful finding that proves usefulness of gSLOPE.
Naturally, as in block diagonal case, the bigger is the value on nonzero elements in precision matrix, the higher power is achieved by all of the methods.

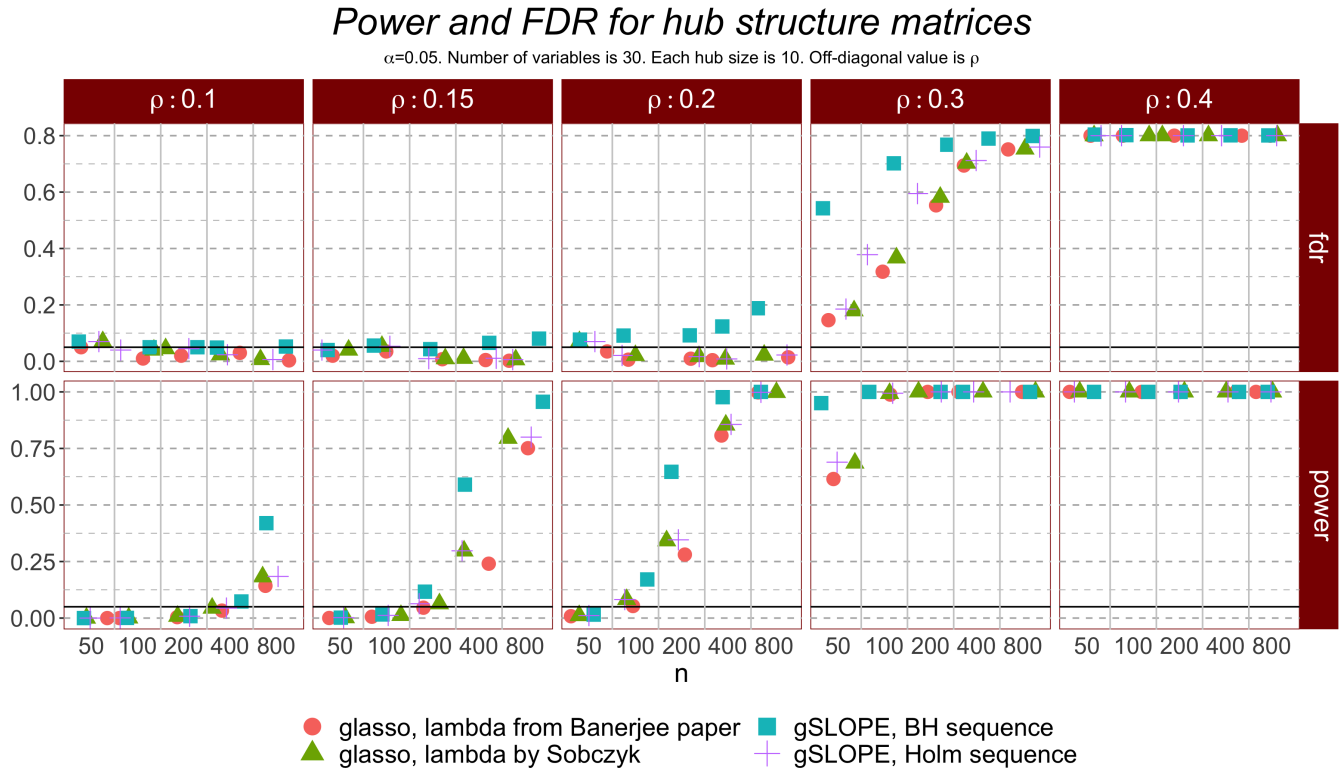


Figure 5.9: Graphical SLOPE. Hub matrix. p is small (30), α is average - 0.05

5.4.6 Banded matrix

In this simulations, data is drawn 100 times according to the algorithm (14).
We compared performance for a grid of parameters

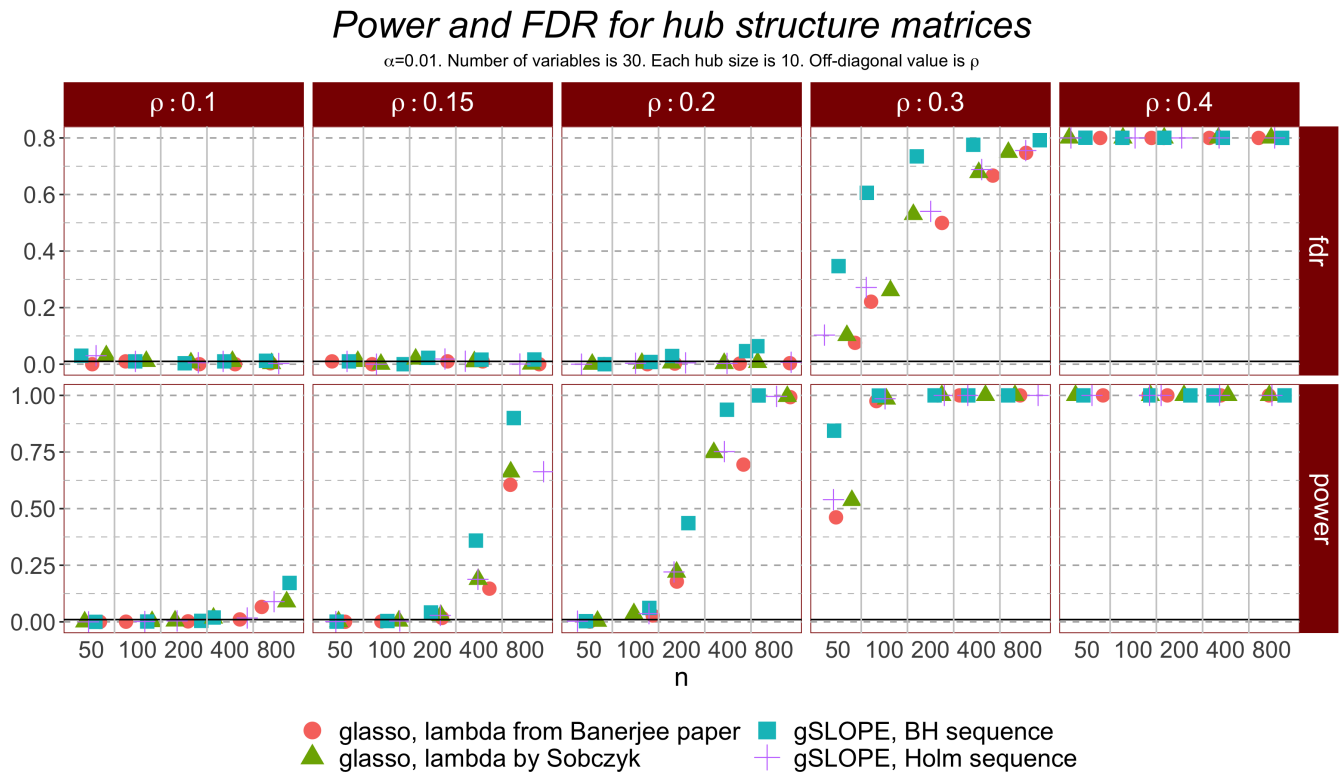


Figure 5.10: Graphical SLOPE. Hub matrix. p is small (30), α is small - 0.01

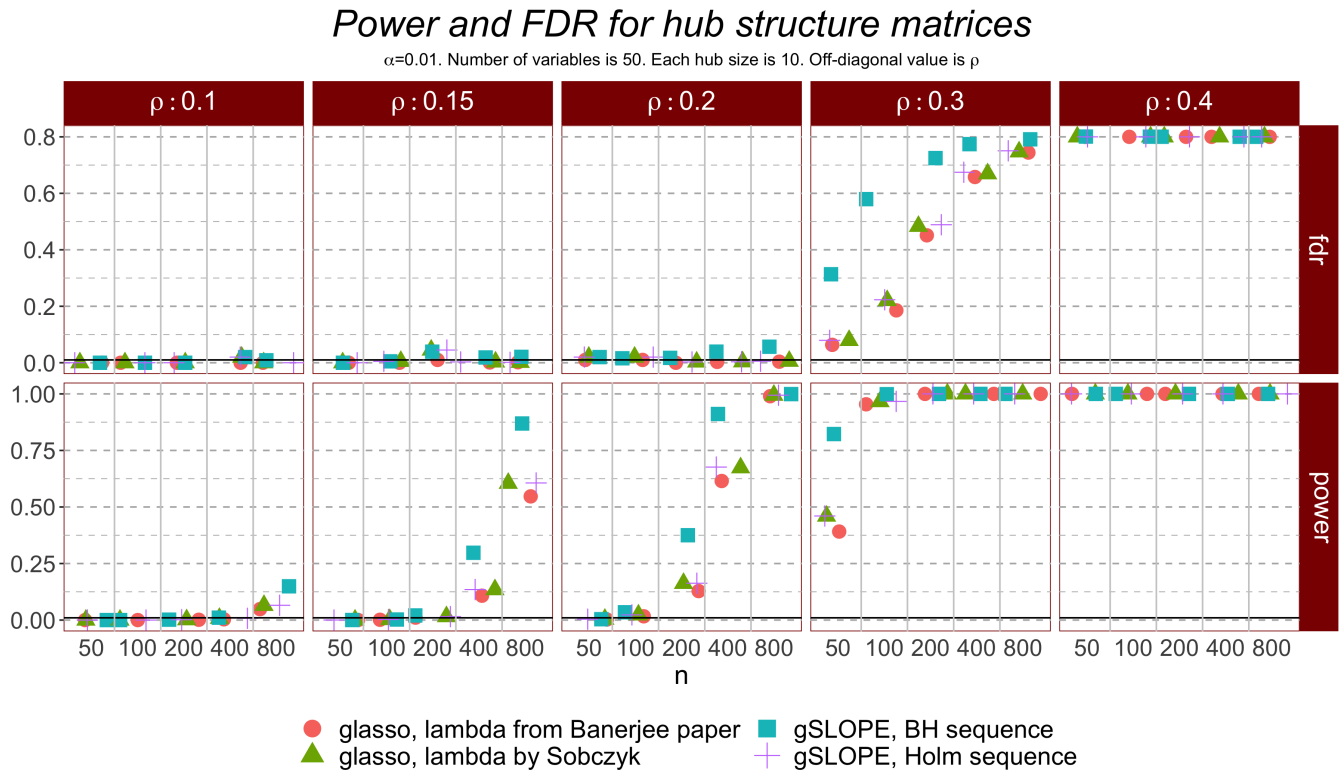


Figure 5.11: Graphical SLOPE. Hub matrix. p is average (50), α is small - 0.01

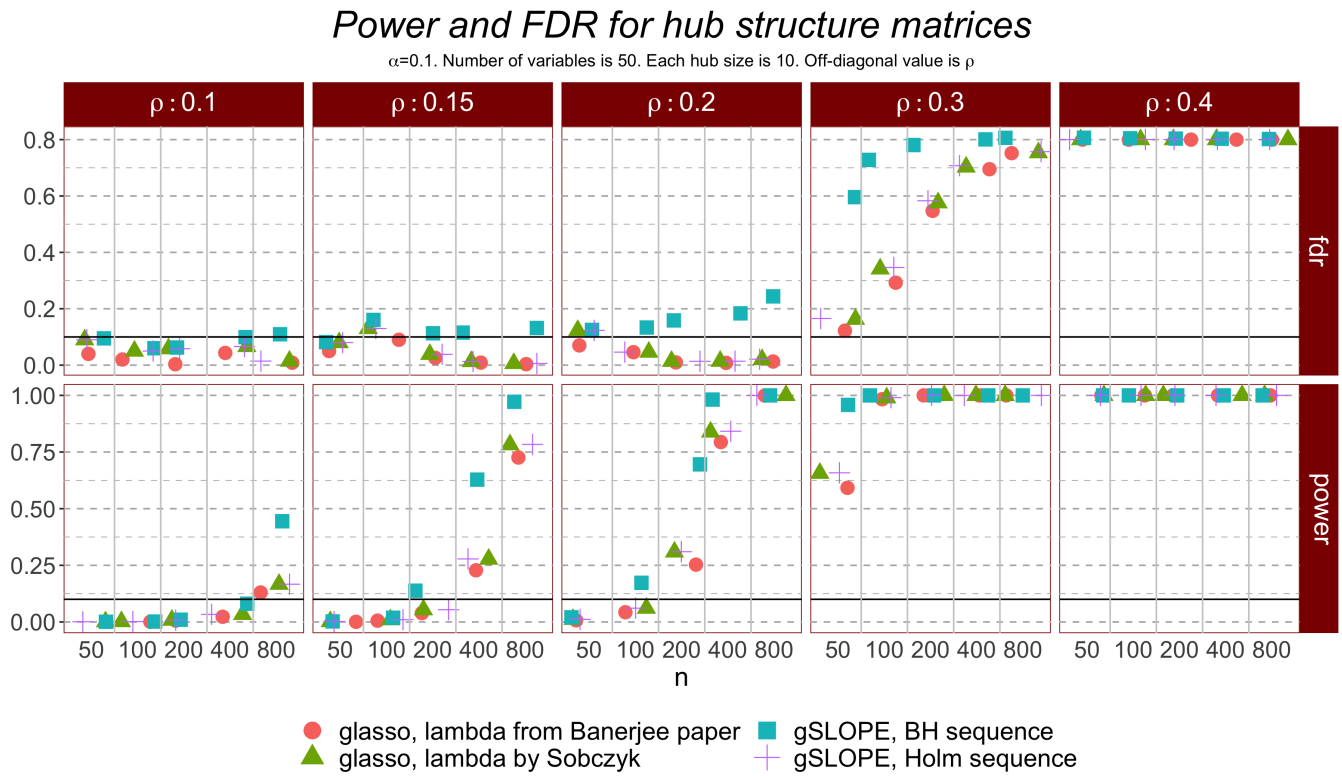


Figure 5.12: Graphical SLOPE. Hub matrix. p is average (50), α is large - 0.1

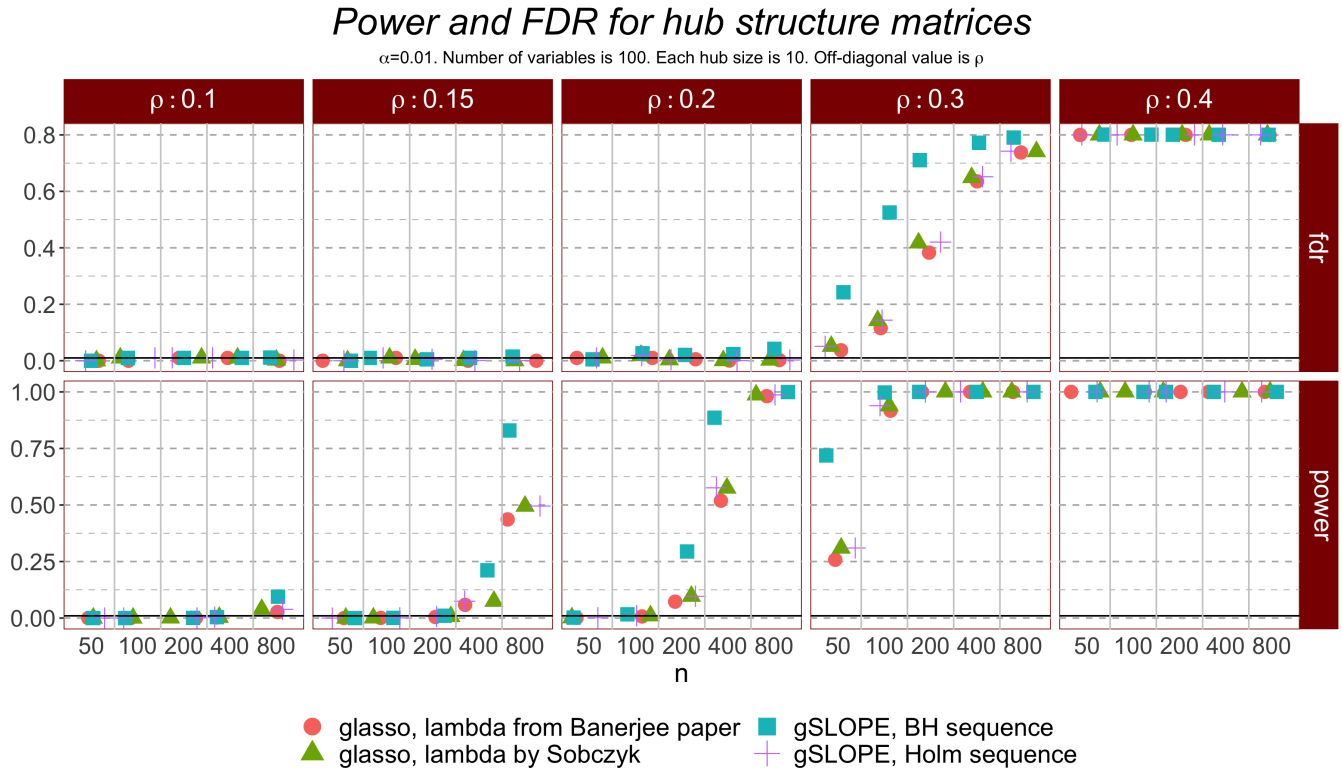


Figure 5.13: Graphical SLOPE. Hub matrix. p is average (100), α is small - 0.01

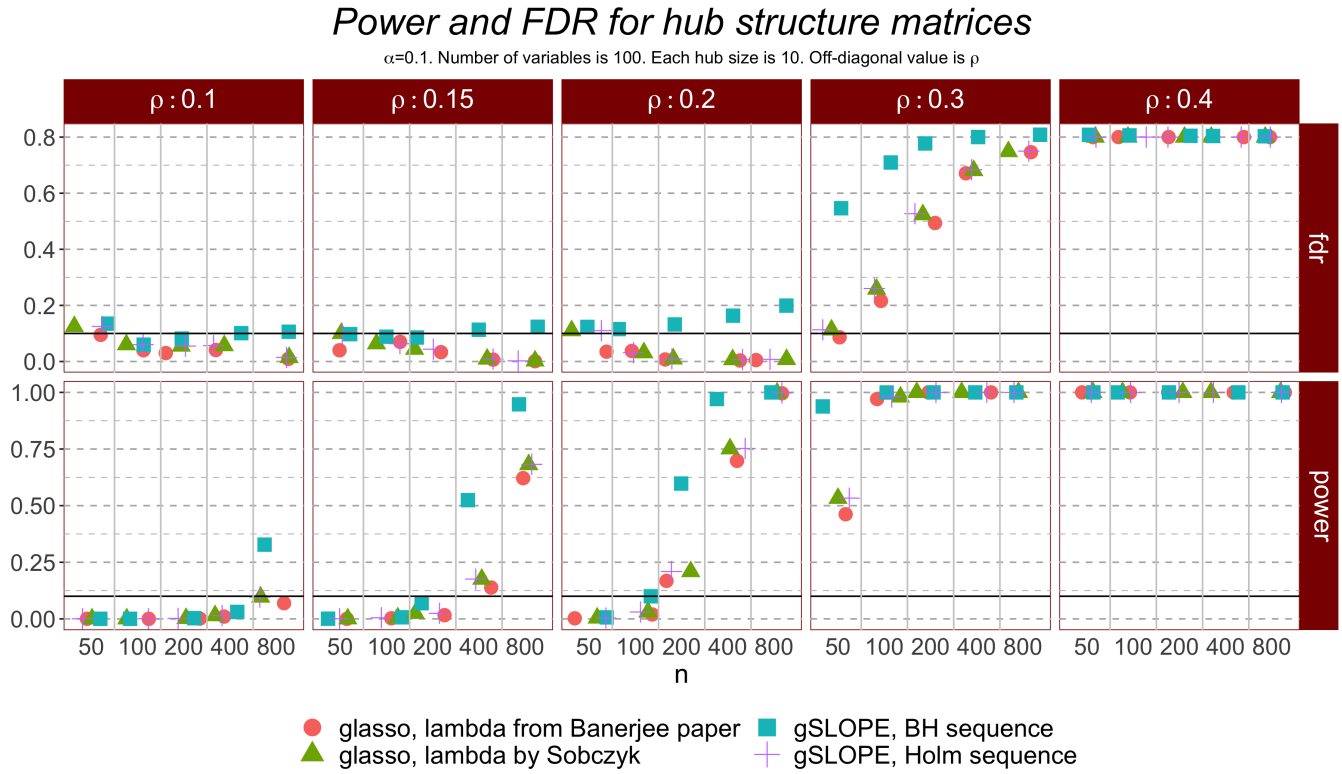


Figure 5.14: Graphical SLOPE. Hub matrix. p is average (100), α is large - 0.1

1. numbers of variables in the data set, varying from 30 to 200
2. number of observations in the data set, varying from 50 to 800
3. value of nonzero elements in precision matrix $\rho \in [0.1, 0.5]$
4. nominal FDR/FWER level in range $\alpha \in (0.01, 0.1)$

Full simulations results are available online. Here we show and comment on selected results. When the number of observations grows, then the signal is relatively stronger, and all methods have higher power. Again, as in block diagonal case, recall that λs from Banerjee et al. [2008a] are getting smaller when n grows. Therefore we expect that power of any statistical method to grow with n. This intuition is backed up by the simulation results, see e.g. (**??**). As expected by the choice of λ sequence for gSLOPE, it has higher power than glasso. This difference can be substantial, even 2 times higher.
Just like in the case of hub structure, neither FWER nor FDR are controlled. Interestingly, also just like for hub structure, when precision matrix entries ρ are smaller than 0.2, FDR seems to be controlled. This is again an insightful finding that proves usefulness of gSLOPE.
Naturally, as in block diagonal case, the bigger is the value on nonzero elements in precision matrix, the higher power is achieved by all of the methods.

5.4.7 FWER block diagonal matrix

In this simulation, data is drawn 100 times according to the Algorithm 12.
We compared performance for a grid of parameters

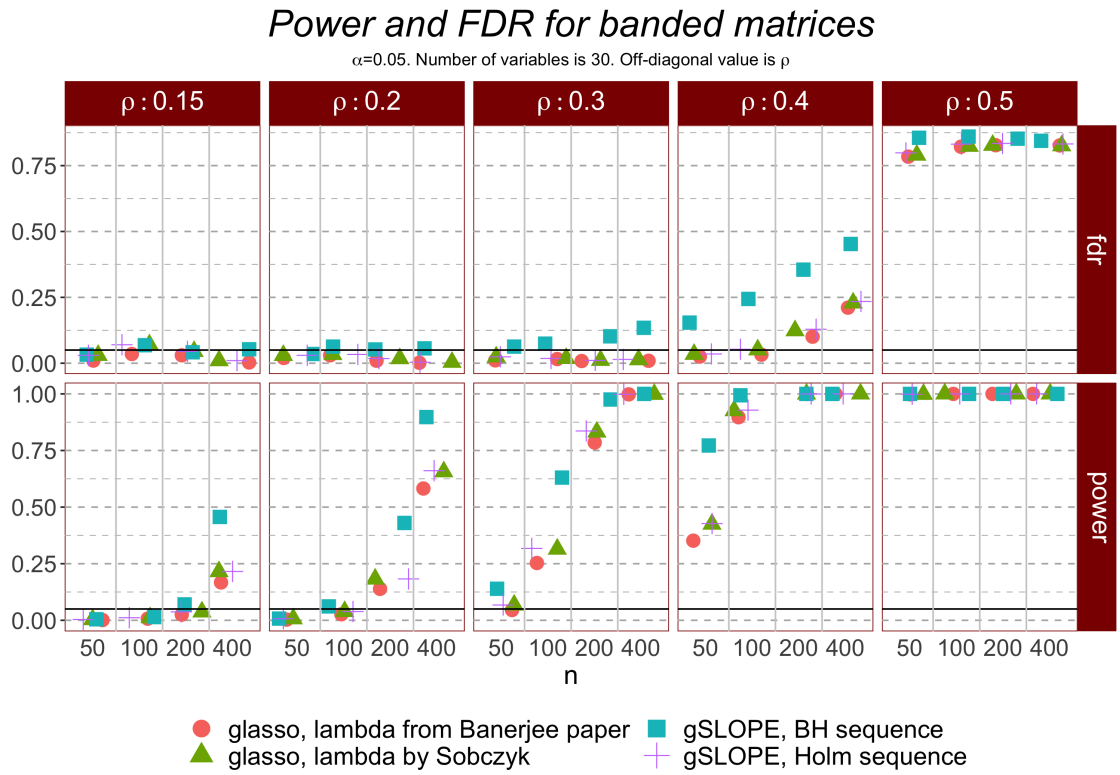


Figure 5.15: Graphical SLOPE. Banded matrix. p is small (30), α is average - 0.05

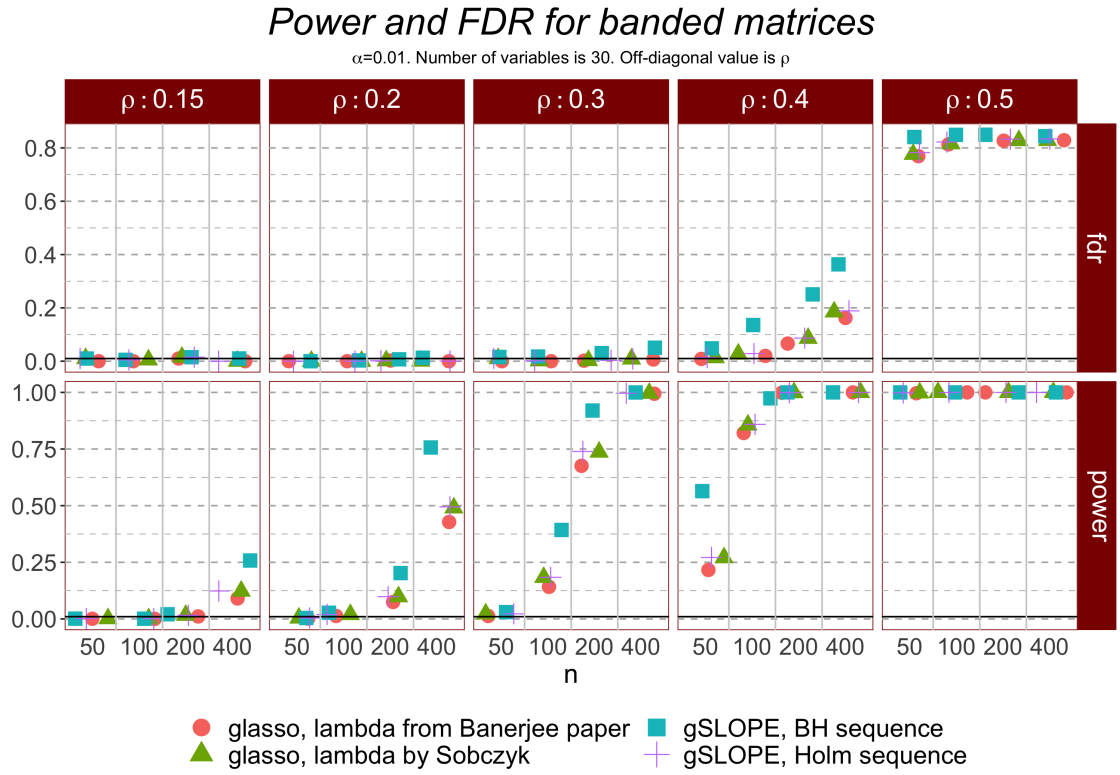


Figure 5.16: Graphical SLOPE. Banded matrix. p is small (30), α is small - 0.01

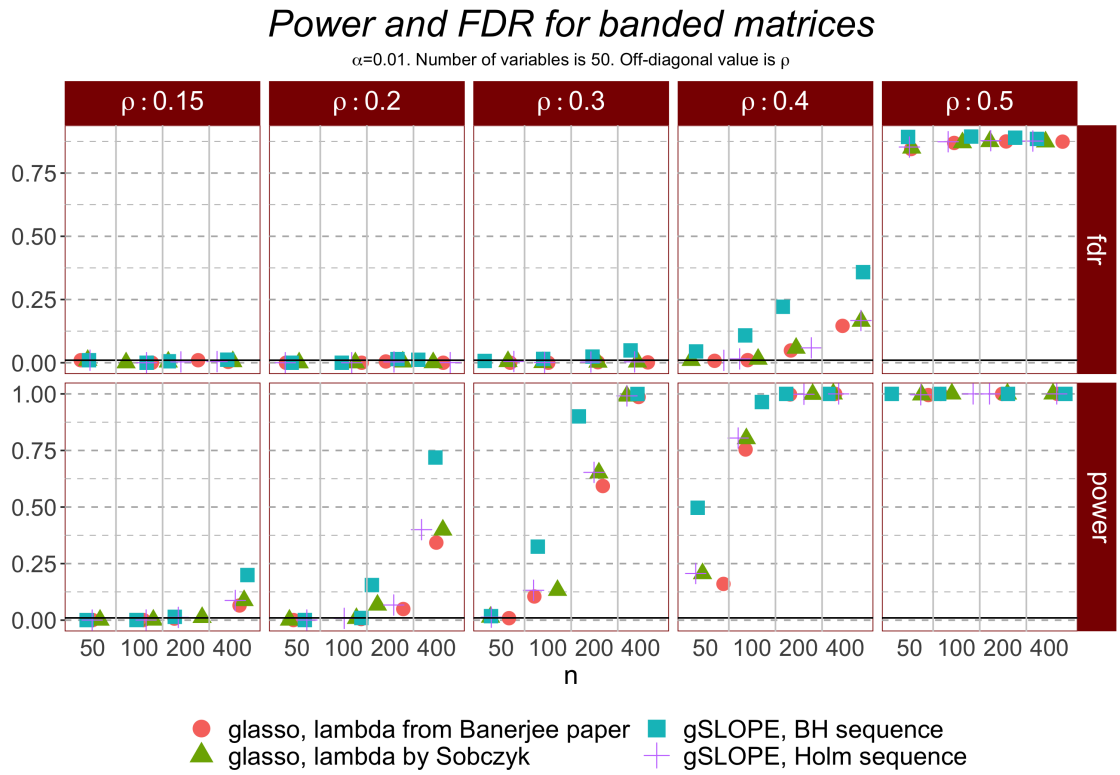


Figure 5.17: Graphical SLOPE. Banded matrix. p is average (50), α is small - 0.01

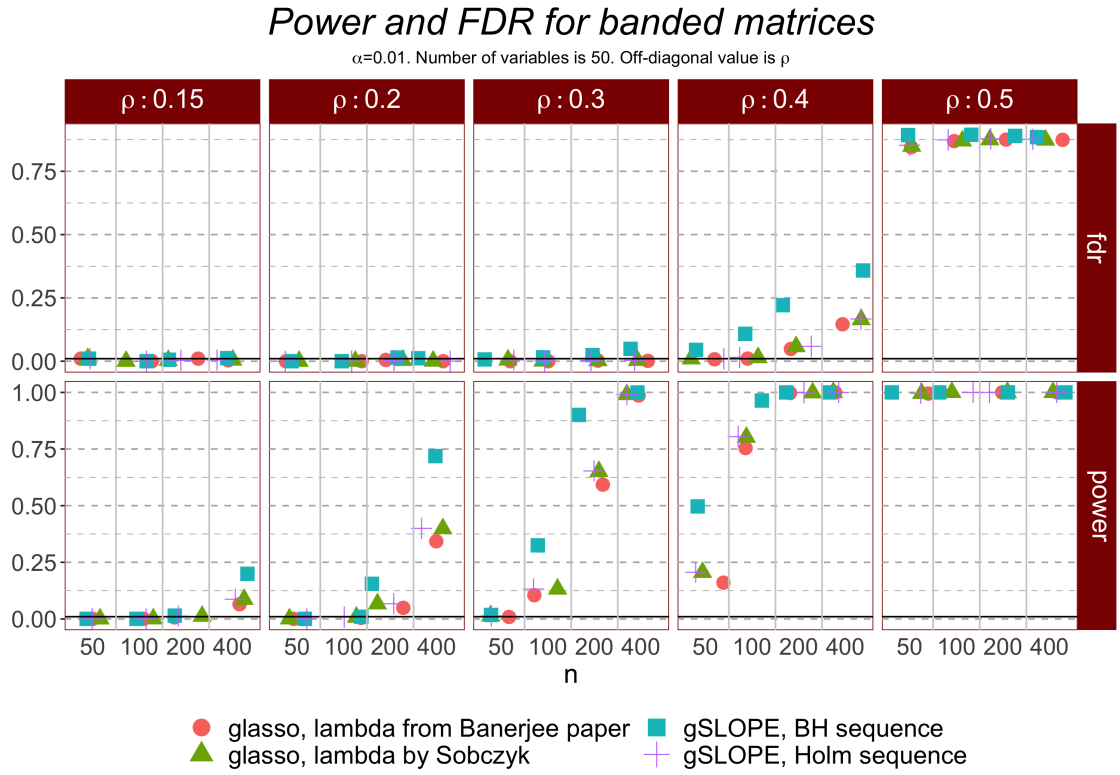


Figure 5.18: Graphical SLOPE. Banded matrix. p is average (50), α is large - 0.1

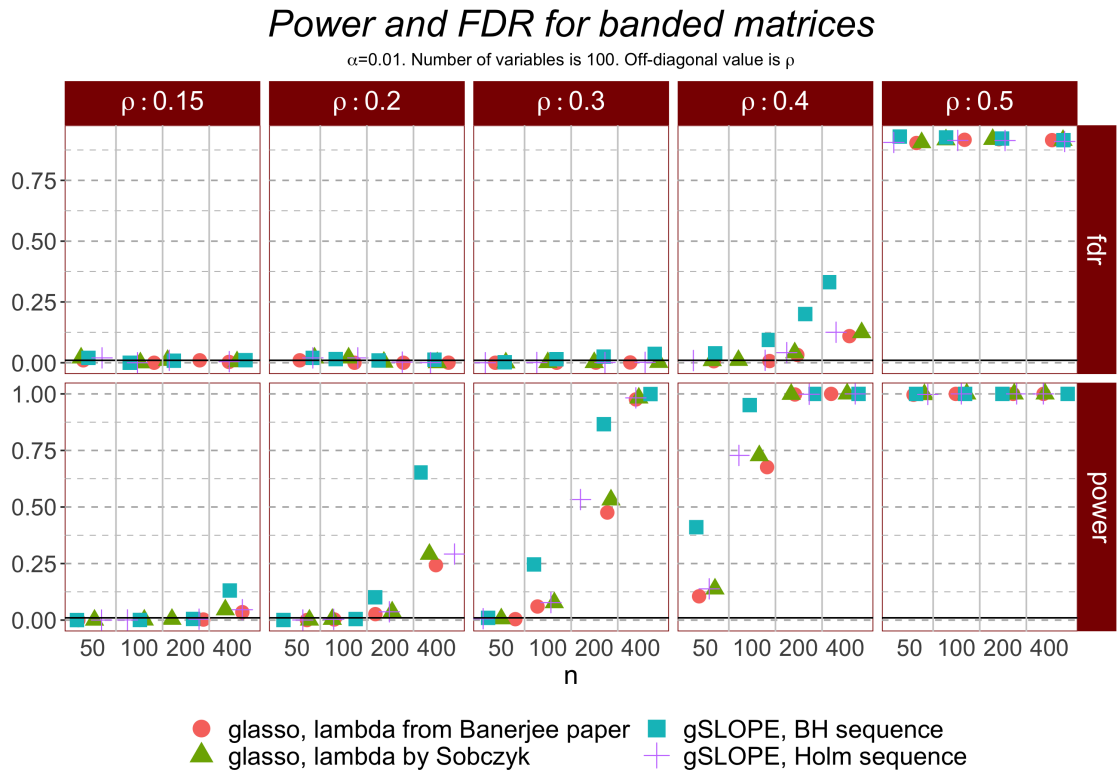


Figure 5.19: Graphical SLOPE. Banded matrix. p is average (100), α is small - 0.01

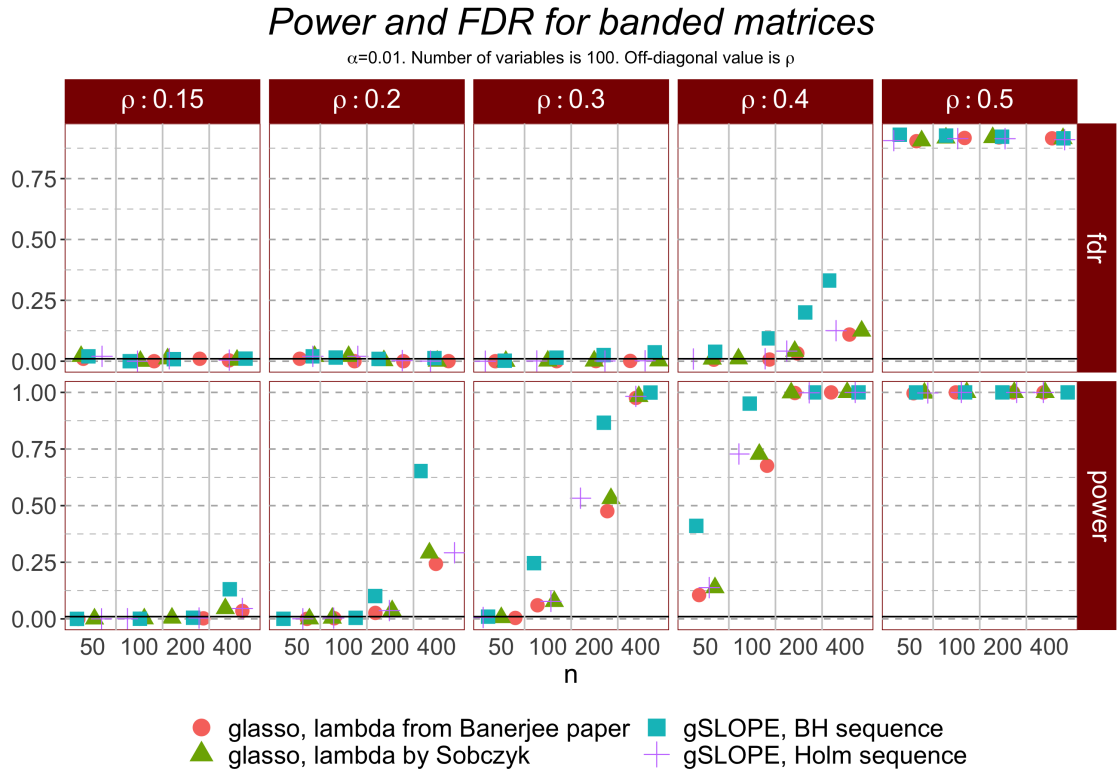


Figure 5.20: Graphical SLOPE. Banded matrix. p is average (100), α is large - 0.1

1. numbers of variables in the data set, varying from 60 to 200
2. number of observations in the data set, varying from 100 to 800
3. value of nonzero elements in precision matrix $\rho \in [0.2, 0.7]$
4. nominal FDR/FWER level in range $\alpha \in (0.05, 0.2)$

Here we show and comment on selected results. We focus only on FWER as we discussed power in the previous sections. Both glasso and gSLOPE with Holm's correction control FWER. This is expected because simulation settings are in accordance with assumptions of Theorem 5.2.5. Glasso with improved λ and gSLOPE with Holm's sequence have FWER very close to level α. When correlation and α are small all methods become very conservative and even BH gSLOPE is close to controlling FWER.

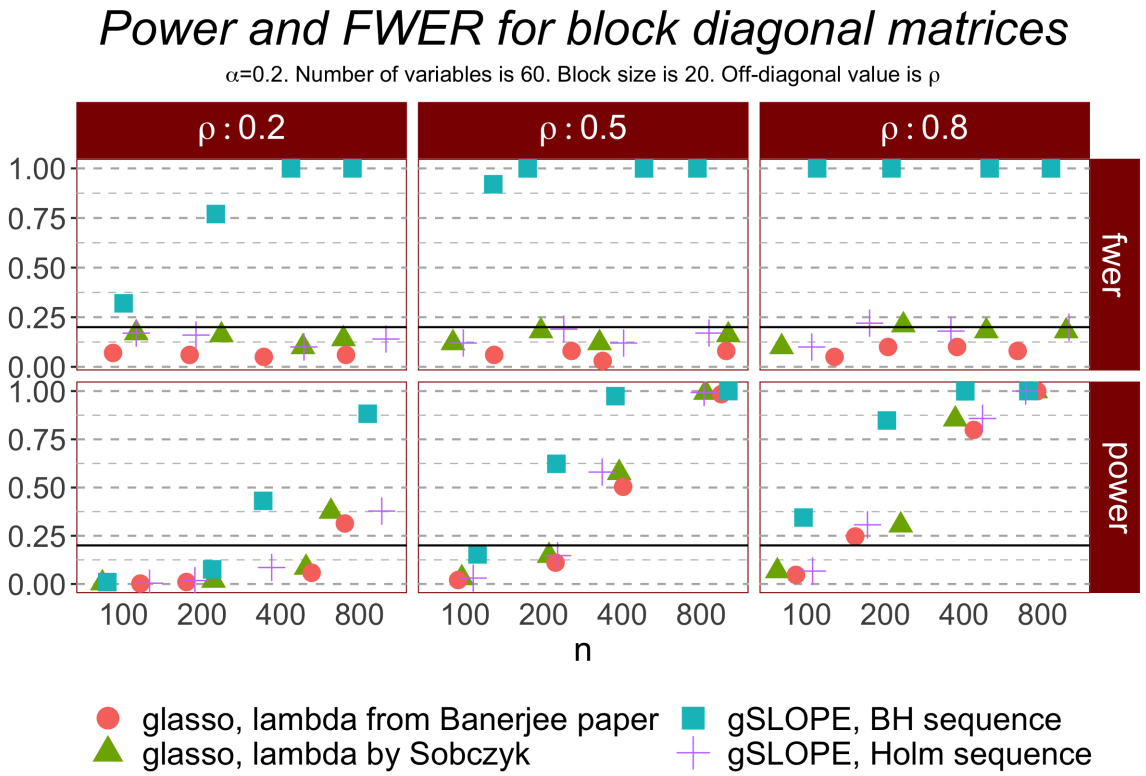


Figure 5.21: Graphical SLOPE. FWER control. p is small (60), α is big - 0.2

5.4.8 ROC curve

In this simulation, data was drawn according to the Algorithm 8. We used it previously in chapter on subspace clustering. It is of interest, as it tries to mimic genetic data, in which groups of variables act together in genetic pathways. We are interested in whether gSLOPE is better estimator than glasso. This time comparison is based on ROC curve. On x axis we have false positive rate, on y axis – true positive rate. The higher the curve is, the better is the estimator as it provides more true discoveries at the same number of false discoveries.
We compared performance for a grid of parameters

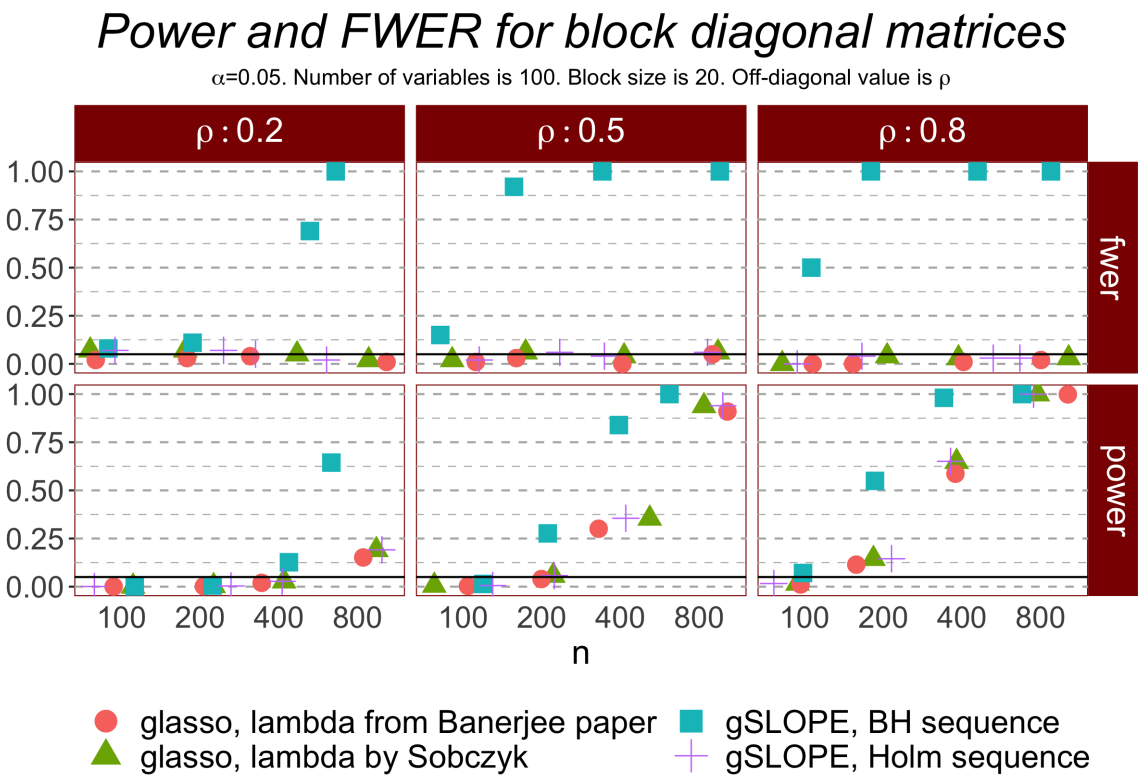


Figure 5.22: Graphical SLOPE. FWER control. p is average (100), α is small - 0.05

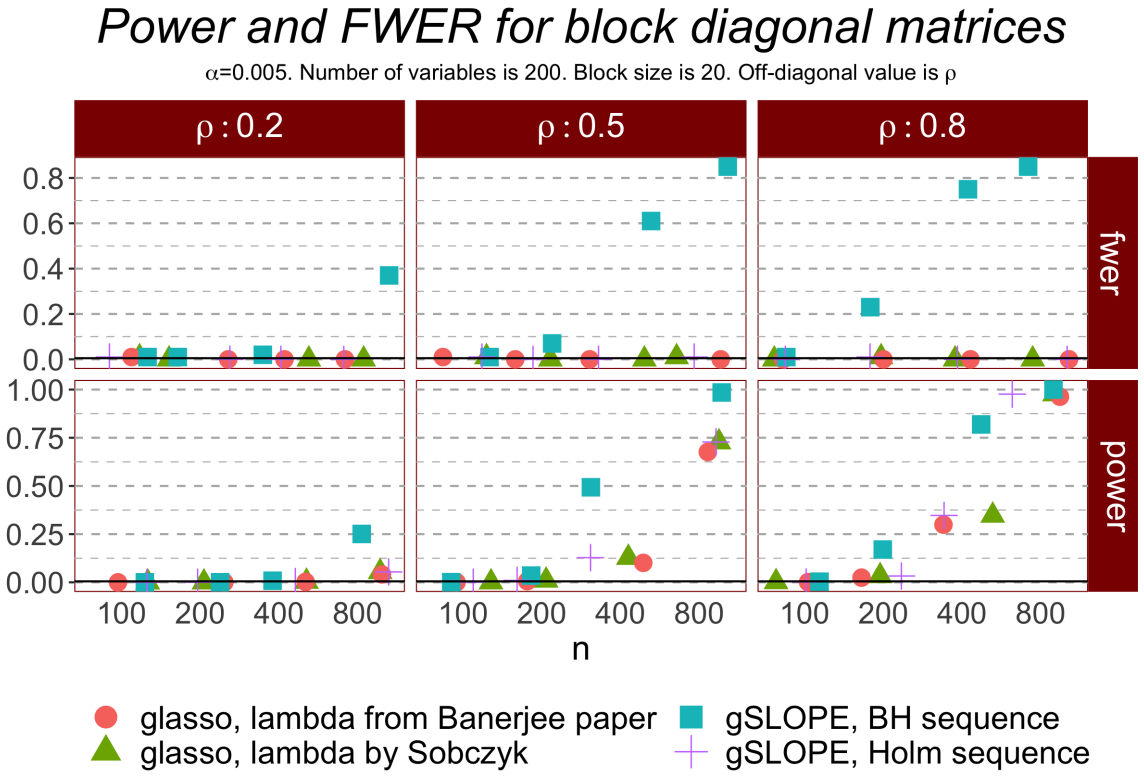


Figure 5.23: Graphical SLOPE. FWER control. p is large (200), α is small - 0.005

1. numbers of variables in the data set, varying from 50 to 100
2. number of observations in the data set, varying from 200 to 800
3. value of signal to noise ratio (defined in section 3.3.2. $SNR \in (1, 2, 3)$
4. nominal FDR/FWER level varies. Each point on the ROC curve corresponds to some α

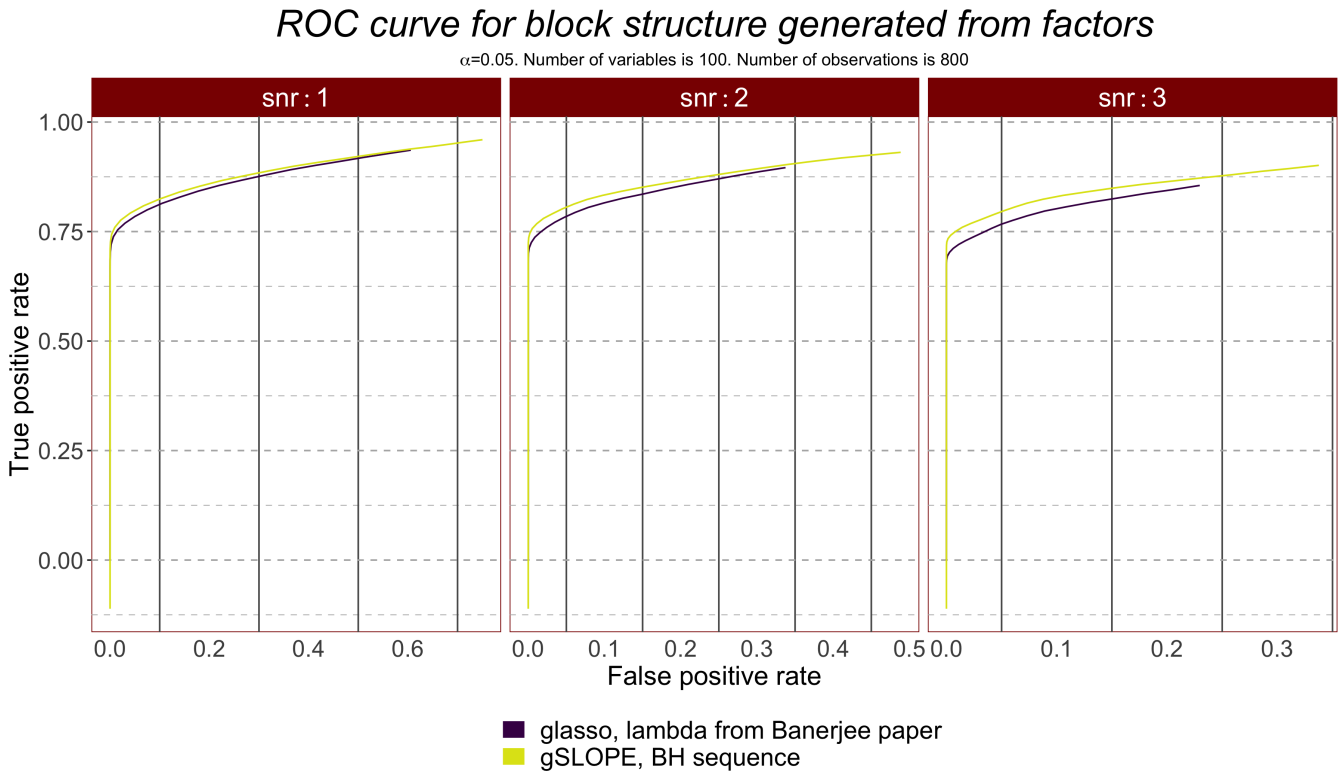


Figure 5.24: Graphical SLOPE. ROC curve. p is average (100), n is 800

It can be seen that ROC curve for graphical SLOPE lies consistently above the curve for glasso with λ from Banerjee et al. [2008a] paper. The stronger the signal gets, the bigger is the difference. This proves that it makes sense to use graphical SLOPEin practice, as area under ROC curve (AUC) is one of the most important metric taken into account in choosing model for the data.

5.4.9 Summary of simulation results

Graphical SLOPE proved to be a competitive method compared to the popular glasso. λ sequence inspired by Holm's correction for multiple testing controls FWER under various scenarios and yields higher power than glasso with λ from Banerjee et al. [2008a]. For specific matrix structures, when correlations are small, λ inspired by Benjamini-Hochberg correction for multiple testing, leads to gSLOPE that controls FDR. At the same time it yields much higher power than glasso. When correlations are high, FDR is not controlled. Preliminary simulation results suggest that gSLOPE might control local FDR (average FDR per variable). This is a topic of an ongoing research.

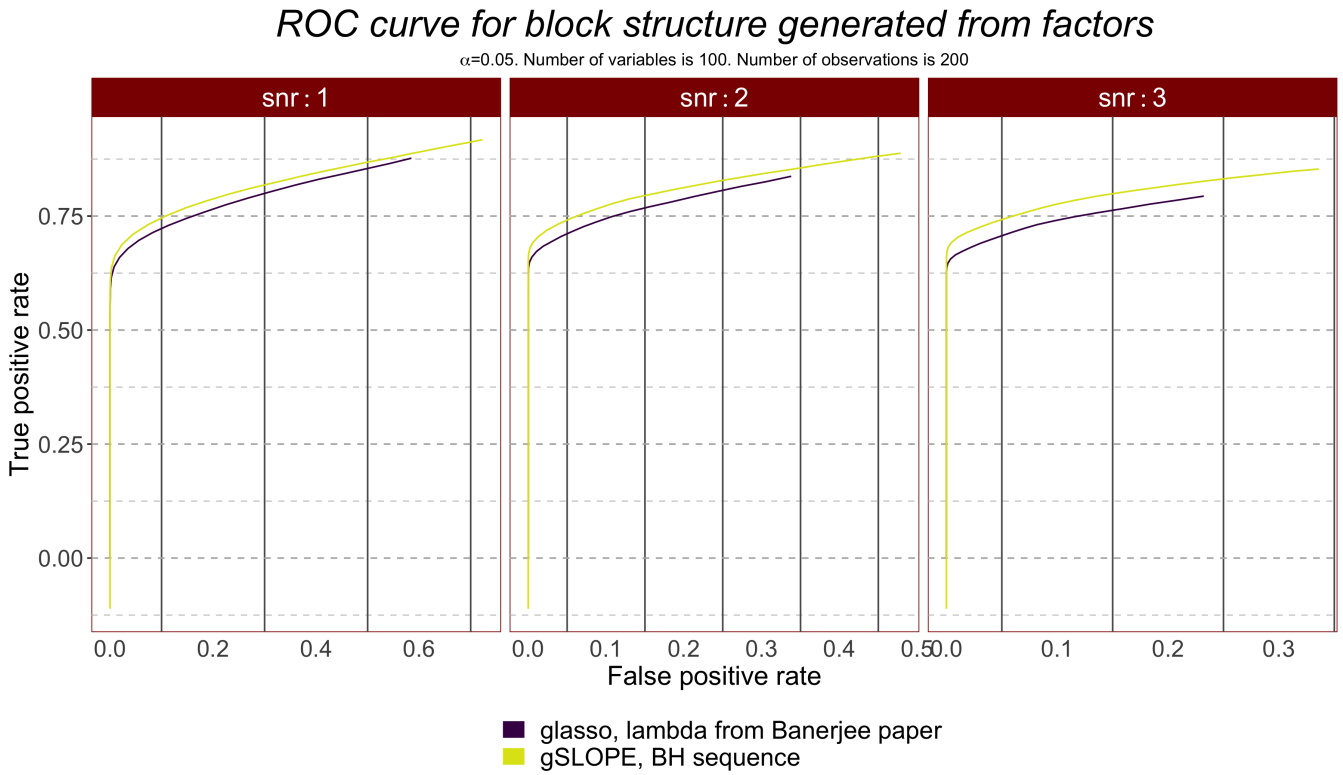


Figure 5.25: Graphical SLOPE. ROC curve. p is average (100), n is 200

Bibliography

Agarwal, Pankaj K. and Mustafa, Nabil H. K-means projective clustering. In *Proceedings of the Twenty-third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '04, pages 155–165, New York, NY, USA, 2004. ACM. ISBN 158113858X.

Akaike, Hirotugu. A new look at the statistical model identification. 19:716 – 723, 01 1975.

Allen, Genevera I.; Grosenick, Logan, and Taylor, Jonathan. A generalized least-square matrix decomposition. *Journal of the American Statistical Association*, 109(505):145–159, 2014.

Bai, Jushan and Ng, Serena. Determining the number of factors in approximate factor models. *Econometrica*, 70(1):191–221, 2002.

Bai, Zhidong and Silverstein, Jack W. *Spectral Analysis of Large Dimensional Random Matrices 2010.*

Baiju, NT. Exciting facts and findings about big data you should know. `http://bigdata-madesimple.com/exciting-facts-and-findings-about-big-data/`. Accessed: 2018-12-29.

Banerjee, Onureena; El Ghaoui, Laurent, and d'Aspremont, Alexandre. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine Learning Research*, 2008a.

Banerjee, Onureena; Ghaoui, Laurent El, and d'Aspremont, Alexandre. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine Learning Research*, 9:485–516, 2008b.

Beck, Amir and Teboulle, Marc. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

Benjamini, Yoav and Hochberg, Yosef. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B*, 57(1):289–300, 1995.

Benjamini, Yoav and Yekutieli, Daniel. The control of the false discovery rate in multiple testing under dependency. *Ann. Statist.*, 29(4):1165–1188, 08 2001. doi: 10.1214/aos/1013699998. URL `http://dx.doi.org/10.1214/aos/1013699998`.

Bishop, Christopher M. Bayesian pca. In *Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems II*, pages 382–388, Cambridge, MA, USA, 1999a. MIT Press. ISBN 0-262-11245-0.

Bishop, Christopher M. Variational principal components. In *Proceedings Ninth International Conference on Artificial Neural Networks, ICANN'99*, volume 1, page 509–514. IEE, January 1999b.

Bogdan, Malgorzata; van den Berg, Ewout; Sabatti, Chiara; Su, Weijie, and Candes, Emmanuel. Slope—adaptive variable selection via convex optimization. *The Annals of Applied Statistics*, 2015a.

Bogdan, Malgorzata; van den Berg, Ewout; Sabatti, Chiara; Su, Weijie, and Candes, Emmanuel J. SLOPE – adaptive variable selection via convex optimization. *Annals of Applied Statistics*, 9(3):1103–1140, 2015b.

Bogdan, Małgorzata; Ghosh, Jayanta K., and R.W., Doerge. Modifying the schwarz bayesian information criterion to locate multipleinteracting quantitative trait loci. *Genetics*, 167:989–999, 2004.

Bogdan, Małgorzata; K. Ghosh, Jayanta, and Zak-Szatkowska, Malgorzata. Selecting explanatory variables with the modified version of the bayesian information criterion. *Quality and Reliability Eng. Int.*, 24:627–641, 10 2008. doi: 10.1002/qre.936.

Bondell, Howard D. and Reich, Brian J. Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with OSCAR. *Biometrics*, 64:115–123, 2008.

Bonferroni, C.E. *Teoria statistica delle classi e calcolo delle probabilità*. Pubblicazioni del R. Istituto superiore di scienze economiche e commerciali di Firenze. Libreria internazionale Seeber, 1936. URL `https://books.google.pl/books?id=3CY-HQAACAAJ`.

Boyd, Stephen and Vandenberghe, Lieven. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521833787.

Boyd, Stephen; Parikh, Neal; Chu, Eric; Peleato, Borja, and Eckstein, Jonathan. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3 (1):1–122, January 2011. ISSN 1935-8237. doi: 10.1561/2200000016. URL `http://dx.doi.org/10.1561/2200000016`.

Brzyski, Damian; Su, Weijie, and Bogdan, Malgorzata. Group SLOPE - adaptive selection of groups of predictors. *arXiv:1511.09078*, 2015.

Caussinus, H. Models and uses of principal component analysis. In de Leeuw, J.; Heiser, W.; Meulman, J., and Critchley, F., editors, *Multidimensional data analysis*, pages 149–178. DSWO Press, 1986.

Chavent, Marie; Kuentz-Simonet, Vanessa; Liquet, Benoît, and Saracco, Jérôme. Clustofvar: An r package for the clustering of variables. *Journal of Statistical Software*, 50(13):1–16, 9 2012. ISSN 1548-7660. URL `http://www.jstatsoft.org/v50/i13`.

Chikuse, Y. *Statistics on special manifolds*. Lecture notes in statistics. Springer, 2003. ISBN 9783540001607.

Choi, Y.; Taylor, J., and Tibshirani, R. Selecting the number of principal components: estimation of the true rank of a noisy matrix. *ArXiv e-prints*, October 2014.

D'agostino, Ralph B. and Russell, Heidy K. Scree test. In *Encyclopedia of Biostatistics*. John Wiley & Sons, Ltd, 2005. ISBN 9780470011812.

d'Aspremont, Alexandre; Banerjee, Onureena, and El Ghaoui, Laurent. First-order methods for sparse covariance selection. *SIAM Journal on Matrix Analysis and Applications*, 30(1):56–66, 2008.

de Bruijn, N.G. *Asymptotic Methods in Analysis*. Bibliotheca mathematica. Dover Publications, 1970. ISBN 9780486642215.

Elhamifar, Ehsan and Vidal, René. Sparse subspace clustering. In *In CVPR*, 2009.

Figueiredo, Mário A. T. and Nowak, Robert D. Ordered weighted L1 regularized regression with strongly correlated covariates: Theoretical aspects. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, Cadiz, Spain, May 9-11, 2016*, pages 930–938, 2016.

Friedman, Jerome; Hastie, Trevor, and Tibshirani, Robert. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008a.

Friedman, Jerome; Hastie, Trevor, and Tibshirani, Robert. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 2008b.

George, Edward I. and McCulloch, Robert E. Variable selection via gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993. ISSN 01621459. URL `http://www.jstor.org/stable/2290777`.

Ghosh, J.K.; Delampady, M., and Samanta, T. *An Introduction to Bayesian Analysis: Theory and Methods*. Springer Texts in Statistics. Springer, 2007. ISBN 9780387354330.

Hastie, Trevor and Mazumder, Rahul. *softImpute: Matrix Completion via Iterative Soft-Thresholded SVD*, 2015. URL `https://CRAN.R-project.org/package=softImpute`. R package version 1.4.

Hastie, Trevor; Tibshirani, Robert, and Friedman, Jerome. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition, 2009. URL `http://www-stat.stanford.edu/~tibs/ElemStatLearn/`.

Hochberg, Yosef. A sharper bonferroni procedure for multiple tests of significance. *Biometrika*, 75(4):800–802, 1988. doi: 10.1093/biomet/75.4.800. URL `http://dx.doi.org/10.1093/biomet/75.4.800`.

Hoerl, A. E. and Kennard, R. W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

Hoff, Peter D. Model averaging and dimension selection for the singular value decomposition. *Journal of the American Statistical Association*, 2007.

Holm, Sture. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2): 65–70, 1979. ISSN 03036898, 14679469. URL `http://www.jstor.org/stable/4615733`.

Hoyle, David C. Automatic pca dimension selection for high dimensional data and small sample sizes. *Journal of Machine Learning Research*, 9(12):2733–2759, 2008.

Hsieh, Cho-Jui; Dhillon, Inderjit S.; Ravikumar, Pradeep K., and Sustik, Mátyás A. Sparse inverse covariance matrix estimation using quadratic approximation. In *Advances in Neural Information Processing Systems 24*, pages 2330–2338. MIT Press, 2011.

Hsieh, Cho-Jui; Banerjee, Arindam; Dhillon, Inderjit S., and Ravikumar, Pradeep K. A divide-and-conquer method for sparse inverse covariance estimation. In *Advances in Neural Information Processing Systems 25*, pages 2330–2338. 2012.

Hsieh, Cho-Jui; Sustik, Matyas A; Dhillon, Inderjit; Ravikumar, Pradeep, and Poldrack, Russell. BIG & QUIC: Sparse inverse covariance estimation for a million variables. In *Advances in Neural Information Processing Systems 26*, pages 3165–3173. MIT Press, 2013.

Hubert, L. and Arabie, P. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

Husson, Francois; Josse, Julie; Le, Sebastien, and Mazet, Jeremy. *FactoMineR: Multivariate Exploratory Data Analysis and Data Mining with R*, 2014. R package version 1.27.

Husson, François; Lê, Sebastien, and Pagès, Jérôme. *Exploratory Multivariate Analysis by Example Using R*. Chapman & Hall, 2010. ISBN 978-1-4398-3580-7.

Ilin, Alexander; Raiko, Tapani, and Jaakkola, Tommi. Practical approaches to principal component analysis in the presence of missing values. *JMLR*, pages 1957–2000, 2010.

Jackson, Donald A. Stopping rules in principal components analysis: a comparison of heuristical and statistical approaches. *Ecology*, pages 2204–2214, 1993.

James, A. T. Normal multivariate analysis and the orthogonal group. *Ann. Math. Statist.*, 25(1):40–75, 03 1954.

Jolliffe, I.T. *Principal Component Analysis*. Springer Series in Statistics. Springer, 2002. ISBN 9780387954424.

Josse, Julie and Husson, François. Selecting the number of components in principal component analysis using cross-validation approximations. *Computational Statistics & Data Analysis*, 56(6):1869 – 1879, 2012.

Josse, Julie; Husson, François, and Pagès, Jérôme. Gestion des données manquantes en analyse en composantes principales. *Journal de la Société Française de Statistique*, 150(2):28–51, 2009.

Josse, Julie; Pagès, Jérôme, and Husson, François. Multiple imputation in principal component analysis. *Advances in Data Analysis and Classification*, 5(3):231–246, 2011. ISSN 1862-5355.

Kendall, M. G.; Stuart, A., and Ord, J. K., editors. *Kendall's Advanced Theory of Statistics*. Oxford University Press, Inc., New York, NY, USA, 1987. ISBN 0-195-20561-8.

Lam, Clifford and Fan, Jianqing. Sparsistency and rates of convergence in large covariance matrix estimation. *The Annals of Statistics*, 37(6B):4254–4278, 12 2009.

Lauritzen, S.L. *Graphical Models*. Oxford Statistical Science Series. Clarendon Press, 1996. ISBN 9780191591228. URL `https://books.google.de/books?id=mGQWkx4guhAC`.

Ledoit, Olivier and Wolf, Michael. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365 – 411, 2004. ISSN 0047-259X. doi: https://doi.org/10.1016/S0047-259X(03)00096-4. URL `http://www.sciencedirect.com/science/article/pii/S0047259X03000964`.

Liu, Guangcan; Lin, Zhouchen; Yan, Shuicheng; Sun, Ju; Yu, Yong, and Ma, Yi. Robust recovery of subspace structures by low-rank representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):171–184, Jan 2013. ISSN 0162-8828.

Luxburg, Ulrike. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, December 2007.

Mazumder, Rahul and Hastie, Trevor. Exact covariance thresholding into connected components for large-scale graphical lasso. *Journal of Machine Learning Research*, 13:781–794, 2012.

McWilliams, Brian and Montana, Giovanni. Subspace clustering of high-dimensional data: a predictive approach. *Data Mining and Knowledge Discovery*, 28(3):736–772, 2014.

Meinshausen, N. and Bühlmann, P. High-dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34:1436–1462, 2006.

Meinshausen, Nicolai and Bühlmann, Peter. Stability selection. *Journal of the Royal Statistical Society (Series B)*, 72(4):417–473, 2010.

Meinshausen, Nicolai and Bühlmann, Peter. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 2006.

Minka, Thomas P. Automatic choice of dimensionality for pca. *NIPS*, 13:514, 2000.

Minka, Thomas P. Old and new matrix algebra useful for statistics, December 2000.

Nakajima, Shinichi; Tomioka, Ryota; Sugiyama, Masashi, and Babacan, S. Derin. Condition for perfect dimensionality recovery by variational bayesian pca. *Journal of Machine Learning Research*, 16:3757–3811, 2015. URL `http://jmlr.org/papers/v16/nakajima15a.html`.

Neath, Andrew A. and Cavanaugh, Joseph E. The bayesian information criterion: background, derivation, and applications. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(2):199–203, 2012.

Ng, A. Y.; Jordan, M. I., and Weiss, Y. On spectral clustering: Analysis and an algorithm. In *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, pages 849–856. MIT Press, 2001.

Owen, Art B. and Perry, Patrick O. Bi-cross-validation of the svd and the nonnegative matrix factorization. *Ann. Appl. Stat.*, 3(2):564–594, 06 2009.

Oztoprak, Figen; Nocedal, Jorge; Rennie, Steven, and Olsen, Peder A. Newton-like methods for sparse inverse covariance estimation. In *Advances in Neural Information Processing Systems 25*, pages 764–772. MIT Press, 2012.

Passemier, Damien; Li, Zhaoyuan, and Yao, Jian-Feng. On estimation of the noise variance in high-dimensional probabilistic principal component analysis. *Journal of the Royal Statistical Society Series B*, 2015.

Pearson, K. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(11): 559–572, 1901.

Petrov, P.V.V. and Petrov, V.V. *Limit Theorems of Probability Theory: Sequences of Independent Random Variables*. Oxford science publications. Clarendon Press, 1995. ISBN 9780198534990. URL https://books.google.pl/books?id=4LkdSaI4xXMC.

Rajan, J.J. and Rayner, P.J.W. Model order selection for the singular value decomposition and the discrete karhunen-loeve transform using a bayesian approach. *Vision, Image and Signal Processing, IEE Proceedings* -, 144(2):116–123, Apr 1997. ISSN 1350-245X.

Raskutti, Garvesh; Yu, Bin; Wainwright, Martin J, and Ravikumar, Pradeep K. Model selection in gaussian graphical models: High-dimensional consistency of ℓ_1-regularized mle. In *Advances in Neural Information Processing Systems 21*, pages 1329–1336. 2009.

Ravikumar, Pradeep; Raskutti, Garvesh; Wainwright, Martin J., and Yu, Bin. Model selection in gaussian graphical models: High-dimensional consistency of l1-regularized mle, 2008.

Rolfs, Benjamin; Rajaratnam, Bala; Guillot, Dominique; Wong, Ian, and Maleki, Arian. Iterative thresholding algorithm for sparse inverse covariance estimation. In *Advances in Neural Information Processing Systems 25*, pages 1574–1582. 2012.

Rothman, Adam J.; Bickel, Peter J.; Levina, Elizaveta, and Zhu, Ji. Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494–515, 2008.

Sarkar, Sanat K. Some probability inequalities for ordered mtp_2 random variables: a proof of the simes conjecture. *Ann. Statist.*, 26(2):494–504, 04 1998. doi: 10.1214/aos/1028144846. URL https://doi.org/10.1214/aos/1028144846.

Schwarz, Gideon. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 03 1978.

Sobczyk, P.; Bogdan, M., and Josse, J. *varclust: Variables Clustering*, 2016. R package version 0.9.30.

Sobczyk, Piotr; Bogdan, Małgorzata, and Josse, Julie. varclust – new iterative subspace clustering method. Konferencja Statystyka Matematyczna, 2014.

Sobczyk, Piotr; Bogdan, Małgorzata, and Josse, Julie. Bayesian dimensionality reduction with pca using penalized semi-integrated likelihood. *Journal of Computational and Graphical Statistics*, 26(4):826–839, 2017a. doi: 10.1080/10618600.2017.1340302. URL https://doi.org/10.1080/10618600.2017.1340302.

Sobczyk, Piotr; Wilczynski, Stanislaw; Josse, Julie, and Bogdan, Malgorzata. *varclust: Variables Clustering*, 2017b. URL https://github.com/psobczyk/varclust. R package version 0.9.4.

Sobczyk, Piotr; Josse, Julie, and Bogdan, Malgorzata. *pesel: Automatic estimation of number of principal components in PCA*, 2018. URL https://github.com/psobczyk/pesel. R package version 0.4.1.

Sobczyk, Piotr; Wilczyński, Stanisław; Bogdan, Małgorzata; Josse, Julie, and Staniak, Mateusz. Varclust: R package for variable clustering. 2019.

Soltanolkotabi, Mahdi; Elhamifar, Ehsan, and Candes, Emmanuel J. Robust subspace clustering, 2013.

Sołtys, M. Metody analizy skupień. Master's thesis, Wrocław University of Technology, 2010.

Steck, G. P. and Owen, D. B. A note on the equicorrelated multivariate normal distribution. *Biometrika*, 49 (1/2):269–271, 1962. ISSN 00063444. URL http://www.jstor.org/stable/2333495.

Tibshirani, Robert. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996. ISSN 00359246. URL http://www.jstor.org/stable/2346178.

Timmerman, Marieke E.; Ceulemans, Eva; De Roover, Kim, and Van Leeuwen, Karla. Subspace k-means clustering. *Behavior Research Methods*, 45(4):1011–1023, 2013.

Tipping, Michael E. and Bishop, Chris M. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61:611–622, 1999a.

Tipping, Michael E. and Bishop, Christopher M. Mixtures of probabilistic principal component analyzers. *Neural Comput.*, 11(2):443–482, February 1999b.

Treister, Eran and Turek, Javier S. A block-coordinate descent approach for large-scale sparse inverse covariance estimation. In *Advances in Neural Information Processing Systems 27*, pages 927–935. 2014.

van der Vaart, A.W. *Asymptotic Statistics*. Asymptotic Statistics. Cambridge University Press, 2000. ISBN 9780521784504. URL `https://books.google.pl/books?id=UEuQEM5RjWgC`.

Vidal, R. Subspace clustering. *Signal Processing Magazine, IEEE*, 28(2):52–68, 2011.

Vidal, René and Favaro, Paolo. Low rank subspace clustering (lrsc). *Pattern Recognition Letters*, 43(0):47 – 61, 2014. {ICPR2012} Awarded Papers.

Vigneau, E. and Qannari, E. M. Clustering of variables around latent components. *Communications in Statistics - Simulation and Computation*, 32(4):1131–1150, 2003.

Witten, R. and Candès, E. J. Randomized algorithms for low-rank matrix factorizations: sharp performance bounds. *Algorithmica*, 72(1):264–281, 2013.

Yuan, Ming. High dimensional inverse covariance matrix estimation via linear programming. *Journal of Machine Learning Research*, 11:2261–2286, 2010.

Yuan, Ming and Lin, Yi. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1): 19–35, 2007.

Zhao, Peng and Yu, Bin. On model selection consistency of lasso. *J. Mach. Learn. Res.*, 7:2541–2563, December 2006. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=1248547.1248637`.