PART I

MOTIVATION AND THE BASICS

CHAPTER 1

INTRODUCTION

Is it possible to estimate a covariance matrix using the regression methodology? If so, then one may bring the vast machinery of regression analysis (regularized estimation, parametric and nonparametric methods, Bayesian analysis, ...) developed in the last two centuries to the service of covariance modeling.

In this chapter, through several examples, we show that sparse estimation of high-dimensional covariance matrices can be reduced to solving a series of regularized regression problems. The examples include sparse principal component analysis (PCA), Gaussian graphical models, and the modified Cholesky decomposition of covariance matrices. The roles of sparsity, the least absolute shrinkage and smooth operator (Lasso) and particularly the soft-thresholding operator in estimating the parameters of linear regression models with a large number of predictors and large covariance matrices are reviewed briefly.

Nowadays, high-dimensional data are collected routinely in genomics, biomedical imaging, functional magnetic resonance imaging (fMRI), tomography, and finance. Let $\boldsymbol{X}$ be an $n \times p$ data matrix where n is the sample size and p is the number of variables. By the *high-dimensional data* usually it is meant that p is bigger than n. Analysis of high-dimensional data often poses challenges which calls for new statistical methodologies and theories (Donoho, 2000). For example, least-squares fitting of linear models and classical multivariate statistical methods cannot handle high-dimensional $\boldsymbol{X}$ since both rely on the inverse of $\boldsymbol{X}'\boldsymbol{X}$ which could be singular or not well-conditioned. It should be noted that increasing n and p each has very different and opposite effects on the statistical results. In general, the focus of multivariate analysis is to make statistical inference about the dependence among variables so

High-Dimensional Covariance Estimation, First Edition. Mohsen Pourahmadi.

that increasing n has the effect of improving the precision and certainty of inference, whereas increasing p has the opposite effect of reducing the precision and certainty. Therefore the level of detail that can be inferred about correlations among variables improves with increasing n but it deteriorates with increasing p.

The dimension reduction and variable selection are of fundamental importance for high-dimensional data analysis. The *sparsity principle* which assumes that only a small number of predictors contribute to the response is frequently adopted as the guiding light in the analysis. Armed with the sparsity principle, a large number of estimation approaches are available to estimate sparse models and select the significant variables simultaneously. The Lasso method introduced by Tibshirani (1996) is one of the most prominent and popular estimation methods for the high-dimensional linear regression models.

Quantifying the interplay between high-dimensionality and sparsity is important in the modern data analysis environment. In the classic setup, usually p is fixed and n grows so that

$$\frac{p}{n} \ll 1.$$

However, in the modern high-dimensional setup where both p and n can grow, estimating accurately a vector β with p parameters is a real challenge. By invoking the *sparsity principle* one assumes that only a small number, say $s(\beta)$, of the entries of β is nonzero and then proceeds in developing algorithms to estimate the nonzero parameters. Of course, it is desirable to establish the statistical consistency of the estimates under a new asymptotic regime where both $n, p \to \infty$. Interestingly, it has emerged from such asymptotic theory that for the consistency to hold in some generic problems, the dimensions n, p of the data and the *sparsity index* of the model must satisfy

$$\frac{\log p}{n} \cdot s(\beta) \ll 1. \tag{1.1}$$

The ratio $\log p/n$ does play a central role in establishing consistency results for variety of covariance estimators proposed for high-dimensional data in recent years (Bühlmann and van de Geer, 2011).

1.1 LEAST SQUARES AND REGULARIZED REGRESSION

The idea of least squares estimation of the regression parameters in the familiar linear model

$$\boldsymbol{Y} = \boldsymbol{X}\beta + \boldsymbol{\varepsilon}, \tag{1.2}$$

has served statistics quite well when the sample size n is large and p is *fixed* and small, say less than 50. The principle of model simplicity or *parsimony* coupled

with the techniques of subset, forward, and backward selections have been developed and used to fit such models either for the purpose of describing the data or for its prediction.

The whole machinery of least-squares fails or does not work well for the high-dimensional data where the ubiquitous $\boldsymbol{X}'\boldsymbol{X}$ matrix is not invertible. The traditional remedy is the ridge regression (Hoerl and Kennard, 1970), which replaces the residual sum of squares of errors by its penalized version:

$$Q(\beta) = \|\boldsymbol{Y} - \boldsymbol{X}\beta\|^2 + \lambda \sum_j |\beta_j|^2, \tag{1.3}$$

where $\lambda > 0$ is a penalty controlling the length of the vector of regression parameters. The unique ridge solution is

$$\widehat{\beta}_{\text{ridge}} = (\boldsymbol{X}'\boldsymbol{X} + \lambda \mathbf{I})^{-1}\boldsymbol{X}'\boldsymbol{Y}, \tag{1.4}$$

which amounts to adding λ to the diagonal entries of $\boldsymbol{X}'\boldsymbol{X}$, and then inverting it. The ridge solution works rather well in the presence of multicollinearity and when p is not too large; however, in general it does not induce sparsity in the model. Nevertheless, it points to the fruitful direction of penalizing a norm of the high-dimensional vector of coefficients (parameters) in the model.

In the modern context of high-dimensional data, the standard goals of regression analysis have also shifted toward:

(I) construction of *good predictors* where the actual values of coefficients in the model are *irrelevant*;

(II) giving causal interpretations of the factors and determining which variables are more *important*.

It turns out that regularization is important for both of these goals, but the appropriate magnitude of the regularization parameter depends on which goal is more important for a given problem. Historically, Goal (II) has been the engine of the statistical developments and the thought of irrelevancy of the parameter values was not imaginable. However, nowadays Goal (I) is the primary focus of developing algorithms in the machine learning theory (Bühlmann and van de Geer, 2011). In general, the pair $(\boldsymbol{Y}, \boldsymbol{X})$ is usually modeled nonparametrically as

$$\boldsymbol{Y} = m(\boldsymbol{X}) + \boldsymbol{\varepsilon}, \tag{1.5}$$

with $E(\boldsymbol{\varepsilon}) = 0$ and $m(\cdot)$ a smooth unknown function. Then using a family of basis functions $b_j(\boldsymbol{X})$, $j = 1, 2, \cdots$, $m(\cdot)$ is approximated closely with the sums: $\sum_{j=1}^{p} \beta_j b_j(\boldsymbol{X})$, for a large p, where $\beta = (\beta_1, \cdots, \beta_p)'$ is a vector of coefficients. Of course, when $b_i(\boldsymbol{X}) = \boldsymbol{X}_j$ is the jth column of the design matrix, then the approximation scheme reduces to the familiar linear regression model.

In the high-dimensional contexts, the ridge or ℓ_2 penalty on $||\beta||_2^2 = \sum_{j=1}^p \beta_j^2$ has lost some of its attractions to competitors like the Lasso which penalizes large values of the sum $\sum_{j=1}^p |\beta_j| = ||\beta||_1$ and hence forces many of the smaller β_j's to be estimated by zero. This zeroing of the coefficients is seen as *model selection* in the sense that only variables with $\beta_j \neq 0$ are included. To this end, perhaps a more natural penalty function is

$$P(\beta) = \sum_{j=1}^{p} I(\beta_j \neq 0) = ||\beta||_0, \tag{1.6}$$

which counts the number of nonzero coefficients. However, the ℓ_0 norm is not an easy function to work with so far as optimization is concerned as it is neither smooth nor convex. Fortunately, the Lasso penalty is the closest convex member of the family of penalty functions of the form $||\beta||_\alpha^\alpha = \sum_{j=1}^p |\beta|^\alpha$, $\alpha > 0$ to (1.6).

1.2 LASSO: SURVIVAL OF THE BIGGER

In this section, we indicate that the Lasso regression which corresponds to replacing the ridge penalty in (1.3) by the ℓ_1 penalty on the coefficients leads to more sparse solutions than the ridge penalty. It forces to zero the smaller coefficients, but keeps the bigger ones around.

The Lasso regression is one of the most popular approaches for selecting significant variables and estimating regression coefficients simultaneously. It corresponds to a penalized least-squares regression with the ℓ_1 penalty on the coefficients (Tibshirani, 1996). Compared to the ridge penalty, it minimizes the sum of squares of residuals subject to a constraint on the sum of absolute values of the regression coefficients:

$$Q(\beta) = \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{X}\beta\|^2 + \lambda \sum_j |\beta_j|, \tag{1.7}$$

where $\lambda > 0$ is a penalty or tuning parameter controlling the sparsity of the model or the magnitude of the estimates. It is evident that for larger values of the tuning parameter λ, the Lasso estimate $\widehat{\beta}(\lambda)$ obtained by minimizing (1.7) shrinks or forces the regression coefficients toward zero. Since the regularization parameter controls the model complexity, its proper selection is of critical importance in the applications of the Lasso and other penalized least-squares/likelihood methods. In most of what follows in the sequel, it is assumed that λ is fixed and known.

Unlike the closed-form ridge solution in (1.4), due to the nature of the constraint in (1.7), its solution is nonlinear in the responses Y_i's, see (1.11). Fundamental to understanding and computing the Lasso solution is the *soft-thresholding operator*.

Its relevance is motivated using the simple problem of minimizing the function (1.7) for $n = p = 1$, or for a generic observation y and $\boldsymbol{X} = 1$:

$$Q(\beta) = \frac{1}{2}(y - \beta)^2 + \lambda|\beta|, \tag{1.8}$$

for a fixed λ. Note that $|\beta|$ is a differentiable function at $\beta \neq 0$ and the derivative $Q(\cdot)$ with respect to such a β is

$$Q'(\beta) = -y + \beta + \lambda \cdot \text{sign}(\beta) = 0, \tag{1.9}$$

where $\text{sign}(\beta)$ is defined through $|\beta| = \beta \cdot \text{sign}(\beta)$. By convention, its value at zero is set to be zero. The explicit solution of β in terms of y, λ from (1.9) is

$$\widehat{\beta}(\lambda) = \text{sign}(y)(|y| - \lambda)_+, \tag{1.10}$$

where $(x)_+ = x$, if $\quad x > 0 \quad$ and 0, otherwise (see Section 2.5 for details). This simple closed-form solution reveals the following two fundamental characteristics of the Lasso solution:

1. Increasing the penalty λ will shrink the Lasso estimate $\widehat{\beta}(\lambda)$ toward 0. In fact, as soon as λ exceeds $|y|$, the Lasso estimate $\widehat{\beta}(\lambda)$ becomes zero and will remain so thereafter
2. The Lasso solution is *piecewise linear* in the penalty λ (see Fig. 1.1).

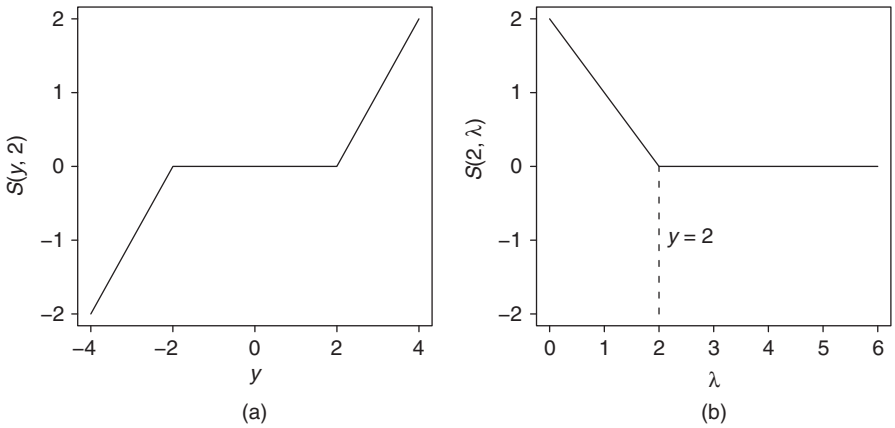

FIGURE 1.1 (a) Plot of the soft-thresholding operator for $\lambda = 2$. (b) Piecewise linearity of Lasso estimator for $y = 2$.

The function (1.10), the building block for solving many penalized regression problems, is called the *soft-thresholding operator*. The following generic notation will be used in the sequel:

$$S(y, \lambda) = \text{sign}(y)(|y| - \lambda)_+, \tag{1.11}$$

where y and λ are real and nonnegative numbers, respectively. There is the related *hard-thresholding function* at threshold λ defined by

$$H(y, \lambda) = yI(|y| > \lambda),$$

where $I(A)$ is the indicator function of the set A, which is also important in the literature of sparse estimation (see Problem 1). Note that these two functions are both nonlinear in y, and have in common the *threshold region* $|y| \leq \lambda$ where the signal is estimated to be zero.

The following example shows the role of the soft-thresholding operator in finding the slope of the simple linear regression with no intercept.

Example 1 (Simple Linear Regression with No Intercept) *Suppose* $\boldsymbol{Y} = (y_1, \cdots, y_n)'$, $\boldsymbol{X} = (x_1, \cdots, x_n)'$, *and consider minimizing:*

$$Q(\beta) = \frac{1}{2}||\boldsymbol{Y} - \beta \boldsymbol{X}||^2 + \lambda|\beta|.$$

For $\lambda = 0$, *the task is to find the least-square estimate of* β *which is given by* $\widehat{\beta} = \sum_{i=1}^n x_i y_i / \sum_{i=1}^n x_i^2$. *In what follows, for simplicity, we assume that the x's are normalized so that* $\sum_{i=1}^n x_i^2 = 1$. *Then, for* $\lambda > 0$, *the Lasso estimate satisfies*

$$Q'(\beta) = -\sum_1^n x_i(y_i - \beta x_i) + \lambda \cdot sign(\beta) = -\sum_{i=1}^n x_i y_i + \beta + \lambda \cdot sign(\beta).$$

Comparing with (1.9), it follows that the Lasso estimate of the slope parameter β *is*

$$\widehat{\beta}(\lambda) = S\left(\sum_{i=1}^n x_i y_i, \lambda\right), \tag{1.12}$$

where the first argument $\widehat{\beta}$ *is the least-squares estimate in a simple linear regression with no intercept.*

The setup of Example 1 and the closed form of its solution play important roles in solving the sparse PCA problem using the singular value decomposition (SVD) of the data matrix $\boldsymbol{X}$ described later in Section 1.4 and Chapter 4.

For a general design matrix $\boldsymbol{X}$, the computational details of the Lasso solution are more involved and will be discussed in Chapter 2. A fast method to compute it is the *coordinate descent algorithm* which minimizes (1.7) over one β_j at a time with the others kept fixed. It then cycles through all the parameters until its convergence (see Example 11, Friedman et al., 2008; Bühlmann and van de Geer, 2011). For an excellent retrospective review of the basic idea of Lasso, its history, computational developments, and generalizations since the publication of the original paper in 1996 see Tibshirani (2011).

Due to its ability to force to zero smaller parameters, the Lasso penalty and method have been applied to sparse PCA, sparse estimation of covariance and precision matrices, Gaussian graphical models, and multivariate regression. Some of these applications are reviewed briefly in this chapter. Many other sparsity-inducing penalties such as the smoothly clipped absolute deviation (SCAD) (Fan and Li, 2001), the elastic net (Zou and Hastie, 2005), and the adaptive Lasso (Zou, 2006), reviewed in Chapter 2, can be used in the context of covariance estimation. However, for clarity and simplification in notation throughout this book we rely on the Lasso penalty.

1.3 THRESHOLDING THE SAMPLE COVARIANCE MATRIX

For a column-centered $n \times p$ data matrix $\boldsymbol{X}$, it is known that the sample covariance matrix $\mathbf{S} = \frac{1}{n}\boldsymbol{X}'\boldsymbol{X}$ performs poorly in high dimensions. In analogy with the nonparametric estimation of mean vectors (Johnstone, 2011), when the population covariance matrix is sparse one may regularize $\mathbf{S}$ by applying the hard- or soft-thresholding operator (1.11) to it elementwise.

More formally, a soft-thresholded covariance estimator is defined by

$$\widehat{\boldsymbol{\Sigma}}_\lambda = S(\mathbf{S}, \lambda),$$

where $\lambda > 0$ is a tuning parameter and $S(\cdot, \lambda)$ acts componentwise. Interestingly, such a regularized covariance estimator is the solution of the following convex optimization problem:

$$\begin{aligned}
\widehat{\boldsymbol{\Sigma}}_\lambda &= \underset{\boldsymbol{\Sigma}}{\operatorname{argmin}} \left\{ \frac{1}{2} ||\boldsymbol{\Sigma} - \mathbf{S}||_F^2 + \lambda |\boldsymbol{\Sigma}|_1 \right\}, \\
&= \underset{\boldsymbol{\Sigma}}{\operatorname{argmin}} \left\{ \frac{1}{2} \sum_{i=1}^{p} \sum_{j=1}^{p} (\sigma_{ij} - s_{ij})^2 + \lambda \sum_{i=1}^{p} \sum_{j \neq i}^{p} |\sigma_{ij}| \right\}, \qquad (1.13) \\
&= \underset{\boldsymbol{\Sigma}}{\operatorname{argmin}} \sum_{i=1}^{p} \sum_{j \neq i}^{p} \left\{ \frac{1}{2} (\sigma_{ij} - s_{ij})^2 + \lambda |\sigma_{ij}| \right\},
\end{aligned}$$

where $||\cdot||_F$ stands for the Frobenius norm and $|\cdot|_1$ stands for the ℓ_1 norm of the nondiagonal elements of its matrix argument. Note that the optimization problem above is the same as minimizing for each $i \neq j$, the expression inside the last curly

bracket which is of the form of (1.8) and hence amounts to soft-thresholding the entry s_{ij}.

It is known that under some regularity conditions such a simple, sparse covariance estimator is consistent in high dimensions (see Chapter 6; Bickel and Levina, 2008b; Rothman et al., 2009). A consequence of the consistency of the thresholded estimator is that its positive-definiteness is guaranteed in the asymptotic setting with high probability. However, the actual estimator is not necessarily a positive-definite matrix in real data analysis. This is illustrated clearly by Xue et al. (2012) using the Michigan lung cancer data with $n = 86$ tumor samples from patients with lung adenocarcinomas and 5217 gene expression values for each sample. Randomly choosing p genes ($p = 200, 500$) they obtained the $p \times p$ soft-thresholding sample correlation matrix for these genes, and repeated the process 10 times for each p, each time the thresholding parameter λ was selected via a five-fold cross-validation. It was found that none of the soft-thresholding estimators were positive-definite. On the average, there were 22 and 124 negative eigenvalues for the soft-thresholding estimator for $p = 200$ and 500, respectively.

To deal with the lack of positive-definiteness, one popular solution is to utilize the eigendecomposition of $\mathbf{S}$ and project it into the convex cone of positive-definite matrices. Let

$$\mathbf{S} = \sum_{i=1}^{p} \widehat{\lambda}_i \widehat{e}_i \widehat{e}_i',$$

be its eigendecomposition then a positive semidefinite estimator can be obtained by setting

$$\tilde{\mathbf{S}}^{+} = \sum_{i=1}^{p} \max(\widehat{\lambda}_i, 0) \widehat{e}_i \widehat{e}_i'.$$

Unfortunately, this strategy does not work well for sparse covariance matrix estimation, because the projection destroys the sparsity pattern of $\mathbf{S}$. For the Michigan lung cancer data, after the semidefinite projection, the soft-thresholding estimator has no zero entry.

In order to simultaneously achieve sparsity and positive-definiteness, a natural approach is to add a positive-definiteness constraint to the objective function (see Chapter 6 and Rothman, 2012).

1.4 SPARSE PCA AND REGRESSION

In this section, we present a simple regression-based procedure for regularizing the principal components (PCs) using the close connections among the SVD of the column-centered data matrix $\boldsymbol{X}$, the eigen-decomposition of its sample covariance matrix $\mathbf{S}$, and the low-rank approximation property of SVD in the Frobenius norm of a matrix (see Section 4.3).

One of the most popular techniques in multivariate statistics is the principal component analysis (PCA) (see Section 3.5). It is used for dimension reduction, data visualization, data compression, and information extraction by relying on the first few eigenvectors of the sample covariance matrix. For a data matrix $\boldsymbol{X}$, PCA finds sequentially unit vectors $\mathbf{v}_1, \cdots, \mathbf{v}_p$ that maximize $\mathbf{v}'\boldsymbol{X}'\boldsymbol{X}\mathbf{v}$ subject to $\mathbf{v}_{i+1}$ being orthogonal to the previous vectors $\mathbf{v}_1, \cdots, \mathbf{v}_i$. These are the (sample) *principal component (PC) loadings* and the vectors $\boldsymbol{X}\mathbf{v}_i$ are the *principal components (PCs)*. A major problem in the high-dimensional data situations is that the classical PCs have poor statistical properties in the sense that the sample eigenvectors are not consistent estimators of their population counterparts (Johnstone and Lu, 2009). Thus, it is necessary to regularize them in some manners.

We focus on the recent growing interest in developing sparse PCA via the SVD of a data matrix using the following simple matrix algebra facts:

1. The normalized right singular vectors of the column-centered data matrix $\boldsymbol{X}$ and the normalized eigenvectors of the sample covariance matrix $\mathbf{S} = 1/n\boldsymbol{X}'\boldsymbol{X}$ are the same (see Section 4.3).
2. The best rank-one approximation to a data matrix $\boldsymbol{X}$ in the Frobenius norm is given by a constant multiple of $\boldsymbol{u}_1\mathbf{v}_1'$ where $\boldsymbol{u}_1$ and $\mathbf{v}_1$ are its first left and right singular vectors.

To regularize an approximant $\boldsymbol{u}\mathbf{v}'$, we restrict $\boldsymbol{u}$ to have unit length, that is $||\boldsymbol{u}||_2^2 = \sum_{i=1}^n u_i^2 = 1$, and assume that $\mathbf{v}$ is sparse with many zeros. Then, imposing a Lasso penalty on the vector of PC loadings the task is to minimize the penalized objective function:

$$\|\boldsymbol{X} - \boldsymbol{u}\mathbf{v}\|_F^2 + \lambda \sum_{j=1}^{p} |v_j| = \sum_j \left\{ \sum_i (x_{ij} - u_i v_j)^2 + \lambda |v_j| \right\}, \tag{1.14}$$

subject to $||\boldsymbol{u}||_2 = 1$, where v_j is the jth entry of the vector of PC loadings.

The full computational advantage of the *regression-based approach* to PCA emerges by noting that the inner sum in (1.14), for each j, is precisely the penalized regression problem studied in Example 1. Thus, for a fixed $\boldsymbol{u}$, the solution vector $\mathbf{v}$ is the (penalized) regression coefficient when the p columns of $\boldsymbol{X}$ are regressed on $\boldsymbol{u}$ one at a time (Shen and Huang, 2008). Of course, for a fixed $\mathbf{v}$ a similar and much simpler statement is true about computing $\boldsymbol{u}$. These useful facts are summarized in the following lemma:

Lemma 1 (Regression-based SVD)

(a) For a fixed $\mathbf{v}$*, the minimizer of (1.14) with* $||\boldsymbol{u}||_2 = 1$ *is given by*

$$\boldsymbol{u} = \frac{\boldsymbol{X}\mathbf{v}}{||\boldsymbol{X}\mathbf{v}||_2}.$$

(b) For a fixed $\boldsymbol{u}$*, the minimizer of (1.14) is given by*

$$\mathbf{v} = S\left(\boldsymbol{X}'\boldsymbol{u}, \lambda\right),$$

where the soft-thresholding operator acts componentwise on the vector $\boldsymbol{X}'\boldsymbol{u}$.

Lemma 1 suggests the following iterative procedure for solving the optimization problem (1.14):

The Shen and Huang (2008) Algorithm for Minimizing (1.14):

1. Initialize $\mathbf{v}$ with $\|\mathbf{v}\|_2 = 1$
2. Iterate until convergence:
   (a) $\boldsymbol{u} \leftarrow \frac{\boldsymbol{X}\mathbf{v}}{\|\boldsymbol{X}\mathbf{v}\|_2}$,
   (b) $\mathbf{v} \leftarrow S(\boldsymbol{X}'\boldsymbol{u}, \lambda)$.

One may initialize $\mathbf{v}$ using the first right singular vector of the standard SVD of $\boldsymbol{X}$. In the literature of PCA, it is common for $\mathbf{v}$ to have unit length for the sake of identifiability. This can be done at the convergence by normalizing the PC loadings to have $||\mathbf{v}||_2 = 1$.

Interestingly, the previous algorithm is closely connected to the *power or alternating least-squares (ALS) method* for computing the standard SVD of a data matrix. In fact, for $\lambda = 0$ since $S\left(\boldsymbol{X}'\boldsymbol{u}, 0\right) = \boldsymbol{X}'\boldsymbol{u}$, it reduces to the *power method* for computing the eigenvectors (see Golub and Van Loan, 1996; Gabriel and Zamir, 1979). Starting with $\mathbf{v}^{(0)}$ it is seen from the algorithm that at the end of the kth iteration

$$\mathbf{v}^{(k)} = \frac{(\boldsymbol{X}'\boldsymbol{X})^k \mathbf{v}^{(0)}}{||(\boldsymbol{X}'\boldsymbol{X})^k \mathbf{v}^{(0)}||_2}, \tag{1.15}$$

which computes the largest eigenvector of $\boldsymbol{X}'\boldsymbol{X}$ or the leading right singular vector of $\boldsymbol{X}$. This connection is made more transparent and discussed further in Section 4.3.

Thus far, our focus has been on estimating a covariance matrix $\boldsymbol{\Sigma}$. In some applications, such as model-based clustering, classification, and regression, the need for the precision matrix $\boldsymbol{\Sigma}^{-1}$ is stronger than that for $\boldsymbol{\Sigma}$ itself. In the standard multivariate statistics, the former is usually computed from the latter. However, since inversion tends to destroy sparsity and may introduce noise for large p, it is advantageous to treat estimating a sparse covariance matrix and sparse precision matrix as completely separate problems. Furthermore, inversion of a $p \times p$ matrix requires $\mathcal{O}(p^3)$ operations which in the high-dimensional situation is computationally expensive and can be avoided by using the techniques explained next.

1.5 GRAPHICAL MODELS: NODEWISE REGRESSION

Gaussian graphical models explore and display conditional dependence relationships among Gaussian random variables. A connection between conditional independence of variables in a Gaussian random vector and graphs is made by identifying the graph's nodes with random variables and translating the graph's edges into a parametrization that relates to the precision matrix of a multivariate normal distribution.

In this section, an overview of a regression-based approach to inducing sparsity in the precision matrix is given. Sparsity of the precision matrix is of great interest in the literature of Gaussian graphical models (cf. Whittaker, 1990; Hastie et al., 2009, Chap. 17). The key idea is that in a multivariate normal random vector $\boldsymbol{Y}$, the conditional dependencies among its entries are related to the off-diagonal entries of its precision matrix $\boldsymbol{\Sigma}^{-1} = (\sigma^{ij})$. More precisely, the variables i and j are conditionally independent given all other variables, if and only if $\sigma^{ij} = 0$, so that the problem of estimating a Gaussian graphical model is equivalent to estimating a precision matrix.

To cast the problem in the language of linear regression, let $\widehat{Y}_i = \sum_{j\neq i} \beta_{ij} Y_j$ be the linear least-squares predictor of Y_i based on the rest of the components of $\boldsymbol{Y}$ and $\varepsilon_i = Y_i - \widehat{Y}_i$ be its prediction error. Then, writing

$$Y_i = \sum_{j\neq i} \beta_{ij} Y_j + \varepsilon_i, \tag{1.16}$$

we show in Section 5.2 that the coefficients $\beta_{i,j}$ of Y_i based on the remaining variables and the corresponding prediction error variances are given by:

$$\beta_{i,j} = -\frac{\sigma^{ij}}{\sigma^{ii}},\ j \neq i, \quad \mathrm{Var}(Y_i | Y_j,\ j \neq i) = \frac{1}{\sigma^{ii}},\ i = 1, \cdots, p. \tag{1.17}$$

This confirms that σ^{ij}, the (i, j)th entry of the precision matrix is, up to a scalar, the coefficient of variable j in the multiple regression of the node or variable i on the rest. As such, each $\beta_{i,j}$ is an unconstrained real number with $\beta_{j,j} = 0$, but note that $\beta_{i,j}$ is not necessarily symmetric in (i, j).

Now, for each node i one may impose an ℓ_1 penalty on the regression coefficients in (1.16). The idea of nodewise Lasso regression, proposed first in Meinshausen and Bühlmann (2006), has been the source of considerable research in sparse estimation of precision matrices leading to the penalized normal likelihood estimation and the graphical Lasso (Glasso) algorithm discussed in Chapter 5.

1.6 CHOLESKY DECOMPOSITION AND REGRESSION

In this section, the close connection between the modified Cholesky decomposition of a covariance matrix and the idea of regression is reviewed when the variables are ordered or there is a notion of distance between variables, as in time series, longitudinal and functional data, and spectroscopic data.

In what follows, it is assumed that $\boldsymbol{Y}$ is a time-ordered random vector with mean zero and a positive-definite covariance matrix $\boldsymbol{\Sigma}$. Let $\widehat{Y}_t$ be the linear least-squares predictor of Y_t based on its predecessors $Y_{t-1}, \ldots, Y_1$ and $\varepsilon_t = Y_t - \widehat{Y}_t$ be its prediction error with variance $\sigma_t^2 = \text{var}(\varepsilon_t)$. Then, there are unique scalars ϕ_{tj} so that

$$Y_t = \sum_{j=1}^{t-1} \phi_{tj} Y_j + \varepsilon_t, \ t = 1, \cdots, p, \tag{1.18}$$

and the regression coefficients ϕ_{tj} can be computed using the covariance matrix as shown in Section 3.6. Let $\varepsilon = (\varepsilon_1, \cdots, \varepsilon_p)'$ be the vector of successive uncorrelated prediction errors with

$$\text{cov}(\varepsilon) = \text{diag}\left(\sigma_1^2, \cdots, \sigma_p^2\right) = D.$$

Then, (1.18) can be rewritten in matrix form as $T\boldsymbol{Y} = \varepsilon$, where T is the following unit lower triangular matrix with $-\phi_{tj}$ as its (t, j)th entry:

$$T = \begin{pmatrix} 1 & & & & \\ -\phi_{21} & 1 & & & \\ -\phi_{31} & -\phi_{32} & 1 & & \\ \vdots & & & \ddots & \\ -\phi_{p1} & -\phi_{p2} & \cdots & -\phi_{p,p-1} & 1 \end{pmatrix}. \tag{1.19}$$

Now, computing the covariance

$$\text{cov}(\varepsilon) = \text{cov}(T\boldsymbol{Y}) = T\boldsymbol{\Sigma}T',$$

one arrives at the decompositions

$$T\boldsymbol{\Sigma}T' = D, \ \boldsymbol{\Sigma}^{-1} = T'D^{-1}T. \tag{1.20}$$

Thus, both $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}^{-1}$ are diagonalized by unit lower triangular matrices; we refer to (1.20) as the modified Cholesky decompositions of the covariance and its precision matrix, respectively.

Since the ϕ_{ij}'s in (1.18) are simply the regression coefficients computed from an unstructured covariance matrix, these coefficients along with $\log \sigma_t^2$ are unconstrained (Pourahmadi, 1999). From (1.18) and the subsequent results, it is evident that the task of modeling a covariance matrix is reduced to that of a sequence of p varying-coefficient and varying-order regression models. Thus, one can bring the whole regression analysis machinery to the service of the unintuitive and challenging task of modeling covariance matrices. In particular, the Lasso penalty can be imposed on the regression coefficients in (1.18) as in Huang et al. (2006). An important

consequence of (1.20) is that for any estimate $(\widehat{T}, \widehat{D})$ of the Cholesky factors, the estimated covariance and precision matrix, $\widehat{\boldsymbol{\Sigma}}^{-1} = \widehat{T}'\widehat{D}^{-1}\widehat{T}$, are guaranteed to be positive-definite.

1.7 THE BIGGER PICTURE: LATENT FACTOR MODELS

A broader framework for covariance matrix reparameterization and modeling is the class of *latent factor models* (Anderson, 2003) or *linear mixed models* (Searle et al., 1992). It has the goal of finding a few common (latent, random) factors that may explain the covariance matrix of the data.

For a random vector $\boldsymbol{Y} = (Y_1, \ldots, Y_p)'$ with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, the classical factor model postulates that entries of $\boldsymbol{Y}$ are linearly dependent on a few *unobservable random variables* $F = (f_1, \ldots, f_q)'$, $q \ll p$, and p additional noises $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_p)'$. The f_i's are called *the common factors* and the ε_i's are the *idiosyncratic errors*. More precisely, a *factor model* for $\boldsymbol{Y}$ is

$$\begin{aligned}
Y_1 - \mu_1 &= \ell_{11} f_1 + \cdots + \ell_{1q} f_q + \varepsilon_1, \\
Y_2 - \mu_2 &= \ell_{21} f_1 + \cdots + \ell_{2q} f_q + \varepsilon_2, \\
&\vdots \\
Y_p - \mu_p &= \ell_{p1} f_1 + \cdots + \ell_{pq} f_q + \varepsilon_p,
\end{aligned}$$

or in matrix notation

$$\boldsymbol{Y} - \boldsymbol{\mu} = LF + \varepsilon, \tag{1.21}$$

where $L = (\ell_{ij})$ is the $p \times q$ matrix of *factor loadings* (see Section 3.7 for more details).

Algebraically, the factor model (1.21) amounts to decomposing a $p \times p$ covariance matrix as

$$\boldsymbol{\Sigma} = L\Lambda L' + \Psi, \tag{1.22}$$

where L is a $p \times q$ matrix, Λ and Ψ are usually $q \times q$ and $p \times p$ diagonal matrices, respectively. Note that the factor model swaps $\boldsymbol{\Sigma}$ by the triplet (L, Λ, Ψ) which leads to a considerable reduction in the number of covariance parameters if q is small relative to p. The generality of (1.22) becomes more evident by choosing the components of the quadruple (q, L, Λ, Ψ) in various ways as indicated below:

1. *Spectral decomposition or PCA* amounts to choosing $q = p$, $\Psi = 0$, L as the orthogonal matrix of eigenvectors, and Λ as the diagonal matrix of eigenvalues of the covariance matrix.
2. *Spiked covariance models* (Paul, 2007; Johnstone and Lu, 2009) are special orthogonal factor models obtained by choosing $q < p$, Λ the identity matrix,

and $\Psi = \sigma^2 I$. In this case, writing $L = (L_1, \cdots, L_q)$ where the columns L_j's are orthogonal with decreasing norms

$$||L_1||_2 > ||L_2||_2 > \cdots > ||L_q||_2,$$

leads to the decomposition

$$\mathbf{\Sigma} = \sum_{j=1}^{q} L_j L_j' + \sigma^2 I, \tag{1.23}$$

where the L_j's are the ordered eigenvectors of the population covariance matrix. The spiked covariance models are ideal for studying consistency of high-dimensional standard PCA. For example, it will be shown in Chapter 4 that the sparse PCA is consistent for the spiked models if the leading L_j's are sparse or concentrated vectors.

3. *Modified Cholesky decomposition* amounts to taking L to be a unit lower triangular matrix, Λ the diagonal matrix of innovation variances, and $\Psi = 0$.

A possible drawback of the standard latent factor models is that the decomposition (1.22) is not unique, see (3.40) and the subsequent discussions. A common approach to ensure uniqueness is to constrain the factor loadings matrix L to be block lower triangular matrix with strictly positive diagonal elements (Geweke and Zhou, 1996). Unfortunately, such a constraint induces order dependence among the responses or the variables in $\boldsymbol{Y}$ as in the Cholesky decomposition. However, for certain tasks such as prediction, identifiability of a unique decomposition may not be necessary.

The *precision matrix* is needed in variety of statistical applications (Anderson, 2003) and in portfolio management (Fan and Lv, 2008). For the latent factor model (1.22), it can be shown that

$$\mathbf{\Sigma}^{-1} = \Psi^{-1} - \Psi^{-1} L(\Lambda + L'\Psi^{-1}L)^{-1} L'\Psi^{-1}, \tag{1.24}$$

which involves inversion of smaller $q \times q$ matrices as Ψ is assumed to be diagonal in the standard factor models. This assumption may not be valid in some economics and financial studies (Chamberlain and Rothschild, 1983) in which case it is more appropriate to assume that Ψ is sparse.

When working with a collection of covariance matrices $\mathbf{\Sigma}_1, \cdots, \mathbf{\Sigma}_K$ sharing some similarities, the factor model framework can be used to model them as

$$\mathbf{\Sigma}_k = \mathbf{\Sigma} + L\Lambda_k L', k = 1, \cdots, K, \tag{1.25}$$

where $\mathbf{\Sigma}$ is a common covariance matrix, L is a $p \times q$ full rank matrix, and Λ_k is a $q \times q$ nonnegative definite matrix which is not necessarily diagonal. The maximum likelihood estimation of such a family of reduced covariance matrices is developed in Schott (2012). An important example of a collection of covariance matrices arises in

multivariate time-varying volatility $\mathbf{\Sigma}_t$, $t = 1, \cdots, T$ for p assets over T time periods. It has many applications in finance including asset allocation and risk management. Here, using a standard factor model one writes

$$\mathbf{\Sigma}_t = L_t \Lambda_t L_t' + \Psi_t, t = 1, \cdots, T, \tag{1.26}$$

where the $p \times q$ matrix of factor loadings L_t is block lower triangular matrix with diagonal elements equal to one, Λ_t, Ψ_t are the diagonal covariance matrices of the common and specific factors, respectively. In the literature of finance, (1.26) is referred to as the *factor stochastic volatility* model.

For the versatility of the latent factor models in dealing with large covariances arising from spatial data, financial data, etc., see Fox and Dunson (2011). They propose a Bayesian nonparametric approach to multivariate covariance regression where L in (1.22) and hence the covariance matrix are allowed to change flexibly with covariates or predictors. The approach readily scales to high-dimensional datasets. The JRSS B discussion paper by Fan et al. (2013) offers the most recent and diverse perspectives of research on the roles of factor models in covariance estimation.

1.8 FURTHER READING

An excellent unified view of regularization methods in statistics is given by Bickel and Li (2006). All regularization methods require a tuning parameter. Choosing it too small keeps many variables in the model and retains too much variance in estimation and prediction, but choosing it too large knocks out many variables and introduces too much bias. Thus, controlling the bias–variance tradeoff is the key in selecting the tuning parameter. Cross-validation is the most popular method which requires data splitting and have been only justified for low-dimensional settings. In the following, an outline of some recent progress and relevant references are provided for interested readers.

Theoretical justifications of most methods for tuning parameter selection are usually based on oracle choices of some related parameters (like the noise variance) that cannot be implemented in practice. Choosing the regularization parameter in a data-dependent way remains a challenging problem in a high-dimensional setup. Since the introduction of the degrees of freedom for Lasso regression (Zou et al., 2007), it seems some older techniques such as the C_p-statistic, Akaike information criterion (AIC), Bayesian information criterion (BIC) are becoming popular in the high-dimensional context. Significant progress has been made recently on developing likelihood-free regularization selection techniques like subsampling (Meinshausen and Bühlmann, 2010), which are computationally expensive and still may lack theoretical guarantees.

There is a growing desire and tendency to avoid data-based methods of selecting the tuning parameter by exploiting certain optimal value of the asymptotic thresholding parameter from the Gaussian sequence models (Johnstone, 2011; Yang et al., 2011;

Cai and Liu, 2011). A new procedure (Liu and Wang, 2012) for estimating high-dimensional Gaussian graphical models, called TIGER (tuning-insensitive graph estimation and regression), enjoys a *tuning-insensitive property*, in the sense that it automatically adapts to the unknown sparsity pattern and is asymptotically tuning-free. In finite sample settings, one only needs to pay minimal attention to tuning the regularization parameter. Its main idea, like that in Glasso, is to estimate the precision matrix one column at a time. For each column, the computation is reduced to a sparse regression problem where unlike the existing methods, the TIGER solves the sparse regression problem using the recent SQRT-Lasso method proposed by Belloni et al. (2012). The main advantage of the TIGER over the existing methods is its asymptotic tuning-free property, which allows one to use the entire data to efficiently estimate the model.

PROBLEMS

1. For $y \in R, \lambda > 0$, consider the objective function

$$f_\lambda(\beta) = \beta^2 - 2y\beta + p_\lambda(|\beta|),$$

where $p_\lambda(\cdot)$ is a penalty function. Let $\hat{\beta}$ be a minimizer of the objective function.

(a) For the ridge penalty $p_\lambda(|\beta|) = \lambda\beta^2$, show that the minimizer of the objective function is

$$R(y, \lambda) = \frac{y}{1+\lambda}.$$

(b) For $p_\lambda(|\beta|) = \lambda^2 I(|\beta| \neq 0)$, plot $f_2(\beta)$ when $y = 3$. Is the function continuous? Differentiable? Convex?

(c) Show that the minimizer of $f_\lambda(y)$ is the hard-thresholding function

$$H(y, \lambda) = yI(|y| > \lambda).$$

(d) Repeat (c) for the penalty function $p_\lambda(|\beta|) = 2\lambda|\beta|$ and show that the minimizer of the objective function is the soft-thresholding function $S(y, \lambda)$.

(e) Plot $H(y, \lambda)$ and $S(y, \lambda)$ as functions of the data y for $\lambda = 1, 5, 7$. What are the similarities and differences between these two penalized least-squares estimators?

2. *Soft–hard-thresholding function*: Plot the function

$$SH(y, \lambda_1, \lambda_2) = \begin{cases} 0 & \text{if} \quad |y| \leq \lambda_1, \\ \text{sign}(y)\frac{\lambda_2(|y|-\lambda_1)}{\lambda_2-\lambda_1} & \text{if} \quad \lambda_1 \leq |y| \leq \lambda_2, \\ y & \text{if} \quad |y| > \lambda_2, \end{cases}$$

for $\lambda_1 = 1, \lambda_2 = 2$. Compare its similarities and differences with the hard- and soft-thresholding functions.

3. *Smoothly Clipped Absolute Deviation (SCAD) Penalty* (Fan and Li, 2001) is given by

$$p_\lambda(|\beta|) = 2\lambda|\beta| I(|\beta| \le \lambda) - \frac{\beta^2 - 2a\lambda|\beta| + \lambda^2}{a-1} I(\lambda < |\beta| \le a\lambda) + (a+1)\lambda^2 I(|\beta| > a\lambda),$$

where $a > 2$ is a second tuning parameter (here it is fixed at 3.7).

(a) Plot the function $p_2(|\beta|)$. Is the function convex (concave)?

(b) Show that the minimizer of the objective function in Problem 1 with the SCAD penalty function is

$$\hat{\beta}(y,\lambda) = \begin{cases} \text{sign}(y)(|y| - \lambda)_+ & \text{if} \quad |y| \le 2\lambda, \\ \{(a-1)y - \text{sign}(y)a\lambda\}/(a-2) & \text{if} \quad 2\lambda < |y| \le a\lambda, \\ y & \text{if} \quad |y| > a\lambda. \end{cases}$$

(c) What are the similarities and differences between the penalized least-squares estimators $S(y,\lambda)$ and $\hat{\beta}(y,\lambda)$ corresponding to the soft-thresholding and SCAD penalties?

(d) Compare the SCAD penalty function with the following simpler penalty functions:

$$p_\lambda(|\beta|) = \lambda|\beta|/(1+|\beta|), \quad p'_\lambda(|\beta|) = (a\lambda - |\beta|)_+/a,$$

where $p'(\cdot)$ stands for the derivative of the function $p(\cdot)$.

4. Consider the problem of minimizing

$$||\boldsymbol{X} - d\boldsymbol{u}\mathbf{v}'||_F^2 + d(\lambda_u||\boldsymbol{u}||_1 + \lambda_v||\mathbf{v}||_1), \text{ subject to } \quad ||\boldsymbol{u}||_2 = ||\mathbf{v}||_2 = 1, \quad (1.27)$$

with respect to $(d, \boldsymbol{u}, \mathbf{v})$ where λ_u, λ_v are fixed tuning parameters.

(a) Show that for $\boldsymbol{u}$ fixed, minimizing the function (1.27) with respect to $(d, \mathbf{v})$ is equivalent to minimizing with respect to $\tilde{\mathbf{v}} = d\mathbf{v}$ the function

$$||\boldsymbol{X} - \boldsymbol{u}\tilde{\mathbf{v}}'||_F^2 + \lambda_v \sum_{j=1}^{p} |\tilde{v}_j| = ||\boldsymbol{Y} - (I_p \otimes \boldsymbol{u})\tilde{\mathbf{v}}||^2 + \lambda_v \sum_{j=1}^{p} |\tilde{v}_j|, \quad (1.28)$$

where $\boldsymbol{Y} = (\mathbf{x}_1', \cdots, \mathbf{x}_p')' \in R^{np}$ with $\mathbf{x}_j$ being the jth column of $\boldsymbol{X}$ and $\otimes$ is the Kronecker product.

(b) Observe that the right-hand side of (1.28) is a Lasso penalized regression problem with the response $\boldsymbol{Y}$, the design matrix $I_p \otimes \boldsymbol{u}$, and the vector of regression coefficients $\tilde{\mathbf{v}}$. Given the first two and λ_v, what algorithm would you use to compute the Lasso estimate of $\tilde{\mathbf{v}}$? Why?

(c) Find the analog of (1.28) for fixed $\mathbf{v}$ when minimizing the function (1.27) with respect to $(d, \boldsymbol{u})$.