Overview of the DESP package

Arnak Dalalyan
CREST/ENSAE

Samuel Balmand
MATIS/ENSG

Abstract

We present briefly the scope of the **DESP** package, which aims to estimate robustly the parameters of a Gaussian distribution—even in high dimension—with particular attention to the diagonal elements of the precision matrix.

Keywords: precision matrix, high-dimension, sparse, robust, diagonal elements, R.

1 Introduction

1.1 Purpose of this package

The **DESP** package[1] is designed to estimate efficiently the parameters of a Gaussian distribution as developed in [1, 2]. To be precise, it estimates the inverse of the covariance matrix (also known as precision or concentration matrix) and the expectation. Its main characteristics are the ability to deal with data contaminated by outliers, with high-dimensional data and the availability of several estimators of the diagonal elements of the precision matrix.

In this document, we zoom in on the function `desp()` which is an interface to most of the features of the package.

1.2 Notation

We denote by $\mathcal{N}_p(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$ the Gaussian distribution in $\mathbb{R}^p$ with mean $\boldsymbol{\mu}^*$ and covariance matrix $\boldsymbol{\Sigma}^*$. The precision matrix $\left(\boldsymbol{\Sigma}^*\right)^{-1}$ is denoted by $\boldsymbol{\Omega}^*$. We denote by $\mathbf{1}_n$ the vector from $\mathbb{R}^n$ with all the entries equal to 1, by $\mathbf{u}_n$ the vector $\mathbf{1}_n/\sqrt{n}$ and by $\mathbf{I}_n$ the $n \times n$ identity matrix. The transpose of the matrix $\mathbf{M}$ is denoted by $\mathbf{M}^\top$. The kth row (resp. jth column) of a $n \times p$ matrix $\mathbf{M}$ is denoted by $\mathbf{M}_{k,\bullet}$ (resp. $\mathbf{M}_{\bullet,j}$). We use the following notation for the (pseudo-)norms of matrices: if $q_1, q_2 > 0$, then

$$\|\mathbf{M}\|_{q_1,q_2} = \left\{ \sum_{i=1}^n \|\mathbf{M}_{i,\bullet}\|_{q_1}^{q_2} \right\}^{1/q_2}.$$

The cardinality of a set S is denoted by $|S|$.

1.3 Estimators

Let us first introduce the proposed estimator. We consider a possible additive contamination of the data by outliers. We assume that the matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ of observed data satisfies

$$\mathbf{X} = \mathbf{Y} + \mathbf{E}^*, \tag{1}$$

where $\mathbf{E}^*$ is the matrix of errors and $\mathbf{Y}$ the outlier-free data matrix. The rows latter are supposed to be independent Gaussian, such that $\mathbf{Y}_{i,\bullet} \sim \mathcal{N}_p(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$. We also assume that most of the rows of the matrix $\mathbf{E}^*$ correspond to inliers, hence are only filled with zeros. The $p \times p$ matrix $\mathbf{B}^*$ corresponds to $\boldsymbol{\Omega}^* \cdot \mathrm{diag}(\{1/\omega_{jj}^*\}_{j \in [p]})$. We introduce the matrix $\boldsymbol{\Theta}^* = \mathbf{E}^*\mathbf{B}^*/\sqrt{n}$ that has the same sparsity pattern as

---

[1]Licensed under GPL (version 3).

$\mathbf{E}^*$. We denote by $\mathbf{X}^{(n)}$ the matrix $\mathbf{X}/\sqrt{n}$, the estimators of the parameters of the Gaussian distribution as defined as follows:

$$\{\widehat{\mathbf{B}}, \widehat{\boldsymbol{\Theta}}\} = \underset{\substack{\mathbf{B}:\mathbf{B}_{jj}=1 \\ \boldsymbol{\Theta} \in \mathbb{R}^{n \times p}}}{\arg\min} \ \underset{\mathbf{c} \in \mathbb{R}^p}{\min} \left\{ \left\| (\mathbf{X}^{(n)}\mathbf{B} - \mathbf{u}_n\mathbf{c}^\top - \boldsymbol{\Theta})^\top \right\|_{2,1} + \lambda\|\boldsymbol{\Theta}\|_{2,1} + \gamma\|\mathbf{B}\|_{1,1} \right\}, \tag{2}$$

where $\lambda \geq 0$ and $\gamma \geq 0$ are tuning parameters respectively promoting robustness and sparsity of the matrix $\mathbf{B}$. The precision matrix $\boldsymbol{\Omega}^*$ can be estimated by

$$\widehat{\omega}_{jj} = \frac{2n}{\pi}\|(\mathbf{I}_n - \mathbf{u}_n\mathbf{u}_n^\top)(\mathbf{X}^{(n)}\widehat{\mathbf{B}}_{\bullet,j} - \widehat{\boldsymbol{\Theta}}_{\bullet,j})\|_1^{-2}; \qquad \widehat{\boldsymbol{\Omega}} = \widehat{\mathbf{B}} \cdot \mathrm{diag}(\{\widehat{\omega}_{jj}\}_{j \in [p]}). \tag{3}$$

We highlight that the estimator of the diagonal entries of the precision matrix stated above – based on average absolute deviation around the mean – is only one of the alternatives proposed in the **DESP** package. The other possibilities rest on residual variance or likelihood maximization (relaxed, symmetry-enforced or penalized). The expectation vector $\boldsymbol{\mu}^*$ can be estimated by

$$\widehat{\boldsymbol{\mu}} = \frac{1}{n}(\mathbf{X} - \widehat{\mathbf{E}})^\top \mathbf{1}_n, \quad \text{where} \quad \widehat{\mathbf{E}} = \sqrt{n}\,\widehat{\boldsymbol{\Theta}}\widehat{\mathbf{B}}^\dagger. \tag{4}$$

The solutions of the problem (2) are obtained iteratively, by optimizing separately with respect to $\mathbf{B}$ and $\boldsymbol{\Theta}$. The details of the algorithm are provided in [3].

2 Implementation

The convex optimization problem (2) is decomposed in p independent sub-problems. When both tuning parameters λ and γ are zero, the solution of problem (2) is obtained using ordinary least squares to estimate each column of $\mathbf{B}$. When $\gamma \neq 0$, each of these p problems corresponding to square-root Lasso can be either cast as an second-order cone program (SOCP), or solved using the coordinate descent algorithm. In the first case, we propose to use the splitting conic solver (SCS, [7]) that solves efficiently convex cone problems. We note however that a more efficient solution in terms of computational time is obtained using the coordinate descend algorithm (stochastic or not). The p square-root Lasso problems can be solved in parallel when the OpenMP application programming interface (API)[2] is supported. Most of the linear algebra operations are performed calling BLAS and LAPACK routines[3]. For the details of available options of the function `desp()`, we refer the user to the package reference manual.

3 Installation

As available on CRAN[4], this package can be simply installed by entering the following instruction:

```
install.packages("DESP")
```

We recommend to use a compiler that supports OpenMP to allow multithreading.

4 Example

We estimate the parameters of the distribution of Fisher's iris data [4] for each of three iris species, assuming that these data are normally distributed.
We first load the package and the data set:

```
library(DESP)
data(iris3)
```

---

[2] OpenMP Architecture Review Board, see http://openmp.org.
[3] These API are for instance implemented by OpenBLAS, available at http://www.openblas.net/; or ATLAS, available at http://math-atlas.sourceforge.net/.
[4] The Comprehensive R Archive Network, https://cran.r-project.org/.

We will use the function `desp.cv()` that relies on the function `desp()` to estimate $\widehat{\boldsymbol{\Omega}}$ and $\widehat{\boldsymbol{\mu}}$, choosing the tuning parameters λ and γ by v-fold cross-validation [5]. To define this function, we have introduced the following partition of the sample $S = \bigcup_{i=1}^{v} S_i$. The values of these parameters are selected over a grid such that the risk (the expectation of the loss) is the lowest. In connexion with the regression model, we might consider a quadratic loss function and select:

$$\{\lambda_{\text{vc}}, \gamma_{\text{vc}}\} = \arg\min_{\lambda,\gamma} \frac{1}{v} \sum_{i=1}^{v} \frac{1}{|S_i|} \sum_{k=1}^{|S_i|} \|\mathbf{X}_{k,\bullet}\widehat{\mathbf{B}}_{(i,\lambda,\gamma)} - \widehat{\boldsymbol{\mu}}_{(i,\lambda,\gamma)}^{\top}\widehat{\mathbf{B}}_{(i,\lambda,\gamma)}\|_2^2$$

where $\widehat{\boldsymbol{\mu}}_{(i,\lambda,\gamma)}$ and $\widehat{\mathbf{B}}_{(i,\lambda,\gamma)}$ are the estimates obtained on the training set $S\backslash S_i$. As the chosen loss function is not robust, we used instead

$$\{\lambda_{\text{vc}}, \gamma_{\text{vc}}\} = \arg\min_{\lambda,\gamma} \frac{1}{v} \sum_{i=1}^{v} \frac{1}{|S_i|} \sum_{k=1}^{|S_i|} \|\mathbf{X}_{k,\bullet}\widehat{\mathbf{B}}_{(i,\lambda,\gamma)} - \widehat{\boldsymbol{\mu}}_{(i,\lambda,\gamma)}^{\top}\widehat{\mathbf{B}}_{(i,\lambda,\gamma)}\|_2$$

that is inspired by the ℓ_1 cross-validation procedure [6].

We choose the estimator based on average absolute deviation around the mean (3) to estimate the diagonal entries. Besides, we choose a number of folds equal to 5.

```
settings <- list(diagElem='AD')
v <- 5
```

Then, we call this function on the first 25 observations of each species of iris:

```
set.seed(1)
categories <- colnames(iris3[1,,])
params <- vector(mode="list", length=length(categories))
for(c in 1:length(categories)){
  obs <- 1:25
  lr <- (9/10)^(0:9)
  gr <- (1/sqrt(2))^(0:4) * sqrt(2*log(ncol(iris3[,,c])))
  params[[c]] <- desp.cv(iris3[obs,,c], v=v, lambda.range=lr,
    gamma.range=gr, settings=settings)
}
```

The estimated parameters corresponding to the species *Iris setosa* are:

```
params[[1]]

## $Omega
##           [,1]      [,2]     [,3]     [,4]
## [1,]  17.12675 -11.32521  0.00000  0.00000
## [2,] -11.32521  20.66644  0.00000  0.00000
## [3,]   0.00000   0.00000 30.03014  0.00000
## [4,]   0.00000   0.00000  0.00000 87.20826
##
## $mu
##           [,1]
## Sepal L. 5.028
## Sepal W. 3.480
## Petal L. 1.460
## Petal W. 0.248
##
## $Theta
##      [,1] [,2] [,3] [,4]
## [1,]    0    0    0    0
## [2,]    0    0    0    0
## [3,]    0    0    0    0
```

```
##  [4,]     0    0    0    0
##  [5,]     0    0    0    0
##  [6,]     0    0    0    0
##  [7,]     0    0    0    0
##  [8,]     0    0    0    0
##  [9,]     0    0    0    0
## [10,]     0    0    0    0
## [11,]     0    0    0    0
## [12,]     0    0    0    0
## [13,]     0    0    0    0
## [14,]     0    0    0    0
## [15,]     0    0    0    0
## [16,]     0    0    0    0
## [17,]     0    0    0    0
## [18,]     0    0    0    0
## [19,]     0    0    0    0
## [20,]     0    0    0    0
## [21,]     0    0    0    0
## [22,]     0    0    0    0
## [23,]     0    0    0    0
## [24,]     0    0    0    0
## [25,]     0    0    0    0
##
## $lambda
## [1] 1
##
## $gamma
## [1] 0.588705
##
## attr(,"class")
## [1] "desp.cv"
```

5   Acknowledgements

The present document is written in LaTeX and uses the R package **knitr**.

References

[1] Balmand, S. and Dalalyan, A. S. (2015a). On estimation of the diagonal elements of a sparse precision matrix. *ArXiv e-prints Posted as http://arxiv.org/abs/1504.04696*, 1–22.

[2] Balmand, S. and Dalalyan, A. S. (2015b). Convex programming approach to robust estimation of a multivariate Gaussian model. *ArXiv e-prints Posted as http://arxiv.org/abs/1512.04734*, 1–31.

[3] Balmand, S. (2016). Large precision matrix estimation. *Phd Thesis*.

[4] Anderson, E. (1935). The irises of the Gaspe Peninsula. *Bulletin of the American Iris Society*, **59**, 2–5.

[5] Geisser, S. (1975). The Predictive Sample Reuse Method with Applications. *Journal of the American Statistical Association*, **70**(350), 320–328.

[6] Wang, F. and Scott, D. (1994). The ℓ_1 method for robust nonparametric regression. *Journal of the American Statistical Association*, **89**(425), 65–76.

[7] O'Donoghue, B. and Chu, E. and Parikh, N. and Boyd, S. (2013). Operator splitting for conic optimization via homogeneous self-dual embedding. *ArXiv e-prints Posted as http://arxiv.org/abs/1312.3039*, 1–30.