

## Topic 15: Sparse Orthogonal Factor Regression

by Sai Zhang

**Key points:** Sparsity and dimensionality reduction for Multivariate Linear Regression models.

**Disclaimer:** The note is built on Prof. *Jinchi Lv*'s lectures of the course at USC, DSO 607, High-Dimensional Statistics and Big Data Problems.

### 15.1 Motivation

Consider a Multivariate Linear Regression (MLR) model

$$\mathbf{Y} = \mathbf{X} \cdot \mathbf{C} + \mathbf{E}$$

$n \times q \quad n \times p \quad p \times q \quad n \times q$

How to apply regularization methods to this model? There are several approaches to consider

- **Shrinkage**: ridge regression to overcome multicollinearity
- **sparsity**: variable selection in multivariate setting
- **Reduced-rank**
  - **Dimension reduction** via reducing rank of  $\mathbf{C}$
  - $\min \|\mathbf{Y} - \mathbf{XC}\|_F^2$  s.t.  $\text{rank}(\mathbf{C}) \leq r$
- **Combinations**
- **Low-rank** plus **sparse decomposition**: robust PCA, latent variable graphical models, covariance estimation
- **Regularized matrix** or **tensor regression**

Or, we can introduce a very attractive sparsity structure to achieve simultaneous dimension reduction and variable selection. This structure should be characterized by

- Having a few **distinct** channels/pathways relating responses and predictors
- Each of such associations may involve only a **smaller subset**, but not all of the responses and predictors

that is

$$\begin{aligned} \mathbf{Y} &= \mathbf{XC} + \mathbf{E} \\ &= \mathbf{X} \cdot \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1q} \\ c_{21} & c_{22} & \cdots & c_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ c_{p1} & c_{p2} & \cdots & c_{pq} \end{pmatrix} + \mathbf{E} \\ &= \mathbf{X} \cdot \begin{pmatrix} 0 & u_{12} & \cdots & u_{1r} \\ u_{21} & 0 & \cdots & c_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ u_{p1} & u_{p2} & \cdots & u_{pr} \end{pmatrix} \cdot \begin{pmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & \cdots & v_{q1} \\ v_{12} & v_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ v_{1r} & v_{2r} & \cdots & v_{qr} \end{pmatrix} + \mathbf{E} \end{aligned}$$

This way, we can have

- **Sparsity**: selection of both latent and original variables
- **Low-rank SVD**: different subsets of responses allowed to be associated with different subsets of predictors

Consider an example:

**Example 15.1.1: Dimension Reduction and Variable Selection via Sparse SVD**

Consider the case where  $p = 1000, q = 100$ , then  $C$ , as a  $p \times q$  matrix, contains 100000 coefficients. Meanwhile, for a rank-3 SVD model:

$$C = d_1 \mathbf{u}_1 \mathbf{v}_1' + d_2 \mathbf{u}_2 \mathbf{v}_2' + d_3 \mathbf{u}_3 \mathbf{v}_3'$$

where  $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$  are all  $p \times 1$ ,  $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$  are all  $q \times 1$ ,  $d_1, d_2, d_3$  are all scalars. Hence, there are only  $3 \times (1000 + 100 + 1) = 3303$  parameters to estimate. If further assume sparsity, the dimension would be even lower.

Now let's develop a scalable procedure for this idea.

## 15.2 Sparse Orthogonal Factor Regression

Consider the singular value decomposition of  $C$

$$C = \mathbf{U} \mathbf{D} \mathbf{V}' = \sum_{k=1}^r d_k \mathbf{u}_k \mathbf{v}_k'$$

where  $\mathbf{U}$  and  $\mathbf{V}$  are both **orthonormal**:  $\mathbf{U}\mathbf{U}' = \mathbf{V}\mathbf{V}' = \mathbf{I}$ . Then we can achieve dimension reduction via **low-dimensional latent model**

$$\tilde{\mathbf{Y}} = \tilde{\mathbf{X}} \mathbf{D} + \tilde{\mathbf{E}}$$

where

- $\tilde{\mathbf{Y}} = \mathbf{Y}\mathbf{V}$ :  $\mathbf{V}$  sparsity leads to response variable selection
- $\tilde{\mathbf{X}} = \mathbf{X}\mathbf{U}$ :  $\mathbf{U}$  sparsity leads to predictor variable selection

How consider

$$(\hat{\mathbf{D}}, \hat{\mathbf{U}}, \hat{\mathbf{V}}) = \arg \min_{\mathbf{D}, \mathbf{U}, \mathbf{V}} \left\{ \frac{1}{2} \|\mathbf{Y} - \mathbf{X} \mathbf{U} \mathbf{D} \mathbf{V}'\|_F^2 + \lambda_d \|\mathbf{D}\|_1 + \lambda_a \rho_a(\mathbf{U} \mathbf{D}) + \lambda_b \rho_b(\mathbf{V} \mathbf{D}) \right\} \quad \text{s.t. } \mathbf{U}' \mathbf{U} = \mathbf{V}' \mathbf{V} = \mathbf{I}_m \quad (15.1)$$

where

- $\rho_a(\cdot), \rho_b(\cdot)$  are penalty functions with regularization parameters  $\lambda_d, \lambda_a, \lambda_b \geq 0$ . These sparsity penalizations on  $\mathbf{U} \mathbf{D}$  and  $\mathbf{V} \mathbf{D}$  can be thought as **importance weighting**
- $\|\cdot\|_F$  is the nuclear norm, defined as the **sum** of its singular values  $\|\mathbf{A}\|_F = \sum_i \sigma_i(\mathbf{A})$ . It encourages sparsity among singular values and achieve **rank reduction**
- The orthogonality on  $\mathbf{U}, \mathbf{V}$  allow a flexible form of sparsity-inducing penalties

If we further enrich this model by introducing an **adaptive weighting  $\mathbf{W}$  matrices**

$$(\hat{\mathbf{\Theta}}, \hat{\mathbf{\Omega}}) = \arg \min_{\mathbf{\Theta}, \mathbf{\Omega}} \left\{ \frac{1}{2} \|\mathbf{Y} - \mathbf{X} \mathbf{U} \mathbf{D} \mathbf{V}'\|_F^2 + \lambda_d \|\mathbf{W}_d \circ \mathbf{D}\|_1 + \lambda_a \rho_a(\mathbf{W}_a \circ \mathbf{A}) + \lambda_b \rho_b(\mathbf{W}_b \circ \mathbf{B}) \right\}$$

s.t.  $\mathbf{U}'\mathbf{U} = \mathbf{V}'\mathbf{V} = \mathbf{I}_m$ ,  $\mathbf{UD} = \mathbf{A}$ ,  $\mathbf{VD} = \mathbf{B}$ . But why? Singular values and singular vectors of **larger magnitude** should be **less penalized** to reduce bias and improve efficiency.

Two applications are

- Biclustering with sparse SVD

$$(\hat{\mathbf{D}}, \hat{\mathbf{U}}, \hat{\mathbf{V}}) = \arg \min_{\mathbf{D}, \mathbf{U}, \mathbf{V}} \left\{ \frac{1}{2} \|\mathbf{X} - \mathbf{UDV}'\|_F^2 + \lambda_d \|\mathbf{D}\|_1 + \lambda_a \rho_a(\mathbf{UD}) + \lambda_b \rho_b(\mathbf{VD}) \right\} \quad \text{s.t. } \mathbf{U}'\mathbf{U} = \mathbf{V}'\mathbf{V} = \mathbf{I}_m$$

- Sparse PCA (sparsity in loadings of principal components)

$$(\hat{\mathbf{A}}, \hat{\mathbf{V}}) = \arg \min_{\mathbf{A}, \mathbf{V}} \left\{ \frac{1}{2} \|\mathbf{X} - \mathbf{XAV}'\|_F^2 + \lambda_a \rho_a(\mathbf{A}) \right\} \quad \text{s.t. } \mathbf{V}'\mathbf{V} = \mathbf{I}_m$$

## 15.3 Nonasymptotic Properties of SOFAR

First, define the robust spark for the regularity conditions

### Definition 15.3.1: The robust spark $\kappa_c$

The robust spark  $\kappa_c$  of the  $n \times p$  design matrix  $\mathbf{X}$  is defined as the smallest possible positive integer such that there exists an  $n \times \kappa_c$  submatrix of  $\frac{1}{\sqrt{n}}\mathbf{X}$  having a **singular value less than** a given positive constant  $c$

The robust spark  $\kappa_c$  here can be at least of order  $O\left(\frac{n}{\log p}\right)$  with large probability for Gaussian design with dependency. With this definition, we characterize the following 5 conditions

- **Parameter space**: True parameters  $(\mathbf{C}^*, \mathbf{D}^*, \mathbf{A}^*, \mathbf{B}^*)$  lie in  $\mathcal{C} \times \mathcal{D} \times \mathcal{A} \times \mathcal{B}$ , where
  - $\mathcal{C} = \{\mathbf{C} \in \mathbb{R}^{p \times q} : \|\mathbf{C}\|_0 < \kappa_{c_2}/2\}$ , with  $\kappa_{c_2}$  being the robust spark of  $\mathbf{X}$
  - $\mathcal{D} = \{\mathbf{D} = \text{diag}\{d_j\} \in \mathbb{R}^{q \times q} : d_j = 0 \text{ or } |d_j| \geq \tau\}$
  - $\mathcal{A} = \{\mathbf{A} = \{a_{ij}\} \in \mathbb{R}^{p \times q} : a_{ij} = 0 \text{ or } |a_{ij}| \geq \tau\}$
  - $\mathcal{B} = \{\mathbf{B} = \{b_{ij}\} \in \mathbb{R}^{p \times q} : b_{ij} = 0 \text{ or } |b_{ij}| \geq \tau\}$
- **Constrained eigenvalue**: It holds that for some constant  $c_3 > 0$

$$\max_{\|\mathbf{u}\|_0 < \frac{\kappa_{c_2}}{2}, \|\mathbf{u}\|_2=1} \|\mathbf{Xu}\|_2^2 \leq c_3 n, \quad \max_{1 \leq j \leq r} \|\mathbf{Xu}_j^*\|_2^2 \leq c_3 n$$

where  $\mathbf{u}_j^*$  is the **left singular vector** of  $\mathbf{C}^*$  corresponding to singular value  $d_j^*$

- **Error term**: The error term  $\mathbf{E} \in \mathbb{R}^{n \times q} \sim \mathcal{N}(\mathbf{0}, \mathbf{I} \otimes \mathbf{\Sigma})$  with the maximum eigenvalue  $\alpha_{\max}$  of  $\mathbf{\Sigma}$  bounded from above and diagonal entries of  $\mathbf{\Sigma}$  being  $\sigma_j^2$
- **Penalty functions**: For matrices  $\mathbf{M}$  and  $\mathbf{M}^*$  of the same size, the penalty functions  $\rho_h$  with  $h \in \{a, b\}$  satisfies

$$|\rho_h(\mathbf{M}) - \rho_h(\mathbf{M}^*)| \leq \|\mathbf{M} - \mathbf{M}^*\|_1$$

- **Relative spectral gap**: The nonzero singular values of  $\mathbf{C}^*$  satisfy that

$$d_{j-1}^{*2} - d_j^{*2} \geq \sqrt{\delta} d_{j-1}^{*2}, \quad 2 \leq j \leq r$$

with a constant  $\delta > 0$ , both  $r$  and  $\sum_{j=1}^r \left(\frac{d_j^*}{d_j^*}\right)^2$  can diverge as  $n \rightarrow \infty$

### How to understand the 5 conditions?

- **Parameter space** and **constrained eigenvalue** are essential for investigating computable solution to non-convex SOFAR optimization problem
- Gaussianity of **error term** can be relaxed
- **Penalty functions** can be many kinds of sparsity-inducing penalties, including entrywise  $L_1$ -norm<sup>1</sup> and row-wise  $(2, 1)$ -norm<sup>2</sup>
- **Relative spectral gap** rules out non-identifiable case where some non-zero singular values are tied with each other and associated singular vectors in matrices  $\mathbf{U}^*, \mathbf{V}^*$  are identifiable only up to some orthogonal transformation

## 15.4 Estimation: Convexity-Assisted Nonconvex Optimization

Non-convexity of SOFAR objective function poses important algorithmic and theoretical challenges, hence consider a **two-step** approach exploiting the framework of convexity-assisted nonconvex optimization (CANO) to obtain SOFAR estimator:

**Step 1** minimize  **$L_1$ -penalized squared loss** for multivariate regression to obtain an initial estimator

### Theorem 15.4.1: Error Bounds for the Initial Estimator

Under some regularity conditions, with large probability the initial estimator satisfies the following error bounds simultaneously:

$$\|\tilde{\mathbf{C}} - \mathbf{C}^*\|_F \leq R_n \equiv c \sqrt{\frac{s \log(pq)}{n}} \quad (\text{A})$$

$$\|\tilde{\mathbf{D}} - \mathbf{D}^*\|_F \leq c \sqrt{\frac{s \log(pq)}{n}} \quad (\text{B})$$

$$\|\tilde{\mathbf{A}} - \mathbf{A}^*\|_F + \|\tilde{\mathbf{B}} - \mathbf{B}^*\|_F \leq c \eta_n \sqrt{\frac{s \log(pq)}{n}} \quad (\text{C})$$

where  $c = \|\mathbf{C}^*\|_0$  and  $\eta_n = 1 + \sqrt{\frac{\sum_{j=1}^r (d_1^*/d_j^*)^2}{\delta}}$

When  $q = 1$ , bound (A) is consistent with the oracle inequality for Lasso. In this step, finer sparse SVD structure of coefficient matrix  $\mathbf{C}^*$  is completely ignored, so intuitively, the second step should be able to improve error bounds.

**Step 2** minimize SOFAR objective function in an **asymptotically shrinking neighborhood** of initial estimator

<sup>1</sup>Entrywise  $L_1$ -norm encourages sparsity among predictor/response effects specific to each rank-1 SVD layer

<sup>2</sup> $(2, 1)$ -norm is defined as the summation of absolute values of all components of a matrix. It promotes predictor/response-wise sparsity **regardless** of specific layer

**Theorem 15.4.2: Nonasymptotic Error Bounds for SOFAR Estimator**

Under some regularity conditions, with large probability the SOFAR estimator satisfies the following error bounds simultaneously:

$$\|\tilde{\mathbf{C}} - \mathbf{C}^*\|_F \leq c \sqrt{\min \{s, (r + s_a + s_b) \eta_n^2\} \cdot \frac{\log(pq)}{n}} \quad (\text{a})$$

$$\|\tilde{\mathbf{D}} - \mathbf{D}^*\|_F + \|\tilde{\mathbf{A}} - \mathbf{A}^*\|_F + \|\tilde{\mathbf{B}} - \mathbf{B}^*\|_F \leq c \eta_n \sqrt{\min \{s, (r + s_a + s_b) \eta_n^2\} \cdot \frac{\log(pq)}{n}} \quad (\text{b})$$

and

$$\|\tilde{\mathbf{D}} - \mathbf{D}^*\|_0 + \|\tilde{\mathbf{A}} - \mathbf{A}^*\|_0 + \|\tilde{\mathbf{B}} - \mathbf{B}^*\|_0 \leq c(r + s_a + s_b) \quad (\text{c})$$

$$\|\tilde{\mathbf{D}} - \mathbf{D}^*\|_1 + \|\tilde{\mathbf{A}} - \mathbf{A}^*\|_1 + \|\tilde{\mathbf{B}} - \mathbf{B}^*\|_1 \leq c(r + s_a + s_b) \eta_n^2 \lambda_{\max} \quad (\text{d})$$

$$\frac{1}{n} \|\mathbf{X}(\hat{\mathbf{C}} - \mathbf{C}^*)\|_F^2 \leq c(r + s_a + s_b) \eta_n^2 \lambda_{\max}^2 \quad (\text{e})$$

where  $r = \|\mathbf{D}^*\|_0$ ,  $s_a = \|\mathbf{A}^*\|_0$ ,  $s_b = \|\mathbf{B}^*\|_0$  and still  $c = \|\mathbf{C}^*\|_0$  and  $\eta_n = 1 + \sqrt{\frac{\sum_{j=1}^r (d_1^*/d_j^*)^2}{\delta}}$

here

- Bound (d) and (e) are the minimum of 2 rates
- $s$  (the sparsity of matrix  $\mathbf{C}^*$ ) comes from the first step of Lasso estimation,  $r + s_a + s_b$  (total sparsity of  $\mathbf{D}^*, \mathbf{A}^*, \mathbf{B}^*$ ) comes from the second step of SOFAR refinement
- Under Frobenius norm,  $s > (r + s_a + s_b) \eta_n^2$  gives that the two-step procedure enhances error rates

also,

- In the case of univariate response with  $q = 1$ ,  $\eta_n = 1 + \delta$ ,  $r = 1$ ,  $s_a = s$ ,  $s_b = 1$ , the upper bounds are then reduced to those for high-dimensional univariate response regressions
- In the case of rank-one  $r = 1$ ,  $\eta_n = 1 + \frac{1}{\sqrt{\delta}}$  and  $s = s_a s_b$ , which leads to

- SOFAR bounds:  $c \sqrt{\frac{(s_a + s_b) \log(pq)}{n}}$ ,  $c \sqrt{\frac{(s_a + s_b) \log(pq)}{n}}$ ,  $c(s_a + s_b)$ ,  $c(s_a + s_b) \sqrt{\frac{\log(pq)}{n}}$  and  $\frac{c(s_a + s_b) \log(pq)}{n}$
- Lasso bound (step 1):  $c \sqrt{\frac{s_a s_b \log(pq)}{n}}$

SOFAR estimator have much improved rates of coverage even in this case.

## 15.5 Implementation with ALM-BCD

Now consider estimation implementation. The idea is to use the **augmented Lagrangian method (ALM)** coupled with **block coordinate descent (BCD)**. The implementation procedure uses variable splitting to separate orthogonality constraints and sparsity-inducing penalties into different subproblems, enabling efficient optimization in a BCD fashion.

Consider  $\Theta = (\mathbf{D}, \mathbf{U}, \mathbf{V})$ ,  $\Omega = (\mathbf{A}, \mathbf{B})$  in the optimization problem

$$\min_{\Theta, \Omega} \left\{ \frac{1}{2} \|\mathbf{Y} - \mathbf{XUDV}'\|_F^2 + \lambda_d \|\mathbf{D}\|_1 + \lambda_a \rho_a(\mathbf{A}) + \lambda_b \rho_b(\mathbf{B}) \right\}$$

$$\text{s.t. } \mathbf{U}'\mathbf{U} = \mathbf{V}'\mathbf{V} = \mathbf{I}_m, \mathbf{UD} = \mathbf{A}, \mathbf{VD} = \mathbf{B}$$