

# Modified SDR for matrix predictors

Chanwoo Lee, May 28, 2020

## 1 Definition of SDR

For a vector predictor  $X \in \mathbb{R}^d$ , sufficient dimension reduction assumes that

$$Y \perp\!\!\!\perp X | \mathbf{B}^T X,$$

where  $\mathbf{B} \in \mathbb{R}^{d \times k}$ . For a matrix predictor  $\mathbf{X} \in \mathbb{R}^{d_1 \times d_2}$ , sufficient dimension reduction assumes that

$$Y \perp\!\!\!\perp \mathbf{X} | \mathbf{X} \times_1 \mathbf{U} \times_2 \mathbf{V}, \quad (1)$$

where  $\mathbf{U} \in \mathbb{R}^{d_1 \times k_1}$ ,  $\mathbf{V} \in \mathbb{R}^{d_2 \times k_2}$ . We can equivalently express (1) as

$$\begin{aligned} Y \perp\!\!\!\perp \mathbf{X} | \{\mathbf{u}_i^T \mathbf{X} \mathbf{v}_j\}_{i \in [k_1], j \in [k_2]}, \text{ or equivalently} \\ Y \perp\!\!\!\perp \mathbf{X} | \{\langle \mathbf{u}_i \mathbf{v}_j^T, \mathbf{X} \rangle\}_{i \in [k_1], j \in [k_2]} \end{aligned} \quad (2)$$

where  $\mathbf{u}_i$  is  $i$ -th column of  $\mathbf{U}$  and  $\mathbf{v}_j$  is  $j$ -th column of  $\mathbf{V}$ . The central subspace in matrix case is defined as

$$S_{Y|\mathbf{X}} = \bigcap_{\{(\mathbf{U}, \mathbf{V}): Y \perp\!\!\!\perp \mathbf{X} | \mathbf{X} \times_1 \mathbf{U} \times_2 \mathbf{V}\}} \text{span}(\mathbf{U}) \times \text{span}(\mathbf{V}),$$

Both lead to equivalent spaces if we set  $k = k_1 * k_2$

**Remark 1.** The assumption (1) is different from the previous note (under structural conditions, they might be the same but different in general). The main difference is the dimension reduction part (1) has  $k_1 k_2$  linear equations but  $k$  in previous notes. There are some reasons I choose the current one. For previous equation, we have to assume all  $\mathbf{B}_i$  has the same rank  $r$  structure, which does not make sense well to me. In addition, we have to choose the rank  $k_1$  and  $k_2$  later from previous definition, which does not have clear connection with the number of equations  $k$ .

not true. We only assume a common “upper bound” for the rank of  $\mathbf{B}$ ’s.

## 2 Estimation of SDR

I’d use rank-r SMM, and choose r by CV.

We use rank 1 SMM method motivated by formula in (2). The weighted SMM finds a matrix  $\mu \mathbf{v}^T$  that optimizes the following problem.

r too large → high variance in the estimated SDR  
r too small → high bias in the estimated hyperplane

For the linear model, the ‘optimal’ r = min(r1, r2).

$$\min_{\mathbf{u} \in \mathbb{R}^{d_1}, \mathbf{v} \in \mathbb{R}^{d_2}} \|\mathbf{u} \mathbf{v}^T\|^2 + \frac{\lambda}{N} \sum_{i=1}^N \omega_\pi(Y_i) (1 - Y_i f(\mathbf{X}_i; \mathbf{u}, \mathbf{v}, \alpha))_+,$$

where  $f(\mathbf{X}_i; \mathbf{u}, \mathbf{v}, \alpha) = \alpha + \langle \mathbf{u} \mathbf{v}^T, \mathbf{X}_i \rangle$ .

By the similar way, we can extend the linear principal weighted vector machine to the matrix case with a pair of random variables  $(\mathbf{X}, Y) \in \mathbb{R}^{d_1 \times d_2} \times \{+1, -1\}$ . We look for optimizer that minimizes

$$\Lambda_\pi(\mathbf{u} \mathbf{v}^T, \alpha) = \text{Vec}(\mathbf{u} \mathbf{v}^T)^T \text{cov}(\text{Vec}(\mathbf{X})) \text{Vec}((\mathbf{u} \mathbf{v}^T) + \lambda \mathbb{E}[\omega_\pi(Y) (1 - Y f(\mathbf{X}; \mathbf{u}, \mathbf{v}, \alpha))_+]) \quad (3)$$

Denote the observed data by  $\{(\mathbf{X}_i, Y_i) : \mathbf{X}_i \in \mathbb{R}^{m \times n}, Y_i \in \{+1, -1\}, i = 1, \dots, N\}$ . The sampled version of  $\Lambda_\pi$  in (3) is,

$$\hat{\Lambda}_{N,\pi} = \text{Vec}(\mathbf{u}\mathbf{v}^T)^T \hat{\Sigma}_{\mathbf{N}} \text{Vec}((\mathbf{u}\mathbf{v}^T) + \frac{\lambda}{N} \sum_{i=1}^N \left[ \omega_\pi(Y_i) (1 - Y_i \hat{f}_N(\mathbf{X}_i; \mathbf{u}, \mathbf{v}, \alpha))_+ \right]), \quad (4)$$

where  $\hat{f}_N(\mathbf{X}_i, \mathbf{u}, \mathbf{v}, \alpha) = \alpha + \langle \mathbf{X}_i - \bar{\mathbf{X}}_N, \mathbf{u}\mathbf{v}^T \rangle$ ,  $\bar{\mathbf{X}}_N$  is the sample mean, and  $\Sigma_{\mathbf{N}}$  denotes the sample covariance matrix of  $\{\text{Vec}(\mathbf{X}_i)\}_{i=1}^N$ . With transformations  $\text{Vec}(\tilde{\mathbf{u}}\tilde{\mathbf{v}}^T) = \hat{\Sigma}_{\mathbf{N}}^{\frac{1}{2}} \text{Vec}(\mathbf{u}\mathbf{v}^T)$  and  $\mathbf{Z}_i = \hat{\Sigma}_{\mathbf{N}}^{-\frac{1}{2}}(\mathbf{X}_i - \bar{\mathbf{X}}_n)$ , (4) becomes

$$\tilde{\Lambda}_{N,\pi} = \|\tilde{\mathbf{u}}\tilde{\mathbf{v}}^T\|^2 + \frac{\lambda}{N} \sum_{i=1}^N \left[ \omega_\pi(Y_i) (1 - Y_i \hat{f}_N(\mathbf{Z}_i; \tilde{\mathbf{u}}, \tilde{\mathbf{v}}, \alpha))_+ \right] \quad (5)$$

Denote the optimizer of (5) as  $(\tilde{\mathbf{u}}_{n,\pi}, \tilde{\mathbf{v}}_{n,\pi})$ , then we can obtain the optimizer of (3) from  $\text{Vec}(\hat{\mathbf{u}}_{n,\pi} \hat{\mathbf{v}}_{n,\pi}^T) = \hat{\Sigma}_{\mathbf{N}}^{-\frac{1}{2}} \text{Vec}(\tilde{\mathbf{u}}_{n,\pi} \tilde{\mathbf{v}}_{n,\pi}^T)$

Given a grid  $0 < \pi_1 < \dots < \pi_H < 1$ , we obtained  $H$ -candidates  $\{\hat{\mathbf{u}}_{n,\pi_h} \hat{\mathbf{v}}_{n,\pi_h}^T\}_{h=1}^H$  for the central subspace.

We can perform principal component analysis to get the  $k_1$  column basis elements and  $k_2$  row basis elements in  $S_{Y|\mathbf{X}}$  with the following procedure.

1. Obtain the candidate tensor  $\hat{\mathcal{M}} \in \mathbb{R}^{H \times d_1 \times d_2}$  such that  $\hat{\mathcal{M}}_{h..} = \hat{\mathbf{u}}_{n,\pi_h} \hat{\mathbf{v}}_{n,\pi_h}^T$
2. From N-th order SVD (Thm 2 in Lathauwer et al. [2000]),

$$\hat{\mathcal{M}} = \hat{\mathcal{C}} \times_1 \hat{\mathbf{U}}_1 \times_2 \hat{\mathbf{U}}_2 \times_3 \hat{\mathbf{U}}_3,$$

where  $\hat{\mathcal{C}} \in \mathbb{R}^{H \times d_1 \times d_2}$ ,  $\hat{\mathbf{U}}_1 \in \mathbb{R}^{H \times H}$ ,  $\hat{\mathbf{U}}_2 \in \mathbb{R}^{d_2 \times d_2}$ , and  $\hat{\mathbf{U}}_3 \in \mathbb{R}^{d_3 \times d_3}$ .

3. Estimate the central subspace as

$$\hat{S}_{Y|\mathbf{X}} = \{[\hat{\mathbf{U}}_2]_i\}_{i=1}^{k_1} \times \{[\hat{\mathbf{U}}_3]_i\}_{i=1}^{k_2}.$$

We can reduce the dimension of  $\mathbf{X}$  as  $\{\mathbf{u}^T \mathbf{X} \mathbf{v} \text{ where } (\mathbf{u}, \mathbf{v}) \in \hat{S}_{Y|\mathbf{X}}\}$

this modification is fine.

**Remark 2.** We consider criterion to estimate  $k_1$  and  $k_2$

$$\begin{aligned} G_{n,1}(m; \rho, \hat{\mathcal{M}}_n) &= \sum_{j=1}^m \|\mathcal{C}_{i_2=j}\| - \rho \frac{m \log n}{\sqrt{n}} \|\mathcal{C}_{i_2=1}\|, \\ G_{n,2}(m; \rho, \hat{\mathcal{M}}_n) &= \sum_{j=1}^m \|\mathcal{C}_{i_3=j}\| - \rho \frac{m \log n}{\sqrt{n}} \|\mathcal{C}_{i_3=1}\|, \end{aligned}$$

where  $\rho$  is a tuning parameter. We have the consistent estimators

$$\begin{aligned} \hat{k}_1 &= \arg \min_m G_{n,1}(m; \rho, \hat{\mathcal{M}}_n), \\ \hat{k}_2 &= \arg \min_m G_{n,2}(m; \rho, \hat{\mathcal{M}}_n). \end{aligned}$$

**Remark 3.** With some structural assumption on  $\mathbf{X}$ , we can simplify (4).

## References

Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Analysis Applications*, 21:1253–1278, 2000.

Q1: how to choose rank in the conditional prob. estimation?  
Proposal: grid search  $r$  using cross-validation.

notations;  
 $r$ : rank bound in SMM  
 $(r_1, r_2)$ : structure dimension in SDR

Training set: perform the sequence of weighted SMM with rank  $r$   
Test set: validate using certain loss function, e.g. Bernoulli deviation (KL) or squared loss

Q2: how to choose  $(r_1, r_2)$  in SDR?  
Proposal: Cross-validation + Tucker

Step 1. choose  $r$  in SMM by CV —> return a coefficient tensor with rank lower bounded by  $(r^2, r, r)$   
Step 2: Choose suitable  $(r_1, r_2)$  from Tucker approximation. Could use your criteria in Remark 2. Note that the  $(r_1, r_2)$  should be at least  $r$

Q3: How does the procedure behave for vector-valued inputs ( $d_1$ -by-1 matrices) with linear classifiers?  
A: consistent with the original paper.  $d_2=1 \rightarrow r_2=1 \rightarrow r=\min(r_1, r_2)=1 \rightarrow$  SVM with rank 1 —> same as your proposal

Q4: how does the procedure behave for matrix-valued inputs but NO dimension reduction?

A:  $(r_1, r_2)=(d_1, d_2) \rightarrow r=\min(r_1, r_2)=\min(d_1, d_2) \rightarrow$  full rank SVM —> same as if we treat matrix  $X$  as  $\text{vec}(X)$ .