SDR for matrix predictors

Chanwoo Lee, May 24, 2020

1 SDR for matrix

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

$$Y \perp\!\!\!\perp X | \boldsymbol{B}^T X, \tag{1}$$

where $\boldsymbol{B} \in \mathbb{R}^{d\times k}$. We define with $\mathcal{X}\bar{\times}_N\mathcal{Y}$ a sequence of contracted products between the $(K+N)$-order tensor $\mathcal{X} \in \mathbb{R}^{J_1\times\cdots\times J_K\times I_1\times\cdots\times I_N}$ and the $(N+M)$-order tensor $\mathcal{Y} \in \mathbb{R}^{I_1\times\cdots\times I_N\times H_1\times\cdots\times H_M}$. Entry-wise, it is defined as

$$(\mathcal{X}\bar{\times}_N\mathcal{Y})_{j_1,\ldots,j_K,h_1,\ldots,h_M} = \sum_{i_1=1}^{I_1}\cdots\sum_{i_N=1}^{I_N}\mathcal{X}_{j_1,\ldots,j_K,i_1,\ldots,i_N}\mathcal{Y}_{i_1,\ldots,i_N,h_1,\ldots,h_M}.$$

For a matrix predictor $\boldsymbol{X} \in \mathbb{R}^{m\times n}$, sufficient dimension reduction assumes that

$$Y \perp\!\!\!\perp \boldsymbol{X} | \mathcal{B}\bar{\times}_2\boldsymbol{X}, \tag{2}$$

where $\mathcal{B} \in \mathbb{R}^{k\times m\times n}$. If we define $\mathcal{B}_{i::} = \boldsymbol{B}_i$, we have

$$\mathcal{B}\bar{\times}_2\boldsymbol{X} = (\langle\boldsymbol{B}_1,\boldsymbol{X}\rangle,\ldots,\langle\boldsymbol{B}_k,\boldsymbol{X}\rangle)^T.$$

Remark 1. The predictor matrix $\boldsymbol{X}$ is a vector where $n=1$, (2) is reduced down to (1)($\boldsymbol{B}\bar{\times}_1 X$). In addition, we can extend to tensor case with order d as

$$Y \perp\!\!\!\perp \mathcal{X} | \mathcal{B}\bar{\times}_d\mathcal{X},$$

where $\mathcal{X} \in \mathbb{R}^{I_1\times\cdots\times I_d}$ and $\mathcal{B} \in \mathbb{R}^{k\times I_1\times\cdots\times I_d}$.

If this holds for B —> this also holds for B\times_1 P, where
P is an orthonormal matrix (i.e. applying rotation to mode 1 of B)

Remark 2. If we do not assume low rank matrix structure on $\boldsymbol{B}_i$, (2) is equivalent to (1) with predictor $\boldsymbol{X}$ replaced by Vec($\boldsymbol{X}$).

Remark 3. My guess of defining the central subspace in matrix case as follows. First, define span of tensor $\mathcal{B}$ as

intuition? Is this Span(B) invariant to orthogonal rotations on mode 1 of B?

$$\text{span}(\mathcal{B}) = \{\boldsymbol{U}\boldsymbol{V}^T : \boldsymbol{U} = \sum_{i=1}^k \alpha_i\boldsymbol{U}_i,\quad \boldsymbol{V} = \sum_{i=1}^k \beta_i\boldsymbol{V}_i \quad \text{where } \boldsymbol{\alpha},\boldsymbol{\beta}\in\mathbb{R}^k \text{ and } \mathcal{B}_{i::} = \boldsymbol{U}_i\boldsymbol{V}_i^T\}$$

From this span of the tensor, the central subspace in matrix case is defined as

$$S_{Y|\boldsymbol{X}} = \bigcap_{\{\mathcal{B}:Y\perp\!\!\!\perp X|\mathcal{B}\bar{\times}_2\boldsymbol{X}\}} \text{span}(\mathcal{B}),$$

How about this modification?

Span(B) = {U \otimes V: U is the column space of B_(2), V is the column space of B_(3)}

We extended weighted SVM to SMM to find the best hyperplane that separate $S_\pi = \{\boldsymbol{X} : \mathbb{P}(\boldsymbol{X}|y=1) > \pi\}$ and $S_{-\pi} = \{\boldsymbol{X} : \mathbb{P}(\boldsymbol{X}|y=1) < \pi\}$ The weighted SMM finds a matrix $\boldsymbol{B} \in \mathbb{R}^{m\times n}$ that optimizes the following problem.

$$\min_{\boldsymbol{B}\in\mathbb{R}^{m\times n}} \|\boldsymbol{B}\|^2 + \frac{\lambda}{N}\sum_{i=1}^N \omega_\pi(Y_i)\left(1 - Y_i f(\boldsymbol{X}_i;\boldsymbol{B},\alpha)\right)_+,$$

where $f(\boldsymbol{X}_i; \boldsymbol{B}, \alpha) = \alpha + \langle \boldsymbol{B}, \boldsymbol{X}_i \rangle$. We make distinction from SVM assuming low rank structure to $\boldsymbol{B} = \boldsymbol{U}\boldsymbol{V}^T$ where $\boldsymbol{U} \in \mathbb{R}^{m\times r}, \boldsymbol{V} \in \mathbb{R}^{n\times r}$.

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

$$\Lambda_\pi(\boldsymbol{B}, \alpha) = \mathrm{Vec}\,(\boldsymbol{B})^T \mathrm{cov}\,(\mathrm{Vec}(\boldsymbol{X}))\,\mathrm{Vec}\,((\boldsymbol{B}) + \lambda \mathbb{E}\left[\omega_\pi(Y)\,(1 - Y f(\boldsymbol{X}; \boldsymbol{B}, \alpha))_+\right] \tag{3}$$

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

$$\hat{\Lambda}_{n,\pi} = \mathrm{Vec}\,(\boldsymbol{B})^T \hat{\boldsymbol{\Sigma}}_{\mathbf{N}}\,\mathrm{Vec}\,((\boldsymbol{B}) + \frac{\lambda}{n}\sum_{i=1}^{N}\left[\omega_\pi(Y_i)\left(1 - Y_i \hat{f}_n(\boldsymbol{X}_i; \boldsymbol{B}, \alpha)\right)_+\right], \tag{4}$$

where $\hat{f}_n(\boldsymbol{X}_i, \boldsymbol{B}, \alpha) = \alpha + \langle \boldsymbol{X}_i - \bar{\boldsymbol{X}}_n, \boldsymbol{B}\rangle$, $\bar{\boldsymbol{X}}_n$ is the sample mean, and $\boldsymbol{\Sigma}_{\mathbf{n}}$ denotes the sample covariance matrix of $\{\mathrm{Vec}(\boldsymbol{X}_i)\}_{i=1}^N$. With transformations $\mathrm{Vec}(\boldsymbol{D}) = \hat{\boldsymbol{\Sigma}}_{\mathbf{N}}^{\frac{1}{2}}\boldsymbol{B}$ and $\boldsymbol{Z}_i = \hat{\boldsymbol{\Sigma}}_{\mathbf{N}}^{-\frac{1}{2}}(\boldsymbol{X}_i - \bar{\boldsymbol{X}}_n)$, (4) becomes

$$\hat{\Lambda}'_{n,\pi} = \|\boldsymbol{D}\|^2 + \frac{\lambda}{n}\sum_{i=1}^{N}\left[\omega_\pi(Y_i)\left(1 - Y_i \hat{f}_n(\boldsymbol{Z}_i; \boldsymbol{D}, \alpha)\right)_+\right] \tag{5}$$

Denote the optimizer of (5) as $\hat{\boldsymbol{D}}_{n,\pi}$, then the optimizer of (3) is $\hat{\boldsymbol{B}}_{n,\pi} = \hat{\boldsymbol{\Sigma}}_{\mathbf{N}}^{-\frac{1}{2}}\hat{\boldsymbol{D}}_{n,\pi}$

Remark 4. If we assumes $\boldsymbol{B}$ as full rank, then all the procedures are reduced down to the linear principal weighted vector machine with sample $\{\mathrm{Vec}(\boldsymbol{X}_i)\}_{i=1}^N$

Remark 5. Since the transformation $\mathrm{Vec}(\boldsymbol{D}) = \hat{\boldsymbol{\Sigma}}_{\mathbf{N}}^{\frac{1}{2}}\boldsymbol{B}$ does not change the rank. we can assume the low rank structure as $\boldsymbol{D} = \boldsymbol{U}\boldsymbol{V}^T$ and solve the weighted SMM problem.

Given a grid $0 < \pi_1 < \cdots < \pi_H < 1$, we obtained H-candidates $\{\hat{\boldsymbol{B}}_{n,\pi_h}\}_{h=1}^H$ of the central subspace. We can perform principal component analysis to get the k basis elements of $S_{Y|\boldsymbol{X}}$ with the following procedure.

1. Obtain column part matrices $\{\hat{\boldsymbol{U}}_h\}_{h=1}^H$ and row part matrices $\{\hat{\boldsymbol{V}}_h\}_{h=1}^H$ through SVD such that

$$\hat{\boldsymbol{B}}_{n,\pi_h} = \hat{\boldsymbol{U}}_h \hat{\boldsymbol{\Sigma}}_h \hat{\boldsymbol{V}}_h^T \quad h = 1, \ldots, H.$$

you could simplify your algorithm using your tensor B

2. Calculate row-candidate matrix $\hat{\boldsymbol{M}}_n^r$ and column-candidate matrix $\hat{\boldsymbol{M}}_n^c$ as

$$\hat{\boldsymbol{M}}_n^r = \sum_{h=1}^H \mathrm{Vec}\,(\hat{\boldsymbol{V}}_h)\,\mathrm{Vec}\,(\hat{\boldsymbol{V}}_h)^T$$

construct tensor B by B(h,..)=matrix B, h=1,…H

$$\hat{\boldsymbol{M}}_n^c = \sum_{h=1}^H \mathrm{Vec}\,(\hat{\boldsymbol{U}}_h)\,\mathrm{Vec}\,(\hat{\boldsymbol{U}}_h)^T.$$

Estimated U: column space of B_(2)
Estimated V: column space of B_(3)

3. The first k eigenmatrices (folded from eigenvectors) of $\hat{\boldsymbol{M}}_n^r$, denoted by $\{\tilde{\boldsymbol{V}}_1, \ldots, \tilde{\boldsymbol{V}}_k\}$, estimates the row-part basis. By the same way, The first k eigenmatrices $\{\tilde{\boldsymbol{U}}_1, \ldots, \tilde{\boldsymbol{U}}_k\}$ of $\hat{\boldsymbol{M}}_n^c$ estimates the column-part basis

4. Estimate the central subspace as

$$S_{Y|\boldsymbol{X}} = \left\{ \left(\sum_{i=1}^{k} \alpha_i \tilde{\boldsymbol{U}}_i \right) \left(\sum_{i=1}^{k} \beta_i \tilde{\boldsymbol{V}}_i \right)^T : \boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^k \right\}.$$

Remark 6. These principal component procedures can be reduced down to the vector case if we standardize the estimated normal vectors as $\{\boldsymbol{\beta}_h / \|\boldsymbol{\beta}_h\|\}_{h=1}^{H}$.

2 Generating matrix valued training data for SDR

We can consider simple model that can show matrix valued SDR performance. First, generate matrix valued $\{\boldsymbol{X}_i\}_{i=1}^{N} \in \mathbb{R}^{m \times n}$ whose entries are from i.i.d. $N(0,1)$. Next, we generate $\mathcal{B} \in \mathbb{R}^{2 \times m \times n}$ such that

$$\mathcal{B}_{1\cdot\cdot} = \boldsymbol{u}_1 \boldsymbol{v}_1^T, \quad \mathcal{B}_{2\cdot\cdot} = \boldsymbol{u}_2 \boldsymbol{v}_2^T,$$

where $\boldsymbol{u}_i \in \mathbb{R}^{m \times r}$ and $\boldsymbol{v}_i \in \mathbb{R}^{n \times r}, i = 1, 2$. Denote $\boldsymbol{Z}_{1i} = \langle \mathcal{B}_{1\cdot\cdot}, \boldsymbol{X}_i \rangle$ and $\boldsymbol{Z}_{2i} = \langle \mathcal{B}_{2\cdot\cdot}, \boldsymbol{X}_i \rangle$. We assign the label $Y_i \in \{+1, -1\}$ as

$$Y_i = \text{sign}(2\boldsymbol{Z}_{1i} + \boldsymbol{Z}_{2i} + 0.2\epsilon) \qquad \text{where } \epsilon \sim N(0,1).$$

In this way, we can generate the training data $\{(\boldsymbol{X}_i, Y_i)\}_{i=1}^{N}$ and check whether estimated the central subspace is close to true one.

If we set the rule of labeling Y_i as

sounds reasonable.

$$Y_i = \text{sign}(\boldsymbol{Z}_{1i}^2 + \boldsymbol{Z}_{2i}^2 - 1)$$

We can check weather the kernel method works well with good visualization which we considered in the last meeting.