Thoughts on space estimation via averaging

Miaoyan Wang, 11/9/2019; updated on 11/15/2019.

- Q1: Is algorithm 4 the same as oversampling approach?
- Q2: How to define "averaged space" from multiple space estimators?

Claim 1: Algorithm 4 is different from Algorithm 6 (oversampling).

Example: Consider a 2-by-2 data matrix

$$\boldsymbol{M} = \boldsymbol{e} \otimes \boldsymbol{e} + \delta \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \delta \end{bmatrix}$$

where $\boldsymbol{e} = (1, 0)^T$ is the signal and $\delta \ll 1$ is the noise. The goal is to estimate $\boldsymbol{e}$ via $\boldsymbol{M}$.

Algorithm 4: Let $\boldsymbol{Z} = [\![z_{ij}]\!] \in \mathbb{R}^{2\times 5}$ denote a Gaussian test matrix. Let $\boldsymbol{M}\boldsymbol{z}_i = (z_{1i}, \delta z_{2i})^T$ be the i-th projection, and $\hat{\boldsymbol{e}}_i = \frac{\boldsymbol{M}\boldsymbol{z}_i}{\|\boldsymbol{M}\boldsymbol{z}_i\|_2}$ the i-th estimator of $\boldsymbol{e}$, where $i = 1, \ldots, 5$. Taking "angle-wise" average of $\{\hat{\boldsymbol{e}}_i\}_{i\in[5]}$ yields the estimator $\hat{\boldsymbol{e}}^*_{\text{normalize}}$:

$$\begin{aligned}
\hat{\boldsymbol{e}}^*_{\text{normalize}} &= \text{leading singularvector of the matrix } \begin{bmatrix} \frac{\boldsymbol{M}\boldsymbol{z}_1}{\|\boldsymbol{M}\boldsymbol{z}_1\|_2} & \cdots & \frac{\boldsymbol{M}\boldsymbol{z}_5}{\|\boldsymbol{M}\boldsymbol{z}_5\|_2} \end{bmatrix} \\
&= \text{leading singularvector of the matrix } \begin{bmatrix} \frac{1}{\sqrt{z_{11}^2+\delta^2 z_{21}^2}} z_{11} & \frac{1}{\sqrt{z_{12}^2+\delta^2 z_{22}^2}} z_{12} & \cdots & \frac{1}{\sqrt{z_{15}^2+\delta^2 z_{25}^2}} z_{15} \\ \frac{1}{\sqrt{z_{11}^2+\delta^2 z_{21}^2}} \delta^2 z_{21} & \frac{1}{\sqrt{z_{12}^2+\delta^2 z_{22}^2}} \delta^2 z_{22} & \cdots & \frac{1}{\sqrt{z_{15}^2+\delta^2 z_{25}^2}} \delta^2 z_{25} \end{bmatrix}.
\end{aligned}$$

Algorithm 6: Let $\boldsymbol{Z} = [\![z_{ij}]\!] \in \mathbb{R}^{2\times 5}$ be a Gaussian test matrix. The oversampling approach takes the leading left singular vector of $\boldsymbol{MZ}$ as the estimator $\boldsymbol{e}^*_{\text{unnormalize}}$; i.e.

$$\begin{aligned}
\hat{\boldsymbol{e}}^*_{\text{unnormalize}} &= \text{leading singular vector of the matrix } \boldsymbol{MZ} \\
&= \text{leading singular vector of the matrix } \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{15} \\ \delta z_{21} & \delta z_{22} & \cdots & \delta z_{25} \end{bmatrix}, \\
&\neq \hat{\boldsymbol{e}}^*_{\text{normalize}} \text{ in general.}
\end{aligned}$$

Claim 2: Entry-wise average brings no additional information to the space estimator.

Algorithm 5: The estimator $\hat{\boldsymbol{e}}^*$ is defined as the entrywise average of $\boldsymbol{M}\boldsymbol{z}_i$ for $i = 1, \ldots, 5$:

$$\hat{\boldsymbol{e}}^* \propto \boldsymbol{M}\boldsymbol{z}_1 + \cdots \boldsymbol{M}\boldsymbol{z}_5 \propto \boldsymbol{M}\boldsymbol{z}^*,$$

where $\boldsymbol{z}^* = \boldsymbol{z}_1 + \cdots + \boldsymbol{z}_5 \in \mathbb{R}^{2\times 1}$ is simply another Gaussian vector. In other words, Algorithm 5 is stochastically equivalent to the naive single projection method.

Summary:

1. Algorithms 4 and 6 are different. Algorithm 4 imposes equal weights on the replicates, whereas Algorithm 6 imposes stochastic weights on the replicates. I have not investigated into the theoretical comparison between these two methods.

2. The entry-wise average makes little sense to me. Intuitively, one should take angle-wise average to define the "average of spaces". Figure 1 shows the angle-wise average in the 1-dimensional case. My conjecture is that the angle-wise average is equivalent to the leading singular vectors of the concatenated singular spaces. That is the reason I use leading singular vectors in Algorithm 4.

3. The average-based approach reduces the variance in the final estimator, thus improving the accuracy. However, decomposing a concatenated matrix of d-by $5k$ incurs additional computational cost. Perhaps we should think of alternative, cheaper ways to find the angle-wise average between spaces. Geometric interoperation may be useful here.

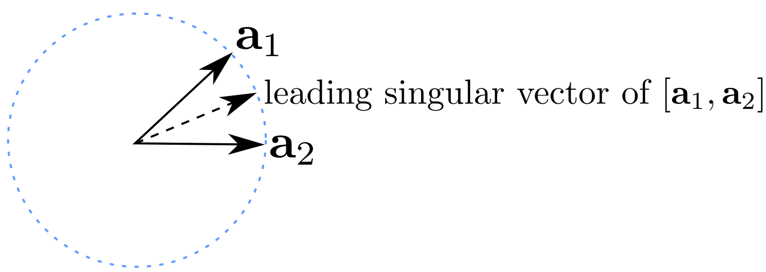

Figure 1: Demonstration of angle-wise average.

Added on Nov 15 2019:

The above sentences in red shall be made more rigorous. We show that the concatenation approach indeed computes the "average" under some natural measurement. As a by-product, we provide another equivalent way to compute the averaged space.

Let $\boldsymbol{V}_1, \boldsymbol{V}_2 \in \mathbb{R}^{d\times r}$ be two matrices with orthogonal columns. Then, the angle-wise average of $\text{Span}(\boldsymbol{V}_1)$ and $\text{Span}(\boldsymbol{V}_2)$ can be obtained in either of the following two ways:

- Approach 1: Define a concatenated matrix $\boldsymbol{V} = [\boldsymbol{V}_1, \boldsymbol{V}_2] \in \mathbb{R}^{d\times 2r}$. Let $\boldsymbol{V}^*_{\text{conc.}} \in \mathbb{R}^{d\times r}$ denote the leading r left singular-vectors of the concatenated matrix $\boldsymbol{V}$.

- Approach 2: Define two gram matrices $\boldsymbol{M}_1 = \boldsymbol{V}_1\boldsymbol{V}_1^T$, $\boldsymbol{M}_2 = \boldsymbol{V}_2\boldsymbol{V}_2^T$ and their entrywise average
$$\boldsymbol{M} = \frac{1}{2}(\boldsymbol{M}_1 + \boldsymbol{M}_2) \in \mathbb{R}^{d\times d}.$$
Let $\boldsymbol{V}^*_{\text{ave.}} \in \mathbb{R}^{d\times r}$ denote the leading r eigen-vectors of the averaged matrix $\boldsymbol{M}$.

We conclude that $\boldsymbol{V}^*_{\text{conc.}} = \boldsymbol{V}^*_{\text{ave.}}$. To see this, note that the left singular spaces of $\boldsymbol{V}$ is the eigenspace of $\boldsymbol{V}\boldsymbol{V}^T$, where

$$\boldsymbol{V}\boldsymbol{V}^T = \begin{bmatrix}\boldsymbol{V}_1, \boldsymbol{V}_2\end{bmatrix} \begin{bmatrix}\boldsymbol{V}_1^T \\ \boldsymbol{V}_2^T\end{bmatrix} = \boldsymbol{V}_1\boldsymbol{V}_1^T + \boldsymbol{V}_2\boldsymbol{V}_2^T = 2\boldsymbol{M}.$$

Therefore $\mathrm{Span}(\boldsymbol{V}^*_{\text{conc.}}) = \mathrm{Span}(\boldsymbol{V}^*_{\text{ave.}})$.

Remark 1. The interpretation of Approach 2 justifies the notion of "average" in our context. Indeed, the average is performed on the Gram matrices. Similar argument applies to computing the average of m spaces. We comment that, in the case $d \gg mr$, Approach 1 requires less flops and lower memory than Approach 2. The complexity is $\mathcal{O}(dmr \min(d, mr))$ for Approach 1, and $\mathcal{O}(d^2(r + m + d))$ for Approach 2.