Supplement to "sJIVE: Supervised Joint and Individual Variation Explained"

Palzer EF[1], Wendt C[2], Bowler R[3], Hersh CP[4], Safo SE[1] and Lock EF[1]

[1] Division of Biostatistics, University of Minnesota, Minneapolis, 55455, USA,
[2] Division of Pulmonary, Allergy and Critical Care, University of Minnesota, Minneapolis, 55455, USA and
[3] Division of Pulmonary, Critical Care and Sleep Medicine, Department of Medicine, National Jewish Health, Denver, CO, USA and
[4] Channing Division of Network Medicine, Brigham and Women's Hospital, Harvard Medical School, Boston, MA.

Abstract

This supplementary document provides further details and validation of the sJIVE method. In Section 1, we confirm the uniqueness of the solution under orthogonality. In Section 2, we give additional details on the cross-validation approach for rank selection. In Section 3, we give additional details on how the simulation data was generated. In Section 4, we describe how we reduce dimensionality to increase computational efficiency.

1 Proof of Theorem 1

Here, we provide a proof for Theorem 1 of the main article. It follows from Lemma 1 of Feng et al. (2018) [1], which is analogous to Theorem 1.1 of Lock et al. (2013) [2], that a decomposition satisfying conditions 1., 2., and 3. of Theorem 1 exists and is unique for $\{\mathbf{X}_1, \ldots, \mathbf{X}_k\}$. Further, because $\tilde{\mathbf{y}} \in \text{row}(\tilde{\mathbf{X}}) \subset (\text{row}(J) + \text{row}(A))$ it follows that $\mathbf{y} = \mathbf{j}_y + \mathbf{a}_y$ where $\mathbf{j}_y \in \text{row}(J)$ and $\mathbf{a}_y \in \text{row}(\mathbf{A})$ (condition 4.), and because $\text{row}(J) \perp \text{row}(A)$ $\mathbf{j}_y$ and $\mathbf{a}_y$ are uniquely defined.

2 Cross-Validation Rank Selection Algorithm

sJIVE selects ranks using a forward selection 5-fold CV algorithm. In the pseudocode below, each time the algorithm instructs to run 5-fold CV, the data was split into 5 folds and sJIVE was fit to 4 of them. The fitted model was then used to predict the outcome for the left-out fold, and the MSE is recorded by comparing the estimated and true $\mathbf{y}$ values. This is repeated for each of the 5 folds.

Pseudocode for sJIVE rank selection

1. Initialize r_J and r_i $\forall i = 1, \ldots, k$ to 0
2. Run 5-fold CV. Record the average test MSE of each fold and label it MSE_{best}
3. Let $r_J = r_J + 1$ and run 5-fold CV, recording the average test MSE of each fold
4. Return r_J back to $r_J - 1$
5. For $i = 1, \ldots, K$
   (a) Let $r_i = r_i + 1$ and run 5-fold CV, recording the average test MSE of each fold
   (b) Return r_i back to $r_i - 1$
6. Determine which rank increase led to the largest reduction of test MSE. Permanently increase that rank by 1, and set MSE_{best} to its new lower value.
7. Repeat steps 3-6 until all rank increases lead to higher MSE_{best} values

3 Generating Simulated Datasets

In this section, we will describe how the datasets were simulated. Our function for generating data allowed us to input the following: the number of datasets, k; the number of predictors in each dataset, $p = (p_1, \cdots, p_k)$; the number of observations, n; the joint and individual ranks, r_J and $r_A = (r_{A_1}, \cdots, r_{A_k})$; the weight of the joint and individual signals, w_J and w_A; the proportion of variance in $\mathbf{X}_i$, $i = 1, \cdots, k$ that contributes to error, X_{err}; the proportion of $\mathbf{y}$ variance contributed to error, Y_{err}; and the proportion of the ranks that are predictive of $\mathbf{y}$, r_{prop}.

Define the following:

- $\mathbf{U}_i = \left[runif(0.5, 1)\right]_{\{p_i \times r_J\}}$
- $\boldsymbol{\theta}_1 = \left[runif(0.5, 1) \quad 0\right]_{\{1 \times r_J\}}$ with the first $r_{prop} \times r_J$ values being non-zero
- take QR decomposition of $\begin{bmatrix} \mathbf{U}_1 \\ \vdots \\ \mathbf{U}_k \\ \boldsymbol{\theta}_1 \end{bmatrix}_{\{\sum p_i + 1 \times r_J\}}$ for new $\mathbf{U}_i$ and $\boldsymbol{\theta}_1$ values
- $\mathbf{S}_J = \text{diag}(w_J)_{\{r_J \times r_J\}} \left[rnorm(0, 1)\right]_{\{r_J \times n\}}$

for each dataset $i = 1, ..., K$

- $\mathbf{W}_i = \left[runif(0.5, 1)\right]_{\{p_i \times r_{A_i}\}}$
- $\boldsymbol{\theta}_{2i} = \left[runif(0.5, 1) \quad 0\right]_{\{1 \times r_{A_i}\}}$ with the first $r_{prop} \times r_{A_i}$ values being non-zero
- take QR decomposition of $\begin{bmatrix} \mathbf{W}_i \\ \boldsymbol{\theta}_{2i} \end{bmatrix}_{\{p_i + 1 \times r_{A_i}\}}$ for new $\mathbf{W}_i$ and $\boldsymbol{\theta}_{2i}$ values
- $\mathbf{S}_i = \text{diag}(w_A)_{\{r_{A_i} \times r_{A_i}\}} \left[rnorm(0, 1)\right]_{\{r_{A_i} \times n\}} \cdot \left(I_{\{n \times n\}} - P_{\mathbf{S}_J}\right)$
  - where $P_{\mathbf{S}_J} = \mathbf{S}_J^T (\mathbf{S}_J \mathbf{S}_J^T)^{-1} \mathbf{S}_J$ to force orthogonality between $\mathbf{S}_J$ and $\mathbf{S}_i$

We then can calculate $\mathbf{X}$ and $\mathbf{y}$

- $\mathbf{X}_i = \mathbf{U}_i\mathbf{S}_J + \mathbf{W}_i\mathbf{S}_i + \mathbf{E}_i$ where $\mathbf{E}_i$ is normal with variance s.t. $var(\mathbf{E}_i)/var(\mathbf{X}_i) = X_{err}$
- $\mathbf{y} = \boldsymbol{\theta}_1\mathbf{S}_J + \sum_i \boldsymbol{\theta}_{2i}\mathbf{S}_i + \mathbf{E}_Y$ where $\mathbf{E}_Y$ is normal with variance s.t. $var(\mathbf{E}_Y)/var(\mathbf{y}) = Y_{err}$
- Scale $\mathbf{X}$ and $\mathbf{y}$ to have variance=1
- Scale components to force $\|[\mathbf{U}^T\boldsymbol{\theta}_1^T]^T\|_F^2 = \|[\mathbf{W}_i^T\boldsymbol{\theta}_{2i}^T]^T\|_F^2 = 1\ \forall i = 1,\ldots,k$ for uniqueness

Return $\mathbf{X}_i$, $i = 1,\ldots,k$ and $\mathbf{y}$.

4 Reducing dimensionality

Though computation time relies on a variety of factors, data dimensions and rank selection are the main drivers. The computation times displayed in Table 6 of the main article were conducted on a 2.4 GHz computer with 8 GB RAM. In high-dimensional scenarios, JIVE maps the data into a lower dimension space before running its optimization function in increase efficiency.

sJIVE uses this same technique. Consider a high-dimensional scenario when $p_i >> n$ for the ith dataset. Then let $\mathbf{X}_i$ be the $p_i \times n$ data matrix. Prior to running the optimization function, map $\mathbf{X}_i$ to an $n \times n$ space using SVD, i.e.,

$$SVD(\mathbf{X}_i) = UDV^T$$
$$\mathbf{X}_i^{\perp} = DV^T$$

This transformation preserves covariance and Euclidean distance between columns in $\mathbf{X}_i$. By implementing this reduction in dimensionality, computation time can significantly decrease. For example in our COPDGene application, sJIVE took 52.6 hours to run without data reduction, but only 66 minutes after utilizing this transformation.

After optimizing the function and obtaining estimates for each of the joint and individual components, $\mathbf{X}_i^{\perp}$ can be mapped back to the original space by multiplying the left singular vectors, U, by the estimated loadings, $\mathbf{U}_i^{\perp}$ and $\mathbf{W}_i^{\perp}$ $i = 1,\ldots,k$. The scores, $\mathbf{S}_J$ and $\mathbf{S}_i$, and the $\boldsymbol{\theta}$ coefficients do not require any transformation.

References

[1] Q. Feng, M. Jiang, J. Hannig, J. Marron, Angle-based joint and individual variation explained, Journal of multivariate analysis 166 (2018) 241–265. doi:10.1016/j.jmva.2018.03.008.

[2] E. F. Lock, K. A. Hoadley, J. S. Marron, A. B. Nobel, Joint and individual variation explained (JIVE) for integrated analysis of multiple data types., The annals of applied statistics 7 (1) (2013) 523–542. doi:10.1214/12-AOAS597.