Supplementary material: Penalized Orthogonal Iteration for Sparse Estimation of Generalized Eigenvalue Problem

Sungkyu Jung
Department of Statistics, Seoul National University
Jeongyoun Ahn
Department of Statistics, University of Georgia
and
Yongho Jeon
Department of Applied Statistics, Yonsei University

October 12, 2018

Contents

S1 Supplement to Section 1: Generalized eigenvalue problems in statistics

A number of multivariate statistical methods can be formulated into a generalized eigenvalue problem (GEP). As referenced in the main article Section 1, we list a few examples of the data-analytic situations and methods that can be cast into a GEP, and for each situation discuss existing approaches for sparse solutions.

S1.1 Linear dimension reduction

Linear dimension reduction finds linear combinations of variables that span a lower-dimensional subspace for data $\mathbf{X}$. The *principal component analysis (PCA)* is a prime example.

Example 1 (PCA). Let a random vector $\mathbf{x}$ has the covariance matrix $\boldsymbol{\Sigma}$. It is well-known that the principal components are given by the eigen-decomposition of $\boldsymbol{\Sigma}$. By setting $\mathbf{A} = \boldsymbol{\Sigma}$, $\mathbf{B} = \mathbf{I}_p$, the problem

$$\mathbf{A}\mathbf{u}_j = \lambda_j \mathbf{B}\mathbf{u}_j, \tag{S1}$$

becomes the ordinary eigen-decomposition problem, and the solution $(\mathbf{u}_j, \lambda_j)$ corresponds to the principal component (PC) direction and variance pair.

There have been a number of proposals for sparse PCA; to name a few, Jolliffe et al. (2003); Zou et al. (2006); d'Aspremont et al. (2008); Shen and Huang (2008); Witten et al. (2009); Ma (2013); Bouveyron et al. (2016). Theoretical guarantees for some of the sparse PCA methods have been given in e.g., Shen et al. (2013) and Ma (2013). In computing sparse PCs, most of these methods utilized the fact that the original data matrix $\mathbf{X}$ is available, and proposed to modify the standard singular value decomposition (e.g., Shen and Huang, 2008; Witten et al., 2009) or forming a penalized regression problem (e.g., Zou et al., 2006). In contrast, we focus on solving the GEP directly, thus require computing the sample covariance matrix $\widehat{\Sigma}$ as an input to our algorithm. As pointed out in Remark 1 (in the main article), a special case of our proposal coincides with the method of Ma (2013), in which $\widehat{\Sigma}$ is also used, and the method is shown to be consistent in a high-dimensional sparse setting.

Some extensions of PCA and factor models that incorporate structures in the data (Jenatton et al., 2010; Allen et al., 2014; Lock et al., 2013; Li and Jung, 2017) have a potential to be cast into a GEP, by e.g. formulating the $\mathbf{B}$ matrix according to the structure given a priori.

Invariant co-ordinate selection (ICS, Tyler et al., 2009) is a general framework, examples of which includes the linear discriminant analysis and the independent component analysis (Hyvärinen et al., 2004). Since the coordinates of ICS are precisely given by the generalized eigenvectors of a pair of general scatter matrices, our algorithm may be used as a sparse estimation method for ICS.

Many moment-based *sufficient dimension reduction* (SDR) methods can also be formulated as a GEP, as shown by Li (2007). As an example, we provide a GEP formulation of the sliced inverse regression (SIR, Li, 1991), the most well-known method of SDR.

Example 2 (SDR). Consider the regression of a univariate response y on p-variate predictor $\mathbf{x}$. SDR aims to find a projection of data $\mathbf{x}$ that is sufficient for (i.e., preserves all information about) the conditional distribution of y given $\mathbf{x}$. By setting $\mathbf{A} = \mathrm{Cov}[\mathrm{E}\{\mathbf{x} - \mathrm{E}(\mathbf{x})|y\}]$ and $\mathbf{B} = \mathrm{Cov}(\mathbf{x})$, the eigenvectors of the GEP, equation (S1), span exactly the predictor subspace given by SIR.

We refer to Cook (2009), Li (2007) and Chen et al. (2010) for various SDR approaches and their relation to the generalized eigenvalue problem.

S1.2 Group mean difference and classification

Suppose that $\mathbf{x}$ follows a K-mixture of multivariate normal distributions, each with mean $\boldsymbol{\mu}_i$ and a common variance $\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}$, for all $i = 1, \ldots, K$, where $K > 1$. Assuming that the group membership y is also observed, common multivariate analyses incurred are classification of observations and testing the hypothesis of equal means.

Example 3 (Multiclass linear discriminant analysis (LDA)). Write $\boldsymbol{\Sigma}_T = \mathrm{Cov}(\mathbf{x})$ for the total-covariance matrix and $\boldsymbol{\Sigma}_W = \boldsymbol{\Sigma}$ for the within-covariance matrix. The between-covariance matrix is $\boldsymbol{\Sigma}_B = \boldsymbol{\Sigma}_T - \boldsymbol{\Sigma}_W$, whose rank is at most $K - 1$. The multiclass LDA finds a discriminant subspace whose basis vectors $\mathbf{u}_j$ are sequentially obtained by maximizing the Rayleigh coefficient:

$$T(\mathbf{u}) = \mathbf{u}^{\mathrm{T}}\boldsymbol{\Sigma}_B\mathbf{u}/\mathbf{u}^{\mathrm{T}}\boldsymbol{\Sigma}_W\mathbf{u}. \tag{S2}$$

This problem is equivalent to the GEP with $\mathbf{A} = \boldsymbol{\Sigma}_B$ and $\mathbf{B} = \boldsymbol{\Sigma}_W$.

There are vast literature on sparse estimation of the linear discriminant rule in high dimensions (Cai and Liu, 2011; Shao et al., 2011; Clemmensen et al., 2011; Mai et al., 2012, 2017; Witten and Tibshirani, 2011; Gaynanova et al., 2016). Witten and Tibshirani (2011) have noticed that the LDA direction in the binary classification is the solution of (S2), and proposed to solve a penalized GEP of the form

$$\max_{\mathbf{u}} \mathbf{u}^{\mathrm{T}}\mathbf{A}\mathbf{u} - p_\rho(\mathbf{u}), \quad \text{subject to } \mathbf{u}^{\mathrm{T}}\mathbf{B}\mathbf{u} = 1,$$

with the lasso or fussed lasso penalty for p_ρ. Clemmensen et al. (2011), Mai et al. (2017) and Gaynanova et al. (2016) focused on the multi-category classification and aimed to estimate the discriminating subspace, similar to us. While Clemmensen et al. (2011) turned the classification problem into a regression setting, thus used the original data matrix $\mathbf{X}$, the methods of Mai et al. (2017) and Gaynanova et al. (2016) are based on the empirical versions of $\boldsymbol{\Sigma}_B$ and $\boldsymbol{\Sigma}_W$. Their objective functions are thus similar to that of our Fast POI; detailed discussion on the similarity can be found in Remark 2 in Section 2.2.2 (in the main article).

In the multiple population situation, the *Multivariate analysis of variance (MANOVA)* provides a simple means of testing the hypothesis $H_0 : \boldsymbol{\mu}_1 = \cdots = \boldsymbol{\mu}_K$, based on the between-covariance matrix $\boldsymbol{\Sigma}_B$ and the within-covariance matrix $\boldsymbol{\Sigma}_W$. Among the choices of test statistics for MANOVA are functions of the generalized eigenvalues λ_j, from the GEP with (S2), replacing $\boldsymbol{\Sigma}_B$ and $\boldsymbol{\Sigma}_W$ by the empirical counterparts. For example, Roy's test statistic is $\lambda_1 = \max(\lambda_j)$, and the Lawley-Hotelling statistic is $\sum_{j=1}^{K-1}\lambda_j$. The test of MANOVA rejects H_0 for larger values of test statistics. Sparse MANOVA using our proposal is not considered in the present paper, but is an interesting future work.

S1.3 Canonical correlation analysis

Canonical correlation analysis (CCA) can be thought of as a special case of linear dimension reduction.

Example 4 (Canonical correlation analysis (CCA)). Let $\mathbf{x}, \mathbf{y}$ be two random vectors of dimensions p and q, respectively. Write $\boldsymbol{\Sigma}_1 = \mathrm{Cov}(\mathbf{x})$, $\boldsymbol{\Sigma}_2 = \mathrm{Cov}(\mathbf{y})$ and $\boldsymbol{\Sigma}_{12} = \mathrm{Cov}(\mathbf{x}, \mathbf{y}) =$

$\mathbf{\Sigma}_{21}^{\mathrm{T}}$. The coefficient vectors for the first pair of canonical variables are $(\mathbf{g}_1, \mathbf{h}_1) \in \mathbb{R}^p \times \mathbb{R}^q$, maximizing the correlation between $\mathbf{g}^{\mathrm{T}}\mathbf{x}$ and $\mathbf{h}^{\mathrm{T}}\mathbf{y}$:

$$\mathrm{Corr}(\mathbf{g}^{\mathrm{T}}\mathbf{x}, \mathbf{h}^{\mathrm{T}}\mathbf{y}) := \rho(\mathbf{g}, \mathbf{h}) = \frac{\mathbf{g}^{\mathrm{T}}\mathbf{\Sigma}_{12}\mathbf{h}}{(\mathbf{g}^{\mathrm{T}}\mathbf{\Sigma}_1\mathbf{g})^{\frac{1}{2}}(\mathbf{h}^{\mathrm{T}}\mathbf{\Sigma}_2\mathbf{h})^{\frac{1}{2}}}. \tag{S3}$$

Note that $\rho(\mathbf{g}, \mathbf{h})$ is invariant under individual scaling of $(\mathbf{g}, \mathbf{h})$. Using this invariance, a Lagrangian formulation of the maximization involves the condition $\mathbf{g}^{\mathrm{T}}\mathbf{\Sigma}_1\mathbf{g} = \mathbf{h}^{\mathrm{T}}\mathbf{\Sigma}_2\mathbf{h} = 1$. The first-order condition for the stationary points of the Lagrangian coincides with the GEP,

$$\begin{pmatrix} \mathbf{0} & \mathbf{\Sigma}_{12} \\ \mathbf{\Sigma}_{21} & \mathbf{0} \end{pmatrix} \mathbf{u} = \lambda \begin{pmatrix} \mathbf{\Sigma}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_2 \end{pmatrix} \mathbf{u}, \tag{S4}$$

where the solution $(\mathbf{u}, \lambda)$ corresponds to the concatenated coefficient vector $(\mathbf{g}_j^{\mathrm{T}}, \mathbf{h}_j^{\mathrm{T}})^{\mathrm{T}}$ and canonical correlation coefficient $\rho(\mathbf{g}_j, \mathbf{h}_j)$, respectively. An alternative formulation of CCA is given by solving the GEP (S4) with respect to either $\mathbf{g}_j$ or $\mathbf{h}_j$, thus leading to two GEPs:

$$\mathbf{\Sigma}_{12}\mathbf{\Sigma}_2^{-1}\mathbf{\Sigma}_{21}\mathbf{g} = \lambda\mathbf{\Sigma}_1\mathbf{g}, \quad \mathbf{\Sigma}_{21}\mathbf{\Sigma}_1^{-1}\mathbf{\Sigma}_{12}\mathbf{h} = \lambda\mathbf{\Sigma}_2\mathbf{h}. \tag{S5}$$

Sparse CCA has recently gained attention. See Witten et al. (2009); Safo et al. (2018); Chen et al. (2018); Gao et al. (2017) and references therein. Among those, it appears that Safo et al. (2018) is the only attempt to use sparse solutions of GEP as estimates for CCA. Witten et al. (2009)'s approach, solving (S3) directly with constraints on $\mathbf{g}$ and $\mathbf{h}$, can be understood as a regularized generalized singular value decomposition (Van Loan, 1976), thus is not equivalent to a GEP. Chen et al. (2018) proposed to modify the power method, a standard numerical algorithm for eigen-decomposition, by transforming the GEP into the standard eigenvalue problem.

S1.4 Nonlinear dimension reduction

Nonlinear dimension reduction methods, such as locally linear embedding, Laplacian eigenmaps, multi-dimensional scaling, and ISOMAP, come down to solving a GEP. Detailed expositions can be found in Kokiopoulou et al. (2011) and references therein.

We point out that our methods may not be immediately applicable to these problem. For example, a number of manifold learning methods are formulated as an n-dimensional GEP, where the matrices $\mathbf{A}$ and $\mathbf{B}$ contain the pairwise distances between observations. In such a case, the assumption of sparse loading translates to zero pairwise distances between many observations, which does not seem to be natural. Therefore it is not advisable to apply any sparse learning directly, not at least with a careful modification.

S2 Supplement to Section 4.1: Principal component Analysis

S2.1 Variable selection performance

The following measures are used in gauging the variable selection performance of generalized eigenvector estimates.

Let $\mathbf{U} = (u_{ij})$ be the $p \times d$ matrix of true eigenvectors, and $\hat{\mathbf{U}} = (\hat{u}_{ij})$ be its estimate of the same size. Note that if the ith variable (or the ith coordinate) is a signal variable if $u_{ij} \neq 0$ for one or more $j = 1, \ldots, d$, and is a non-signal variable if $u_{ij} = 0$ for all j. Thus, to measure the variable selection performance, for any matrix $\mathbf{V}$, the set of "positive" indices is defined by for $\epsilon \geq 0$,

$$s_\epsilon(\mathbf{V}) = \{i : \sum_{j=1}^{d} v_{ij}^2 > \epsilon, i = 1, \ldots, p\},$$

and the set of "negative" indices by

$$s_\epsilon^C(\mathbf{V}) = \{i : \sum_{j=1}^{d} v_{ij}^2 \leq \epsilon, i = 1, \ldots, p\}.$$

Note that $s_\epsilon(\mathbf{V}) \cup s_\epsilon^C(\mathbf{V}) = \{1, \ldots, p\}$ for any $p \times d$ matrix $\mathbf{V}$. For the truth $\mathbf{U}$ and an estimate $\hat{\mathbf{U}}$, we compute $s_0(\mathbf{U})$ and $s_\epsilon(\hat{\mathbf{U}})$ for $\epsilon = 10^{-10}$, which then leads to the following:

1. Total positive: $\mathrm{P} = \#\mathrm{s}_\epsilon(\hat{\mathbf{U}})$,

2. True positive: $\mathrm{TP} = \#\mathrm{s}_0(\mathbf{U}) \cap \mathrm{s}_\epsilon(\hat{\mathbf{U}})$,

From the false positive count, $\mathrm{FP} = \#\mathrm{s}_0^\mathrm{C}(\mathbf{U}) \cap \mathrm{s}_\epsilon(\hat{\mathbf{U}})$, the true negative count, $\mathrm{TN} = \#\mathrm{s}_0^\mathrm{C}(\mathbf{U}) \cap \mathrm{s}_\epsilon^\mathrm{C}(\hat{\mathbf{U}})$, the false negative count, $\mathrm{FN} = \#\mathrm{s}_0(\mathbf{U}) \cap \mathrm{s}_\epsilon^\mathrm{C}(\hat{\mathbf{U}})$ and the total negative count, $\mathrm{N} = \#\mathrm{s}_\epsilon^\mathrm{C}(\hat{\mathbf{U}}) = \mathrm{TN} + \mathrm{FN}$, we compute

3. Sensitivity: TP/P,

4. Specificity: TN/N,

5. Matthews correlation coefficient: $\frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{[\mathrm{P}(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})\mathrm{N}]^{1/2}}$

It may be of interest to inspect the sparsity patterns of $\hat{\mathbf{U}}$ in terms of each element of $\hat{\mathbf{U}}$. For this, instead of inspecting the row ℓ_2 norms of the matrix, we define the set of positive "loadings" of the matrix $\mathbf{V}$ by $s_\epsilon(\mathrm{vec}(\mathbf{V}))$, where $\mathrm{vec}(\mathbf{V})$ "vectorizes" $\mathbf{V}$ by stacking the column vectors vertically (i.e. $\mathrm{vec}(\mathbf{V})$ is a matrix of size $(pd) \times 1$). The five measures defined above can be now used to extract the loading-wise sparsity pattern.

From the simulation studies in Section 4.1 (of the main article), we have computed the five measures of variable selection performance, both in terms of coordinate-wise sparsity and the loading-wise sparsity, for each method considered. The performances of the cross-validated estimates are presented in Tables S1, S2 and S3 based on 100 repetition.

As noted in the main article, FastPOI-C shows the best variable selection performance for Model I, while POI-C is a close contender; see Table S1. We note that both POI-L and FastPOI-L tend to include more variables, and also more loadings, than needed, indicated by low specificity. The solutions of POI-L and FastPOI-L are in general not element-wise sparse. This may be counter-intuitive, but is not. Since we utilize the QR decomposition in the POI algorithm, column-specific sparse loadings are not preserved.

The eigenvectors for Models II and III are specifically designed to reduce the effect of QR decomposition in the algorithm in destroying column-specific sparsity patterns; the QR decomposition of the true (element-wise sparse) eigenvector matrix results in the same, element-wise sparse, matrix. In Tables S2 and S3, we find that POI-C and POI-L (or FastPOI-L and FastPOI-C) have similar variable selection performances. For Model II, Shen and Huang (2008)'s method performed the best while our methods are comparable to it; for Model III, POI-L, FastPOI-C and Shen and Huang (2008)'s method performed comparable to each other.

Note that Song et al. (2015)'s method did not provide a sparse estimate, when tuned by our tuning procedure. We checked that by choosing a large tuning parameter λ, Song et al. (2015)'s estimate is sparse, but the quality of estimate become progressively worse for larger values of λ.

S2.2 Computation times

We report computation times for sparse PCA. While both FastPOI-L and FastPOI-C are among the fastest, both POI-L and POI-C require significantly longer computation times than other methods.

Model I ($d = 3, p = 200$)		POI-L	POI-C	FastPOI-L	FastPOI-C	Zou et al.	Song et al.	Shen & Huang
Coord.	Total pos.	56.50	30.12	44.18	28.95	50.65	200.00	50.53
	True pos.	10.00	10.00	10.00	10.00	10.00	10.00	10.00
	Sensitivity	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Specificity	0.76	0.89	0.82	0.90	0.79	0.00	0.79
	Matthews	0.39	0.59	0.46	0.59	0.42	0.00	0.42
Loading	Total pos.	169.50	90.36	132.54	86.85	68.48	403.16	68.01
	True pos.	30.00	30.00	30.00	30.00	23.99	23.16	23.63
	Sensitivity	1.00	1.00	1.00	1.00	0.80	0.77	0.79
	Specificity	0.76	0.89	0.82	0.90	0.92	0.33	0.92
	Matthews	0.39	0.59	0.46	0.59	0.52	0.05	0.51

Model I ($d = 5, p = 500$)		POI-L	POI-C	FastPOI-L	FastPOI-C	Zou et al.	Song et al.	Shen & Huang
Coord.	Total pos.	109.01	21.86	63.45	18.37	61.51	500.00	48.69
	True pos.	10.00	10.00	10.00	10.00	10.00	10.00	10.00
	Sensitivity	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Specificity	0.80	0.98	0.89	0.98	0.89	0.00	0.92
	Matthews	0.30	0.74	0.44	0.78	0.48	0.00	0.53
Loading	Total pos.	545.05	109.30	317.25	91.85	91.07	2002.96	75.53
	True pos.	50.00	50.00	50.00	50.00	34.76	42.96	34.32
	Sensitivity	1.00	1.00	1.00	1.00	0.70	0.86	0.69
	Specificity	0.80	0.98	0.89	0.98	0.98	0.20	0.98
	Matthews	0.30	0.74	0.44	0.78	0.56	0.02	0.59

Table S1: Measures of variable selection performance of the cross-validated estimates, based on the simulations for PCA model I. See text for description of the measures and methods involved. Shown are averages from 100 repetitions. For Tables S1, S2 and S3, the standard errors of the total positive and true positive counts are at most 17.5 and 3.5, respectively; the standard errors of the sensitivity, specificity, and Mathews correlation coefficient are at most 0.03.

Model II ($d = 3, p = 200$)		POI-L	POI-C	FastPOI-L	FastPOI-C	Zou et al.	Song et al.	Shen & Huang
Coord.	Total pos.	20.94	23.01	21.58	21.71	24.04	200.00	19.45
	True pos.	15.00	15.00	15.00	15.00	15.00	15.00	15.00
	Sensitivity	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Specificity	0.97	0.96	0.96	0.96	0.95	0.00	0.98
	Matthews	0.87	0.82	0.86	0.84	0.84	0.00	0.90
Loading	Total pos.	62.82	69.03	64.74	65.13	31.40	405.00	21.04
	True pos.	15.00	15.00	15.00	15.00	13.30	14.10	12.53
	Sensitivity	1.00	1.00	1.00	1.00	0.89	0.94	0.84
	Specificity	0.92	0.91	0.91	0.91	0.97	0.33	0.99
	Matthews	0.49	0.46	0.48	0.47	0.66	0.09	0.72

Model II ($d = 5, p = 500$)		POI-L	POI-C	FastPOI-L	FastPOI-C	Zou et al.	Song et al.	Shen & Huang
Coord.	Total pos.	76.37	88.74	42.60	40.25	56.71	500.00	43.68
	True pos.	24.46	24.17	24.02	24.04	24.65	25.00	24.71
	Sensitivity	0.98	0.97	0.96	0.96	0.99	1.00	0.99
	Specificity	0.89	0.86	0.96	0.97	0.93	0.00	0.96
	Matthews	0.59	0.55	0.77	0.78	0.71	0.00	0.81
Loading	Total pos.	381.85	443.70	213.00	201.25	71.85	2004.95	48.79
	True pos.	24.46	24.17	24.02	24.04	22.14	24.29	21.81
	Sensitivity	0.98	0.97	0.96	0.96	0.89	0.97	0.87
	Specificity	0.86	0.83	0.92	0.93	0.98	0.20	0.99
	Matthews	0.26	0.24	0.34	0.34	0.58	0.04	0.69

Table S2: Measures of variable selection performance of the cross-validated estimates, based on the simulations for PCA model II. See text for description of the measures and methods involved. Shown are averages from 100 repetitions.

Model III ($d = 3, p = 200$)		POI-L	POI-C	FastPOI-L	FastPOI-C	Zou et al.	Song et al.	Shen & Huang
Coord.	Total pos.	17.16	21.42	26.41	20.69	24.86	200.00	23.08
	True pos.	15.00	15.00	15.00	15.00	15.00	15.00	15.00
	Sensitivity	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Specificity	0.99	0.97	0.94	0.97	0.95	0.00	0.96
	Matthews	0.95	0.85	0.77	0.87	0.83	0.00	0.87
Loading	Total pos.	56.38	69.04	83.88	66.87	43.97	405.57	34.56
	True pos.	45.00	45.00	45.00	45.00	32.25	33.92	24.86
	Sensitivity	1.00	1.00	1.00	1.00	0.72	0.75	0.55
	Specificity	0.98	0.96	0.93	0.96	0.98	0.33	0.98
	Matthews	0.89	0.80	0.73	0.81	0.72	0.04	0.63

Model III ($d = 5, p = 500$)		POI-L	POI-C	FastPOI-L	FastPOI-C	Zou et al.	Song et al.	Shen & Huang
Coord.	Total pos.	111.97	39.94	94.45	36.60	66.04	500.00	41.24
	True pos.	25.00	25.00	25.00	25.00	25.00	25.00	25.00
	Sensitivity	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Specificity	0.82	0.97	0.85	0.98	0.91	0.00	0.97
	Matthews	0.57	0.81	0.52	0.83	0.65	0.00	0.85
Loading	Total pos.	576.81	220.77	490.82	204.22	121.38	2004.37	73.31
	True pos.	119.63	120.00	120.00	120.00	73.47	95.06	53.30
	Sensitivity	1.00	1.00	1.00	1.00	0.61	0.79	0.44
	Specificity	0.81	0.96	0.84	0.96	0.98	0.20	0.99
	Matthews	0.50	0.71	0.46	0.73	0.59	-0.00	0.56

Table S3: Measures of variable selection performance of the cross-validated estimates, based on the simulations for PCA model III. See text for description of the measures and methods involved. Shown are averages from 100 repetitions.

Model	d	p	POI-L	POI-C	FastPOI-L	FastPOI-C	Zou et al.	Song et al.	Shen & Huang
I	3	200	7.93	2.27	0.12	0.13	0.19	0.23	0.59
	3	500	63.17	26.30	0.44	0.35	0.26	0.58	0.87
	5	200	14.74	10.09	0.24	0.22	0.30	0.70	1.44
	5	500	90.31	72.94	0.60	0.41	0.37	1.20	1.81
II	3	200	6.59	2.80	0.11	0.11	0.15	0.20	0.48
	3	500	56.50	27.20	0.36	0.30	0.22	0.49	0.76
	5	200	10.88	9.60	0.17	0.16	0.20	0.53	1.08
	5	500	117.42	88.52	0.76	0.53	0.40	1.40	2.13
III	3	200	6.88	2.62	0.12	0.11	0.16	0.22	0.56
	3	500	59.13	25.94	0.38	0.31	0.20	0.50	0.75
	5	200	12.16	9.38	0.16	0.15	0.20	0.52	1.08
	5	500	96.51	75.24	0.57	0.38	0.31	1.09	1.67

Table S4: Computation times (in seconds) for sparse PCA, averaged over 100 repetitions. For our methods, shown are the computation times needed to complete the cross-validation with $|L| = 31$ candidate values of tuning parameter λ. All other methods also computed over 31 different values of their tuning parameters.

S3 Supplement to Section 4.2: Multiclass LDA

We report additional simulation results from the same simulation reported in Section 4.2 (in the main article).

S3.1 Variable selection performance

The variable selection performance is measured to investigate the coordinate-wise sparsity pattern of the estimates. We report the results from the cross-validated estimates in Table S5. In Table S5, it is seen that our methods tend to choose more variables than needed, but shows better sensitivity than other methods. Overall, both FastPOI-C and Gaynanova et al. (2016)'s method show better performances than other methods, and there is no clear winner among FastPOI-C and Gaynanova et al. (2016)'s.

S3.2 Extension of main article Tables 2 and 4

We report an extension of Tables 3 and 4, in the main article. We included the numerical results from two more choices of tuning parameter, $\tilde{\lambda}$ (for the ideal choice) and $\breve{\lambda}$ (for minimizing misclassification rates), as well as using $\hat{\lambda}$ that maximizes the sum of predicted

Number of total positives

Model	FastPOI-L	FastPOI-C	Mai et al.	Clemmensen et al.	Gaynanova et al.
I	11.90	10.61	12.45	9.10	8.52
II	112.25	14.91	94.88	6.85	6.31
III	17.23	12.23	9.40	7.75	8.76
IV	83.31	10.40	50.98	6.97	6.95
V	108.15	23.10	16.42	15.34	12.89

Number of true positives

Model	FastPOI-L	FastPOI-C	Mai et al.	Clemmensen et al.	Gaynanova et al.
I	4.97	4.98	4.80	4.38	4.39
II	4.99	4.39	4.23	3.42	3.66
III	4.93	4.94	4.59	3.35	3.86
IV	3.85	3.61	2.95	3.30	3.02
V	4.99	4.87	3.09	4.88	4.32

Sensitivity

Model	FastPOI-L	FastPOI-C	Mai et al.	Clemmensen et al.	Gaynanova et al.
I	0.99	1.00	0.96	0.88	0.88
II	1.00	0.88	0.85	0.68	0.73
III	0.99	0.99	0.92	0.67	0.77
IV	0.96	0.90	0.74	0.82	0.76
V	1.00	0.97	0.62	0.98	0.86

Specificity

Model	FastPOI-L	FastPOI-C	Mai et al.	Clemmensen et al.	Gaynanova et al.
I	0.96	0.97	0.96	0.98	0.98
II	0.45	0.95	0.54	0.98	0.99
III	0.94	0.96	0.98	0.98	0.97
IV	0.59	0.97	0.75	0.98	0.98
V	0.47	0.91	0.93	0.95	0.96

Matthews correlation coefficient

Model	FastPOI-L	FastPOI-C	Mai et al.	Clemmensen et al.	Gaynanova et al.
I	0.75	0.79	0.69	0.69	0.72
II	0.14	0.56	0.20	0.59	0.66
III	0.61	0.72	0.80	0.55	0.69
IV	0.17	0.65	0.32	0.66	0.61
V	0.19	0.56	0.56	0.58	0.64

Table S5: Measures of variable selection performance, based on the simulations for MLDA models. See text for description of the measures and methods involved. Shown are averages from 100 repetitions. The standard errors of each measure are at most 6.88, 0.13, 0.03, 0.04 and 0.02, respectively.

eigenvalues (our proposal). We briefly explain $\tilde{\lambda}$ and $\breve{\lambda}$ below.

The ideal choice of tuning parameter is given by using $\mathbf{U}(\tilde{\lambda})$ where $\tilde{\lambda} = \arg\min_{\lambda \in L} \rho(\widehat{\mathbf{U}}(\lambda), \mathbf{U})$ is chosen to minimize the distance to the true subspace.

Given that the classification is the goal of analysis, one could use a tuning set to directly tune the performance of the classification. To implement this alternative method of tuning λ, the multiclass LDA is trained for $\mathbf{X}\widehat{\mathbf{U}}(\lambda)$ for each choice of λ. The tuned parameter $\breve{\lambda}$ is the value of λ for which the misclassification error rate (MCE) of the tuning data set is the smallest. All three choices of tuning parameters are used for our methods (FastPOI-L and FastPOI-C) and the competing methods.

In Tables S6 and S7, the choice of λ by “Ideal” stands for $\tilde{\lambda}$, the ideal choice; by “Pred” we mean $\hat{\lambda}$; and by “MCE” we mean $\breve{\lambda}$, minimizing the misclassification error rate. Numerical results in the tables are based on 100 repetitions. We note that our numerical results are similar to each other for different choices of tuning parameter. In particular, FastPOI-L solution has the smallest possible distance to truth, among all methods considered, for Models I–IV, and the performance of FastPOI-C from either choice of tuning is comparable.

S3.3 Computation times

We report computation times for sparse LDA methods in Table S8. All methods except Gaynanova et al's were implemented in Matlab. For Gaynanova et al's method, we used the R package `MGSDA`. While Clemmensen et al's method requires significantly longer computation times, there was no clear winner among the other methods.

S4 Supplement to Section 4.3: Sufficient dimension reduction

We provide numerical evidences for numerical instability of Chen et al. (2010)'s method and that our approach is much faster than Chen et al. (2010)'s.

In Table S9, we compare the average computation times needed to estimate the 2-dimensional sufficient subspace, computed from variants of sliced inverse regression (SIR). As the dimension p increases, the computation times for all methods also increase. The difference in computation times between SIR (Li, 1991) and SIR with POI-C (POI with coordinate-wise sparsity) is exactly the extra time needed to replace the standard gener-

Model	Choice of λ	FastPOI-L	FastPOI-C	Mai et al.	Clemmensen et al.	Gaynanova et al.
I	Ideal	0.303	0.292	0.340	0.614	0.311
	Pred.	0.328	0.313	0.371	0.622	0.377
	MCE	0.366	0.342	0.406	0.630	0.389
II	Ideal	0.392	0.553	0.864	0.779	0.555
	Pred.	0.839	0.570	0.936	0.817	0.611
	MCE	0.797	0.605	0.914	0.858	0.607
III	Ideal	0.511	0.418	0.465	0.771	0.536
	Pred.	0.644	0.437	0.542	0.784	0.608
	MCE	0.634	0.468	0.516	0.806	0.612
IV	Ideal	0.372	0.437	0.854	0.662	0.451
	Pred.	0.852	0.478	0.916	0.689	0.514
	MCE	0.808	0.507	0.916	0.697	0.540
V	Ideal	0.381	0.323	0.823	0.333	0.289
	Pred.	0.712	0.359	0.869	0.365	0.411
	MCE	0.720	0.421	0.852	0.389	0.359

Table S6: The projection distance from the estimate, averaged from 100 repetitions, for sparse discriminant basis learning. The standard errors are at most 0.024. Smaller distance indicates more precise estimation.

Model	Choice of λ	FastPOI-L	FastPOI-C	Mai et al.	Clemmensen et al.	Gaynanova et al.
I	Ideal	7.28	7.15	7.75	10.08	7.24
	Pred.	7.46	7.27	9.41	10.23	12.30
	MCE	7.88	7.69	8.28	10.38	7.84
II	Ideal	37.67	8.60	23.72	10.79	9.16
	Pred.	30.50	8.72	21.74	9.70	12.49
	MCE	16.45	9.12	20.70	9.52	9.13
III	Ideal	22.65	12.19	14.40	18.03	15.63
	Pred.	17.10	12.41	17.23	18.41	19.19
	MCE	17.26	12.64	14.70	18.14	14.65
IV	Ideal	39.62	15.69	28.90	18.59	15.87
	Pred.	35.84	16.03	23.62	18.00	19.68
	MCE	28.59	16.39	23.84	18.49	15.91
V	Ideal	40.47	16.35	28.02	16.55	12.88
	Pred.	32.40	16.13	26.80	16.57	15.98
	MCE	22.69	14.90	23.56	16.65	13.16

Table S7: Misclassification rates (in percent) of the test set, averaged from 100 repetitions. The standard errors are at most 1.29. Smaller error rate indicates better classification.

Model	FastPOI-L	FastPOI-C	Mai et al.	Clemmensen et al.	Gaynanova et al.
I	0.56	0.82	0.78	11.27	0.60
II	1.12	2.13	0.68	7.62	0.82
III	0.52	0.77	0.74	5.11	0.61
IV	1.08	2.13	0.71	8.55	0.73
V	1.36	2.20	0.80	10.72	1.29

Table S8: Computation times (in seconds) for sparse LDA methods compared, averaged over 100 repetitions. For FastPOI-L and FastPOI-C, shown are the computation times needed to complete the cross-validation with $|L| = 31$ candidate values of tuning parameter λ. All other methods also computed over 31 different values of their tuning parameters.

alized eigen-decomposition by the sparse generalized eigen-decomposition, computed using POI-C algorithm. Note that for rank-deficit cases, e.g. $(n, p) = (100, 100)$ or $(100, 500)$, both the standard and penalized estimation required more computation times than for the case with full-rank matrices. For those rank-deficit cases, Chen et al. (2010)'s method did not converge in an hour $(3, 600, 000$ milliseconds), so we had to terminate the process and omitted the result. Even when it converged for other cases, the computation times are about 100 times longer than POI-C.

(n, p)	SIR	SIR with POI-C	Chen et al.
(100, 10)	3.22	2.99	537.81
(100, 100)	6.58	17.96	Did not finish
(100, 500)	62.85	904.33	Did not finish
(1000, 10)	2.83	3.02	335.17
(1000, 100)	5.50	10.15	3,488.63
(1000, 500)	43.15	119.89	104,579.02

Table S9: Sliced inverse regression for Tai-Chi data. Computation times in milliseconds (average of 10 trials). Computation was done using Matlab 2015 on a standard desktop computer (Intel i7-4770 CPU 2.40GHz with 16 gigabytes of RAM). Both SIR and Chen et al. (2010)'s method were implemented using the Matlab package of Coordinate-independent Sparse Estimation (Chen, 2018).

In Table S10, we compare the average accuracy of the estimated sufficient subspace. Note that SIR with POI-C have perfectly recovered the true subspace in all situations.

While Chen et al. (2010)'s method sometimes exhibits the exact recovery of true subspace, the algorithm is highly unstable. This can be seen in the table for $p = 10$, in which cases, the iterated solution of Chen et al. (2010) seemed to have converged to a local optimum, for roughly a half of time.

(n, p)	SIR	SIR with POI-C	Chen et al.
(100, 10)	0.98	0.00	0.89
(100, 100)	1.00	0	Did not finish
(100, 500)	1.00	0	Did not finish
(1000, 10)	0.96	0.00	0.60
(1000, 100)	0.99	0.00	0.00
(1000, 500)	1.00	0.00	0.00

Table S10: Sliced inverse regression for Tai-Chi data. Projection distance from the estimates, averaged from 10 trials.

S5 Supplement to Section 4.4: Canonical correlation analysis

We provide the model and simulation setting used in sparse estimation of canonical correlation analysis (CCA) and the numerical results.

We borrow the model used in Safo et al. (2018). In particular, the concatenated random vector $\mathbf{z}^{\mathrm{T}} = (\mathbf{x}^{\mathrm{T}}, \mathbf{y}^{\mathrm{T}})$ follows the multivariate normal distribution with mean $\mathbf{0}$ and covariance matrix

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_1 & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{12}^{\mathrm{T}} & \boldsymbol{\Sigma}_2 \end{pmatrix}.$$

For $0 \leq \rho < 1$ and a natural number s, let $C_s(\rho) = (1-\rho)\mathbf{I}_s + \rho\mathbf{J}_s$, $\mathbf{J}_s = \mathbf{1}_s\mathbf{1}_s^{\mathrm{T}}$. We set $\boldsymbol{\Sigma}_1$ as the block diagonal matrix of $C_{20}(.7)$ and $\mathbf{I}_{180}$, $\boldsymbol{\Sigma}_2$ as the block diagonal matrix of $C_{15}(.7)$ and $\mathbf{I}_{135}$, and $\boldsymbol{\Sigma}_{12}$ as the block diagonal matrix of $.6\mathbf{1}_{20}\mathbf{1}_{15}^{\mathrm{T}}$ and $\mathbf{0}_{180\times 135}$. Under this model, there is only one canonical pair $(\mathbf{g}, \mathbf{h}) \in \mathbb{R}^{200} \times \mathbb{R}^{150}$ in which only first 10% of coefficients are nonzero. The true canonical correlation is approximately $\rho = 0.8362$. This model corresponds to Setting I in Safo et al. (2018).

We applied the POI. Note that since there is only one vector to evaluate, POI-L is the same as to POI-C. The performances in estimating $(\mathbf{g}, \mathbf{h})$ and ρ are evaluated separately, using the measures defined in Section S2, based on 100 repetition. We also directly compare the performance of Gao et al. (2017)'s method. These are contained in Table S11. Safo et al. (2018) reported that their proposed method, called "SELP-I" performed the best in the Setting I of the paper, compared to methods of Gao et al. (2017), Witten et al. (2009), Parkhomenko et al. (2009) and Chalise and Fridley (2012). For reference we also list the numerical results of "SELP-I" in Table S11. In this setting, our method has a potential to provide much more accurate estimates. Our estimate using the proposed tuning procedure performs inferior in terms of accuracy to "SELP-I" of Safo et al. (2018), but shows a similar performance in terms of variable selection and canonical correlation estimation. Both our methods and Safo et al's performed much better than those of Gao et al. (2017), Witten et al. (2009), Parkhomenko et al. (2009) and Chalise and Fridley (2012).

		POI (min)	POI (CV)	Gao et al.	Safo et al.
α	Projection distance	0.135	0.239	0.997	0.144
	Sensitivity	1.000	1.000	0.749	1.000
	Specificity	0.975	0.968	0.920	0.993
	Matthews	0.907	0.902	0.642	0.964
β	Projection distance	0.129	0.225	0.994	0.144
	Sensitivity	1.000	1.000	0.711	1.000
	Specificity	0.976	0.962	0.923	0.988
	Matthews	0.910	0.895	0.624	0.945
	$\hat{\rho}$	0.838	0.847	0.904	0.839

Table S11: Performance in sparse CCA by the POI. POI (min) refers to the choice of tuning parameter by the minimum distance to truth; POI (CV) refers to the choice of tuning parameter by using the proposed cross validation procedure. The largest standard errors are 0.1, 0.26, 0.11, 0.21, 0.05 for Projection distance, Sensitivity, Specificity, Matthews correlation coefficient and $\hat{\rho}$, respectively.

S6 Supplment to Section 4: Genomic data analysis

We report an application of the proposed method in the exploratory data analysis of a large-scale genomic data. The data set was introduced in Ciriello et al. (2015), and consists of

16,615 gene expression levels measured for 817 breast cancer tumor samples. These tumor samples were pre-classified by a pathology committee, and grouped into five subtypes of lobular breast cancer—Luminal A, Basal-like, Luminal B, HER2-enriched, and normal-like. For this dataset, we apply sparse PCA and multiclass LDA using the proposed POI algorithm.

S6.1 Feature selection by sparse principal component analysis

We first used the data to understand the behavior of sparse PCA estimates by the POI. For this study, we kept the 500 variables with the largest standard deviations, and added to each observation 500 noise variables, sampled from the standard normal distribution. For any "sparse" estimation methods in this context, the estimated basis vectors should not include the 500 noise variables. To evaluate the performance in the smaller sample size situation, we use one third of the sample, consisting of $n = 272$ observations. The data are then standardized (so that each variable has mean zero and unit variance).

For this data set of size $(n, p) = (272, 1000)$, we use the POI with coordinate-wise sparse penalty (POI-C) with the tuning parameter given by $\lambda = \lambda_{\max}/2$ in estimation of principal subspace of dimension d. To glimpse the stability of the estimates against varying dimension d, we have repeated the analysis for $d = 1$ to $d = 20$, and have collected the estimated eigenvalues and eigenvectors. The result of analysis is graphically summarized in Fig. S1, which also appears in the main article.

In Fig. S1, notice that the estimated eigenvalues (shown in the left panel) are slowly decreasing but are stable across a range of d. It appears that the first three or four largest eigenvalue estimates "stand out" among others, indicating a potentially small number of true principal components.

In the middle panel, the sparsity patterns of estimated eigenspaces are shown. As desired, the latter 500 coordinates are estimated to be zero. Moreover, the number of nonzero coordinates seems to be stable as d increases.

The right panel of Fig. S1 shows the absolute value of the inner product between $\hat{\mathbf{q}}_{i,i}$ and $\hat{\mathbf{q}}_{i,d}$ for $d \geq i$, where $\hat{\mathbf{q}}_{i,d}$ is the ith principal component (PC) direction vector when estimating d PCs. For clarity, we show the first three PC directions, and they are stable against increasing d.

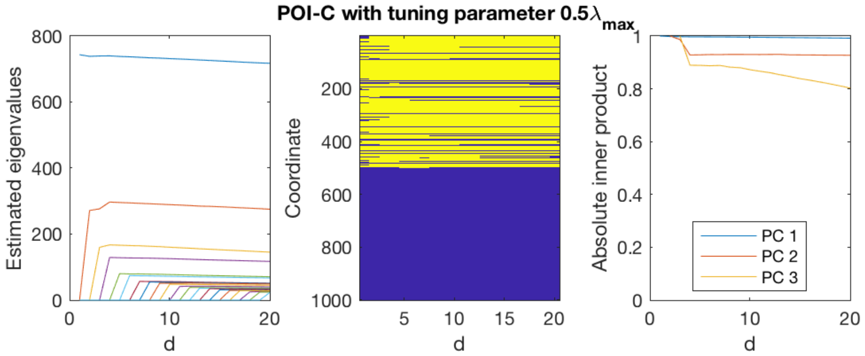

Figure S1: Sparse principal component analysis by POI with coordinate-wise sparse penalty for the genomic data set. The analysis is repeated for subspace dimension $d = 1$ to 20. Shown are estimated eigenvalues (left), nonzero coordinates shown as lighter color (middle) and $|\hat{\mathbf{q}}_{i,i}^{\mathrm{T}}\hat{\mathbf{q}}_{i,d}|$ for $i = 1, 2, 3$ (right).

S6.2 Linear classification

We now demonstrate the application of Fast POI in learning sparse discriminant basis from the data.

The data were split in half at random, where the first half with 409 cases was used for training, and the other half was used for testing. We kept the 2,000 variables with the largest standard deviations, and then standardized the data (so that each variable has mean zero and unit variance). The $\mathbf{B}$ and $\mathbf{A}$ matrices were then prepared by the sample estimates of the within-group and between group covariance matrices, $\mathbf{\Sigma}_W$ and $\mathbf{\Sigma}_B$, respectively. For this large data, sparse estimation of the generalized eigenvectors by Fast POI algorithms took only 2.09 seconds on average over a range of λ (on a standard Macbook). This is much faster than, e.g., estimating by Clemmensen et al. (2011)'s method (using the spaSM package, Sjöstrand et al., 2012) which took about a minute.

We further considered applying our methods to linear classification. For this experiment, the data were divided into three equal-sized groups: training, tuning and testing sets of size 272. Due to heavy-computation cost (mostly from using the spaSM package), only the 500 variables with the largest raw standard deviations were kept. We compared FastPOI-L, FastPOI-C, Mai et al. (2017)'s method, Clemmensen et al. (2011)'s method and Gaynanova et al. (2016)'s method, as used in the simulation study in the main article Section 4.2 .

The training set was used to estimate the subspace $\widehat{\mathbf{U}}$, while the tuning set was used

	Pred.		MCE	
	Error	#Signal	Error	#Signal
FastPOI-L	17.86 (0.23)	303.2 (58.2)	18.35 (0.19)	327.2 (129.5)
FastPOI-C	16.64 (0.19)	205.1 (45.5)	17.25 (0.19)	235.4 (126.1)
Mai et al.	17.40 (0.20)	311.6 (83.8)	17.36 (0.20)	365.3 (142.3)
Clemmensen et al.	17.89 (0.20)	218.2 (49.8)	18.21 (0.21)	199.9 (99.9)
Gaynanova et al.	22.64 (9.49)	83.37 (65.39)	18.06 (1.99)	79.4 (36.8)

Table S12: Basis learning for classification on the lobula breast cancer data. Column "Error" contains the means (standard errors) of the misclassification rates (in percent) of the test data set. Column "#Signal" contains the means (standard deviations) of the number of non-zero coordinates in the estimated basis. Data are randomly split for 100 times. "Pred" and "MCE" refer to the cross-validation method used.

to compute the cross-validation score. The testing set was used to compute estimates of misclassification rate. The tuning parameters were chosen by two different standards: one maximizing the predicted sum of eigenvalues, and one minimizing the tuning classification error. The performances of classification are summarized in Table S12. The results are mixed. All methods turn out to be equally well-performing. On the other hand, Gaynanova's method provides the smallest number of non-zero coordinates in the estimated eigenvector matrix.

References

Allen, G. I., L. Grosenick, and J. Taylor (2014). A generalized least-square matrix decomposition. *Journal of the American Statistical Association 109*(505), 145–159.

Bouveyron, C., P. Latouche, and P.-A. Mattei (2016). Bayesian variable selection for globally sparse probabilistic PCA. Preprint HAL 01310409, Université Paris Descartes. arXiv:1605.05918.

Cai, T. and W. Liu (2011). A direct estimation approach to sparse linear discriminant analysis. *Journal of the American Statistical Association 106*(496), 1566–1577.

Chalise, P. and B. L. Fridley (2012). Comparison of penalty functions for sparse canonical correlation analysis. *Computational Statistics and Data Analysis 56*, 245–254.

Chen, M., C. Gao, Z. Ren, and H. H. Zhou (2018). Sparse cca via precision adjusted iterative thresholding. In L. Y. Shing-Tung Yau and S.-Y. Cheng (Eds.), *Proceedings of International Congress of Chinese Mathematicians*, Volume to appear.

Chen, X. (2018). Matlab package for Coordinate-independent Sparse Estimation. Web link at https://www.stat.nus.edu.sg/ stacx/cise.zip, retrieved on May 25, 2018.

Chen, X., C. Zou, R. D. Cook, et al. (2010). Coordinate-independent sparse sufficient dimension reduction and variable selection. *The Annals of Statistics 38*(6), 3696–3723.

Ciriello, G., M. L. Gatza, A. H. Beck, M. D. Wilkerson, S. K. Rhie, A. Pastore, H. Zhang, M. McLellan, C. Yau, C. Kandoth, et al. (2015). Comprehensive molecular portraits of invasive lobular breast cancer. *Cell 163*(2), 506–519.

Clemmensen, L., T. Hastie, D. Witten, and B. Ersbll (2011). Sparse discriminant analysis. *Technometrics 53*(4), 406–413.

Cook, R. D. (2009). *Regression graphics: Ideas for studying regressions through graphics*, Volume 482. John Wiley & Sons.

d'Aspremont, A., F. Bach, and L. E. Ghaoui (2008). Optimal solutions for sparse principal component analysis. *Journal of Machine Learning Research 9*(Jul), 1269–1294.

Gao, C., Z. Ma, H. H. Zhou, et al. (2017). Sparse cca: Adaptive estimation and computational barriers. *The Annals of Statistics 45*(5), 2074–2101.

Gaynanova, I., J. G. Booth, and M. T. Wells (2016). Simultaneous sparse estimation of canonical vectors in the $p \gg n$ setting. *Journal of the American Statistical Association 111*(514), 696–706.

Hyvärinen, A., J. Karhunen, and E. Oja (2004). *Independent component analysis*, Volume 46. John Wiley & Sons.

Jenatton, R., G. Obozinski, and F. Bach (2010). Structured sparse principal component analysis. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 366–373.

Jolliffe, I. T., N. T. Trendafilov, and M. Uddin (2003). A modified principal component technique based on the lasso. *Journal of computational and Graphical Statistics 12*(3), 531–547.

Kokiopoulou, E., J. Chen, and Y. Saad (2011). Trace optimization and eigenproblems in dimension reduction methods. *Numerical Linear Algebra with Applications 18*(3), 565–602.

Li, G. and S. Jung (2017). Incorporating covariates into integrated factor analysis of multi-view data. *Biometrics 73*(4), 1433–1442.

Li, K.-C. (1991). Sliced inverse regression for dimension reduction (with discussion). *Journal of the American Statistical Association 86*(414), 316–342.

Li, L. (2007). Sparse sufficient dimension reduction. *Biometrika 94*(3), 603–613.

Lock, E. F., K. A. Hoadley, J. S. Marron, and A. B. Nobel (2013). Joint and individual variation explained (jive) for integrated analysis of multiple data types. *The Annals of Applied Statistics 7*(1), 523.

Ma, Z. (2013). Sparse principal component analysis and iterative thresholding. *The Annals of Statistics 41*(2), 772–801.

Mai, Q., Y. Yang, and H. Zou (2017). Multiclass sparse discriminant analysis. *Statistica Sinica* (to appear). arXiv:1504.05845.

Mai, Q., H. Zou, and M. Yuan (2012). A direct approach to sparse discriminant analysis in ultra-high dimensions. *Biometrika 99*(1), 29–42.

Parkhomenko, E., D. Tritchler, and J. Beyene (2009). Sparse canonical correlation analysis with application to genomic data integration. *Statistical Applications in Genetics and Molecular Biology 8.*

Safo, S. E., J. Ahn, Y. Jeon, and S. Jung (2018). Sparse generalized eigenvalue problem with application to canonical correlation analysis for integrative analysis of methylation and gene expression data. *Biometrics in press.*

Shao, J., Y. Wang, X. Deng, and S. Wang (2011). Sparse linear discriminant analysis by thresholding for high dimensional data. *The Annals of statistics*, 1241–1265.

Shen, D., H. Shen, and J. S. Marron (2013). Consistency of Sparse PCA in High Dimension , Low Sample Size Contexts. *Journal of Multivariate Analysis 115*, 317—-333.

Shen, H. and J. Z. Huang (2008). Sparse principal component analysis via regularized low rank matrix approximation. *Journal of multivariate analysis 99*(6), 1015–1034.

Sjöstrand, K., L. H. Clemmensen, R. Larsen, and B. Ersbøll (2012). Spasm: A matlab toolbox for sparse statistical modeling. *URL www2.imm.dtu.dk/projects/spasm/*.

Song, J., P. Babu, and D. P. Palomar (2015). Sparse generalized eigenvalue problem via smooth optimization. *IEEE Transactions on Signal Processing 63*(7), 1627–1642.

Tyler, D. E., F. Critchley, L. Dümbgen, and H. Oja (2009). Invariant co-ordinate selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 71*(3), 549–592.

Van Loan, C. F. (1976). Generalizing the singular value decomposition. *SIAM Journal on Numerical Analysis 13*(1), 76–83.

Witten, D. and R. Tibshirani (2011). Penalized classification using fisher's linear discriminant. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73*(5), 753–772.

Witten, D. M., R. Tibshirani, and T. Hastie (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics 10*(3), 515–534.

Zou, H., T. Hastie, and R. Tibshirani (2006). Sparse principal component analysis. *Journal of Computational and Graphical Statistics 15*(2), 265–286.