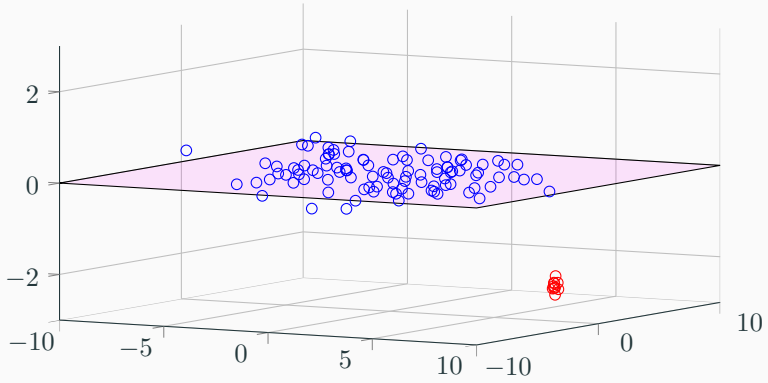
Flavors of sparse PCA

Arnaud Breloy, 2025

le cnam

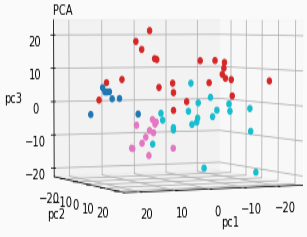
Subspace learning grounds

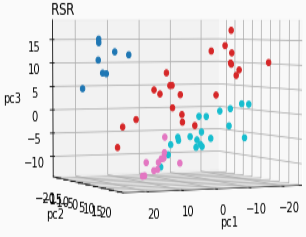
$$\mathbf{x}_i \simeq \mathbf{U}\mathbf{U}^\top \mathbf{x}_i, \text{ with } \mathbf{U} \in \mathrm{St}(p,k) \triangleq \{\mathbf{U} \in \mathbb{R}^{p\times k} \mid \mathbf{U}^\top\mathbf{U} = \mathbf{I}\}$$

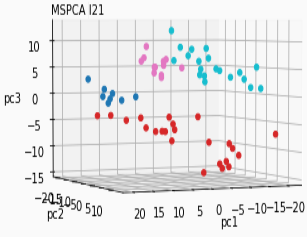


Can we find the best $\mathbf{U}$?

Khan data: $n = 2308$ genes expression of tumors of $n = 63$ patients, with 4 cancer classes







Khan et al., "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks," Nature medicine, 2001

Can we define the best $\mathbf{U}$?

$$\underset{\mathbf{U} \in \mathrm{St}(p,k)}{\text{minimize}} \quad f(\mathbf{U})$$

- **Design** the objective function f
- **Solve** the constrained minimization problem
- **Analyze** the expected performance, convergence speed, etc.
- **Apply** the result to some data

In this talk

✓ **Design**

- Generalize fitting costs from PCA to robust variants
- Motivating sparse PCA and sparsity promoting penalties

✓ **Solve**

- Quick panorama of optimization methods for $\mathrm{St}(p, k)$

✗ **Analyze**

✗ **Apply**

✓ **Try something else**

- False discovery rate driven variable selection in SPCA

How to design a new “PCA”

Principal component analysis (PCA)

"**Vanilla**" **PCA** of **rank** k

- Singular value decomposition (**SVD**) of the data matrix $\mathbf{x} = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \mathbb{R}^{p \times n}$

$$\mathbf{X} \overset{\text{SVD}}{=} [\mathbf{U}|\mathbf{U}_\perp]\,\mathbf{D}\mathbf{V}^\top$$

- **Loading vectors** $\mathbf{U} \in \mathrm{St}(p, k)$
- **Principal components** $\mathbf{x}_i^k = \mathbf{U}^\top \mathbf{x}_i \in \mathbb{R}^k$
- **Projected data** $\tilde{\mathbf{x}}_i = \mathbf{U}\mathbf{x}_i^k = \mathbf{U}\mathbf{U}^\top \mathbf{x}_i$

Beyond “vanilla” PCA

Generalizing PCA?

- Find a **model** or **framework** that yields PCA as a solution

$$\underset{\mathbf{U}\in\mathrm{St}(p,k)}{\text{minimize}} \quad f(\mathbf{U})$$

- Leverage corresponding **tools** and **extensions**

$$\underset{\mathbf{U}\in\mathrm{St}(p,k)}{\text{minimize}} \quad \mathfrak{f}(\mathbf{U})$$

A major motivation: being robust to **heavy-tailed** distributions and **outliers**

Probabilistic and Bayesian PCA

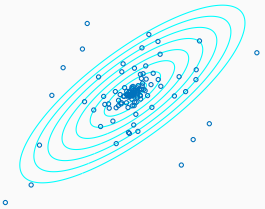
- **Signal** plus **noise** model

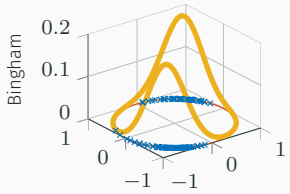
$$\mathbf{x} \overset{d}{=} \mathbf{U}\mathbf{s} + \mathbf{n}$$

- **Probabilistic PCA** "MLE of $\mathbf{U}$ defines f"

$$\mathbf{s} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_s) \quad \text{and} \quad \mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$$
$$\implies \quad \mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{U}\mathbf{\Sigma}_s\mathbf{U}^\top + \sigma^2\mathbf{I})$$

- **Extensions** by **changing assumptions**
  - **Elliptical model** for $\mathbf{x}$, $\mathbf{s}$, or $\mathbf{n}$
  - **Prior distribution** on $\mathbf{U}$ directional statistics





Geometric PCA

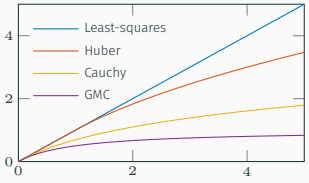
- **Euclidean distance** from $\mathbf{x}$ to $\mathrm{span}(\mathbf{U})$

$$\mathrm{dist}(\mathbf{U}, \mathbf{x}) = \sqrt{\mathbf{x}^\top \mathbf{x} - \mathbf{x}^\top \mathbf{U}\mathbf{U}^\top \mathbf{x}}$$

- **Geometric PCA**

$$\underset{\mathbf{U} \in \mathrm{St}(p,k)}{\text{minimize}} \quad \sum_{i=1}^{n} \mathrm{dist}^2(\mathbf{U}, \mathbf{x}_i)$$

- **Extensions** using **alternate distances**
  - Robust costs: $f(\mathbf{U}) = \sum_{i=1}^{n} \rho(\mathrm{dist}^2(\mathbf{U}, \mathbf{x}_i))$



Algebraic PCA

- **Low-rank approximation**

$$\begin{aligned} \underset{\mathbf{Y}}{\text{minimize}} \quad & ||\mathbf{X} - \mathbf{Y}||_F^2 \\ \text{subject to} \quad & \text{rank}(\mathbf{Y}) = k \end{aligned}$$

- **Extensions** using **alternate decompositions/structures**
  - Low-rank plus sparse recovery (*Robust PCA*)
  - Matrix completion (missing entries)
  - Non-negative matrix factorization
  - Additional structure in the principal components $\rightarrow$ SPCA?

Sparse PCA

Why Sparse PCA?

Recalling

- **Loading vectors** $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_k] \in \mathrm{St}(p, k)$
- **Principal components**

$$[\tilde{\mathbf{x}}_i]_j = \mathbf{u}_j^\top \mathbf{z}_i \in \mathbb{R}$$

- The j^{th}-PC is a **linear combination** of initial variables **weighted by the loading vector**

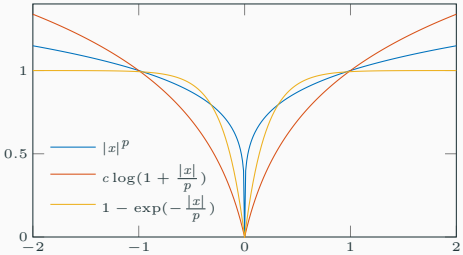
Loading vectors should be sparse

- Improved PC's **interpretation**
- Double duty: **dimension reduction & variable selection**

Sparse PCA from the optimization point of view

$$\underset{\mathbf{U}\in\mathrm{St}(p,k)}{\text{minimize}} \; f(\{\mathbf{x}_i\}_{i=1}^n,\, \mathbf{U}) + \lambda h(\mathbf{U})$$

- f is a **fitting cost** cf. many examples
- h is a **sparsity-promoting penalty** mimics ℓ_1- or $\ell_{2,1}$-norm



Entry-wise sparse penalty h

Optimization on $\mathrm{St}(p, k)$

Existing solutions

- **Solution #1**: Riemannian optimization on $\mathrm{St}(p, k)$

BOU23 Boumal, "An introduction to optimization on smooth manifolds," Cambridge University Press, 2023

ABS09 Absil, Mahony, Sepulchre, "Optimization algorithms on matrix manifolds," Princeton Univ. Press, 2009

EDE98 Edelman, Arias, Smith, "The geometry of algorithms with orthogonality constraints," SIMAX, 1998

- **Solution #2**: Majorization-Minimization ticks

KIE02 Kiers, "Setting up alternating least squares and iterative majorization algorithms for solving various matrix optimization problems," Computational statistics & data analysis, 2002

BRE21 Breloy et al. "MM on the Stiefel Manifold With Application to Robust Sparse PCA", IEEE TSP, 2021

- **Solution #3**: ADMM splitting tricks

BRE23 Brehier et al. "Robust and globally sparse PCA via MM and variable splitting," ICASSP, 2023

UEM19 Uematsu, Fan, Chen, Lv, Lin, "SOFAR: Large-Scale Association Network Learning," IEEE Trans. on IT, 2019

#1 Riemannian optimization

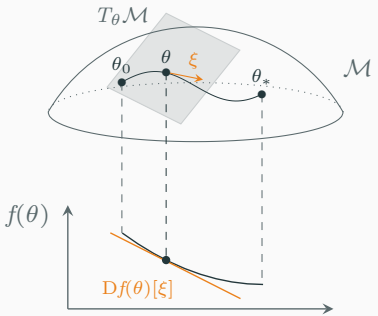
Tools from differential geometry

- Manifold $\mathcal{M}$, tangent spaces $T_\theta\mathcal{M}$
- Metric $\langle\cdot,\cdot\rangle_\theta$
- Retraction $\mathrm{R}_\theta : T_\theta\mathcal{M} \to \mathcal{M}$

Riemannian gradient descent

$$\langle \mathrm{grad}_{\mathcal{M}} f(\theta), \xi \rangle_\theta = \mathrm{D}f(\theta)[\xi]$$

$$\theta_{i+1} = R_{\theta_i}(-t_i \mathrm{grad}_{\mathcal{M}} f(\theta_i))$$



Pros: elegant, generic and flexible framework, links with information geometry

Cons: ca be tedious and computationally expensive, no convincing proximal method (yet?)

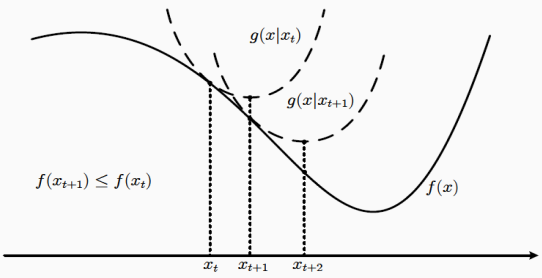
#2 Majorization-Minimization

Majorize f at $\mathbf{x}_t$ by a surrogate g

$$f(\mathbf{x}) \leq g(\mathbf{x}|\mathbf{x}_t)$$

Minimize g to iterate

$$\mathbf{x}_{t+1} = \underset{\mathbf{x}\in\mathcal{X}}{\operatorname{argmin}} \; g(\mathbf{x}|\mathbf{x}_t)$$



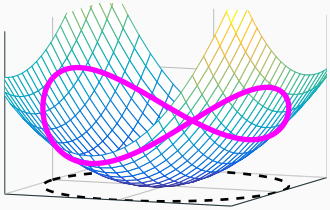
Pros: super fast (in time), covers a large class of functions

Cons: yet not universal, has to be tailored (or proxy the objective)

#2 Majorization-Minimization framework for $\mathrm{St}(p, k)$

Proposal in BRE21

- A systematic trick to deal with orthonormality
- A catalog of surrogates for usual costs
- Applications to probabilistic and sparse PCA

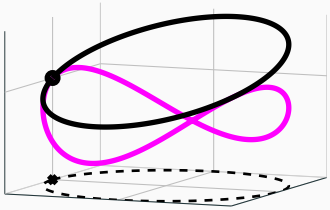


Majorize f on $\mathrm{St}(p, k)$ by a linear surrogate

$$g\left(\mathbf{U}|\mathbf{U}^t\right) = -2\mathfrak{Re}\left\{\mathrm{Tr}\{\mathbf{R}_{\mathbf{U}^t}^H\mathbf{U}\}\right\} + \mathrm{const.}$$

Minimize g on $\mathrm{St}(p, k) \Leftrightarrow$ Projection

$$\mathbf{U}^{t+1} = \mathcal{P}_{\mathrm{St}}\left\{\mathbf{R}_{\mathbf{U}^t}^H\right\}$$



#3 ($\mathcal{M}$)-ADMM and distributed algorithms

Variable splitting

$$\begin{aligned} \underset{\mathbf{U},\mathbf{V}}{\text{minimize}} \quad & f_{\mathrm{u}}(\mathbf{U}) + f_{\mathrm{v}}(\mathbf{V}) \\ \text{subject to} \quad & \mathbf{U} \in \mathrm{St}(p,k) \\ & \mathbf{U} = \mathbf{V} \end{aligned}$$

Cyclic updates over $\{\mathbf{U},\ \mathbf{V},\ \mathbf{\Gamma}\}$ on **augmented Lagrangian** problem

$$\underset{\{\mathbf{U}_i\}\in\mathrm{St}(p,k),\mathbf{V}}{\text{minimize}} \quad f_{\mathrm{u}}(\mathbf{U}) + f_{\mathrm{v}}(\mathbf{V}) + \gamma||\mathbf{U}-\mathbf{V}||_F^2 + \langle \mathbf{\Gamma}, \mathbf{U}-\mathbf{V}\rangle$$

Pros: generalization to distributed setting, splits problems into simple ones

Cons: relaxation trick, still requires to handle $\mathbf{U}$ with previous methods

Some experiments

Optimization benchmark

Robust subspace recovery

$$\underset{\mathbf{U}\in\mathrm{St}(p,k)}{\text{minimize}} \quad \sum_{i=1}^{n} \rho_H(\mathrm{dist}^2(\mathbf{U}, \mathbf{x}_i))$$

with **Huber loss**

$$\rho_{\mathrm{H}}(t) = \begin{cases} t/\sqrt{T} & \text{if } \leq T \\ 2\sqrt{t} - \sqrt{T} & \text{if } t > T \end{cases}$$

$\rightarrow$ **compare iterative algorithms**

Different algorithms (and computational bottlenecks)

LER17 Quadratic MM, data matrix version — rank-k SVD($p \times n$)

$$\mathbf{U}^{t+1} = \mathcal{P}_k\{\tilde{\mathbf{Z}}_t\}, \text{ with } [\tilde{\mathbf{Z}}_t]_{:,i} = \sqrt{\rho'\left(\mathrm{dist}^2\left(\mathbf{U}^t, \mathbf{z}_i\right)\right)}\mathbf{z}_i$$

MAR05 Fixed point heuristic, covariance matrix version — rank-k SVD($p \times p$)

$$\mathbf{U}^{t+1} = \mathcal{P}_k\left\{\mathbf{M}\left(\mathbf{U}^t\right)\right\}, \text{ with } \mathbf{M}\left(\mathbf{U}^t\right) = \tilde{\mathbf{Z}}_t\tilde{\mathbf{Z}}_t^H$$

MAN02 Steepest descent on Stiefel — $\times$thin-SVDs($p \times k$)

$$\mathbf{U}^{t+1} = \mathcal{P}_{\mathrm{St}}\left\{\mathbf{U}^t + \gamma\, \nabla_f(\mathbf{U}^t)\right\}, \text{ with the right } \gamma$$

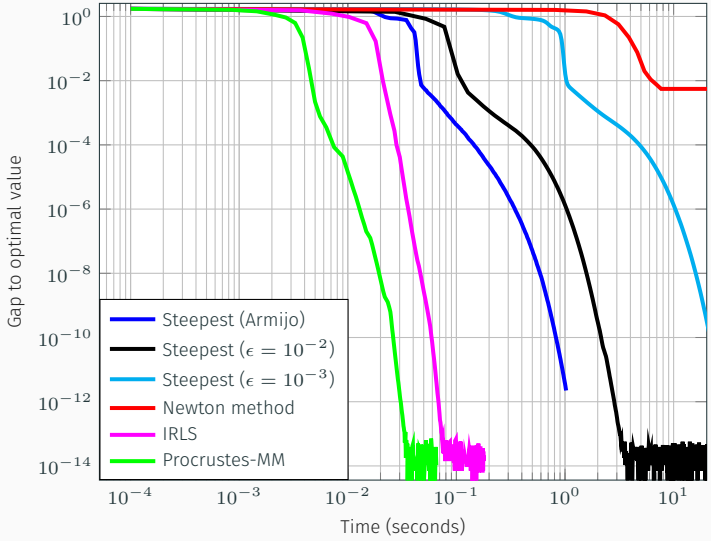
MAN02 Newton method on Stiefel — $(p \times k)^2$ system

$$\mathbf{U}^{t+1} = \mathcal{P}_{\mathrm{St}}\{\mathbf{U}^t + \mathbf{Y}\}, \text{ with } \mathbf{Y} = \mathrm{cpoint}(\mathbf{U}^t, \nabla_f(\mathbf{U}^t), \mathbf{H}_f(\mathbf{U}^t))$$

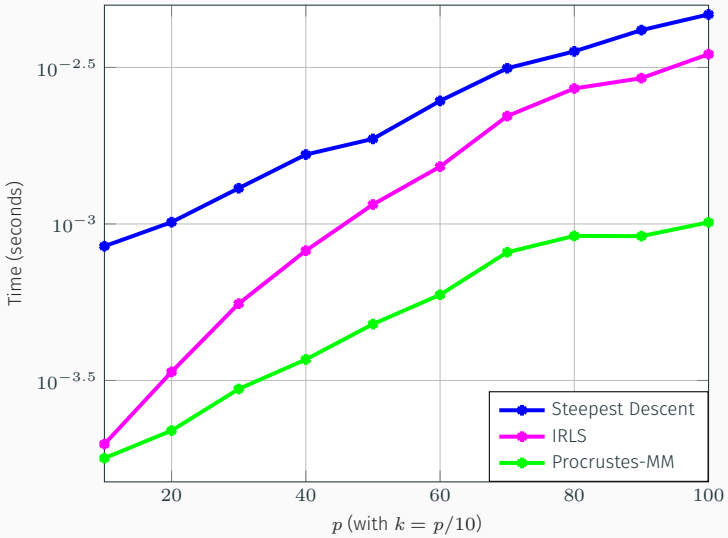
DIN06 Procrustes-MM — thin-SVD($p \times k$)

$$\mathbf{U}^{t+1} = \mathcal{P}_{\mathrm{St}}\left\{\mathbf{M}\left(\mathbf{U}^t\right)\mathbf{U}^t\right\}$$

Objective value (-optimal value) versus CPU time ($p = 30$, $k = 5$, $n = 100$)



Average CPU time of an iteration versus size and rank



Robustness and sparsity

$$\underset{\mathbf{U}\in\mathrm{St}(p,k)}{\text{minimize}} \quad \sum_{i=1}^{n} \rho(\mathrm{dist}^2(\mathbf{U}, \mathbf{x}_i)) + \lambda h(\mathbf{U})$$

- **Fitting**
  - **Least-squares** $\rho(x) = x$
  - **Huber Loss** $\rho_H(x)$
- **Penalty**
  - "**PCA**" $\lambda = 0$
  - **SPCA** $\lambda \neq 0$ + h is smooth-ℓ_1

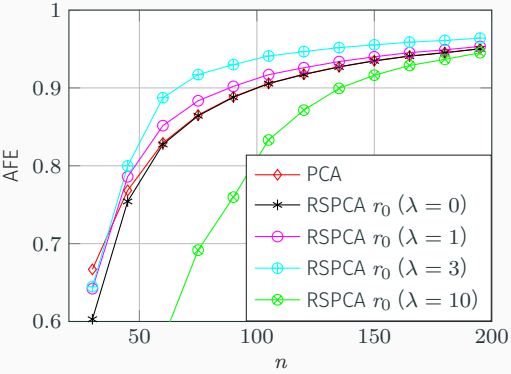
Subspace recovery

Sampling

$$
\begin{aligned}
\{\mathbf{x}_i\}_{i=1}^{n} &= \{\{\mathbf{x}_i^{\text{in}}\}_{i=1}^{n_{\text{in}}}, \{\mathbf{x}_i^{\text{out}}\}_{i=n_{\text{in}}+1}^{n}\} \\
\mathbf{x}^{\text{in}} &\sim \mathcal{CN}(\mathbf{0}, \text{SNR} \times \mathbf{U}\mathbf{U}^H + \mathbf{I}) \\
\mathbf{x}^{\text{out}} &\sim \mathcal{CN}(\mathbf{0}, \text{ONR} \times \mathbf{U}_\perp \mathbf{U}_\perp^H + \mathbf{I})
\end{aligned}
$$

Metric

$$
\text{AFE} = \mathbb{E}\left[\text{Tr}\{\hat{\mathbf{U}}^H \mathbf{U}\mathbf{U}^H \hat{\mathbf{U}}\}/k\right]
$$



$p = 100$, $k = 10$, $\text{SNR} = 10$, *dense loadings*

Subspace recovery

Sampling

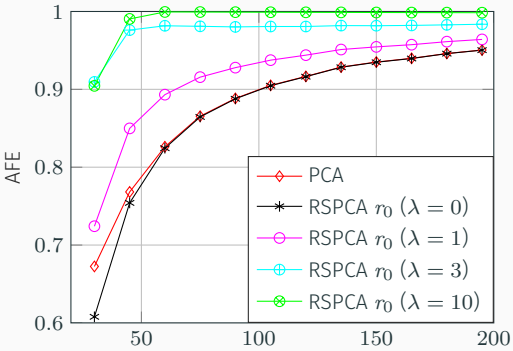
$$\{\mathbf{x}_i\}_{i=1}^{n} = \{\{\mathbf{x}_i^{\text{in}}\}_{i=1}^{n_{\text{in}}}, \{\mathbf{x}_i^{\text{out}}\}_{i=n_{\text{in}}+1}^{n}\}$$

$$\mathbf{x}^{\text{in}} \sim \mathcal{CN}(\mathbf{0}, \text{SNR} \times \mathbf{U}\mathbf{U}^H + \mathbf{I})$$

$$\mathbf{x}^{\text{out}} \sim \mathcal{CN}(\mathbf{0}, \text{ONR} \times \mathbf{U}_\perp \mathbf{U}_\perp^H + \mathbf{I})$$

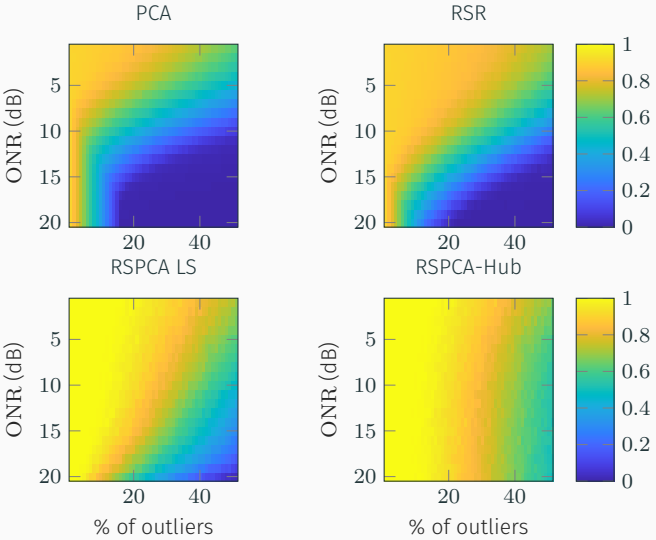
Metric

$$\text{AFE} = \mathbb{E}\left[\text{Tr}\{\hat{\mathbf{U}}^H \mathbf{U}\mathbf{U}^H \hat{\mathbf{U}}\}/k\right]$$



$n, p = 100$, $k = 10$, SNR $= 10$, *sparse loadings*

Robustness to outliers

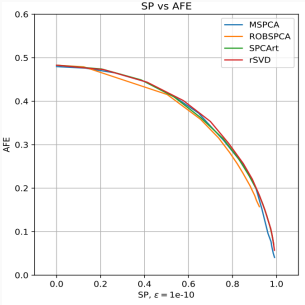


FDR controlled SPCA

Sparse PCA: EV vs SP trade-off

Limitation

- SPCA is only driven by **explained variance** (EV)
- Not always relevant for variable selection
- Might capture high EV **outliers**



New point of view in this work

- Links with **SPCA**
- Focuses on **variable selection** step
- Reformulates the optimization with promising criterion

FDR = “% of allowed false variable selection”

Back to the roots of Sparse PCA

Inspiration:

Zou, Hastie, Tibshirani, "Sparse principal component analysis," J. of computational and graphical statistics, 2006

2-step **SPCA** given **pre-computed plug-in PCs** $\tilde{\mathbf{X}}$

$$\underset{\mathbf{U}\in \mathrm{St}(p,k)}{\text{minimize}} \quad ||\tilde{\mathbf{X}} - \mathbf{U}^\top \mathbf{X}||_F^2 + \lambda ||\mathbf{U}||_1$$

Relaxation of orthogonality constraint $\longrightarrow$ series of k **elastic-net** problems

$$\underset{\boldsymbol{\beta}_j}{\text{minimize}} \quad ||[\tilde{\mathbf{X}}]_{j,:}^\top - \mathbf{X}^\top \boldsymbol{\beta}_j||_F^2 + \lambda ||\boldsymbol{\beta}_j||^2 + \lambda_1 ||\boldsymbol{\beta}_j||_1$$

Sparse loading vectors $\mathbf{U}_{\mathrm{SPCA}} = [\boldsymbol{\beta}_1, \ \ldots, \ \boldsymbol{\beta}_k]$

T-Rex PCA: motivations

SPCA boils down to solve k problems

$$\underset{\boldsymbol{\beta}_j}{\text{minimize}} \quad ||[\tilde{\mathbf{X}}]_{j,:}^\top - \mathbf{X}^\top \boldsymbol{\beta}_j||_F^2 + \lambda||\boldsymbol{\beta}_j||^2 + \lambda_1||\boldsymbol{\beta}_j||_1$$

and **tune** λ_1 to achieve a **good SP-vs-EV**

Can we change the paradigm?

- move away from this trade-off
- have guarantees on false-discovery-rate
- be more computationally efficient

We are going to ask all of this to the **T-Rex** 🦖

False Discovery Rate (FDR) and True Positive Rate (TPR)

- # samples: n
- # candidate variables: $p > n$
- Active variables: $\mathcal{A} \subseteq \{1, \dots, p\}$ $\leftarrow$ the "true" support of our loadings
- Selected variables: $\widehat{\mathcal{A}} \subseteq \{1, \dots, p\}$

$$\mathrm{FDR} \coloneqq \mathbb{E}\big[\,\mathrm{FDP}\,\big] \coloneqq \mathbb{E}\left[\frac{|\widehat{\mathcal{A}} \backslash \mathcal{A}|}{1 \vee |\widehat{\mathcal{A}}|}\right] \qquad \text{and} \qquad \mathrm{TPR} \coloneqq \mathbb{E}\big[\,\mathrm{TPP}\,\big] \coloneqq \mathbb{E}\left[\frac{|\mathcal{A} \cap \widehat{\mathcal{A}}|}{1 \vee |\mathcal{A}|}\right]$$

Goal

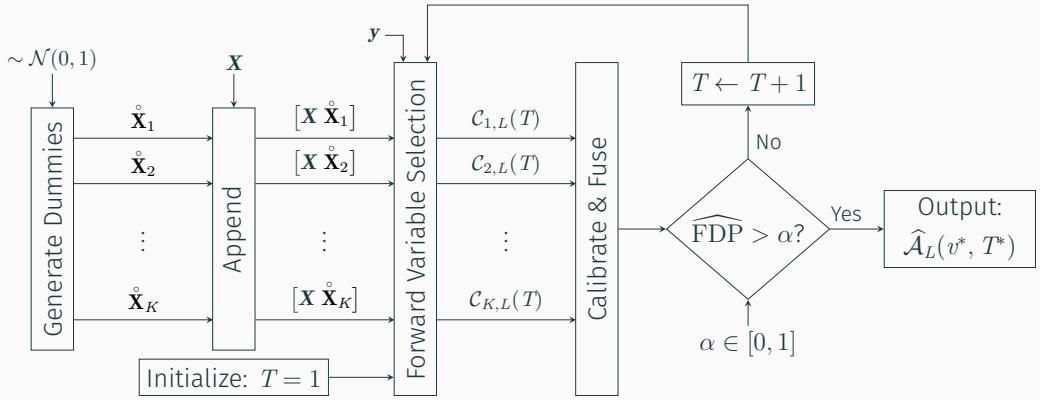
$$\max_{\widehat{\mathcal{A}}} \mathrm{TPR} \quad \text{s.t.} \quad \mathrm{FDR} \leq \alpha,$$

where $\alpha \in [0, 1]$ is the user-defined target FDR level.

Terminating-Random Experiments (T-Rex): the ingredients

- **LARSalgorithm**
  - Solves **LASSO** or **Elastic-Net** **iteratively**
  - Produces **a sequence of selected variables**
- **Terminated random experiment**
  - Append L dummies to $\mathbf{X} := [\mathbf{X}; \mathring{\mathbf{X}}]$ usually $\sim \mathcal{N}(0, 1)$
  - Apply **LARS** and stop when T **dummies are selected**
- **T-Rex selector**
  - Run **term. rand. exp. in parallel** and **vote for** $\hat{\mathcal{A}}$ threshold $v \in (0.5, 1]$
  - **FDP can be estimated** at this step
  - **Calibrate** T and v to **maximize TPP** and **achieve desired FDR target**

T-Rex Selector



- $L \in \mathbb{N}_+$ generated dummies
- $T \in \{1, \dots, L\}$ included dummies before stopping
- $v \in [0.5, 1)$ voting level
- $\Phi_{T,L}(j)$ relative occurrence of variable j
- $\widehat{\mathcal{A}}_L(v, T) \coloneqq \{j : \Phi_{T,L}(j) > v\}$

Back to Sparse PCA

Inspiration:

Zou, Hastie, Tibshirani, "Sparse principal component analysis," J. of computational and graphical statistics, 2006

2-step **SPCA** given **pre-computed plug-in PCs** $\tilde{\mathbf{X}}$

$$\underset{\mathbf{U}\in \mathrm{St}(p,k)}{\text{minimize}} \quad ||\tilde{\mathbf{X}} - \mathbf{U}^\top \mathbf{X}||_F^2 + \lambda||\mathbf{U}||_1$$

Relaxation of orthogonality constraint $\longrightarrow$ series of k **elastic-net** problems

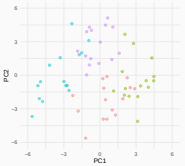
$$\underset{\boldsymbol{\beta}_j}{\text{minimize}} \quad ||[\tilde{\mathbf{X}}]_{j,:}^\top - \mathbf{X}^\top \boldsymbol{\beta}_j||_F^2 + \lambda||\boldsymbol{\beta}_j||^2 + \lambda_1||\boldsymbol{\beta}_j||_1 \quad \rightarrow$$

Sparse loading vectors $\mathbf{U}_{\mathrm{SPCA}} = [\boldsymbol{\beta}_1, \ \ldots, \ \boldsymbol{\beta}_k]$

Cleaning PCA with the T-Rex Selector

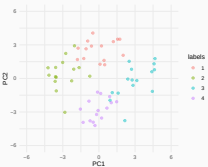
Setup (see setup in R pack. scPCA): 4 Classes, $n = 60$, $p = 150$, $p_1 = 10$.

Standard PCA



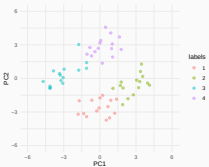
(a) $\mathrm{FDP} = 1$, $\mathrm{TPP} = 1$, #Var $= 150$.

T-Rex, tFDR: $= 0.5 \ / \ 2$



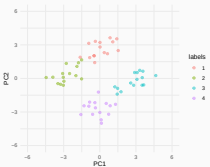
(b)

T-Rex, tFDR: $= 0.4 \ / \ 2$



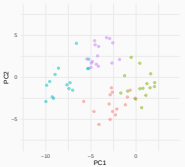
(c)

T-Rex, tFDR: $= 0.3 \ / \ 2$



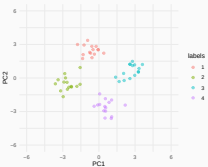
(d)

Sparse PCA



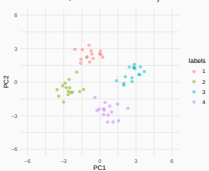
(h) $\mathrm{FDP} \approx 0.9$, $\mathrm{TPP} = 0.5$, #Var $= 48$.

Oracle PCA



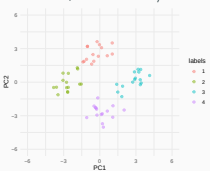
(g) $\mathrm{FDP} = 0$, $\mathrm{TPP} = 1$, #Var $= 10$.

T-Rex, tFDR: $= 0.1 \ / \ 2$



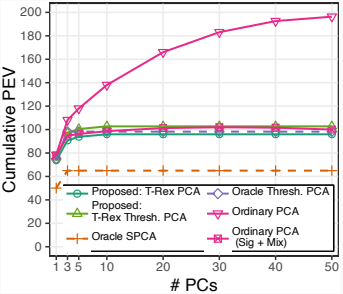
(f) $\mathrm{FDP} \approx 0.09$, $\mathrm{TPP} = 1$, #Var $= 11$.

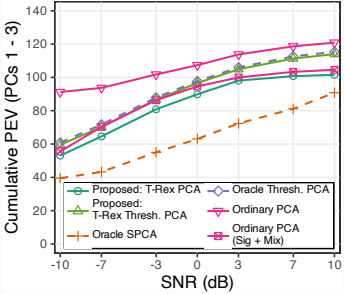
T-Rex, tFDR: $= 0.2 \ / \ 2$

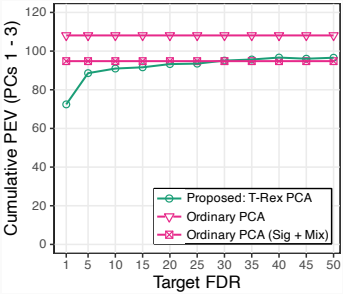


(e)

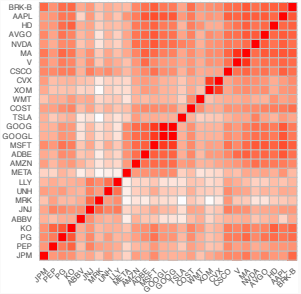
More experiments



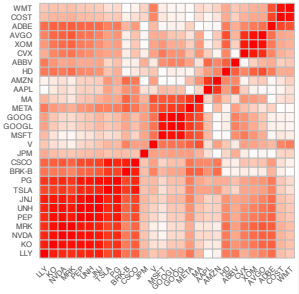




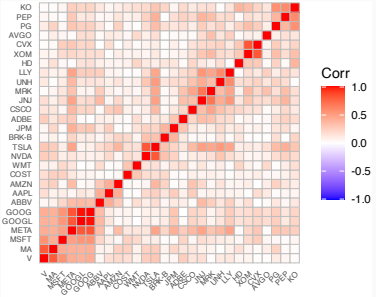
Experiments on the 28 most influential S&P 500 stocks



Raw covariance matrix



T-Rex SPCs removed



PCs removed

Covered so far

✓ **Design**

- Generalize fitting costs from PCA to robust variants
- Motivating sparse PCA and sparsity promoting penalties

✓ **Solve**

- Quick panorama of optimization methods for $\mathrm{St}(p, k)$

✗ **Analyze**

✗ **Apply**

✓ **Try something else**

- False discovery rate driven variable selection in SPCA

Thanks!

Papers in this talk

Machkour, J., Breloy, A., Muma, M., Palomar, D. P., & Pascal, F. (2024, April). Sparse PCA with false discovery rate controlled variable selection. In ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 9716-9720). IEEE.

Brehier, H., Breloy, A., El Korso, M. N., & Kumar, S. (2023, June). Robust and globally sparse PCA via majorization-minimization and variable splitting. In ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 1-5). IEEE.

Collas, A., Bouchard, F., Breloy, A., Ginolhac, G., Ren, C., & Ovarlez, J. P. (2021). Probabilistic PCA from heteroscedastic signals: geometric framework and application to clustering. IEEE Transactions on Signal Processing, 69, 6546-6560.

Breloy, A., Kumar, S., Sun, Y., & Palomar, D. P. (2021). Majorization-minimization on the Stiefel manifold with application to robust sparse PCA. IEEE Transactions on Signal Processing, 69, 1507-1520.

T-Rex demo page `https://cran.r-project.org/web/packages/TRexSelector/vignettes/TRexSelector_usage_and_simulations.html`