Iterative Supervised Principal Components

Juho Piironen **Aki Vehtari**
Helsinki Institute for Information Technology, HIIT
Department of Computer Science, Aalto University, Finland

Aalto University
School of Science

Introduction

- Supervised learning in high dimensions: model target variable y with predictors $\mathbf{x} = (x_1, \dots, x_D)$, where D is large (often $n \ll D$)
- We want to reduce the computational burden via dimensionality reduction: find a new set of features $\mathbf{z} = (z_1, \dots, z_K)$, where $K \ll D$, and use these for learning
- Standard principal components (PCA) not necessarily good
- Better method is supervised PCA (SPCA): screen only those features with univariate score (correlation) with y above some threshold, and perform PCA for those features
- We propose an iterative version of this method: iterative supervised PCA (ISPCA)

Algorithm

- Iterate the following steps K times:

1. Compute the univariate scores $s_j = S(\mathbf{x}_j, \mathbf{y})$ for each feature $\mathbf{x}_j$.
2. Retain only features with score $s_j > \gamma$, and compute the first principal component $\mathbf{v}_\gamma$ of these features $\mathbf{X}_\gamma$. Choose γ so that the projection of $\mathbf{X}_\gamma$ onto this vector $\mathbf{z}_\gamma = \mathbf{X}_\gamma \mathbf{v}_\gamma$ maximises the score $S(\mathbf{z}_\gamma, \mathbf{y})$. Denote the extracted feature by $\mathbf{z}$.
3. Subtract the variation explained by $\mathbf{z}$ from each column in $\mathbf{X}$ as $\mathbf{x}'_j = \mathbf{x}_j - b_j \mathbf{z}$ where $b_j = (\mathbf{z}^\mathsf{T}\mathbf{z})^{-1}(\mathbf{x}_j^\mathsf{T}\mathbf{z})$. This yields a modified feature matrix $\mathbf{X}'$.
4. Set $\mathbf{X} \leftarrow \mathbf{X}'$ and go to step 1.

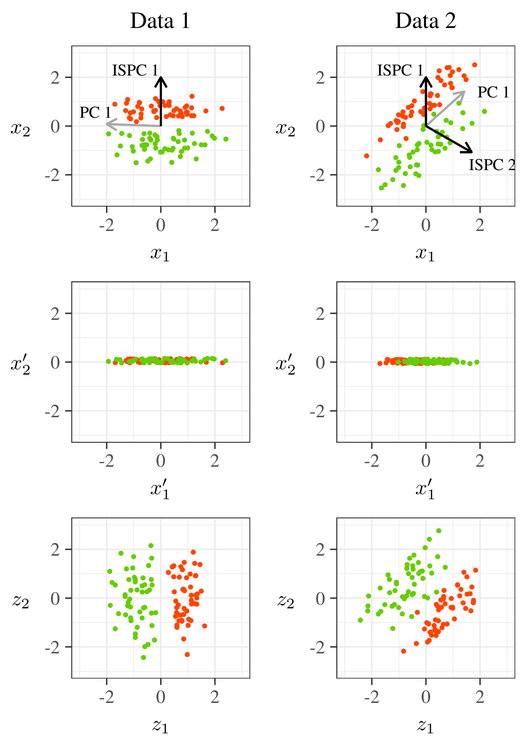
More details

- Like PCA and SPCA, the method can be written as

$$\mathbf{Z} = \mathbf{X}\mathbf{W}$$

  where the new features $\mathbf{Z}$ are orthogonal, but columns of $\mathbf{W}$ are not
- It is possible to combine supervised and unsupervised features
- Number of supervised iterations can be decided based on a permutation test

Toy example

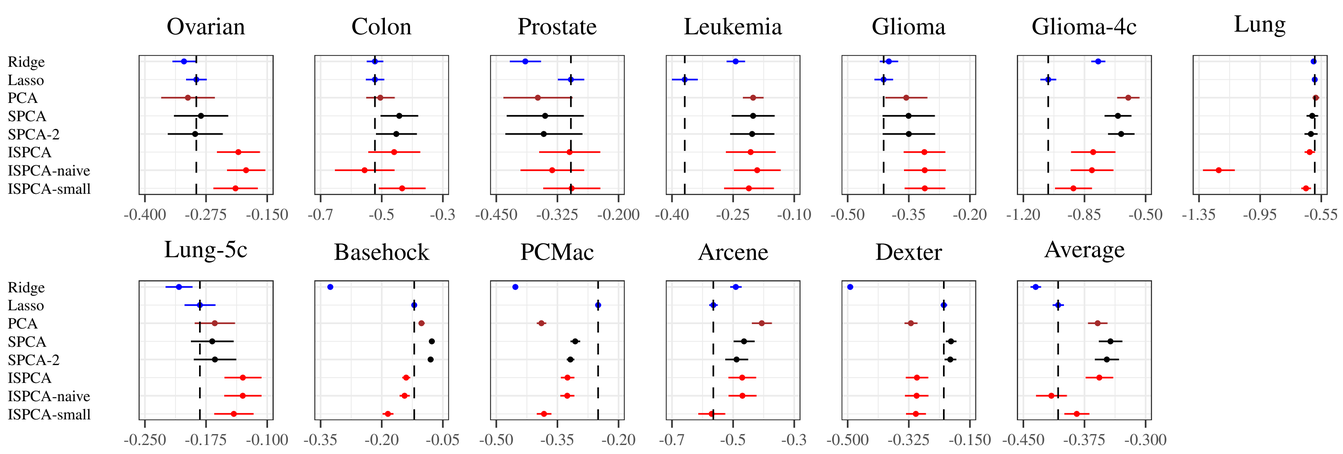


Top row: original dataset, ISPCs found, and the first PC. Middle row: feature matrix obtained after subtracting the variation related to the first ISPC from $\mathbf{X}$. Bottom row: transformed features $\mathbf{Z}$.

Binary classification examples

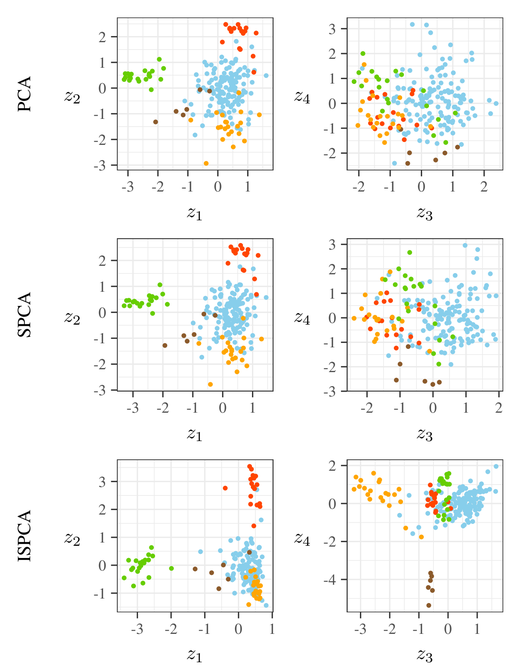
Some of the binary classification datasets visualized using the first two latent features from PCA, SPCA and ISPCA.

Predictive accuracy



Test mean log predictive densities (larger is better) with 95% intervals for Bayesian logistic regression using first 50 latent features from different dimension reduction approaches. Results for ridge and Lasso using original features are shown for comparison (dashed vertical line denotes Lasso). The last plot denotes the average over all the datasets.

Multiclass example



Visualization of Lung-5c cancer data ($n = 203$, $D = 3312$) using the first four latent features from PCA, SPCA and ISPCA.

Datasets

Dataset	Type	Classes	n	D
Ovarian	Gene	2	54	1536
Colon	Gene	2	62	2000
Prostate	Gene	2	102	5966
Leukemia	Gene	2	72	7129
Glioma	Gene	2	85	22283
Glioma-4c	Gene	4	50	4434
Lung	Gene	2	187	19993
Lung-5c	Gene	5	203	3312
Arcene	Other	2	200	10000
Dexter	Text	2	600	20000
Basehock	Text	2	1993	4862
PCMac	Text	2	1943	3289

Computation times

Dataset	Classes	n	D	Computation time			
				PCA	SPCA	ISPCA	Lasso
Leukemia	2	72	7129	9.6 (2%)	8.3 (21%)	8.4 (24%)	1.0
Glioma	2	85	22283	14.6 (5%)	16.6 (33%)	14.5 (28%)	2.7
Lung-5c	5	203	3312	81.0 (1%)	82.2 (12%)	89.0 (19%)	5.2
PCMac	2	1943	3289	511.4 (2%)	303.3 (4%)	565 (22%)	18.9

Average computation time (in seconds) over five repeated runs for a representative set of datasets. For PCA, SPCA and ISPCA, the time contains both the dimension reduction and model fitting (the number in the parenthesis indicating the relative amount of time spent in the dimension reduction), and for Lasso the cross-validation of the regularization parameter.

Conclusions

- ISPCA is useful for visualizing high-dimensional data and for predictive model construction
- None of the dimension reduction techniques is optimal for all the datasets
- The method is implemented in the R-package *dimreduce* (https://github.com/jpiironen/dimreduce)

Contact: juho.piironen@aalto.fi, web page: https://users.aalto.fi/jtpiiron/