High Dimensional Data and Statistical Learning

Lecture 8: PCA and Its Extensions

Weixing Song

Department of Statistics
Kansas State University

Outline

Outline

Principal Component Analysis (PCA) is a technique for examining relationships among a set of variables (generally correlated).

PCA linearly transforms the original set of, say, p variables $\mathbf{x}$ to a new set of p uncorrelated variables, and orders the variables in decreasing order of "importance", determined by the portion of explained total variance in $\mathbf{x}$. The new variables are called the principal components (PC).

The PCs of $\mathbf{x}$ takes the form $\mathbf{v}_k^T\mathbf{x}$, $\|\mathbf{v}_k\| = 1$. Loosely speaking, the PCs are the choices that maximize the variance, subject to being uncorrelated with all the others.

The usual objective of PCA is to see if the first few PCs account for most of the variation in the original data. If they do, then it is argued that the effective dimensionality is less than p.

Suppose a random vector $\mathbf{x}$ has covariance matrix Σ with its eigenvalue-eigenvector pairs $(d_k^2, \mathbf{v}_k)$, where $d_1 \geq \cdots \geq d_p \geq 0$. That is, $\Sigma = \mathbf{V}\mathbf{D}^2\mathbf{V}^T$ and $D = \text{diag}(d_k)$.

The p PCs are given by $z_k = \mathbf{v}_k^T\mathbf{x}$, $k = 1, \ldots, p$, and

$$\text{Var}(z_k) = d_k^2, \qquad \text{Cov}(z_k, z_j) = \delta_{j,k}.$$

With this background in mind, PCA is the action of decomposing a process Z as a superposition of its principal components. The analysis consists of two steps.

- Analysis Step: This step finds the orthonormal eigenvectors $\mathbf{v}_k$ and projects $\mathbf{x}$ onto this basis, i.e.,
$$z_k = \mathbf{v}_k^T\mathbf{x}, \quad \mathbf{z} = \mathbf{V}^T\mathbf{x}.$$
- Synthesis Step: This step reconstructs the process from the principal components using the orthonormal eigenvectors, i.e.,
$$x_j = \sum_{k=1}^{p} v_{jk} z_k, \quad \mathbf{x} = \mathbf{V}\mathbf{z}.$$

PCA finds a linear combination of the original variables such that the derived variables capture maximal variance.

With observed data, the sample PCA can be computed via a SVD of the data matrix.

Let the data $\mathbf{X}$ be a $n \times p$ matrix where n and p are the number of observations and variables, respectively. Assume the column means of $\mathbf{X}$ are zeros, and the SVD of $\mathbf{X}$ be

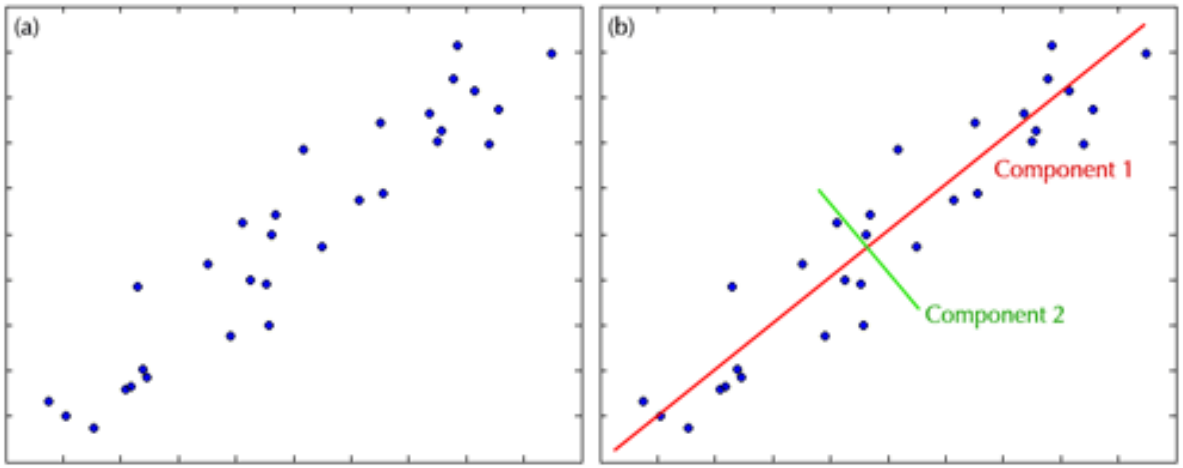
$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T.$$

Then $\mathbf{Z} = \mathbf{U}\mathbf{D}$ are the PCs. The k-th PC accounts for a proportion $d_i^2/\text{trace}(\mathbf{X}^T\mathbf{X})$ of the total variation in the data.

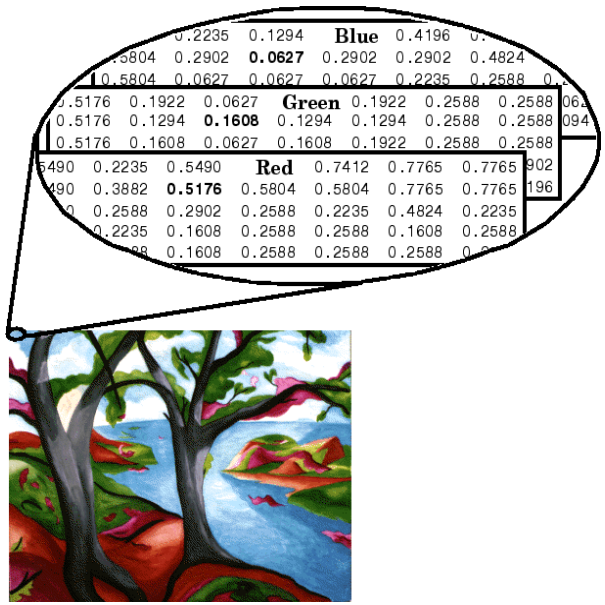
The vectors $\mathbf{v}_k$ (the columns of $\mathbf{V}$) are the principal component directions, or loadings, and they describe the transformation process by which the new variables are created as a linear combination of the old ones.

The vectors $\mathbf{u}_k$ (the columns of $\mathbf{U}$) are the normalized PCs or PC scores.

The singular values d_k indicate the relative importance of the PCs.

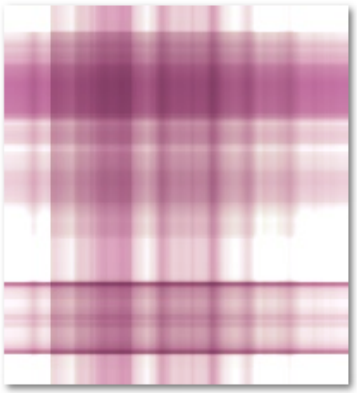


Example (Image Compression): The following KSU-Wildcat picture is stored as a jpeg file which is a $200 \times 200 \times 3$ array in R.

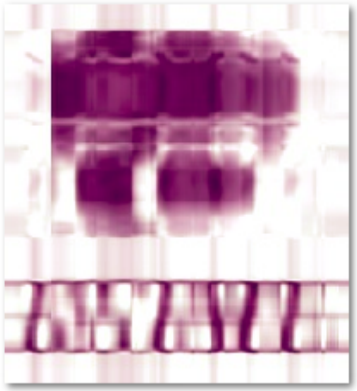


The first, second and third layer contains the intensity of red, green and blue colors, respectively, which are real numbers between 0 and 1. The figure on the right is an illustration.

We can use PCA to compress a figure. For the KSU-Wildcat jpeg file, the following are the compressed versions by taking the number of singular values to be 1, 5, 20 and 50.



No. of SVs: 1
Size: 10.1 KB



No. of SVs: 5
Size: 14.4KB

No. of SVs: 20
Size: 18.2KB

No. of SVs: 50
Size: 18.6KB

The following program is used for the analysis in the example.

```
library(rARPACK);
library(jpeg);

svdeco=function(m, k)
  {
   r=svds(m[, , 1], k);  g=svds(m[, , 2], k); b=svds(m[, , 3], k);
   return(list(r=r, g=g, b=b));
  }

recover0=function(fac, k)
     {
       dmat=diag(k); diag(dmat)=fac$d[1:k];  m=fac$u[,1:k]%*%dmat%*%t(fac$v[, 1:k]);
       m[m<0]=0; m[m>1]=1;
       return(m);
      }

recoverimg=function(lst, k)
 {
   r=recover0(lst$r, k); g=recover0(lst$g, k); b=recover0(lst$b, k);
   m=array(0, c(nrow(r), ncol(r), 3));
   m[, , 1]=r; m[, , 2]=g; m[, , 3]=b;
   return(m);
 }
#rawimg=readJPEG("C://Users//Weixing Song//Dropbox//Teaching//Stat905//Programs//pca_wall.jpg");
#rawimg=readJPEG("C://Users//Weixing Song//Dropbox//Teaching//Stat905//Programs//pca_face.jpg");
rawimg=readJPEG("C://Users//Weixing Song//Dropbox//Teaching//Stat905//Programs//pca_cat.jpg");
lst=svdeco(rawimg,50);
neig=c(1,5,20,50);
for(i in neig)
 {
   m=recoverimg(lst, i);
   writeJPEG(m, sprintf("C://Users//Weixing Song//Desktop//svd_%d.jpg", i), 0.95);
 }
```

The success of PCA is due to the following two optimal properties:

- PCA sequentially captures the maximum variability among the columns of $\mathbf{X}$, guaranteeing minimal information loss.
- PCs are uncorrelated, so we can talk about one PC without referring to others.

However, PCA has drawbacks.

One drawback of PCA is that it is hard to interpret the derived PCs. Each PC is linear combinations of all p variables and the loadings are typically nonzero.

Deriving a "sparse" PCA has been an interesting research problem.

Zou et al. (2004) introduced a sparse PCA (SPCA), which builds on the fact that PCA can be written as a regression-type optimization problem.

Outline

Previous solutions to enforce sparsity includes

- Rotation techniques have been used to interpret the PCs (Jolliffe, 1995).
- Only consider loadings from a small set, such as $\{-1, 0, 1\}$ (Vines, 2000).
- Threshold the loadings (Cadima and Jolliffe, 1995).
- ScoTLASS gets modified PCs with possible zero loadings (Jolliffe, Trendafilov and Uddin, 2003).

Review: Lasso and Elastic Net

Consider a linear regression model with n observations and p predictors. Let $\mathbf{y}$ be the response vector and $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_p)$ be the design matrix.

We assume all the $\mathbf{x}_j$ and $\mathbf{y}$ are centered.

The Lasso criterion:

$$\hat{\beta}_{\text{Lasso}} = \text{argmin}_{\boldsymbol{\beta}} \left\{ \frac{1}{2n} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda \|\boldsymbol{\beta}\|_1 \right\}.$$

The Elastic Net criterion:

$$\hat{\boldsymbol{\beta}}_{\text{eNet}} = \text{argmin}_{\boldsymbol{\beta}} \left\{ \frac{1}{2n} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda \|\boldsymbol{\beta}\|_1 + \lambda_2 \|\boldsymbol{\beta}\|^2 \right\}.$$

The Lars-EN algorithm (Zou and Hastie, 2005) efficiently solves the elastic net problem for all λ_1 with the computational cost of a single least squares fit.

ScoTLASS

The ScoTLASS proposed by Jolliffe, Trendafilov and Uddin (2003) directly imposes an L_1 constraint on PCA.

ScoTLASS successively maximizes $\mathbf{a}_k^T \mathbf{X}^T \mathbf{X} \mathbf{a}_k$ subject to $\mathbf{a}_j^T \mathbf{a}_k = \delta_{j,k}$ and $\|\mathbf{a}_k\|_1 \leq t$ for $k = 1, 2, \ldots, p$.

No guidance of how to choose the t was provided and the computational cost of the optimization problem is high due to nonconvexity.

Motivation: Direct Sparse Approximation

Observe that each PC is a linear combination of the p variables, thus its loadings can be recovered by regressing the PC on the p variables.

Theorem 1

For each i, denote by $\mathbf{z}_i = d_i \mathbf{u}_i$ the i-th principal component. Consider a positive λ and the ridge estimates $\hat{\beta}^{\text{ridge}}$ given by

$$\hat{\beta}^{\text{ridge}} = \text{argmin}_{\beta}\{\|\mathbf{z}_i - \mathbf{X}\beta\|^2 + \lambda\|\beta\|^2\}.$$

Let $\hat{\mathbf{v}} = \hat{\beta}^{\text{ridge}}/\|\hat{\beta}^{\text{ridge}}\|$. Then $\hat{\mathbf{v}} = \mathbf{v}_i$.

This theorem shows the connection between PCA and a regression method. Note that after normalization, the coefficients are independent of λ.

Proof: Note that $\mathbf{X} = \mathbf{U}D\mathbf{V}^T$, $\mathbf{X}^T\mathbf{X} = \mathbf{V}D^2\mathbf{V}^T$, and $\mathbf{z}_i = d_i\mathbf{u}_i = \mathbf{X}\mathbf{v}_i$. We have

$$\begin{aligned}\hat{\beta}^{\text{ridge}} &= (\mathbf{X}^T\mathbf{X} + \lambda I)^{-1}\mathbf{X}^T\mathbf{z}_i = (\mathbf{X}\mathbf{X}^T + \lambda I)^{-1}\mathbf{X}^T\mathbf{X}\mathbf{v}_i \\ &= (\mathbf{V}D^2\mathbf{V}^T + \lambda I)^{-1}\mathbf{V}D^2\mathbf{V}^T\mathbf{v}_i = [d_i^2/(d_i^2 + \lambda)]\mathbf{v}_i.\end{aligned}$$

Now consider adding a L_1 penalty.

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}}[\|\mathbf{z}_i - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda_2\|\boldsymbol{\beta}\|^2 + \lambda_1\|\boldsymbol{\beta}\|_1],$$

where $\|\boldsymbol{\beta}\|_1$ is the L_1 norm of $\boldsymbol{\beta}$.

We call $\hat{\mathbf{v}}_i = \hat{\boldsymbol{\beta}}/\|\hat{\boldsymbol{\beta}}\|$ an approximation to $\mathbf{v}_i$ and $\mathbf{X}\hat{\mathbf{v}}_i$ is the i-th approximated PC. Clearly, a large enough λ_1 gives a sparse $\hat{\boldsymbol{\beta}}$, hence a sparse $\mathbf{v}_i$.

Theorem 1 depends on the results of PCA, so it is not a genuine alternative.

We shall introduce a "self-contained" regression type criterion for deriving principal components.

Theorem 2

For any $\lambda > 0$, let

$$(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}) = \text{argmin}_{\boldsymbol{\alpha},\boldsymbol{\beta}}\{\|\mathbf{X} - \mathbf{X}\boldsymbol{\beta}\boldsymbol{\alpha}^T\|^2 + \lambda\|\boldsymbol{\beta}\|^2\}, \quad \text{subject to } \|\boldsymbol{\alpha}\|^2 = 1.$$

Then $\hat{\boldsymbol{\beta}} \propto \mathbf{v}_1$.

Proof: Note that

$$\begin{aligned} \|\mathbf{X} - \mathbf{X}\boldsymbol{\beta}\boldsymbol{\alpha}^T\|^2 &= \text{trace}\left[(\mathbf{X} - \mathbf{X}\boldsymbol{\beta}\boldsymbol{\alpha}^T)(\mathbf{X} - \mathbf{X}\boldsymbol{\beta}\boldsymbol{\alpha}^T)^T\right] \\ &= \text{trace}(\mathbf{X}^T\mathbf{X}) + \boldsymbol{\beta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} - 2\boldsymbol{\alpha}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\beta}. \end{aligned}$$

So the target function becomes

$$\text{trace}(\mathbf{X}^T\mathbf{X}) + \boldsymbol{\beta}^T(\mathbf{X}^T\mathbf{X} + \lambda I)\boldsymbol{\beta} - 2\boldsymbol{\alpha}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\beta}.$$

For any fixed $\boldsymbol{\alpha}$, $\boldsymbol{\beta}(\boldsymbol{\alpha}) = (\mathbf{X}^T\mathbf{X} + \lambda I)^{-1}\mathbf{X}^T\mathbf{X}\boldsymbol{\alpha}$. Plugging back to the target function, we have

$$\hat{\boldsymbol{\alpha}} = \text{argmin}_{\boldsymbol{\alpha}}\boldsymbol{\alpha}^T\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda I)^{-1}\mathbf{X}^T\mathbf{X}\boldsymbol{\alpha},$$

which leads to

$$\hat{\boldsymbol{\alpha}} = \mathbf{v}_1, \quad \hat{\boldsymbol{\beta}} = \frac{d_1^2}{d_1^2 + \lambda}\mathbf{v}_1.$$

For any matrices A, B, C, if dimensions match, then

$$\text{vec}(ABC) = (C^T \otimes A)\text{vec}(B), \quad (A \otimes B)(C \otimes D) = (AC) \otimes (BC).$$

Use these facts, we can obtain $\boldsymbol{\beta}(\boldsymbol{\alpha})$ as follows:

$$\begin{aligned} & \|\mathbf{X} - \mathbf{X}\boldsymbol{\beta}\boldsymbol{\alpha}^T\|^2 + \lambda\|\boldsymbol{\beta}\|^2 = \|\text{vec}(\mathbf{X} - \mathbf{X}\boldsymbol{\beta}\boldsymbol{\alpha}^T)\|^2 + \lambda\|\boldsymbol{\beta}\|^2 \\ = \quad & \|\text{vec}(\mathbf{X}) - \text{vec}(\mathbf{X}\boldsymbol{\beta}\boldsymbol{\alpha}^T)\|^2 + \lambda\|\boldsymbol{\beta}\|^2 = \|\vec{(}\mathbf{X}) - (\boldsymbol{\alpha} \otimes \mathbf{X})\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|^2. \end{aligned}$$

So,

$$\hat{\boldsymbol{\beta}} = [(\boldsymbol{\alpha}^T \otimes \mathbf{X}^T)(\boldsymbol{\alpha} \otimes \mathbf{X}) + \lambda I]^{-1}(\boldsymbol{\alpha}^T \otimes \mathbf{X}^T)\text{vec}(\mathbf{X}) = (\mathbf{X}^T\mathbf{X} + \lambda I)^{-1}\mathbf{X}^T\mathbf{X}\boldsymbol{\alpha}.$$

The next theorem extends Theorem 2 to derive the whole sequence of PCs.

Theorem 3

Consider the first k PCs. Let $\mathbf{A}_{p\times k} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k)$ and $\mathbf{B}_{p\times k} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_k)$. For any $\lambda > 0$, let

$$(\hat{\mathbf{A}}, \hat{\mathbf{B}}) = \text{argmin}_{\mathbf{A},\mathbf{B}} \left\{ \|\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^T\|^2 + \lambda \sum_{j=1}^{k} \|\boldsymbol{\beta}_j\|^2 \right\},$$

subject to $\mathbf{A}^T\mathbf{A} = I_{k\times k}$. Then $\hat{\boldsymbol{\beta}}_j \propto \mathbf{v}_j$ for $j = 1, \ldots, k$.

Proof: Let

$$C_\lambda(\mathbf{A}, \mathbf{B}) = \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{A}\mathbf{B}^T\mathbf{x}_i\|^2 + \lambda \sum_{j=1}^{k} \|\boldsymbol{\beta}_j\|^2.$$

Since $\mathbf{A}$ is orthonormal, so for $\mathbf{A}_\perp$ be any orthonormal matrix such that $[\mathbf{A}; \mathbf{A}_\perp]$ is $p \times p$ orthonormal, we have

$$\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{A}\mathbf{B}^T\mathbf{x}_i\|^2 = \|\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^T\|^2 = \|\mathbf{X}\mathbf{A}_\perp\|^2 + \|\mathbf{X}\mathbf{A} - \mathbf{X}\mathbf{B}\|^2.$$

Hence, with $\mathbf{A}$ fixed, solving $\text{argmin}_{\mathbf{B}} C_\lambda(\mathbf{A}, \mathbf{B})$ is equivalent to solving the series of ridge regressions

$$\text{argmin}_{\boldsymbol{\beta}_j, j=1,\ldots,k} \sum_{j=1}^{k} \left\{ \|\mathbf{X}\boldsymbol{\alpha}_j - \mathbf{X}\boldsymbol{\beta}_j\|^2 + \lambda \|\boldsymbol{\beta}_j\|^2 \right\}.$$

It is easy to show that

$$\hat{\mathbf{B}} = (\mathbf{X}^T\mathbf{X} + \lambda I)^{-1}\mathbf{X}^T\mathbf{X}\mathbf{A}.$$

Therefore, we have the partially optimized penalized criterion

$$C_\lambda(\mathbf{A}, \hat{\mathbf{B}}) = \|\mathbf{X}\mathbf{A}_\perp\|^2 + \text{trace}((\mathbf{X}\mathbf{A})^T(I - \mathbf{S}_\lambda)(\mathbf{X}\mathbf{A})),$$

where $\mathbf{S}_\lambda = \mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda I)^{-1}\mathbf{X}^T$.

Rearranging the terms, we get

$$C_\lambda(\mathbf{A}, \hat{\mathbf{B}}) = \text{trace}(\mathbf{X}^T\mathbf{X}) - \text{trace}(\mathbf{A}^T\mathbf{X}^T\mathbf{S}_\lambda\mathbf{X}\mathbf{A}),$$

which must be minimized w.r.t. $\mathbf{A}$ with $\mathbf{A}^T\mathbf{A} = I$. Hence $\mathbf{A}$ should be taken to be the largest k eigenvectors $\mathbf{X}^T\mathbf{S}_\lambda\mathbf{X}$. If the SVD of $\mathbf{X}$ is $\mathbf{U}\mathbf{D}\mathbf{V}^T$, it is easy to show that $\mathbf{X}^T\mathbf{S}_\lambda\mathbf{X} = \mathbf{V}\mathbf{D}^2(\mathbf{D}^2 + \lambda I)^{-1}\mathbf{D}^2\mathbf{V}^T$, hence $\hat{\mathbf{A}} = \mathbf{V}[, 1:k]$. Likewise, plugging the SVD of $\mathbf{X}$ into the formula of $\mathbf{B}$, we see that each $\boldsymbol{\beta}_j$ are scaled elements of the corresponding V_j.

Sparse PCA Criterion

We now add an L_1 penalty to produce sparse loadings or regression coefficients, yielding the following optimization problem:

$$(\hat{\mathbf{A}}, \hat{\mathbf{B}}) = \text{argmin}_{\mathbf{A},\mathbf{B}} \left\{ \|\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^T\|^2 + \lambda \sum_{j=1}^{k} \|\boldsymbol{\beta}_j\|^2 + \sum_{j=1}^{k} \lambda_{1,j} \|\boldsymbol{\beta}_j\|_1 \right\},$$

subject to $\mathbf{A}^T\mathbf{A} = I_{k\times k}$.

Computation:

- For fixed $\boldsymbol{\alpha}_j$:

  For each j, let $\mathbf{y}_j^* = \mathbf{X}\boldsymbol{\alpha}_j$. Then $\hat{\mathbf{B}} = (\hat{\boldsymbol{\beta}}_1, \ldots, \hat{\boldsymbol{\beta}}_k)$, where each $\hat{\boldsymbol{\beta}}_j$ is an elastic net estimate

  $$\hat{\boldsymbol{\beta}}_j = \text{argmin}_{\boldsymbol{\beta}_j} \|\mathbf{y}_j^* - \mathbf{X}\boldsymbol{\beta}_j\|^2 + \lambda\|\boldsymbol{\beta}_j\|^2 + \lambda_{1,j}\|\boldsymbol{\beta}_j\|_1.$$

- For fixed $\boldsymbol{\beta}_j$:

  If $\mathbf{B}$ is fixed, we can ignore the penalty part and only minimize $\|\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^T\|$ subject to $\mathbf{A}^T\mathbf{A} = I$. The solution is shown in Theorem 4.

Theorem 4 (Reduced Rank Procrustes Rotation)

Let $\mathbf{M}_{n\times p}$ and $\mathbf{N}_{n\times k}$ be two matrices. Consider the constrained maximization problem

$$\hat{\mathbf{A}} = \text{argmin}_{\mathbf{A}} \|\mathbf{M} - \mathbf{N}\mathbf{A}^T\|^2$$

subject to $\mathbf{A}^T\mathbf{A} = I_{k\times k}$. Suppose the SVD of $\mathbf{M}^T\mathbf{N}$ is $\mathbf{U}D\mathbf{V}^T$, then $\hat{\mathbf{A}} = \mathbf{U}\mathbf{V}^T$.

Proof: Expand the matrix norm

$$\|\mathbf{M} - \mathbf{N}\mathbf{A}^T\|^2 = \text{trace}(\mathbf{M}^T\mathbf{M}) - 2\text{trace}(\mathbf{M}^T\mathbf{N}\mathbf{A}^T) + \text{trace}(\mathbf{A}\mathbf{N}^T\mathbf{N}\mathbf{A}^T).$$

Since $\mathbf{A}^T\mathbf{A} = I$, the last term is equal to $\text{trace}(\mathbf{N}^T\mathbf{N})$, and hence we need to maximize the middle term. With the SVD $\mathbf{M}^T\mathbf{N} = \mathbf{U}\mathbf{D}\mathbf{V}^T$, the middle term becomes

$$\text{trace}(\mathbf{M}^T\mathbf{N}\mathbf{A}^T) = \text{trace}(\mathbf{U}\mathbf{D}\mathbf{V}^T\mathbf{A}^T) = \text{trace}((\mathbf{A}\mathbf{V})^T\mathbf{U}\mathbf{D}).$$

where $(AV)^TAV = I$. Since $\mathbf{D}$ is diagonal, the above is maximized when the diagonal of $(AV)^T\mathbf{U}$ is positive and maximum. By Cauchy-Schwartz inequality, this is achieved when $AV = \mathbf{U}$, in which case the diagonal elements are all 1. Hence $\hat{\mathbf{A}} = \mathbf{U}\mathbf{V}^T$.

SPCA Algorithm:

1. Initialize $\mathbf{A}$ at $\mathbf{V}[:; 1:k]$, the loadings of the first k ordinary PCs.
2. Given a fixed $\mathbf{A} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k)$, solve the following elastic net problem for $j = 1, 2, \ldots, k$

$$\hat{\boldsymbol{\beta}}_j = \text{argmin}_{\boldsymbol{\beta}_j} \|\mathbf{y}_j^* - \mathbf{X}\boldsymbol{\beta}_j\|^2 + \lambda\|\boldsymbol{\beta}_j\|^2 + \lambda_{1,j}\|\boldsymbol{\beta}_j\|_1.$$

3. For fixed $\mathbf{B} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_k)$, compute the SVD of $\mathbf{X}^T\mathbf{X}\mathbf{B} = \mathbf{U}D\mathbf{V}^T$, then update $\mathbf{A} = \mathbf{U}\mathbf{V}^T$.
4. Repeat Steps 2-3 until convergence.
5. Normalize: $\hat{\boldsymbol{\beta}}_j = \boldsymbol{\beta}_j / \|\boldsymbol{\beta}_j\|$, $j = 1, 2, \ldots, k$.

Remarks:

(a). Various methods have been developed for SPCA problem. The method presented here is the first of the kind.

(b). A good SPCA method should possess the following properties:

- Without any sparsity constraint, the method should reduce to PCA.
- It should be computationally efficient for both small p and big p data.
- It should avoid misidentifying the important variables.
- Orthogonality?

Outline

The SVD of a matrix $\mathbf{X}$ is

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T = \sum_{k=1}^{r} \mathbf{X}_k = \sum_{k=1}^{r} d_k \mathbf{u}_k \mathbf{v}_k^T, \quad \mathbf{V}^T\mathbf{V} = \mathbf{U}^T\mathbf{U} = I,$$

where

- $\mathbf{D} = \text{diag}(d_1, \ldots, d_r)$ with singular values $d_1 > \cdots > d_r > 0$.
- $\text{rank}(\mathbf{X}) = r$.
- $\mathbf{X}_k = d_k \mathbf{u}_k \mathbf{v}_k^T$ is the layer-k unit-rank matrix.
- $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_r)$ consists of r orthonormal left singular vectors.
- $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_r)$ consists of r orthonormal right singular vectors.

One method for sparse singular value decomposition: Minimizing

$$\frac{1}{2}\|\mathbf{X} - d\mathbf{u}\mathbf{v}^T\|^2 + \lambda \sum_{i=1}^{p}\sum_{j=1}^{q} w_{ij}|du_i v_j| \tag{1}$$

where

- $d\mathbf{u}\mathbf{v}^T$ is a unit-rank SVD, and $\|\mathbf{u}\| = \|\mathbf{v}\| = 1$.
- $w_{ij} = w^{(d)} w_i^{(u)} w_j^{(v)}$ are data driven weights. Let $\tilde{d}\tilde{\mathbf{u}}\tilde{\mathbf{v}}^T$ be a unit-rank initial estimator, the weights can be chosen as

$$\begin{cases} w^{(d)} = |\tilde{d}|^{-\gamma}, \\ \mathbf{w}^{(u)} = (w_1^{(u)}, \ldots, w_p^{(u)})^T = |\tilde{\mathbf{u}}|^{-\gamma}, \\ \mathbf{w}^{(v)} = (w_1^{(v)}, \ldots, w_p^{(v)})^T = |\tilde{\mathbf{v}}|^{-\gamma}. \end{cases}$$

- $\mu = 2$ in Zou (2006).
- Other layers can be obtained by sequentially fitting previous residuals. See Chen et al. (2012) for details.
- Special case of sparse unit-rank regression (SURR), see Chen et al. (2012).
- Various extensions.

The objective function (1) admits a multiconvex structure. For fixed $\mathbf{u}$, minimization of function (1) w.r.t. $(d, \mathbf{v})$ becomes minimization with respect to $\check{\mathbf{v}} = \text{diag}(d\mathbf{w}^{(v)})\mathbf{v}$ of

$$\frac{1}{2}\|\mathbf{y} - \mathbf{X}^{(v)}\check{\mathbf{v}}\|^2 + \lambda^v \sum_{j=1}^{q} |\check{v}_j|, \tag{2}$$

where

$$\mathbf{y} = \text{vec}(\mathbf{X}), \quad \mathbf{X}^{(v)} = \text{diag}(\mathbf{w}^{(v)})^{-1} \otimes \mathbf{u}, \quad \lambda^{(v)} = \lambda w^{(d)} \left(\sum_{i=1}^{p} w_i^{(u)} |u_i| \right)$$

and $\otimes$ is the Kronecker product. Model (2) can be recognized as a Lasso regression w.r.t. $\check{\mathbf{v}}$. Moreover, not that $\mathbf{X}^{(v)}$ is always an orthogonal matrix; hence the solution of problem (2) is explicit.

In contrast, for fixed $\mathbf{v}$, minimization of function (1) w.r.t. $(d, \mathbf{u})$ becomes minimization w.r.t. $\check{\mathbf{u}} = \text{diag}(d\mathbf{w}^{(u)})\mathbf{u}$ of

$$\frac{1}{2}\|\mathbf{y} - \mathbf{X}^{(u)}\check{\mathbf{u}}\|^2 + \lambda^u \sum_{i=1}^{p} |\check{u}_i|, \tag{3}$$

where

$$\mathbf{X}^{(u)} = \mathbf{v} \otimes \text{diag}(\mathbf{w}^{(u)})^{-1}, \quad \lambda^{(u)} = \lambda w^{(d)} \left(\sum_{j=1}^{q} w_j^{(v)} |v_j| \right).$$

Again, this is a lasso regression problem with respect to $\check{\mathbf{u}}$.

The following are the steps of the numerical sparse unit rank regression algorithm for a fixed λ.

(a). Choose a non-zero initial value for $\hat{\mathbf{u}}$.

(b). Given $\mathbf{u} = \hat{\mathbf{u}}$, minimize function (2) to obtain $\check{\mathbf{v}}$. Let

$$\hat{d} = \|\text{diag}(\mathbf{w}^{(v)})^{-1}\check{\mathbf{v}}\|, \quad \check{\mathbf{v}} = \text{diag}(\hat{d}\mathbf{w}^{(v)})^{-1}\check{\mathbf{v}}.$$

(c). Given $\mathbf{v} = \hat{\mathbf{v}}$, minimize function (3) to obtain $\check{\mathbf{u}}$. Let

$$\hat{d} = \|\text{diag}(\mathbf{w}^{(u)})^{-1}\check{\mathbf{u}}\|, \quad \check{\mathbf{v}} = \text{diag}(\hat{d}\mathbf{w}^{(u)})^{-1}\check{\mathbf{u}}.$$

(d). Repeat steps (b) and (c), until $\hat{\mathbf{C}} = \hat{d}\hat{\mathbf{u}}\hat{\mathbf{v}}^T$ converges, i.e. $\|\hat{\mathbf{C}}_c - \hat{\mathbf{C}}_p\|_F / \|\hat{\mathbf{C}}_p\|_F < \varepsilon$, where $\hat{\mathbf{C}}_c$ is the current fit, $\hat{\mathbf{C}}_p$ is the previous fit and ε is the level of tolerance, e.g. $\varepsilon = 10^{-6}$.

Application: Microarray Biclustering

Background: High dimensional microarray data analysis.

Goal: Identify sets of genes that are significantly expressed for certain cancer types. See Busygin et al. (2008), Lee et al. (2010).

Data: Expression levels of thousands of genes ($p = 12625$), measured from a few subjects (n=56). The cancer type of each subject is known (Carcinoid[20], Colon[13], Normal[17] and Small Cell[6]).

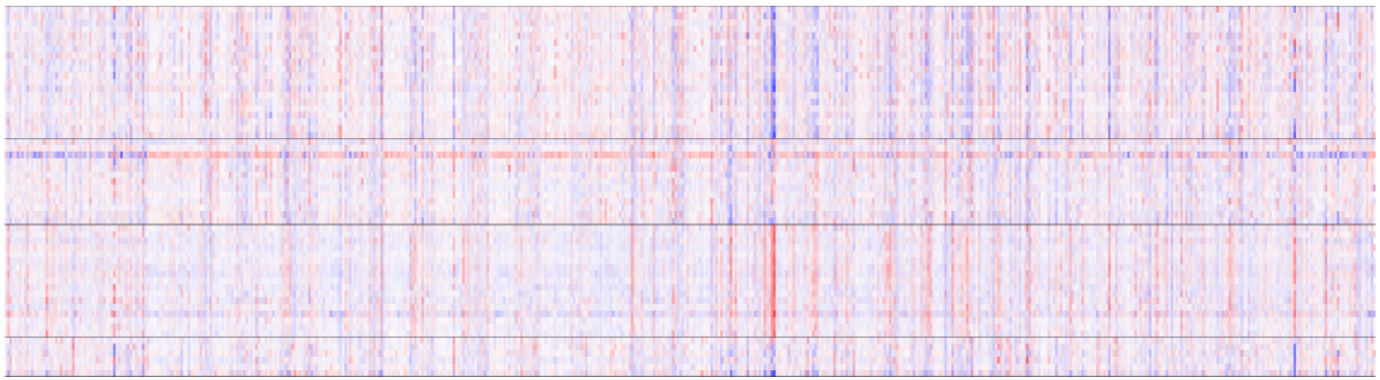


Figure: The original gene expression matrix

Method 1: Unsupervised Learning

- Penalized matrix decomposition (Witten et al. 2009, Lee et al. 2010).
- Sparse SVD (Chen et al. 2012).

Result:

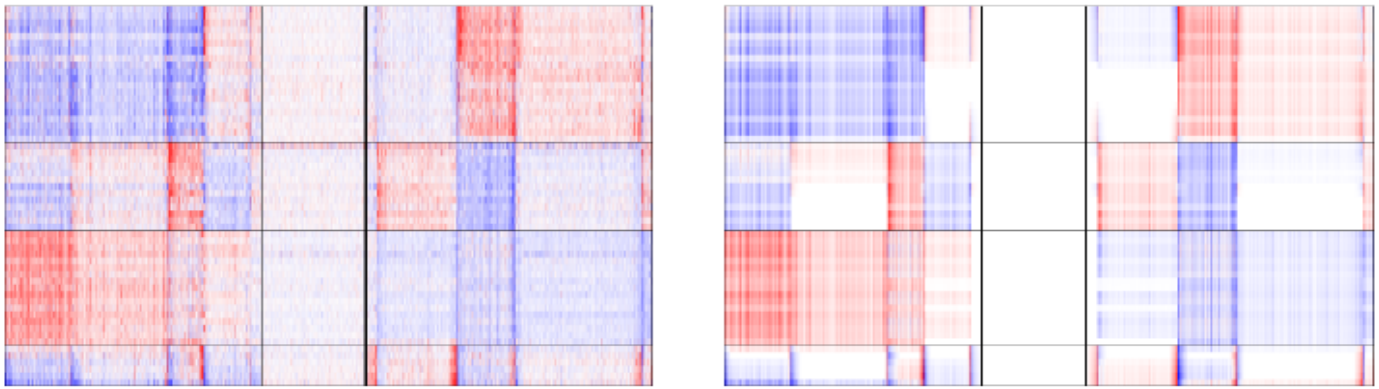


Figure: The original expression matrix (left) and the sparse estimate (right)

Method 1: Unsupervised Learning

- Penalized matrix decomposition (Witten et al. 2009, Lee et al. 2010).
- Sparse SVD (Chen et al. 2012).

Result:

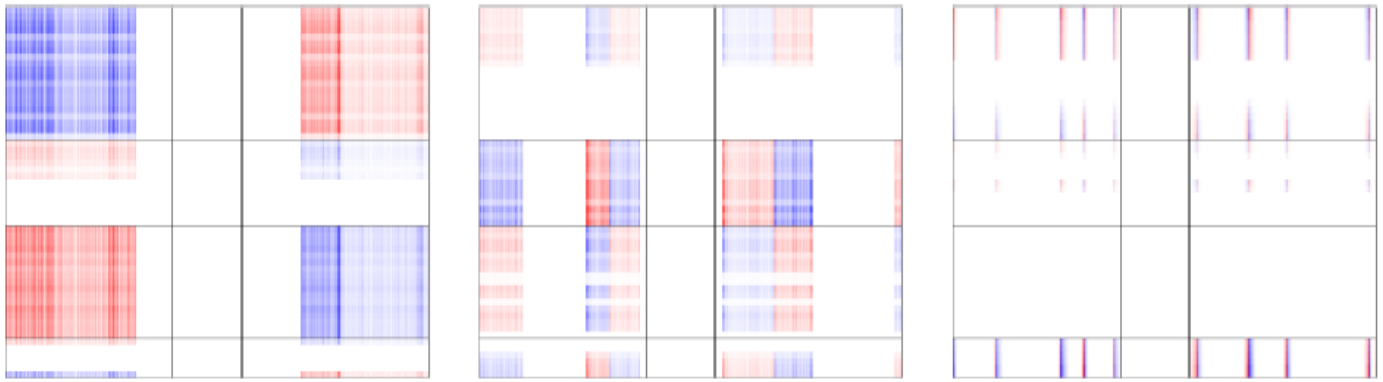


Figure: Estimated SVD layers by SSVD

Method 2: Supervised Learning

- Incorporate cancer type information.
- Extract only the associations between genes and cancer types.
- Sparse SVD (Chen et al. 2012).

Result:

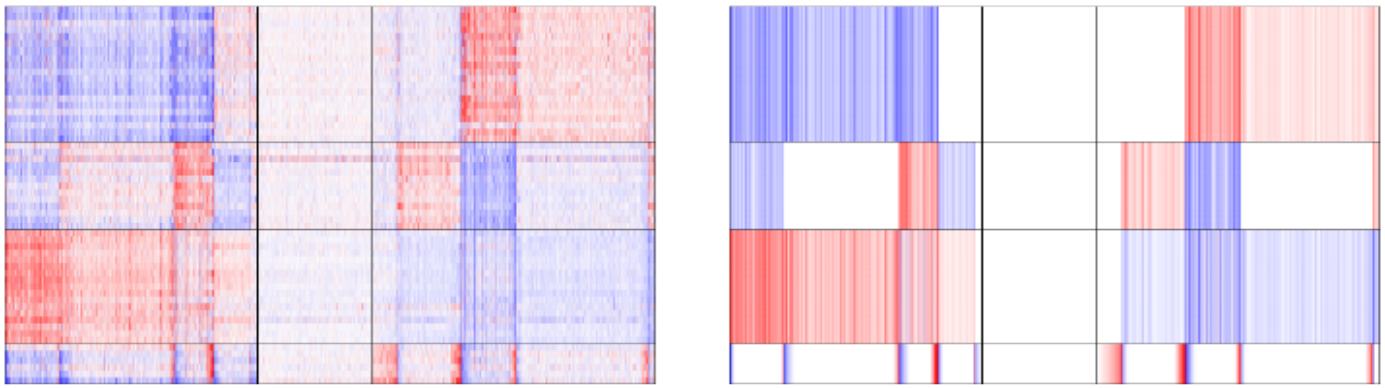


Figure: The original expression matrix (left) and the sparse estimate (right)

Method 2: Supervised Learning

- Incorporate cancer type information.
- Extract only the associations between genes and cancer types.
- Sparse SVD (Chen et al. 2012).

Result:

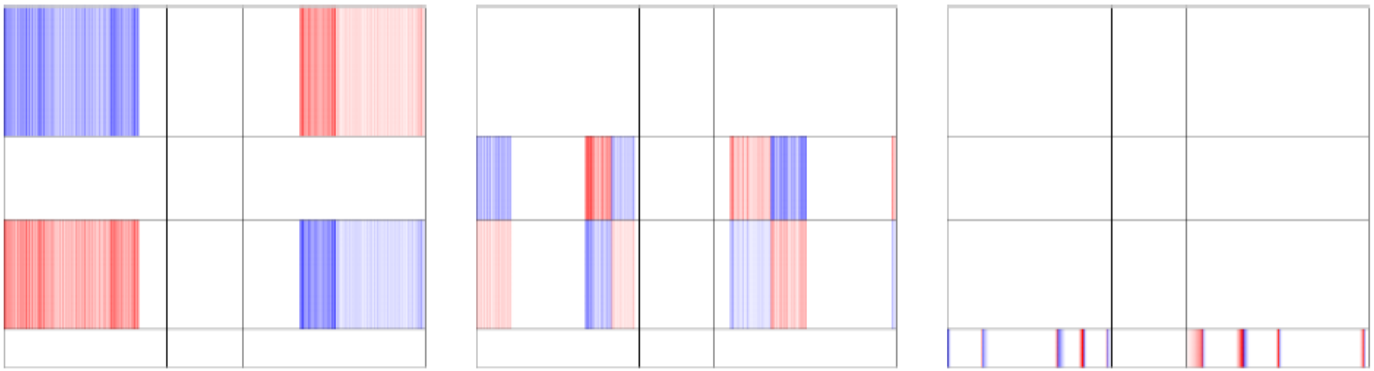


Figure: Estimated SVD layers by RRR-SSVD

Outline

References

- Busygin, S., O. Prokopyev, and P. M. Pardalos (2008). Biclustering in data mining. Computers & Operations Research 35(9), 2964-2987.
- Chen, K., K.-S. Chan, and N. C. Stenseth (2012). Reduced rank stochastic regression with a sparse singular value decomposition. Journal of the Royal Statistical Society: Series B. 74(2), 203-221.
- Jolliffe, I. T., N. T. Trendafilov, and M. Uddin (2003, September). A Modified Principal Component Technique Based on the LASSO. Journal of Computational and Graphical Statistics 12(3), 531-547.
- Lee, M., H. Shen, J. Z. Huang, and J. S. Marron (2010). Biclustering via sparse singular value decomposition. Biometrics 66, 1087-1095.
- 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. (2006). The adaptive lasso and its oracle properties. Journal of the American Statistical Association 101, 1418-1429.

- Zou, H. and T. Hastie (2005). Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society Series B 67(2), 301-320.
- Zou, H., T. Hastie, and R. Tibshirani (2004). Sparse Principal Component Analysis. Journal of Computational and Graphical Statistics 15(2), 265-286.