

# Singular Value Decomposition (SVD) Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) Machine Learning for Finance (FIN 570)

Instructor: Jaehyuk Choi

Peking University HSBC Business School, Shenzhen, China

2019-20 Module 3 (Spring 2020)

# Eigen(spectral) decomposition

For a matrix  $A$ , eigenvalue  $\lambda_k$  and eigenvector  $v_k$  satisfy

$$Av_k = \lambda_k v_k.$$

The matrix  $A$  can be decomposed into

$$A = Q\Lambda Q^{-1},$$

where  $\Lambda$  is a diagonal matrix with values  $\lambda_k$  and  $Q = (v_1 \cdots v_n)$ , i.e.,  $Q_{*j} = v_j$ .  
When  $A$  is real and symmetric,  $Q$  is an orthonormal matrix,  $QQ^T = I$ ,

$$A = Q\Lambda Q^T,$$

# Singular Value Decomposition (SVD)

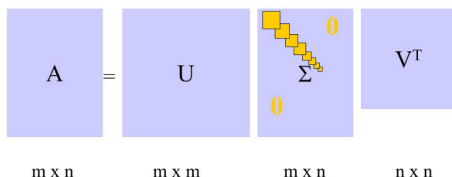
The single most useful practical concept in linear algebra:

- Any matrix (even rectangular) has a SVD.
- SVD tells everything on a matrix.

For any  $m \times n$  matrix  $A$ , there is a unique decomposition:

$$A = USV^T, \quad \text{where}$$

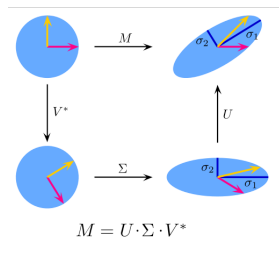
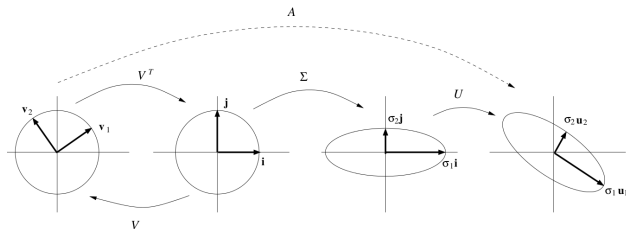
- $U$  ( $m \times m$ ): orthonormal ( $UU^T = U^TU = I$ )
- $S$  ( $m \times n$ ): diagonal. Singular values,  $s_k \geq 0$ , are in decreasing order for  $1 \leq k \leq \min(m, n)$
- $V$  ( $n \times n$ ): orthonormal ( $VV^T = V^TV = I$ )



# SVD: Intuition

Linear transformation  $A$  is decomposed into

- a rotation by  $V^T$
- a scaling by  $S$
- a rotation by  $U$



# SVD: Compact Form, Low Rank Approximation

$$\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix} \begin{bmatrix} \bullet & & & & & \\ & \bullet & & & & \\ & & \bullet & & & \\ & & & \bullet & & \\ & & & & \bullet & \\ & & & & & \bullet \end{bmatrix} \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}$$

$$\begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix} = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix} \begin{bmatrix} \bullet & & & & & \\ & \bullet & & & & \\ & & \bullet & & & \\ & & & \bullet & & \\ & & & & \bullet & \\ & & & & & \bullet \end{bmatrix} \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix}$$

$$A = U \times S \times V^T$$

$$A_k = U_k \times S_k \times V_k^T$$

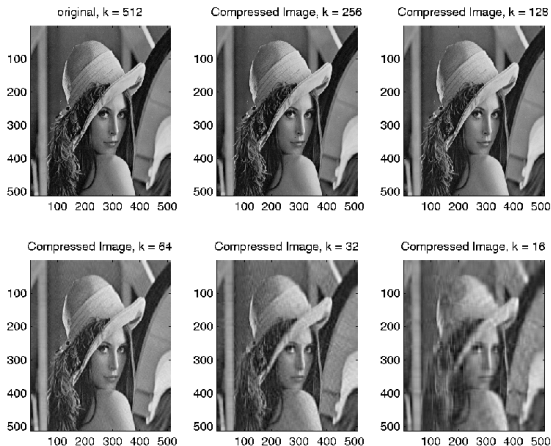
- For a non-square matrix, a compact form is enough:  
 $U$  ( $m \times r$ ),  $S$  ( $r \times r$ ),  $V$  ( $n \times r$ ) where  $r = \min(m, n)$ .
- If the rank is  $k$  ( $\leq r$ ),  $s_{j>k} = 0$ :  
 $U$  ( $m \times k$ ),  $S$  ( $k \times k$ ),  $V$  ( $n \times k$ )
- Using the first  $j$  ( $\leq k$ ) biggest singular values,

$$A_j = U_j S_j V_j^T = \sum_{i=1}^j \mathbf{u}_i s_i \mathbf{v}_i^T, \quad U_j (m \times j), S_j (j \times j), V_j (n \times j)$$

is the best approximation with rank  $j$  minimizing the norm  $\|A - A_j\|_F$

# SVD: Image Compression

An image file is nothing but a matrix, so the low-rank approximation of SVD works as an image compression method. The storage is reduced from  $mn$  to  $(m + n + 1)k$ .



# Principal Component Analysis (PCA)

If  $\mathbf{X}$  is a matrix of  $n$  samples of  $p$  features ( $n \times p$ ), the covariance matrix is

$$\Sigma = \frac{1}{n} \mathbf{X}^T \mathbf{X} : (p \times p) \text{ symmetric matrix}$$

The covariance matrix of the transformed space  $\mathbf{Z} = \mathbf{X}\mathbf{W}$  is

$$\text{Cov}(\mathbf{Z}) = \frac{1}{n} (\mathbf{X}\mathbf{W})^T (\mathbf{X}\mathbf{W}) = \frac{1}{n} \mathbf{W}^T (\mathbf{X}^T \mathbf{X}) \mathbf{W} = \mathbf{W}^T \Sigma \mathbf{W}$$

If we pick  $\mathbf{W}$  to be the orthogonal transformation of  $SVD$ , i.e.,  $\Sigma = \mathbf{W}\mathbf{S}\mathbf{W}^T$ ,

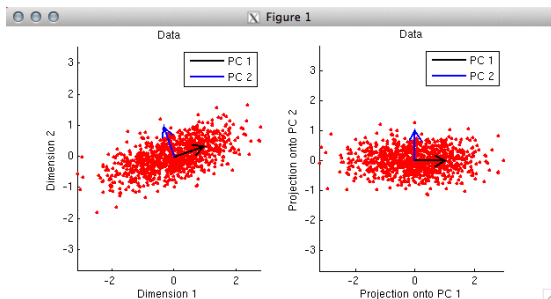
$$\text{Cov}(\mathbf{Z}) = \mathbf{S} = \text{diag}(S_{11}, \dots, S_{pp}).$$

Notice that  $\text{Cov}(Z_i, Z_j) = \mathbf{W}_{*i}^T \Sigma \mathbf{W}_{*j} = S_{ij}$  is zero if  $i \neq j$ , so the extracted features are orthogonal.

# Process of finding $W$

Let  $W = (W_{*1} \ W_{*2} \ \cdots \ W_{*p})$ .

- Find  $W_{*1}$  such that  $|W_{*1}| = 1$  and  $|W_{*1}^T \Sigma W_{*1}|$  is maximized.
- Find  $W_{*2}$  such that  $|W_{*2}| = 1$ ,  $|W_{*2}^T \Sigma W_{*2}|$  is maximized and  $W_{*1}^T W_{*2} = 0$ .
- ...
- Find  $W_{*k}$  such that  $|W_{*k}| = 1$ ,  $|W_{*k}^T \Sigma W_{*k}|$  is maximized and  $W_{*k}$  is orthogonal to  $\{W_{*j}\}$  for  $j < k$ .





# Total and Explained Variance

The total variance is the variance of all original features. Under PCA,

$$\sum_{k=1}^p \text{Var}(X_k) = \sum_{k=1}^p S_{kk}.$$

Therefore the ratio

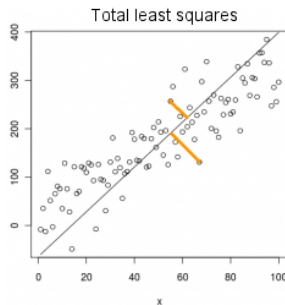
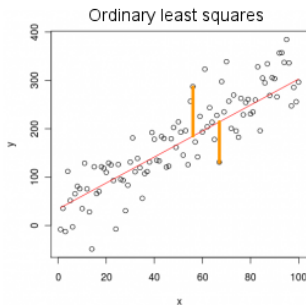
$$\frac{\sum_{j=1}^k S_{jj}}{\sum_{j=1}^p S_{jj}}$$

indicates how much of the total variance is *explained* by the first  $k$  PCA factors. Extracting features from PCA is an unsupervised learning, NOT supervised learning, because the response variable is not associated.

# PCA vs Simple Linear Regression for $(x, y)$

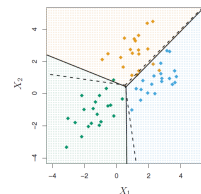
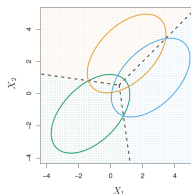
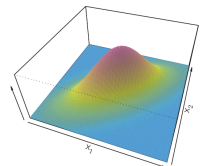
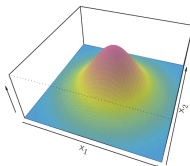
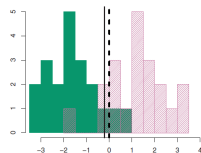
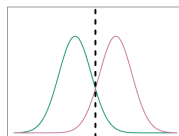
PCA is not same as Simple Linear regression (OLS)!

- **Linear Regression** minimize the the (squared) distance in  $y$ -axis.
- **PCA** (1st component) minimize the (squared) shortest distance.



# Linear Discriminant Analysis (LDA) as a classifier

- Assume the samples in each class follow normal (Gaussian) distribution.
- Estimate mean  $\hat{\mu}_k$  and variance  $\hat{\Sigma}_k$  of class  $k$ :
- Obtain multivariate normal PDF:  
 $f_k(\mathbf{x}) = n(\mathbf{x}|\hat{\mu}_k, \hat{\Sigma}_k)$
- LDA if  $\Sigma_W = \sum_{k=1}^K \Sigma_k$  (within covariance) is used for all  $\Sigma_k$ .
- QDA if  $\Sigma_k$  is estimated for each class  $k$
- A test sample  $\mathbf{x}$  is classified to the class  $k$  for which  $f_k(\mathbf{x})$  is largest.



# LDA as a dimensionality reduction

- Given the LDA assumptions, which direction  $\mathbf{w}$  best separates the feature?
- $\mathbf{w} \approx \boldsymbol{\mu}_2 - \boldsymbol{\mu}_1$  ? Probably not the best.
- If  $(\mu_{1,2}, \sigma_{1,2}^2)$  is the mean and variance pair of the samples (1-D) projected on  $\mathbf{w}$ ,  $y = \mathbf{x}\mathbf{w}$ , with  $|\mathbf{w}| = 1$ , we want to maximize the Fisher criterion:

$$J(\mathbf{w}) = \frac{(\mu_2 - \mu_1)^2}{N_1\sigma_1^2 + N_2\sigma_2^2} = \frac{\mathbf{w}^T(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^T(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)\mathbf{w}}{\mathbf{w}^T(N_1\boldsymbol{\Sigma}_1 + N_2\boldsymbol{\Sigma}_2)\mathbf{w}} = \frac{\mathbf{w}^T\mathbf{S}_B\mathbf{w}}{\mathbf{w}^T\mathbf{S}_W\mathbf{w}},$$

where  $\mathbf{S}_W$  and  $\mathbf{S}_B$  are *within*- and *between*-class variance matrices

$$\mathbf{S}_W = \sum_{k=1,2} N_k \boldsymbol{\Sigma}_k, \quad \mathbf{S}_B = (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^T(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$$

# LDA as a dimensionality reduction

- The direction  $\mathbf{w}$  maximizing  $J(\mathbf{w})$  is  $\propto \mathbf{S}_W^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$ .
- In general, the eigenvectors,  $\mathbf{W}$ , of  $\mathbf{S}_W^{-1}\mathbf{S}_B$  in the decreasing order of eigenvalue (similar to PCA) are the best directions to discriminate features.
- The transformation  $\mathbf{z} = \mathbf{x}\mathbf{W}$  is the extracted factors with the best separability, which can be used for other ML methods.

