

# Principal Component Analysis

**Christopher Ting**

<http://cting.x10host.com/TMU/TMU.html>

<http://www.mysmu.edu/faculty/christophert/>

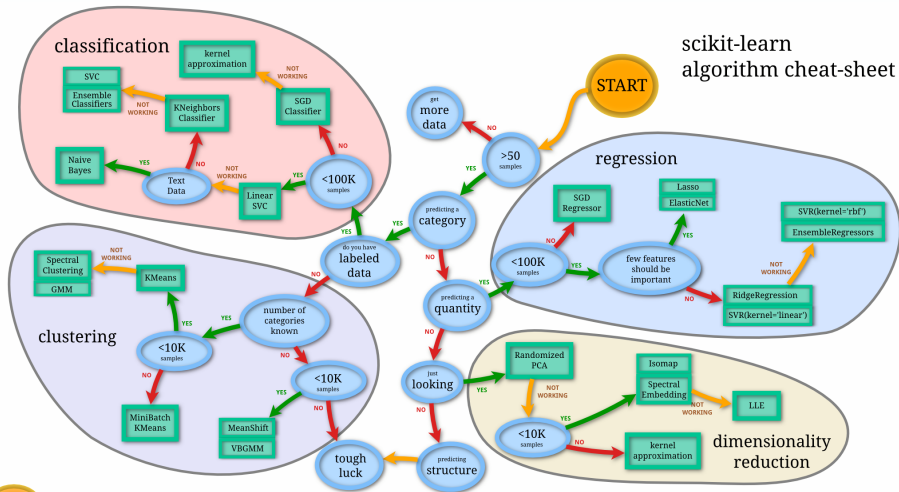
✉: [christophert@smu.edu.sg](mailto:christophert@smu.edu.sg)

June 7, 2018

# Broad Lesson Plan

- 1 Introduction
- 2 Matrix Calculus
- 3 Intuitive PCA
- 4 General PCA
- 5 Linear Algebra
- 6 SVD
- 7 Yield Curves
- 8 Takeaways

# Machine Learning with Python



Back



# Introduction

- Principal component analysis (PCA) is a technique that is useful for the **compression** and **classification** of data.
- The intent is to **reduce the dimensionality** of a data set (sample) by finding a new set of variables, smaller than the original set of variables, that nonetheless retains most of the sample's information.
- PCA is an application of **linear algebra**.
- PCA is basically rotating and shifting the axes to one that presents important information.
- PCA has a lot of applications across different fields.

# Definition of Vector Differentiation

✿ Let  $\mathbf{x}$  be a column  $k$ -vector. Consider the function

$$g(\mathbf{x}) = g(x_1, x_2, \dots, x_k) : \mathfrak{R}^k \longrightarrow \mathfrak{R}.$$

✿ Vector derivative of a scalar  $g(\mathbf{x})$  is

$$\frac{\partial}{\partial \mathbf{x}} g(\mathbf{x}) = \begin{bmatrix} \frac{\partial}{\partial x_1} g(\mathbf{x}) \\ \frac{\partial}{\partial x_2} g(\mathbf{x}) \\ \vdots \\ \frac{\partial}{\partial x_k} g(\mathbf{x}) \end{bmatrix}. \quad (1)$$

and

$$\frac{\partial}{\partial \mathbf{x}^\top} g(\mathbf{x}) = \left[ \frac{\partial}{\partial x_1} g(\mathbf{x}) \quad \frac{\partial}{\partial x_2} g(\mathbf{x}) \quad \dots \quad \frac{\partial}{\partial x_k} g(\mathbf{x}) \right] \quad (2)$$

## Definition of Vector Differentiation (Cont'd)

- ✿ Vector derivative of a vector function, e.g.  $\begin{bmatrix} f(\mathbf{x}) \\ g(\mathbf{x}) \end{bmatrix}$  is to treat each element of the vector as a scalar and apply the rules (1) and (2) for vector derivative of a scalar.

$$\frac{\partial}{\partial \mathbf{x}} \begin{bmatrix} f(\mathbf{x}) \\ g(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial \mathbf{x}} f(\mathbf{x}) \\ \frac{\partial}{\partial \mathbf{x}} g(\mathbf{x}) \end{bmatrix} \quad (3)$$

and

$$\frac{\partial}{\partial \mathbf{x}^\top} \begin{bmatrix} f(\mathbf{x}) \\ g(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial \mathbf{x}^\top} f(\mathbf{x}) & \frac{\partial}{\partial \mathbf{x}^\top} g(\mathbf{x}) \end{bmatrix} \quad (4)$$

- ✿ Likewise for a matrix of  $k$ -valued functions

# Basic Properties

✿ For constant vector  $\mathbf{a}$  and matrix  $\mathbf{A}$ ,

$$\frac{\partial}{\partial \mathbf{x}} (\mathbf{a}^\top \mathbf{x}) = \frac{\partial}{\partial \mathbf{x}} (\mathbf{x}^\top \mathbf{a}) = \mathbf{a}, \quad \frac{\partial}{\partial \mathbf{x}^\top} (\mathbf{a}^\top \mathbf{x}) = \mathbf{a}^\top$$

$$\frac{\partial}{\partial \mathbf{x}^\top} (\mathbf{A}\mathbf{x}) = \mathbf{A}$$

$$\frac{\partial}{\partial \mathbf{x}} (\mathbf{x}^\top \mathbf{A}\mathbf{x}) = (\mathbf{A} + \mathbf{A}^\top)\mathbf{x}$$

$$\frac{\partial^2}{\partial \mathbf{x} \partial \mathbf{x}^\top} (\mathbf{x}^\top \mathbf{A}\mathbf{x}) = \mathbf{A} + \mathbf{A}^\top$$

# Vector of Random Variables

✿ Let  $\mathbf{x}$  be a vector of  $m$  random variables ( $m$ -variate) with mean vector  $\boldsymbol{\mu} = \mathbb{E}(\mathbf{x})$  and covariance matrix  $\boldsymbol{\Sigma} = \mathbb{C}(\mathbf{x})$ .

✿ Example:  $m = 2$

$$\mathbf{x} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}, \quad \boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{pmatrix}$$

- $\sigma_{ii} = \sigma_i^2$  is the variance of  $r_i$ ,  $i = 1, 2$ .
- The covariance of  $r_1$  with  $r_2$  is  $\sigma_{12} = \sigma_{21}$ .

✿ Consider a linear combination of  $m$  random variables:

$$c = a_1 x_1 + a_2 x_2 + \cdots + a_m x_m$$

✿ Variance of  $c$ ,

$$\mathbb{V}(c) = \mathbb{V}(\mathbf{a}^\top \mathbf{x}) = \mathbf{a}^\top \mathbb{C}(\mathbf{x}) \mathbf{a} = \mathbf{a}^\top \boldsymbol{\Sigma} \mathbf{a}$$

## Example: $n = 2$ (Bivariate)

✿ Note that  $\mathbb{V}(\mathbf{c}) = \mathbb{C}(\mathbf{c}, \mathbf{c})$

✿ Hence,

$$\begin{aligned}
 \mathbb{C}(a_1\mathbf{r}_1 + a_2\mathbf{r}_2, a_1\mathbf{r}_1 + a_2\mathbf{r}_2) &= a_1^2 \mathbb{C}(\mathbf{r}_1, \mathbf{r}_1) + a_1a_2 \mathbb{C}(\mathbf{r}_1, \mathbf{r}_2) \\
 &\quad + a_2a_1 \mathbb{C}(\mathbf{r}_2, \mathbf{r}_1) + a_2^2 \mathbb{C}(\mathbf{r}_2, \mathbf{r}_2) \\
 &= a_1^2 \mathbb{V}(\mathbf{r}_1) + 2a_1a_2 \mathbb{C}(\mathbf{r}_1, \mathbf{r}_2) + a_2^2 \mathbb{V}(\mathbf{r}_2) \\
 &= a_1\sigma_{11} + a_1a_2\sigma_{12} + a_2a_1\sigma_{21} + a_2^2\sigma_{22} \\
 &= (a_1 \quad a_2) \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \\
 &= \mathbf{a}^\top \boldsymbol{\Sigma} \mathbf{a}
 \end{aligned}$$

# Maximization with Constraints

- ✿ Problem formulation

$$\max_{\mathbf{a}} \mathbf{a}^T \Sigma \mathbf{a} \quad \text{subject to } \mathbf{a}^T \mathbf{a} = 1.$$

- ✿ Use the Lagrange multiplier technique. Consider instead a cost function  $f$

$$f(\mathbf{a}, \lambda; \mathbf{x}) = \mathbf{a}^T \Sigma \mathbf{a} + \lambda(1 - \mathbf{a}^T \mathbf{a})$$

- ✿ The first-order condition is

$$\frac{\partial f}{\partial \mathbf{a}} = 2\Sigma \mathbf{a} - 2\lambda \mathbf{a} = \mathbf{0}$$

$$\frac{\partial f}{\partial \lambda} = 1 - \mathbf{a}^T \mathbf{a} = 0$$

- ✿ We obtain the characteristic equation of eigenvalue problem:

$$\Sigma \mathbf{a} = \lambda \mathbf{a}$$

# First Principal Component

- With respect to the eigenvector  $\mathbf{a}$ ,

$$\mathbb{V}(\mathbf{c}) = \mathbf{a}^\top \Sigma \mathbf{a} = \mathbf{a}^\top (\lambda \mathbf{a}) = \lambda \mathbf{a}^\top \mathbf{a} = \lambda$$

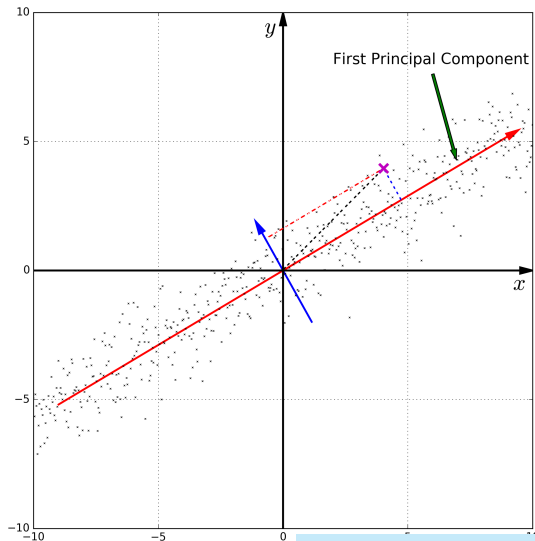
- Therefore,  $\mathbb{V}(\mathbf{c})$  is maximized if  $\lambda$  is the largest eigenvalue.
- Correspondingly,  $\mathbf{c}$  is the first principal component for the first eigenvector  $\mathbf{a}$ .
- In general, suppose there are  $m$  random variables, eigenvectors  $\mathbf{a}_i$  of eigenvalues  $\lambda_i$ ,  $i = 1, 2, \dots, m$ , we have,

$$\mathbb{V}(\mathbf{a}_i^\top \mathbf{x}) = \lambda_i \tag{5}$$

# PCA: Coordinate Transformation

- Variance along the direction of first principal component ( $u$  axis) is largest.
- The axis perpendicular to the  $u$  axis is second principal component ( $v$  axis).
- Since the radius  $r$  (length of dashed black lines) should not change,

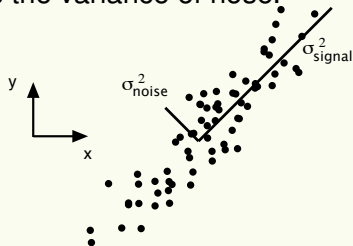
$$r^2 = x^2 + y^2 = u^2 + v^2$$



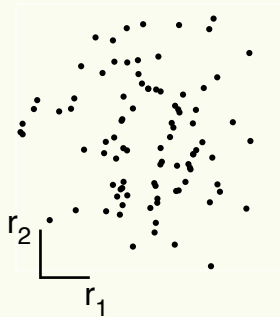
# Insight!

- In standard asset pricing since Markowitz, variance (or standard deviation) is a measure of risk (more appropriately uncertainty).
- In PCA, variance  $\sigma_u^2$  is a measure of information (signal). The larger dispersion of data along the  $u$  axis suggests that more bits are needed to describe it.
- In the two-dimension case, the variance  $\sigma_v^2$  of dispersion in the direction of the  $u$  axis can be considered as the variance of noise.

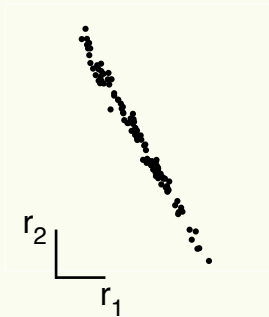
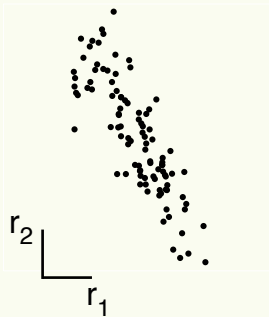
- **Signal-to-noise ratio**  $\text{SNR} = \frac{\sigma_u^2}{\sigma_v^2}$



# Redundancy



low redundancy

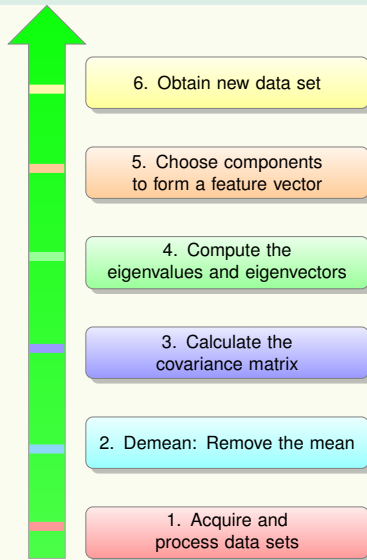


high redundancy

⇒ Central idea: correlation, given  $r_1$ , you know quite well about  $r_2$  and vice versa

⇒ Implication: **Dimension reduction!**

# PCA Algorithm



# Demo Python Codes

```

from __future__ import print_function, division
import numpy as np
import matplotlib.pyplot as plt

# Step 1: Collect data
year = range(2007,2017)
x_original = [2.5,0.4,2.2,1.9,3.1,2.3,2.0,1.0,1.5,1.1] # Annual return of Stock X in %
y_original = [2.4,0.6,2.9,2.2,3.0,2.7,1.6,1.1,1.6,0.9] # Annual return of Stock Y in %

# Step 2: Demean
mx = np.mean(x_original)
my = np.mean(y_original)

x = np.asarray(x_original - mx)
y = np.asarray(y_original - my)

X = np.matrix([x, y])

# Sep 3: Compute the variance-covariance matrix
Sigma = np.cov(X, ddof=1)

# Step 4: Compute the eigenvalues and eigenvectors
eigenvalues, eigenvectors = np.linalg.eig(Sigma)

# Step 5: Choose components to form features
features2 = eigenvectors

```

## Demo Python Codes (Cont'd)

```
# Step 6: Get the data back after transformation
```

```
fdata2 = features2.dot(X)
fx = np.asarray(fdata2[0])
fy = np.asarray(fdata2[1])
fx = fx[0]
fy = fy[0]
```

```
# Visual presentation of results
```

```
plt.plot(x, y, 'bo',
markersize=7, label="demeaned")
plt.plot(fx, -fy, 'r^',
markersize=7, label="pca transformed")
```

```
slope = Sigma[0,1]/np.var(y, ddof=1)
lx = np.linspace(-2, 2, 100)
ly = lx * slope
plt.plot(lx, ly, '-b', linewidth=2)
```

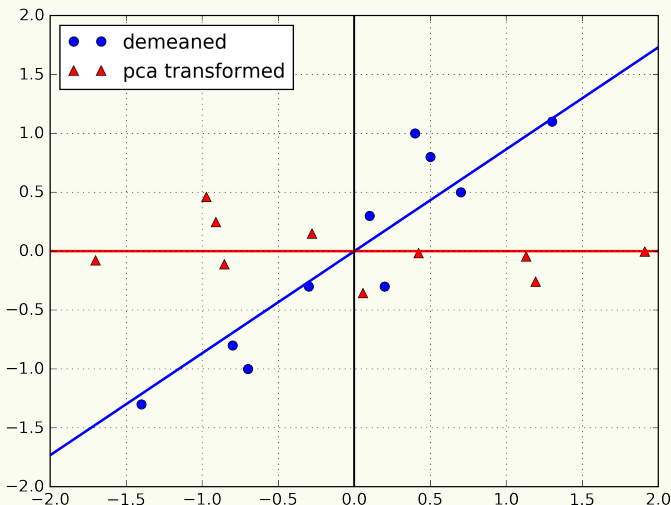
```
ly2 = np.zeros(100, dtype=float)
plt.plot(lx, ly2, '-r', linewidth=2)
plt.plot(ly2, lx, '-k', linewidth=1.5)
plt.grid()
plt.legend(loc="upper left")
plt.show()
```

# Results

Blue line is the principal component, obtained by OLS regression.

Data (in red) obtained back from PCA have no slope.

PCA eigenvectors are invariant with respect to sign change.



## Remarks

- Linearly sets up the problem as a change of basis.
- Large variances have important structure.
- The principal components are orthogonal.

# Basis

→ A coordinate system can be expressed as  $m$  vectors that are **orthogonal** to each other.

→  $m$ -dimensional coordinate system:  $m \times m$  identity matrix

$$B = \begin{bmatrix} \mathbf{b}_1^\top \\ \mathbf{b}_2^\top \\ \vdots \\ \mathbf{b}_m^\top \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} =: I$$

where each row is an **orthonormal basis vector**  $\mathbf{b}_i^\top$  with  $m$  components.

→ Examples:

- ⌘ A portfolio or index with  $m$  component stocks
- ⌘ Yield curve with  $m$  anchor maturities

## Change of Basis

- The goal of principal component analysis is to identify the most meaningful **basis** to re-express a data set.
- Is there another basis, which is a linear combination of the original basis, that best re-expresses our data set?
- Let  $X$  and  $Y$  be  $m \times n$  matrices related by a linear transformation  $P$ .

$$PX = Y$$

- $X$  is the original recorded data set and  $Y$  is a **re-representation** of that data set.

## Covariance Matrix from Data Matrix

- Data matrix  $\mathbf{X}$ :  $n$  sets of observations. Each set has  $m$  observations for  $m$  variables of interest
- Sample variance-covariance matrix

$$\mathbf{C}_X = \frac{1}{n-1} \mathbf{X} \mathbf{X}^\top$$

- $\mathbf{C}_X$  is a square symmetric  $m \times m$  matrix.
- The diagonal terms of  $\mathbf{C}_X$  are the variances of particular measurement types.
- The off-diagonal terms of  $\mathbf{C}_X$  are the covariances between measurement types.

# Covariance Matrix in PCA

- The diagonal terms, by assumption, correspond to principal structure, especially for the largest ones.
- For the off-diagonal terms, large magnitudes correspond to high redundancy.
- Our goals are to
  - 1 minimize redundancy, measured by the magnitude of the covariance
  - 2 maximize the signal, measured by the variance.
- What should be the properties of  $C_Y$  for PCA?

# Diagonalization of Covariance Matrix

- ‡ All off-diagonal terms in  $C_Y$  should be zero.
- ‡ Thus,  $C_Y$  must be a diagonal matrix. Or, said another way,  $Y$  is **decorrelated**.
- ‡ Each successive dimension in  $Y$  should be rank-ordered according to variance.

## How PCA Works

- ‡  $P$  acts as a **generalized rotation** to align a basis with the axis of maximal variance. In multiple dimensions this could be performed by a simple algorithm.
- ‡ Select a normalized direction in  $m$ -dimensional space along which the variance in  $X$  is maximized. Save this vector as  $p_1$ .
- ‡ Repeat this procedure until  $m$  vectors are selected.
- ‡ The resulting ordered set of  $p$ 's are the **principal components**.

# PCA Eigenvector Decomposition

- Find some orthonormal matrix  $P$  in  $Y = PX$  such that  $C_Y$  is a diagonal matrix.
- The  $m$ -dimensional rows of  $P$  are the principal components of  $X$ .
- Relation between  $C_Y$  and  $C_X$ :

$$\begin{aligned}C_Y &= \frac{1}{n-1}YY^T = \frac{1}{n-1}PX(PX)^T \\ &= \frac{1}{n-1}PXX^T P^T = P \left( \frac{1}{n-1}XX^T \right) P^T \\ &= PC_X P^T.\end{aligned}$$

# Orthogonal

## Definition 1

A  $m \times n$  matrix  $\mathbf{A} = [\mathbf{a}_1 \quad \mathbf{a}_2 \quad \cdots \quad \mathbf{a}_n]$  is said to be orthogonal

$$\mathbf{a}_i^\top \mathbf{a}_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases},$$

where each  $\mathbf{a}_i, i = 1, 2, \dots, n$  is a column vector of  $m$  rows.

## Theorem 1

The inverse of an orthogonal matrix is its transpose.

## Proof

Note that the  $(\mathbf{A}^\top \mathbf{A})_{ij} = \mathbf{a}_i^\top \mathbf{a}_j = \delta_{ij}$ , where  $\delta_{ij}$  is Kronecker's delta function. Since  $\mathbf{A}^\top \mathbf{A} = \mathbf{I}$ , it follows that  $\mathbf{A}^{-1} = \mathbf{A}^\top$ .

# Symmetric Matrix

## Definition 2

A  $m \times m$  square matrix  $A$  is said to be symmetric if  $A_{ij} = A_{ji}$ , i.e., row index and column index are interchangeable:  $A^T = A$ .

## Theorem 2

For any  $m \times n$  matrix  $A$ ,  $A^T A$  and  $AA^T$  are symmetric.

## Proof

$$(AA^T)^T = A^{TT}A^T = AA^T$$

$$(A^T A)^T = A^T A^{TT} = A^T A$$

# Symmetric and Orthogonally Diagonalizable

## Definition 3

A matrix  $A$  is said to be diagonalizable if there exists some  $E$  such that  $A = EDE^T$ , where  $D$  is a diagonal matrix and  $E$  is some special matrix that diagonalizes  $A$ . Additionally, if  $E$  is orthogonal, then  $A$  is said to be orthogonally diagonalizable.

## Theorem 3

A matrix is symmetric if it is orthogonally diagonalizable.

## Proof

Suppose  $A$  is orthogonally diagonalizable. Let us compute  $A^T$ .

$$A^T = (EDE^T)^T = E^{TT}D^TE^T = EDE^T = A.$$

Hence, if  $A$  is orthogonally diagonalizable, it must also be symmetric.

# Orthonormal Eigenvectors

Let  $A$  be a square  $n \times n$  symmetric matrix with associated eigenvectors  $\{e_i\}_{i=1}^n$ . Let  $E = [e_1 \ e_2 \ \cdots \ e_n]$ , and  $D_{ij} = \lambda_i \delta_{ij}$

## Theorem 4

A symmetric matrix  $A$  is diagonalized by a matrix of its orthonormal eigenvectors.

## Proof (Part 1)

This theorem asserts that there exists a diagonal matrix  $D$  such that  $A = EDE^T$ .

Let  $A$  be any matrix, not necessarily symmetric, and let it have independent eigenvectors  $e_i$  (i.e. no degeneracy).

$$AE = [Ae_1 \ Ae_2 \ \cdots \ Ae_n] = [\lambda_1 e_1 \ \lambda_2 e_2 \ \cdots \ \lambda_n e_n] = ED.$$

Since  $AE = ED$ , it follows that  $A = EDE^{-1}$ .

## Orthonormal Eigenvectors (Cont'd)

### Proof (Part 2)

We will show that a symmetric matrix  $A$  always has orthogonal eigenvectors. For any  $i$  and  $j$  such that  $i \neq j$ ,

$$\lambda_i \mathbf{e}_i^\top \mathbf{e}_j = (\mathbf{A}\mathbf{e}_i)^\top \mathbf{e}_j = \mathbf{e}_i^\top \mathbf{A}^\top \mathbf{e}_j = \mathbf{e}_i^\top \mathbf{A}\mathbf{e}_j = \mathbf{e}_i^\top (\lambda_j \mathbf{e}_j)$$

$$\implies \lambda_i \mathbf{e}_i^\top \mathbf{e}_j = \lambda_j \mathbf{e}_i^\top \mathbf{e}_j$$

$$\implies (\lambda_i - \lambda_j) \mathbf{e}_i^\top \mathbf{e}_j = 0$$

Since we have assumed that the eigenvalues are unique, it must be that  $\mathbf{e}_i^\top \mathbf{e}_j = 0$ . Therefore, the eigenvectors of a symmetric matrix are orthogonal. By theorem 1 that  $\mathbf{E}^\top = \mathbf{E}^{-1}$  and we can rewrite the final result

$$\mathbf{A} = \mathbf{E}\mathbf{D}\mathbf{E}^\top$$

# Orthogonal Basis

## Theorem 5

For any  $m \times n$  matrix  $\mathbf{X}$ , the symmetric matrix  $\mathbf{X}^\top \mathbf{X}$  has a set of orthonormal eigenvectors  $\{\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \dots, \hat{\mathbf{v}}_n\}$  and a set of associated eigenvalues  $\{\lambda_1, \lambda_2, \dots, \lambda_n\}$ . The set of vectors  $\{\mathbf{X}\hat{\mathbf{v}}_1, \mathbf{X}\hat{\mathbf{v}}_2, \dots, \mathbf{X}\hat{\mathbf{v}}_n\}$  then form an orthogonal basis, where each vector  $\mathbf{X}\hat{\mathbf{v}}_i$  is of length  $\sqrt{\lambda_i}$ .

## Proof

$$\begin{aligned} (\mathbf{X}\hat{\mathbf{v}}_i)^\top (\mathbf{X}\hat{\mathbf{v}}_j) &= \hat{\mathbf{v}}_i^\top \mathbf{X}^\top \mathbf{X} \hat{\mathbf{v}}_j = \hat{\mathbf{v}}_i^\top (\lambda_j \hat{\mathbf{v}}_j) = \lambda_j \hat{\mathbf{v}}_i^\top \hat{\mathbf{v}}_j \\ &= \lambda_j \delta_{ij} \end{aligned}$$

The squared length of each vector is

$$\|\mathbf{X}\hat{\mathbf{v}}_i\|^2 = (\mathbf{X}\hat{\mathbf{v}}_i)^\top (\mathbf{X}\hat{\mathbf{v}}_i) = \lambda_i.$$

## Application to PCA

For a symmetric matrix  $A$ , Theorem 4 provides  $A = EDE^T$ , where  $D$  is a diagonal matrix and  $E$  is a matrix of eigenvectors of  $A$  arranged as columns.

We select the matrix  $P$  to be a matrix where each row  $p_i$  is an eigenvector of  $\frac{1}{n-1}XX^T$ , i.e.,  $P = E^T$ .

With this relation and Theorem 1 ( $P^{-1} = P^T$ ),

$$\begin{aligned} C_Y &= PC_XP^T = P(E^TDE)P^T = P(P^TDP)P^T \\ &= (PP^T)D(PP^T) = (PP^{-1})D(PP^{-1}) \\ &= D \end{aligned}$$

It is evident that the choice of  $P$  diagonalizes  $C_X$ , which is the goal for PCA.

# Summary

- ~ The principal components of  $\mathbf{X}$  are the eigenvectors of 
$$\mathbf{C}_X = \frac{1}{n-1} \mathbf{X} \mathbf{X}^\top.$$
- ~ The  $i$ -th diagonal value of  $\mathbf{C}_Y$  is the variance of  $\mathbf{X}$  along  $p_i$ .
- ~ Note that  $\mathbf{C}_X$  is symmetric, and the same analysis works for 
$$\mathbf{C}_X = \mathbf{C}_X^\top = \frac{1}{n-1} \mathbf{X}^\top \mathbf{X}.$$

# Framework for Singular Vector Decomposition

Let  $\mathbf{X}$  be an arbitrary  $n \times m$  matrix and  $\mathbf{X}^\top \mathbf{X}$  be a rank  $r$ , square, and symmetric  $m \times m$  matrix.

$\{\hat{\mathbf{v}}_i\}_{i=1}^r$  is the set of orthonormal  $m \times 1$  eigenvectors with associated eigenvalues  $\{\lambda_i\}_{i=1}^r$  for the symmetric matrix  $\mathbf{X}^\top \mathbf{X}$

$$\mathbf{X}^\top \mathbf{X} \hat{\mathbf{v}}_i = \lambda_i \hat{\mathbf{v}}_i$$

$\sigma_i := \sqrt{\lambda_i}$  is positive and real. It is called the **singular values**.

$\{\hat{\mathbf{u}}_i\}_{i=1}^r$  is the set of  $n \times 1$  vectors defined by

$$\hat{\mathbf{u}}_i := \frac{1}{\sigma_i} \mathbf{X} \hat{\mathbf{v}}_i$$

By Theorem 5, we have  $\hat{\mathbf{u}}_i^\top \hat{\mathbf{u}}_j = \delta_{ij}$  and  $\|\mathbf{X} \hat{\mathbf{v}}_i\| = \sigma_i$

# Singular Value Decomposition

- ↳ The scalar version of singular value decomposition is just a restatement of the definition

$$X\hat{\mathbf{v}}_i = \sigma_i\hat{\mathbf{u}}_i$$

- ↳ Now, eigenvectors  $\{\hat{\mathbf{v}}_i\}_{i=1}^r$  and vectors  $\{\hat{\mathbf{u}}_i\}_{i=1}^r$  are both orthonormal sets or bases in  $r$ -dimensional space.
- ↳ For accompanying orthogonal matrices,

$$V = [\hat{\mathbf{v}}_1 \quad \hat{\mathbf{v}}_2 \quad \cdots \quad \hat{\mathbf{v}}_{\tilde{m}}] \quad U = [\hat{\mathbf{u}}_1 \quad \hat{\mathbf{u}}_2 \quad \cdots \quad \hat{\mathbf{u}}_{\tilde{n}}]$$

where we have appended an additional  $(m - r)$  and  $(n - r)$  orthonormal vectors to “fill up” the matrices for  $V$  and  $U$  respectively (i.e. to deal with degeneracy issues).

- ↳ **Singular Value Decomposition:**  $XV = U\Sigma$

## Final Form of SVD

- Because  $V$  is orthogonal, we can multiply both sides by  $V^{-1} = V^T$  to arrive at the final form of the decomposition.

$$X = U\Sigma V^T$$

- Any arbitrary matrix  $X$  can be decomposed to an orthogonal matrix, a diagonal matrix and another orthogonal matrix.
- Any matrix performs a rotation, a stretch, and another rotation!



# Interpreting SVD

↳ Let  $Z := \Sigma V^T$ . Then

$$X = U \Sigma V^T \implies U^T X = \Sigma V^T \implies U^T X = Z$$

↳  $U^T$  performs a change of basis from  $X$  to  $Z$  in the column space.

↳ On the other hand, in the row space, define  $W := U^T \Sigma$ , and

$$(XV)^T = (\Sigma U)^T \implies V^T X^T = V^T \Sigma \implies V^T X^T = W.$$

# SVD and PCA

- Given a  $m \times n$  data matrix  $X$ , define a new  $n \times m$  matrix  $Y$  as

$$Y := \frac{1}{\sqrt{n-1}} X^\top,$$

where each column of  $Y$  has zero mean.

- Properties

$$Y^\top Y = \left( \frac{1}{\sqrt{n-1}} X^\top \right)^\top \left( \frac{1}{\sqrt{n-1}} X^\top \right) = \frac{1}{n-1} X X^\top = C_X$$

- If we calculate the SVD of  $Y$ , the columns of matrix  $V$  contain the eigenvectors of  $Y^\top Y = C_X$ . Therefore, the columns of  $V$  are the principal components of  $X$ .
- Finding the principal components amounts to finding an orthonormal basis that spans the column space of  $X$ .

## Quick Algorithm of PCA by SVD

- 1 Organize data as an  $m \times n$  matrix, where  $m$  is the number of measurement types and  $n$  is the number of samples.
- 2 Subtract off the mean for each measurement type.
- 3 Calculate the SVD or the eigenvectors of the covariance.

## Daily Treasury Yield Curve Rates

- Commonly referred to as “**Constant Maturity Treasury**” rates
- The yield curve, which relates the yield on a security to its time to maturity, is based on the closing market bid yields on actively traded Treasury securities in the over-the-counter market.
- These market yields are calculated from composites of quotations obtained by the Federal Reserve Bank of New York.
- The Treasury yield curve is estimated daily using a cubic spline model. Inputs to the model are primarily bid-side yields for on-the-run Treasury securities.
- Sample period: From 9 Feb, 2006, all maturities from 1 month to 30 years have data.

# PCA of Yield Curve

## ∞ Yield Curve Level

The first principal component can explain 92.49% of the variation in the data set.

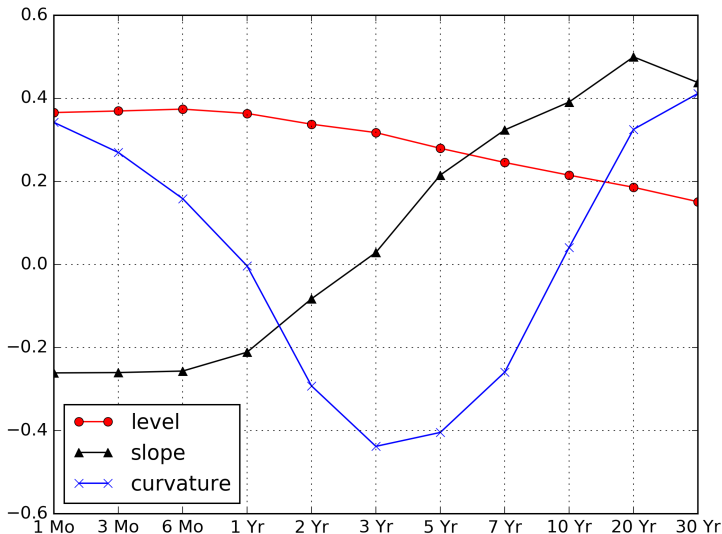
## ∞ Yield Curve Slope

The first and second components can explain 99.07% of the variation.

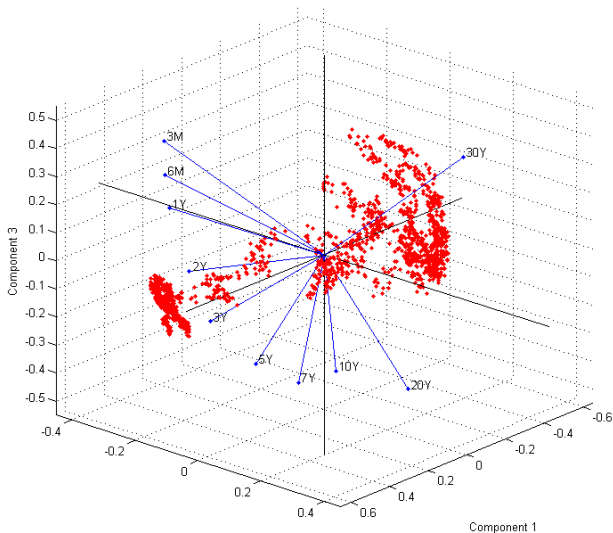
## ∞ Yield Curve Curvature

The first three components explain 99.79% of the variation.

# Three Principal Components of Yield Curve



# Mapping of Principal Components



## Takeaways

- + PCA is a linear algorithm for finding the dimensions that capture most information in data.
- + Information in the PCA framework is measured by variance.

$$\mathbb{V}(e_i^\top \mathbf{x}) = \lambda_i, \quad i = 1, 2, \dots, m.$$

- + The eigenvectors in PCA are orthonormal, and form an uncorrelated basis.
- + PCA show that U.S. Treasury Yield Curve can be explained by three “factors”: level, slope, and curvature.

## Additional Reading

- 1 [A Tutorial on Principal Component Analysis](#), Jonathon Shlens, Google Research, Working Paper (2014)