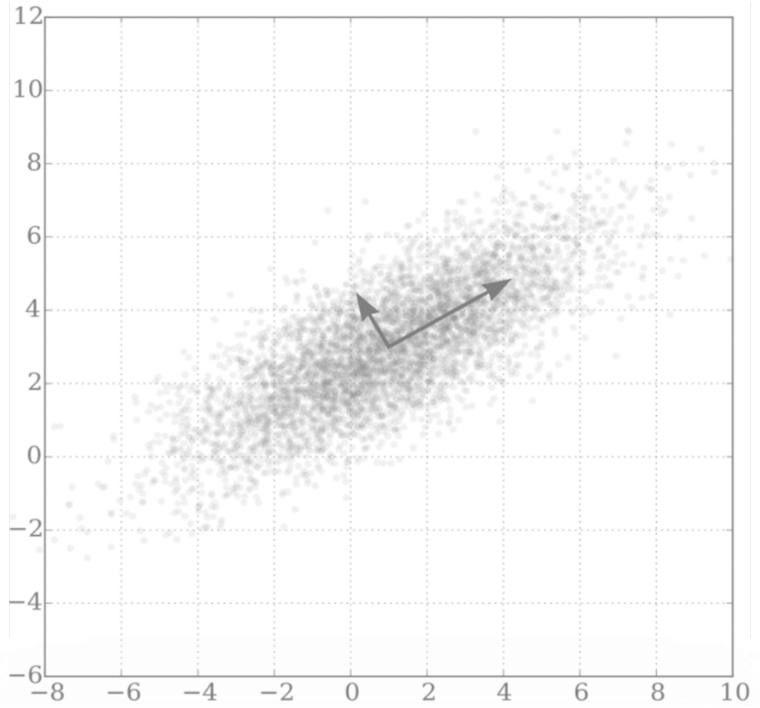
Lecture 14:
Covariance & SPSD Matrices
Introduction to PCA



CS 111: Intro to Computational Science

Spring 2023

Ziad Matni, Ph.D.

Dept. of Computer Science, UCSB

Administrative

- New homework due Monday
- Lab tomorrow

- Quiz 4 grades will be on Canvas later today

Correlated Variables

- Recall the meaning of "independent variables" (IV)
  and "dependent variables" (DV)

- What does it mean if 2 variables are ***highly correlated***?
  - The usual measure for this is Pearson's Correlation Coefficient (***r***).
  - Correlation of x to y = Correlation of y to x … it means the same thing

Variance

- A measure of how spread out the data is
  - Similar to average of the squares
  - Standard deviation is sqrt(variance)

$$\sigma^2 = \frac{\sum(\chi - \mu)^2}{N}$$

- In 2D, you can measure variance in x-dim (x-variance) and in y-dim (y-variance)

- In a dataset, where each column is a separate variable (dimension), each column has:
  - Some measure of centrality (**mean**, median, mode, etc…) `np.mean()`
  - Some measure of spread (**variance**, std. deviation, etc…) `np.var()`

Covariance

- How much **one column** (i.e. vector) of numbers varies with another
  - Similar to average of the sum of the squares of the coordinates

$$\operatorname{cov}(x, y) = \sigma_{xy} = \frac{1}{n}(\mathbf{x} - \mu_x)^T (\mathbf{y} - \mu_y)$$

n = # of items *μ = mean*

  - cov(x , y) = cov(y , x) i.e. it means the same thing…

- Correlation measures the same thing, but is scaled **-1** to **1**
  - Covariance domain is **(-∞, +∞)**

Covariance Matrix

cov(x, y) ➔ *The covariance of (column) vectors x_i and x_j*

$$C = \begin{pmatrix} cov(x_0, x_0) & cov(x_0, x_1) & cov(x_0, x_2) & \dots \\ cov(x_1, x_0) & cov(x_1, x_1) & cov(x_1, x_2) & \dots \\ cov(x_2, x_0) & cov(x_2, x_1) & cov(x_2, x_2) & \dots \\ \dots & \dots & \dots & \dots \end{pmatrix}$$

Same values! *Same values!* *Same values!*

This is why C is symmetrical…

Also, note that: $cov(x_i, x_i) = var(x_i)$

Finding the Covariance in a Data Set

- Very useful metric
  - It tells us which (if any) variables in our dataset are "telling us the same thing"

- So, our Data Set can be thought of a Matrix!
  - Each column is a variable

Column 0 = Number of Cars
Column 1 = Monthly Income
Column 2 = Eats Caviar for Breakfast at Least Once a Week (1 = Yes, 0 = No)

$$\begin{pmatrix} 1 & 5000 & 0 \\ 1 & 6000 & 0 \\ 2 & 10000 & 0 \\ 3 & 11000 & 0 \\ 192 & 9999999 & 1 \\ 2 & 22000 & 0 \end{pmatrix}$$

Symmetrical Positive Semi-Definite Matrices (**SPSD**)

- If a matrix's eigenvalues are all ≥ 0, then we call that matrix

  Positive Semi-Definite

- For any *square* matrix ***A***, $\boldsymbol{A}.\boldsymbol{A}^T$ is symmetrical
  - Proof: $(A.A^T)^T = (A^T)^T.A^T = A.A^T$
- Fun fact: Symmetrical matrices' ***eigenvectors** are orthogonal*
- If ***A*** is also *invertible and real*, then $\boldsymbol{A}.\boldsymbol{A}^T$ is also *Positive Semi-Definite*

More Revelations!!!

For any SPD or SPSD square and real matrix, ***M***:

- The ***eigenvalues*** and the ***singular values*** of ***M*** are the same!
  - But generating them in **numpy** will not give you equal lists – why?
- When performing SVD(***M***) = $\mathbf{U\Sigma V^T}$, the matrices ***U*** and ***V*** (not $\boldsymbol{V}^T$) are the same!
- The ***eigenvectors*** and the columns of ±**U** are the same!

Calculating the Covariance of a Matrix

Start with your data matrix, D, which is nxm

$$D = \begin{pmatrix} 2 & 5 & 10 \\ 6 & 3 & 8 \\ 5 & 4 & 3 \end{pmatrix}$$

1. Find the mean of each column (μ_i) in the matrix **D** and create an **m**-element row vector $\boldsymbol{\mu}^T$ (has all μ_i in it)

$$\mu^T = [4.33, \quad 4, \quad 7]$$

2. Create the matrix **M** that's μ^T stacked **n** times.

$$M = \begin{pmatrix} 4.33 & 4 & 7 \\ 4.33 & 4 & 7 \\ 4.33 & 4 & 7 \end{pmatrix}$$

3. Calculate **(D – M)**, which is matrix **D**, but each entry has the mean removed (i.e. each entry is centered around its mean)

$$D - M = \begin{pmatrix} -2.33 & 1 & 3 \\ 1.67 & -1 & 1 \\ 0.67 & 0 & 4 \end{pmatrix}$$

4. Calculate the covariance of matrix **C**, defined as:

$$C = \text{cov}(D) = \frac{1}{n}(D - M)^T(D - M)$$

*Side note: Use **(n-1)** instead of **n** only if **n** is very large.*
*This is known as using **Bessel's correction** or **Bessel's bias**.*

OR! For step-3,
just use: `np.cov(X)`

Where X is $(D – M)^T$

$$C = \begin{pmatrix} 2.89 & -1.33 & -2.67 \\ -1.33 & 0.67 & 0.67 \\ -2.67 & 0.67 & 8.67 \end{pmatrix}$$

Properties of Covariance of a Matrix

- **C** is ***symmetrical***
- **C** is also ***positive semi-definite*** if it is also real

Therefore:

- The ***eigenvalues*** and the ***singular values*** of ***C*** are the same!
- The ***eigenvectors*** of **C** and the columns of ±**U** (gotten from SVD(**C**)) are the same!

Additionally:

- **C** has a matrix rank of at most ***n – 1***
  - Has mathematical proof (uses rank-nullity theorem), but we won't cover it.
  - What is the **det(C)** then?

Principle Component Analysis (PCA)

- *The process of finding the **principal components** of a data set (i.e. a matrix) and using only the first few principal components to explain the **data outcomes** and ignoring the rest*
  - This is a technique called **variable reduction**

- The principal components are ***eigenvectors*** of the **data's covariance matrix, C**
  - These happen to ALSO be the vectors in the **U** matrix (resulting from running SVD on **C**)

- Applications:
  - Quant finance (risk management, financial derivatives)
  - Big Data mining
  - Eigenfaces/facial recognition

Principal Component Analysis (PCA)

- PCA is based on the SVD of the covariance matrix **C** of a data set **D***

If **C = U**Σ**U**^{T} (gotten thru SVD), then the columns of **U** are called the ***principal components of D***.

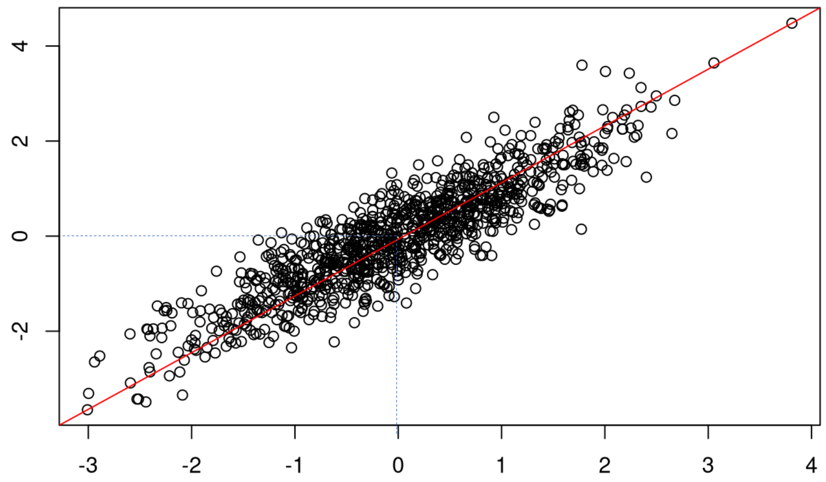
- If we take the ***k*** first principal components, we get this approximation:

$$\mathbf{d} \approx \mu + a_0\mathbf{e}_0 + a_1\mathbf{e}_1 + \cdots + a_{k-1}\mathbf{e}_{k-1}$$

  - Where $\boldsymbol{a_i}$ are *projection values* onto the eigenvectors $\boldsymbol{e_i}$
  - *Do these have anything to do with singular values of matrix C?* ***Yes!***

PCA's Key Point

- PCA helps us find the closest line thru the data points, once we center them at the origin (0,0)



- How? Take **D** and subtract the median (per column), i.e. **D – M**

- Claim: This line will be in the direction of the ***first singular vector*** $\boldsymbol{u_1}$ of the *covariance* of (**D – M**)

- When we visualize the data after PCA treatment, we can see which PCs tell me more about which dependent variables (i.e. outcome variables)

Your TO DOs!

- Finish new assignment by Monday
- Lab tomorrow!

</LECTURE>