Applied Machine Learning

Lecture 14
Dimensionality Reduction: Principal Components Analysis

Ekarat Rattagan, Ph.D.

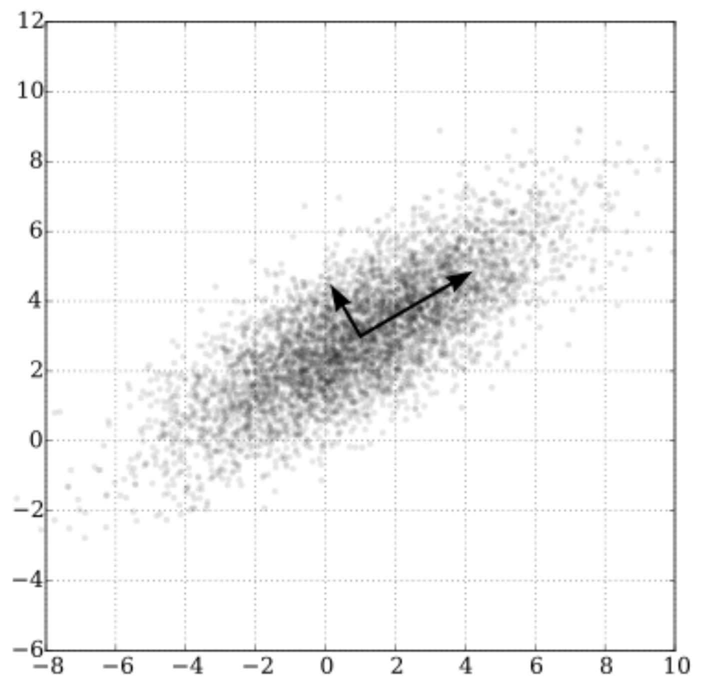
Principal Components Analysis (PCA)

- PCA: most popular instance of second main class of unsupervised learning methods, projection methods, aka dimensionality-reduction methods

- We have some data $X \in \mathbb{R}^{N \times D}$
- D may be huge, etc.
- We would like to find a new representation $Z \in \mathbb{R}^{N \times K}$ where K << D.

Principal Components Analysis (PCA)

- Aim: find a small number of "directions" in input space that explain variation in input data; re-represent data by projecting along those directions
- Important assumption: variation contains information



Principal Components Analysis (PCA)

- Can be used to:
  - Reduce number of dimensions in data
  - Find patterns in high-dimensional data
  - Visualise data of high dimensionality
- Example applications:
  - Face recognition
  - Image compression

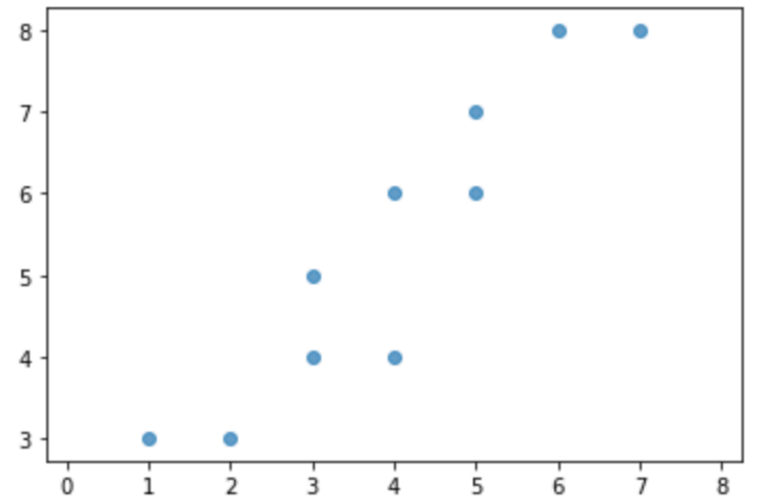
Steps of PCA

1. Let $\bar{X}$ be the mean vector (taking the mean of all rows)
2. Adjust the original data by the mean $X' = X - \bar{X}$
3. Compute the covariance matrix A of X'
4. Find the eigenvectors and eigenvalues of A.

Example

Step 1 & 2

	X1	X2	X1'	X2'
	1	3	-3	-2.4
	2	3	-2	-2.4
	3	4	-1	-1.4
	3	5	-1	-0.4
	4	4	0	-1.4
	4	6	0	0.6
	5	6	1	0.6
	5	7	1	1.6
	6	8	2	2.6
	7	8	3	2.6
Mean	4	5.4		



Mean1=4
Mean2=5.4

Covariance Matrix

Covariance: measures the correlation between X and Y

- Cov(X,Y)=0: independent
- Cov(X,Y)>0: move same direction
- Cov(X,Y)<0: move oppo dirrection

$$\operatorname{cov}(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)}$$

$$\begin{bmatrix} cov(X_1,X_1) & cov(X_1,X_2) & \cdots & cov(X_1,X_n) \\ cov(X_2,X_1) & cov(X_2,X_2) & \cdots & cov(X_2,X_n) \\ \vdots & \vdots & \vdots & \vdots \\ cov(X_n,X_1) & cov(X_n,X_2) & \cdots & cov(X_n,X_n) \end{bmatrix}$$

https://dmitry.ai/t/topic/242

Step 3

- A = $\begin{bmatrix} 3.33 & 3.22 \\ 3.22 & 3.60 \end{bmatrix}$

$$\text{cov}(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)}$$

	X1	X2	X1'	X2'		
	1	3	-3	-2.4		
	2	3	-2	-2.4		
	3	4	-1	-1.4		
	3	5	-1	-0.4		
	4	4	0	-1.4		
	4	6	0	0.6		
	5	6	1	0.6		
	5	7	1	1.6		
	6	8	2	2.6		
	7	8	3	2.6		
Mean	4	5.4				

Eigenvalues & eigenvectors

- Vectors x having same direction as Ax are called eigenvectors of A. (A is a cov matrix)
- In the equation Ax=λx, λ is called an eigenvalue of A.

$$\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} x \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 12 \\ 8 \end{pmatrix} = 4 x \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

Eigenvectors make understanding linear transformations easy. They are the "axes" (directions) along which a linear transformation acts simply by "stretching/compressing" and/or "flipping"; **eigenvalues** give you the factors by which this compression occurs.

The more directions you have along which you understand the behavior of a linear transformation, the easier it is to understand the linear transformation

Arturo Magidin (https://math.stackexchange.com/users/742/arturo-magidin), What is the importance of eigenvalues/eigenvectors?, URL (version: 2018-01-15): https://math.stackexchange.com/q/23325

Eigenvalues & eigenvectors

- We want to find x and λ.
- Ax=λx ⇔ (A-λI)x = 0, let say x != 0, then
- How to calculate x and λ:
  - Calculate det(A-λI), yields a polynomial (degree n)
  - Determine roots to det(A-λI)=0, roots are eigenvalues λ
  - Solve (A- λI) x=0 for each λ to obtain eigenvectors x

– **Why det(A-λI) ?**

1 An **eigenvector** $\boldsymbol{x}$ lies along the same line as $A\boldsymbol{x}$: $\boxed{A\boldsymbol{x} = \lambda\boldsymbol{x}.}$ The **eigenvalue** is λ.

2 If $A\boldsymbol{x} = \lambda\boldsymbol{x}$ then $A^2\boldsymbol{x} = \lambda^2\boldsymbol{x}$ and $A^{-1}\boldsymbol{x} = \lambda^{-1}\boldsymbol{x}$ and $(A + cI)\boldsymbol{x} = (\lambda + c)\boldsymbol{x}$: the same $\boldsymbol{x}$.

3 If $A\boldsymbol{x} = \lambda\boldsymbol{x}$ then $(A-\lambda I)\boldsymbol{x} = \mathbf{0}$ and $A-\lambda I$ is singular and $\boxed{\det(\boldsymbol{A}-\boldsymbol{\lambda I}) = \mathbf{0}.}$ n eigenvalues.

Step 4

- Python
  - Eigenvectors:
  - $x1 = (-0.722, \ 0.692)$, $\lambda 1 = 0.24$
  - $x2 = (0.6923, \ 0.722)$, $\lambda 2 = 6.69$
  - Thus the second eigenvector is more important!

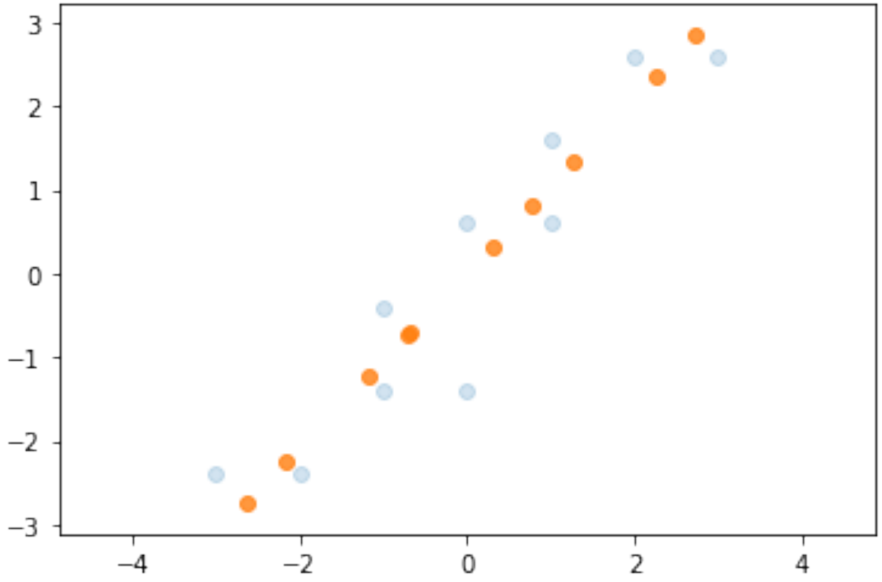
Interesting !!!

- https://lpsa.swarthmore.edu/MtrxVibe/EigMat/MatrixEigen.html
- Test
    - https://octave-online.net/

Assume we keep only one dimension

- We keep the dimension of

  $x2 = (\ 0.6923,\ 0.722),\ \ \lambda 2 = 6.69$
- Finally, we can obtain the data as



$x2 * X'$

```
[[-3.81]
 [-3.12]
 [-1.70]
 [-0.98]
 [-1.01]
 [0.43]
 [1.13]
 [1.85]
 [3.26]
 [3.95]]
```

PCA –> Original Data

- Retrieving old data (x_1, x_2)

$$
\begin{bmatrix} -3.81 \\ -3.12 \\ -1.70 \\ -0.98 \\ -1.01 \\ 0.43 \\ 1.13 \\ 1.85 \\ 3.26 \\ 3.95 \end{bmatrix} * (0.6923, 0.722) + \bar{X}
$$

```
[[-3.81]
 [-3.12]
 [-1.70]
 [-0.98]
 [-1.01]
 [0.43]
 [1.13]
 [1.85]
 [3.26]
 [3.95]]
```

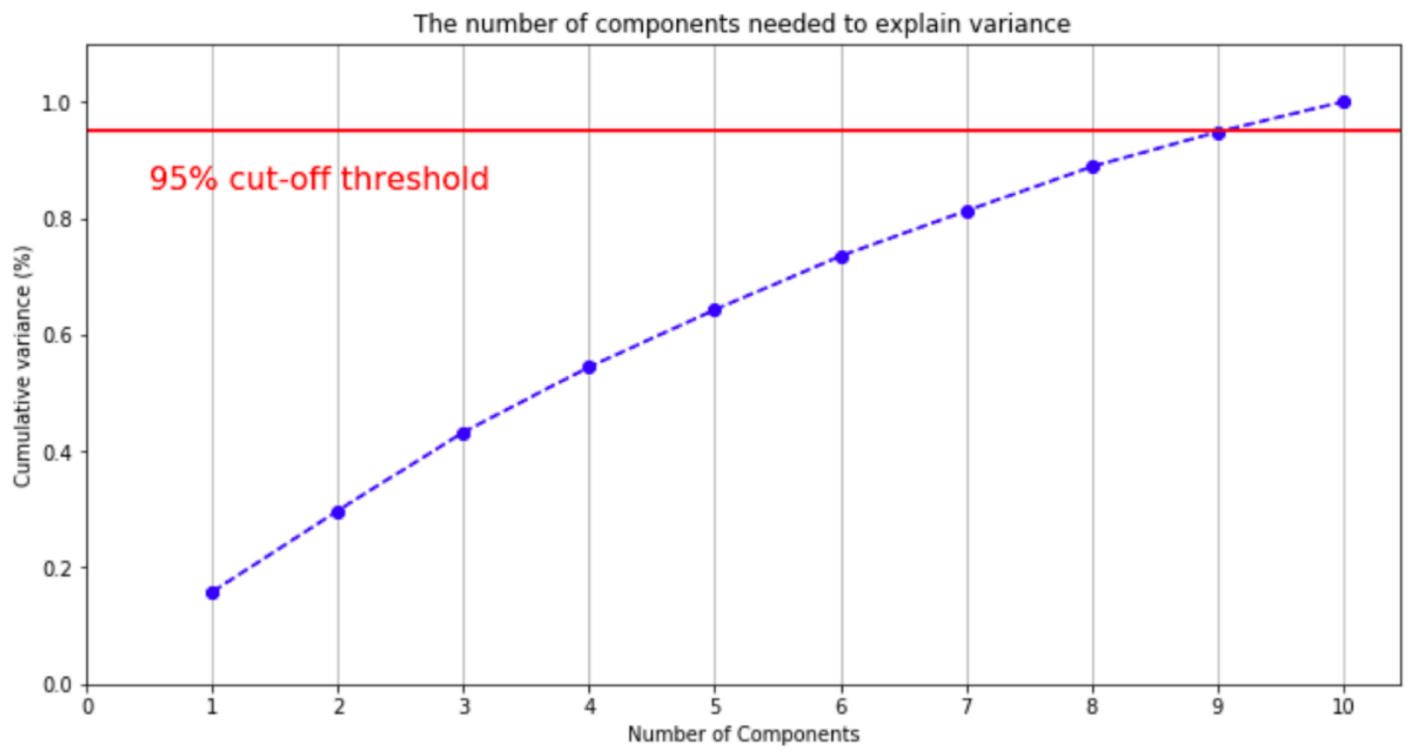
PCA –> Original Data

- Retrieving old data (x_1, x_2)

$$
\begin{bmatrix} -2.64 & -2.75 \\ -2.16 & -2.25 \\ -1.18 & -1.23 \\ -0.68 & -0.71 \\ -0.70 & -0.73 \\ 0.30 & 0.31 \\ 0.78 & 0.81 \\ 1.28 & 1.33 \\ 2.26 & 2.35 \\ 2.74 & 2.85 \end{bmatrix} + \bar{X} = \begin{bmatrix} 1.36 & 2.65 \\ 1.84 & 3.15 \\ 2.82 & 4.17 \\ 3.32 & 4.69 \\ 3.30 & 4.67 \\ 4.30 & 5.71 \\ 4.78 & 6.21 \\ 5.28 & 6.73 \\ 6.26 & 7.75 \\ 6.74 & 8.25 \end{bmatrix} \sim \begin{bmatrix} 1.0, & 3.0 \\ 2.0, & 3.0 \\ 3.0, & 4.0 \\ 3.0, & 5.0 \\ 4.0, & 4.0 \\ 4.0, & 6.0 \\ 5.0, & 6.0 \\ 5.0, & 7.0 \\ 6.0, & 8.0 \\ 7.0, & 8.0 \end{bmatrix}
$$

Mean1=4
Mean2=5.4

PCA — how to choose the number of components?



In this case, to get 95% of variance explained I need 9 principal components.

https://www.mikulskibartosz.name/pca-how-to-choose-the-number-of-components/

Applications