

# Problem Formulation

*Principal Component Analysis*

Unsupervised Learning

## Choosing $k$ (number of principal components)

Average squared projection error:

Total variation in the data:

## Choosing $k$ (number of principal components)

Average squared projection error:  $\frac{1}{m} \sum_{i=1}^m \|x^{(i)} - x_{approx}^{(i)}\|^2$

Total variation in the data:  $\frac{1}{m} \sum_{i=1}^m \|x^{(i)}\|^2$

Typically, choose  $k$  to be smallest value so that

$$\frac{\frac{1}{m} \sum_{i=1}^m \|x^{(i)} - x_{approx}^{(i)}\|^2}{\frac{1}{m} \sum_{i=1}^m \|x^{(i)}\|^2} \leq 0.01 \quad (1\%)$$

“99% of variance is retained”

Windows'u Etkinleştir  
Windows'u etkinleştirmek için Ayarlar'a gidin.

## Choosing $k$ (number of principal components)

Average squared projection error:  $\frac{1}{m} \sum_{i=1}^m \|x^{(i)} - x_{approx}^{(i)}\|^2$

Total variation in the data:  $\frac{1}{m} \sum_{i=1}^m \|x^{(i)}\|^2$

Typically, choose  $k$  to be smallest value so that

$$\begin{aligned} &\rightarrow \frac{\frac{1}{m} \sum_{i=1}^m \|x^{(i)} - x_{approx}^{(i)}\|^2}{\frac{1}{m} \sum_{i=1}^m \|x^{(i)}\|^2} \leq \frac{\cancel{0.01}}{\cancel{0.05} \quad 0.10} \quad \frac{\cancel{(1\%)}}{\cancel{5\%} \quad (10\%)} \end{aligned}$$

$\rightarrow$  “99% of variance is retained”  
~~95%~~ 90%

# Choosing $k$ (number of principal components)

Algorithm:

Try PCA with  $k = 1$

~~$k=2$~~   ~~$k=3$~~   
 $k=4$   
 $\vdots$

Compute  $U_{reduce}, \underline{z}^{(1)}, \underline{z}^{(2)}, \dots, \underline{z}^{(m)}, x_{approx}^{(1)}, \dots, x_{approx}^{(m)}$

Check if

$$\frac{\frac{1}{m} \sum_{i=1}^m \|x^{(i)} - x_{approx}^{(i)}\|^2}{\frac{1}{m} \sum_{i=1}^m \|x^{(i)}\|^2} \leq 0.01?$$

$k=17$

# Choosing $k$ (number of principal components)

Algorithm:

Try PCA with  $k = 1$   ~~$k=2$~~   ~~$k=3$~~   $k=4$   $\vdots$

Compute  $U_{reduce}, z^{(1)}, z^{(2)}, \dots, z^{(m)}, x_{approx}^{(1)}, \dots, x_{approx}^{(m)}$

Check if

$$\frac{\frac{1}{m} \sum_{i=1}^m \|x^{(i)} - x_{approx}^{(i)}\|^2}{\frac{1}{m} \sum_{i=1}^m \|x^{(i)}\|^2} \leq 0.01?$$

$k=17$

$$\rightarrow [U, \boxed{S}, V] = \text{svd}(\text{Sigma})$$

$$S = \begin{bmatrix} s_{11} & & & \\ & s_{22} & & \\ & & s_{33} & \\ & & & \ddots \\ & & & & s_{nn} \end{bmatrix}$$

For given  $k$

$$1 - \frac{\sum_{i=1}^k s_{ii}}{\sum_{i=1}^n s_{ii}}$$

Windows'u Etkinleştir  
Windows'u etkinleştirmek için Ayarlar'a gidin.



# Choosing $k$ (number of principal components)

Algorithm:

Try PCA with  $k = 1$   ~~$k=2$~~   ~~$k=3$~~   $k=4$   $\vdots$

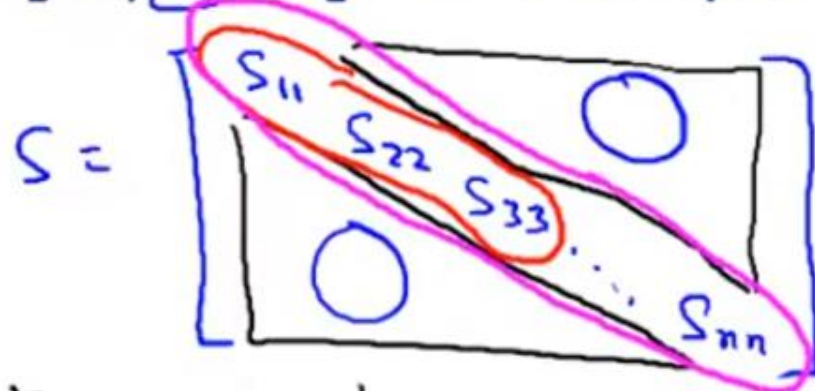
Compute  $U_{reduce}, z^{(1)}, z^{(2)}, \dots, z^{(m)}, x_{approx}^{(1)}, \dots, x_{approx}^{(m)}$

Check if

$$\frac{\frac{1}{m} \sum_{i=1}^m \|x^{(i)} - x_{approx}^{(i)}\|^2}{\frac{1}{m} \sum_{i=1}^m \|x^{(i)}\|^2} \leq 0.01?$$

$k=17$

$$\rightarrow [U, S, V] = \text{svd}(\text{Sigma})$$



For given  $k$

$$1 - \frac{\sum_{i=1}^k S_{ii}}{\sum_{i=1}^n S_{ii}}$$

$k=3$

# Choosing $k$ (number of principal components)

Algorithm:

Try PCA with  $k = 1$   ~~$k=2$~~   ~~$k=3$~~   $k=4$   $\vdots$

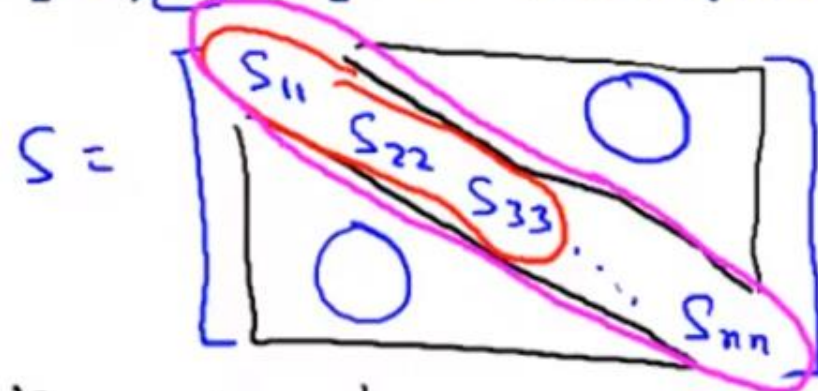
Compute  $U_{reduce}, z^{(1)}, z^{(2)}, \dots, z^{(m)}, x_{approx}^{(1)}, \dots, x_{approx}^{(m)}$

Check if

$$\frac{\frac{1}{m} \sum_{i=1}^m \|x^{(i)} - x_{approx}^{(i)}\|^2}{\frac{1}{m} \sum_{i=1}^m \|x^{(i)}\|^2} \leq 0.01?$$

$k=17$

$$\rightarrow [U, S, V] = \text{svd}(\text{Sigma})$$



For given  $k$

$$1 - \frac{\sum_{i=1}^k S_{ii}}{\sum_{i=1}^n S_{ii}} \leq 0.01$$

$$\frac{\sum_{i=1}^k S_{ii}}{\sum_{i=1}^n S_{ii}} \geq 0.99$$

Windows'u Etkinleştir  
Windows'u etkinleştirmek için Ayarlar'a gidin.



## Choosing $k$ (number of principal components)

→ `[U,S,V] = svd(Sigma)`

Pick smallest value of  $k$  for which

$$\left[ \frac{\sum_{i=1}^k S_{ii}}{\sum_{i=1}^m S_{ii}} \geq 0.99 \right.$$

(99% of variance retained)

- `from sklearn.decomposition import PCA`
- `# Make an instance of the Model`
- `pca = PCA(.99)`

# Advice for Applying PCA

*Principal Component Analysis*

Unsupervised Learning

# Application of PCA

- Compression

- Reduce memory/disk needed to store data
  - Speed up learning algorithm ←

Choose  $k$  by % of variance retain

- Visualization

$k=2$  or  $k=3$

Windows'u Etkinleştir  
Windows'u etkinleştirmek için Ayarlar'a gidin.

## Bad use of PCA: To prevent overfitting

→ Use  $z^{(i)}$  instead of  $x^{(i)}$  to reduce the number of features to  $k < n$ . — 1000 — 10000

Thus, fewer features, less likely to overfit.

Bad!

This might work OK, but isn't a good way to address overfitting. Use regularization instead.

$$\rightarrow \min_{\theta} \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2 \leftarrow$$

Windows'u Etkinleştir  
Windows'u etkinleştirmek için Ayarlar'a gidin.

## PCA is sometimes used where it shouldn't be

Design of ML system:

- - Get training set  $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$
- - ~~Run PCA to reduce  $x^{(i)}$  in dimension to get  $z^{(i)}$~~
- - Train logistic regression on  $\{(\cancel{z^{(1)}}), y^{(1)}), \dots, (\cancel{z^{(m)}}), y^{(m)})\}$
- - Test on test set: Map  $x_{test}^{(i)}$  to  $z_{test}^{(i)}$ . Run  $h_{\theta}(z)$  on  $\{(z_{test}^{(1)}, y_{test}^{(1)}), \dots, (z_{test}^{(m)}, y_{test}^{(m)})\}$

→ How about doing the whole thing without using PCA?

→ Before implementing PCA, first try running whatever you want to do with the original/raw data  $x^{(i)}$ . Only if that doesn't do what you want, then implement PCA and consider using  $z^{(i)}$ .

Windows'u Etkinleştir  
Windows'u etkinleştirmek için Ayarlar'a gidin.