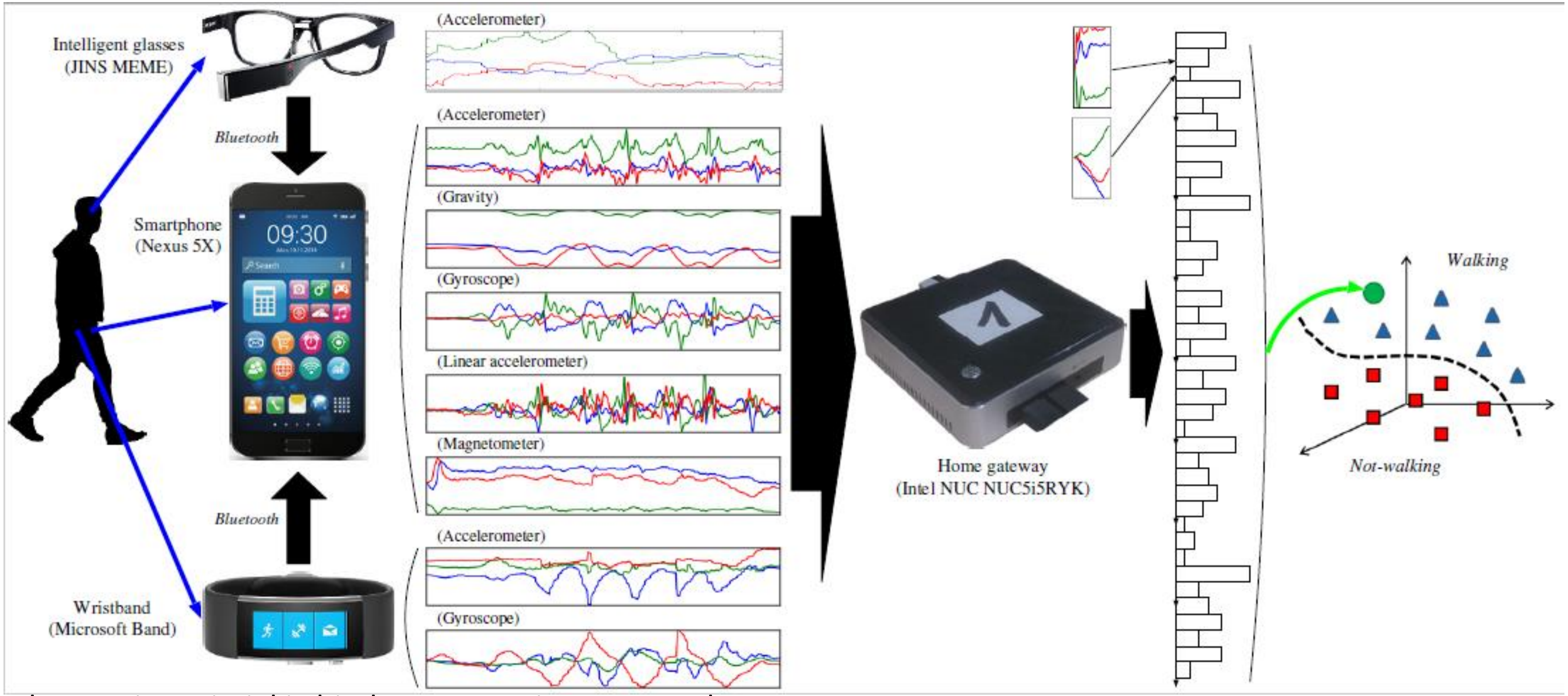
Data Science for Assistive Health Technologies

Dr. Muhammad Adeel Nisar

Assistant Professor – Department of IT,
Faculty of Computing and Information Technology,
University of the Punjab, Lahore

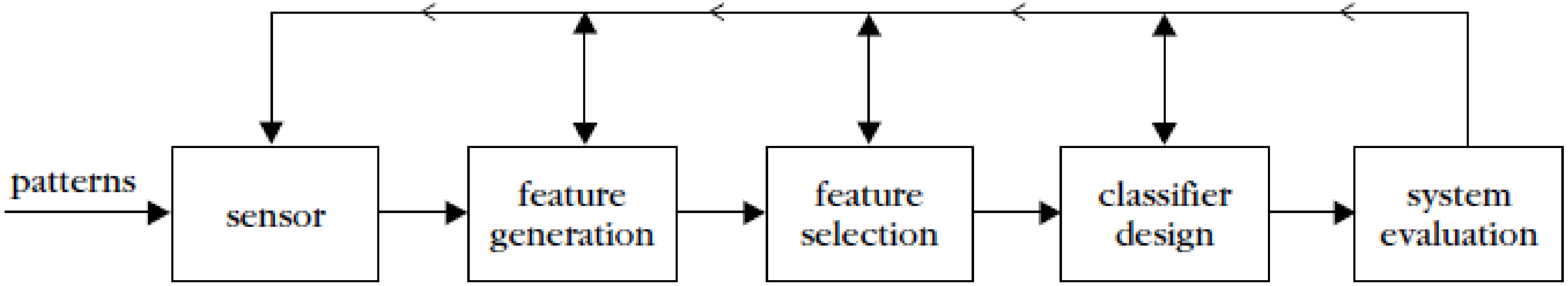
Contents

Recap – Sensor Data Acquisition System



Lukas Köping, Kimiaki Shirahama, Marcin Grzegorzek,
A general framework for sensor-based human activity recognition, Computers in Biology and Medicine, Volume 95, 2018, Pages 248-260, ISSN 0010-4825, https://doi.org/10.1016/j.compbiomed.2017.12.025.

Basic Stages of Pattern Analysis



Feature Generation – Introductory Statements

- Given a set of measurements, the goal is to discover compact and informative representations of the obtained data – features.

- The representations are generated after processing a large amount of sensory data.

- Measurements are transformed to a new set of features.

- In good features, the classification-related information is “squeezed” in a relatively small number of features.

Feature Generation – Introductory Statements

- Appropriately chosen feature transform can exploit and remove information redundancies.
- For example, using image pixels as features would be highly inefficient as pixels have a large degree of correlation.
- However, for instance, the Fourier transform turns out to be much more efficient for feature extraction.
- Fourier transform is just one of the tools from a palette of possible transforms.

Manual Feature Engineering vs. Feature Learning

- **Manual Feature Engineering:** features manually selected by experts from a certain application domain.

- **Feature Learning:** a set of techniques allowing a system to automatically discover the raw data representation needed for classification.

Manual Feature Engineering

Hand-Crafted Features		
Maximum	Percentile 50	First-order mean
Minimum	Percentile 80	Norm of the first-order mean
Average	Interquartile	Second-order mean
Standard-deviation	Skewness	Norm of the second-order mean
Zero-crossing	Kurtosis	Spectral energy
Percentile 20	Auto-correlation	Spectral entropy

Li, F.; Shirahama, K.; Nisar, M.A.; Köping, L.; Grzegorzek, M. Comparison of Feature Learning Methods for Human Activity Recognition Using Wearable Sensors. *Sensors* **2018**, *18*, 679. https://doi.org/10.3390/s18020679

Supervised vs. Unsupervised Feature Learning

- In **supervised feature learning,** features are learned using labelled input data. Examples include supervised neural networks, multilayer perceptron and (supervised) dictionary learning.

- In **unsupervised feature learning,** features are learned with unlabeled input data. Examples include dictionary learning, independent component analysis, autoencoders, matrix factorization and various forms of clustering.

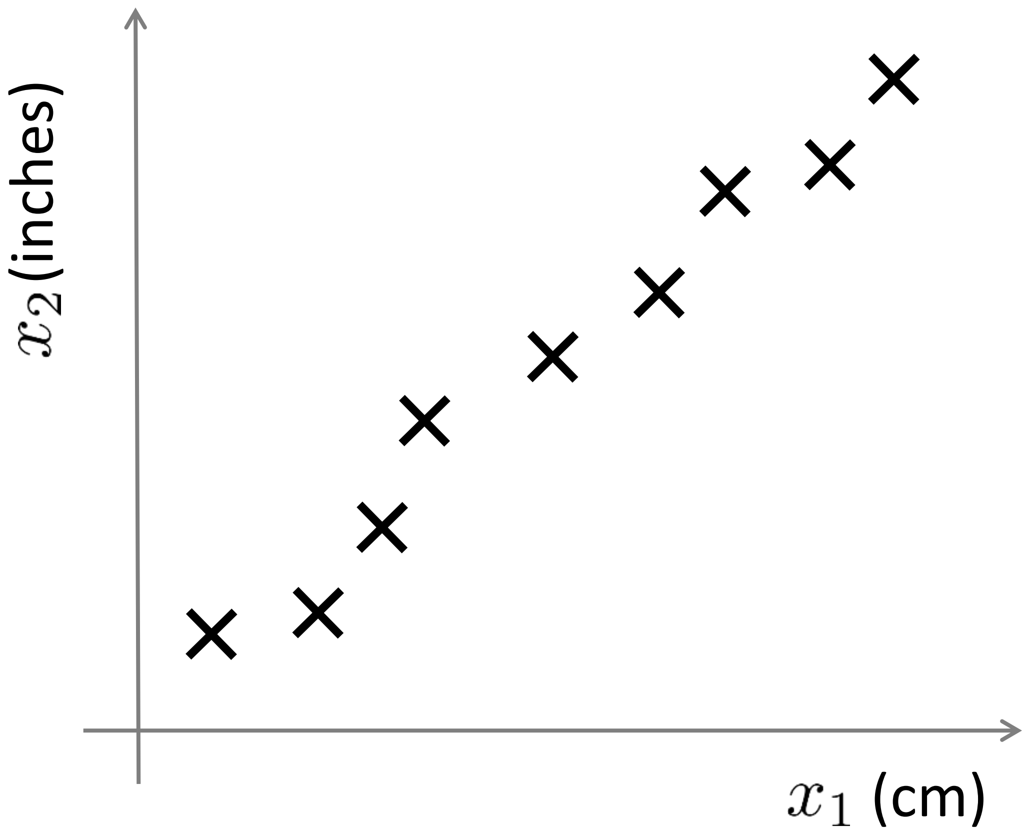
PCA – Introductory Statements

- The Principal Component Analysis (PCA) is one of the most popular methods for feature generation and dimensionality reduction in pattern recognition.

- The computation of the transformation matrix exploits the statistical information describing the data.

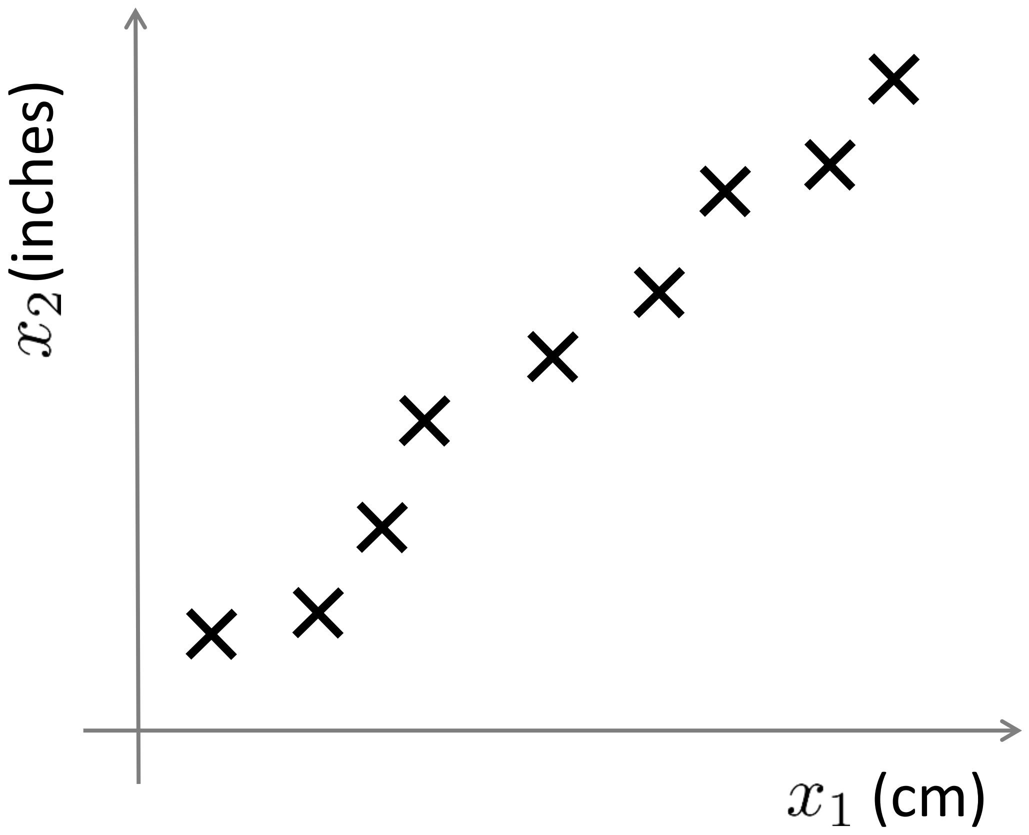
- The labels of the training samples are not used (unsupervised mode).

Data Compression



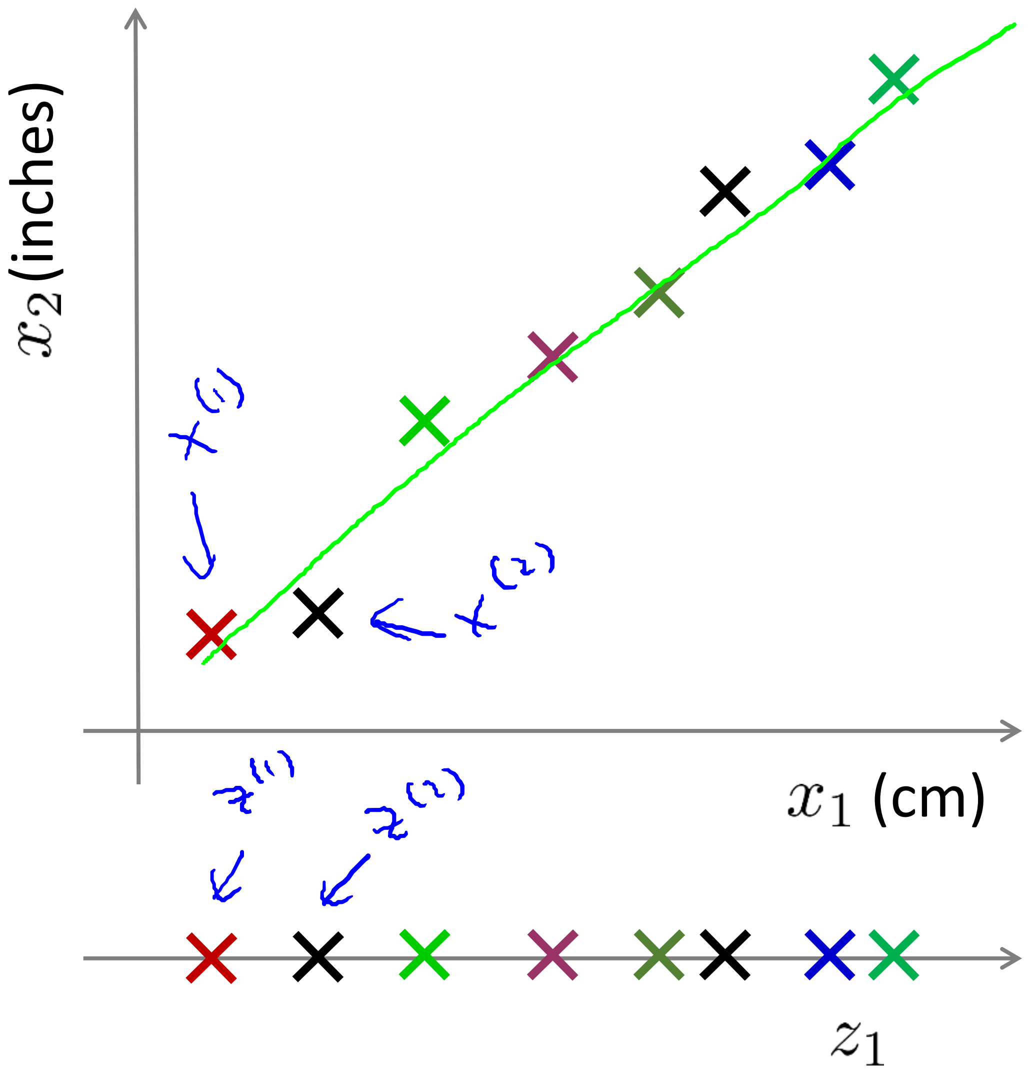
Reduce data from 2D to 1D

Data Compression



Reduce data from 2D to 1D

Data Compression



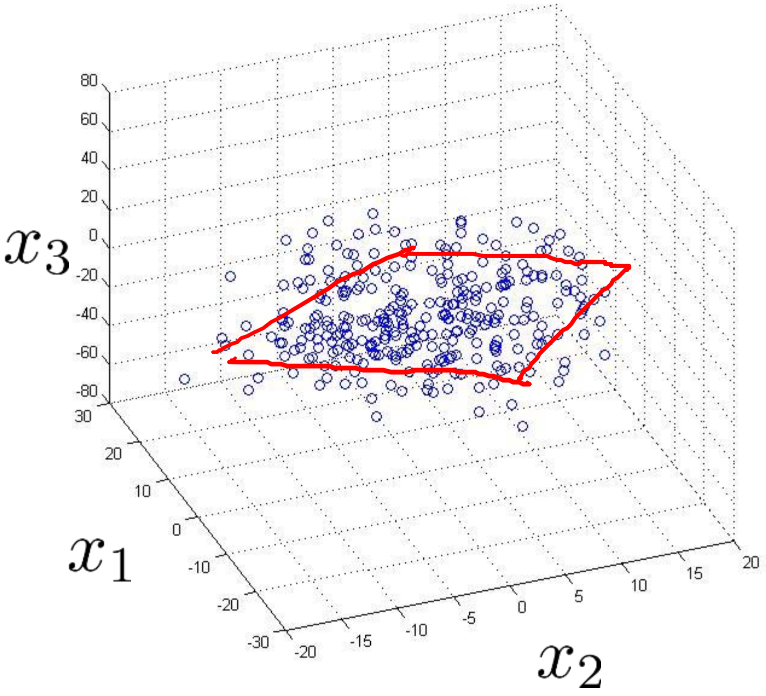
Reduce data from 2D to 1D

$$x^{(1)} \in \mathbb{R}^2 \rightarrow z^{(1)} \in \mathbb{R}$$
$$x^{(2)} \in \mathbb{R}^2 \rightarrow z^{(2)} \in \mathbb{R}$$
$$\vdots$$
$$x^{(m)} \in \mathbb{R}^2 \rightarrow z^{(m)} \in \mathbb{R}$$

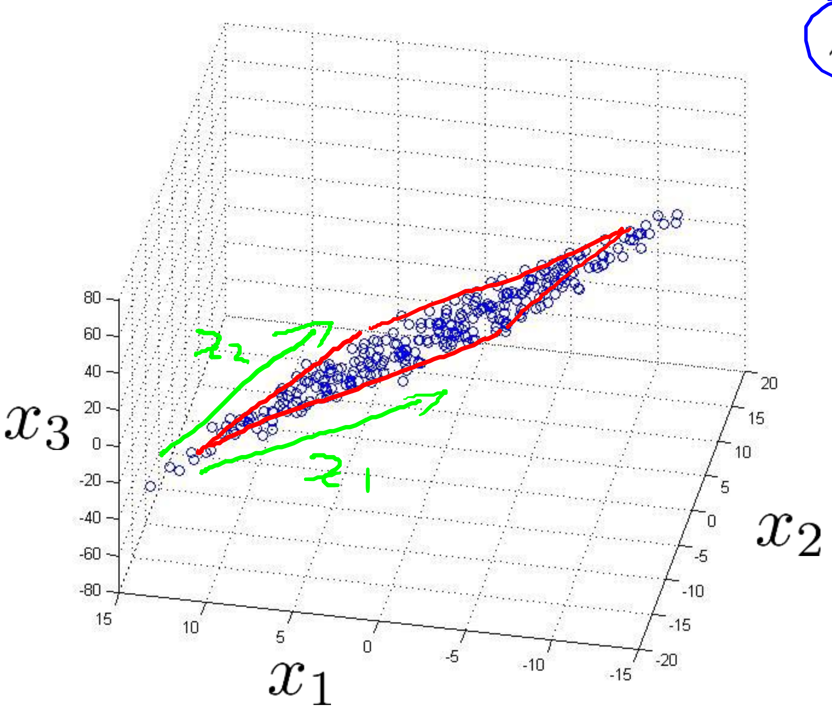
Data Compression

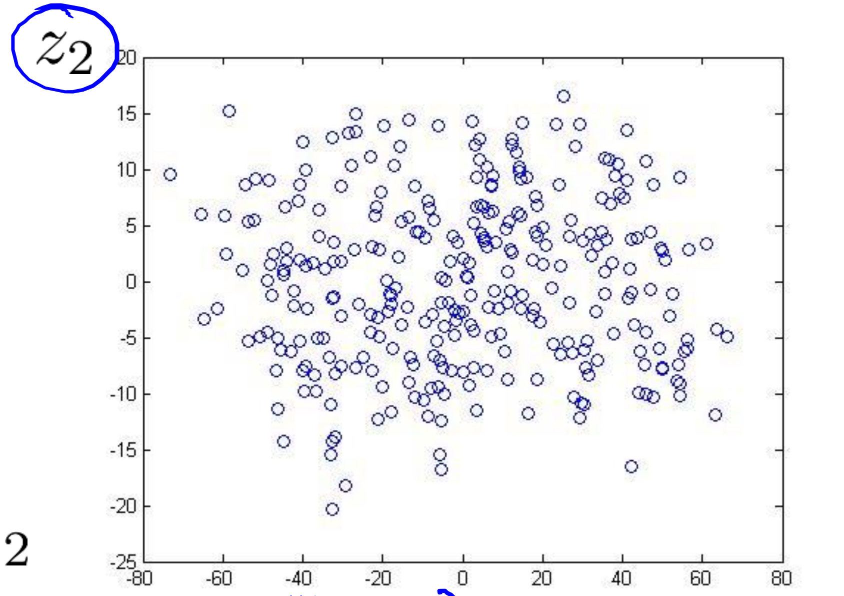
$1000D \rightarrow 100D$

Reduce data from 3D to 2D



$x^{(i)} \in \mathbb{R}^3$





$z^{(i)} \in \mathbb{R}^2$

$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \qquad z^{(i)} = \begin{bmatrix} z_1^{(i)} \\ z_2^{(i)} \end{bmatrix}$$

Data Visualization

$x \in \mathbb{R}^{50}$ $\quad x^{(i)} \in \mathbb{R}^{50}$

Country	x_1 GDP (trillions of US$)	x_2 Per capita GDP (thousands of intl. $)	x_3 Human Develop-ment Index	x_4 Life expectancy	x_5 Poverty Index (Gini as percentage)	x_6 Mean household income (thousands of US$)	…
Canada	1.577	39.17	0.908	80.7	32.6	67.293	…
China	5.878	7.54	0.687	73	46.9	10.22	…
India	1.632	3.41	0.547	64.7	36.8	0.735	…
Russia	1.48	19.84	0.755	65.5	39.9	0.72	…
Singapore	0.223	56.69	0.866	80	42.5	67.1	…
USA	14.527	46.86	0.91	78.3	40.8	84.3	…
…	…	…	…	…	…	…	

[resources from en.wikipedia.org]

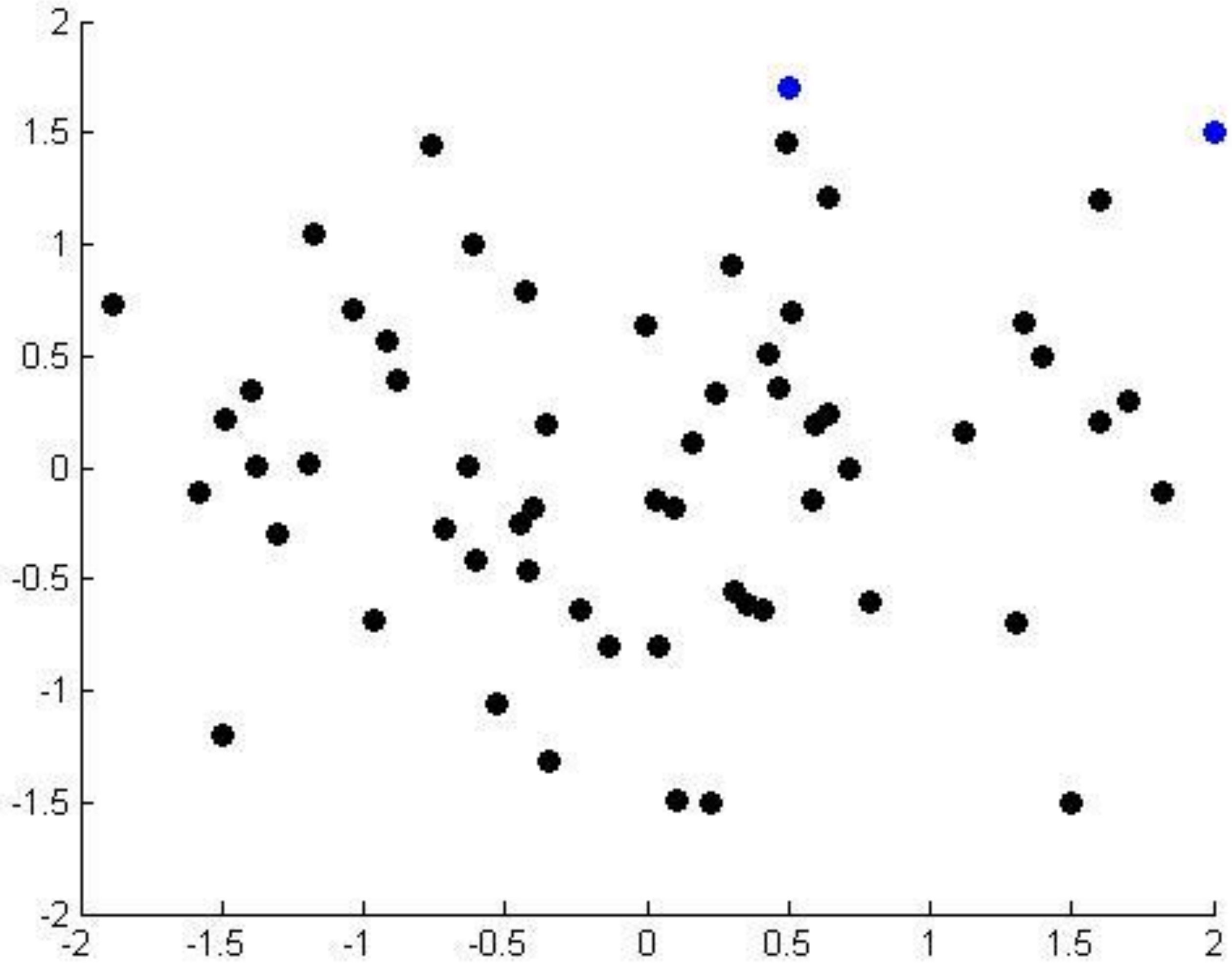
Data Visualization

Country	z_1	z_2
Canada	1.6	1.2
China	1.7	0.3
India	1.6	0.2
Russia	1.4	0.5
Singapore	0.5	1.7
USA	2	1.5
…	…	…

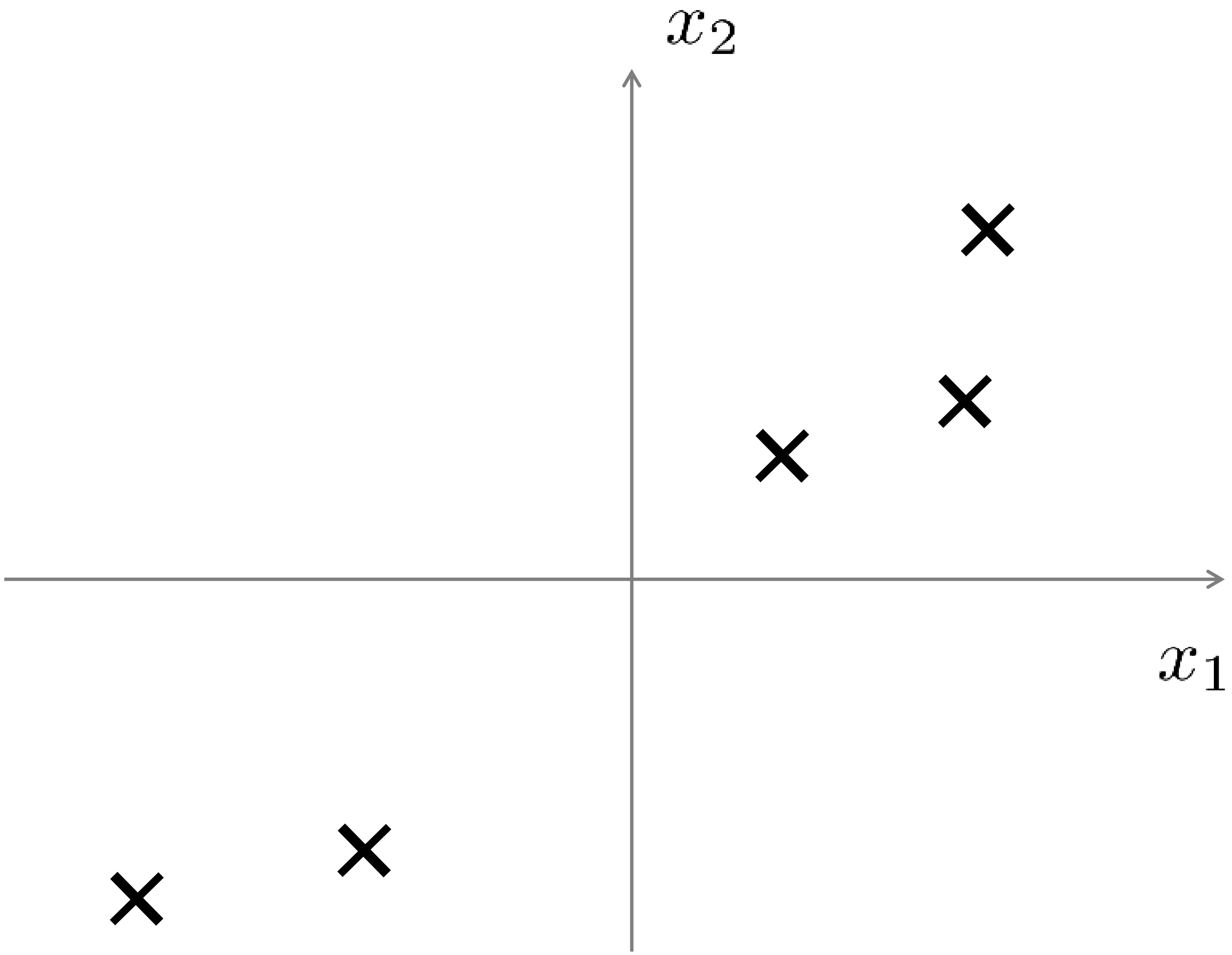
$z^{(i)} \in \mathbb{R}^2$

Reduce data from 50D to 2D

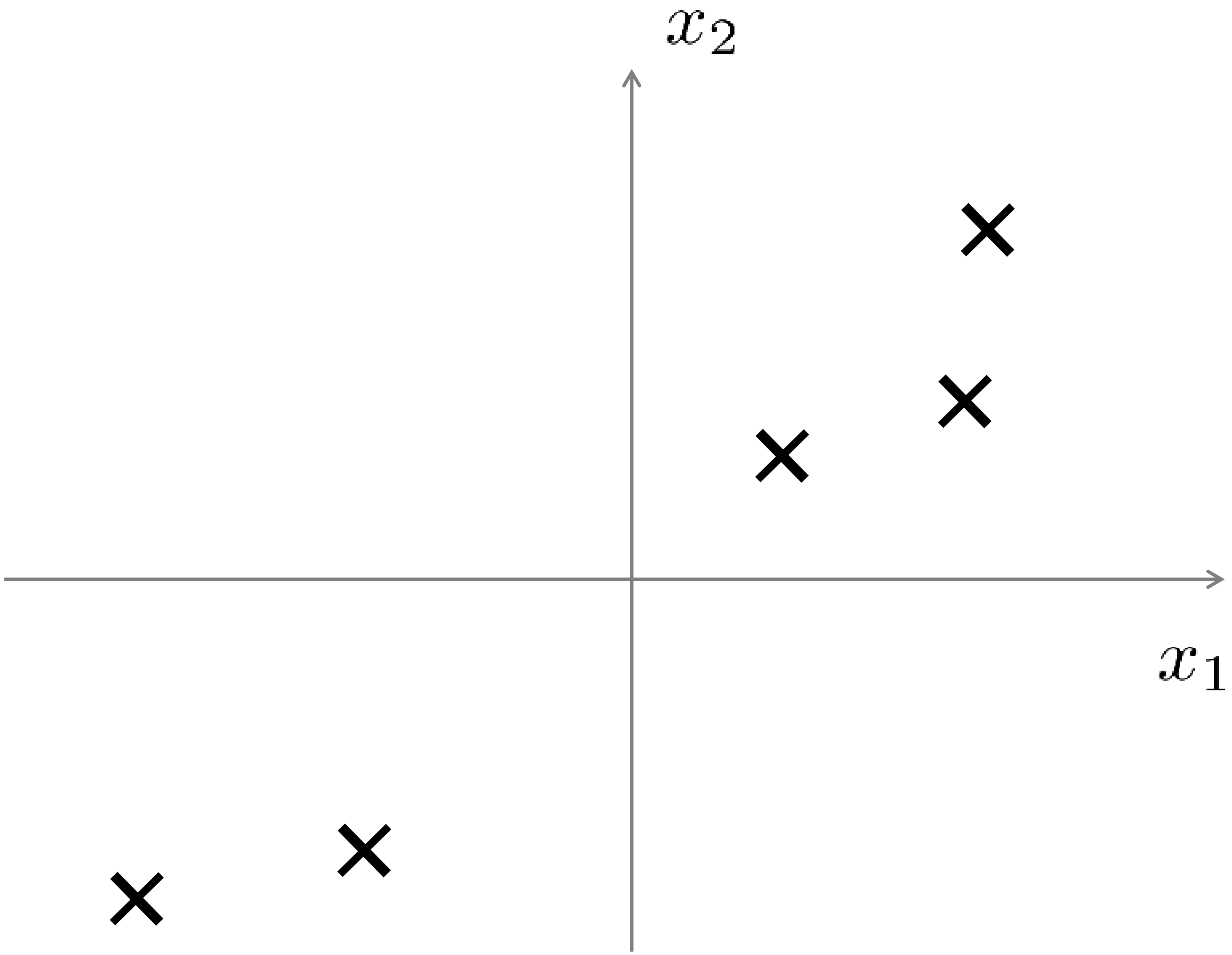
Data Visualization



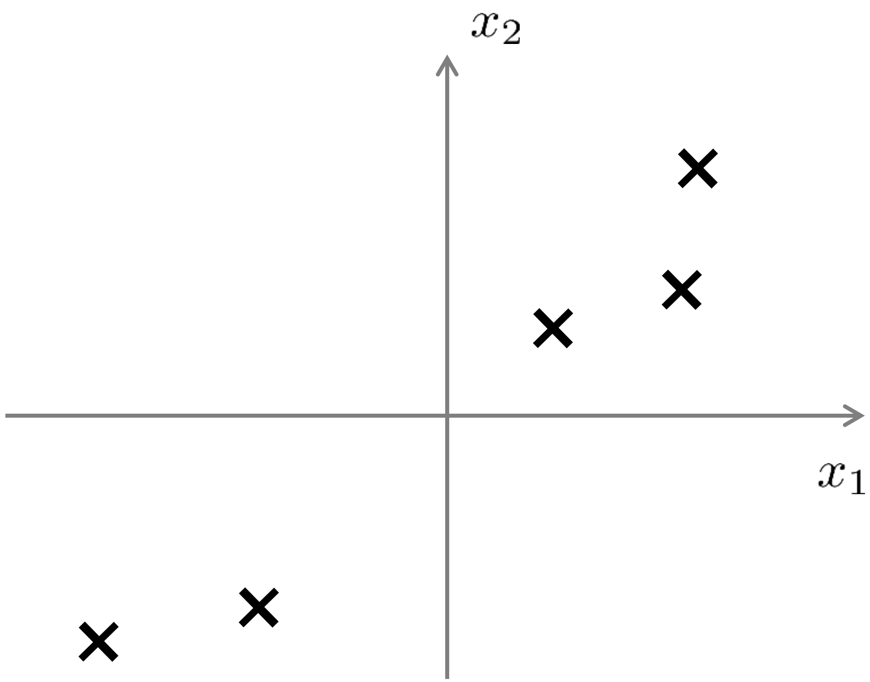
Principal Component Analysis (PCA) problem formulation



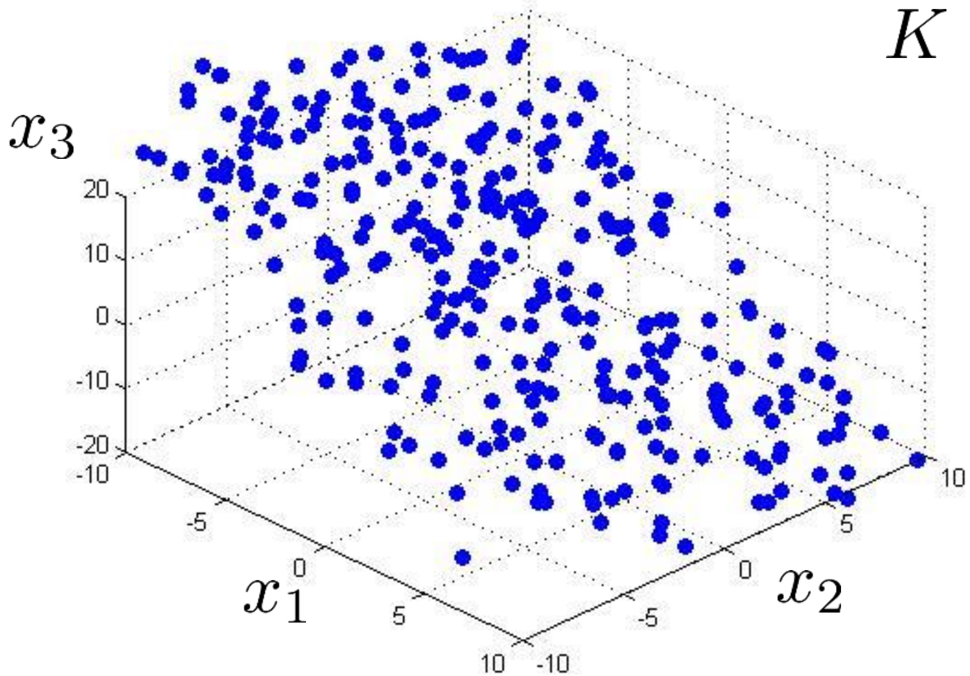
Principal Component Analysis (PCA) problem formulation



Principal Component Analysis (PCA) problem formulation

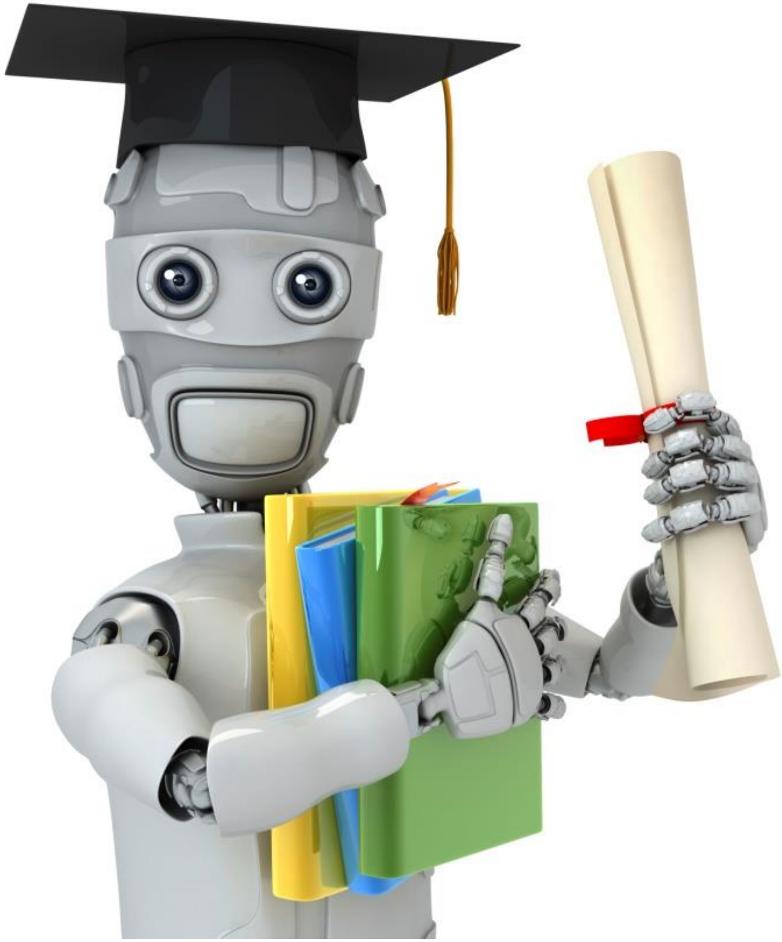


$3D \rightarrow 2D$
$K = 2$



Reduce from 2-dimension to 1-dimension: Find a direction (a vector $u^{(1)} \in \mathbb{R}^n$) onto which to project the data so as to minimize the projection error.

Reduce from n-dimension to k-dimension: Find k vectors $u^{(1)}, u^{(2)}, \ldots, u^{(k)}$ onto which to project the data, so as to minimize the projection error.



Machine Learning

Dimensionality Reduction

Principal Component Analysis algorithm

Data preprocessing

Training set: $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$

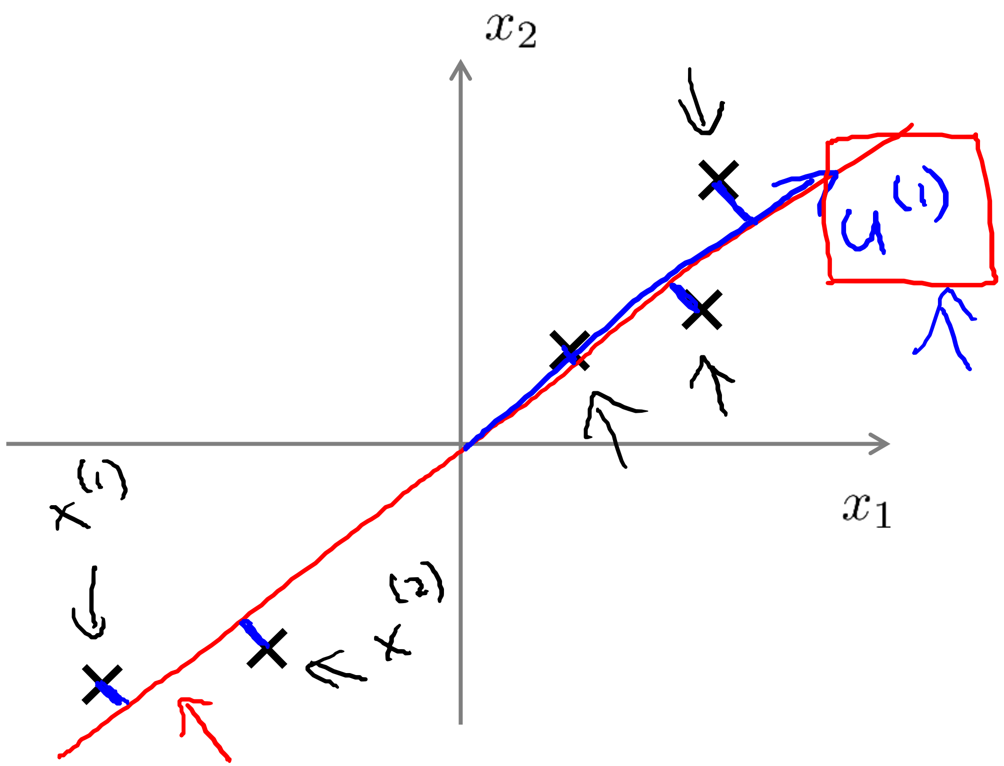
Preprocessing (feature scaling/mean normalization):

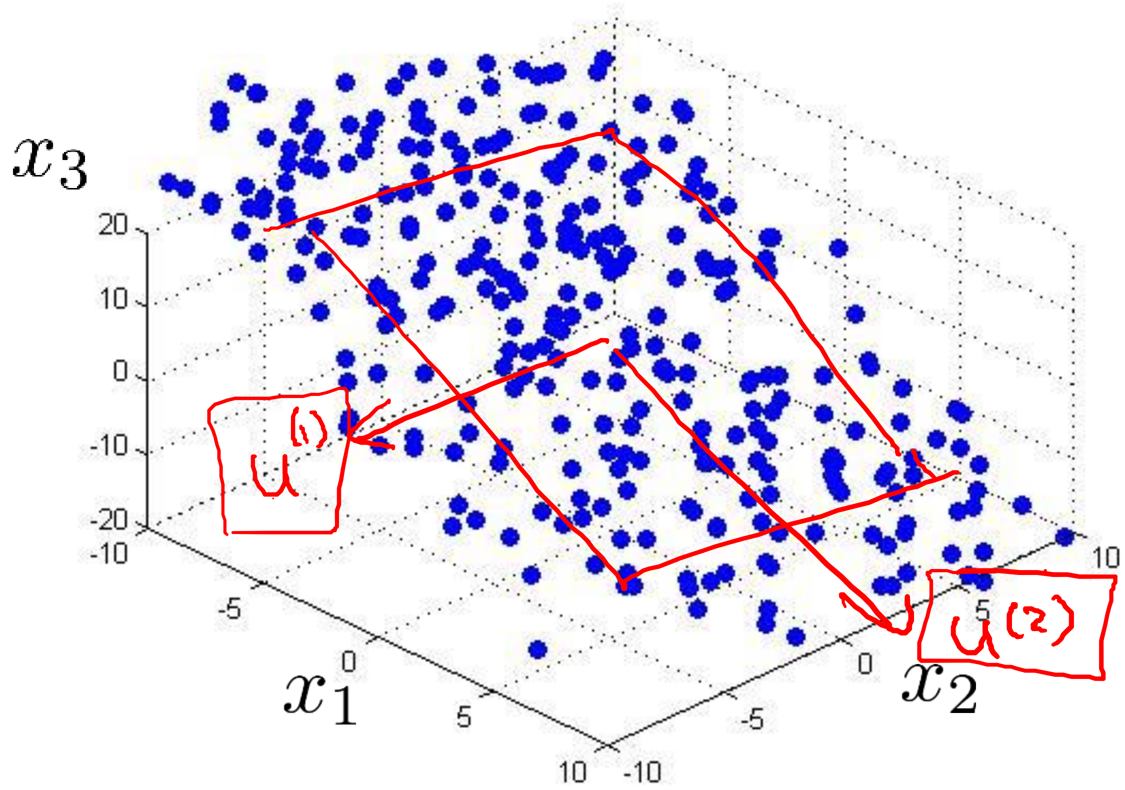
$$\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)}$$

Replace each $x_j^{(i)}$ with $x_j - \mu_j$

If different features on different scales (e.g., $x_1 =$ heart rate, $x_2 =$ skin conductance), scale features to have comparable range of values.

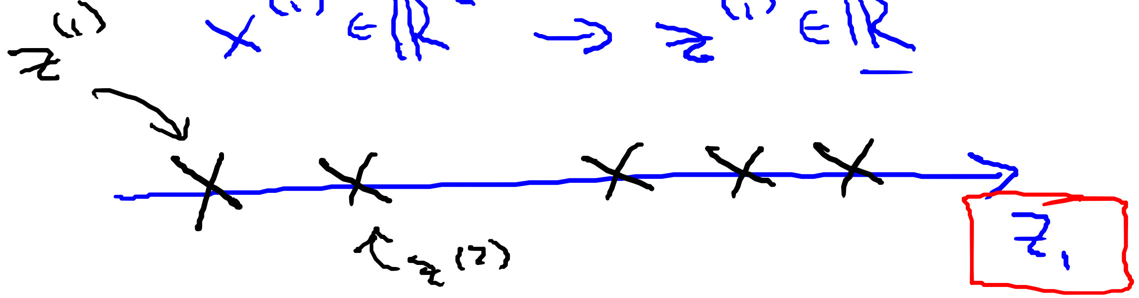
Principal Component Analysis (PCA) algorithm





Reduce data from 2D to 1D

$x^{(i)} \in \mathbb{R}^2 \rightarrow z^{(i)} \in \mathbb{R}$



Reduce data from 3D to 2D

$x^{(i)} \in \mathbb{R}^3 \rightarrow z^{(i)} \in \mathbb{R}^2$

$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$

Principal Component Analysis (PCA) algorithm

Reduce data from n-dimensions to k-dimensions

Compute “covariance matrix”:

$$\Sigma = \frac{1}{m}\sum_{i=1}^{n} \boxed{\underbrace{(x^{(i)})}_{n\times 1}\underbrace{(x^{(i)})^T}_{1\times n}} \quad n \times n$$

Sigma

Compute “eigenvectors” of matrix Σ:

```
[U,S,V] = svd(Sigma);
```

→ Singular value decomposition

eig(Sigma)

$n \times n$ matrix.

$$U = \begin{bmatrix} | & | & | & & | \\ \underbrace{u^{(1)} & u^{(2)} & u^{(3)}}_{k} & \dots & u^{(m)} \\ | & | & | & & | \end{bmatrix}$$

$U \in \mathbb{R}^{n \times n}$

$u^{(1)}, \dots, u^{(k)}$

Principal Component Analysis (PCA) algorithm

From `[U,S,V] = svd(Sigma)` we get:

$$U = \Bigg[\underbrace{\boxed{\begin{matrix} | & | \\ u^{(1)} & u^{(2)} \\ | & | \end{matrix}}}_{k} \quad \ldots \quad \begin{matrix} | \\ u^{(n)} \\ | \end{matrix}\Bigg] \in \mathbb{R}^{n\times n}$$

$$x \in \mathbb{R}^n \longrightarrow z \in \mathbb{R}^k$$

$$z^{(i)} = \underbrace{\begin{bmatrix} | & | & & | \\ u^{(1)} & u^{(2)} & \ldots & u^{(k)} \\ | & | & & | \end{bmatrix}}_{\substack{n\times k \\ U_{reduce}}}^T x^{(i)} = \underbrace{\underbrace{\begin{bmatrix} \text{---}\,(u^{(1)})^T\,\text{---} \\ \vdots \\ \text{---}\,(u^{(k)})^T\,\text{---} \end{bmatrix}}_{k\times n} \underbrace{x^{(i)}}_{n\times 1}}_{k\times 1}$$

$$z \in \mathbb{R}^k$$

Principal Component Analysis (PCA) algorithm summary

- After mean normalization (ensure every feature has zero mean) and optionally feature scaling:

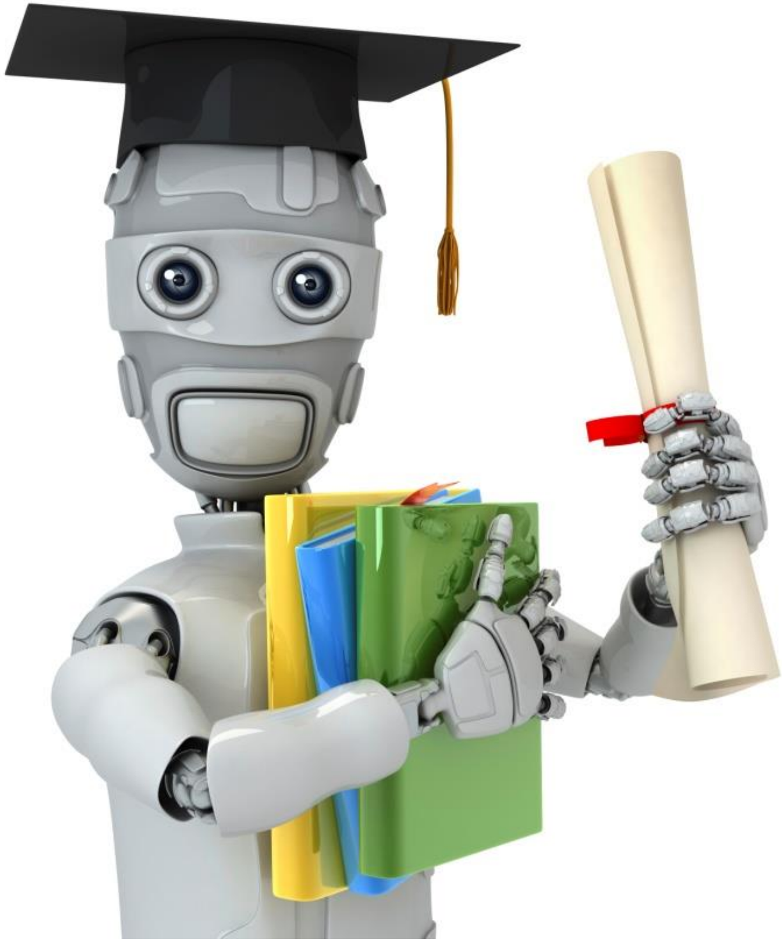
$$\texttt{Sigma} = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)})(x^{(i)})^T$$

$$\underline{X} = \begin{bmatrix} — & x^{(1)T} & — \\ & \vdots & \\ — & x^{(m)T} & — \end{bmatrix}$$

$$\boxed{\texttt{Sigma = (1/m) * X' * X;}}$$

```
[U,S,V] = svd(Sigma);
Ureduce = U(:,1:k);
z = Ureduce'*x;
```

$x \in \mathbb{R}^n$ $\cancel{x_0 = 1}$

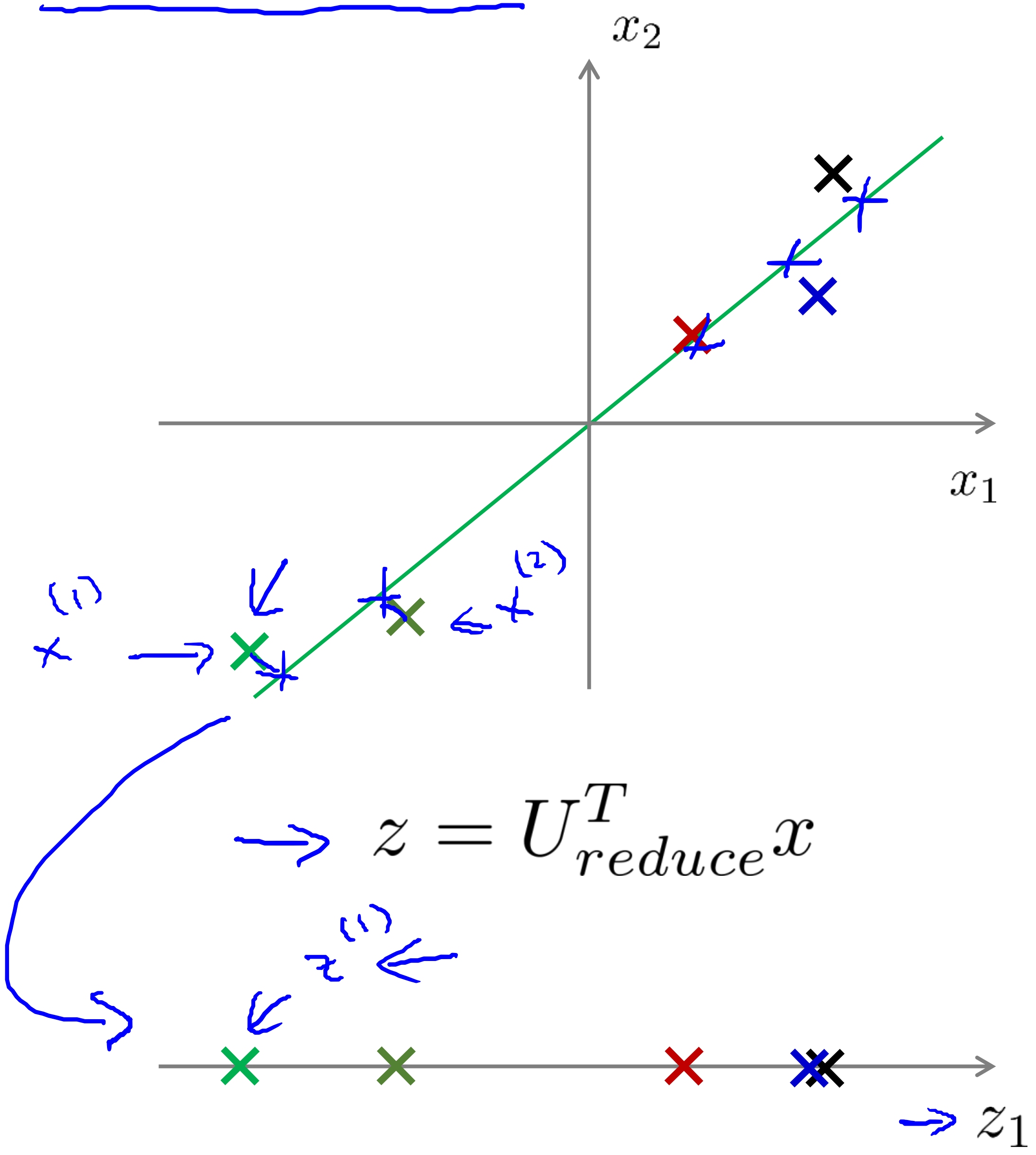


Machine Learning

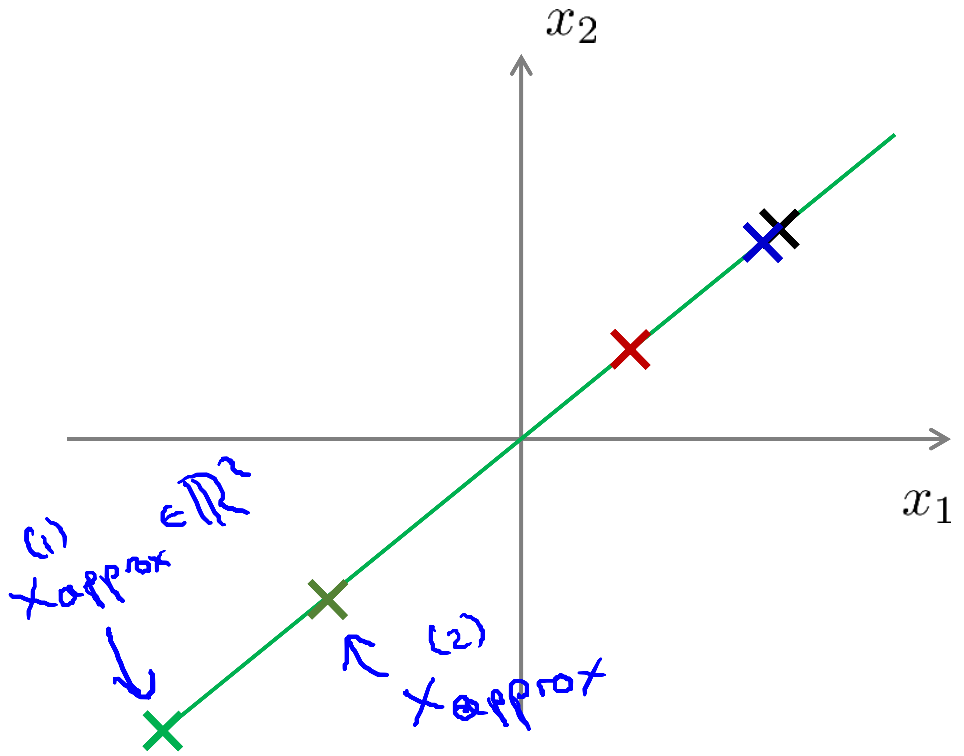
Dimensionality Reduction

Reconstruction from compressed representation

Reconstruction from compressed representation



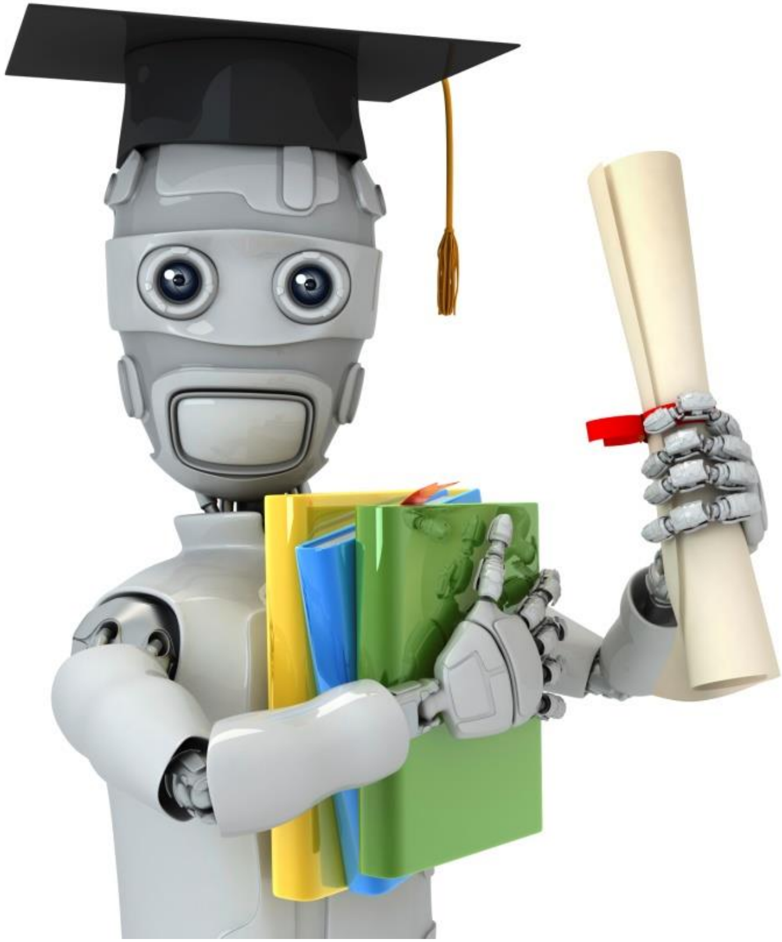
$$z = U_{reduce}^T x$$



$z \in \mathbb{R} \rightarrow x \in \mathbb{R}^2$

$\tilde{x}$

$\underbrace{x_{approx}^{(1)}}_{\mathbb{R}^n} = \underbrace{\underbrace{U_{reduce}}_{n \times k} \cdot \underbrace{z^{(1)}}_{k \times 1}}_{n \times 1}$



Machine Learning

Dimensionality Reduction

Choosing the number of principal components

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

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

~~0.05~~ ~~5%~~

0.10 (10%)

"~~99%~~ of variance is retained"

~~95%~~ 90%

Choosing k (number of principal components)

Algorithm:

Try PCA with $\cancel{k=1}$ $\cancel{k=2}$ $\cancel{k=3}$ $k=4$ $\vdots$

Compute $U_{reduce}, z^{(1)}, z^{(2)}, \ldots, z^{(m)}, x^{(1)}_{approx}, \ldots, x^{(m)}_{approx}$

Check if

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

$k = 17$

```
[U,S,V] = svd(Sigma)
```

$$S = \begin{bmatrix} S_{11} & & & & 0 \\ & S_{22} & & & \\ & & S_{33} & & \\ & & & \ddots & \\ 0 & & & & S_{nn} \end{bmatrix}$$

$k=3$

For given k

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

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

Choosing k (number of principal components)

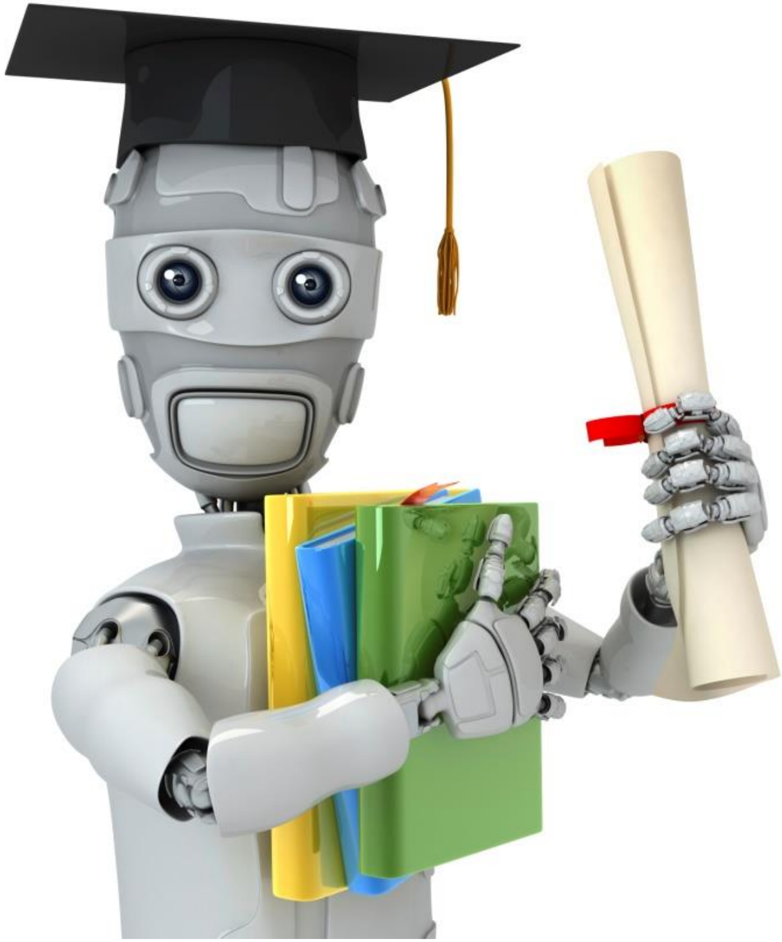
```
[U,S,V] = svd(Sigma)
```

Pick smallest value of k for which

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

$k=100$

(99% of variance retained)



Machine Learning

Dimensionality Reduction

Advice for applying PCA

Supervised learning speedup

$x^{(i)} \in \mathbb{R}^{10,000}$

100

100

$\boxed{(x^{(1)}, y^{(1)})}, (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})$

Extract inputs:

Unlabeled dataset: $x^{(1)}, x^{(2)}, \ldots, x^{(m)} \in \mathbb{R}^{10000}$

$\downarrow PCA$

x

$\downarrow$

z

U_{reduce}

$z^{(1)}, z^{(2)}, \ldots, z^{(m)} \in \mathbb{R}^{1000}$

New training set:

$\boxed{(z^{(1)}, y^{(1)})}, (z^{(2)}, y^{(2)}), \ldots, (z^{(m)}, y^{(m)})$

$h_\theta(z) = \frac{1}{1+e^{-\theta^T z}}$

Note: Mapping $x^{(i)} \rightarrow z^{(i)}$ should be defined by running PCA only on the training set. This mapping can be applied as well to the examples $x_{cv}^{(i)}$ and $x_{test}^{(i)}$ in the cross validation and test sets.

$x \rightarrow z$

Application of PCA

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

Choose k by % of variance retain

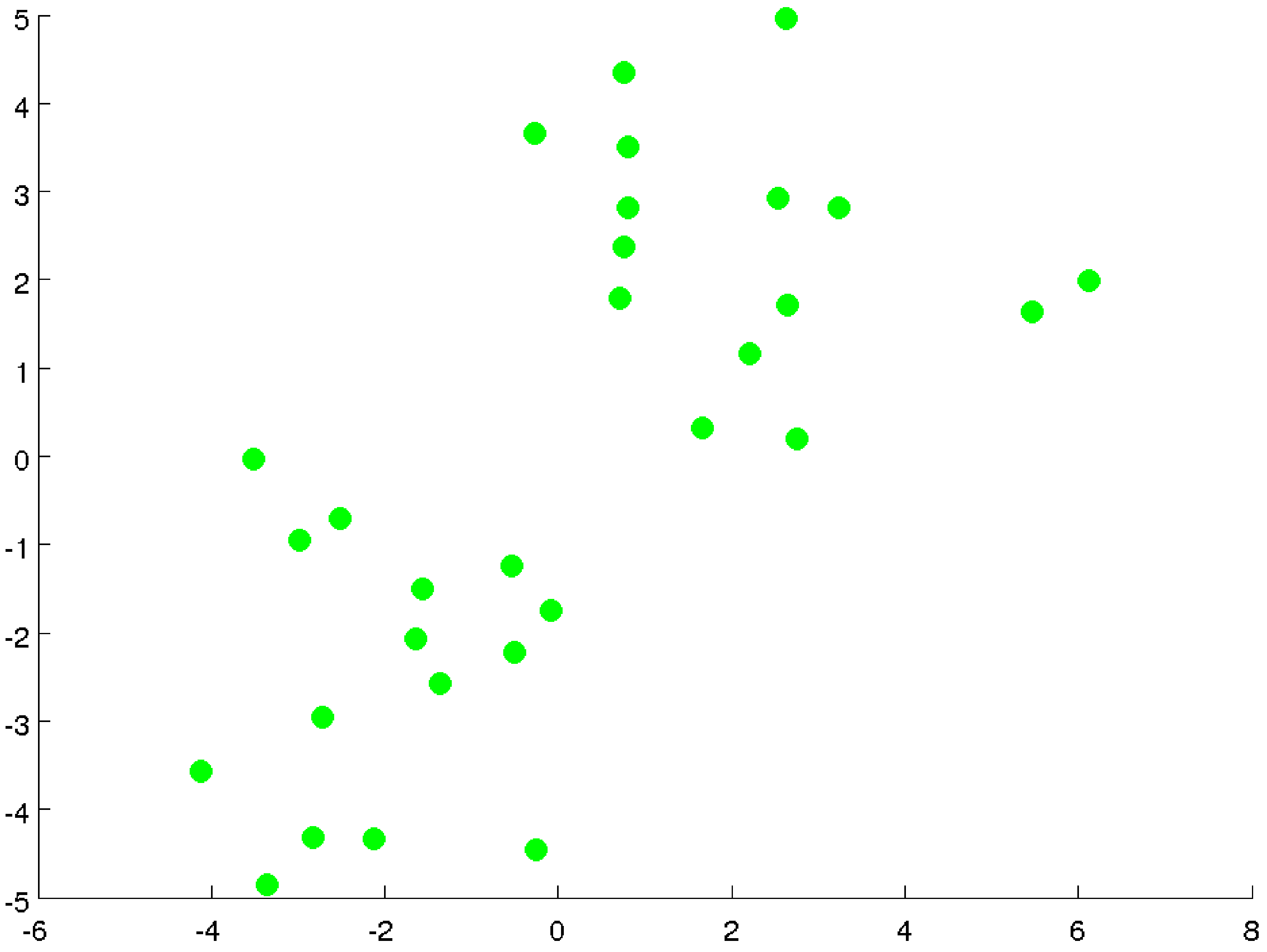
- Visualization

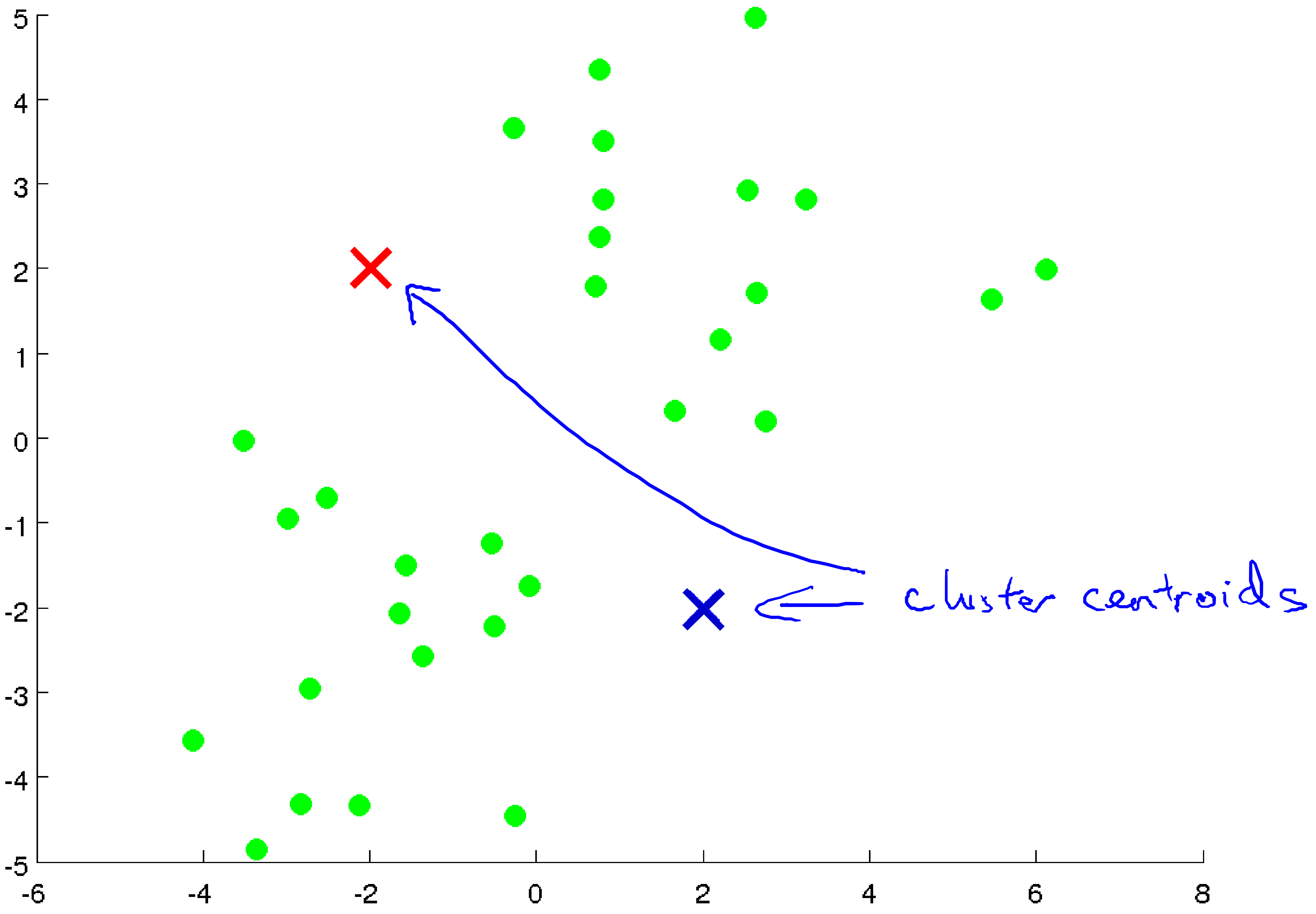
k = 2 or k = 3

Codebook Approach

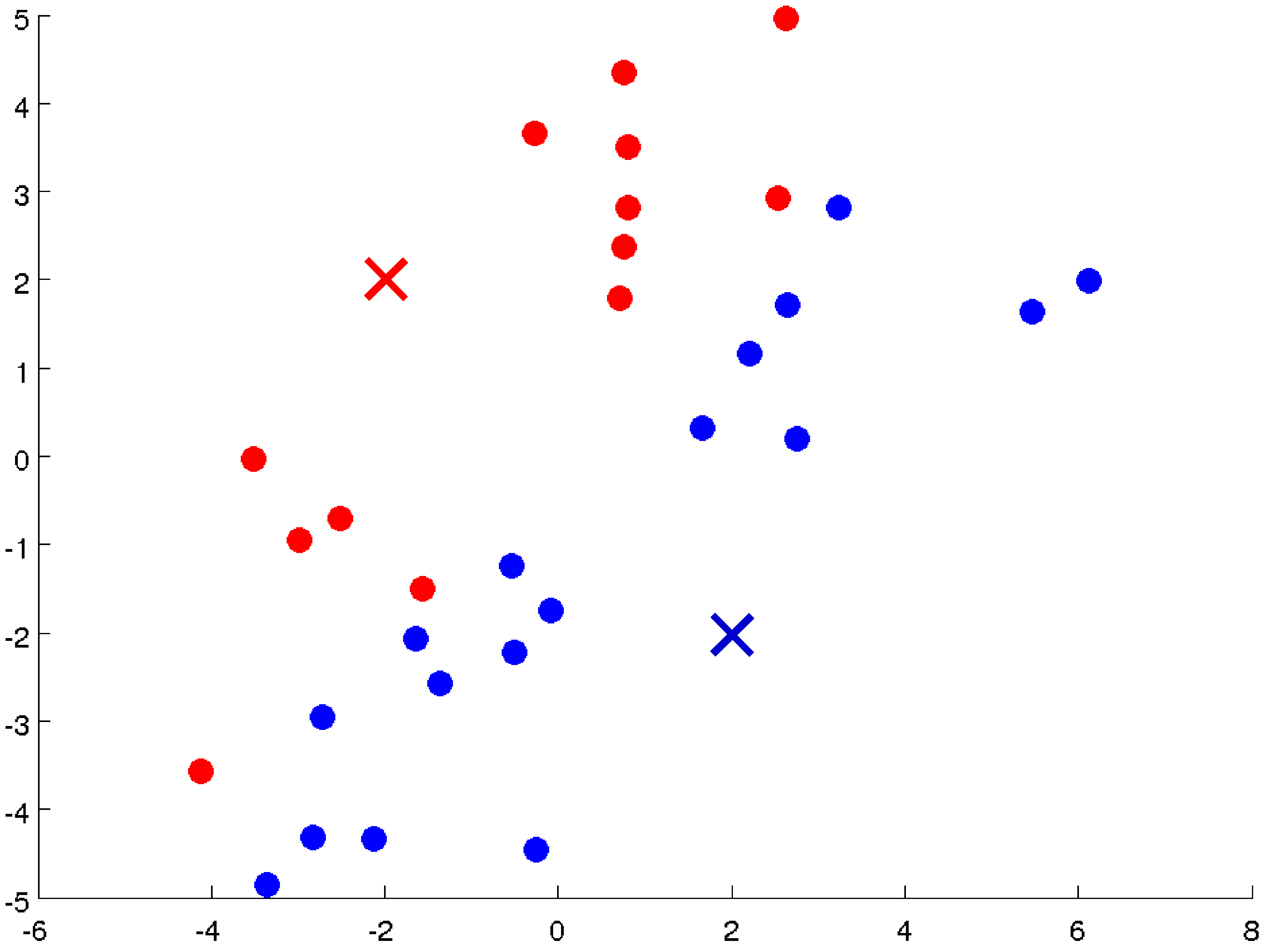
Clustering

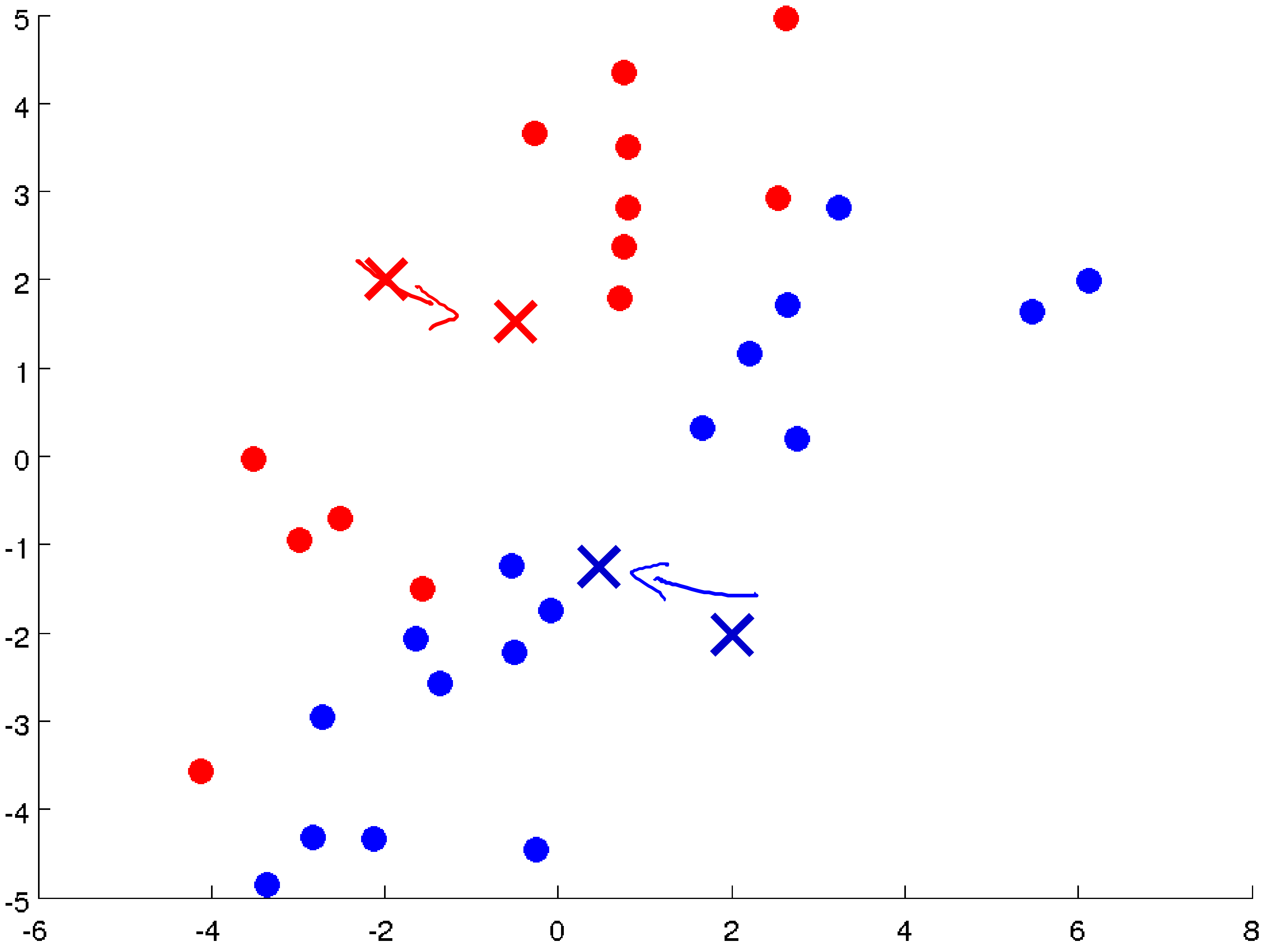
K-means algorithm

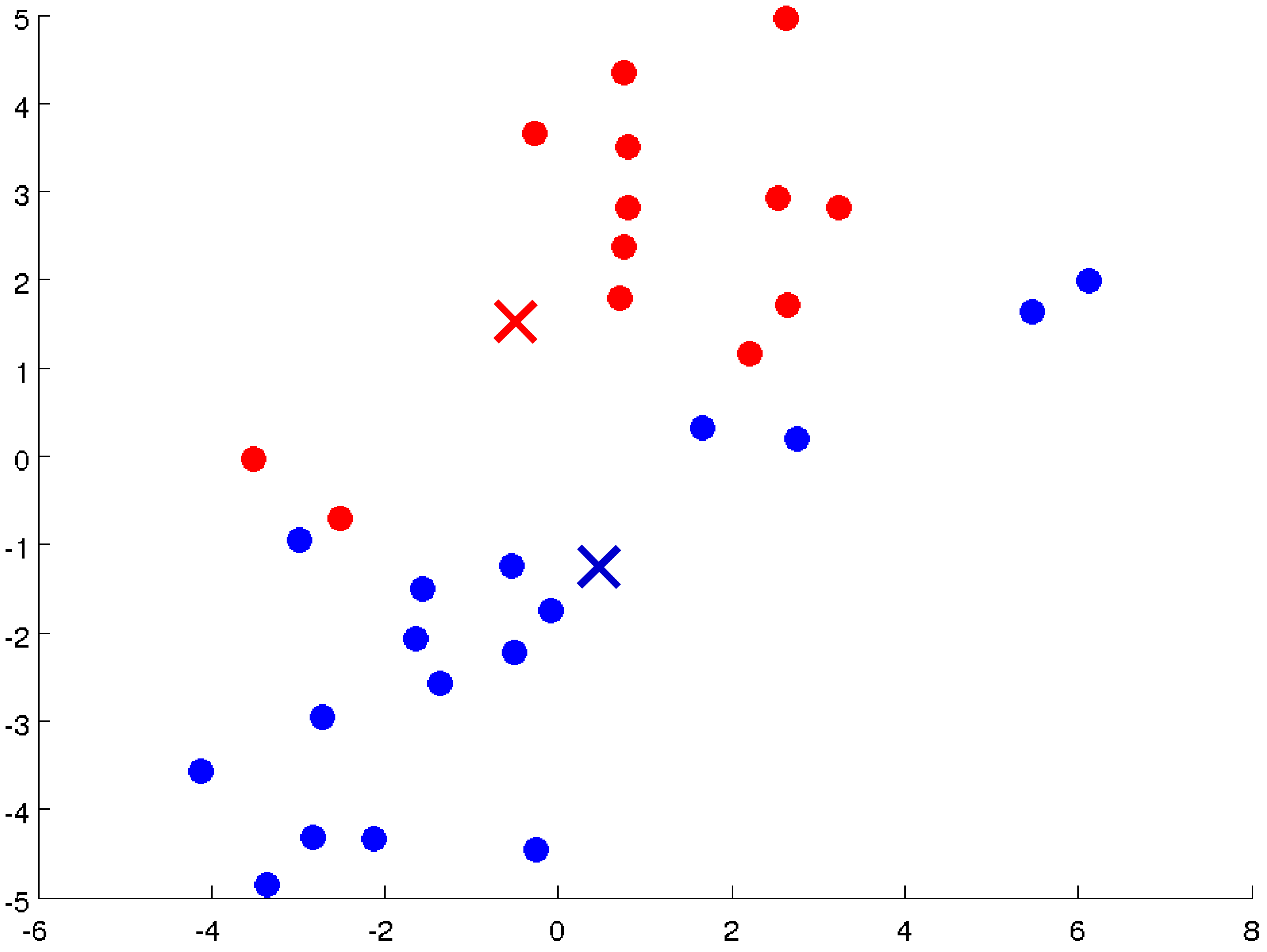

5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8

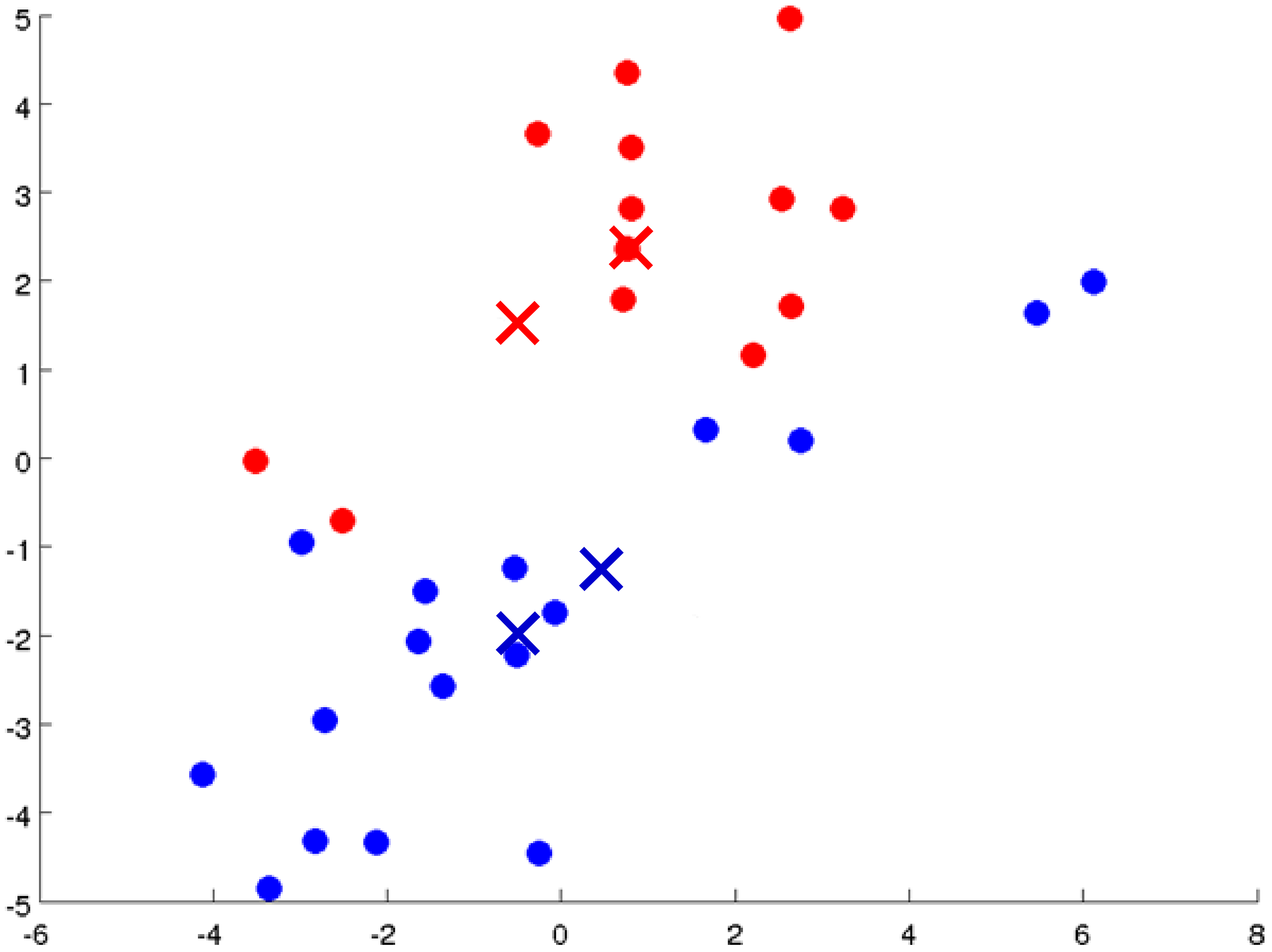


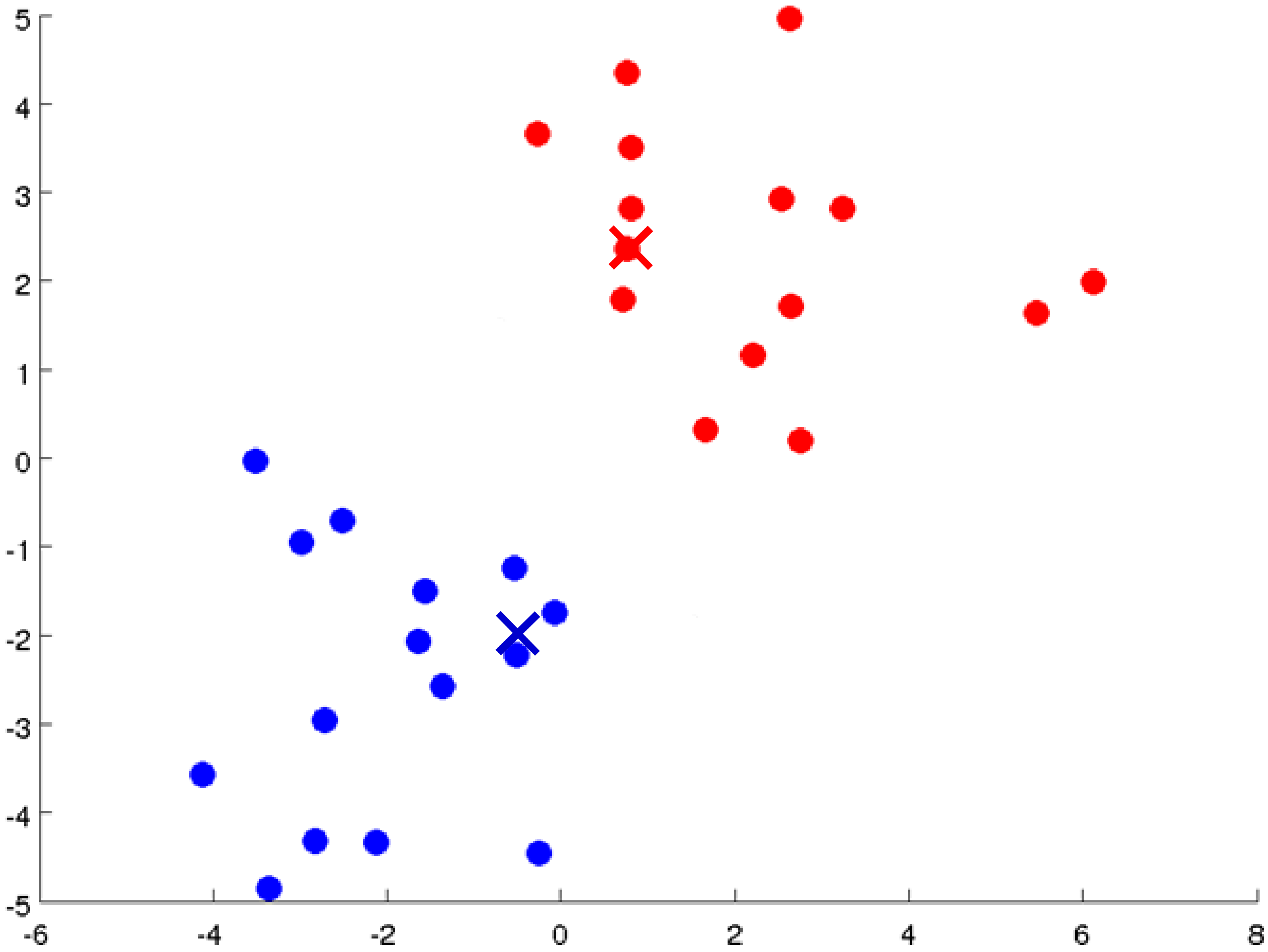
cluster centroids

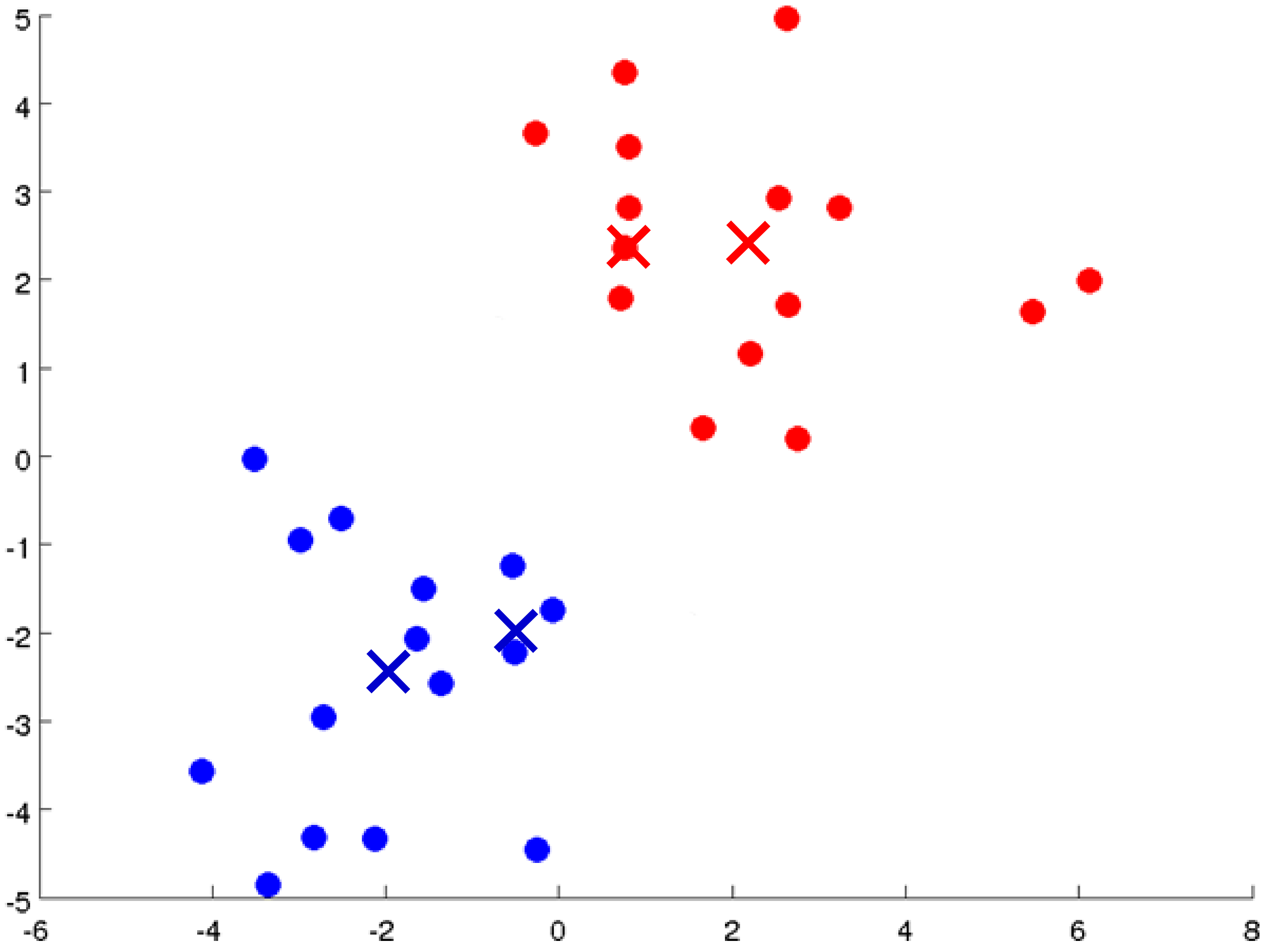

5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8


5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8

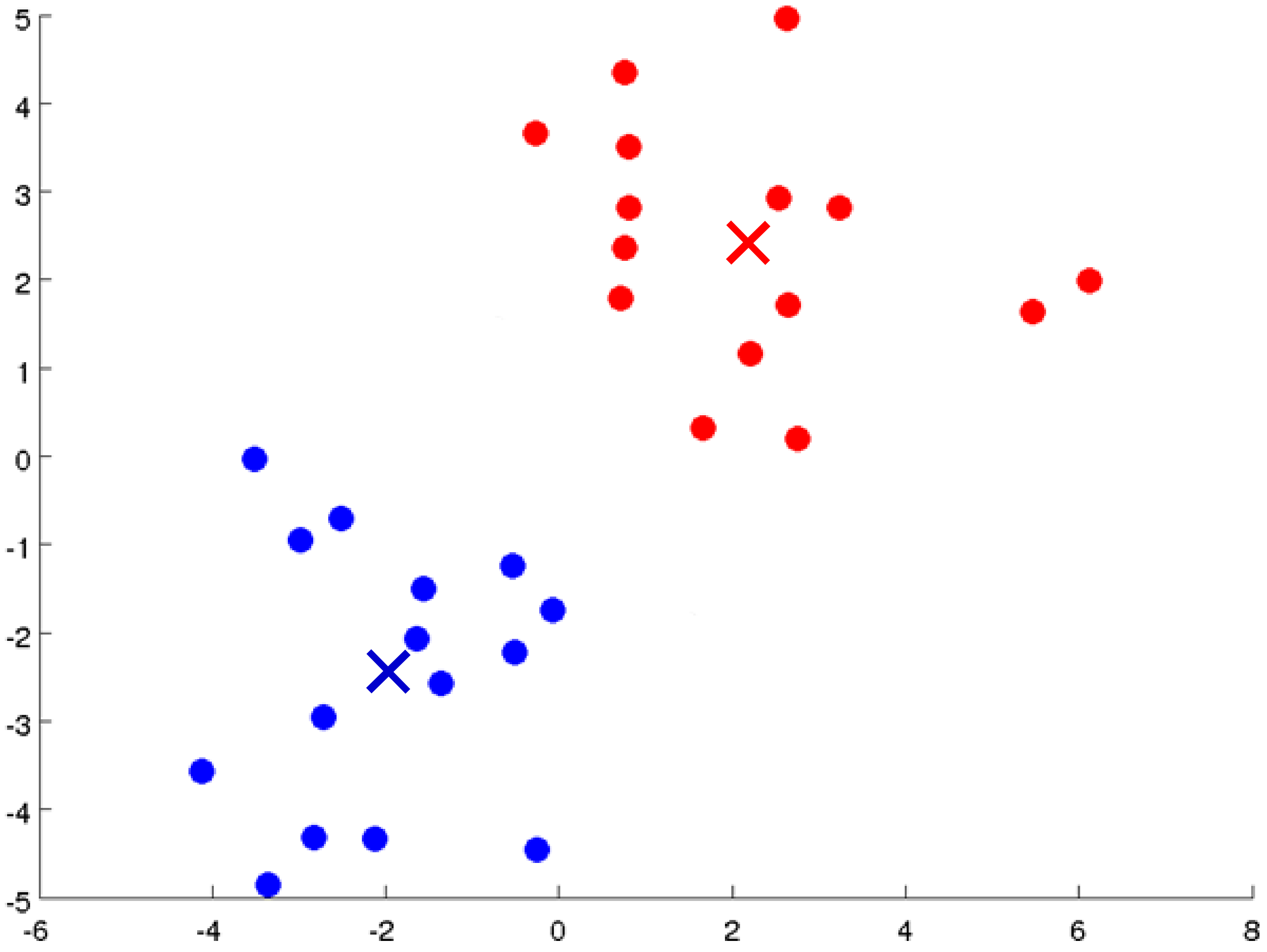

5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8


5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8


5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8



5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8


5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-4
-2
0
2
4
6
8

K-means algorithm

Input:

- K (number of clusters)
- Training set $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$

$x^{(i)} \in \mathbb{R}^n$

K-means algorithm

μ_1 $\times$ μ_2 $\times$

Randomly initialize K cluster centroids $\mu_1, \mu_2, \dots, \mu_K \in \mathbb{R}^n$

Repeat {

Cluster assignment step

for i = 1 to m

$c^{(i)}$:= index (from 1 to K) of cluster centroid closest to $x^{(i)}$

$\min_k \|x^{(i)} - \mu_k\|^2 \rightarrow c^{(i)}$

Move centroid

for k = 1 to K

μ_k := average (mean) of points assigned to cluster k

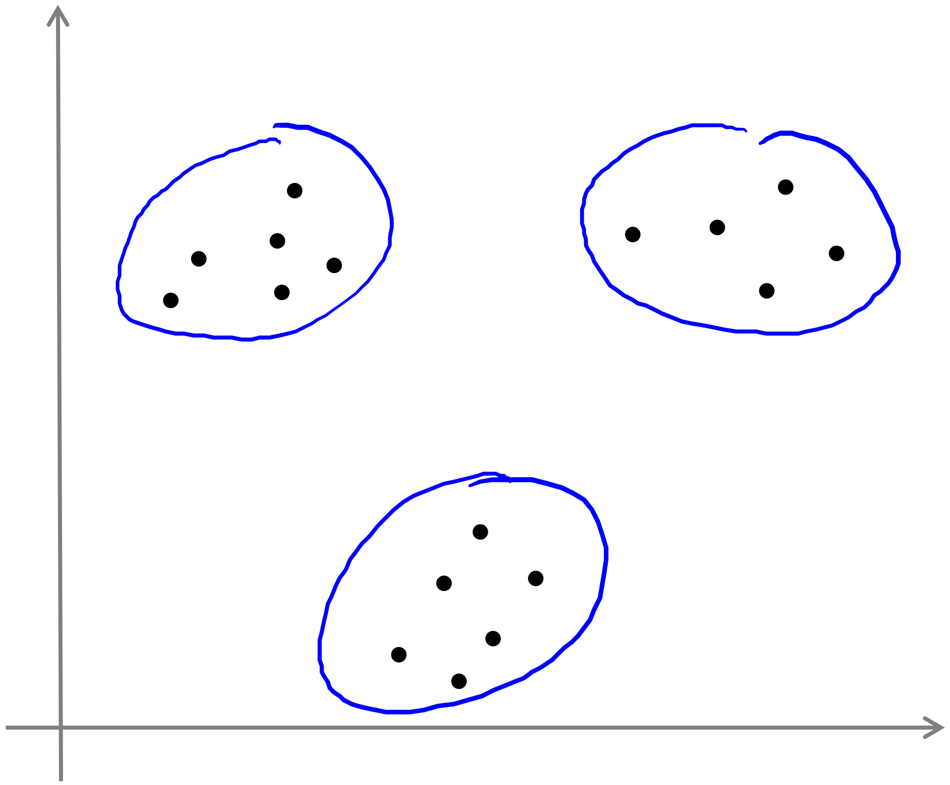
$x^{(1)}, x^{(5)}, x^{(6)}, x^{(10)} \rightarrow c^{(1)}=2, c^{(5)}=2, c^{(6)}=2, c^{(10)}=2$

$$\mu_2 = \frac{1}{4}\left[x^{(1)} + x^{(5)} + x^{(6)} + x^{(10)}\right] \in \mathbb{R}^n$$

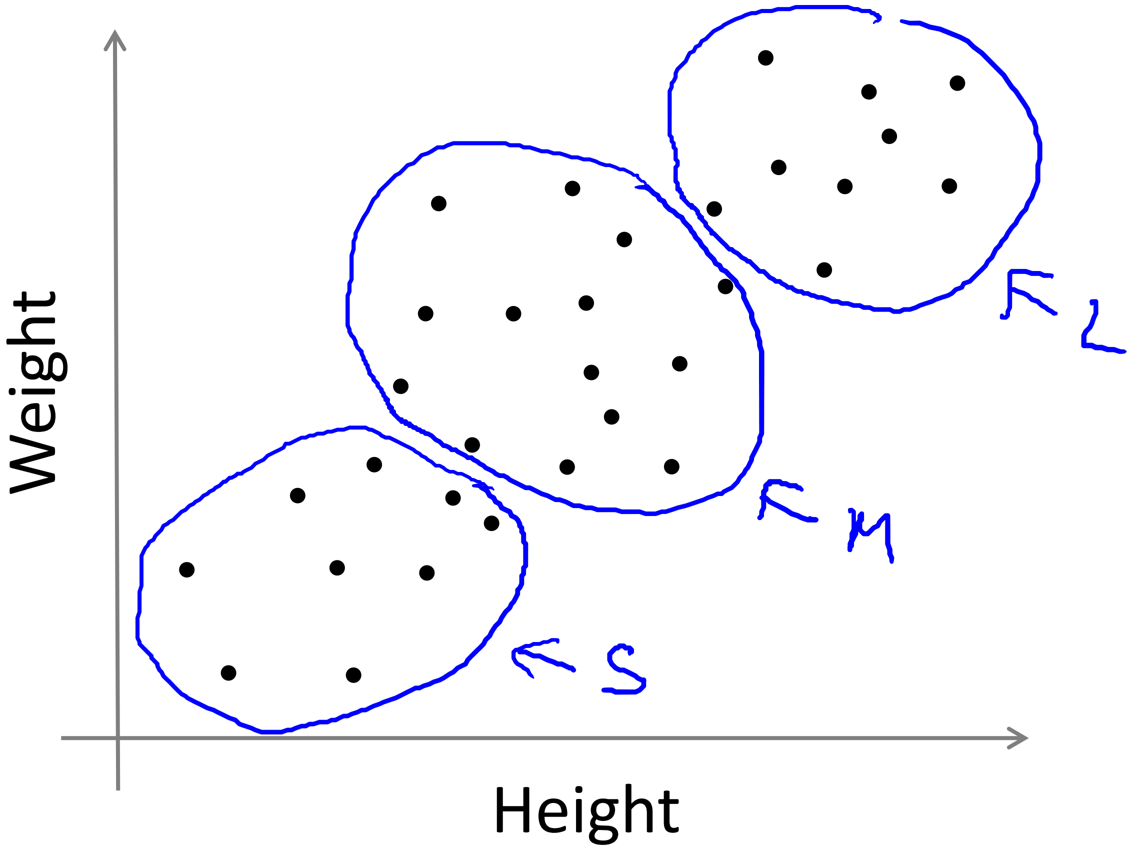
}

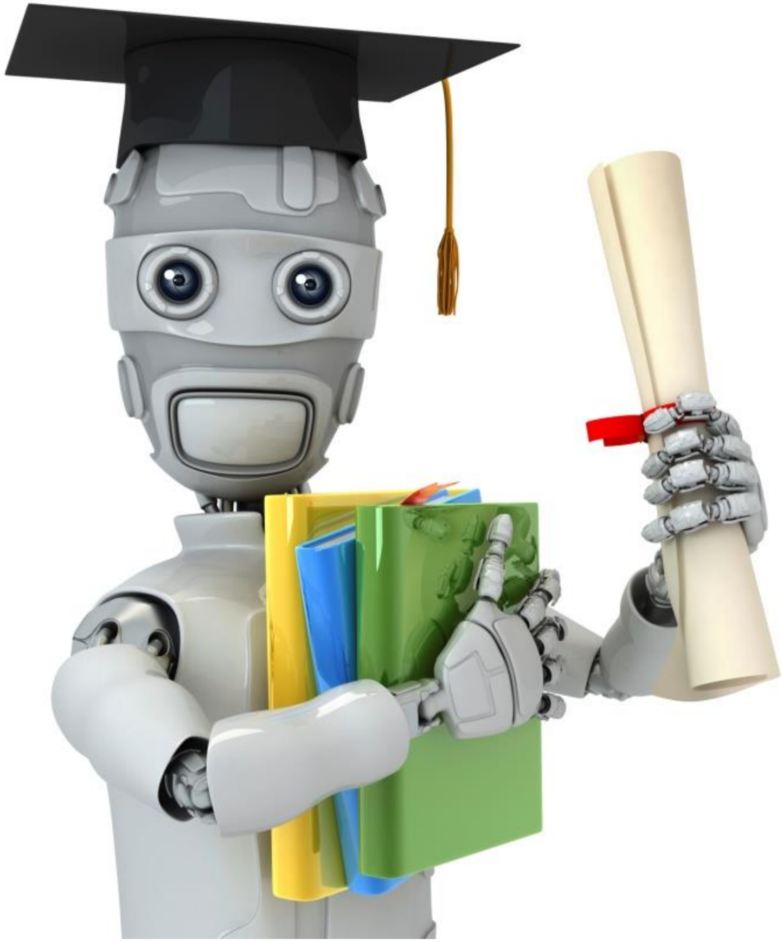
K-means for non-separated clusters

S, M, L



T-shirt sizing





Machine Learning

Clustering

Optimization objective

K-means optimization objective

$c^{(i)}$ = index of cluster (1,2,...,K) to which example $x^{(i)}$ is currently assigned

μ_k = cluster centroid $\underline{k}$ ($\mu_k \in \mathbb{R}^n$) K $k \in \{1,2,\ldots,K\}$

$\mu_{c^{(i)}}$ = cluster centroid of cluster to which example $x^{(i)}$ has been assigned

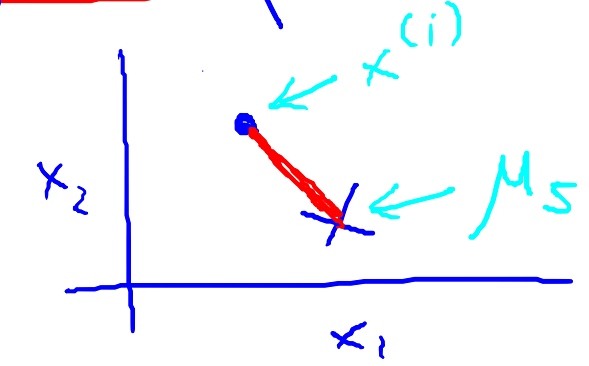
$x^{(i)} \rightarrow \underline{5}$ $\underline{c^{(i)} = 5}$ $\underline{\mu_{c^{(i)}}} = \mu_5$

Optimization objective:

$$J(\underline{c^{(1)},\ldots,c^{(m)},\mu_1,\ldots,\mu_K}) = \frac{1}{m}\sum_{i=1}^{m} \boxed{\|x^{(i)} - \mu_{c^{(i)}}\|^2}$$

$$\min_{\substack{c^{(1)},\ldots,c^{(m)},\\ \mu_1,\ldots,\mu_K}} J(c^{(1)},\ldots,c^{(m)},\mu_1,\ldots,\mu_K)$$

Distortion



K-means algorithm

Randomly initialize K cluster centroids $\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^n$

Cluster assignment step

Minimize $J(\ldots)$ wrt $\boxed{c^{(1)}, c^{(2)}, \ldots, c^{(m)}}$ ←

(holding $\mu_1, \ldots, \mu_K$ fixed)

Repeat {

for i = 1 to m

$c^{(i)}$:= index (from 1 to K) of cluster centroid closest to $x^{(i)}$

move centroid

for k = 1 to K

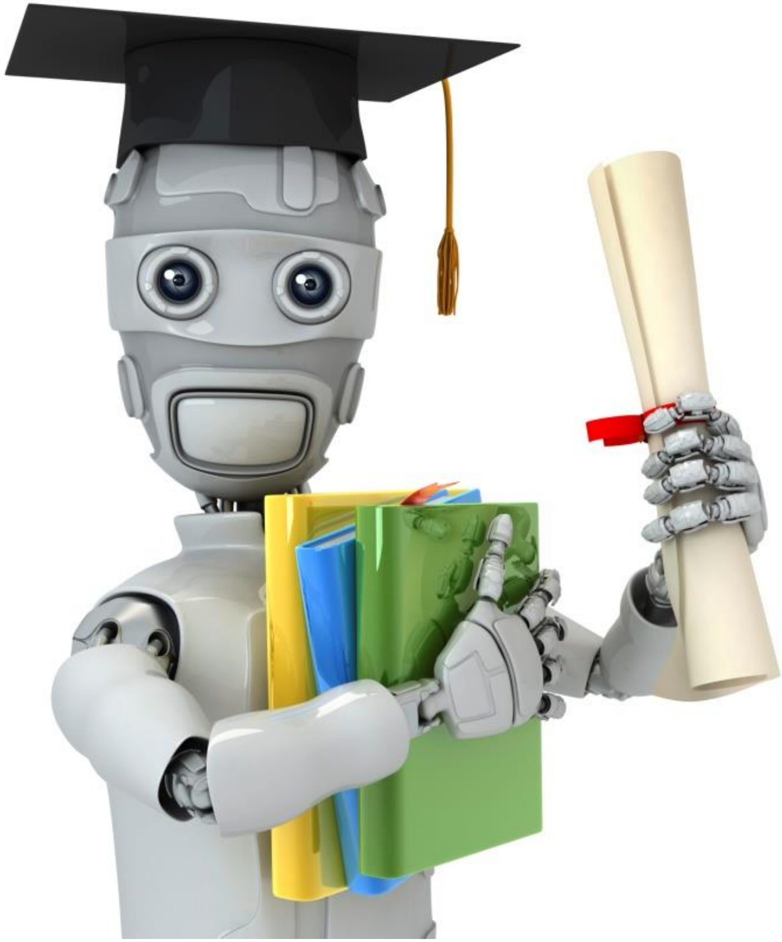
μ_k := average (mean) of points assigned to cluster k

}

minimize $J(\ldots)$ wrt $\boxed{\mu_1, \ldots, \mu_K}$

Clustering

Random initialization



Machine Learning

K-means algorithm

Randomly initialize K cluster centroids $\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^n$

Repeat {

for i = 1 to m

$c^{(i)}$:= index (from 1 to K) of cluster centroid closest to $x^{(i)}$

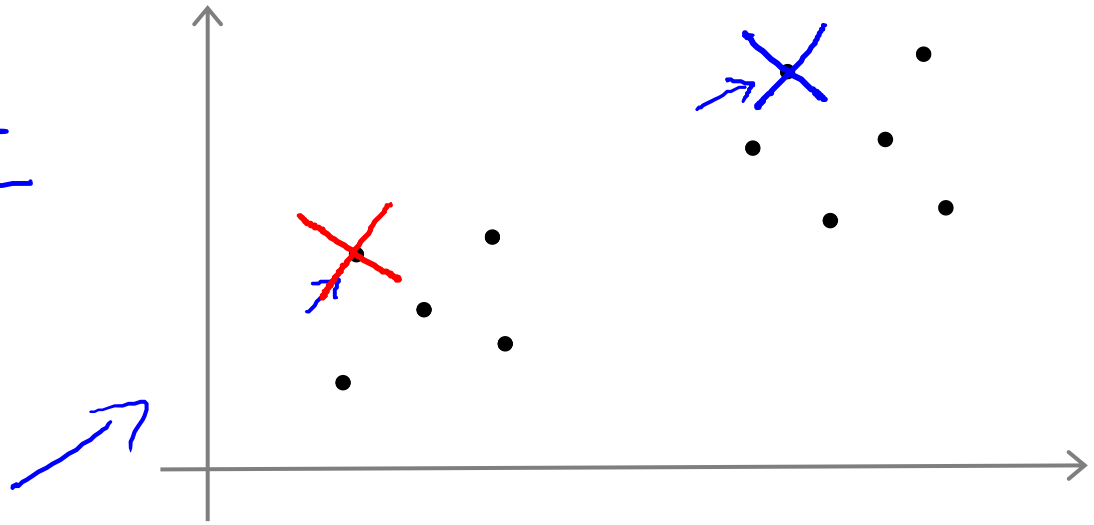
for k = 1 to K

μ_k := average (mean) of points assigned to cluster k

}

Random initialization

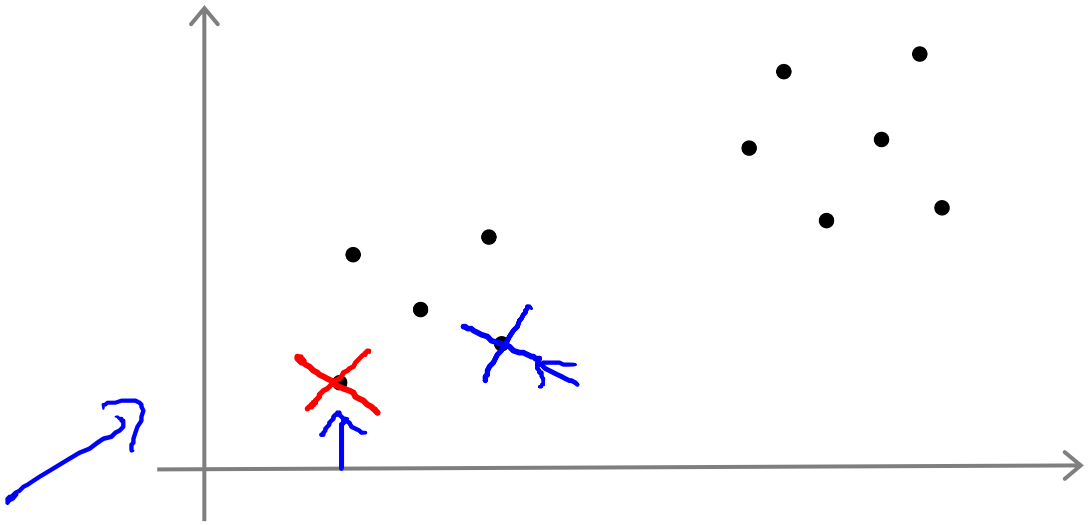
$K=2$



Should have $K < m$

Randomly pick K training examples.

Set $\mu_1, \ldots, \mu_K$ equal to these K examples.

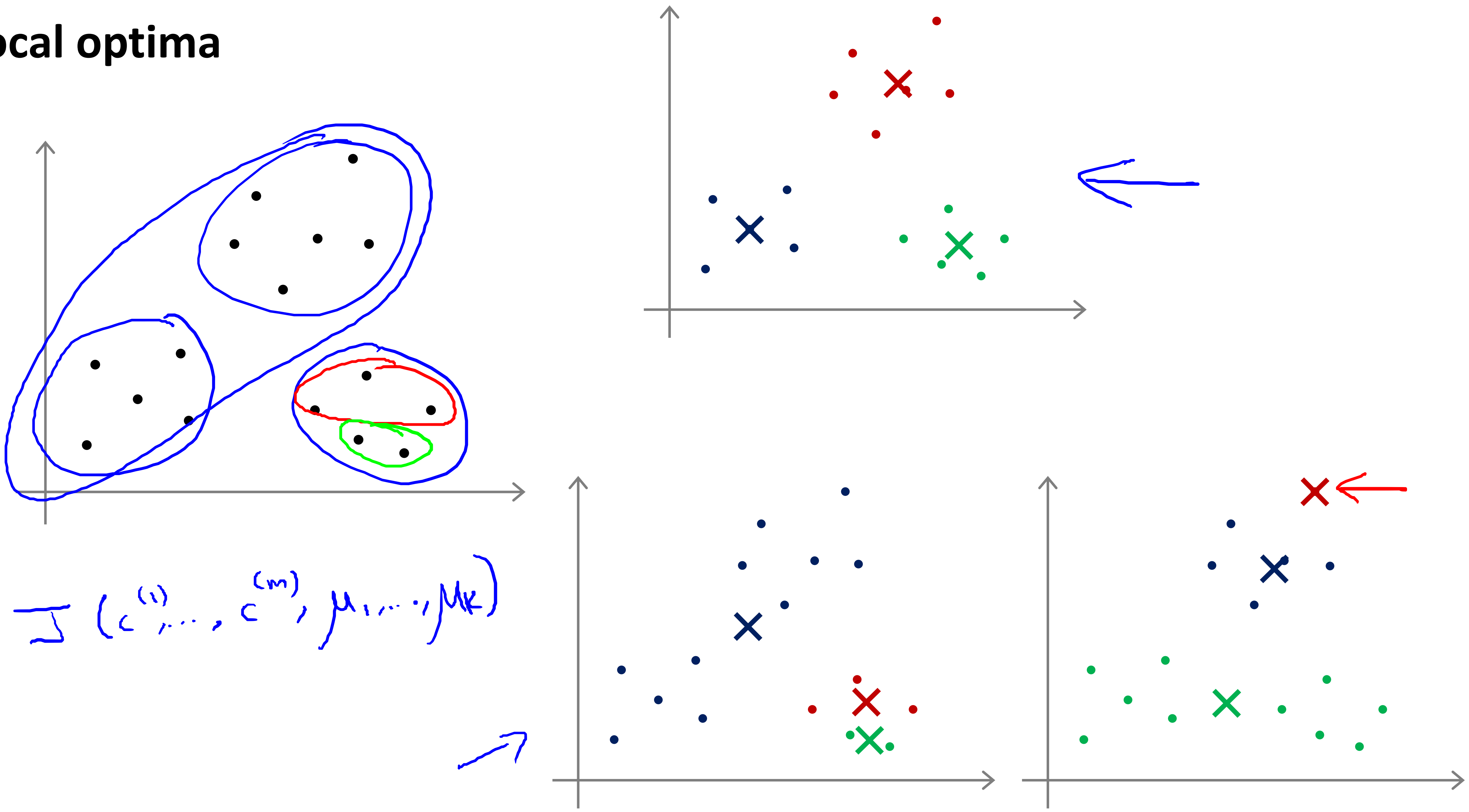


$\mu_1 = x^{(i)}$

$\mu_2 = x^{(j)}$

$\vdots$

Local optima



$$J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$$

Random initialization

For i = 1 to 100 {

Randomly initialize K-means.

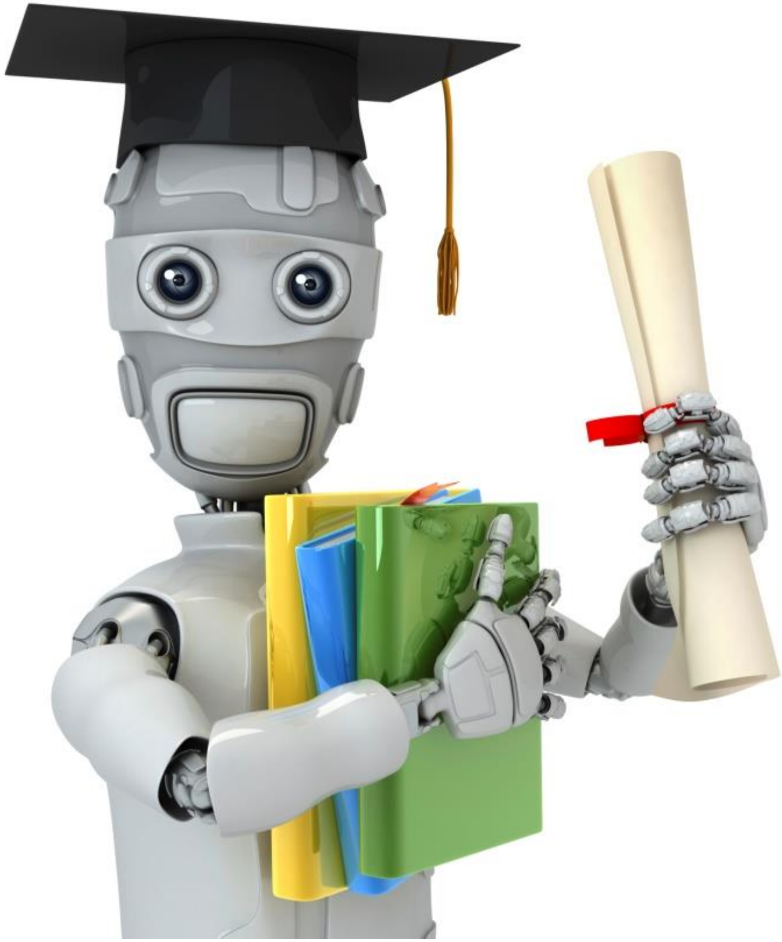
Run K-means. Get $c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K$

Compute cost function (distortion)

$$J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$$

}

Pick clustering that gave lowest cost $J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$

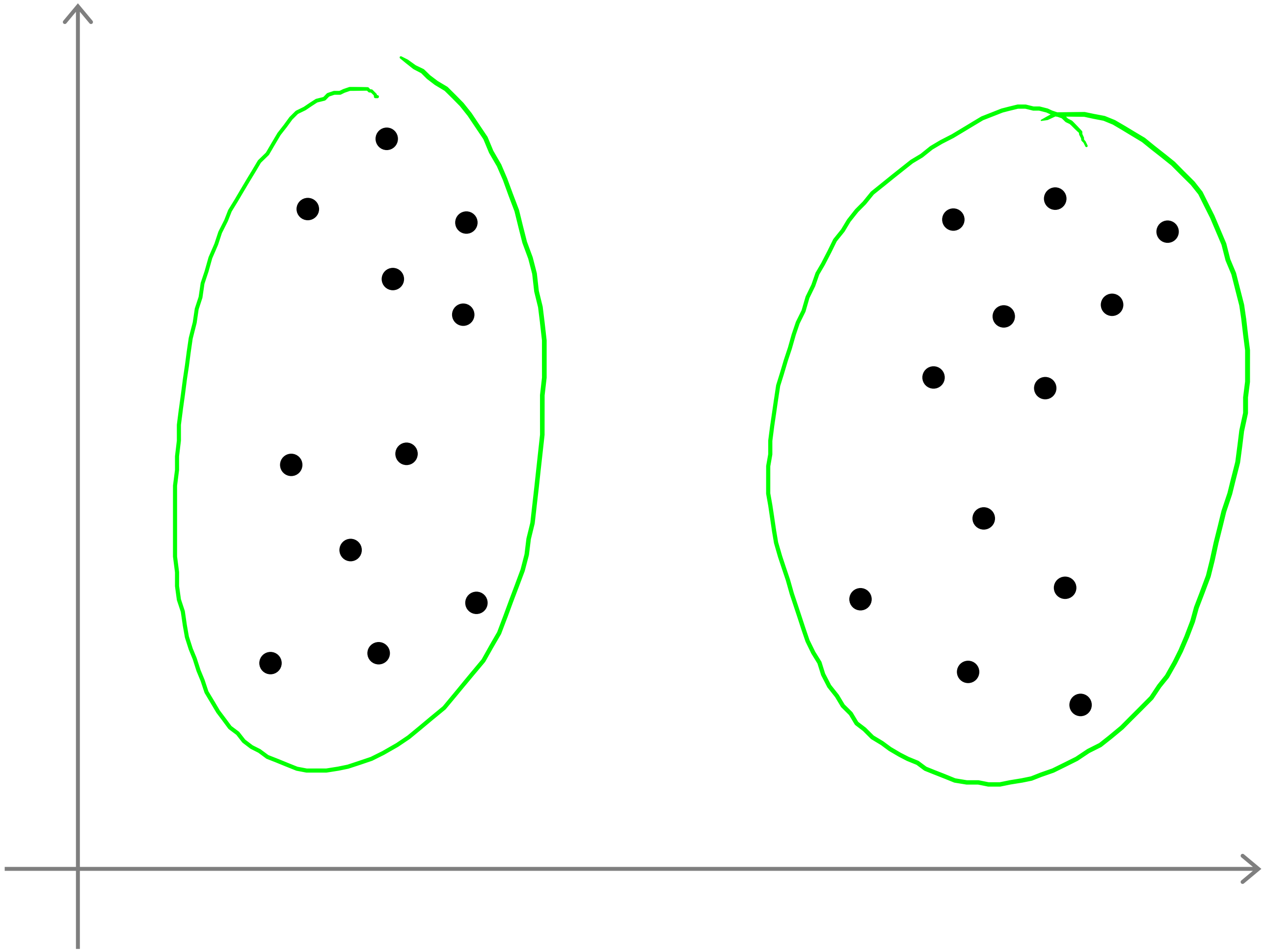


Machine Learning

Clustering

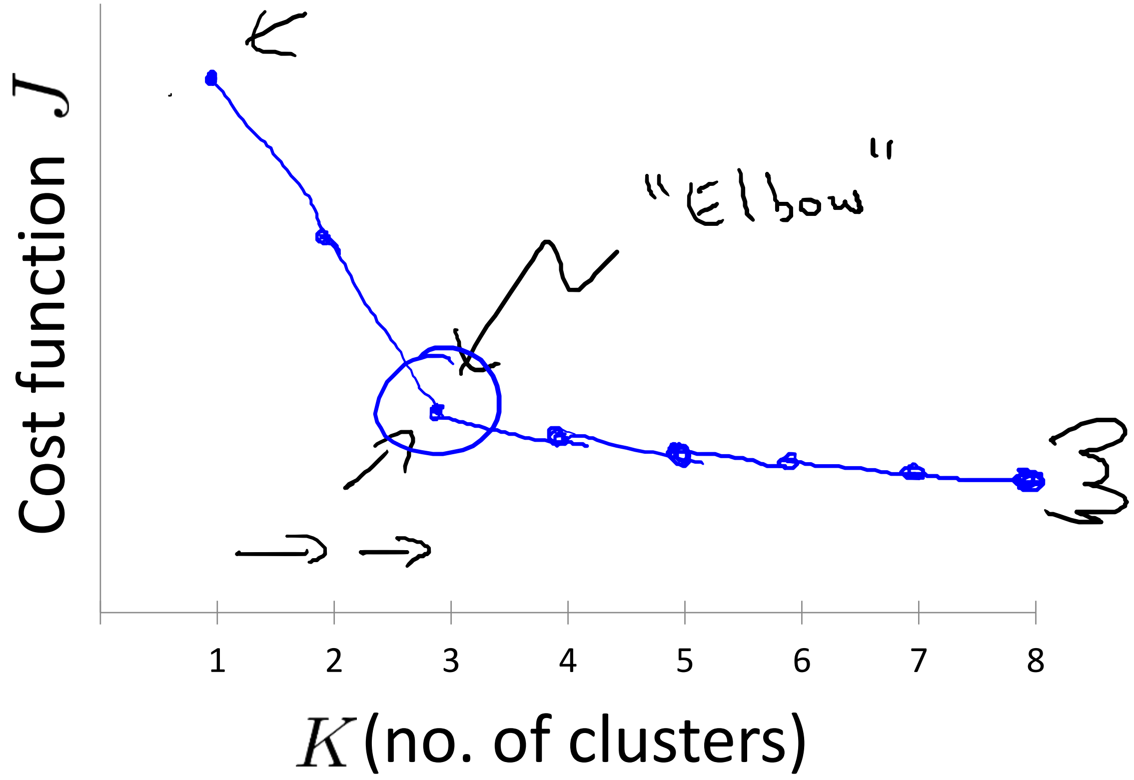
Choosing the number of clusters

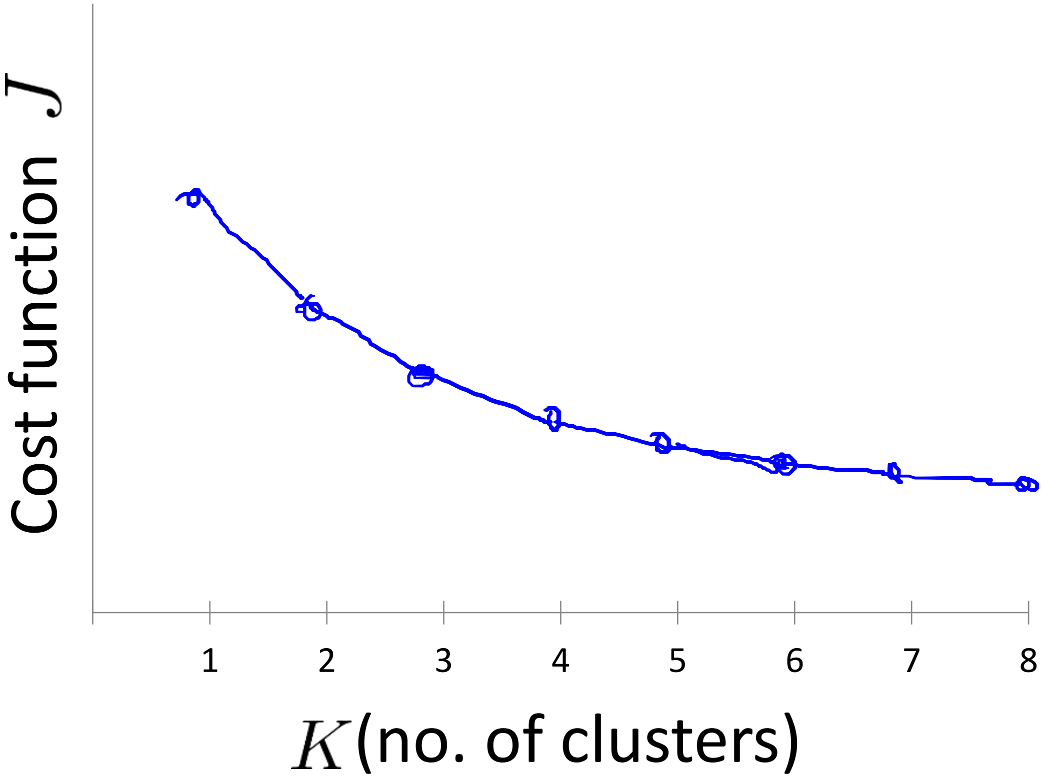
What is the right value of K?



Choosing the value of K

Elbow method:





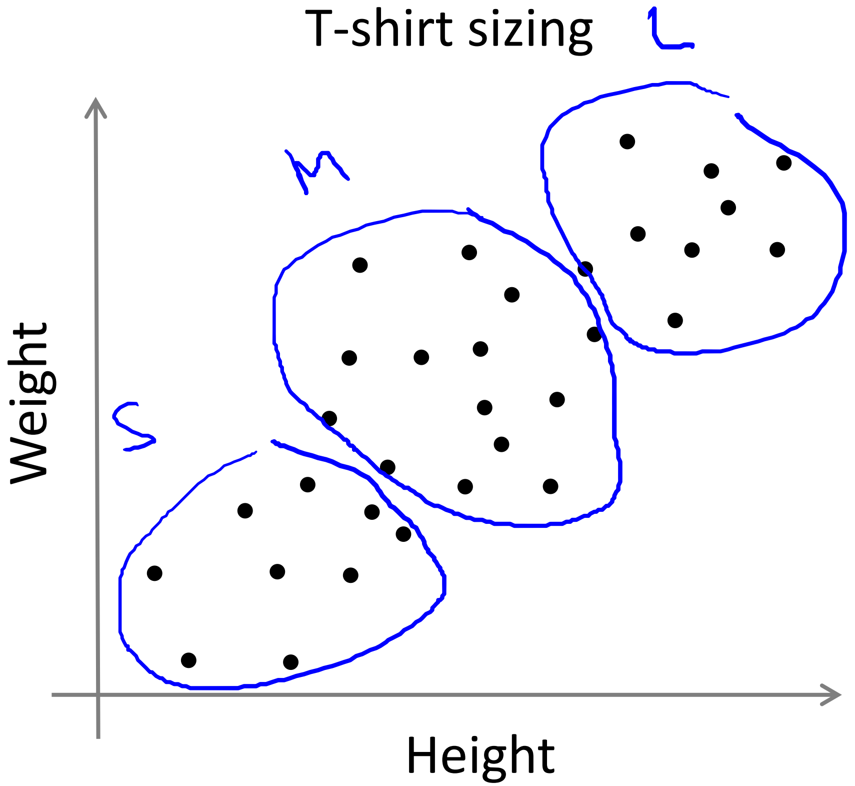
Choosing the value of K

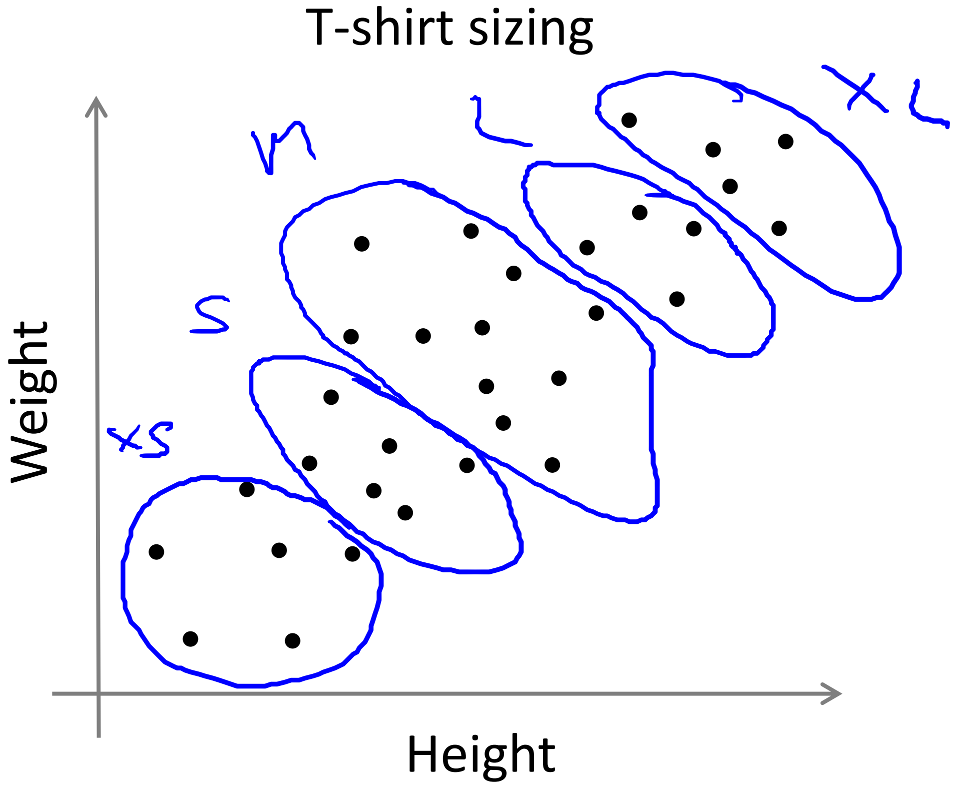
Sometimes, you're running K-means to get clusters to use for some later/downstream purpose. Evaluate K-means based on a metric for how well it performs for that later purpose.

K=3 S, M, L

K=5 XS, S, M, L, XL

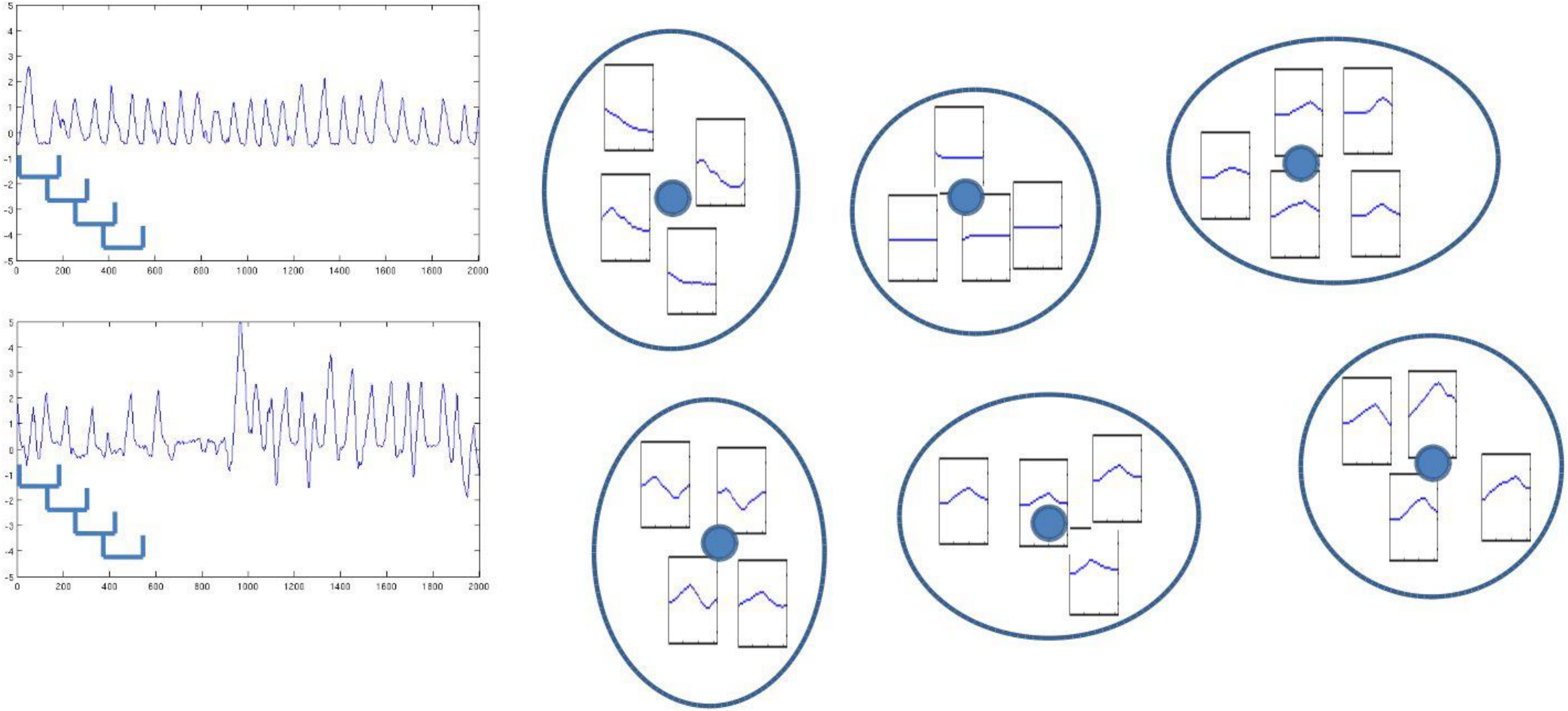
E.g.





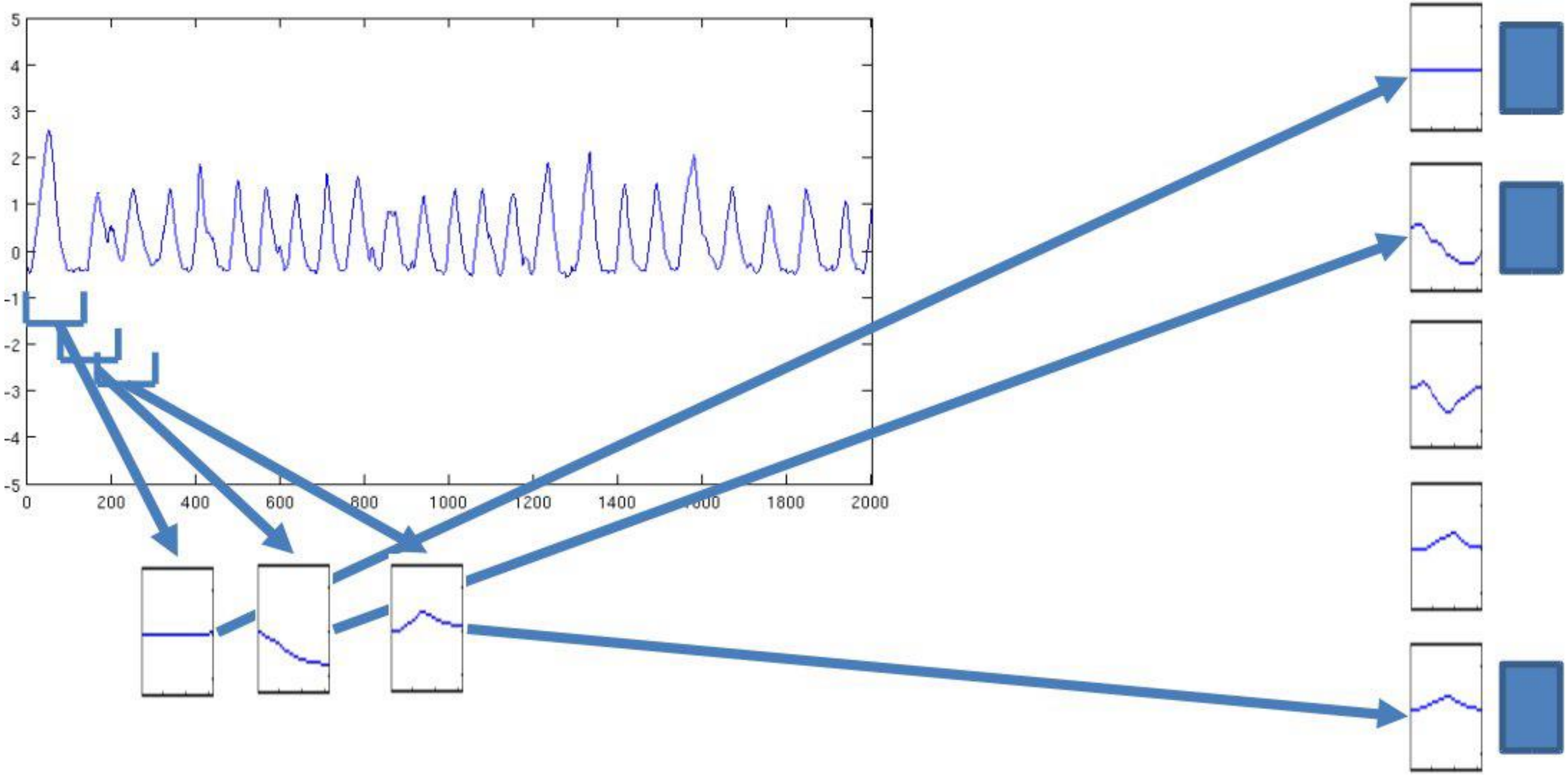
Codebook - Construction

- Clustering of Time Series subsequences



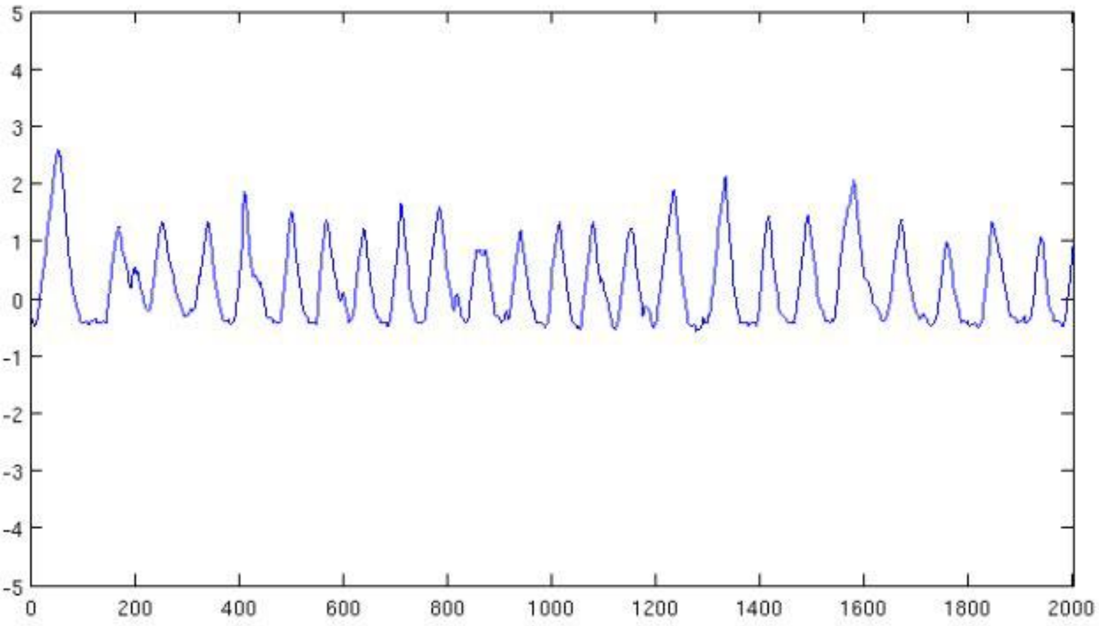
Codeword-Assignment

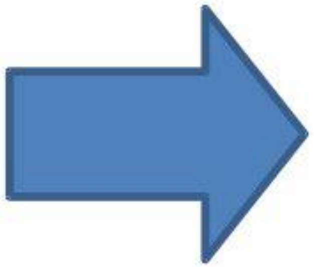
- Assignment of Codewords to the Constructed Clusters

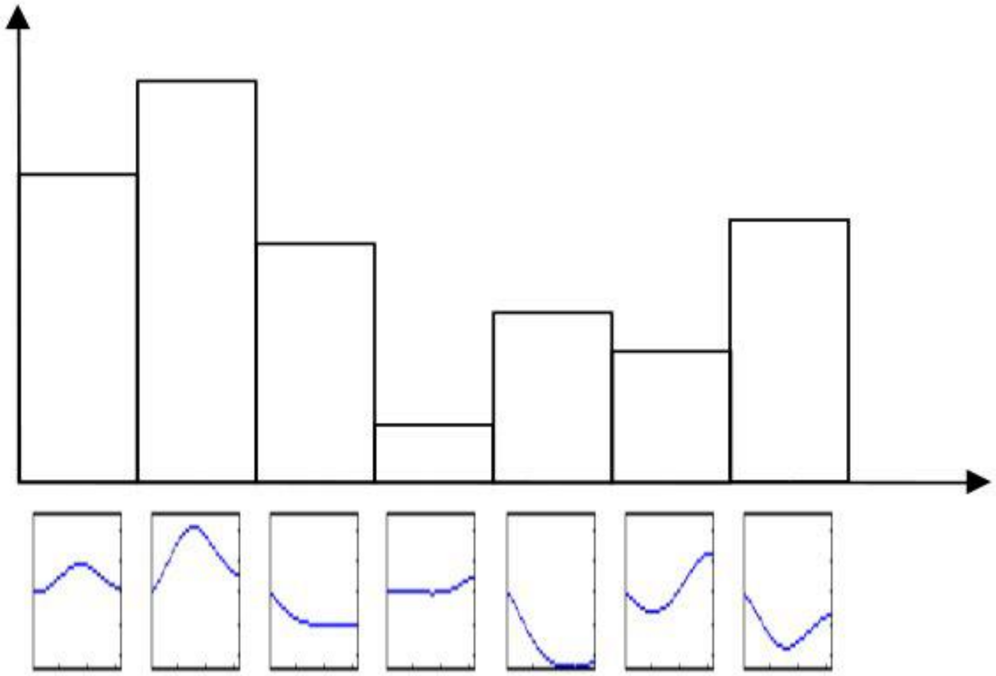


Histogram Generation

- Time Series Sequence Representation by a Histogram

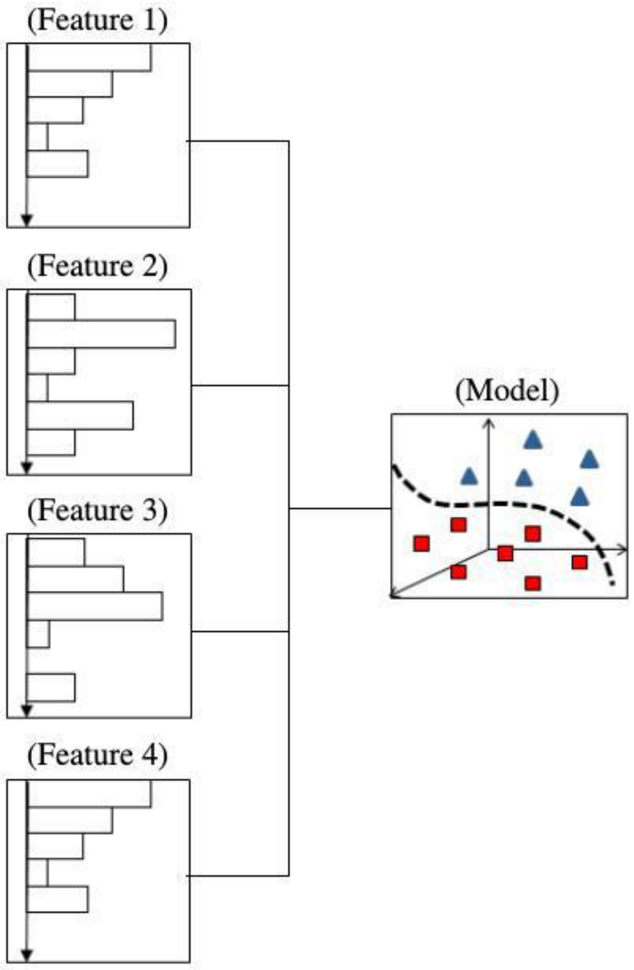




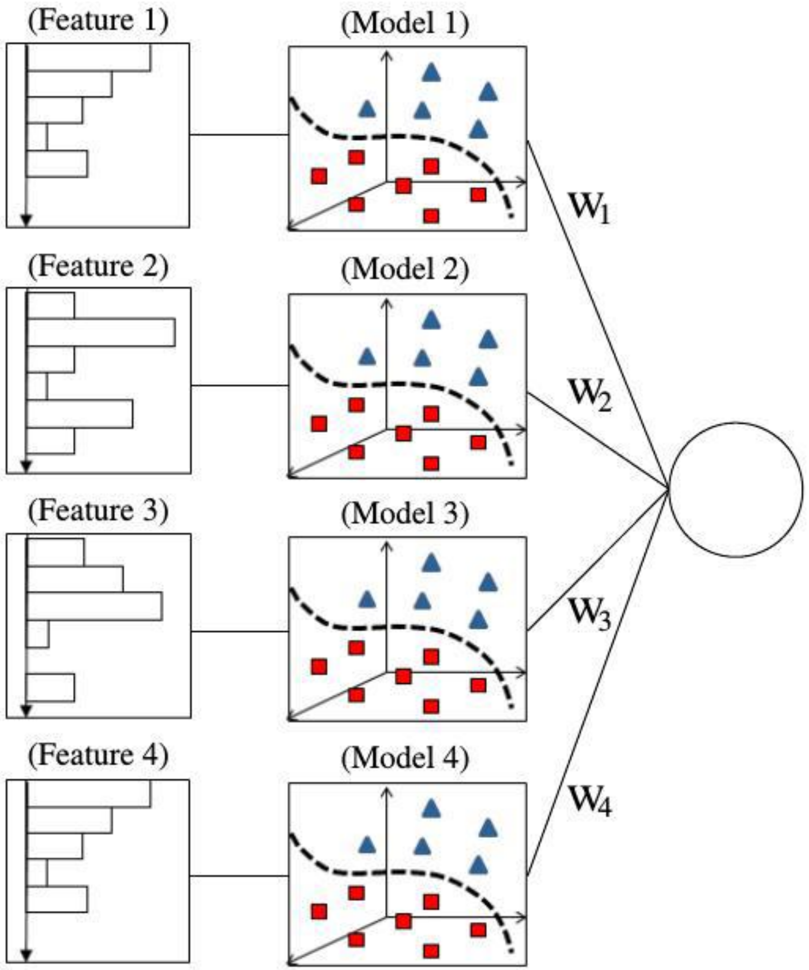


Fusion and Classification

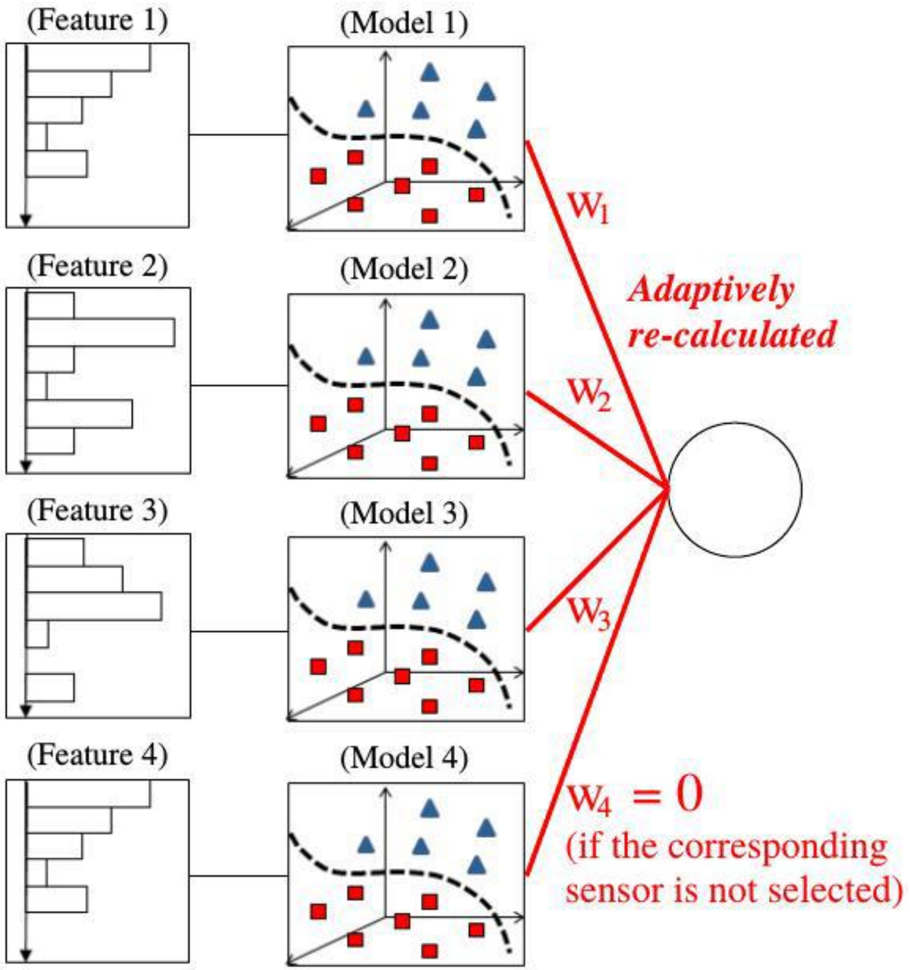
- Early Fusion



Late Fusion



Dynamic Late Fusion



Final Statements

- Feature learning delivers an optimum representation of the data, however, the semantic meaning of single feature dimensions gets often lost.

- Although feature learning algorithms seem to take over, manual feature engineering will remain important in the future, especially in the medical domain.

- Deep Neural Networks are a very powerful technique to automatically learn discriminative data representation in a supervised scenario.

Reading Homework

Article

Comparison of Feature Learning Methods for Human Activity Recognition Using Wearable Sensors