UNIVERSITY OF COPENHAGEN

Unsupervised Learning-1

Elements of Machine Learning, 2021

Raghavendra Selvan

raghav@di.ku.dk
Machine Learning Section
@raghavian

Video Instructions

- Embedded quizzes
- Pause and ponder
- Make notes of unclear concepts
- Post doubts/insights/queries on Absalon
- Teachers will engage

Pause-to-Ponder cue

Real or Fake?

(1)

(3)

(2)

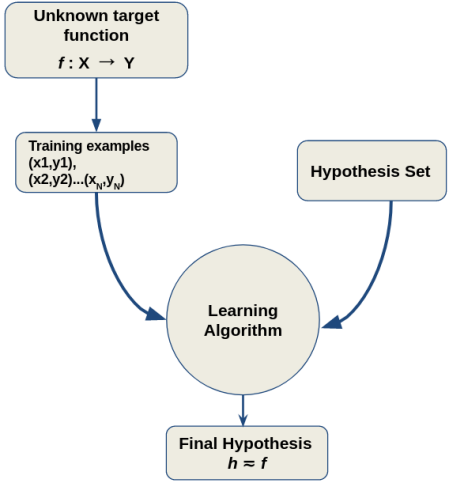
(4)

https://thispersondoesnotexist.com/

Karras, Tero, et al. "Analyzing and improving the image quality of stylegan." arXiv preprint arXiv:1912.04958 (2019).

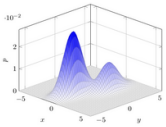
Learning from Data

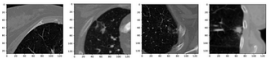
- Supervised learning
- Semi-supervised learning
- Reinforcement/Online Learning
- Self-supervised learning
- Unsupervised learning

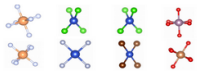


Based on Fig. 1.9, from Mostafa et al.

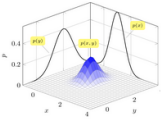
Formalizing Machine Learning



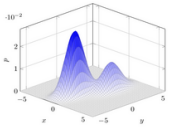




≃

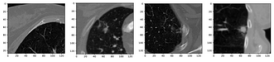


Formalizing Machine Learning



Unknown Data Distribution

P(X)



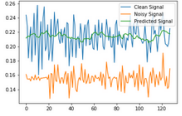
Observed data {X}

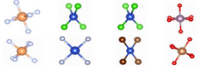
Supervised: {(x1,y1),(x2,y2)...(xN,yN)}
Unsupervised: {x1,x2,...xN}

≈

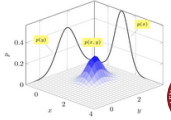
Modelled Data Distribution

$P_\theta(X)$





≈



Why Unsupervised Learning?

- Labelled data is expensive & difficult to obtain
- Abundant unlabelled data
- Generative modelling
- Density estimation **Current research trends**
- Representation learning[1]
- Contrastive Self-supervised Learning[2]

[1] Bengio, Yoshua, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives." IEEE transactions on pattern analysis and machine intelligence 35.8 (2013): 1798-1828.

[2] `https://github.com/jason718/awesome-self-supervised-learning`

Overview

Lecture-1

- Unsupervised learning
- PCA
- K-means clustering

Lecture-2

- Autoencoders
- Variational inference
- Variational Autoencoders

Literature

- **K-Means and PCA**
  Sections: 9.1, 9.2, 9.3.1, 9.3.2 & 12.1
  Christopher M Bishop, Pattern Recognition and Machine Learning
- **Autoencoders**
  Chapter 14
  Ian Goodfellow et al. Deep Learning
- **Variational Inference**
  Sections: 10.1 & 10.2
  Christopher M Bishop, Pattern Recognition and Machine Learning
- **Variational Autoencoders**
  Chapters: 1 & 2
  Kingma and Welling, An Introduction to Variational Autoencoders

Notations

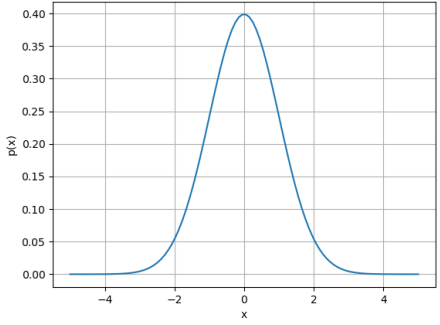
Data distribution	$P_X(\mathbf{x})$ or $p(\mathbf{x})$
Joint distribution	$P_{XY}(\mathbf{x}, \mathbf{y})$ or $p(\mathbf{x}, \mathbf{y})$
Observed data	$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} : \mathbf{x}_i \in \mathbb{R}^F$
Labels	$\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\} : \mathbf{y}_i \in \mathbb{R}^L$
Decision functions/ Models	$f_\theta(\cdot) : \mathbf{X} \rightarrow \mathbf{Y}$
Modeled distribution	$P_\theta(\mathbf{x})$ or $p_\theta(\mathbf{x})$

Principal Component Analysis (PCA)

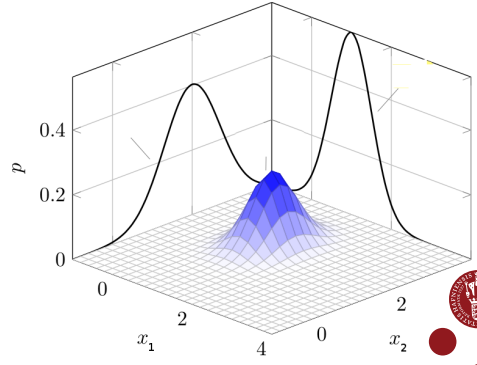
- Widely used *linear* dimensionality reduction method
- Powerful *feature extractor*
- *Lossy* compression method
- Useful tool for visualizations
- A first *Unsupervised* model

Curse of Dimensionality

- Consider a standard Normal distribution: $\mathcal{N}(x; 0, 1)$
- What is the value of PDF evaluated at $x = 0$?



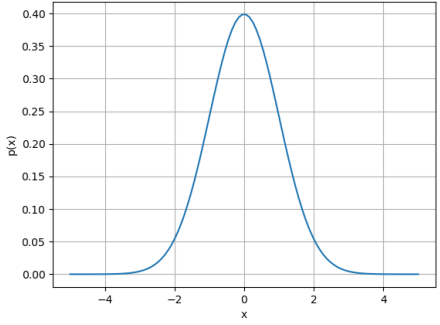
- Consider a standard Normal distribution in **2D**: $\mathcal{N}(\mathbf{x}; \mathbf{0}, \mathbf{I})$
- What is the value of PDF evaluated at $\mathbf{x} = \mathbf{0}$?



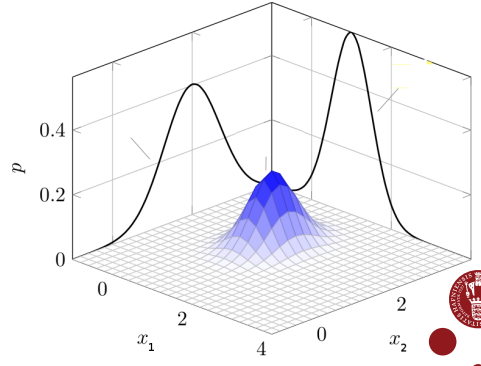
Pause-to-Ponder cue

Curse of Dimensionality

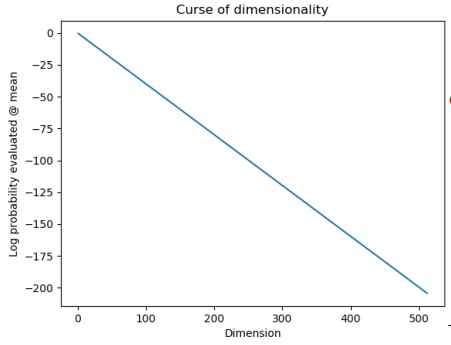
- Consider a standard Normal distribution: $\mathcal{N}(x; 0, 1)$
- What is the value of PDF evaluated at $x = 0$



- Consider a standard Normal distribution in **2D**: $\mathcal{N}(\mathbf{x}; \mathbf{0}, \mathbf{I})$
- What is the value of PDF evaluated at $\mathbf{x} = \mathbf{0}$



Curse of Dimensionality



- First introduced by Bellman[a]
- Points to the strange behaviours of high dimensional spaces.
- Indicates that number of samples needed to estimate an arbitrary function[b] grows exponentially with respect to the dimensionality of the function.

[a] Bellman R.E. Adaptive Control Processes. Princeton University 1961.

[b] with a given level of accuracy

PCA Objective

Consider a dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$: $\mathbf{x}_i \in \mathbb{R}^F$, we are interested in

- Obtained data representation in lower dimension, $\tilde{\mathbf{x}}_i \in \mathbb{R}^D : D < F$
- While maximizing the variance in the data

Solution

Obtain the orthogonal projection of the data onto a lower dimensional linear space constrained to maximize variance of the projected data [a]

[a] Hotelling, Harold. "Analysis of a complex of statistical variables into principal components." Journal of educational psychology 24.6 (1933): 417.

PCA: Maximum Variance Formulation

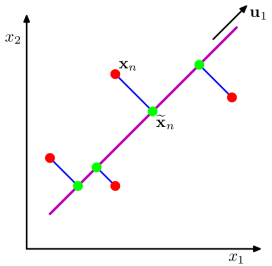


Fig. 12.2 Christopher Bishop, PRML

- Consider a dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \quad : \mathbf{x}_i \in \mathbb{R}^F$
- To begin with consider a one dimensional-space i.e, $D = 1$
- Direction of this space can be defined using an F=dimensional unit vector $\mathbf{u}_1$ i.e $\mathbf{u}_1^T \mathbf{u}_1 = 1$
- Each data point $\mathbf{x}_i$ is projected onto a scalar $\tilde{x}_i = \mathbf{u}_1^T \mathbf{x}_i$

PCA: Maximum Variance Formulation

- If each data point $\mathbf{x}_i$ is projected onto a scalar $\tilde{x}_i = \mathbf{u}_1^T \mathbf{x}_i$
- Mean of the projected data is $\mathbf{u}_1^T \bar{\mathbf{x}}$ where $\bar{\mathbf{x}}$ is the sample mean

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i$$

- And variance of the projected data is given by

$$\frac{1}{N} \sum_{i=1}^{N} \left(\tilde{x}_i - \bar{\tilde{x}}_i \right)^2 = \frac{1}{N} \sum_{i=1}^{N} \left(\mathbf{u}_1^T \mathbf{x}_i - \mathbf{u}_1^T \bar{\mathbf{x}} \right)^2 = \mathbf{u}_1^T \mathbf{S} \mathbf{u}_1, \textit{where}$$

$$\mathbf{S} = \frac{1}{N} \sum_{i=1}^{N} \left(\mathbf{x}_i - \bar{\mathbf{x}} \right) \left(\mathbf{x}_i - \bar{\mathbf{x}} \right)^T \text{ is the data covariance matrix.}$$

PCA: Maximum Variance Formulation

Remember that we not only want to reduce the dimensionality but want to maximize the projected variance $\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1$ How can we maximize this?

- Trivial solution would be to allow $||\mathbf{u}_i|| \rightarrow \infty$
- This is the reason we choose $\mathbf{u}_1$ to be unit vector i.e $\mathbf{u}_1^T \mathbf{u}_1 = 1$
- We now have a constrained optimization problem

$$\max_{\mathbf{u}_1} \quad \mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 \quad \text{such that } (1 - \mathbf{u}_1^T \mathbf{u}) = 0$$

  which can be relaxed into an unconstrained optimization prob.

- Using Lagrangian multiplier λ_1 the optimization becomes:

$$\max_{\mathbf{u}_1} \quad \mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 + \lambda_1 (1 - \mathbf{u}_1^T \mathbf{u}_1)$$

PCA: Maximum Variance Formulation

- Setting the derivative wrt $\mathbf{u}_1$ equal to zero,

$$\mathbf{S}\mathbf{u}_1 = \lambda_1 \mathbf{u}_1$$

- This relation says that $\mathbf{u}_1$ must be an eigenvector of $\mathbf{S}$.
- Note that $\mathbf{u}_1^T \mathbf{S}\mathbf{u}_1 = \lambda_1$
- Variance will be maximum when we choose the eigenvector with the largest eigenvalue λ_1 as the principal component

PCA in Practice [3]

Algorithm (when $N > F$)

Input: Data $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ $\quad : \mathbf{x}_i \in \mathbb{R}^F$

Number of dimensions of the projected data D

1. Compute sample mean $\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i$
2. Compute sample data covariance $\mathbf{S} = \frac{1}{N} \sum_{i=1}^{N} \left(\mathbf{x}_i - \bar{\mathbf{x}}\right)\left(\mathbf{x}_i - \bar{\mathbf{x}}\right)^T$
3. Perform eigenvalue decomposition as $\mathbf{S} = \mathbf{V}\Lambda\mathbf{V}^T$
4. Collect the first D eigenvectors of $\mathbf{S}$ from $\mathbf{V}$ sorted by decreasing eigenvalue into $\mathbf{U}$
5. Compute $\tilde{x}_i = \mathbf{U}^T\mathbf{x}_i$ for $i = 1, \ldots, N$

Output: Principal components $\mathbf{U}$, projected data $\{\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_N\}$, eigenvalues of principal components

where $\mathbf{V} \in \mathbb{R}^{N \times N}$: Matrix with eigen vectors , $\Lambda \in \mathbb{R}^{N \times N}$: Diagonal matrix with eigenvalues

[3] Adapted from C.Igel

Example of PCA based dimensionality reduction

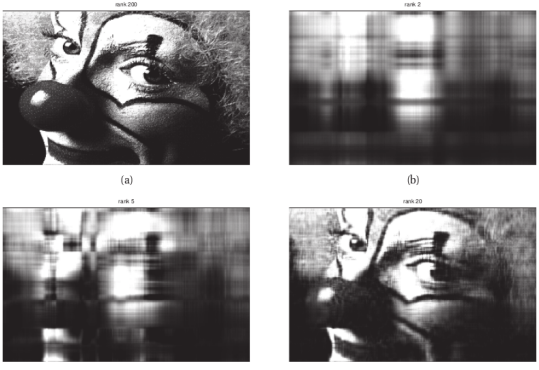


Figure 12.9 Low rank approximations to an image. Top left: The original image is of size 200×320, so has rank 200. Subsequent images have ranks 2, 5, and 20.

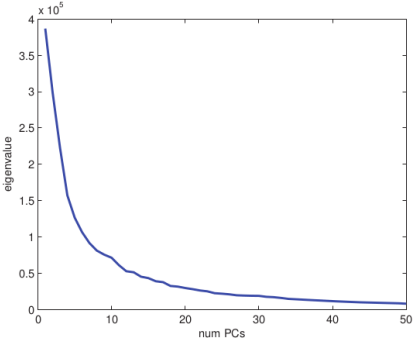
Figure 12.9 from Kevin Murphy, Probabilistic Machine Learning

Quantifying Dimensionality Reduction

- Is PCA lossy?
- How to quantify loss of information?
- When can all variance be retained?

Pause-to-Ponder cue

Fraction of Explained variance



Eigenvalue spectrum. (Fig. 12. 16 from Kevin Murphy, PML)

- Fraction of explained variance: $F_e = \frac{\sum_{i=1}^{D} \lambda_i}{\sum_{i=1}^{N} \lambda_i}$
  Denominator is due to $\text{Tr}(\mathbf{S}) = \text{Tr}(\mathbf{V}\Lambda\mathbf{V}^T) = \sum_{i=1}^{N} \lambda_i$
- $F_e = 1$ when $D = N$

Summary: PCA

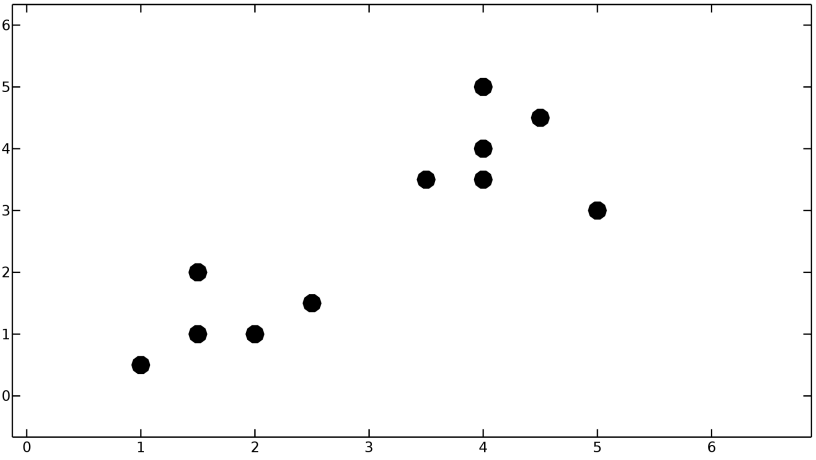
-+ Curse of dimensionality
+ Dimensionality reduction while maximizing variance
+ Data is projected into *linear*, orthogonal subspace
+ Quantifiable loss of information with "explained variance"
+ Singular Value Decomposition for cheaper computation
- Lossy
- For some datasets $D \approx F$

Unsupervised Learning: Clustering

We focus on k-Means clustering

- Process of grouping similar objects together
- Detecting patterns
- Either based on similarity or features
- Representing data at higher abstractions
- Applications like image segmentation

Toy example:

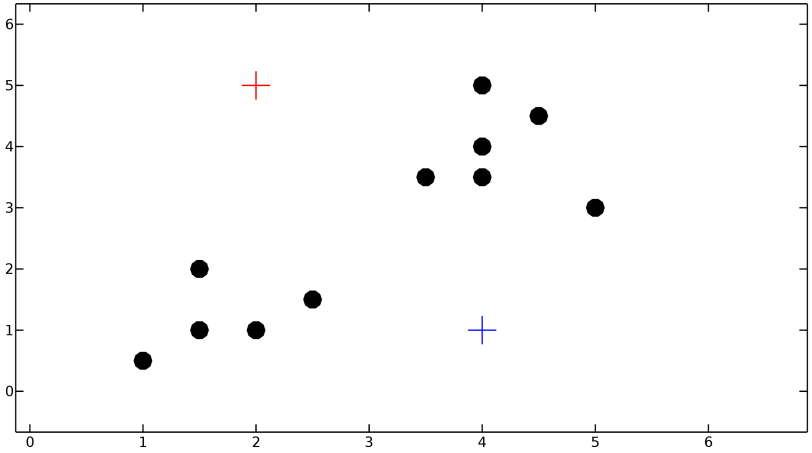


How would you cluster these points?

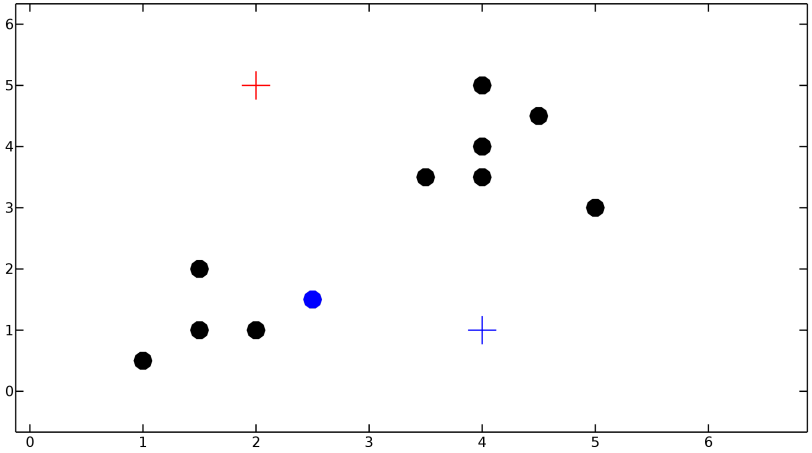
Think on these lines...

- Measure of similarity
- Number of clusters
- Complexity

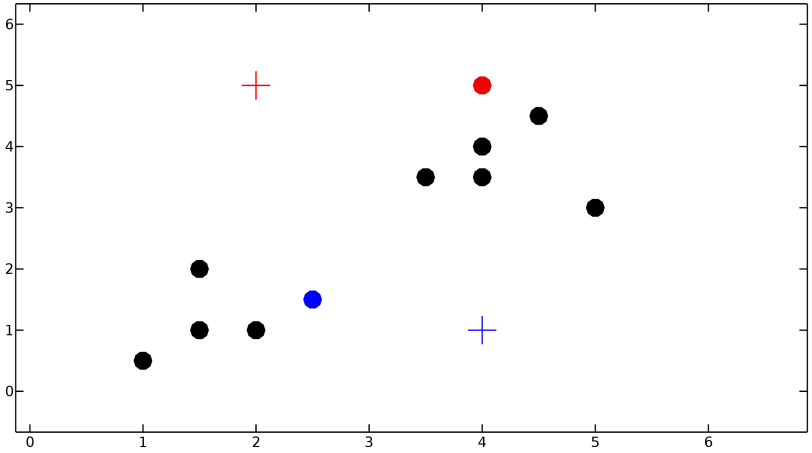
Initialize centroids, randomly!



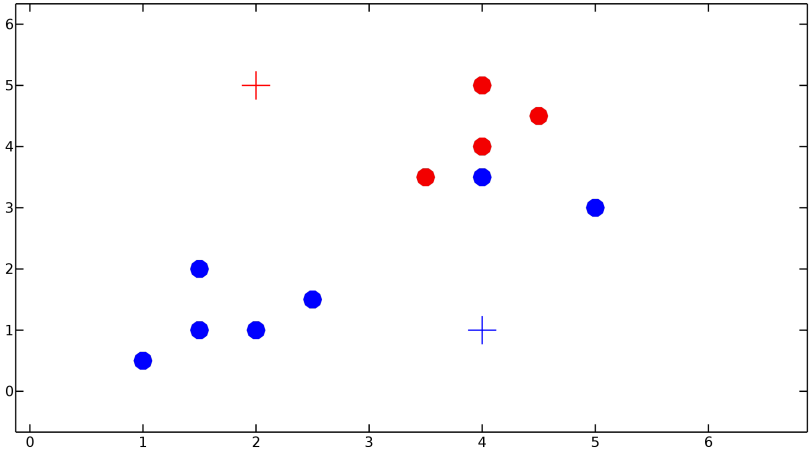
Assign points to nearest centroid!



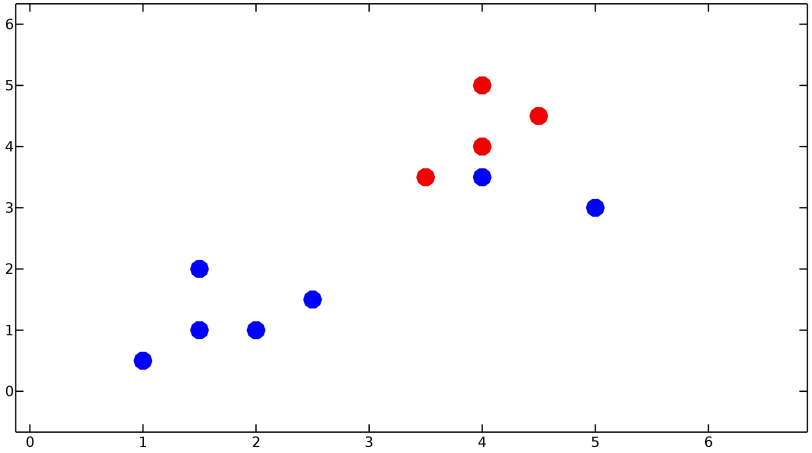
Assign points to nearest centroid!



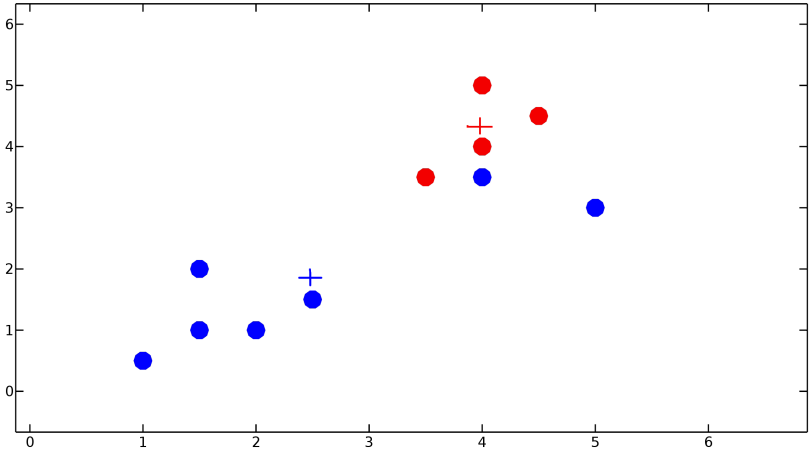
Assign points to nearest centroid!



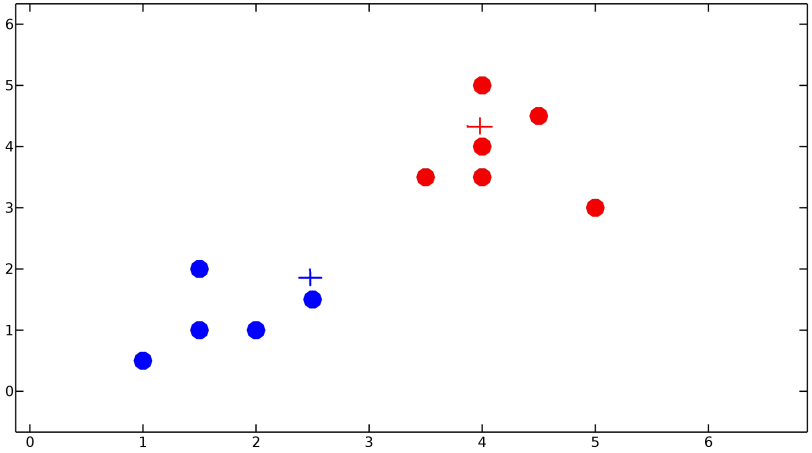
Recompute centroids!



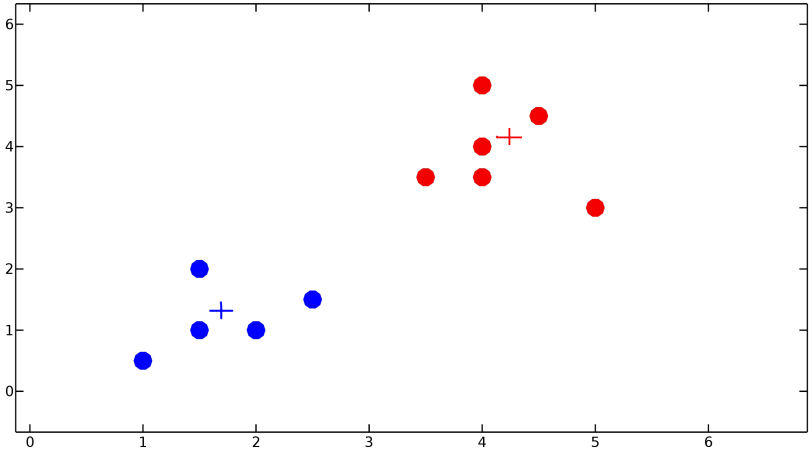
Recompute centroids!



Iterate, until convergence!



Iterate, until convergence!



Formalizing k-Means Clustering[4]

Given a dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ $: \mathbf{x}_i \in \mathbb{R}^F$

Objective is:

$$\min_{\substack{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k \\ \mathbf{X}_1, \ldots, \mathbf{X}_k : \mathbf{X} = \\ \mathbf{X}_1 \cup \cdots \cup \mathbf{X}_k}} \sum_{i=1}^{k} \sum_{\boldsymbol{x} \in \mathbf{X}_i} \|\boldsymbol{x} - \boldsymbol{\mu}_i\|^2$$

Iterate:

Data assignment: Assign each data point to cluster represented by the most similar prototype. This leads to a new partitioning of the data.

Centroid relocation: Recompute cluster centroids as mean of data points assigned to respective cluster.

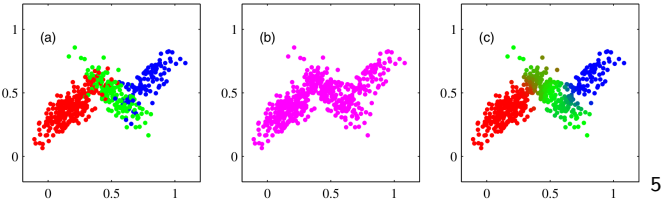
[4] Adapted from from C.Igel

k-Means clustering based Image Segmentation

Figure 9.3 Two examples of the application of the K-means clustering algorithm to image segmentation showing the initial images together with their K-means segmentations obtained using various values of K. This also illustrates of the use of vector quantization for data compression, in which smaller values of K give higher compression at the expense of poorer image quality.

Figure from Christopher Bishop, PRML

Summary: k-Means Clustering



5

+ Simple with good performance
+ Single hyperparameter
+ Cross validation for parameter selection
+ Flexibile similarity measures
+ **Hard EM**: Assigns hard labels
+ Powerful unsupervised method when used with PCA
- k has to be pre-selected; Sensitive to initialization

[5] Fig. 9.5 from Christopher Bishop, PRML