Unsupervised Learning

MLS 2025, Data Science Lab, UCPH

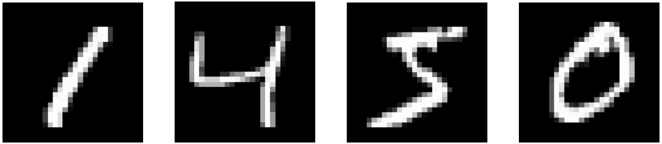
Tong Chen (toch@di.ku.dk)

Postdoc
Dept. of Computer Science (ML Section)
University of Copenhagen

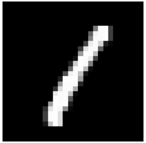
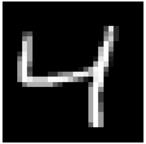
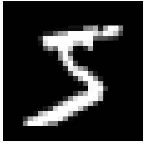
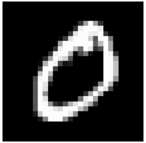
With Raghav (raghav@di.ku.dk)

November 27, 2025

Supervised v.s. Unsupervised Learning

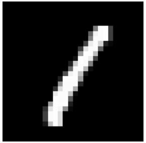
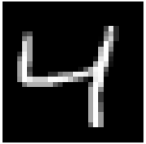
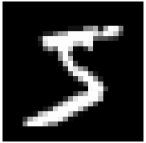
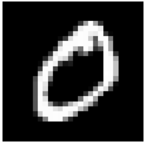


Supervised v.s. Unsupervised Learning

Classification: label "1" label "4" label "5" label "0"

Supervised v.s. Unsupervised Learning

				
Classification:	label “1”	label “4”	label “5”	label “0”
Clustering:	cluster 1	cluster 2	cluster 3	cluster 4

Generative Modeling: Real or Fake?

https://thispersondoesnotexist.com/

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

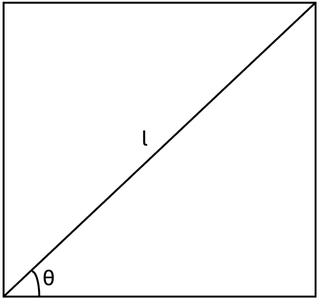
Overview: Unsupervised Learning

- Curse of Dimensionality
- Principal Component Analysis (PCA)
- K-means clustering

Curse of Dimensionality

Consider the diagonal of a unit square.

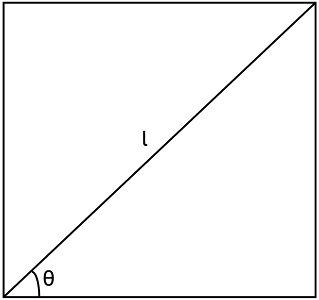
- Length of the diagonal l?
- Value of $\sin(\theta)$?
- Area of the enclosed circle?



Curse of Dimensionality

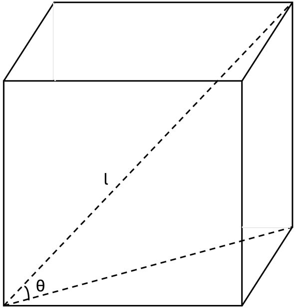
Consider the diagonal of a unit square.

- Length of the diagonal l?
- Value of $\sin(\theta)$?
- Area of the enclosed circle?



Consider the diagonal of a unit cube.

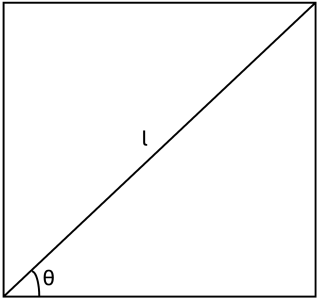
- Length of the diagonal l?
- Value of $\sin(\theta)$?
- Volume of the enclosed ball?



Curse of Dimensionality

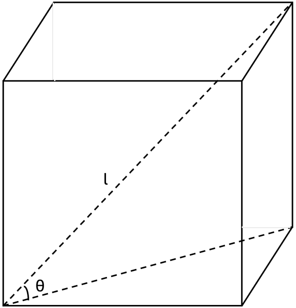
Consider the diagonal of a unit square.

- Length of the diagonal l?
- Value of $\sin(\theta)$?
- Area of the enclosed circle?



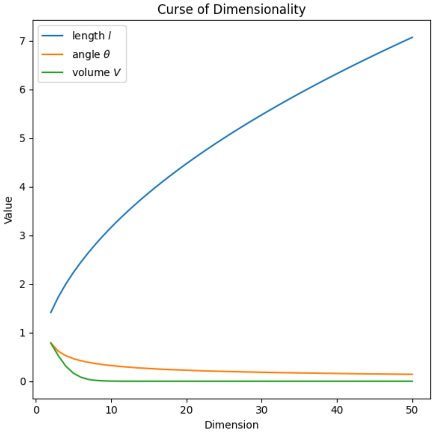
Consider the diagonal of a unit cube.

- Length of the diagonal l?
- Value of $\sin(\theta)$?
- Volume of the enclosed ball?



Discuss: How about a unit hyper-box in dimension n?

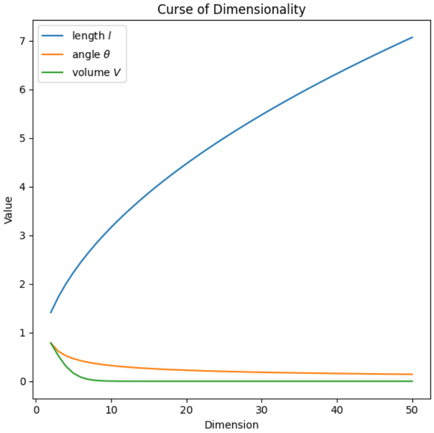
Curse of Dimensionality [1]



Strange behaviors in high dimensional spaces:

[1] First introduced by Bellman R.E.: Adaptive Control Processes. Princeton University Press, Princeton, NJ, 1961.

Curse of Dimensionality [1]



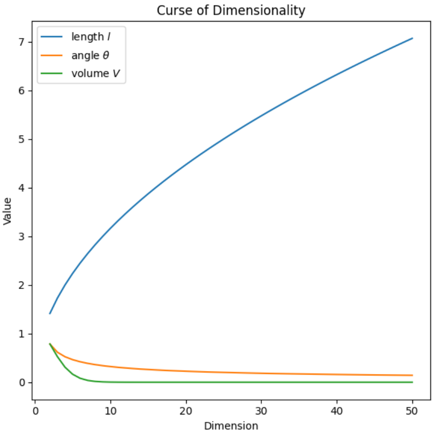
Strange behaviors in high dimensional spaces:

- Length of the diagonal

$$l_n = \sqrt{n} \to \infty;$$

[1] First introduced by Bellman R.E.: Adaptive Control Processes. Princeton University Press, Princeton, NJ, 1961.

Curse of Dimensionality [1]



Strange behaviors in high dimensional spaces:

- Length of the diagonal

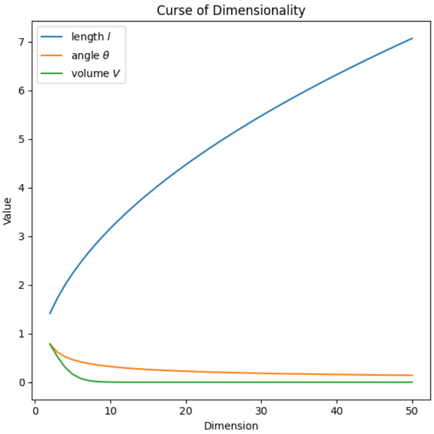
$$l_n = \sqrt{n} \to \infty;$$

- Value of the angle

$$\theta_n = \arcsin(1/\sqrt{n}) \to 0;$$

[1] First introduced by Bellman R.E.: Adaptive Control Processes. Princeton University Press, Princeton, NJ, 1961.

Curse of Dimensionality [1]



Strange behaviors in high dimensional spaces:

- Length of the diagonal

$$l_n = \sqrt{n} \to \infty;$$

- Value of the angle

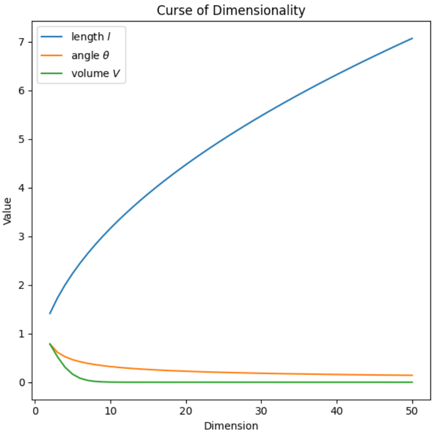
$$\theta_n = \arcsin(1/\sqrt{n}) \to 0;$$

- Volume of the enclosed ball

$$V_n = \frac{\pi^{n/2}}{\Gamma(n/2+1)}\left(\frac{1}{2}\right)^n \to 0.$$

[1] First introduced by Bellman R.E.: Adaptive Control Processes. Princeton University Press, Princeton, NJ, 1961.

Curse of Dimensionality [1]



Strange behaviors in high dimensional spaces:

- Length of the diagonal

$$l_n = \sqrt{n} \to \infty;$$

- Value of the angle

$$\theta_n = \arcsin(1/\sqrt{n}) \to 0;$$

- Volume of the enclosed ball

$$V_n = \frac{\pi^{n/2}}{\Gamma(n/2+1)} \left(\frac{1}{2}\right)^n \to 0.$$

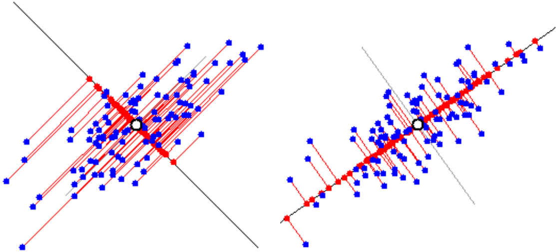
- Sampling complexity ...

[1] First introduced by Bellman R.E.: Adaptive Control Processes. Princeton University Press, Princeton, NJ, 1961.

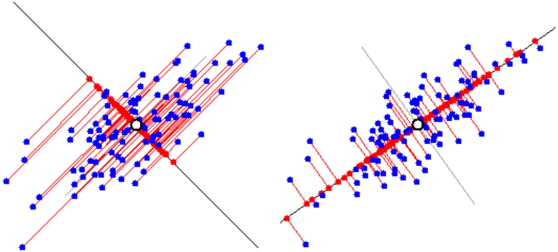
Principal Component Analysis (PCA)

- *Linear* dimensionality reduction method;
- Powerful *feature extractor*;
- *Lossy* compression method;
- Widely used for data compression, visualization, and noise reduction.

Intuition: Orthogonal Projection

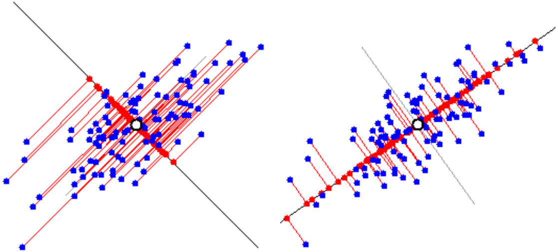


Intuition: Orthogonal Projection



Which projection is “good”? Why?

Intuition: Orthogonal Projection



Which projection is "good"? Why?

- More variance in projections;
- Less distance to the line.

PCA Objective: Find a “Good” Projection

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

PCA Objective: Find a “Good” Projection

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

- Project data from high-dimensional space $\mathbb{R}^n$ to low-dimensional space $\mathbb{R}^m$: $m < n$;

PCA Objective: Find a "Good" Projection

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

- Project data from high-dimensional space $\mathbb{R}^n$ to low-dimensional space $\mathbb{R}^m$: $m < n$;
- Preserve as much information from the original data as possible.

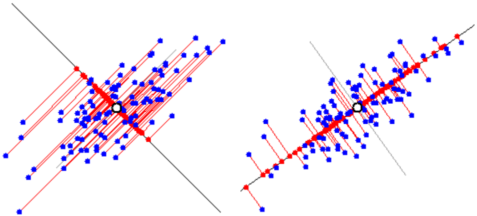
PCA Objective: Find a "Good" Projection

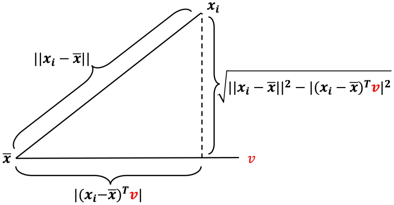
Consider a dataset $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \subseteq \mathbb{R}^n$, we are interested in:

- Project data from high-dimensional space $\mathbb{R}^n$ to low-dimensional space $\mathbb{R}^m$: $m < n$;
- Preserve as much information from the original data as possible.

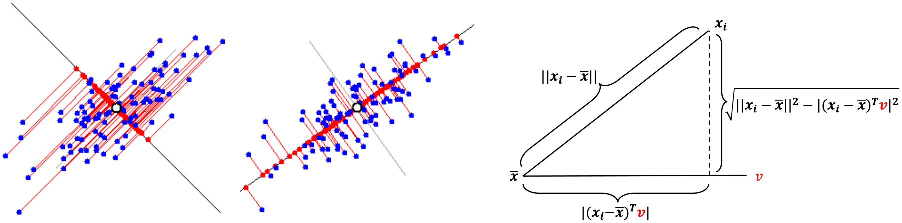
Discuss: How do we measure the amount of information we preserve/loose?

A "Good" Projection $= \cdots$





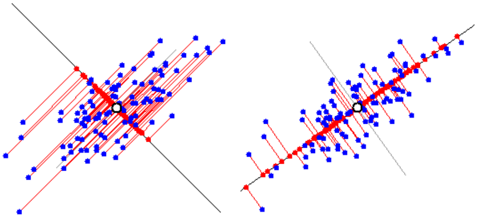
A "Good" Projection $= \cdots$

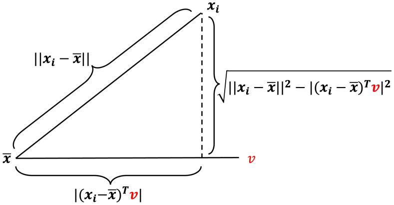


- Minimize the average distance to projections:

$$\min_{\mathbf{v}} \frac{1}{N} \sum_{i=1}^{N} \left(\|\mathbf{x}_i - \bar{\mathbf{x}}\|^2 - |(\mathbf{x}_i - \bar{\mathbf{x}})^T \mathbf{v}|^2 \right) = \min_{\mathbf{v}} \operatorname{Var}(\mathcal{X}) - \operatorname{Var}(\mathcal{X}\mathbf{v});$$

A "Good" Projection $= \cdots$





- Minimize the average distance to projections:

$$\min_{\mathbf{v}} \frac{1}{N}\sum_{i=1}^{N}\left(\|\mathbf{x}_i - \bar{\mathbf{x}}\|^2 - |(\mathbf{x}_i - \bar{\mathbf{x}})^T\mathbf{v}|^2\right) = \min_{\mathbf{v}} \mathrm{Var}(\mathcal{X}) - \mathrm{Var}(\mathcal{X}\mathbf{v});$$

- Maximize the average variance of projections:

$$\max_{\mathbf{v}} \frac{1}{N}\sum_{i=1}^{N}|(\mathbf{x}_i - \bar{\mathbf{x}})^T\mathbf{v}|^2 = \max_{\mathbf{v}} \mathrm{Var}(\mathcal{X}\mathbf{v}).$$

PCA: Maximum Variance Formulation

- Covariance matrix $\mathbf{S} := \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}}) \cdot (\mathbf{x}_i - \bar{\mathbf{x}})^T$,

PCA: Maximum Variance Formulation

- Covariance matrix $\mathbf{S} := \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}}) \cdot (\mathbf{x}_i - \bar{\mathbf{x}})^T$,

$$
\begin{aligned}
\max_{\mathbf{v}} \mathrm{Var}(\mathcal{X}\mathbf{v}) \quad &= \max_{\mathbf{v}} \frac{1}{N}\sum_{i=1}^{N} |(\mathbf{x}_i - \bar{\mathbf{x}})^T \mathbf{v}|^2 \\
&= \max_{\mathbf{v}} \frac{1}{N}\sum_{i=1}^{N} (\mathbf{x}_i - \bar{\mathbf{x}})^T \mathbf{v} \cdot (\mathbf{x}_i - \bar{\mathbf{x}})^T \mathbf{v} \\
&= \max_{\mathbf{v}} \frac{1}{N}\sum_{i=1}^{N} \mathbf{v}^T (\mathbf{x}_i - \bar{\mathbf{x}}) \cdot (\mathbf{x}_i - \bar{\mathbf{x}})^T \mathbf{v} \\
&= \max_{\mathbf{v}} \mathbf{v}^T \left(\frac{1}{N}\sum_{i=1}^{N} (\mathbf{x}_i - \bar{\mathbf{x}}) \cdot (\mathbf{x}_i - \bar{\mathbf{x}})^T \right) \mathbf{v} \qquad = \max_{\mathbf{v}} \mathbf{v}^T \mathbf{S} \mathbf{v}
\end{aligned}
$$

PCA: Maximum Variance Formulation

- Covariance matrix $\mathbf{S} := \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}}) \cdot (\mathbf{x}_i - \bar{\mathbf{x}})^T$,

$$
\begin{aligned}
\max_{\mathbf{v}} \operatorname{Var}(\mathcal{X}\mathbf{v}) \quad &= \max_{\mathbf{v}} \frac{1}{N}\sum_{i=1}^{N} |(\mathbf{x}_i - \bar{\mathbf{x}})^T \mathbf{v}|^2 \\
&= \max_{\mathbf{v}} \frac{1}{N}\sum_{i=1}^{N} (\mathbf{x}_i - \bar{\mathbf{x}})^T \mathbf{v} \cdot (\mathbf{x}_i - \bar{\mathbf{x}})^T \mathbf{v} \\
&= \max_{\mathbf{v}} \frac{1}{N}\sum_{i=1}^{N} \mathbf{v}^T (\mathbf{x}_i - \bar{\mathbf{x}}) \cdot (\mathbf{x}_i - \bar{\mathbf{x}})^T \mathbf{v} \\
&= \max_{\mathbf{v}} \mathbf{v}^T \left(\frac{1}{N}\sum_{i=1}^{N} (\mathbf{x}_i - \bar{\mathbf{x}}) \cdot (\mathbf{x}_i - \bar{\mathbf{x}})^T \right) \mathbf{v} \qquad = \max_{\mathbf{v}} \mathbf{v}^T \mathbf{S} \mathbf{v}
\end{aligned}
$$

- $\max_{\mathbf{v}} \mathbf{v}^T \mathbf{S} \mathbf{v} = \lambda_{\max} = \mathbf{v}_*^T \mathbf{S} \mathbf{v}_*$, where $\mathbf{v}_*$ is the eigenvector corresponding to eigenvalue $\lambda_{\max}$.

PCA in Practice [2]

Algorithm (when $n > m$)

Input: Data $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \subseteq \mathbb{R}^n$, number of dimensions of the projected data m.

[2] Adapted from C.Igel

PCA in Practice [2]

Algorithm (when $n > m$)

Input: Data $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} \subseteq \mathbb{R}^n$, number of dimensions of the projected data m.

1. Compute sample mean $\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i$;

[2] Adapted from C.Igel

PCA in Practice [2]

Algorithm (when $n > m$)

Input: Data $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \subseteq \mathbb{R}^n$, number of dimensions of the projected data m.

1. Compute sample mean $\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$;
2. Compute sample data covariance $\mathbf{S} = \frac{1}{N}\sum_{i=1}^{N} \left(\mathbf{x}_i - \bar{\mathbf{x}}\right)\left(\mathbf{x}_i - \bar{\mathbf{x}}\right)^T$;

[2] Adapted from C.Igel

PCA in Practice [2]

Algorithm (when $n > m$)

Input: Data $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \subseteq \mathbb{R}^n$, number of dimensions of the projected data m.

1. Compute sample mean $\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$;
2. Compute sample data covariance $\mathbf{S} = \frac{1}{N}\sum_{i=1}^{N} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$;
3. Perform eigenvalue decomposition as $\mathbf{S} = \mathbf{V}\Lambda\mathbf{V}^T$, where $\mathbf{V} \in \mathbb{R}^{n\times n}$ is a matrix with eigenvectors, $\Lambda \in \mathbb{R}^{n\times n}$ is a diagonal matrix with eigenvalues;

[2] Adapted from C.Igel

PCA in Practice [2]

Algorithm (when $n > m$)

Input: Data $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} \subseteq \mathbb{R}^n$, number of dimensions of the projected data m.

1. Compute sample mean $\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$;
2. Compute sample data covariance $\mathbf{S} = \frac{1}{N}\sum_{i=1}^{N} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$;
3. Perform eigenvalue decomposition as $\mathbf{S} = \mathbf{V}\Lambda\mathbf{V}^T$, where $\mathbf{V} \in \mathbb{R}^{n\times n}$ is a matrix with eigenvectors, $\Lambda \in \mathbb{R}^{n\times n}$ is a diagonal matrix with eigenvalues;
4. Collect the first m eigenvectors of $\mathbf{S}$ from $\mathbf{V}$ sorted by decreasing eigenvalue into $\mathbf{U}$;

[2] Adapted from C.Igel

PCA in Practice [2]

Algorithm (when $n > m$)

Input: Data $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \subseteq \mathbb{R}^n$, number of dimensions of the projected data m.

1. Compute sample mean $\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$;
2. Compute sample data covariance $\mathbf{S} = \frac{1}{N}\sum_{i=1}^{N} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$;
3. Perform eigenvalue decomposition as $\mathbf{S} = \mathbf{V}\Lambda\mathbf{V}^T$, where $\mathbf{V} \in \mathbb{R}^{n\times n}$ is a matrix with eigenvectors, $\Lambda \in \mathbb{R}^{n\times n}$ is a diagonal matrix with eigenvalues;
4. Collect the first m eigenvectors of $\mathbf{S}$ from $\mathbf{V}$ sorted by decreasing eigenvalue into $\mathbf{U}$;
5. Compute $\tilde{\mathbf{x}}_i = \mathbf{U}^T\mathbf{x}_i$ for $i = 1, \ldots, N$.

[2] Adapted from C.Igel

PCA in Practice [2]

Algorithm (when $n > m$)

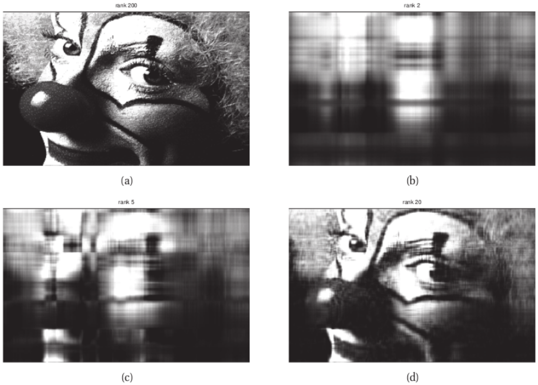
Input: Data $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \subseteq \mathbb{R}^n$, number of dimensions of the projected data m.

1. Compute sample mean $\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$;
2. Compute sample data covariance $\mathbf{S} = \frac{1}{N}\sum_{i=1}^{N} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$;
3. Perform eigenvalue decomposition as $\mathbf{S} = \mathbf{V}\Lambda\mathbf{V}^T$, where $\mathbf{V} \in \mathbb{R}^{n\times n}$ is a matrix with eigenvectors, $\Lambda \in \mathbb{R}^{n\times n}$ is a diagonal matrix with eigenvalues;
4. Collect the first m eigenvectors of $\mathbf{S}$ from $\mathbf{V}$ sorted by decreasing eigenvalue into $\mathbf{U}$;
5. Compute $\tilde{\mathbf{x}}_i = \mathbf{U}^T\mathbf{x}_i$ for $i = 1, \ldots, N$.

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

[2] Adapted from C.Igel

Example of PCA based dimensionality reduction [3]



(a) (b) (c) (d)

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.

[3] from Kevin Murphy, Probabilistic Machine Learning

Summary: PCA

-+ Curse of dimensionality;

+ Data is projected orthogonally into *linear* subspace;

+ Dimensionality reduction while maximizing variance;

+ Quantifiable loss of information with "explained variance";

+ Singular Value Decomposition for cheaper computation;

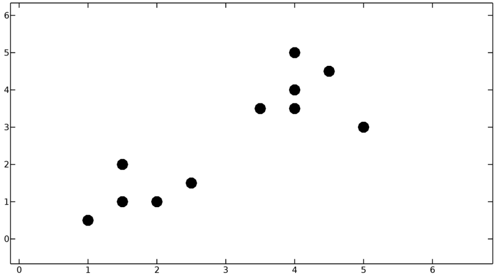
- Lossy compression;

- For some datasets $m \approx n$.

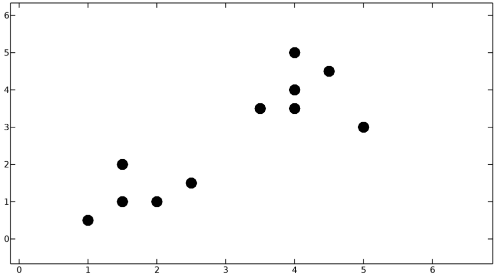
K-Means Clustering

- Process of grouping similar objects together;
- Detecting similar patterns or features;
- Representing data at higher abstractions;
- Applications like image segmentation.

Toy example:

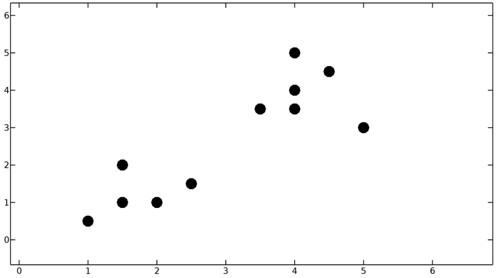


Toy example:



How would you cluster these points?

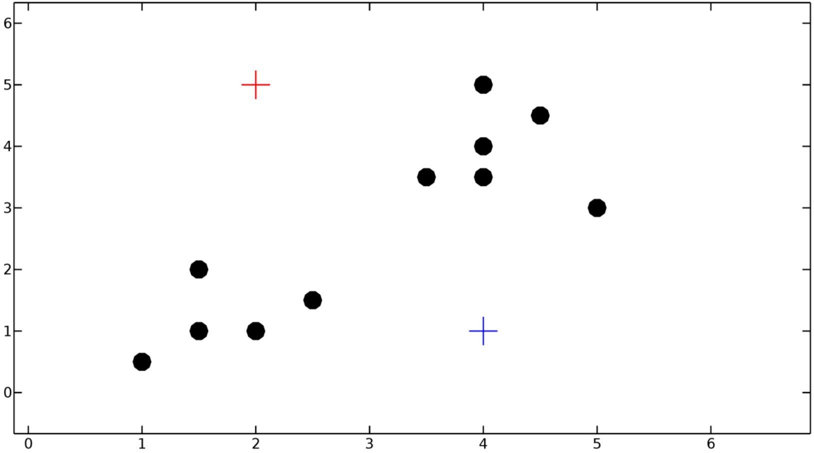
Toy example:



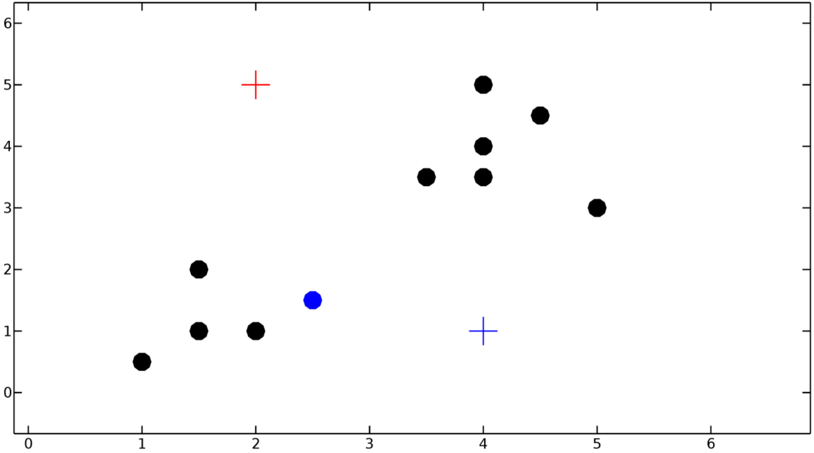
How would you cluster these points?

- Location of centroids;
- Assign labels by closest neighbors;
- Within-cluster variance.

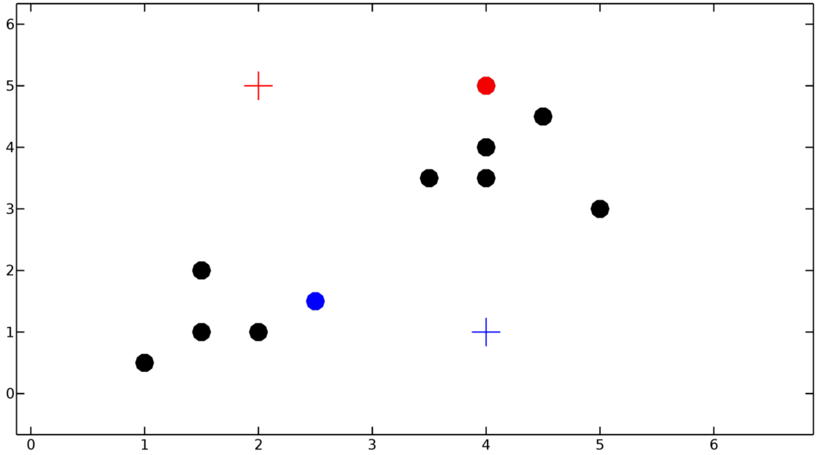
Initialize centroids, randomly!



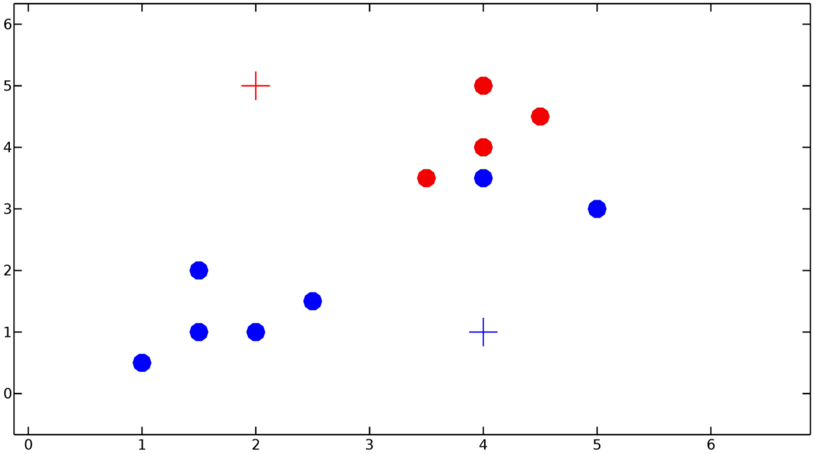
Assign points to nearest centroid!



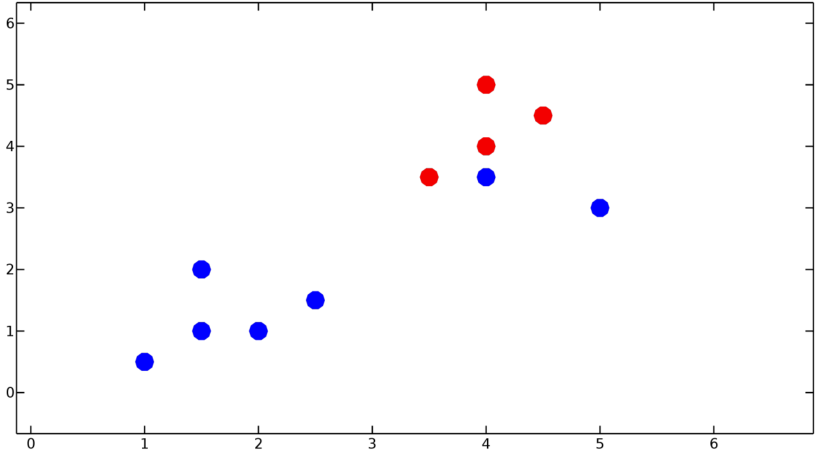
Assign points to nearest centroid!



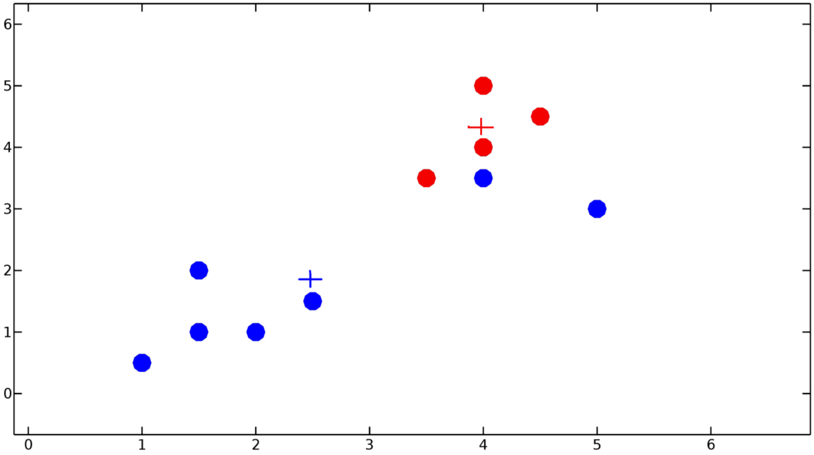
Assign points to nearest centroid!



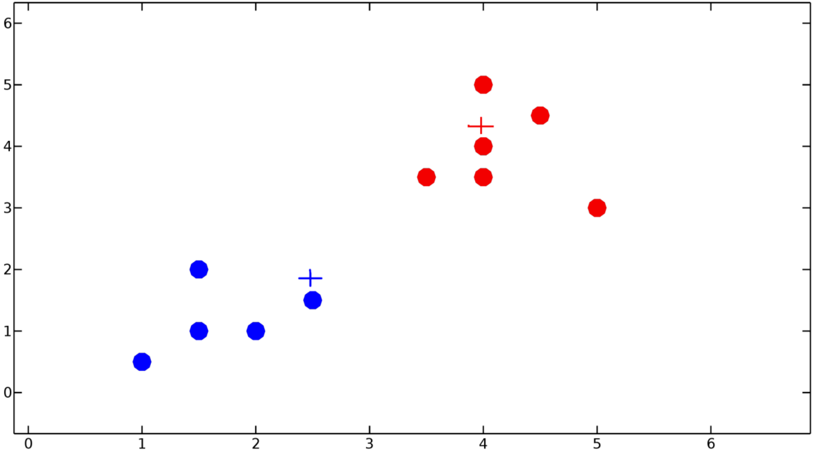
Recompute centroids!



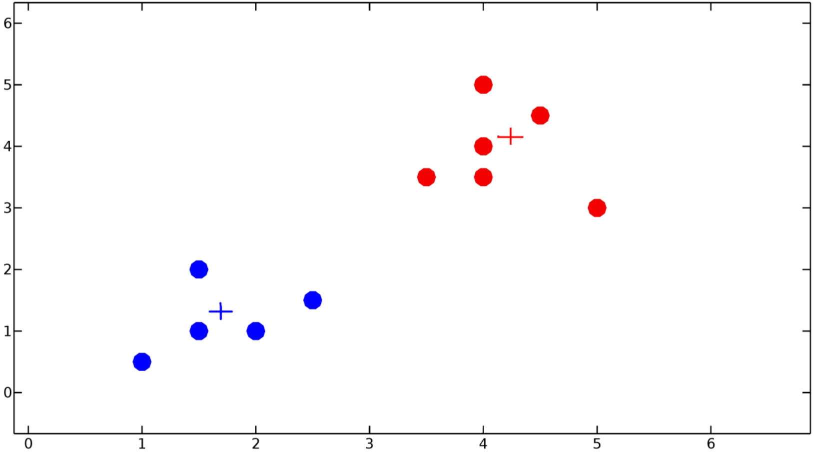
Recompute centroids!



Iterate, until convergence!



Iterate, until convergence!



Formalizing K-Means Clustering[5]

Given a dataset $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \subseteq \mathbb{R}^n$. Our objective is:

[4] Naive K-means, aka. Lloyd's algorithm

[5] Adapted from from C.Igel

Formalizing K-Means Clustering[5]

Given a dataset $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \subseteq \mathbb{R}^n$. Our objective is:

$$\min_{\mathcal{X}_i} \sum_{i=1}^{k} \sum_{\mathbf{x}\in\mathcal{X}_i} \|\mathbf{x} - \mu_i\|^2 = \min_{\mathcal{X}_i} \sum_{i=1}^{k} |\mathcal{X}_i| \operatorname{Var}(\mathcal{X}_i), \text{ where } \mu_i = \frac{1}{|\mathcal{X}_i|} \sum_{\mathbf{x}\in\mathcal{X}_i} \mathbf{x}.$$

[4] Naive K-means, aka. Lloyd's algorithm

[5] Adapted from from C.Igel

Formalizing K-Means Clustering[5]

Given a dataset $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \subseteq \mathbb{R}^n$. Our objective is:

$$\min_{\mathcal{X}_i} \sum_{i=1}^{k} \sum_{\mathbf{x} \in \mathcal{X}_i} \|\mathbf{x} - \mu_i\|^2 = \min_{\mathcal{X}_i} \sum_{i=1}^{k} |\mathcal{X}_i| \operatorname{Var}(\mathcal{X}_i), \text{ where } \mu_i = \frac{1}{|\mathcal{X}_i|} \sum_{\mathbf{x} \in \mathcal{X}_i} \mathbf{x}.$$

Iterate[4]:

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

[4] Naive K-means, aka. Lloyd's algorithm
[5] Adapted from from C.Igel

Formalizing K-Means Clustering[5]

Given a dataset $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \subseteq \mathbb{R}^n$. Our objective is:

$$\min_{\mathcal{X}_i} \sum_{i=1}^{k} \sum_{\mathbf{x} \in \mathcal{X}_i} \|\mathbf{x} - \mu_i\|^2 = \min_{\mathcal{X}_i} \sum_{i=1}^{k} |\mathcal{X}_i| \operatorname{Var}(\mathcal{X}_i), \text{ where } \mu_i = \frac{1}{|\mathcal{X}_i|} \sum_{\mathbf{x} \in \mathcal{X}_i} \mathbf{x}.$$

Iterate[4]:

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] Naive K-means, aka. Lloyd's algorithm

[5] Adapted from from C.Igel

Formalizing K-Means Clustering[5]

Given a dataset $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \subseteq \mathbb{R}^n$. Our objective is:

$$\min_{\mathcal{X}_i} \sum_{i=1}^{k} \sum_{\mathbf{x} \in \mathcal{X}_i} \|\mathbf{x} - \mu_i\|^2 = \min_{\mathcal{X}_i} \sum_{i=1}^{k} |\mathcal{X}_i| \operatorname{Var}(\mathcal{X}_i), \text{ where } \mu_i = \frac{1}{|\mathcal{X}_i|} \sum_{\mathbf{x} \in \mathcal{X}_i} \mathbf{x}.$$

Iterate[4]:

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.

Can we formulate K-means as $\min_{\mathcal{X}_i} \sum_{i=1}^{k} \operatorname{Var}(\mathcal{X}_i)$? Why?

[4] Naive K-means, aka. Lloyd's algorithm

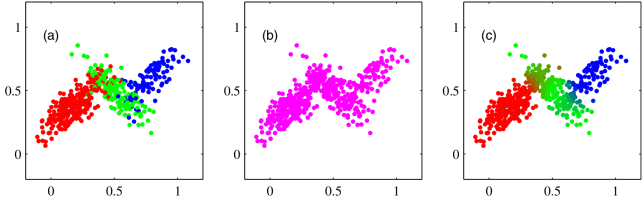
[5] Adapted from from C.Igel

K-Means clustering based Image Segmentation [6]

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.

[6] from Christopher Bishop, PRML

Summary: K-Means Clustering [7]



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

[7] Fig. 9.5 from Christopher Bishop, PRML