Course 3: Unsupervised Learning

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Summary

Last session

1. Supervised learning - learning from labeled examples
2. Bias/variance tradeoff
3. Overfitting and cross-validation
4. Curse of dimensionality
5. Computational requirements

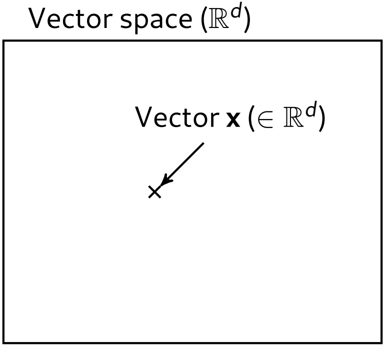
Today's session

1. Learning from Unlabeled examples
2. Clustering
3. Decomposition
4. Manifold learning
5. Feature Selection and preprocessing

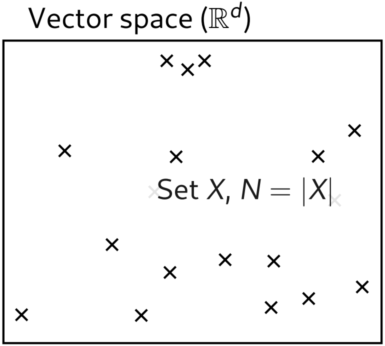
Notations

Vector space ($\mathbb{R}^d$)

Notations



Notations



Unsupervised learning

Goal

Discover patterns/structure in X,

Unsupervised learning

- Unsupervised = no expert, no labels
- Main approaches:
  - Clustering = find a partition of X in K subsets,
  - Decomposition using K vectors,
  - Manifold Learning.
- Applications :
  - Dimensionality reduction, Quantization
  - Visualization…

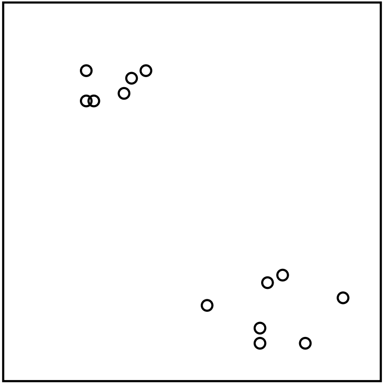
Unsupervised learning

Goal

Discover patterns/structure in X,

Unsupervised learning

- Unsupervised = no expert, no labels
- Main approaches:
  - Clustering = find a partition of X in K subsets,
  - Decomposition using K vectors.
  - Manifold Learning.
- Applications :
  - Dimensionality reduction, Quantization
  - Visualization...

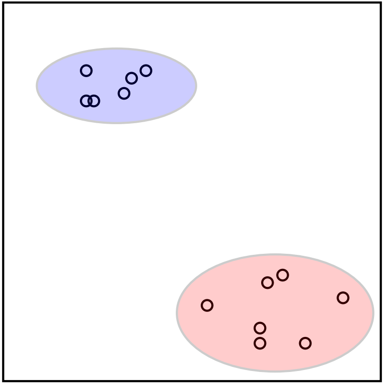


Goal

Discover patterns/structure in X,

Unsupervised learning

- Unsupervised = no expert, no labels
- Main approaches:
  - Clustering = find a partition of X in K subsets,
  - Decomposition using K vectors.
  - Manifold Learning.
- Applications :
  - Dimensionality reduction, Quantization
  - Visualization...

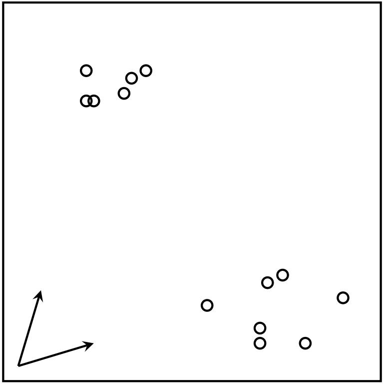


Goal

Discover patterns/structure in X,

Unsupervised learning

- Unsupervised = no expert, no labels
- Main approaches:
  - Clustering = find a partition of X in K subsets,
  - Decomposition using K vectors.
  - Manifold Learning.
- Applications :
  - Dimensionality reduction, Quantization
  - Visualization...



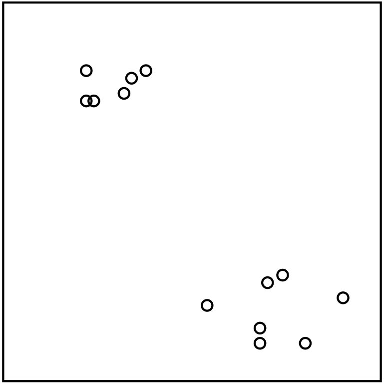
Unsupervised learning

Goal

Discover patterns/structure in X,

Unsupervised learning

- Unsupervised = no expert, no labels
- Main approaches:
  - Clustering = find a partition of X in K subsets,
  - Decomposition using K vectors.
  - Manifold Learning.
- Applications :
  - Dimensionality reduction, Quantization
  - Visualization…

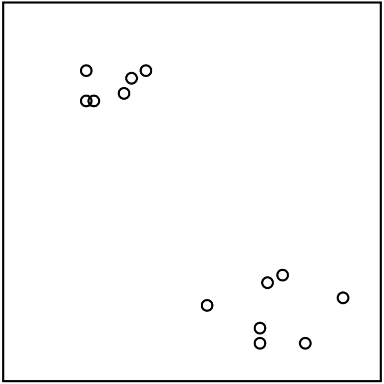


Goal

Discover patterns/structure in X,

Unsupervised learning

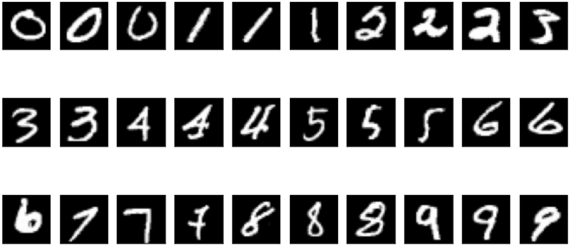
- Unsupervised = no expert, no labels
- Main approaches:
  - Clustering = find a partition of X in K subsets,
  - Decomposition using K vectors.
  - Manifold Learning.
- Applications :
  - Dimensionality reduction, Quantization
  - Visualization…



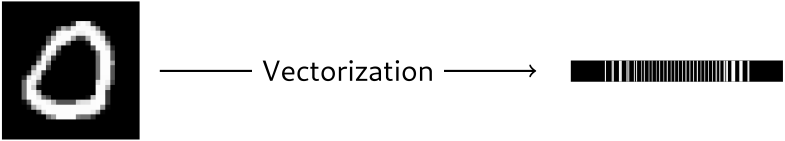
A classical dataset: MNIST dataset (1/2)

MNIST Dataset

- "Toy" dataset (=small and easy)
- 60000 + 10000 handwritten digits

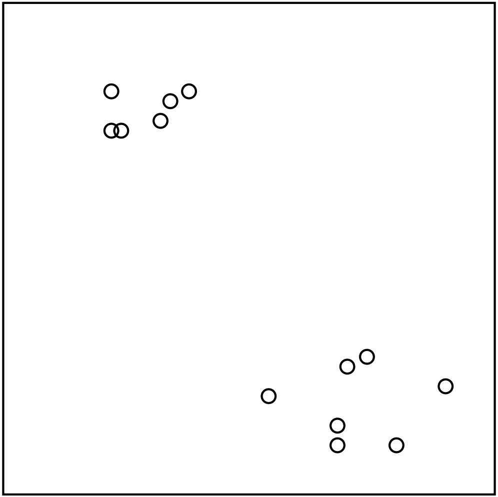


A classical dataset: MNIST dataset (2/2)

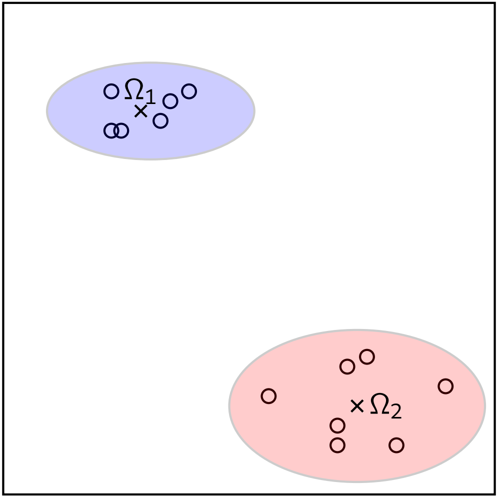


Hence, all images are interpreted as 1D vectors!

Example: clustering using L_2 norm (1/8)



Example: clustering using L_2 norm (1/8)



Example: clustering using L_2 norm (2/8)

An example to perform clustering is to rely on distances to centroids.
We define K *cluster centroids* $\Omega_k, \forall k \in [1..K]$.
Here, each vector is associated with the cluster whose centroid is of minimal distance.

Definitions

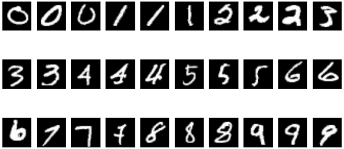
We denote $q : \mathbb{R}^d \to [1..K]$ a function that associates a vector $\mathbf{x}$ with the index of (one of) its closest centroid $\Omega_{q(\mathbf{x})}$. Formally:

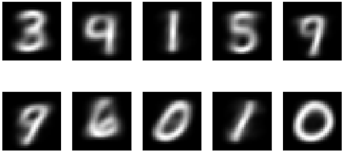
- $\forall k \in [1..K], \Omega_k \in \mathbb{R}^d$
- $\forall \mathbf{x} \in X, \forall j \in [1..K], \|\mathbf{x} - \Omega_{q(\mathbf{x})}\|_2 \leq \|\mathbf{x} - \Omega_j\|_2$
- Error $E(q) \triangleq \sum_{\mathbf{x} \in X} \|\mathbf{x} - \Omega_{q(\mathbf{x})}\|_2$
- $X = \bigcup_k \underbrace{\{\mathbf{x} \in X, q(\mathbf{x}) = k\}}_{\text{cluster } k}$

Clustering using L_2 norm (3/8)

Clustering MNIST

Using K-means algorithm with $K = 10$





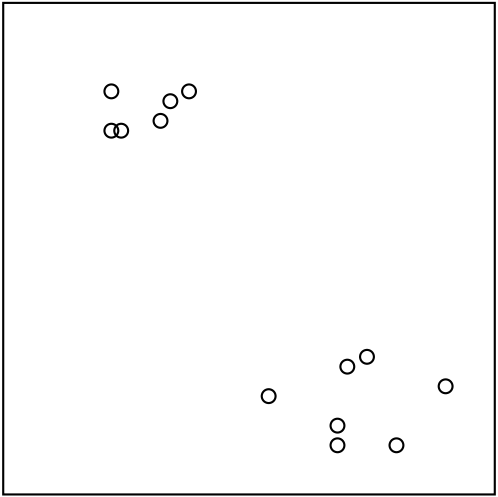
Note: we recall that images are vectorized for the clustering to make sense!
They are only displayed in 2D to be interpretable.

Clustering using L_2 norm (4/8)

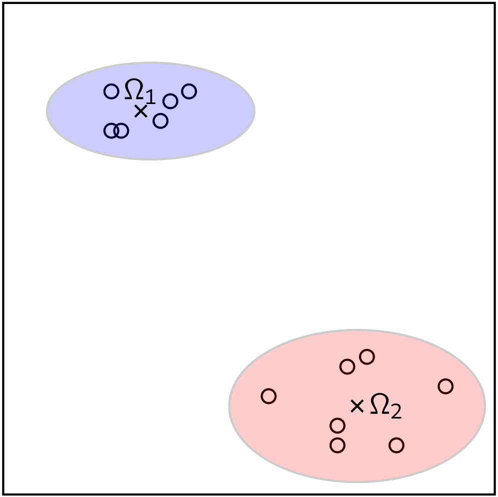
Quantizing MNIST

- Replace $\mathbf{x}$ by $\Omega_{k(\mathbf{x})}$
- Compression factor $\kappa = 1 - K/N$

Clustering using L_2 norm (5/8): Choosing K

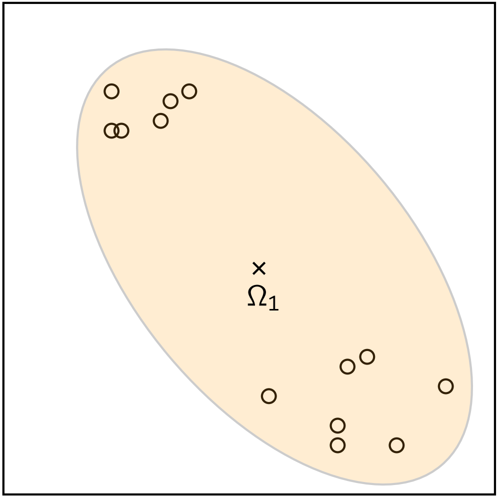


Clustering using L_2 norm (5/8): Choosing K



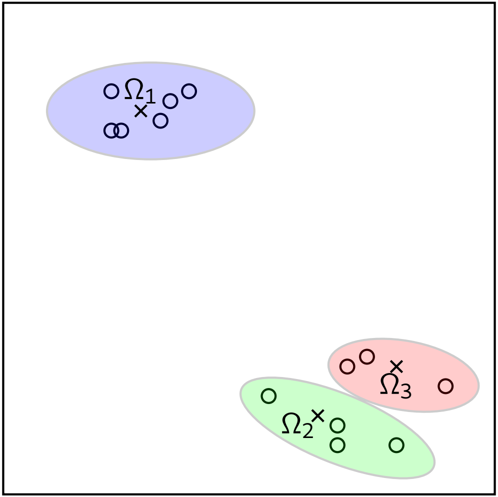
$K = 2$

Clustering using L_2 norm (5/8): Choosing K



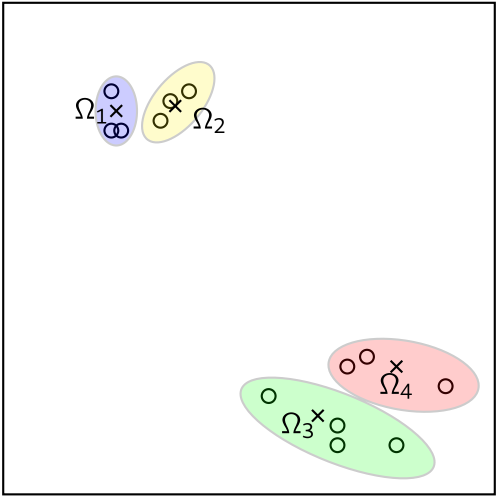
$K = 1$

Clustering using L_2 norm (5/8): Choosing K



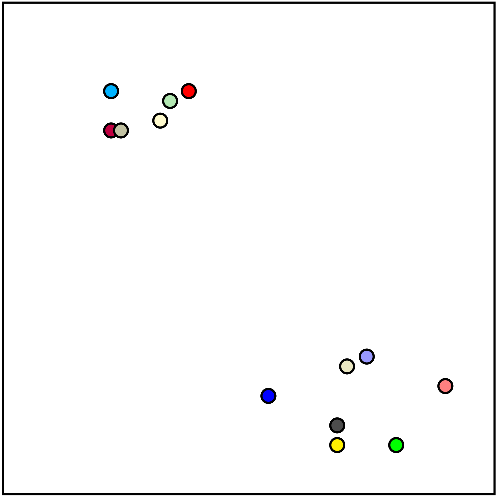
$K = 3$

Clustering using L_2 norm (5/8): Choosing K



$K = 4$

Clustering using L_2 norm (5/8): Choosing K



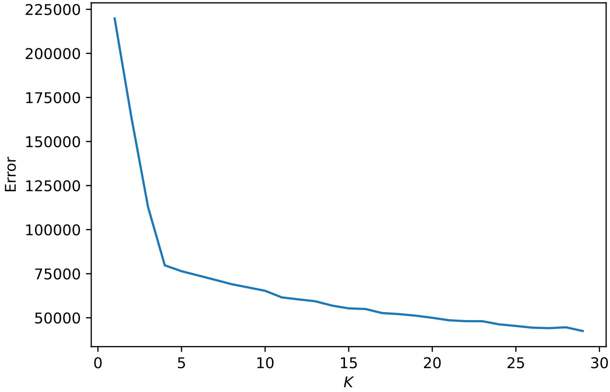
$K = N$

(each data point is its own centroid)

Clustering using L_2 norm (6/8)

Choosing K

- Finding a compromise between error and compression,
- Simple practical method : "elbow".



Clustering using L_2 norm (7/8)

Optimal clustering

- Define $E_{opt_K}(q^*) \triangleq \arg \min_{q:\mathbb{R}^d \to [1..K]} E(q)$,
- Finding an optimal clustering is an NP-hard problem.

Properties

- $0 = E_{opt_N}(q^*) \leq E_{opt_{N-1}}(q^*) \leq \cdots \leq E_{opt_1}(q^*) = var(X)$,
  - Proof: monotonicity by particularization, extremes with identity function (left) and variance (right).
- $0 \leq K \leq \frac{N-1}{N}$.

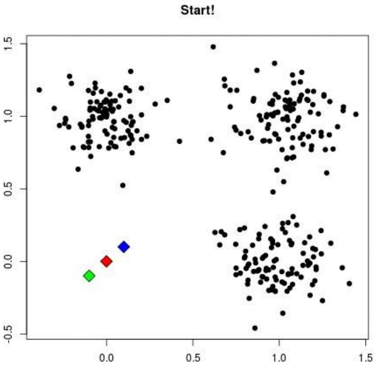
Changing the number of centroids changes the clustering... And the signification of clusters.

Clustering using L_2 norm (8/8)

K-means algorithm

First: initialize K cluster centroids.

1. Assign each data point to the cluster of closest centroid.
2. Compute the new centroids as the average of the data points in each cluster.
3. Repeat.

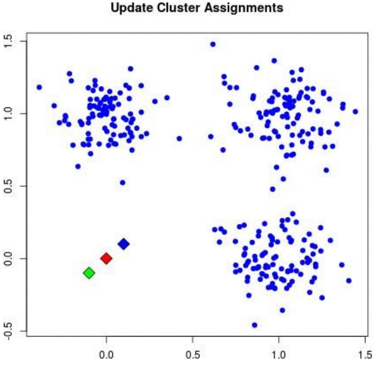


Reference: https://mubaris.com/posts/kmeans-clustering/

K-means algorithm

First: initialize K cluster centroids.

1. Assign each data point to the cluster of closest centroid.
2. Compute the new centroids as the average of the data points in each cluster.
3. Repeat.



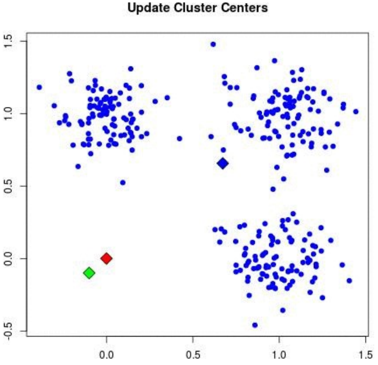
Reference: https://mubaris.com/posts/kmeans-clustering/

Clustering using L_2 norm (8/8)

K-means algorithm

First: initialize K cluster centroids.

1. Assign each data point to the cluster of closest centroid.
2. Compute the new centroids as the average of the data points in each cluster.
3. Repeat.

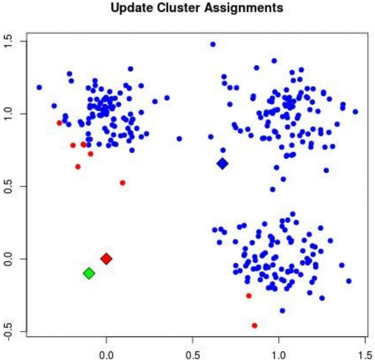


Reference: https://mubaris.com/posts/kmeans-clustering/

K-means algorithm

First: initialize K cluster centroids.

1. Assign each data point to the cluster of closest centroid.
2. Compute the new centroids as the average of the data points in each cluster.
3. Repeat.



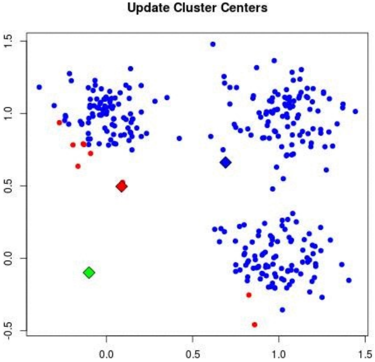
Reference: https://mubaris.com/posts/kmeans-clustering/

Clustering using L_2 norm (8/8)

K-means algorithm

First: initialize K cluster centroids.

1. Assign each data point to the cluster of closest centroid.
2. Compute the new centroids as the average of the data points in each cluster.
3. Repeat.



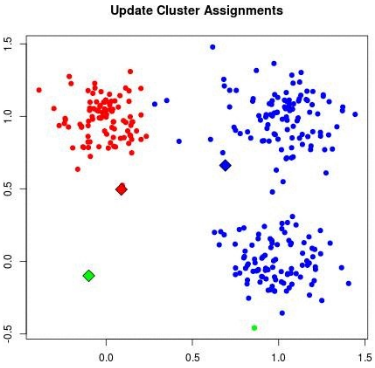
Reference: https://mubaris.com/posts/kmeans-clustering/

Clustering using L_2 norm (8/8)

K-means algorithm

First: initialize K cluster centroids.

1. Assign each data point to the cluster of closest centroid.
2. Compute the new centroids as the average of the data points in each cluster.
3. Repeat.



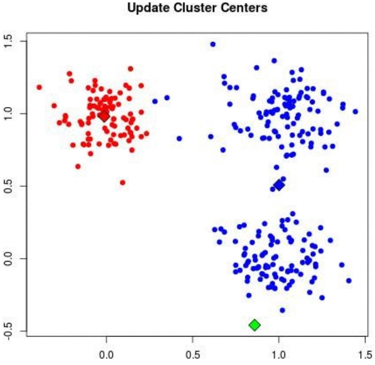
Reference: https://mubaris.com/posts/kmeans-clustering/

Clustering using L_2 norm (8/8)

K-means algorithm

First: initialize K cluster centroids.

1. Assign each data point to the cluster of closest centroid.
2. Compute the new centroids as the average of the data points in each cluster.
3. Repeat.

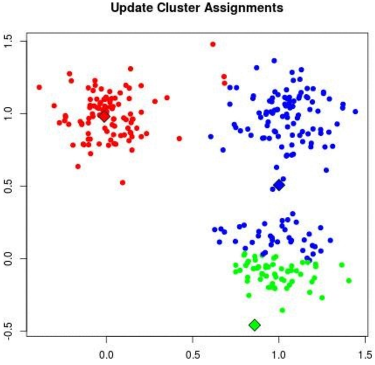


Reference: https://mubaris.com/posts/kmeans-clustering/

K-means algorithm

First: initialize K cluster centroids.

1. Assign each data point to the cluster of closest centroid.
2. Compute the new centroids as the average of the data points in each cluster.
3. Repeat.



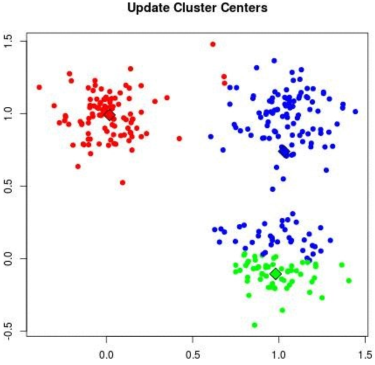
Reference: https://mubaris.com/posts/kmeans-clustering/

Clustering using L_2 norm (8/8)

K-means algorithm

First: initialize *K* cluster centroids.

1. Assign each data point to the cluster of closest centroid.
2. Compute the new centroids as the average of the data points in each cluster.
3. Repeat.



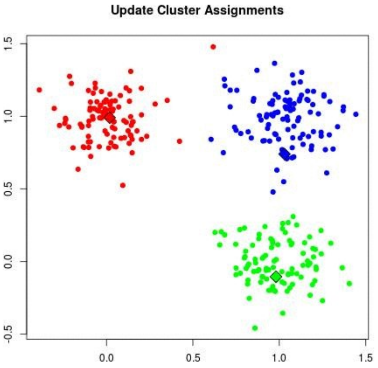
Reference: https://mubaris.com/posts/kmeans-clustering/

Clustering using L_2 norm (8/8)

K-means algorithm

First: initialize K cluster centroids.

1. Assign each data point to the cluster of closest centroid.
2. Compute the new centroids as the average of the data points in each cluster.
3. Repeat.



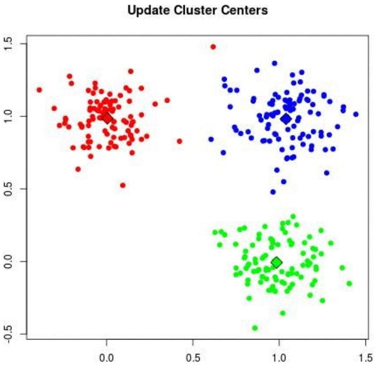
Reference: https://mubaris.com/posts/kmeans-clustering/

Clustering using L_2 norm (8/8)

K-means algorithm

First: initialize K cluster centroids.

1. Assign each data point to the cluster of closest centroid.
2. Compute the new centroids as the average of the data points in each cluster.
3. Repeat.



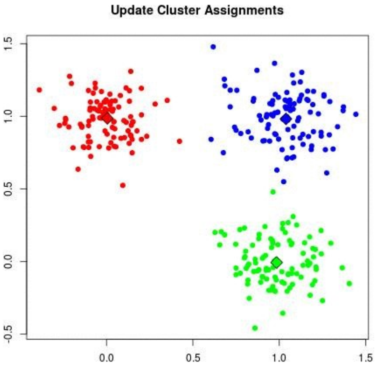
Reference: https://mubaris.com/posts/kmeans-clustering/

Clustering using L_2 norm (8/8)

K-means algorithm

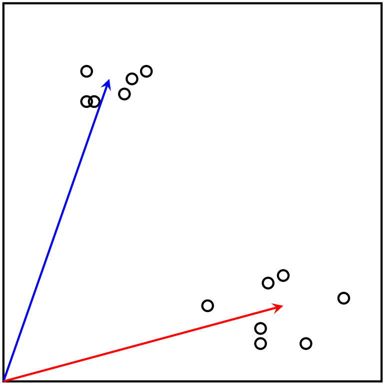
First: initialize K cluster centroids.

1. Assign each data point to the cluster of closest centroid.
2. Compute the new centroids as the average of the data points in each cluster.
3. Repeat.

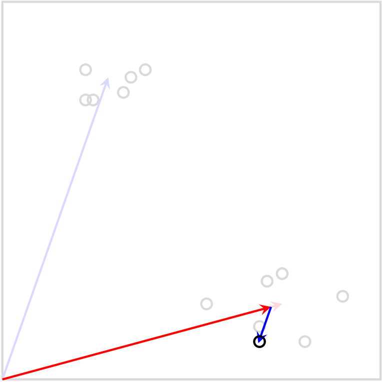


Reference: https://mubaris.com/posts/kmeans-clustering/

Decomposition

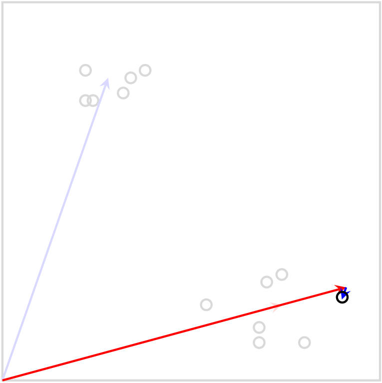


Decomposition



$$x = {\color{red}0.96} \times {\color{red}\longrightarrow} + {\color{blue}-0.12} \times {\color{blue}\longrightarrow}$$

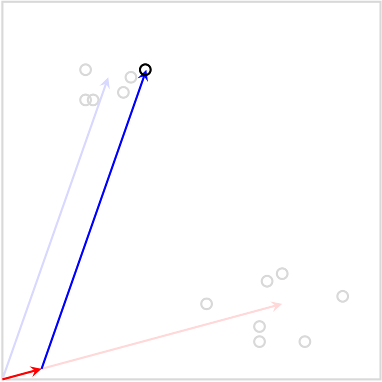
Decomposition



x = 1.23 × 

\+ -0.04 × 

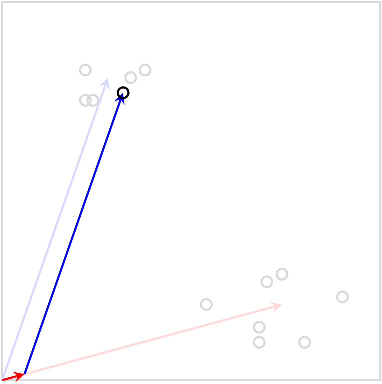
Decomposition



x = 0.14 × 

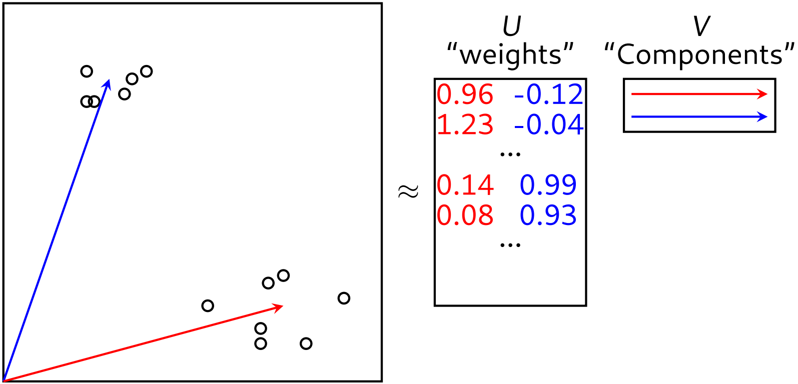
+ 0.99 × 

Decomposition



$$x = 0.08 \times \longrightarrow$$
$$+\ 0.93 \times \longrightarrow$$

Decomposition

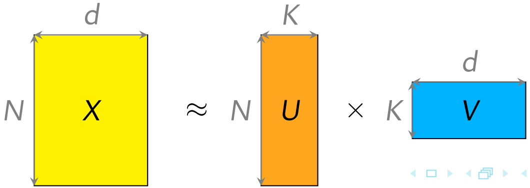


Principal Components Analysis

Definitions

Principal components analysis solves the following matrix factorization problem:

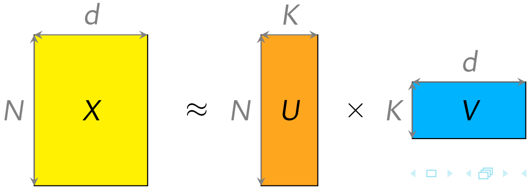
- The set X is considered as a matrix $X \in \mathcal{M}_{N\times d}(\mathbb{R})$,
- We consider decompositions using components $V \in \mathcal{M}_{K\times d}(\mathbb{R})$ and weights $U \in \mathcal{M}_{N\times k}(\mathbb{R})$,
- PCA estimates K components that are orthogonal and ordered by importance (variance explained)
- It is based on the Singular Value Decomposition (SVD) of the covariance matrix $XX^\top$



Definitions

Principal components analysis solves the following matrix factorization problem:

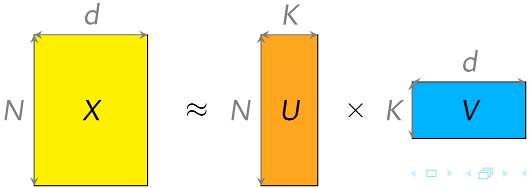
- The set X is considered as a matrix $X \in \mathcal{M}_{N\times d}(\mathbb{R})$,
- We consider decompositions using components $V \in \mathcal{M}_{K\times d}(\mathbb{R})$ and weights $U \in \mathcal{M}_{N\times k}(\mathbb{R})$,
- PCA estimates K components that are orthogonal and ordered by importance (variance explained)
- It is based on the Singular Value Decomposition (SVD) of the covariance matrix $XX^\top$



Definitions

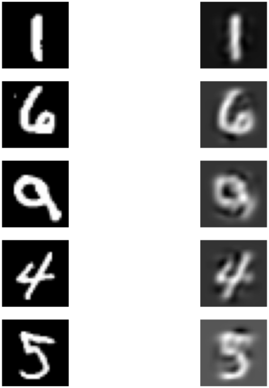
Principal components analysis solves the following matrix factorization problem:

- The set X is considered as a matrix $X \in \mathcal{M}_{N\times d}(\mathbb{R})$,
- We consider decompositions using components $V \in \mathcal{M}_{K\times d}(\mathbb{R})$ and weights $U \in \mathcal{M}_{N\times k}(\mathbb{R})$,
- PCA estimates K components that are orthogonal and ordered by importance (variance explained)
- It is based on the Singular Value Decomposition (SVD) of the covariance matrix $XX^\top$



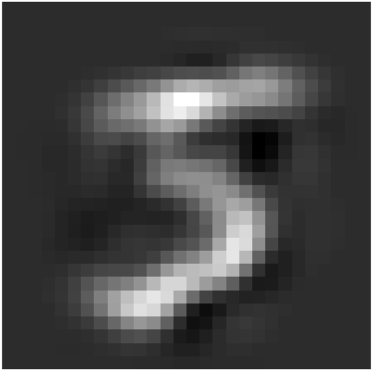
Principal Components Analysis

Example of reconstructions on MNIST with $K = 32$

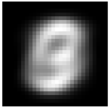
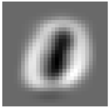


Recall that each image is vectorized, hence each of these images correspond to a row in V.

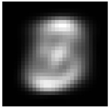
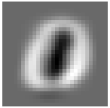
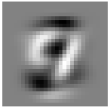
Detailed example of a reconstruction



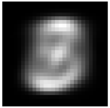
Principal Components Analysis

 $= 122.3\times$ 

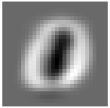
Principal Components Analysis

 $= 122.3\times$  $-316.2\times$ 

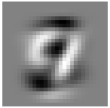
Principal Components Analysis



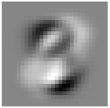
$= 122.3\times$



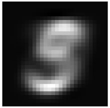
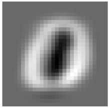
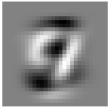
$-316.2\times$

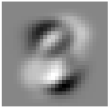


$-51.13\times$

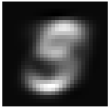


Principal Components Analysis

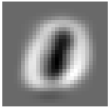
 $= 122.3\times$  $-316.2\times$ 

$-51.13\times$  $-556.9\times$ 

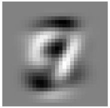
Principal Components Analysis



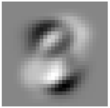
$= 122.3\times$



$-316.2\times$

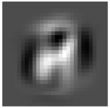


$-51.13\times$

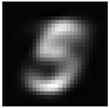
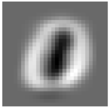
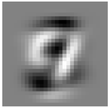


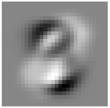
$-556.9\times$

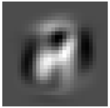
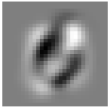
$-49.2\times$



Principal Components Analysis

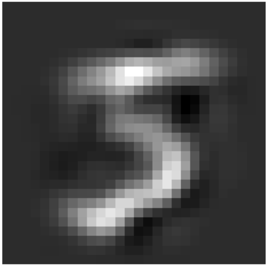
 $= 122.3\times$  $-316.2\times$ 

$-51.13\times$  $-556.9\times$ 

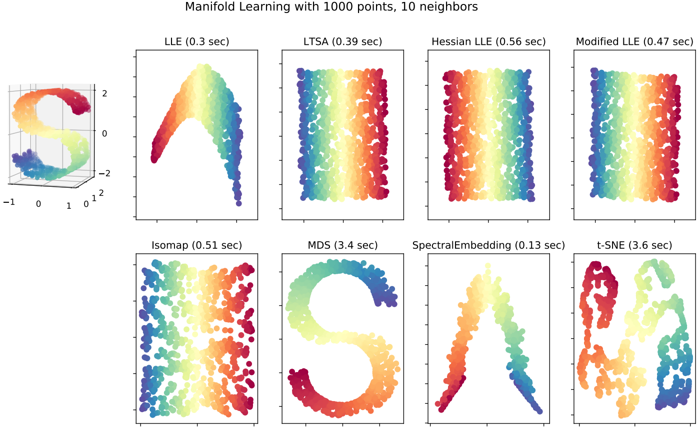
$-49.2\times$  $-217.1\times$  $\dots$

Principal Components Analysis

Reconstruction with all 32 components:



Example 3: Manifold Learning



Approaches to uncover lower dimensional structure of high dimensional data. Source : Manifold module, sklearn website

Working with features

N.b. : valid in unsupervised and supervised settings.

Feature preprocessing

Objective : change the statistical distribution of the features

- Scaling / Normalization
- Power transform
- Encode, discretization
- Manual feature engineering
- See more `https://scikit-learn.org/stable/modules/preprocessing.html`

Many techniques need or are greatly helped when features are on the unit sphere.

Working with features

N.b. : valid in unsupervised and supervised settings.

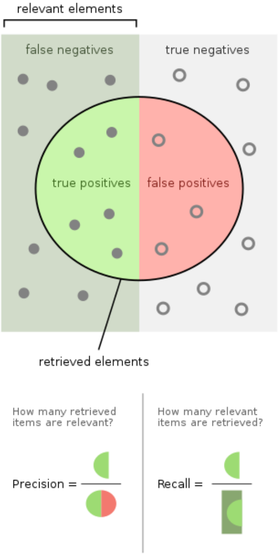
Feature selection

Objective : remove features

- Remove features with low variance
- Select features according to their explained variance towards labels (e.g. SelectKBest)
- See more `https://scikit-learn.org/stable/modules/feature_selection.html`

Helps to adress the dimensionality curse.

In supervised learning : per class metric



Metrics

Clustering Metrics :

- Error defined slide 8 : similar to inertia (sum of squared distances)
- The Silhouette score (per sample) is $(b - a)/max(a, b)$, with mean intra-cluster distance (a) and the mean nearest-cluster distance (b).

Clustering metrics using labels :

- Random Index : measures the similarity of two assignments, ignoring permutations
- Homogeneity : each cluster contains only members of a single class.
- Completeness : all members of a given class are assigned to the same cluster.

See more on sklearn website and in the lab session

Metrics

Clustering Metrics :

- Error defined slide 8 : similar to inertia (sum of squared distances)
- The Silhouette score (per sample) is $(b - a)/max(a, b)$, with mean intra-cluster distance (a) and the mean nearest-cluster distance (b).

Clustering metrics using labels :

- Random Index : measures the similarity of two assignments, ignoring permutations
- Homogeneity : each cluster contains only members of a single class.
- Completeness : all members of a given class are assigned to the same cluster.

See more on sklearn website and in the lab session

Metrics

Clustering Metrics :

- Error defined slide 8 : similar to inertia (sum of squared distances)
- The Silhouette score (per sample) is $(b - a)/max(a, b)$, with mean intra-cluster distance (a) and the mean nearest-cluster distance (b).

Clustering metrics using labels :

- Random Index : measures the similarity of two assignments, ignoring permutations
- Homogeneity : each cluster contains only members of a single class.
- Completeness : all members of a given class are assigned to the same cluster.

See more on sklearn website and in the lab session

Metrics

Clustering Metrics :

- Error defined slide 8 : similar to inertia (sum of squared distances)
- The Silhouette score (per sample) is $(b - a)/max(a, b)$, with mean intra-cluster distance (a) and the mean nearest-cluster distance (b).

Clustering metrics using labels :

- Random Index : measures the similarity of two assignments, ignoring permutations
- Homogeneity : each cluster contains only members of a single class.
- Completeness : all members of a given class are assigned to the same cluster.

See more on sklearn website and in the lab session

Lab Session 3 and assignment (1/2)

Lab Unsupervised Learning

- Feature selection and preprocessing
- K-means clustering
- Principal Component Analysis (PCA)
- Tests on the modality chosen in Lab 1 (text, vision or audio), based on the same foundation model as in Lab 1.

Project 2 (P2)

You will choose a couple of unsupervised learning methods among those available (see Lab 3). You will present

- A brief description of the theory behind these methods,
- Basic tests on these techniques for your modality.

During Session 4 you will have 15 minutes to present.

Lab Session 3 and assignment (2/2)

List of Unsupervised Learning Methods

- Non-Negative Matrix Factorization & DBSCAN
- Spectral Clustering & Gaussian Mixture Models
- Agglomerative Clustering & UMAP