Course 3: Unsupervised Learning

2025-10-16

Course 3: Unsupervised Learning

Course 3: Unsupervised Learning

Summary

Last session

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

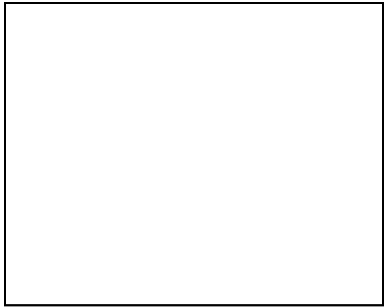
Today's session

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

Summary
Last session
1 Supervised learning - learning from labeled examples
2 Bias/variance tradeoff
3 Overfitting
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

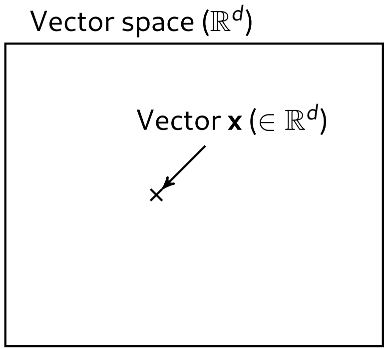
Notations

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

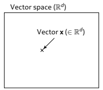


Notations
Vector space ($\mathbb{R}^d$)

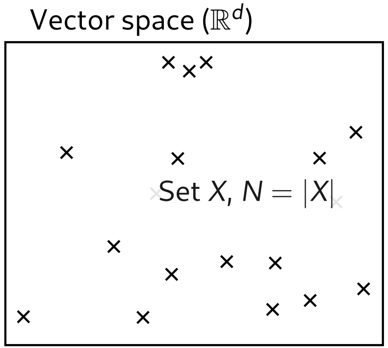
Notations



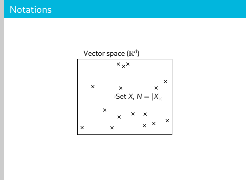
Notations



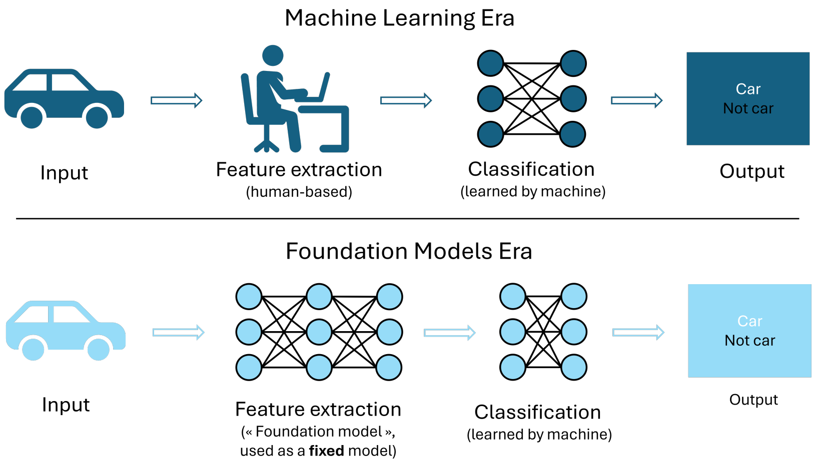
Notations



2025-10-16
Course 3: Unsupervised Learning
Notations



What is the vector x? (1/2)

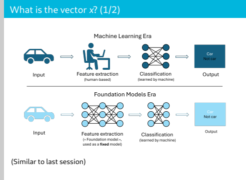


(Similar to last session)

2025-10-16

Course 3: Unsupervised Learning

└What is the vector x? (1/2)



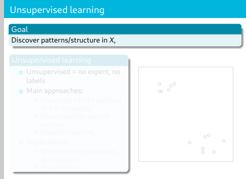
Unsupervised learning

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…

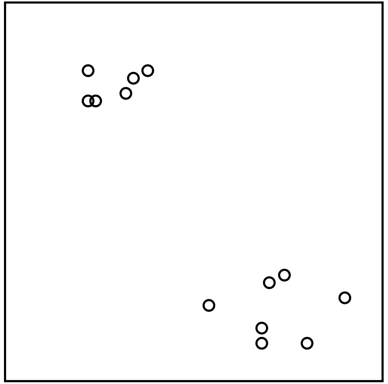


Goal

Discover patterns/structure in X,

Unsupervised learning

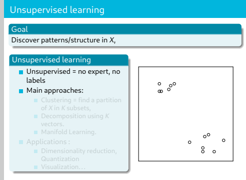
- Unsupervised = no expert, no labels
- Main approaches:
  - Clustering = find a partition of X in K subsets,
  - Decomposition using K vectors.
  - Manifold Learning.
- Applications :
  - Dimensionality reduction, Quantization
  - Visualization…



2025-10-16

Course 3: Unsupervised Learning

└Unsupervised learning



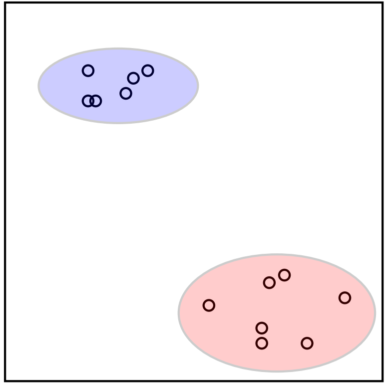
Unsupervised learning

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…

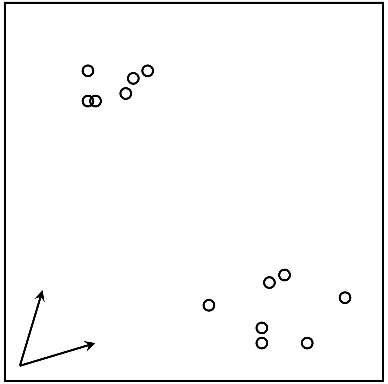
Unsupervised learning

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...

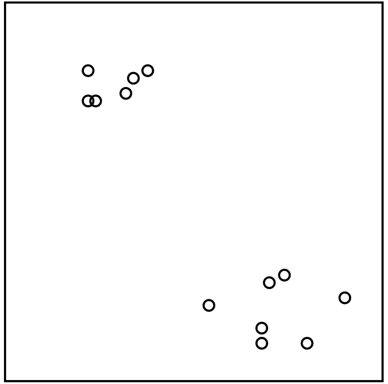
Unsupervised learning

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...



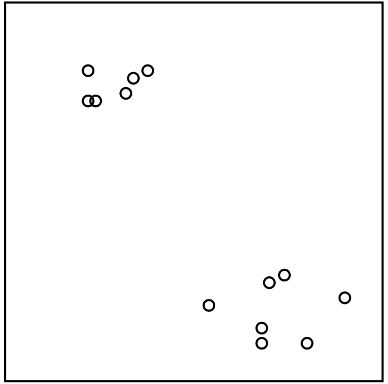
Unsupervised learning

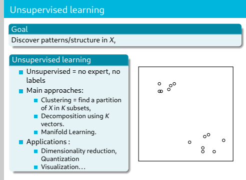
Goal

Discover patterns/structure in X,

Unsupervised learning

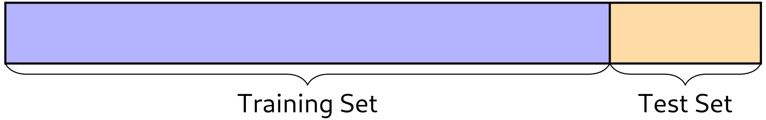
- Unsupervised = no expert, no labels
- Main approaches:
  - Clustering = find a partition of X in K subsets,
  - Decomposition using K vectors.
  - Manifold Learning.
- Applications :
  - Dimensionality reduction, Quantization
  - Visualization...




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...

Training and Test sets?

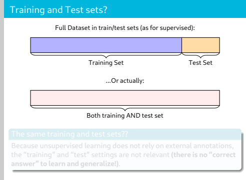
Full Dataset in train/test sets (as for supervised):



...Or actually:

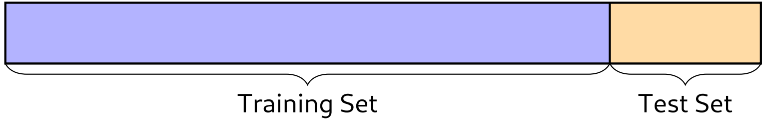
The same training and test sets??

Because unsupervised learning does not rely on external annotations, the “training” and “test” settings are not relevant **(there is no “correct answer” to learn and generalize!)**.



Training and Test sets?

Full Dataset in train/test sets (as for supervised):



...Or actually:

The same training and test sets??

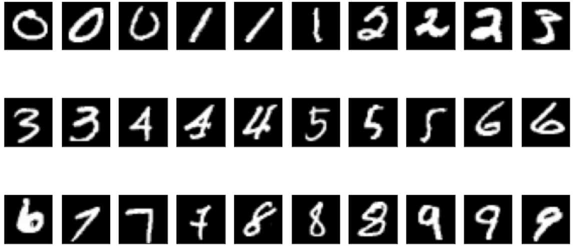
Because unsupervised learning does not rely on external annotations, the “training” and “test” settings are not relevant **(there is no “correct answer” to learn and generalize!)**.

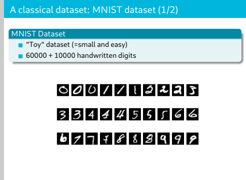
Training and Test sets?
Full Dataset in train/test sets (as for supervised):
Training Set
Test Set
...Or actually:
Both training AND test set
The same training and test sets??
Because unsupervised learning does not rely on external annotations, the “training” and “test” settings are not relevant **(there is no “correct answer” to learn and generalize!)**.

A classical dataset: MNIST dataset (1/2)

MNIST Dataset

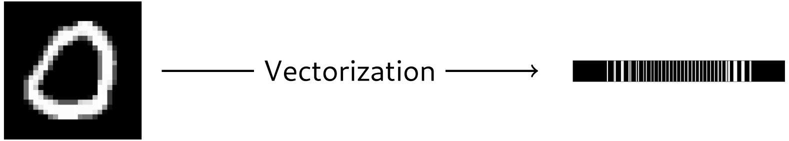
- "Toy" dataset (=small and easy)
- 60000 + 10000 handwritten digits





Let's look at an example that looks a little bit more like real data. The MNIST dataset is small dataset of handwritten digits. It used to be an important benchmark, but it is considered too easy today to be a serious machine learning benchmark, so that is why we say it is a "toy" dataset. MNIST is composed of 60000 examples of digits that are used for training, and 10000 that are used for test.

A classical dataset: MNIST dataset (2/2)

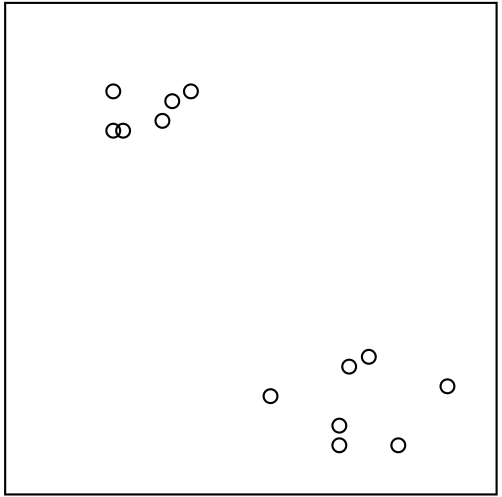


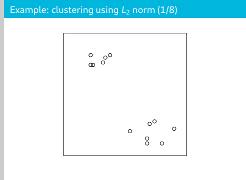
Hence, all images are interpreted as 1D vectors!

A classical dataset: MNIST dataset (2/2)

A classical dataset: MNIST dataset (2/2)
Vectorization
Hence, all images are interpreted as 1D vectors!

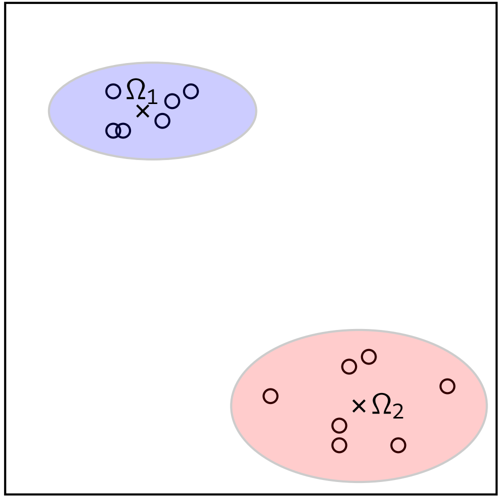
Example: clustering using L_2 norm (1/8)

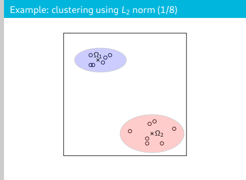




Here is a visual example. If we have the following set of points, then the following two centroids Ω_1 and Ω_2 would be reasonable candidates for a clustering with two clusters.

Example: clustering using L_2 norm (1/8)





Here is a visual example. If we have the following set of points, then the following two centroids Ω_1 and Ω_2 would be reasonable candidates for a clustering with two clusters.

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

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

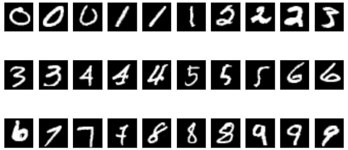
We denote $q : \mathbb{R}^d \rightarrow [1..K]$ a function that associates a vector $\mathbf{x}$ with the index of (one of) its closest centroid $\Omega_{q(\mathbf{x})}$. Formally:

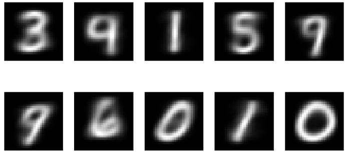
- $\forall k \in [1..K], \Omega_k \in \mathbb{R}^d$
- $\forall \mathbf{x} \in X, \forall j \in [1..K], \|\mathbf{x} - \Omega_{q(\mathbf{x})}\|_2 \leq \|\mathbf{x} - \Omega_j\|_2$
- Error $E(q) \triangleq \sum_{\mathbf{x} \in X} \|\mathbf{x} - \Omega_{q(\mathbf{x})}\|_2$
- $X = \bigcup_k \underbrace{\{\mathbf{x}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

Here, we provide a formal definition of clustering using centroids. Note that there are other ways to define clustering, using regions, using density of spaces, using probabilities, etc...
The second point is the way to define the closest centroid.
The important point to note here is the definition of the error, which can be defined as the sum of all distances between points and their closest cluster centroid.

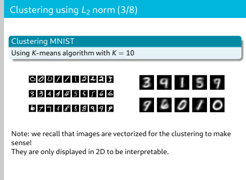
Clustering MNIST

Using K-means algorithm with $K = 10$





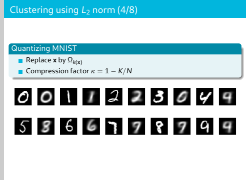
Note: we recall that images are vectorized for the clustering to make sense!
They are only displayed in 2D to be interpretable.



Let's look at an example that looks a little bit more like real data. The MNIST dataset is small dataset of handwritten digits. It used to be an important benchmark, but it is considered too easy today to be a serious machine learning benchmark, so that is why we say it is a "toy" dataset.
MNIST is composed of 60000 examples of digits that are used for training, and 10000 that are used for test.
We can do a simple clustering test on this dataset, by using the K-Means algorithm.
Briefly, the K-means algorithm iterates between (a) assigning each point to a cluster by considering the distance to centroids, and (b) calculating the centroids for the next iteration by computing the average in each cluster. Centroid clusters can be initilized randomly.
The K-means algorithm stops when a certain criterion is met (number of iterations, or difference between iterations is small enough).
See here `https://upload.wikimedia.org/wikipedia/commons/f/fb/K-means.png` (picture is nice) or `https://en.wikipedia.org/wiki/K-means_clustering`
Maybe a very quick explanation of Kmeans on the board is good if the time enables it.
The bottom left figure represent original examples of MNIST. The bottom right figure shows the obtained cluster centroids with Kmeans. We can comment that some of the clusters seem to capture one digit (6, 1, 2, 0), but that other digits can correspond to several clusters (8, 4, 3).
The next figure will illustrate this more precisely.

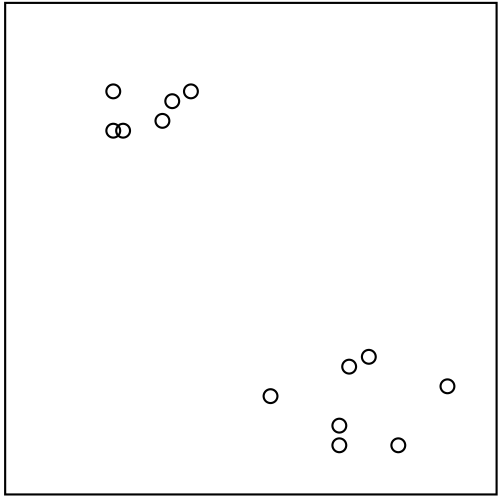
Quantizing MNIST

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

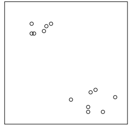


We have chosen here a random example of each digit, and we show the closest cluster centroid. We see that there are issues with 3, 4, 5, 7 and 8, even though we have tried to find 10 clusters.
In the top part of the slide, we also explain that we can actually use Clustering for compression; we just have to store the centroids, and the cluster label.

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

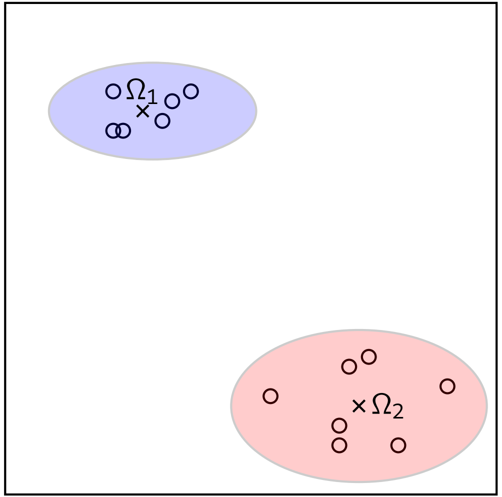


Clustering using L_2 norm (5/8): Choosing K

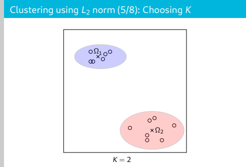


Changing the number of centroids changes the clustering... And the signification of clusters.

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

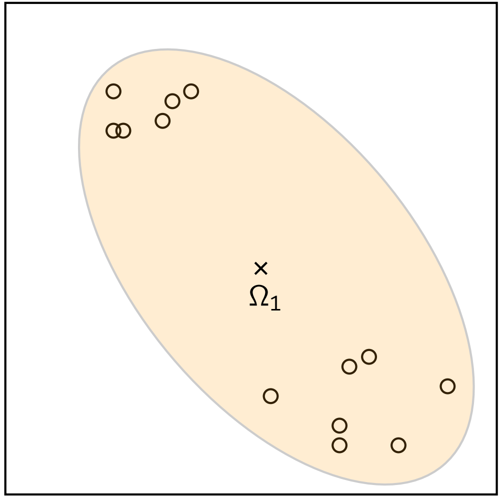


$K = 2$

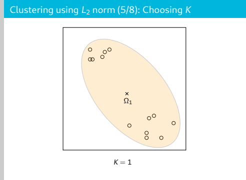


Changing the number of centroids changes the clustering... And the signification of clusters.

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

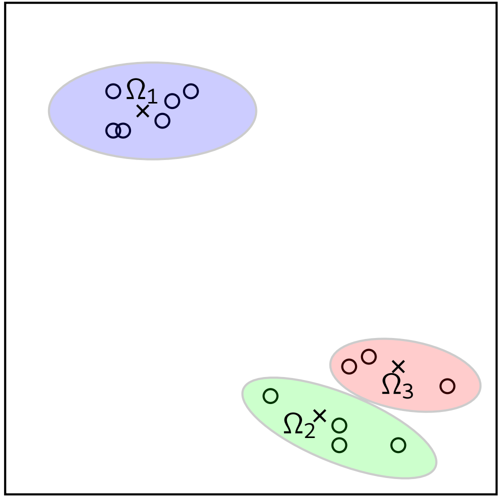


$K = 1$

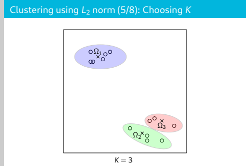


Changing the number of centroids changes the clustering... And the signification of clusters.

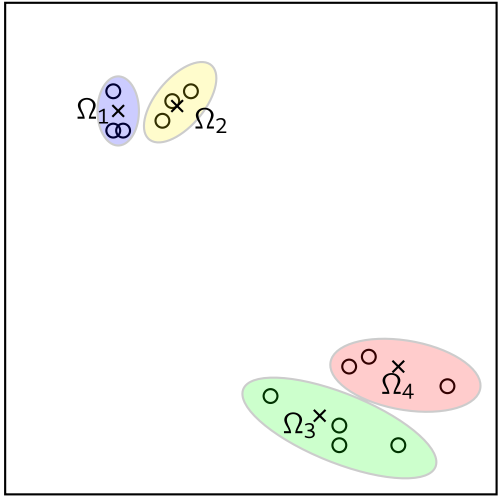
Clustering using L_2 norm (5/8): Choosing K



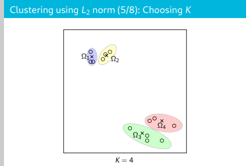
$K = 3$



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

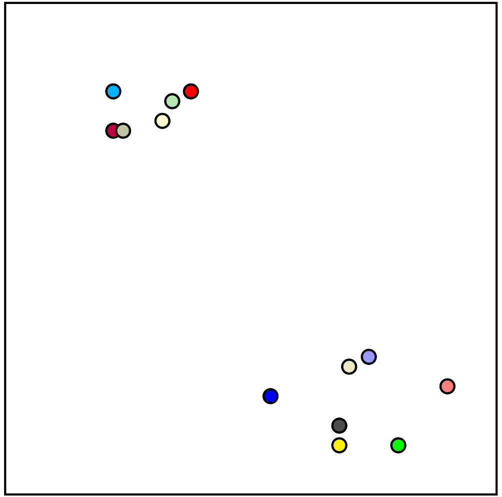


$K = 4$

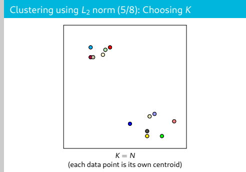


Changing the number of centroids changes the clustering... And the signification of clusters.

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



$K = N$
(each data point is its own centroid)

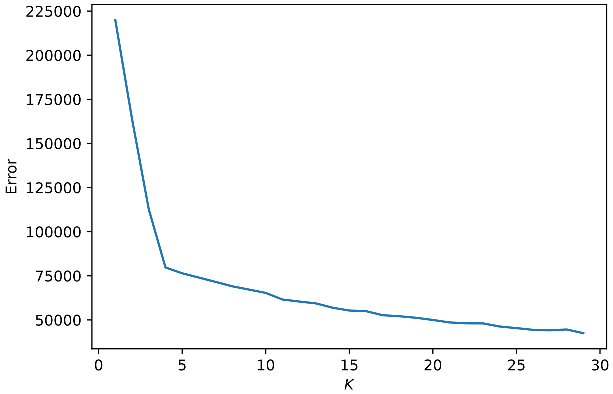


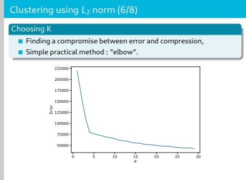
Changing the number of centroids changes the clustering... And the signification of clusters.

Clustering using L_2 norm (6/8)

Choosing K

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





It is important to say that this is the ideal case! Here, we clearly see a value of K after which it is not necessary to add more clusters.

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 (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.

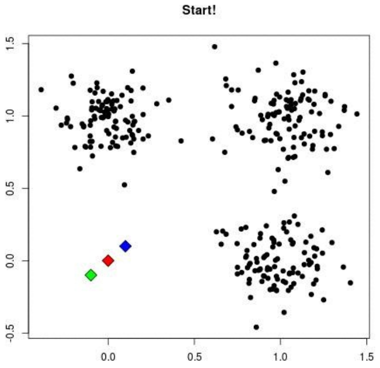
About the properties :
On the left side, if we take a cluster for each point in the space (N cluster centroids), then obviously the error is 0.
On the right side, if we take only one cluster, then the best cluster that can be chosen is the average of all points, in which case the error is exactly the variance across X.

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 .

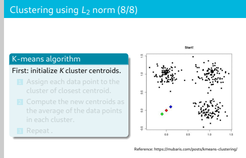


Reference: https://mubaris.com/posts/kmeans-clustering/

Course 3: Unsupervised Learning

2025-10-16

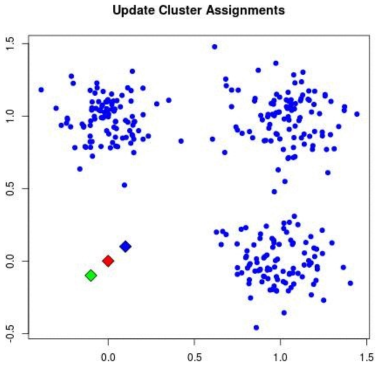
Clustering using L_2 norm (8/8)



K-means algorithm

First: initialize K cluster centroids.

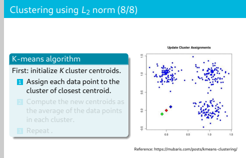
1. Assign each data point to the cluster of closest centroid.
2. Compute the new centroids as the average of the data points in each cluster.
3. Repeat .



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

2025-10-16 Course 3: Unsupervised Learning

└─Clustering using L_2 norm (8/8)

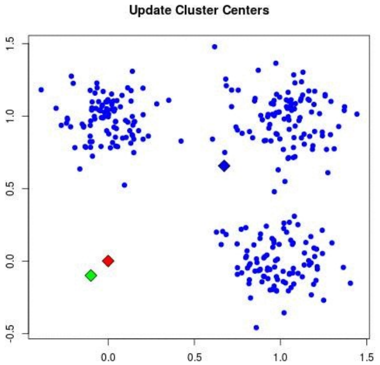


Clustering using L_2 norm (8/8)

K-means algorithm

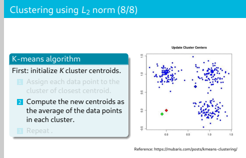
First: initialize K cluster centroids.

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)

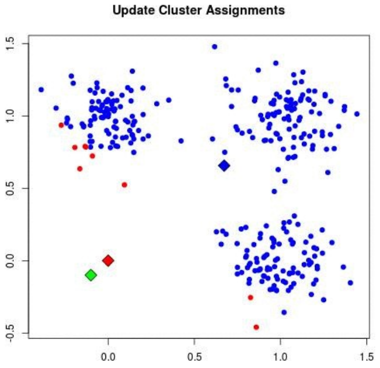


Clustering using L_2 norm (8/8)

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 .

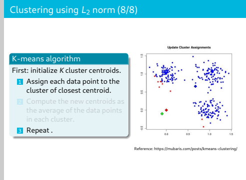


Reference: https://mubaris.com/posts/kmeans-clustering/

2025-10-16

Course 3: Unsupervised Learning

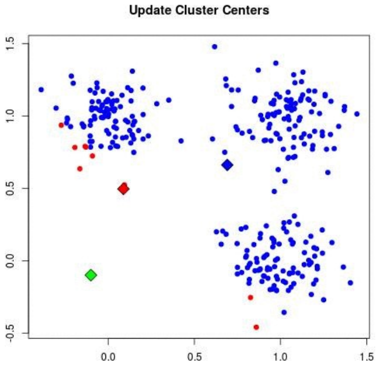
└─Clustering using L_2 norm (8/8)



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 .

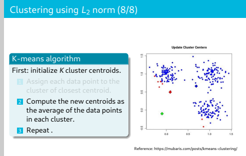


Reference: https://mubaris.com/posts/kmeans-clustering/

2025-10-16

Course 3: Unsupervised Learning

└─Clustering using L_2 norm (8/8)

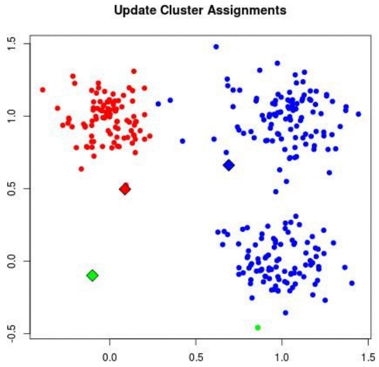


Clustering using L_2 norm (8/8)

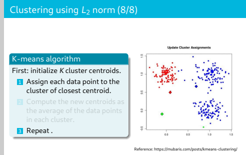
K-means algorithm

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/

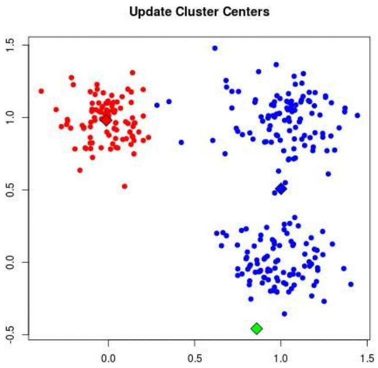


Clustering using L_2 norm (8/8)

K-means algorithm

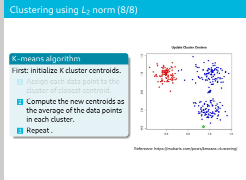
First: initialize K cluster centroids.

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/

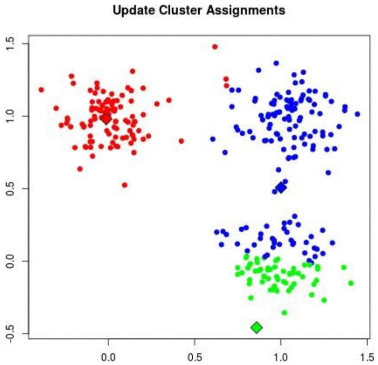
└─Clustering using L_2 norm (8/8)



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 .

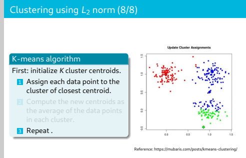


Reference: https://mubaris.com/posts/kmeans-clustering/

2025-10-16

Course 3: Unsupervised Learning

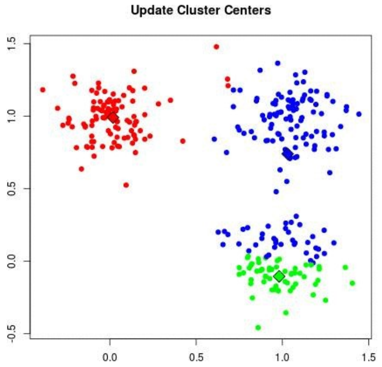
└─Clustering using L_2 norm (8/8)



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 .



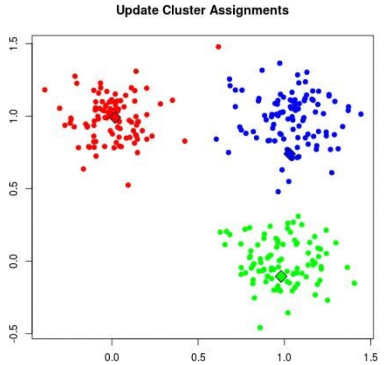
Reference: https://mubaris.com/posts/kmeans-clustering/

2025-10-16 Course 3: Unsupervised Learning
└ 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 .

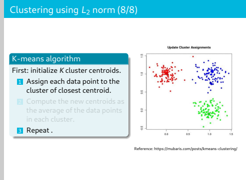


Reference: https://mubaris.com/posts/kmeans-clustering/

2025-10-16

Course 3: Unsupervised Learning

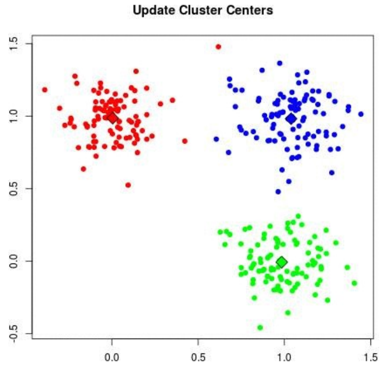
└─Clustering using L_2 norm (8/8)



K-means algorithm

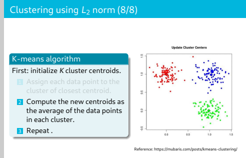
First: initialize K cluster centroids.

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)

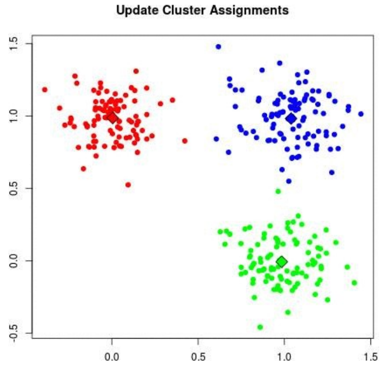


Clustering using L_2 norm (8/8)

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 until convergence.

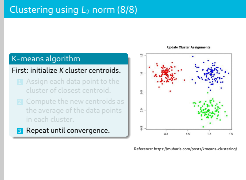


Reference: https://mubaris.com/posts/kmeans-clustering/

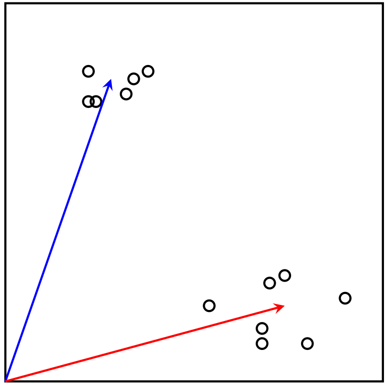
2025-10-16

Course 3: Unsupervised Learning

└ Clustering using L_2 norm (8/8)



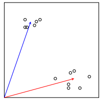
Decomposition



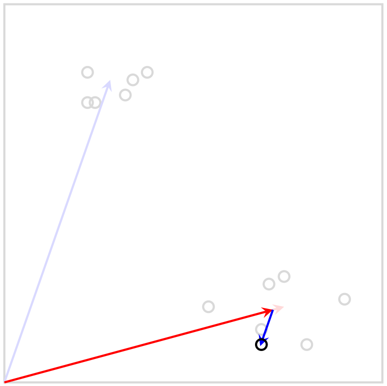
2025-10-16 Course 3: Unsupervised Learning

└─Decomposition

Decomposition

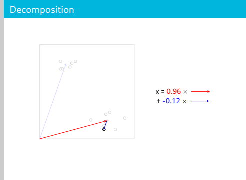


Decomposition

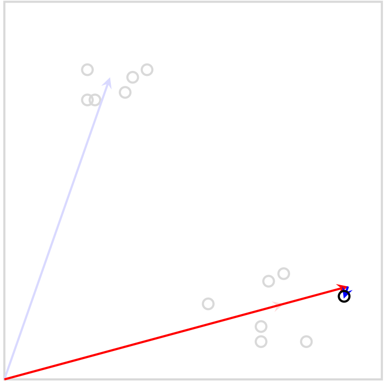


x = 0.96 ×

\+ -0.12 ×

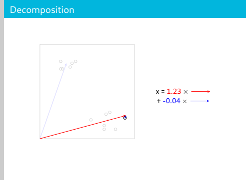


Decomposition

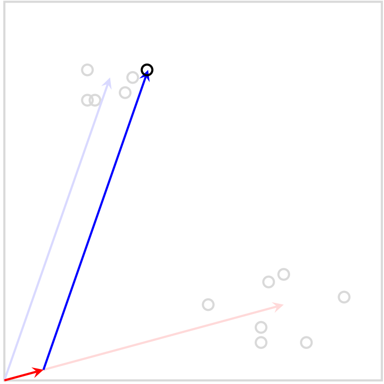


x = 1.23 × $\longrightarrow$
+ -0.04 × $\longrightarrow$

2025-10-16 Course 3: Unsupervised Learning
└Decomposition

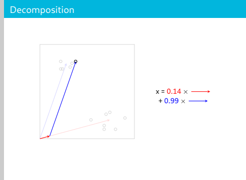


Decomposition

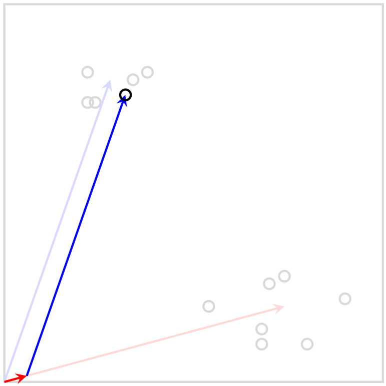


$x = \color{red}{0.14} \times \color{red}{\longrightarrow}$

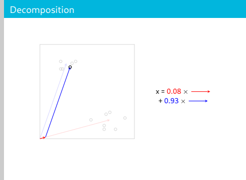
$+ \color{blue}{0.99} \times \color{blue}{\longrightarrow}$



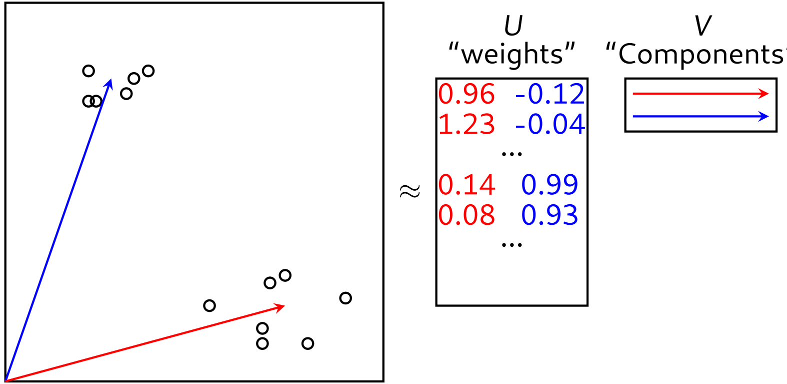
Decomposition

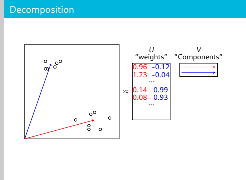


x = 0.08 × ⟶
+ 0.93 × ⟶



Decomposition

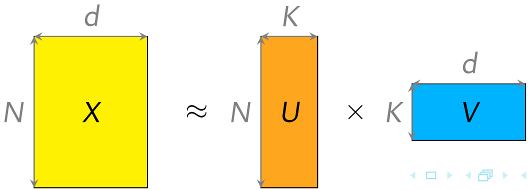


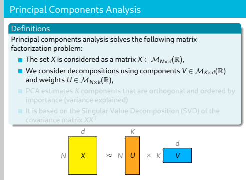


Definitions

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 $XX^\top$

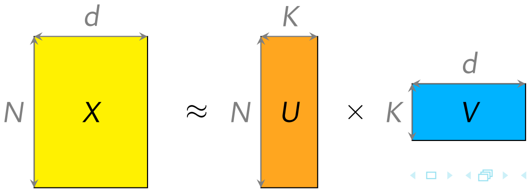


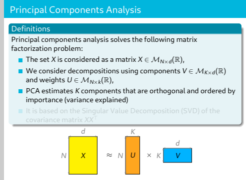


Definitions

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 $XX^\top$



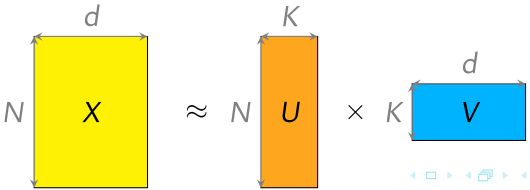


Principal Components Analysis

Definitions

Principal components analysis solves the following matrix factorization problem:

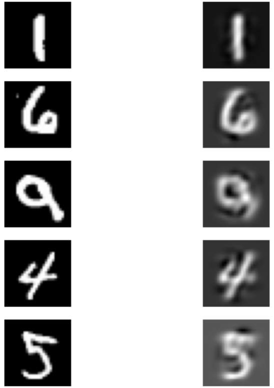
- The set X is considered as a matrix $X \in \mathcal{M}_{N\times d}(\mathbb{R})$,
- We consider decompositions using components $V \in \mathcal{M}_{K\times d}(\mathbb{R})$ and weights $U \in \mathcal{M}_{N\times k}(\mathbb{R})$,
- PCA estimates K components that are orthogonal and ordered by importance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



Principal Components Analysis
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 $K = 32$

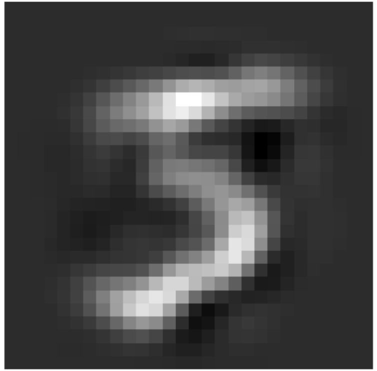


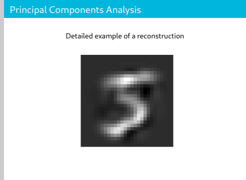
Recall that each image is vectorized, hence each of these images correspond to a row in V.

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.

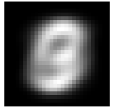
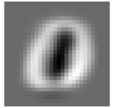
Detailed example of a reconstruction

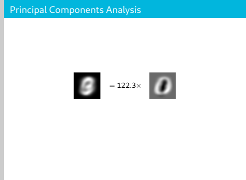




In this slide we show the result of reconstructing the original vectors using the learnt components. In the first slide, we only show the results of reconstruction. In the following slides, we unroll the decomposition of a particular example, by adding the combination of successive components.

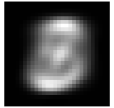
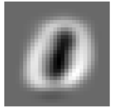
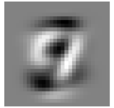
Principal Components Analysis

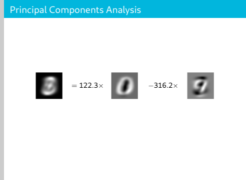
 $= 122.3 \times$ 



In this slide we show the result of reconstructing the original vectors using the learnt components. In the first slide, we only show the results of reconstruction. In the following slides, we unroll the decomposition of a particular example, by adding the combination of successive components.

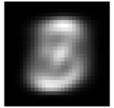
Principal Components Analysis

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

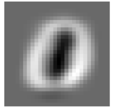


In this slide we show the result of reconstructing the original vectors using the learnt components. In the first slide, we only show the results of reconstruction. In the following slides, we unroll the decomposition of a particular example, by adding the combination of successive components.

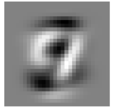
Principal Components Analysis



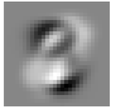
$= 122.3\times$

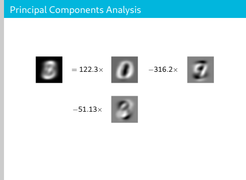


$-316.2\times$



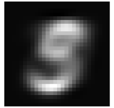
$-51.13\times$



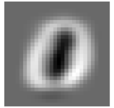


In this slide we show the result of reconstructing the original vectors using the learnt components. In the first slide, we only show the results of reconstruction. In the following slides, we unroll the decomposition of a particular example, by adding the combination of successive components.

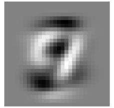
Principal Components Analysis



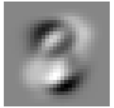
$= 122.3\times$



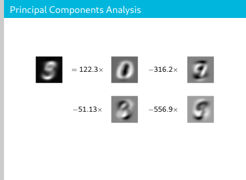
$-316.2\times$



$-51.13\times$

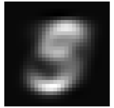
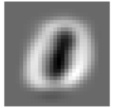
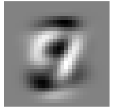


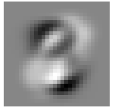
$-556.9\times$

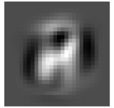


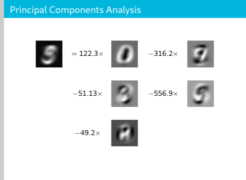
In this slide we show the result of reconstructing the original vectors using the learnt components. In the first slide, we only show the results of reconstruction. In the following slides, we unroll the decomposition of a particular example, by adding the combination of successive components.

Principal Components Analysis

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

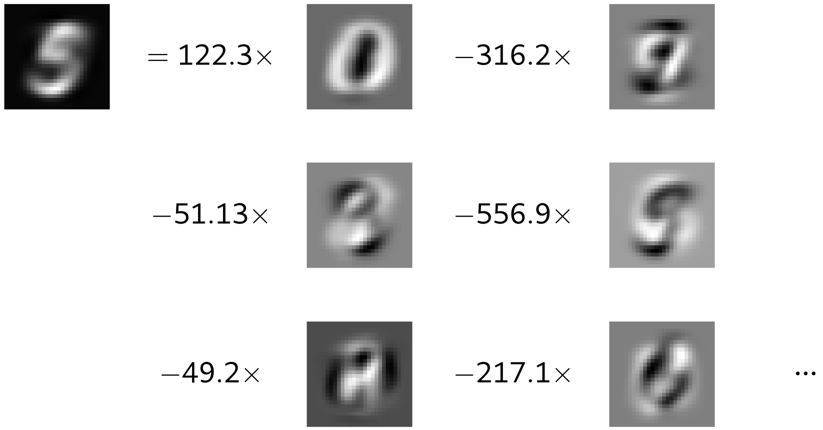
$-51.13\times$  $-556.9\times$ 

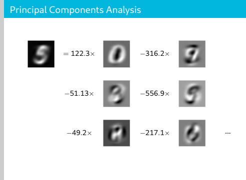
$-49.2\times$ 



In this slide we show the result of reconstructing the original vectors using the learnt components. In the first slide, we only show the results of reconstruction. In the following slides, we unroll the decomposition of a particular example, by adding the combination of successive components.

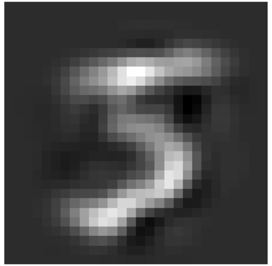
Principal Components Analysis





In this slide we show the result of reconstructing the original vectors using the learnt components. In the first slide, we only show the results of reconstruction. In the following slides, we unroll the decomposition of a particular example, by adding the combination of successive components.

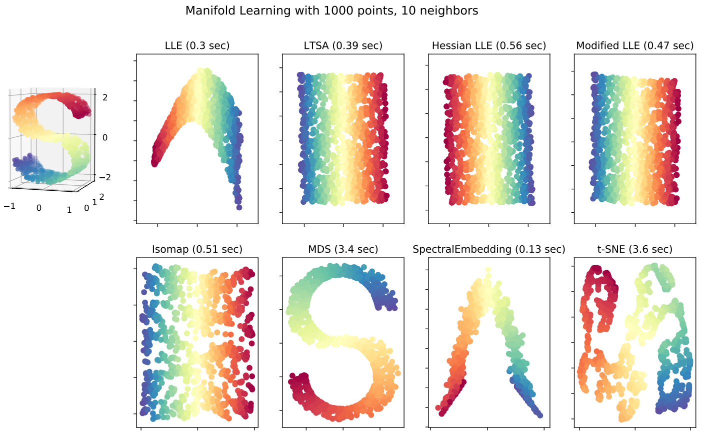
Reconstruction with all 32 components:



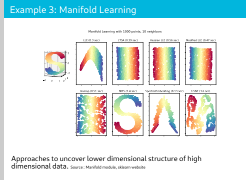
Principal Components Analysis
Reconstruction with all 32 components:

In this slide we show the result of reconstructing the original vectors using the learnt components. In the first slide, we only show the results of reconstruction. In the following slides, we unroll the decomposition of a particular example, by adding the combination of successive components.

Example 3: Manifold Learning



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



Tell them here that we don't have time to investigate in detail how these different methods work. The important thing is to explain the range of methods that can uncover the lower dimensional topology, in an unsupervised way.

Re-explain the original data (the swiss roll in the top right corner) and explain that there are methods that use different metrics (potentially non linear ones) that try to project in lower d.

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 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.

Don't hesitate to state that this lab is not easy, and that we value exploration and justification of the tests over results.

Working with features

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

Feature selection

Objective : remove features

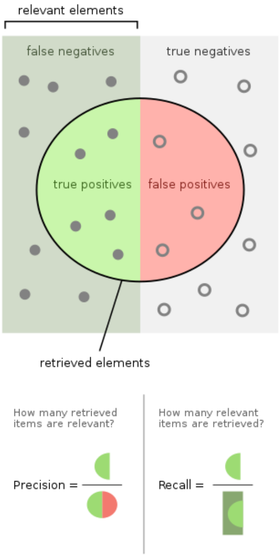
- Remove features with low variance
- Select features according to their explained variance towards labels (e.g. SelectKBest)
- See more `https://scikit-learn.org/stable/modules/feature_selection.html`

Helps to adress the dimensionality curse.

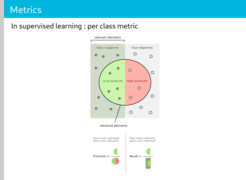
Working with features
N.b. : valid in unsupervised and supervised settings.
Feature selection
Objective : remove features
Remove features with low variance
Select features according to their explained variance towards labels (e.g. SelectKBest)
See more https://scikit-learn.org/stable/modules/feature_selection.html
Helps to adress the dimensionality curse.

Don't hesitate to state that this lab is not easy, and that we value exploration and justification of the tests over results.

In supervised learning : per class metric



2025-10-16 Course 3: Unsupervised Learning
Metrics



Metrics

Clustering Metrics :

- Error defined slide 10 : 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 10 : 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

See more on sklearn website and in the lab session

Metrics

Clustering Metrics :

- Error defined slide 10 : 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 10 : 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 10 : 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 10 : 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 10 : 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 10 : 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).

Project 2 (P2)

You will choose one unsupervised learning method from the available options (see Lab 3). You will present

- A brief description of the theory behind the method,
- Basic tests on this technique for your modality.

During Session 4, even binome numbers will have 7 minutes to present.

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).
Project 2 (P2)
You will choose one unsupervised learning method from the available options (see Lab 3). You will present
- A brief description of the theory behind the method,
- Basic tests on this technique for your modality.
During Session 4, even binome numbers will have 7 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

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