W9. Unsupervised Learning - Clustering & PCA

Guang Cheng

University of California, Los Angeles

guangcheng@ucla.edu

Week 9

Motivation

- Picture this – you are working on a large-scale data science project. What happens when the given data set has too many variables?

Motivation

- Picture this – you are working on a large-scale data science project. What happens when the given data set has too many variables?
- There are a few possible situations that you might come across. For instance, you find that most of the variables are correlated on analysis, and you become indecisive about what to do; hence you lose patience and decide to run a model on the whole data (curse of dimensionality!).

Motivation

- Picture this – you are working on a large-scale data science project. What happens when the given data set has too many variables?
- There are a few possible situations that you might come across. For instance, you find that most of the variables are correlated on analysis, and you become indecisive about what to do; hence you lose patience and decide to run a model on the whole data (curse of dimensionality!).
- This returns poor accuracy, and you feel terrible and start thinking of some strategic method to find a few important variables.

Motivation

- Picture this – you are working on a large-scale data science project. What happens when the given data set has too many variables?
- There are a few possible situations that you might come across. For instance, you find that most of the variables are correlated on analysis, and you become indecisive about what to do; hence you lose patience and decide to run a model on the whole data (curse of dimensionality!).
- This returns poor accuracy, and you feel terrible and start thinking of some strategic method to find a few important variables.
- That's where Principal Component Analysis (PCA) is used.

What is Principal Component Analysis (PCA)?

- Principal component analysis, or PCA, is a statistical procedure that allows you to summarize the information content in large data tables by means of a smaller set of "summary indices" that can be more easily visualized and analyzed.

What is Principal Component Analysis (PCA)?

- Principal component analysis, or PCA, is a statistical procedure that allows you to summarize the information content in large data tables by means of a smaller set of "summary indices" that can be more easily visualized and analyzed.
- The underlying data can be measurements describing properties of production samples, chemical compounds or reactions, process time points of a continuous process, batches from a batch process, biological individuals or trials of a DOE-protocol, for example.

What is Principal Component Analysis (PCA)?

- Principal component analysis, or PCA, is a statistical procedure that allows you to summarize the information content in large data tables by means of a smaller set of "summary indices" that can be more easily visualized and analyzed.
- The underlying data can be measurements describing properties of production samples, chemical compounds or reactions, process time points of a continuous process, batches from a batch process, biological individuals or trials of a DOE-protocol, for example.
- PCA is often used in the preliminary data analytics, before running any machine learning tasks.

How PCA works

- Consider a matrix X with N rows (aka "observations") and K columns (aka "variables").

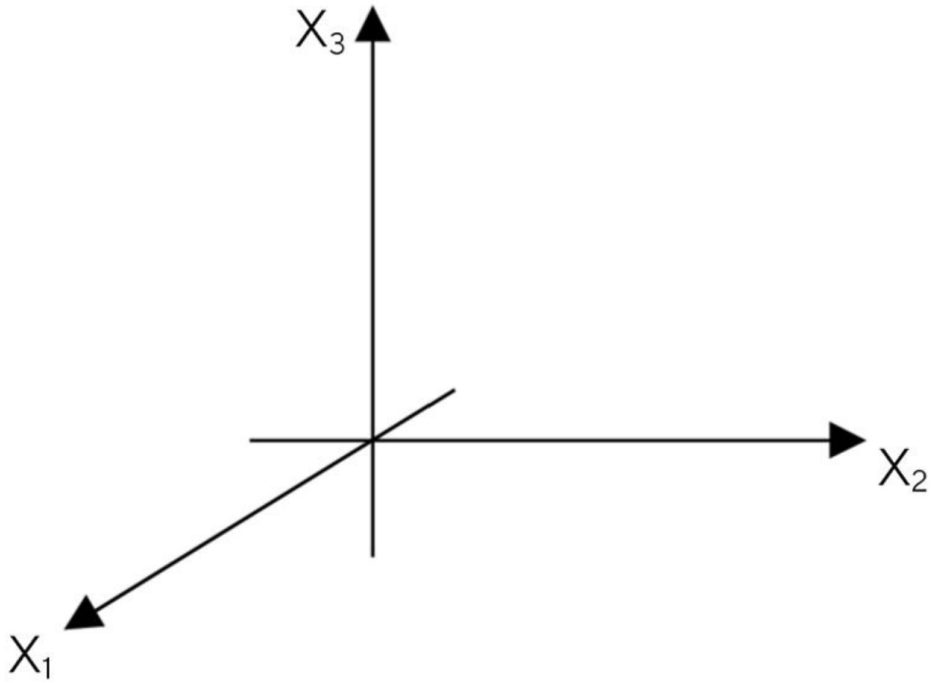
How PCA works

- Consider a matrix X with N rows (aka "observations") and K columns (aka "variables").
- For this matrix, we construct a variable space with as many dimensions as there are variables (see figure in the next page).

How PCA works

- Consider a matrix X with N rows (aka "observations") and K columns (aka "variables").
- For this matrix, we construct a variable space with as many dimensions as there are variables (see figure in the next page).
- Each variable represents one coordinate axis. For each variable, the length has been standardized according to a scaling criterion, normally by scaling to unit variance.

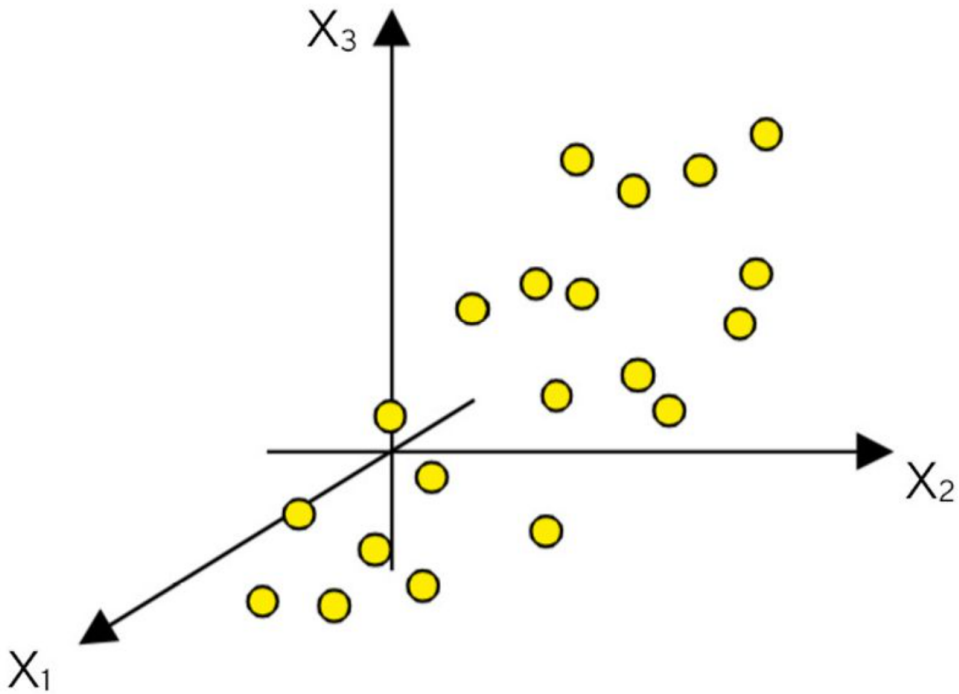
How PCA works



- A K-dimensional variable space. For simplicity, only three variables axes are displayed. The "length" of each coordinate axis has been standardized according to a specific criterion, usually unit variance scaling.

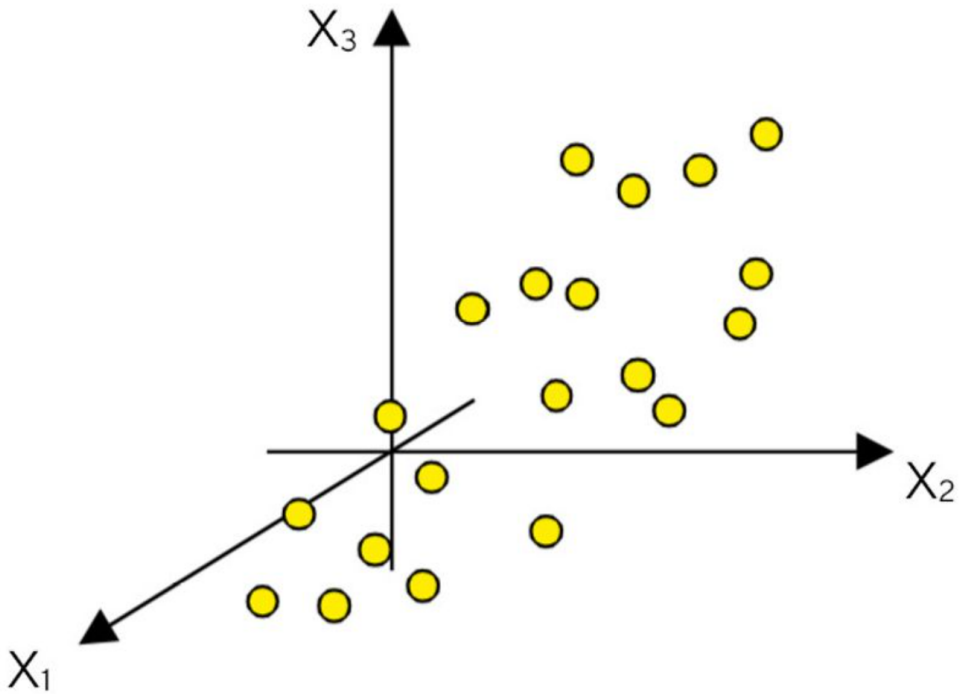
How PCA works

- In the next step, each observation (row) of the X-matrix is placed in the K-dimensional variable space. Consequently, the rows in the data table form a swarm of points in this space.



How PCA works

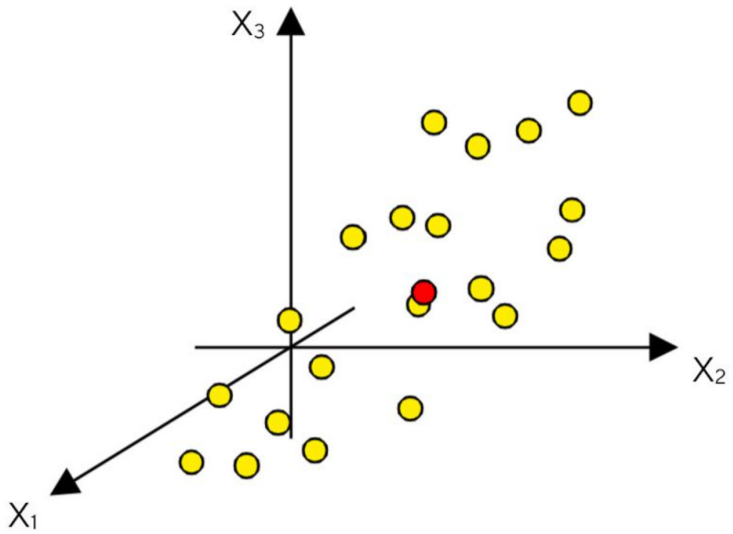
- In the next step, each observation (row) of the X-matrix is placed in the K-dimensional variable space. Consequently, the rows in the data table form a swarm of points in this space.



- The observations (rows) in the data matrix X can be understood as a swarm of points in the variable space (K-space).

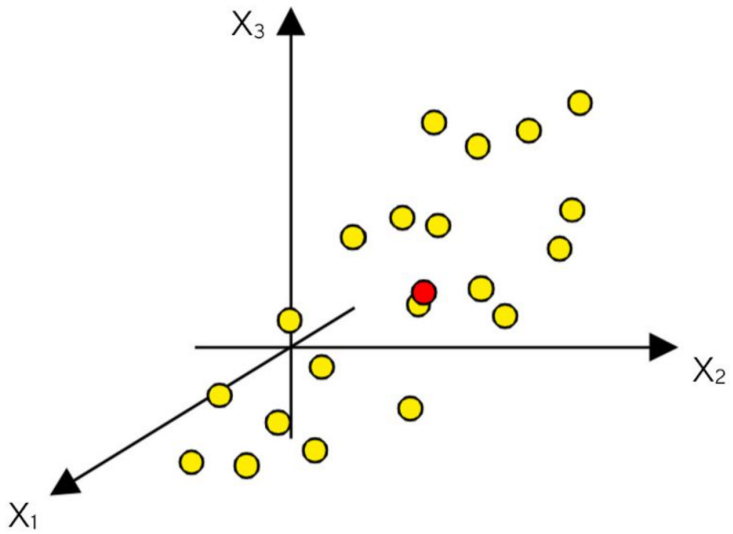
How PCA works

- Next, mean-centering involves the subtraction of the variable averages from the data. The vector of averages corresponds to a point in the K-space.



How PCA works

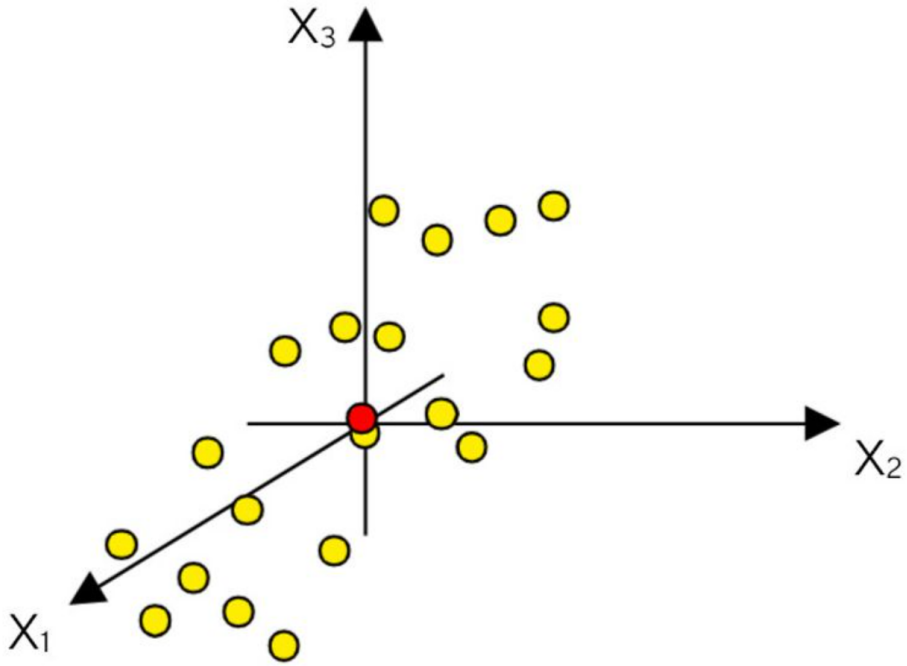
- Next, mean-centering involves the subtraction of the variable averages from the data. The vector of averages corresponds to a point in the K-space.



- In the mean-centering procedure, you first compute the variable averages. This vector of averages is interpretable as a point (here in red) in space. The point is situated in the middle of the point swarm (at the center of gravity).

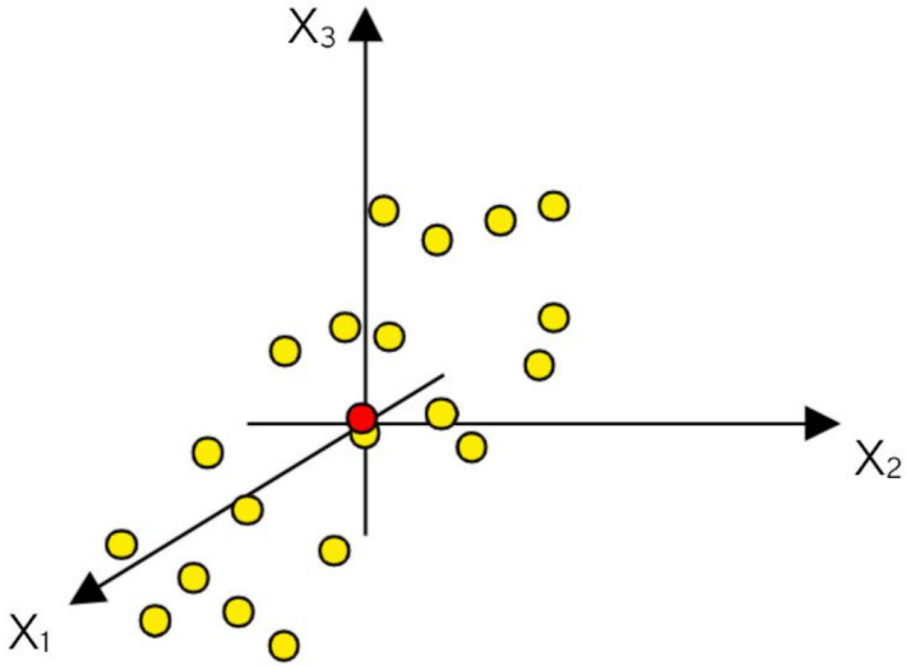
How PCA works

- The subtraction of the averages from the data corresponds to a re-positioning of the coordinate system, such that the average point now is the origin.



How PCA works

- The subtraction of the averages from the data corresponds to a re-positioning of the coordinate system, such that the average point now is the origin.



- The mean-centering procedure corresponds to moving the origin of the coordinate system to coincide with the average point (here in red).

The first principal component

- After mean-centering and scaling to unit variance, the data set is ready for computation of the first summary index, the first principal component (PC1).

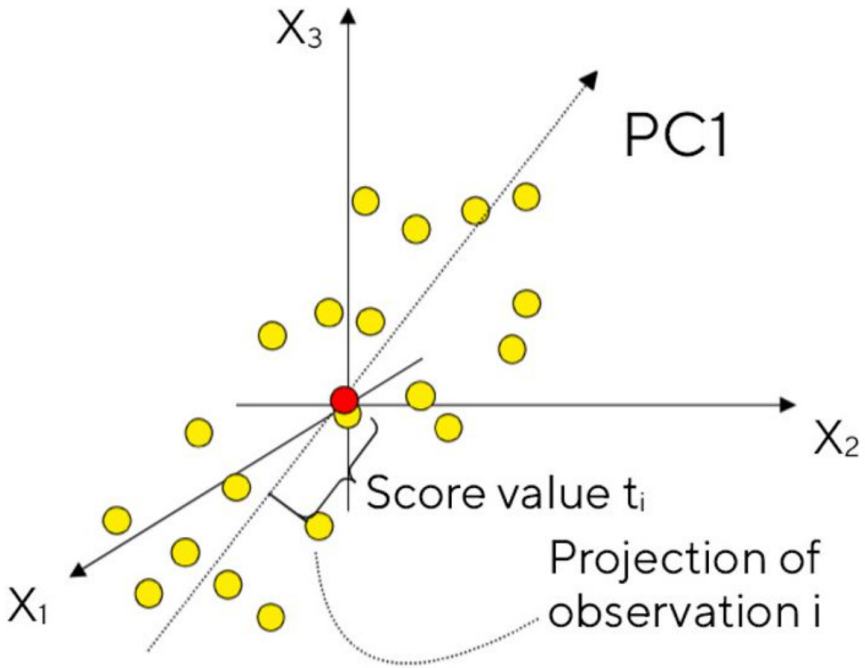
The first principal component

- After mean-centering and scaling to unit variance, the data set is ready for computation of the first summary index, the first principal component (PC1).
- This component is the line in the K-dimensional variable space that best approximates the data in the least squares sense. This line goes through the average point.

The first principal component

- After mean-centering and scaling to unit variance, the data set is ready for computation of the first summary index, the first principal component (PC1).
- This component is the line in the K-dimensional variable space that best approximates the data in the least squares sense. This line goes through the average point.
- Each observation (yellow dot) may now be projected onto this line in order to get a coordinate value along the PC-line. This new coordinate value is also known as the score.

The first principal component



- The first principal component (PC1) is the line that best accounts for the shape of the point swarm. It represents the *maximum variance direction* in the data.

The second principal component

- Usually, one summary index or principal component is insufficient to model the systematic variation of a data set.

The second principal component

- Usually, one summary index or principal component is insufficient to model the systematic variation of a data set.
- Thus, a second summary index – a second principal component (PC2) – is calculated.

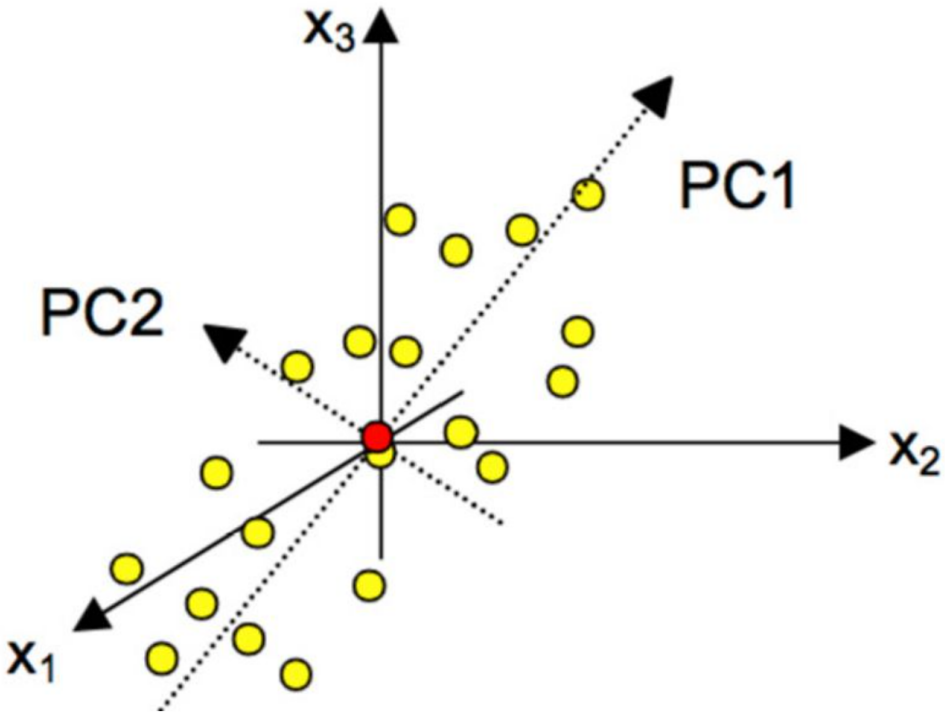
The second principal component

- Usually, one summary index or principal component is insufficient to model the systematic variation of a data set.
- Thus, a second summary index – a second principal component (PC2) – is calculated.
- The second PC is also represented by a line in the K-dimensional variable space, which is *orthogonal* to the first PC.

The second principal component

- Usually, one summary index or principal component is insufficient to model the systematic variation of a data set.
- Thus, a second summary index – a second principal component (PC2) – is calculated.
- The second PC is also represented by a line in the K-dimensional variable space, which is *orthogonal* to the first PC.
- This line also passes through the average point.

The second principal component



- The second principal component (PC2) is oriented such that it reflects the second largest source of variation in the data while being *orthogonal* to the first PC. PC2 also passes through the average point.

How to calculate the PC1 and PC2 ?

- **Standardize the data**: Ensure that each feature has a mean of 0 and a standard deviation of 1. This is important for PCA because it is sensitive to the scales of the variables.

How to calculate the PC1 and PC2 ?

- **Standardize the data**: Ensure that each feature has a mean of 0 and a standard deviation of 1. This is important for PCA because it is sensitive to the scales of the variables.
- **Calculate the covariance matrix**: The covariance matrix is a square matrix that captures the covariance between each pair of features in the dataset. The covariance matrix (with dimension $k \times k$) is

$$\widehat{\Sigma} = \frac{1}{n-1} X^T X,$$

where X is the standardized data matrix (dimnesion $n \times k$) with n observations and k features.

How to calculate the PC1 and PC2 ?

- **Calculate the eigenvalues and eigenvectors of the covariance matrix**: The eigenvectors represent the directions of maximum variance in the dataset, and the eigenvalues represent the magnitude of the variance in those directions.

How to calculate the PC1 and PC2 ?

- **Calculate the eigenvalues and eigenvectors of the covariance matrix**: The eigenvectors represent the directions of maximum variance in the dataset, and the eigenvalues represent the magnitude of the variance in those directions.
- The eigenvectors $\{v_1, v_2, .., v_k\}$ and eigenvalues $\{\lambda_1, \lambda_2, ..\lambda_k\}$ are the set of vectors/values such that

$$\widehat{\Sigma} v_i = \lambda_i v_i.$$

How to calculate the PC1 and PC2 ?

- **Calculate the eigenvalues and eigenvectors of the covariance matrix**: The eigenvectors represent the directions of maximum variance in the dataset, and the eigenvalues represent the magnitude of the variance in those directions.
- The eigenvectors $\{v_1, v_2, .., v_k\}$ and eigenvalues $\{\lambda_1, \lambda_2, ..\lambda_k\}$ are the set of vectors/values such that

$$\widehat{\Sigma} v_i = \lambda_i v_i.$$

- The eigen-decomposition of $\widehat{\Sigma}$ can be written as

$$\widehat{\Sigma} = V \Lambda V^T,$$

where V is a matrix containing the eigen-vectors and Λ is a diagonal matrix with diagonal elements as eigenvalues.

How to calculate the PC1 and PC2 ?

- **Sort the eigenvectors by their corresponding eigenvalues in descending order**: The eigenvector associated with the largest eigenvalue is the first principal component, and the eigenvector associated with the second largest eigenvalue is the second principal component.

How to calculate the PC1 and PC2 ?

- **Sort the eigenvectors by their corresponding eigenvalues in descending order**: The eigenvector associated with the largest eigenvalue is the first principal component, and the eigenvector associated with the second largest eigenvalue is the second principal component.
- Then PC1 and PC2 can be represented as

$$PC1 = v_1$$

$$PC2 = v_2$$

where v_1 and v_2 are the eigenvectors corresponding to the largest and second largest eigenvalues, respectively.

Two principal components define a model plane

- When two principal components have been derived, they together define a place, a window into the K-dimensional variable space.

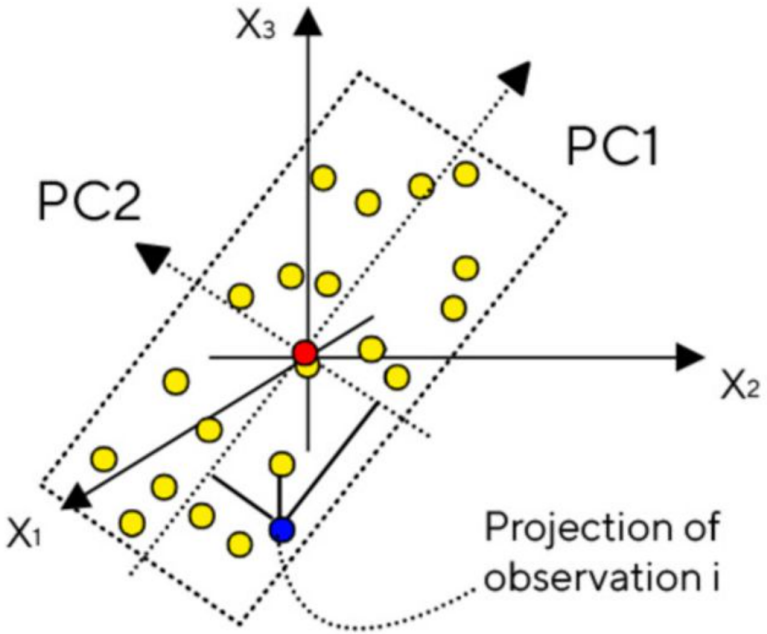
Two principal components define a model plane

- When two principal components have been derived, they together define a place, a window into the K-dimensional variable space.
- By projecting all the observations onto the low-dimensional sub-space and plotting the results, it is possible to visualize the structure of the investigated data set.

Two principal components define a model plane

- When two principal components have been derived, they together define a place, a window into the K-dimensional variable space.
- By projecting all the observations onto the low-dimensional sub-space and plotting the results, it is possible to visualize the structure of the investigated data set.
- The coordinate values of the observations on this plane are called scores, and hence the plotting of such a projected configuration is known as a score plot.

Two principal components define a model plane



- Two PCs form a plane. This plane is a window into the multidimensional space, which can be visualized graphically. Each observation may be projected onto this plane, giving a **score** for each.

What is the score ?

- The principal component scores of (standardized) X are obtained by multiplying X by the loadings (eigenvectors) of the covariance of X, say $\widehat{\Sigma}$.

What is the score ?

- The principal component scores of (standardized) X are obtained by multiplying X by the loadings (eigenvectors) of the covariance of X, say $\widehat{\Sigma}$.
- Recall that V is the matrix of eigenvectors (loadings) of $\widehat{\Sigma}$. We order the columns of V by their corresponding eigenvalues in descending order, then the scores T can be calculated as:

$$T_{n\times k} = XV$$

What is the score ?

- The principal component scores of (standardized) X are obtained by multiplying X by the loadings (eigenvectors) of the covariance of X, say $\widehat{\Sigma}$.
- Recall that V is the matrix of eigenvectors (loadings) of $\widehat{\Sigma}$. We order the columns of V by their corresponding eigenvalues in descending order, then the scores T can be calculated as:

$$T_{n\times k} = XV$$

- Here, the first column of T contains the scores for the first principal component (PC1), the second column contains the scores for the second principal component (PC2), and so on.

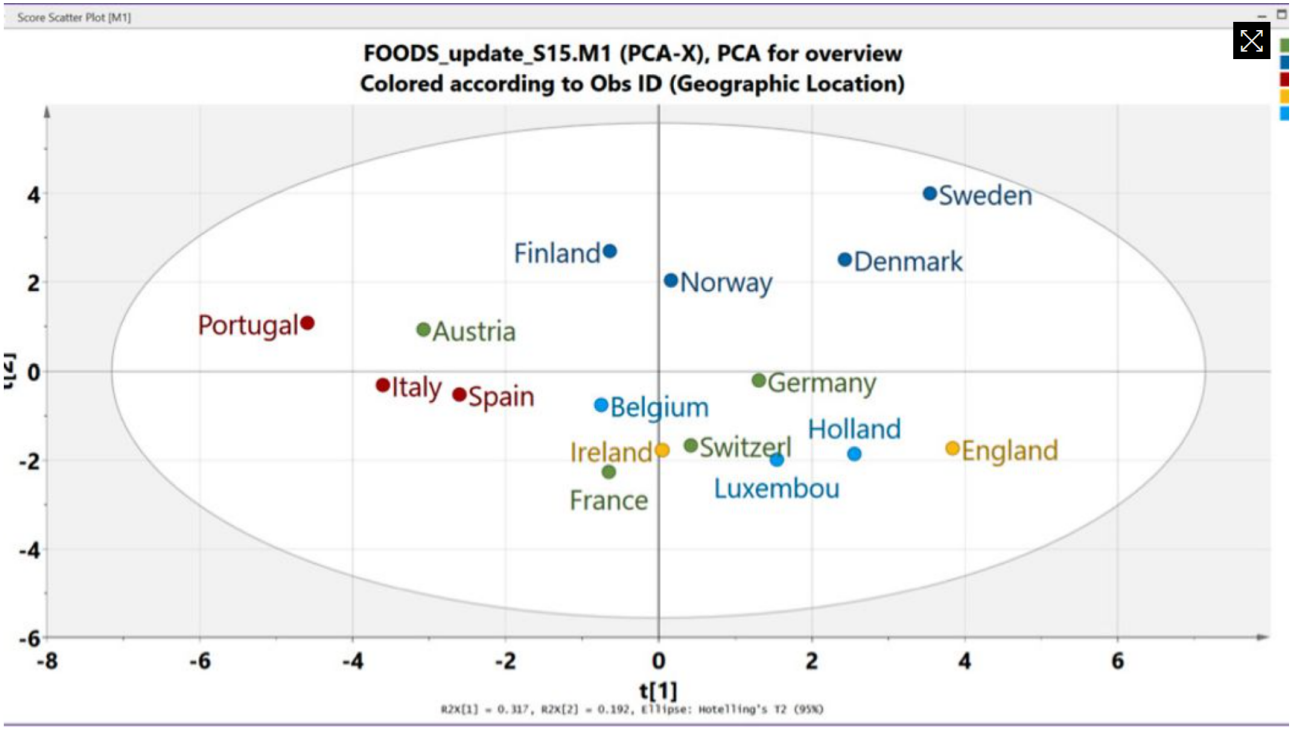
A real example to model a dataset

- Now, let's consider what this looks like using a data set of foods commonly consumed in different European countries. The figure in the next page displays the score plot of the first two principal components. These scores are called t_1 and t_2.

A real example to model a dataset

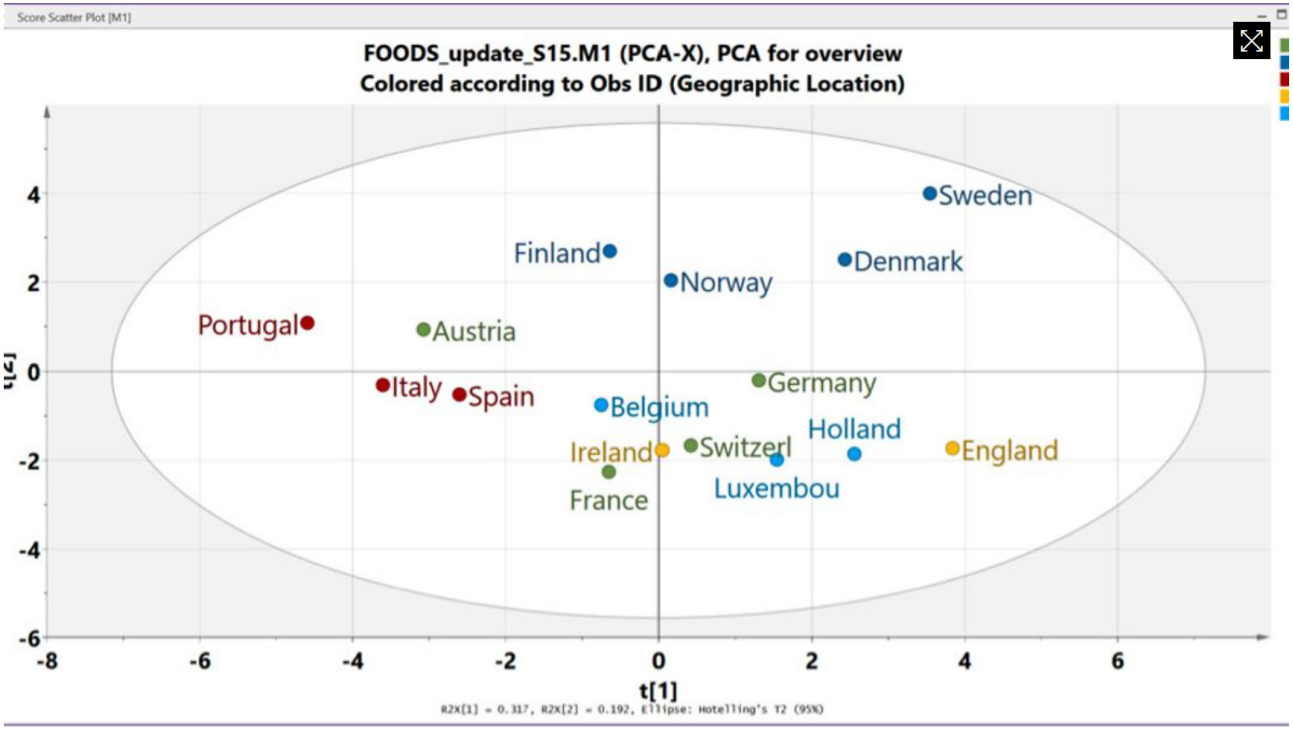
- Now, let's consider what this looks like using a data set of foods commonly consumed in different European countries. The figure in the next page displays the score plot of the first two principal components. These scores are called t_1 and t_2.
- The score plot is a map of 16 countries. Countries close to each other have similar food consumption profiles, whereas those far from each other are dissimilar.

A real example to model a dataset



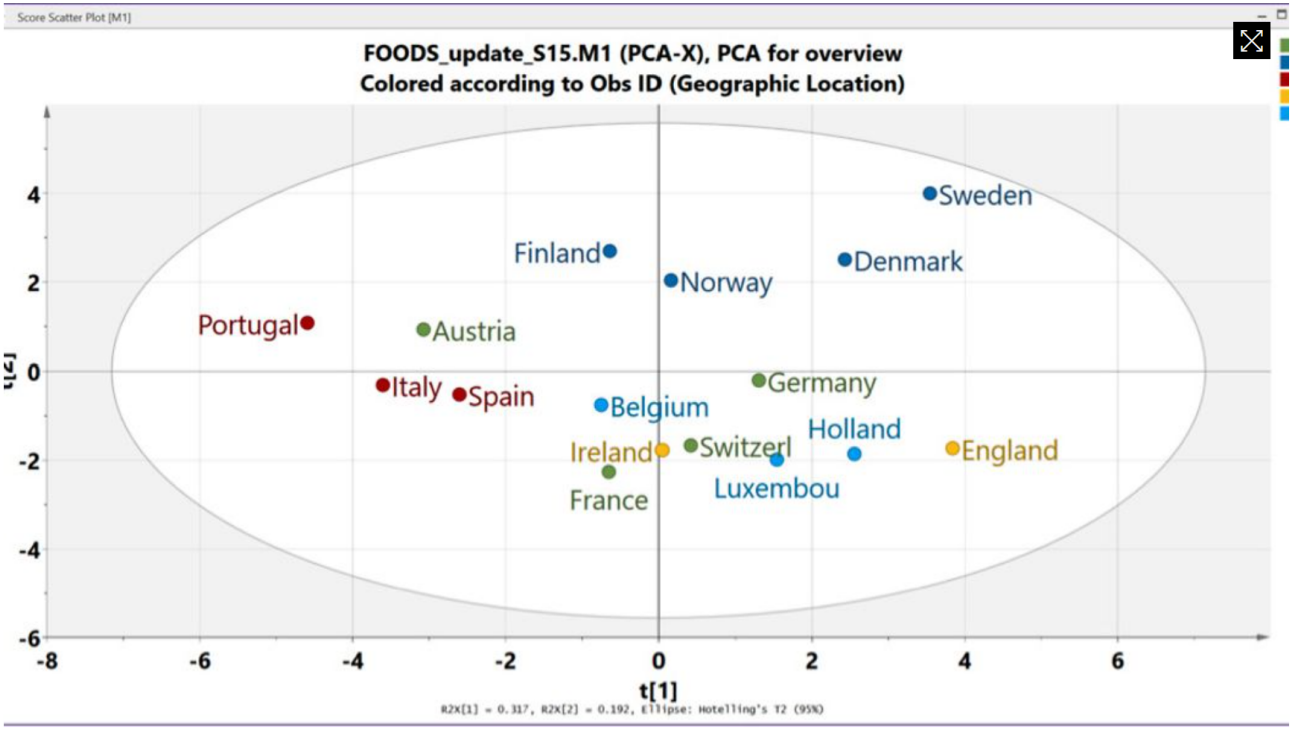
- The PCA score plot provides a map of how the countries relate to each other.

A real example to model a dataset



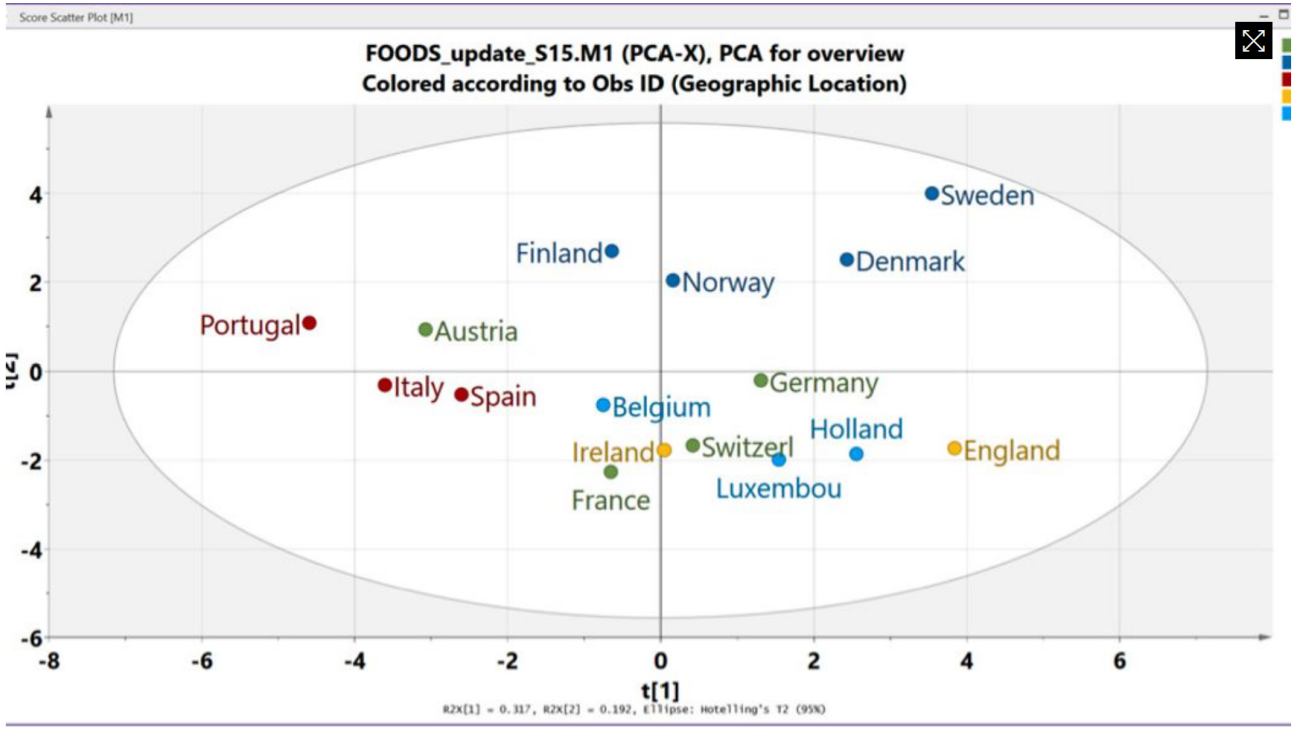
- The PCA score plot provides a map of how the countries relate to each other.
- The first component explains 32% of the variation, and the second component 19%.

A real example to model a dataset



- The Nordic countries (Finland, Norway, Denmark and Sweden) are located together in the upper right-hand corner, thus representing a group of nations with some similarity in food consumption.

A real example to model a dataset



- The Nordic countries (Finland, Norway, Denmark and Sweden) are located together in the upper right-hand corner, thus representing a group of nations with some similarity in food consumption.
- Belgium and Germany are close to the center (origin) of the plot, which indicates they have average properties.

Exercise: PCA

- Consider the following dataset with two variables X and Y:

X	Y
1	2
3	4
5	6

Exercise: PCA

- Consider the following dataset with two variables X and Y:

X	Y
1	2
3	4
5	6

- Perform PCA on this dataset by following these steps:

Exercise: PCA

- Consider the following dataset with two variables X and Y:

X	Y
1	2
3	4
5	6

- Perform PCA on this dataset by following these steps:
- **Center the data**: Subtract the mean of each variable from the corresponding values.

Exercise: PCA

- Consider the following dataset with two variables X and Y:

X	Y
1	2
3	4
5	6

- Perform PCA on this dataset by following these steps:
- **Center the data**: Subtract the mean of each variable from the corresponding values.
- **Calculate the covariance matrix** of the centered data.

Exercise: PCA

- Consider the following dataset with two variables X and Y:

X	Y
1	2
3	4
5	6

- Perform PCA on this dataset by following these steps:
- **Center the data**: Subtract the mean of each variable from the corresponding values.
- **Calculate the covariance matrix** of the centered data.
- **Find the eigenvalues and eigenvectors** of the covariance matrix.

Exercise: PCA

- Consider the following dataset with two variables X and Y:

X	Y
1	2
3	4
5	6

- Perform PCA on this dataset by following these steps:
- **Center the data**: Subtract the mean of each variable from the corresponding values.
- **Calculate the covariance matrix** of the centered data.
- **Find the eigenvalues and eigenvectors** of the covariance matrix.
- **Choose the principal component(s)**: Select the eigenvector(s) associated with the largest eigenvalue(s) as the principal component(s).

Exercise: PCA

- Consider the following dataset with two variables X and Y:

X	Y
1	2
3	4
5	6

- Perform PCA on this dataset by following these steps:
- **Center the data**: Subtract the mean of each variable from the corresponding values.
- **Calculate the covariance matrix** of the centered data.
- **Find the eigenvalues and eigenvectors** of the covariance matrix.
- **Choose the principal component(s)**: Select the eigenvector(s) associated with the largest eigenvalue(s) as the principal component(s).
- **Calculate the scores**: Project the centered data onto the principal component(s).

Exercise: PCA

- Centered data table

X	Y
-2	-2
0	0
2	2

Exercise: PCA

- Centered data table

X	Y
-2	-2
0	0
2	2

- covariance matrix:

$$\begin{bmatrix} 4 & 4 \\ 4 & 4 \end{bmatrix}$$

Exercise: PCA

- Centered data table

X	Y
-2	-2
0	0
2	2

- covariance matrix:

$$\begin{bmatrix} 4 & 4 \\ 4 & 4 \end{bmatrix}$$

- Find the eigenvalues and eigenvectors:

Exercise: PCA

- Centered data table

X	Y
-2	-2
0	0
2	2

- covariance matrix:

$$\begin{bmatrix} 4 & 4 \\ 4 & 4 \end{bmatrix}$$

- Find the eigenvalues and eigenvectors:
  - eigenvalues : 0,8

Exercise: PCA

- Centered data table

X	Y
-2	-2
0	0
2	2

- covariance matrix:

$$\begin{bmatrix} 4 & 4 \\ 4 & 4 \end{bmatrix}$$

- Find the eigenvalues and eigenvectors:
  - eigenvalues : 0,8
  - eigenvectors:

$$\begin{bmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$$

,

$$\begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$$

Exercise: PCA

- Choose the principal component(s): The eigenvector associated with the largest eigenvalue (8) is $\begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$, so this is the first principal component.

Exercise: PCA

- Choose the principal component(s): The eigenvector associated with the largest eigenvalue (8) is $\begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$, so this is the first principal component.
- Calculate the scores: Project the centered data onto the principal component: Scores = Centered data $\times$ Principal component = $\begin{bmatrix} -2\sqrt{2} & 2\sqrt{2} \\ 0 & 0 \\ 2\sqrt{2} & -2\sqrt{2} \end{bmatrix}$

What is the shortcoming of the standard PCA ?

- Principal Component Analysis (PCA) is a widely used technique for dimensionality reduction. It works by identifying the principal components (directions of maximum variance) in the data.

What is the shortcoming of the standard PCA ?

- Principal Component Analysis (PCA) is a widely used technique for dimensionality reduction. It works by identifying the principal components (directions of maximum variance) in the data.
- However, standard PCA is sensitive to outliers and noise, which can significantly affect the computed principal components.

What is the shortcoming of the standard PCA ?

- Principal Component Analysis (PCA) is a widely used technique for dimensionality reduction. It works by identifying the principal components (directions of maximum variance) in the data.
- However, standard PCA is sensitive to outliers and noise, which can significantly affect the computed principal components.
- Need for Robust PCA: To address the limitations of standard PCA, Robust PCA was developed.

What is the shortcoming of the standard PCA ?

- Principal Component Analysis (PCA) is a widely used technique for dimensionality reduction. It works by identifying the principal components (directions of maximum variance) in the data.
- However, standard PCA is sensitive to outliers and noise, which can significantly affect the computed principal components.
- Need for Robust PCA: To address the limitations of standard PCA, Robust PCA was developed.
- Unlike standard PCA, Robust PCA is designed to separate the low-rank structure of the data (representing the true signal) from the sparse noise or outliers. This makes it more suitable for real-world datasets that are often corrupted by noise or contain anomalies.

Mathmatical Formulation of Robust PCA

- Objective: Decompose a data matrix X into a low-rank matrix L and a sparse matrix S, such that $X = L + S$.

Mathmatical Formulation of Robust PCA

- Objective: Decompose a data matrix X into a low-rank matrix L and a sparse matrix S, such that $X = L + S$.
- Optimization Problem: The decomposition is achieved by solving:

$$\min_{L,S} \quad ||L||_* + \lambda ||S||_1$$

$$\text{subject to} \quad X = L + S$$

where $||L||_*$ is the trace norm of L, $||S||_1$ is the L_1 norm of S, and λ is a regularization parameter.

Mathmatical Formulation of Robust PCA

- Objective: Decompose a data matrix X into a low-rank matrix L and a sparse matrix S, such that $X = L + S$.
- Optimization Problem: The decomposition is achieved by solving:

$$\min_{L,S} \quad ||L||_* + \lambda ||S||_1$$

$$\text{subject to} \quad X = L + S$$

  where $||L||_*$ is the trace norm of L, $||S||_1$ is the L_1 norm of S, and λ is a regularization parameter.
- The trace norm of a matrix A is defined as the sum of its singular values (eigenvalues).

Mathmatical Formulation of Robust PCA

- Objective: Decompose a data matrix X into a low-rank matrix L and a sparse matrix S, such that $X = L + S$.
- Optimization Problem: The decomposition is achieved by solving:

$$\min_{L,S} \quad ||L||_* + \lambda ||S||_1$$

$$\text{subject to} \quad X = L + S$$

  where $||L||_*$ is the trace norm of L, $||S||_1$ is the L_1 norm of S, and λ is a regularization parameter.
- The trace norm of a matrix A is defined as the sum of its singular values (eigenvalues).
- The L-1 norm of a matrix A is defined as the sum of the absolute values of all its entries.

Clustering

- First, let us see an example below:

Clustering

- First, let us see an example below:
- A bank wants to give credit card offers to its customers. Currently, they look at the details of each customer and, based on this information, decide which offer should be given to which customer.

Clustering

- First, let us see an example below:
- A bank wants to give credit card offers to its customers. Currently, they look at the details of each customer and, based on this information, decide which offer should be given to which customer.
- Now, the bank can potentially have millions of customers. Does it make sense to look at the details of each customer separately and then make a decision?

Clustering

- First, let us see an example below:
- A bank wants to give credit card offers to its customers. Currently, they look at the details of each customer and, based on this information, decide which offer should be given to which customer.
- Now, the bank can potentially have millions of customers. Does it make sense to look at the details of each customer separately and then make a decision?
- Certainly not! It is a manual process and will take a huge amount of time.

Clustering

- First, let us see an example below:
- A bank wants to give credit card offers to its customers. Currently, they look at the details of each customer and, based on this information, decide which offer should be given to which customer.
- Now, the bank can potentially have millions of customers. Does it make sense to look at the details of each customer separately and then make a decision?
- Certainly not! It is a manual process and will take a huge amount of time.
- So what can the bank do?

Clustering

- One option is to segment its customers into different groups. For instance, the bank can group the customers based on their income:

Clustering

- One option is to segment its customers into different groups. For instance, the bank can group the customers based on their income:

Clustering

- One option is to segment its customers into different groups. For instance, the bank can group the customers based on their income:

Clustering

- One option is to segment its customers into different groups. For instance, the bank can group the customers based on their income:

- The bank can now make three different strategies or offers, one for each group, instead of creating different strategies for individual customers.

Clustering

- One option is to segment its customers into different groups. For instance, the bank can group the customers based on their income:

- The bank can now make three different strategies or offers, one for each group, instead of creating different strategies for individual customers.
- Note that "high income," "average income," "low income" are not pre-specified lable, but the outcome of clustering. So, this is un-supervised learning.

Clustering

- Clustering is the process of dividing the entire data into groups (also known as clusters) based on the patterns in the data.

Clustering

- Clustering is the process of dividing the entire data into groups (also known as clusters) based on the patterns in the data.
- Specifically, we are given a training set $x^{(1)}, ..., x^{(m)}$, and want to group the data into a few cohesive "clusters."

Clustering

- Clustering is the process of dividing the entire data into groups (also known as clusters) based on the patterns in the data.
- Specifically, we are given a training set $x^{(1)}, ..., x^{(m)}$, and want to group the data into a few cohesive "clusters."
- Here, we are given feature vectors for each data point $x^{(i)} \in \mathbb{R}^n$ as usual; but no labels $y^{(i)}$ (making this an unsupervised learning problem).

Clustering

- Clustering is the process of dividing the entire data into groups (also known as clusters) based on the patterns in the data.
- Specifically, we are given a training set $x^{(1)}, ..., x^{(m)}$, and want to group the data into a few cohesive "clusters."
- Here, we are given feature vectors for each data point $x^{(i)} \in \mathbb{R}^n$ as usual; but no labels $y^{(i)}$ (making this an unsupervised learning problem).
- Our goal is to predict k centroids and a label "$c^{(i)}$" (corresponding to different clusters) for each datapoint. The k-means clustering algorithm is as follows:

K-Means

- K-Means is one of the most popular "clustering" algorithms.

K-Means

- K-Means is one of the most popular "clustering" algorithms.
- K-Means stores k centroids that it uses to define clusters. A point is considered to be in a particular cluster if it is closer to that cluster's centroid than any other centroid.

K-Means

- K-Means is one of the most popular "clustering" algorithms.
- K-Means stores k centroids that it uses to define clusters. A point is considered to be in a particular cluster if it is closer to that cluster's centroid than any other centroid.
- K-Means finds the best centroids by alternating between

K-Means

- K-Means is one of the most popular "clustering" algorithms.
- K-Means stores k centroids that it uses to define clusters. A point is considered to be in a particular cluster if it is closer to that cluster's centroid than any other centroid.
- K-Means finds the best centroids by alternating between
  - (1) assigning data points to clusters based on the current centroids

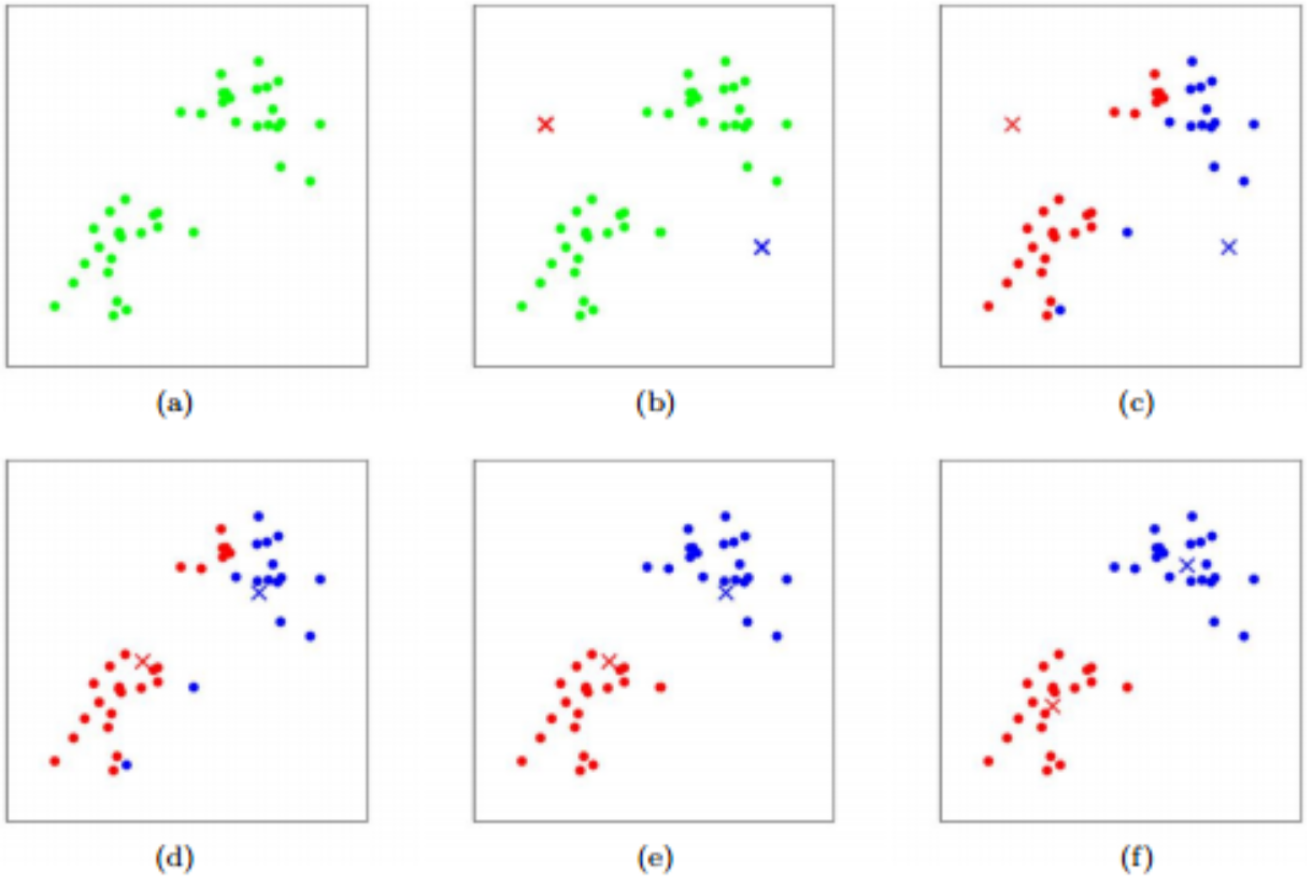
K-Means

- K-Means is one of the most popular "clustering" algorithms.
- K-Means stores k centroids that it uses to define clusters. A point is considered to be in a particular cluster if it is closer to that cluster's centroid than any other centroid.
- K-Means finds the best centroids by alternating between
  - (1) assigning data points to clusters based on the current centroids
  - (2) choosing centroids (points which are the center of a cluster) based on the current assignment of data points to clusters.

K-Means

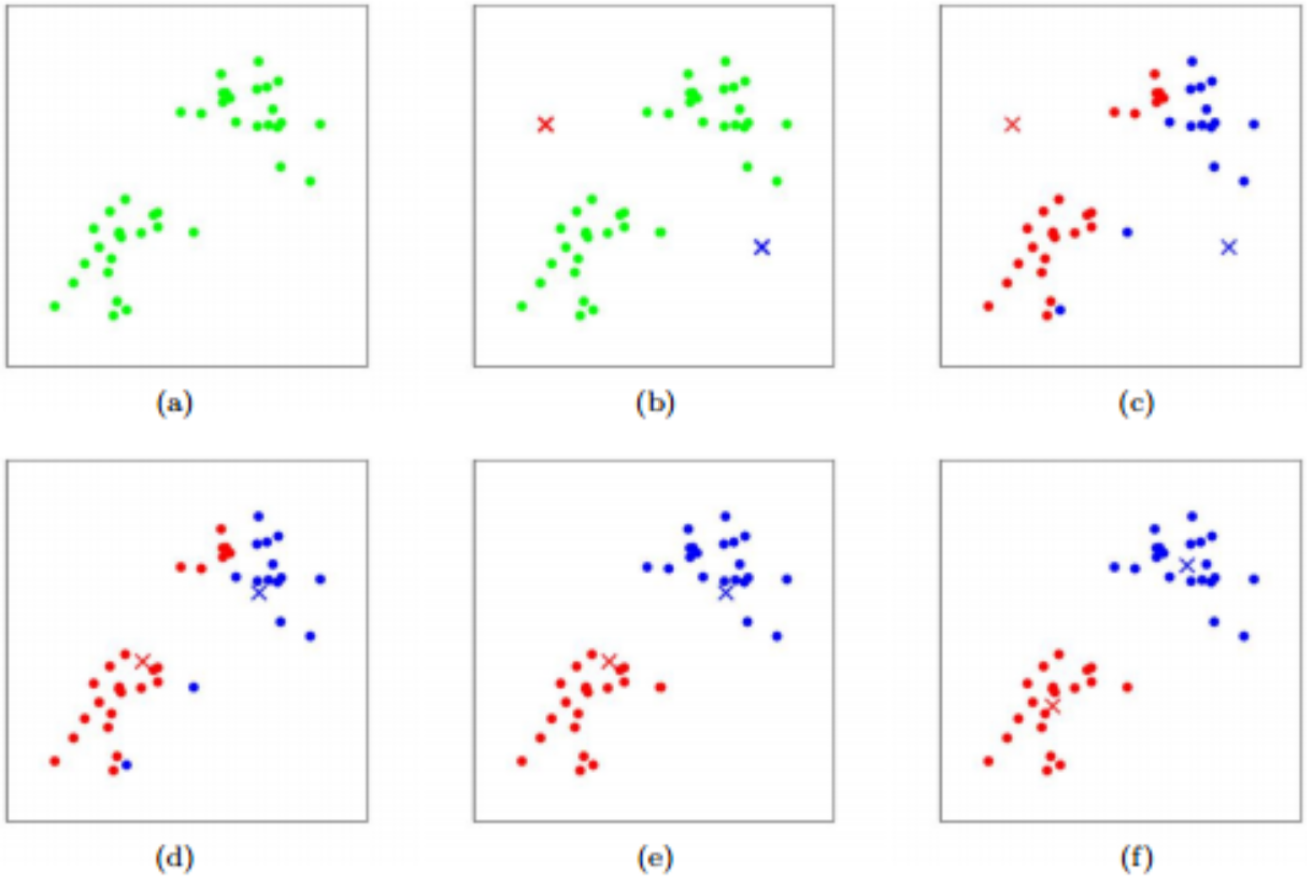
- K-Means is one of the most popular "clustering" algorithms.
- K-Means stores k centroids that it uses to define clusters. A point is considered to be in a particular cluster if it is closer to that cluster's centroid than any other centroid.
- K-Means finds the best centroids by alternating between
  - (1) assigning data points to clusters based on the current centroids
  - (2) choosing centroids (points which are the center of a cluster) based on the current assignment of data points to clusters.
- The value of K is often pre-specififed.

K-Means



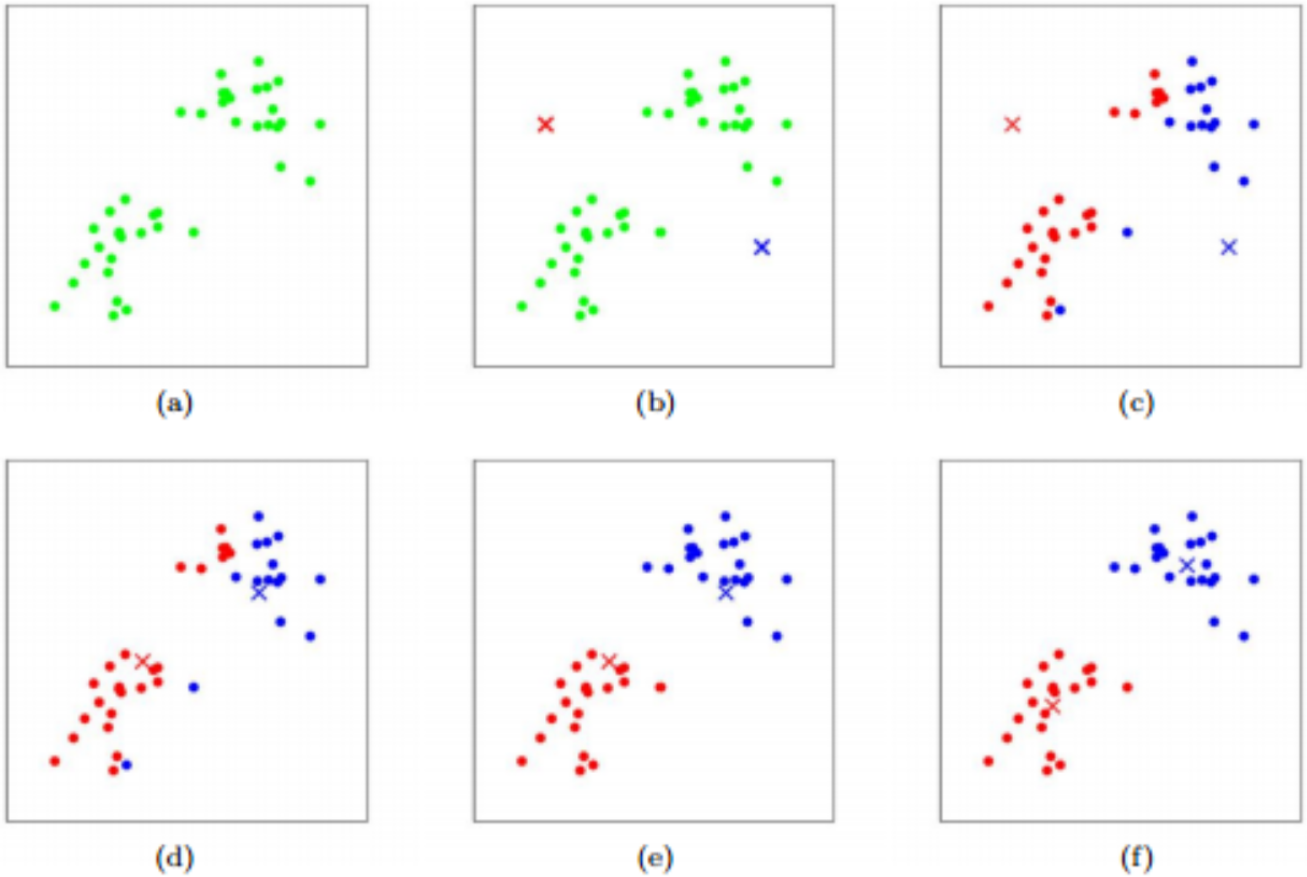
- Training examples are shown as dots, and cluster centroids are shown as crosses.

K-Means



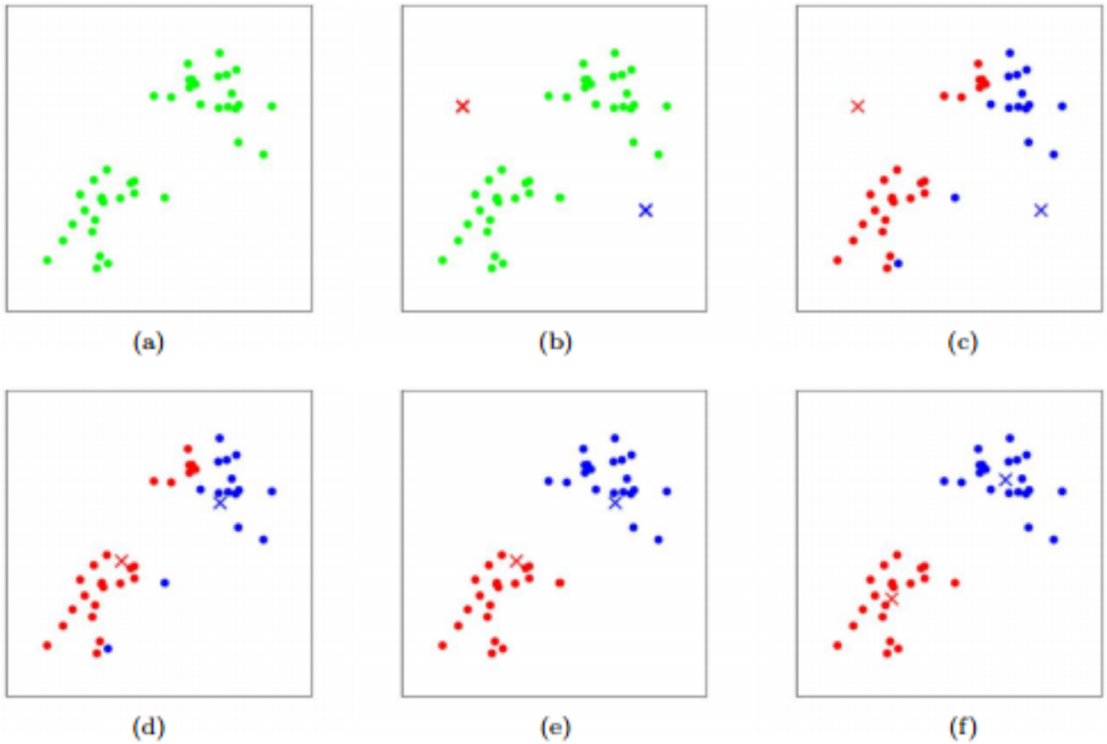
- (a) Original dataset.

K-Means



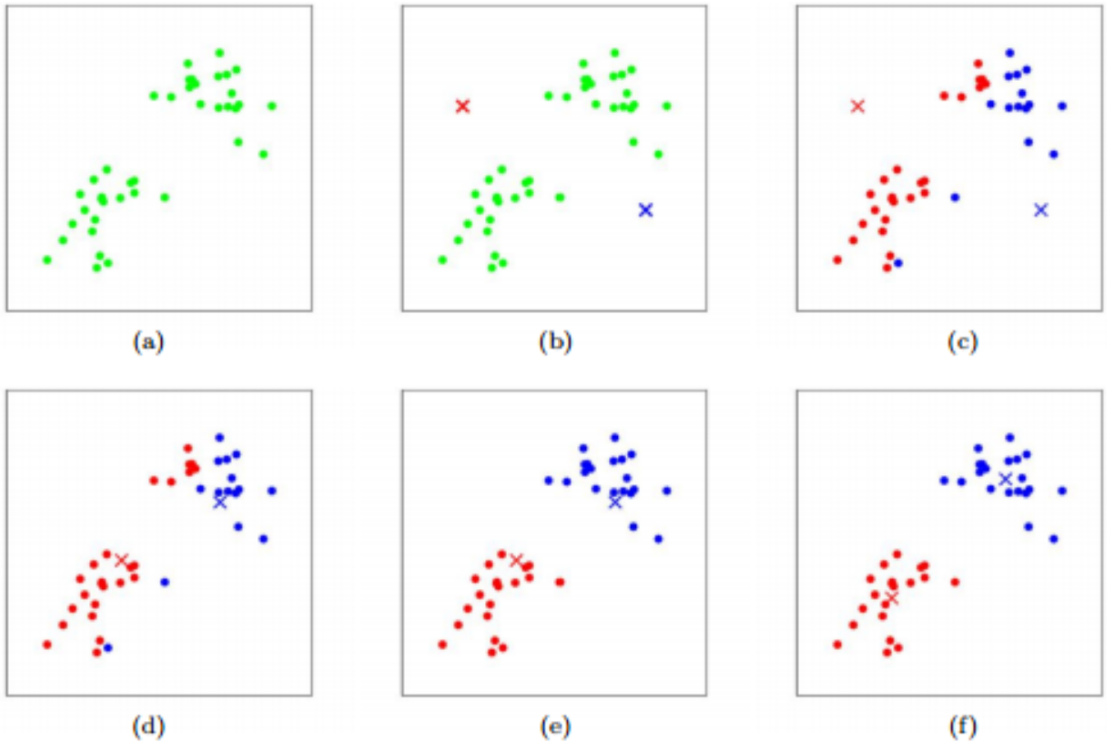
- (b) Random initial cluster centroids.

K-Means



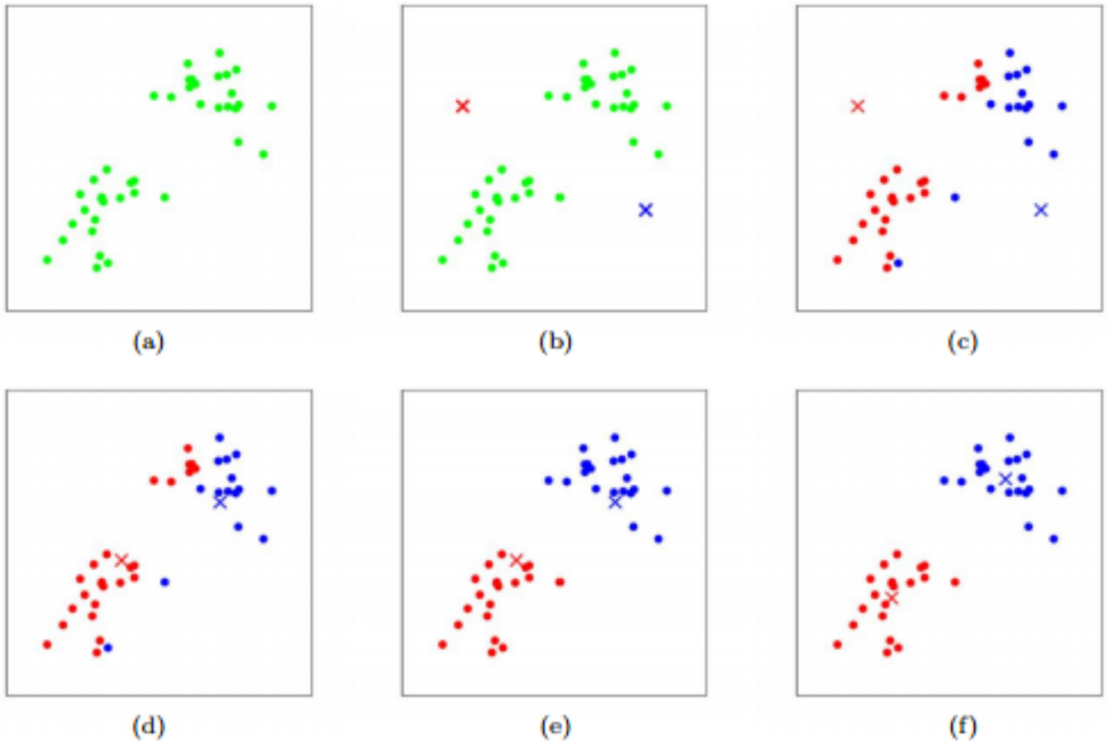
- (c-f) Illustration of running two iterations of k-means.

K-Means



- In each iteration, we assign each training example to the closest cluster centroid (shown by "painting" the training examples the same color as the cluster centroid to which is assigned);

K-Means



- In each iteration, we assign each training example to the closest cluster centroid (shown by "painting" the training examples the same color as the cluster centroid to which is assigned);
- Then we move each cluster centroid to the mean of the points assigned to it.

K-Means: the algorithm

- Intialize cluster centroids $\mu_1, \mu_2, \cdots, \mu_k \in \mathbb{R}^n$ randomly.

K-Means: the algorithm

- Intialize cluster centroids $\mu_1, \mu_2, \cdots, \mu_k \in \mathbb{R}^n$ randomly.
- Repeat until convergence:

K-Means: the algorithm

- Intialize cluster centroids $\mu_1, \mu_2, \cdots, \mu_k \in \mathbb{R}^n$ randomly.
- Repeat until convergence:
  - For every i, set $c^{(i)} = \text{argmin}_j ||x^{(i)} - \mu_j||^2$

K-Means: the algorithm

- Intialize cluster centroids $\mu_1, \mu_2, \cdots, \mu_k \in \mathbb{R}^n$ randomly.
- Repeat until convergence:
  - For every i, set $c^{(i)} = \text{argmin}_j ||x^{(i)} - \mu_j||^2$
  - For each j, set $\mu_j = \frac{\sum_{i=1}^{m} 1\{c^{(i)}=j\} x^{(i)}}{\sum_{i=1}^{m} 1\{c^{(i)}=j\}}$

Exercise: K-Means Clustering on a Small Dataset

- Objective: Apply K-Means clustering to a small dataset.

Exercise: K-Means Clustering on a Small Dataset

- Objective: Apply K-Means clustering to a small dataset.
- Dataset: $X = [(1,1),(1,2),(2,1),(2,2),(4,4),(4,5),(5,4),(5,5)]$

Exercise: K-Means Clustering on a Small Dataset

- Objective: Apply K-Means clustering to a small dataset.
- Dataset: $X = [(1,1),(1,2),(2,1),(2,2),(4,4),(4,5),(5,4),(5,5)]$
- Tasks:

Exercise: K-Means Clustering on a Small Dataset

- Objective: Apply K-Means clustering to a small dataset.
- Dataset: $X = [(1,1),(1,2),(2,1),(2,2),(4,4),(4,5),(5,4),(5,5)]$
- Tasks:
  - (1) Initialize Centroids: Choose $k = 2$ and initial centroids as $(1,1)$ and $(1,2)$.

Exercise: K-Means Clustering on a Small Dataset

- Objective: Apply K-Means clustering to a small dataset.
- Dataset: $X = [(1,1),(1,2),(2,1),(2,2),(4,4),(4,5),(5,4),(5,5)]$
- Tasks:
  - (1) Initialize Centroids: Choose $k = 2$ and initial centroids as $(1,1)$ and $(1,2)$.
  - (2) Assign Points to Clusters: Assign each point to the nearest centroid.

Exercise: K-Means Clustering on a Small Dataset

- Objective: Apply K-Means clustering to a small dataset.
- Dataset: $X = [(1,1),(1,2),(2,1),(2,2),(4,4),(4,5),(5,4),(5,5)]$
- Tasks:
  - (1) Initialize Centroids: Choose $k = 2$ and initial centroids as $(1,1)$ and $(1,2)$.
  - (2) Assign Points to Clusters: Assign each point to the nearest centroid.
  - (3) Update Centroids: Recalculate the centroids as the mean of the points in each cluster.

Exercise: K-Means Clustering on a Small Dataset

- Objective: Apply K-Means clustering to a small dataset.
- Dataset: $X = [(1,1),(1,2),(2,1),(2,2),(4,4),(4,5),(5,4),(5,5)]$
- Tasks:
  - (1) Initialize Centroids: Choose $k = 2$ and initial centroids as $(1,1)$ and $(1,2)$.
  - (2) Assign Points to Clusters: Assign each point to the nearest centroid.
  - (3) Update Centroids: Recalculate the centroids as the mean of the points in each cluster.
  - (4) Repeat Steps 2 and 3 until the centroids do not change.

Answer: K-Means Clustering on a Small Dataset

- Initial Centroids:

Answer: K-Means Clustering on a Small Dataset

- Initial Centroids:
  - Centroid 1: $(1, 1)$

Answer: K-Means Clustering on a Small Dataset

- Initial Centroids:
  - Centroid 1: $(1, 1)$
  - Centroid 2: $(1, 2)$

Answer: K-Means Clustering on a Small Dataset

- Initial Centroids:
  - Centroid 1: $(1, 1)$
  - Centroid 2: $(1, 2)$
- First Iteration:

Answer: K-Means Clustering on a Small Dataset

- Initial Centroids:
  - Centroid 1: $(1, 1)$
  - Centroid 2: $(1, 2)$
- First Iteration:
  - Cluster 1: $[(1, 1), (1, 2), (2, 1), (2, 2)]$

Answer: K-Means Clustering on a Small Dataset

- Initial Centroids:
  - Centroid 1: $(1, 1)$
  - Centroid 2: $(1, 2)$
- First Iteration:
  - Cluster 1: $[(1, 1), (1, 2), (2, 1), (2, 2)]$
  - Cluster 2: $[(4, 4), (4, 5), (5, 4), (5, 5)]$

Answer: K-Means Clustering on a Small Dataset

- Initial Centroids:
  - Centroid 1: $(1, 1)$
  - Centroid 2: $(1, 2)$
- First Iteration:
  - Cluster 1: $[(1, 1), (1, 2), (2, 1), (2, 2)]$
  - Cluster 2: $[(4, 4), (4, 5), (5, 4), (5, 5)]$
  - New Centroids: Centroid 1: $(1.5, 1.5)$, Centroid 2: $(4.5, 4.5)$

Answer: K-Means Clustering on a Small Dataset

- Initial Centroids:
  - Centroid 1: $(1, 1)$
  - Centroid 2: $(1, 2)$
- First Iteration:
  - Cluster 1: $[(1, 1), (1, 2), (2, 1), (2, 2)]$
  - Cluster 2: $[(4, 4), (4, 5), (5, 4), (5, 5)]$
  - New Centroids: Centroid 1: $(1.5, 1.5)$, Centroid 2: $(4.5, 4.5)$
- Second Iteration:

Answer: K-Means Clustering on a Small Dataset

- Initial Centroids:
  - Centroid 1: $(1, 1)$
  - Centroid 2: $(1, 2)$
- First Iteration:
  - Cluster 1: $[(1, 1), (1, 2), (2, 1), (2, 2)]$
  - Cluster 2: $[(4, 4), (4, 5), (5, 4), (5, 5)]$
  - New Centroids: Centroid 1: $(1.5, 1.5)$, Centroid 2: $(4.5, 4.5)$
- Second Iteration:
  - No change in cluster assignment.

Answer: K-Means Clustering on a Small Dataset

- Initial Centroids:
  - Centroid 1: $(1, 1)$
  - Centroid 2: $(1, 2)$
- First Iteration:
  - Cluster 1: $[(1, 1), (1, 2), (2, 1), (2, 2)]$
  - Cluster 2: $[(4, 4), (4, 5), (5, 4), (5, 5)]$
  - New Centroids: Centroid 1: $(1.5, 1.5)$, Centroid 2: $(4.5, 4.5)$
- Second Iteration:
  - No change in cluster assignment.
  - Algorithm converges.

Answer: K-Means Clustering on a Small Dataset

- Initial Centroids:
  - Centroid 1: $(1, 1)$
  - Centroid 2: $(1, 2)$
- First Iteration:
  - Cluster 1: $[(1, 1), (1, 2), (2, 1), (2, 2)]$
  - Cluster 2: $[(4, 4), (4, 5), (5, 4), (5, 5)]$
  - New Centroids: Centroid 1: $(1.5, 1.5)$, Centroid 2: $(4.5, 4.5)$
- Second Iteration:
  - No change in cluster assignment.
  - Algorithm converges.
- Final Centroids:

Answer: K-Means Clustering on a Small Dataset

- Initial Centroids:
  - Centroid 1: $(1, 1)$
  - Centroid 2: $(1, 2)$
- First Iteration:
  - Cluster 1: $[(1, 1), (1, 2), (2, 1), (2, 2)]$
  - Cluster 2: $[(4, 4), (4, 5), (5, 4), (5, 5)]$
  - New Centroids: Centroid 1: $(1.5, 1.5)$, Centroid 2: $(4.5, 4.5)$
- Second Iteration:
  - No change in cluster assignment.
  - Algorithm converges.
- Final Centroids:
  - Centroid 1: $(1.5, 1.5)$

Answer: K-Means Clustering on a Small Dataset

- Initial Centroids:
  - Centroid 1: $(1, 1)$
  - Centroid 2: $(1, 2)$
- First Iteration:
  - Cluster 1: $[(1, 1), (1, 2), (2, 1), (2, 2)]$
  - Cluster 2: $[(4, 4), (4, 5), (5, 4), (5, 5)]$
  - New Centroids: Centroid 1: $(1.5, 1.5)$, Centroid 2: $(4.5, 4.5)$
- Second Iteration:
  - No change in cluster assignment.
  - Algorithm converges.
- Final Centroids:
  - Centroid 1: $(1.5, 1.5)$
  - Centroid 2: $(4.5, 4.5)$

Answer: K-Means Clustering on a Small Dataset

- Initial Centroids:
  - Centroid 1: $(1, 1)$
  - Centroid 2: $(1, 2)$
- First Iteration:
  - Cluster 1: $[(1, 1), (1, 2), (2, 1), (2, 2)]$
  - Cluster 2: $[(4, 4), (4, 5), (5, 4), (5, 5)]$
  - New Centroids: Centroid 1: $(1.5, 1.5)$, Centroid 2: $(4.5, 4.5)$
- Second Iteration:
  - No change in cluster assignment.
  - Algorithm converges.
- Final Centroids:
  - Centroid 1: $(1.5, 1.5)$
  - Centroid 2: $(4.5, 4.5)$
- Number of Iterations: 2.

Fuzzy C-Means clustering

- Fuzzy C-Means (FCM) is a clustering algorithm that allows one piece of data to belong to two or more clusters.

Fuzzy C-Means clustering

- Fuzzy C-Means (FCM) is a clustering algorithm that allows one piece of data to belong to two or more clusters.
- This method is frequently used in pattern recognition and is an extension of the traditional k-means clustering algorithm.

Fuzzy C-Means clustering

- Fuzzy C-Means (FCM) is a clustering algorithm that allows one piece of data to belong to two or more clusters.
- This method is frequently used in pattern recognition and is an extension of the traditional k-means clustering algorithm.
- Unlike k-means, where each data point belongs to exactly one cluster, FCM introduces the concept of membership levels, allowing data points to have varying degrees of belonging to multiple clusters.

Fuzzy C-Means clustering

- Given a dataset $X = \{x_1, x_2, \cdots, x_n\}$ consisting of n data points, the the goal of FCM is to partition the data into c fuzzy clusters, where c is a predefined number of clusters.

Fuzzy C-Means clustering

- Given a dataset $X = \{x_1, x_2, \cdots, x_n\}$ consisting of n data points, the the goal of FCM is to partition the data into c fuzzy clusters, where c is a predefined number of clusters.
- Each data point x_i has a degree of membership u_{ij} in each cluster j, where u_{ij} is a number between 0 and 1 representing the degree to which x_i belongs to cluster j.

Fuzzy C-Means clustering

- The FCM algorithm aims to minimize the following objective function:

$$J(U, V) = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^{m} \|x_i - v_j\|^2$$

where

Fuzzy C-Means clustering

- The FCM algorithm aims to minimize the following objective function:

$$J(U, V) = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^{m} \|x_i - v_j\|^2$$

where

- $U = [u_{ij}]$ is the membership matrix

Fuzzy C-Means clustering

- The FCM algorithm aims to minimize the following objective function:

$$J(U, V) = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^{m} \|x_i - v_j\|^2$$

where

- $U = [u_{ij}]$ is the membership matrix
- $V = \{v_1, v_2, \cdots, v_C\}$ is the set of cluster centers

Fuzzy C-Means clustering

- The FCM algorithm aims to minimize the following objective function:

$$J(U, V) = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^{m} \|x_i - v_j\|^2$$

where

- $U = [u_{ij}]$ is the membership matrix
- $V = \{v_1, v_2, \cdots, v_C\}$ is the set of cluster centers
- m is the fuzziness parameter which controls the level of cluster fuzziness

Fuzzy C-Means clustering

- The FCM algorithm aims to minimize the following objective function:

$$J(U, V) = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^{m} \|x_i - v_j\|^2$$

where

- $U = [u_{ij}]$ is the membership matrix
- $V = \{v_1, v_2, \cdots, v_C\}$ is the set of cluster centers
- m is the fuzziness parameter which controls the level of cluster fuzziness
- $||x_i - v_j||$ is the Euclidean distance between the data point x_i and the cluster centerv_j.

Fuzzy C-Means clustering

- The algorithm iteratively updates the membership matrix U and the cluster centers V until convergence, typically using the following update rules:

Fuzzy C-Means clustering

- The algorithm iteratively updates the membership matrix U and the cluster centers V until convergence, typically using the following update rules:
- Membership update:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|x_i - v_j\|}{\|x_i - v_k\|} \right)^{\frac{2}{m-1}}}$$

Fuzzy C-Means clustering

- The algorithm iteratively updates the membership matrix U and the cluster centers V until convergence, typically using the following update rules:
- Membership update:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|x_i - v_j\|}{\|x_i - v_k\|} \right)^{\frac{2}{m-1}}}$$

- Cluster center update:

$$v_j = \frac{\sum_{i=1}^{n} u_{ij}^m x_i}{\sum_{i=1}^{n} u_{ij}^m}$$

Fuzzy C-Means clustering

- The algorithm iteratively updates the membership matrix U and the cluster centers V until convergence, typically using the following update rules:
- Membership update:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|x_i - v_j\|}{\|x_i - v_k\|} \right)^{\frac{2}{m-1}}}$$

- Cluster center update:

$$v_j = \frac{\sum_{i=1}^{n} u_{ij}^m x_i}{\sum_{i=1}^{n} u_{ij}^m}$$

- The algorithm stops when the changes in the membership matrix between consecutive iterations are below a specified threshold.

How to understand the meaning of membership?

- Let us see an example below:

How to understand the meaning of membership?

- Let us see an example below:
- Suppose we have a dataset with four points: A, B, C, and D. We want to cluster these points into two clusters using Fuzzy C-means. After running the algorithm, we might get the following membership values for each point in each cluster:

Point	Cluster 1	Cluster 2
A	0.8	0.2
B	0.3	0.7
C	0.6	0.4
D	0.1	0.9

How to understand the meaning of membership?

- Point A has a high membership value (0.8) in Cluster 1 and a low membership value (0.2) in Cluster 2. This means that Point A is strongly associated with Cluster 1 but has a slight association with Cluster 2.

How to understand the meaning of membership?

- Point A has a high membership value (0.8) in Cluster 1 and a low membership value (0.2) in Cluster 2. This means that Point A is strongly associated with Cluster 1 but has a slight association with Cluster 2.
- Point B has a membership value of 0.3 in Cluster 1 and 0.7 in Cluster 2, indicating that it is more closely associated with Cluster 2.

How to understand the meaning of membership?

- Point A has a high membership value (0.8) in Cluster 1 and a low membership value (0.2) in Cluster 2. This means that Point A is strongly associated with Cluster 1 but has a slight association with Cluster 2.
- Point B has a membership value of 0.3 in Cluster 1 and 0.7 in Cluster 2, indicating that it is more closely associated with Cluster 2.
- Point C has a membership value of 0.6 in Cluster 1 and 0.4 in Cluster 2, suggesting that it belongs more to Cluster 1 but still has some association with Cluster 2.

How to understand the meaning of membership?

- Point A has a high membership value (0.8) in Cluster 1 and a low membership value (0.2) in Cluster 2. This means that Point A is strongly associated with Cluster 1 but has a slight association with Cluster 2.
- Point B has a membership value of 0.3 in Cluster 1 and 0.7 in Cluster 2, indicating that it is more closely associated with Cluster 2.
- Point C has a membership value of 0.6 in Cluster 1 and 0.4 in Cluster 2, suggesting that it belongs more to Cluster 1 but still has some association with Cluster 2.
- Point D has a very low membership value (0.1) in Cluster 1 and a high value (0.9) in Cluster 2, showing that it is strongly associated with Cluster 2.

Fuzzy C-Means clustering

- Advantages

Fuzzy C-Means clustering

- Advantages
  - 1) Gives best result for overlapped data set and comparatively better then k-means algorithm.

Fuzzy C-Means clustering

- Advantages
  - 1) Gives best result for overlapped data set and comparatively better then k-means algorithm.
  - 2) Unlike k-means where data point must exclusively belong to one cluster center, here data point is assigned membership to each cluster center as a result of which data point may belong to more then one cluster center.

Fuzzy C-Means clustering

- Disadvantages

Fuzzy C-Means clustering

- Disadvantages
  - 1) Apriori specification of the number of clusters.

Fuzzy C-Means clustering

- Disadvantages
  - 1) Apriori specification of the number of clusters.
  - 2) Euclidean distance measures can unequally weight underlying factors.