

Econ 425 Week 9 Part I

Clustering and PCA

Grigory Franguridi

UCLA Econ

USC CESR

franguri@usc.edu

Motivation

- what if a dataset has too many variables? *e.g. image data*
- e.g., most of the variables are correlated on analysis
变量和另一边的变量~
- may lead to poor accuracy in estimation
- **dimension reduction** methods
 - Principal Component Analysis (PCA)

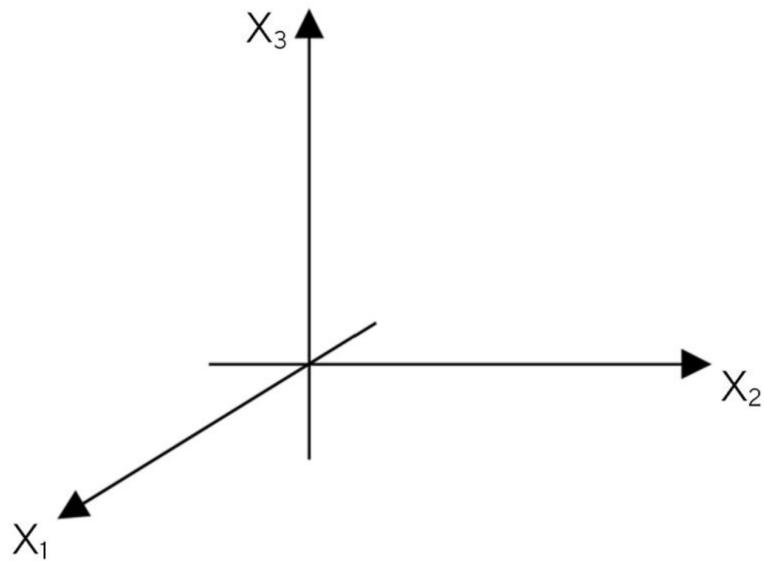
PCA

- summarizes the information content in large datasets with a smaller dataset of “summary indices” that can be more easily visualized and analyzed
- underlying data can be measurements describing properties of production samples, chemical compounds or reactions, time points of a continuous process, batches from a batch process, biological individuals, or trials of a DOE protocol
- often used in the preliminary data analysis, before running any ML tasks

How PCA works

- X is data matrix with N rows (observations) and K columns (features)
- construct a variable space with as many dimensions as there are variables (see figure on the next slide)
- each variable represents a coordinate axis; for each variable, the length is standardized, typically by scaling to unit variance

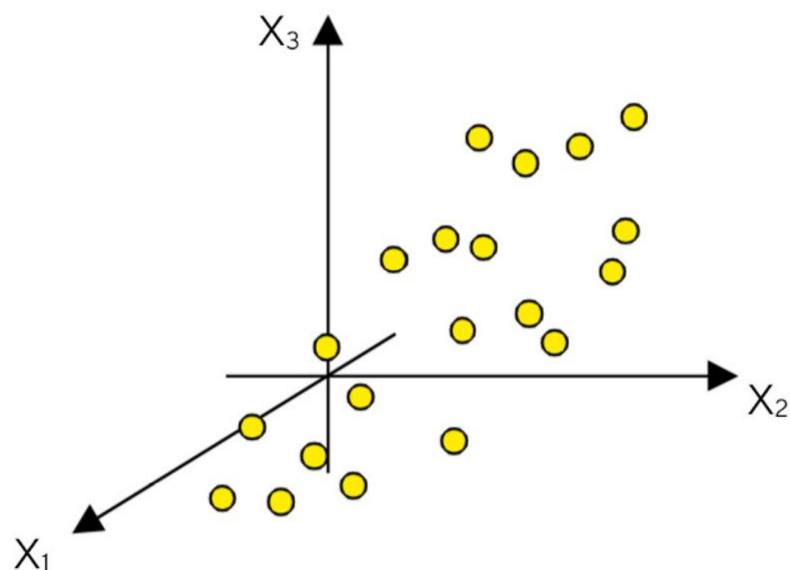
How PCA works



Feature space \mathbb{R}^K . Only three variable axes displayed. The “length” of each coordinate is standardized
坐标

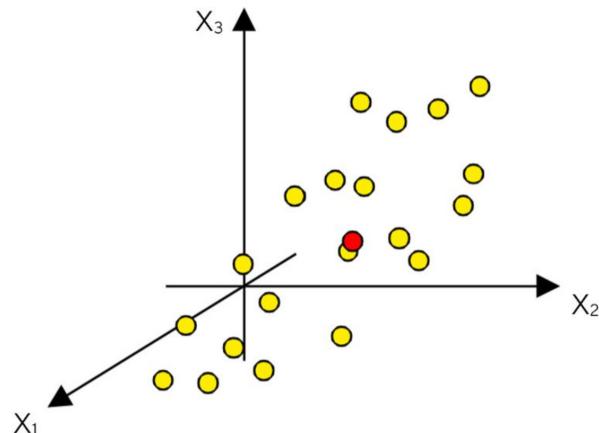
How PCA works

- each observation in X is a point in the feature space \mathbb{R}^K



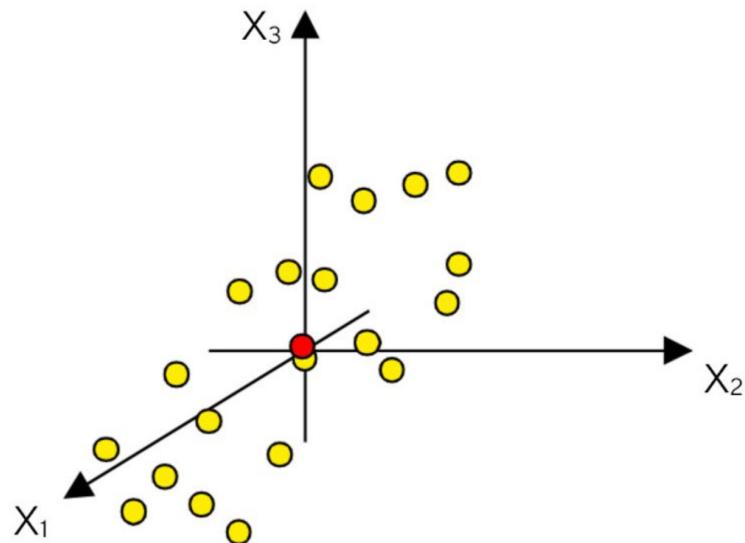
How PCA works

- **centering:** subtract variable averages from the data. The vector of averages is the red point in \mathbb{R}^K



How PCA works

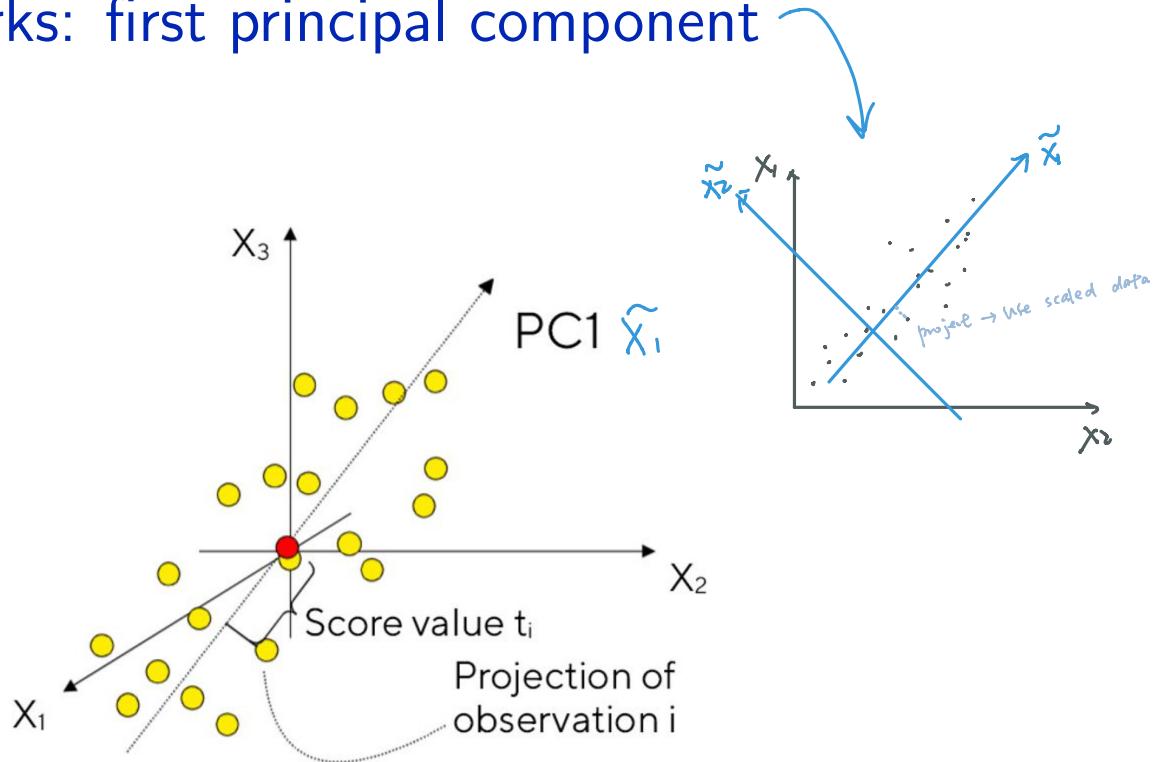
- subtraction of the average corresponds to a re-positioning of the origin of the coordinate system to the average point



How PCA works: first principal component

- ready to compute the first principal component (PC1)
- PC1 is the line through the average point that **best approximates the data in the least squares sense**
- each observation (yellow dot) may now be projected onto this **line** to get the coordinate value along the PC-line (**score**)

How PCA works: first principal component

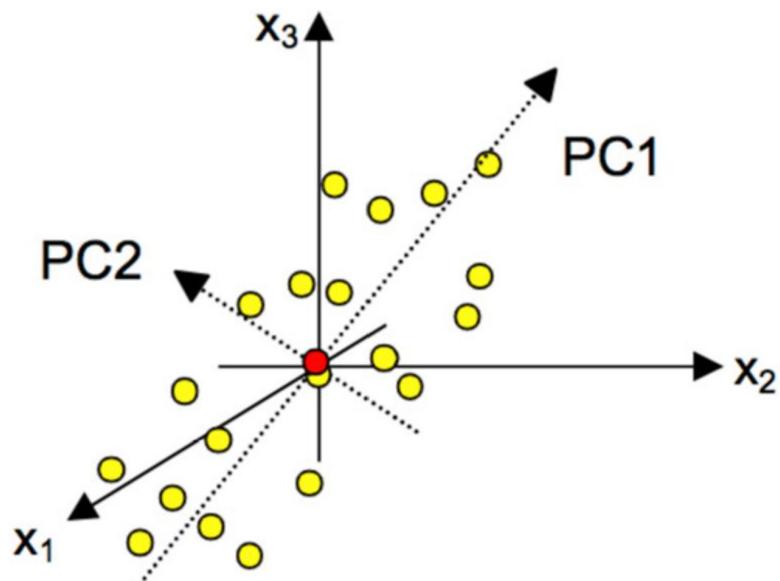


- the first principal component (PC1) represents the *maximum variance direction* in the data

How PCA works: second principal component

- usually one summary index or principal component is insufficient to model the systematic variation of data
- second principal component (PC2) is represented by a line through the average point *orthogonal* to the first PC

The second principal component



- second principal component (PC2) reflects the second largest source of variation in the data while being *orthogonal* to the first PC *Cause the residuals that aren't caught by PC1*
- PC2 also passes through the average point

How to calculate PC1 and PC2?

$$\text{cov}(x, z) = E[(x - \bar{x})(z - \bar{z})] = Exz - EX\bar{z}$$

Data: x_1, x_2, \dots, x_n $\bar{x} = \frac{1}{n} \sum x_i$

$$\hat{\text{cov}}(x, z) = \frac{1}{n} \sum_i (x_i - \bar{x})(z_i - \bar{z}) = \frac{1}{n} \sum_i x_i z_i - \bar{x}\bar{z}$$

Data (already centered)

$$X = \begin{pmatrix} x_{11} & \dots & x_{1K} \\ \vdots & \ddots & \vdots \\ x_{n1} & \dots & x_{nK} \end{pmatrix}$$

$$\hat{\Sigma}_{K \times K} = \begin{pmatrix} \hat{\text{cov}}(x_1, x_1) & \dots & \hat{\text{cov}}(x_1, x_K) \\ \vdots & \ddots & \vdots \\ \hat{\text{cov}}(x_K, x_1) & \dots & \hat{\text{cov}}(x_K, x_K) \end{pmatrix} = \frac{1}{n} X'X$$

- **standardize the data:** each variable should be mean 0 and SD 1 (PCA is sensitive to scaling)
- **calculate the sample covariance matrix**

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

doesnt matter if they're 1-1 (?)

- elements are covariances between each pair of features

How to calculate PC1 and PC2?

- **计算协方差矩阵的特征值和特征向量**: 特征向量 / 特征值表示数据中的最大方差方向 / 方差的大小
- 特征向量 $\{v_1, v_2, \dots, v_K\}$ 和特征值 $\{\lambda_1, \lambda_2, \dots, \lambda_K\}$ 满足

$$\widehat{\Sigma}v_k = \lambda_k v_k$$

- 特征分解:

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

where V is a matrix of eigenvectors, Λ is a diagonal matrix with eigenvalues on the diagonal

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:

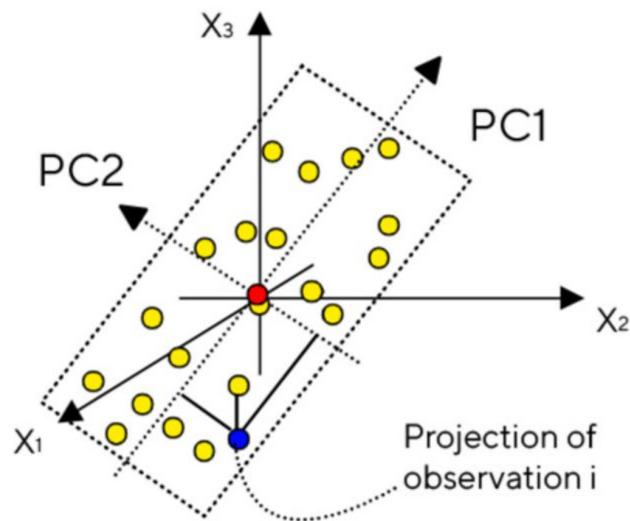
$$PC1 = v_1$$

$$PC2 = v_2$$

The model plane .

- PC1 and PC2 together define a plane in \mathbb{R}^K
- visualize the data by projecting observations onto this low-dimensional subspace and plotting (**score plot**)
- coordinate values of the observations on this plane **scores**

The model plane



- PC1 and PC2 form a plane, which can be visualized graphically. Projections of observations onto the plane are called **scores**

What is the score?

- eigenvectors $\{v_1, v_2, \dots, v_K\}$ and eigenvalues $\{\lambda_1, \lambda_2, \dots, \lambda_K\}$ satisfy

$$\hat{\Sigma}v_k = \lambda_k v_k$$

- eigendecomposition:

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

where V is a matrix of eigenvectors, Λ is a diagonal matrix with eigenvalues on the diagonal

- PC scores of (standardized) X are obtained by multiplying X by the loadings (eigenvectors) of the covariance of X , say $\hat{\Sigma}$
- recall that V is the matrix of eigenvectors (loadings) of $\hat{\Sigma}$
- order the columns of V by their corresponding eigenvalues in descending order
- scores:

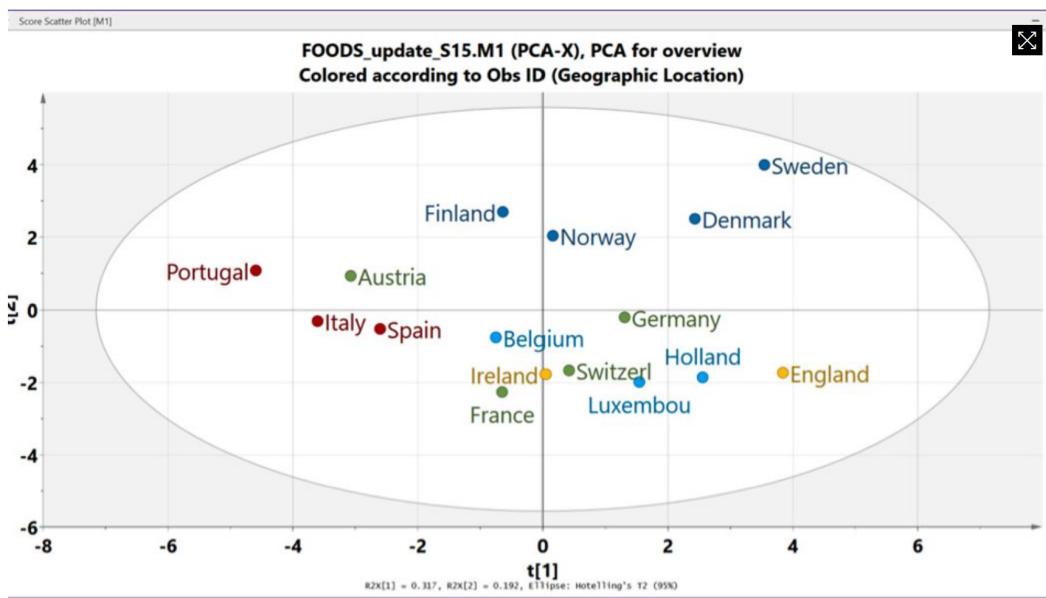
$$T_{n \times k} = X V$$

- the first column of T contains the scores for PC1, the second column contains the scores for PC2, etc.

PCA: example

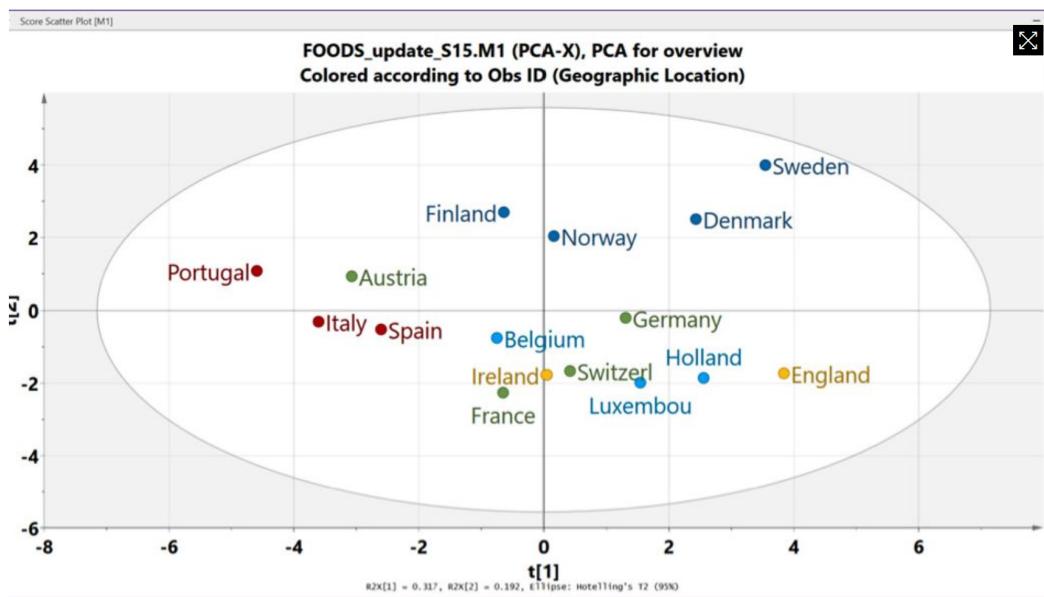
- data: food consumption in European countries
- figure on the next slide displays the score plot of the first two principal components (scores t_1 and t_2)
- the score plot is a map of 16 countries: those close to each other have similar food consumption profiles

PCA: example



- score plot illustrates relations between food consumption profiles
- PC1 (PC2) explains 32% (19%) of the variation of the data

PCA: example



- Nordic countries (Finland, Norway, Denmark, and Sweden) are in the upper right corner
- Belgium and Germany are close to the center (origin) of the plot

PCA: exercise

- dataset:

x_1	x_2
1	2
3	4
5	6

- perform PCA on this dataset by following these steps:
- demean 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 PCs
- calculate the scores**: project the centered data onto PCs

Exercise: PCA

- demeaned data:

x_1	x_2
-2	-2
0	0
2	2

- covariance matrix:

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

- eigenvalues and eigenvectors:

- eigenvalues: 0, 8
- eigenvectors:

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

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

$\Rightarrow \text{product} = 0 \Rightarrow \text{垂直}$

PCA: exercise

- choose the principal component(s): eigenvector associated with the largest eigenvalue (8) is $\begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}$ (PC1)
- calculate the scores (projections of the centered data onto PC1):

$$\text{scores} = \text{centered data} \times \text{PC} = \begin{pmatrix} -2\sqrt{2} & 2\sqrt{2} \\ 0 & 0 \\ 2\sqrt{2} & -2\sqrt{2} \end{pmatrix}$$

Shortcomings of standard PCA

- sensitive to outliers and noise, which can significantly affect the computed PCs
- **robust PCA** is designed to separate the low-rank structure of the data (true signal) from the sparse noise or outliers
 - more suitable for datasets corrupted by noise or anomalies

Robust PCA

- objective: decompose X into a **low-rank** matrix L and a **sparse** matrix S such that $X = L + S$
- achieved by solving

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

$$\text{s.t. } X = L + S$$

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

- $\|L\|_*$ is the sum of **singular values** (eigenvalues of $L'L$)
- $\|S\|_1$ is the sum of **absolute values** of entries of S

Clustering

- a bank wants to give credit card offers to its customers; currently, they use customer data to decide which offer should be given to which customer
- the bank can potentially have millions of customers; should it use customer-level data?
- what can the bank do?

Clustering

- segment the customers into different groups, e.g. income groups:



- now only three strategies are required, one for each income group
- “high”, “average”, “low” are not prespecified labels, but outcomes of clustering (**unsupervised learning**)

K-means clustering

簇... - 一组点的平均中心

- one of the most popular clustering algorithms
- stores K **centroids** used to define clusters
- a point is in a cluster if it is closer to that cluster's centroid than any other centroid
- finds the best centroids by alternating between
 - 1 assigning data points to clusters based on current centroids
 - 2 choosing centroids based on the current assignment of data points to clusters
- K is either prespecified or tuned, e.g., by cross-validation

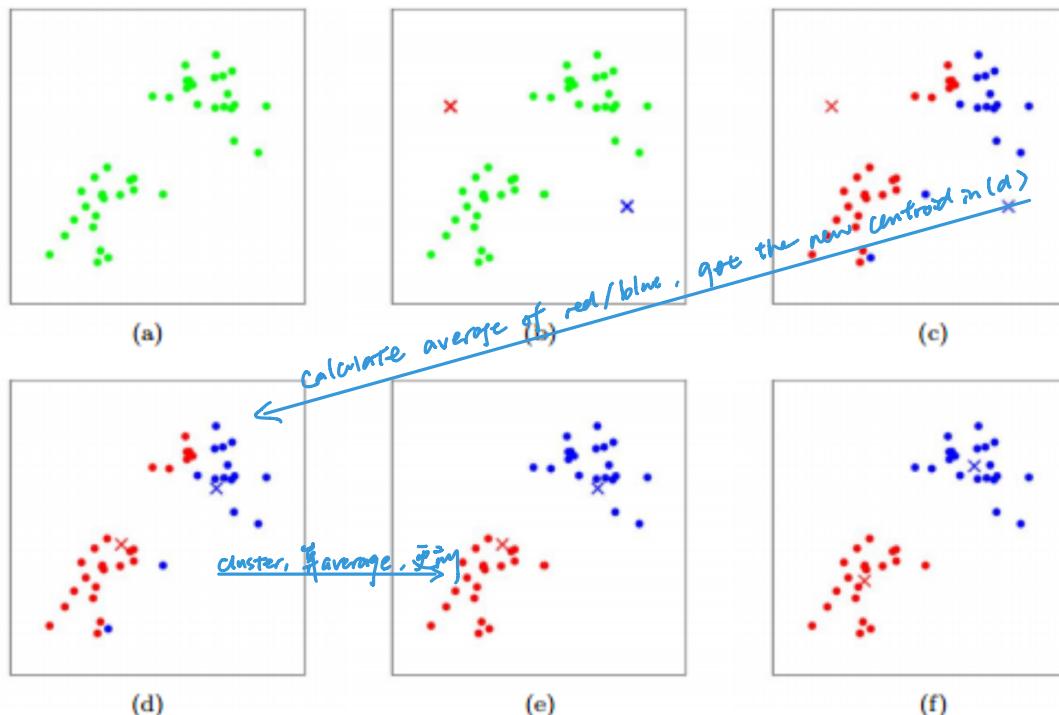
K均值聚类算法的主要步骤如下：

1. **初始化质心**：随机选择K个数据点作为初始质心。
2. **分配数据点**：对于每个数据点，计算其与每个质心的距离，并将其分配给距离最近的质心所代表的簇。
3. **更新质心**：对于每个簇，计算其中所有数据点的平均值，并将该平均值作为新的质心位置。
4. **迭代**：重复步骤2和步骤3，直到质心不再变化或达到预定的迭代次数。

K均值聚类算法的目标是最小化簇内数据点与其所属质心之间的距离，即最小化簇内的方差。因此，在更新质心的步骤中，质心被移动到簇内所有数据点的平均位置，以减小数据点与质心之间的距离。

最终，经过迭代调整后得到的质心就是聚类分析中得出的中心点，它们代表了每个簇的中心位置，用于描述数据集的聚类结构。

K-Means

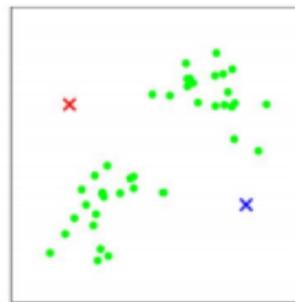


- training examples are dots, cluster centroids are crosses

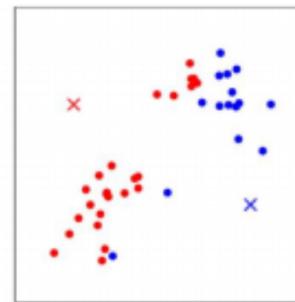
K-Means



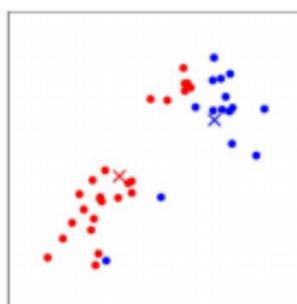
(a)



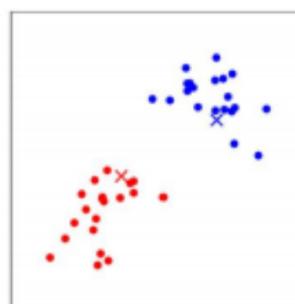
(b)



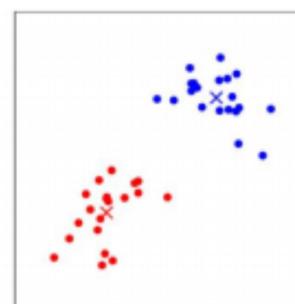
(c)



(d)



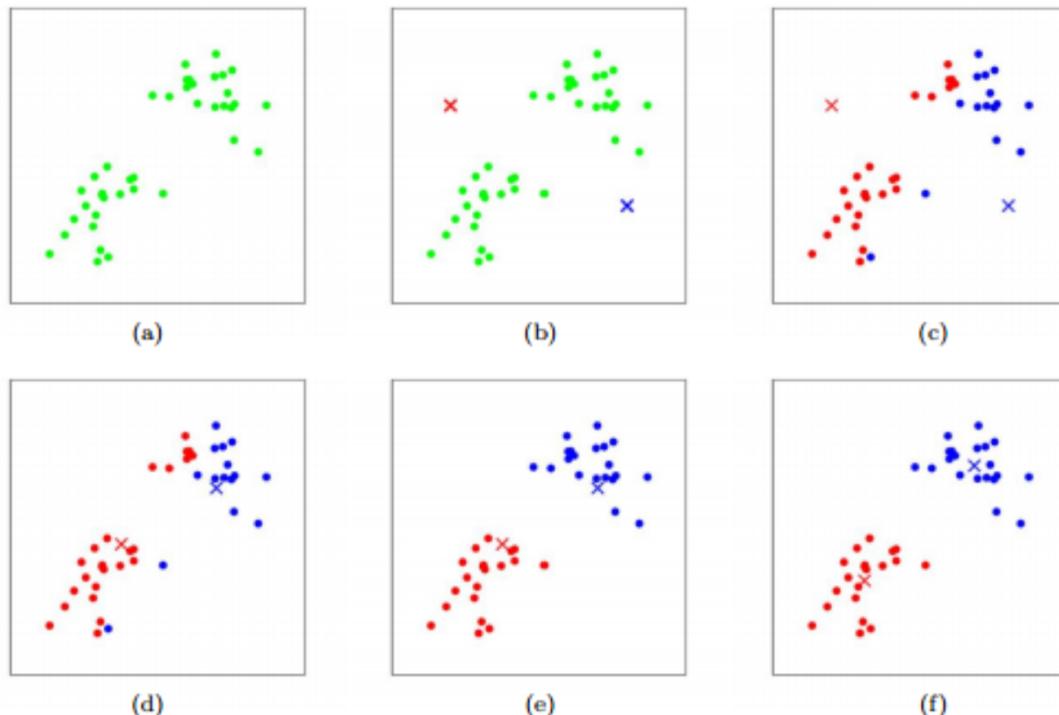
(e)



(f)

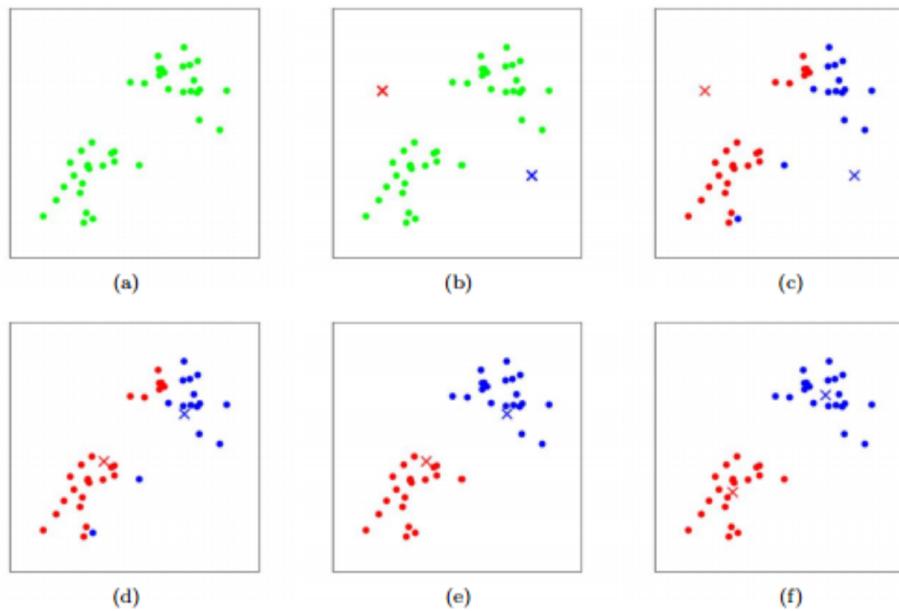
(a) original data

K-Means



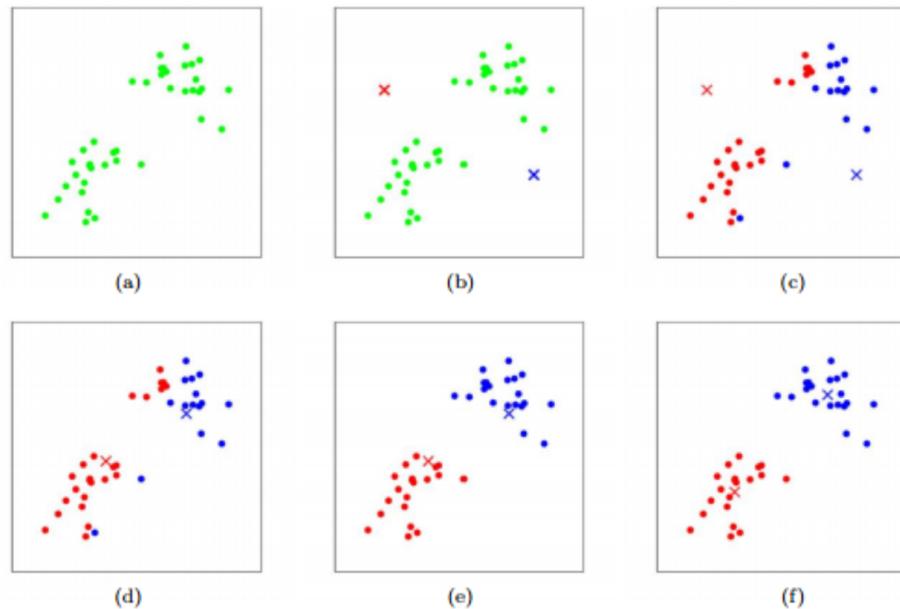
(b) random initialization of cluster centroids

K-Means



(c)-(f) two iterations of K -means clustering

K-Means



- in each iteration, training example are assigned to the closest cluster centroid (shown by coloring the training examples with the same color as the cluster centroid to which is assigned)
- then each cluster centroid is moved to the mean of the points assigned to it

K-Means algorithm

- initialize cluster centroids $\mu_1, \mu_2, \dots, \mu_K \in \mathbb{R}^d$ randomly
- until convergence, repeat:
 - for each i , set **membership**

$$c^{(i)} = \operatorname{argmin}_j \|x^{(i)} - \mu_j\|^2$$

- for each j , set **centroids**

$$\mu_j = \frac{\sum_{i=1}^m 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^m 1\{c^{(i)} = j\}}$$

Exercise: K-Means on a small dataset

- objective: apply K-Means clustering to a small dataset
- data:
$$X = [(1, 1), (1, 2), (2, 1), (2, 2), (4, 4), (4, 5), (5, 4), (5, 5)]$$
- steps:
 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 convergence

Exercise: K-Means on a small dataset

- initial centroids:
 - centroid 1: (1, 1)
 - centroid 2: (1, 2)
- iteration 1:
 - 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)
- iteration 2:
 - no change in cluster assignment
 - convergence achieved
- final centroids:
 - centroid 1: (1.5, 1.5)
 - centroid 2: (4.5, 4.5)
- number of iterations: 2

收敛条件

Fuzzy C-Means clustering 允许数据点属于多个簇

- K means: each observation only belongs to one cluster
- fuzzy C -means: each observation may belong to two or more clusters
 - degree of **membership** of x_i in cluster c
- often used in pattern recognition and is an extension of the traditional K -means clustering

Fuzzy C-Means clustering

- given a dataset $X = \{x_1, x_2, \dots, x_n\}$, FCM partitions the data into C fuzzy clusters
- each data point x_i has a degree of **membership** $u_{ic} \in [0, 1]$ in each cluster c

Fuzzy C-Means (FCM) 聚类是一种基于原型的聚类算法，它类似于K均值聚类，但允许数据点属于多个簇，而不是仅属于一个簇。这使得FCM在某些情况下比K均值聚类更灵活，适用于某些数据分布不明显或模糊的情况。

FCM的主要思想是将每个数据点分配到每个簇的概率，而不是硬性地将其分配到一个特定的簇。具体而言，对于每个数据点，FCM会计算其与每个簇中心的距离，并根据距离计算每个簇的隶属度（membership degree）。隶属度表示数据点属于每个簇的概率，其中隶属度越大，表示数据点越可能属于该簇。

FCM的基本步骤如下：

1. **初始化隶属度**：随机初始化每个数据点对于每个簇的隶属度。
2. **计算簇中心**：根据当前隶属度计算每个簇的中心，即加权平均值。
3. **更新隶属度**：根据当前簇中心，重新计算每个数据点对于每个簇的隶属度。
4. **重复迭代**：重复步骤2和步骤3，直到隶属度不再变化或达到收敛条件。

与K均值聚类不同，FCM允许数据点根据其与簇中心的距离被分配到多个簇，因此可以更灵活地适应数据分布不均匀或模糊的情况。然而，FCM的计算复杂度较高，且对初始隶属度的选择敏感，需要仔细调参。

Fuzzy C-Means clustering

FCM objective function:

$$J(U, V) = \sum_{i=1}^n \sum_{c=1}^C u_{ic}^m \|x_i - v_c\|^2,$$

where

- $U = [u_{ic}]$ is the membership matrix Describing.
- $V = \{v_1, v_2, \dots, v_C\}$ are cluster centers
- m is a parameter that controls the level of cluster fuzziness
- $\|x_i - v_c\|$ is the Euclidean distance between the data point x_i and the cluster center v_c

Fuzzy C-Means clustering

- iteratively updates membership matrix U and cluster centers V until convergence
- **membership** update:

$$u_{ic}^m = \frac{1}{\sum_{k=1}^C \left(\frac{\|x_i - v_c\|}{\|x_i - v_k\|} \right)^{\frac{2}{m-1}}}$$

- **cluster center** update:

$$v_c = \frac{\sum_{i=1}^n u_{ic}^m x_i}{\sum_{i=1}^n u_{ic}^m}$$

- stops when changes in membership matrix between consecutive iterations are below a specified threshold

Fuzzy C-Means: example

- dataset with four points: A, B, C, and D
- cluster these points into $C = 2$ clusters using Fuzzy C-means

- membership values:

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

Fuzzy C-Means: example

- **point A** has a high membership value (0.8) in cluster 1 and a low membership value (0.2) in cluster 2, i.e. 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:

- good for overlapping data
- fuzzy membership

Disadvantages:

- number of clusters should pre-specified
- Euclidean distance may unequally weight underlying factors