KIOS
Research and Innovation Center of Excellence

MSc on Intelligent Critical Infrastructure Systems

Machine Learning
Lecture 9

Christos Kyrkou
Research Lecturer
KIOS Research and Innovation Center of Excellence
University of Cyprus

Imperial College London

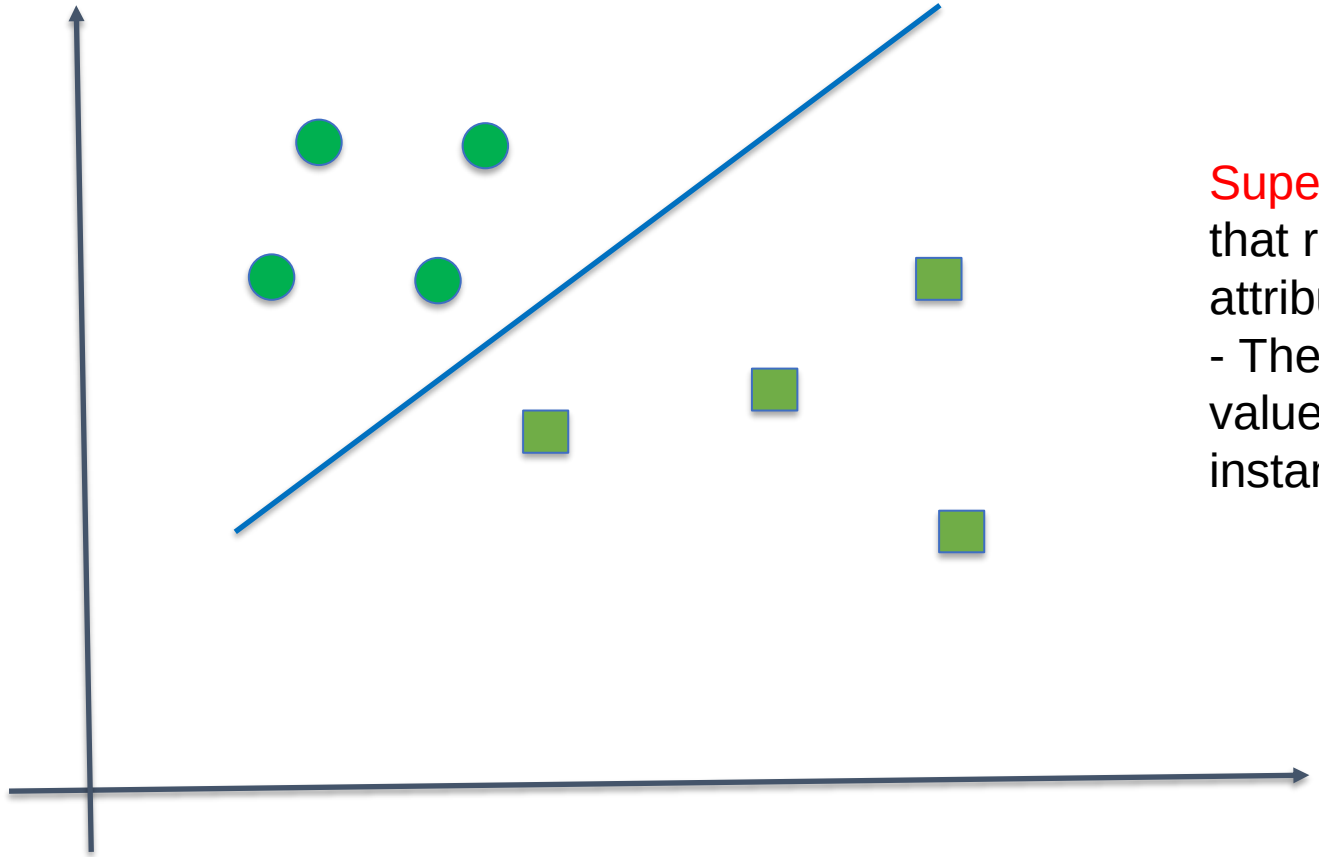
University of Cyprus

funded by:

Outline

- Unsupervised Learning
  - Clustering
    - K-means Algorithm
  - Dimensionality Reduction
    - Principal Component Analysis (PCA)

Supervised Learning

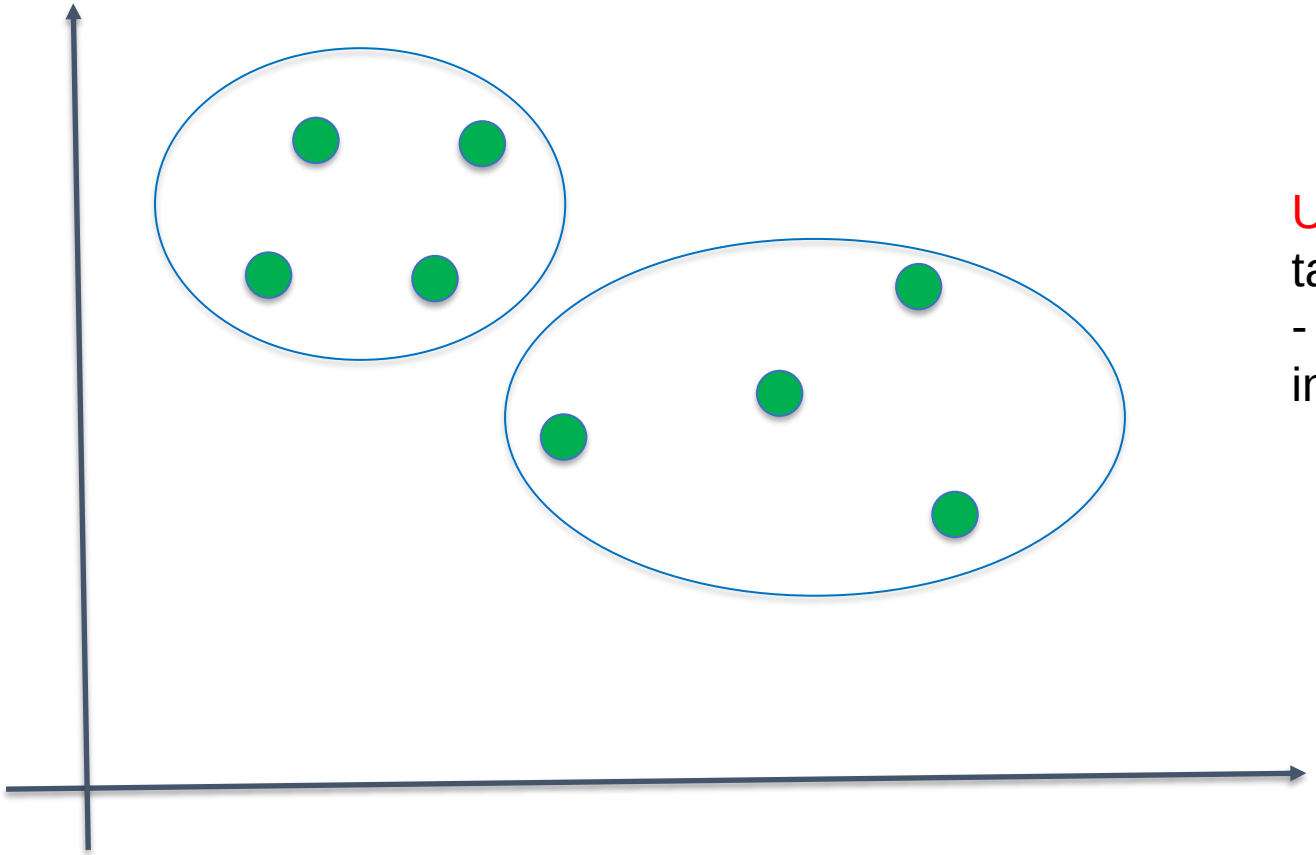


Supervised learning: discover patterns in the data that relate data attributes with a target (class) attribute.

- These patterns are then utilized to predict the values of the target attribute in future data instances.

Training set: $\{(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots\ldots (x_N, y_N)\}$

Unsupervised Learning



Unsupervised learning: The data have no target attribute.
- We want to explore the data to find some intrinsic structures in them.

Training set: $\{x_1, x_2, x_3, \ldots x_N\}$

Unsupervised Learning (UL) Applications

- Methods
  - Clustering
  - Dimensionality Reduction
  - Many more
- Applications
  - Marketing and sales (personalization and market targeting)
  - Identifying fake news
  - Social network analysis
  - Classifying network traffic
  - Fraud detection

What is UL for?

- Let us see some real-life examples
- Example 1: groups people of similar sizes together to make “small”, “medium” and “large” T-Shirts.
  - Tailor-made for each person: too expensive
  - One-size-fits-all: does not fit all.
- Example 2: In marketing, segment customers according to their similarities
  - To do targeted marketing.

What is UL for? (cont…)

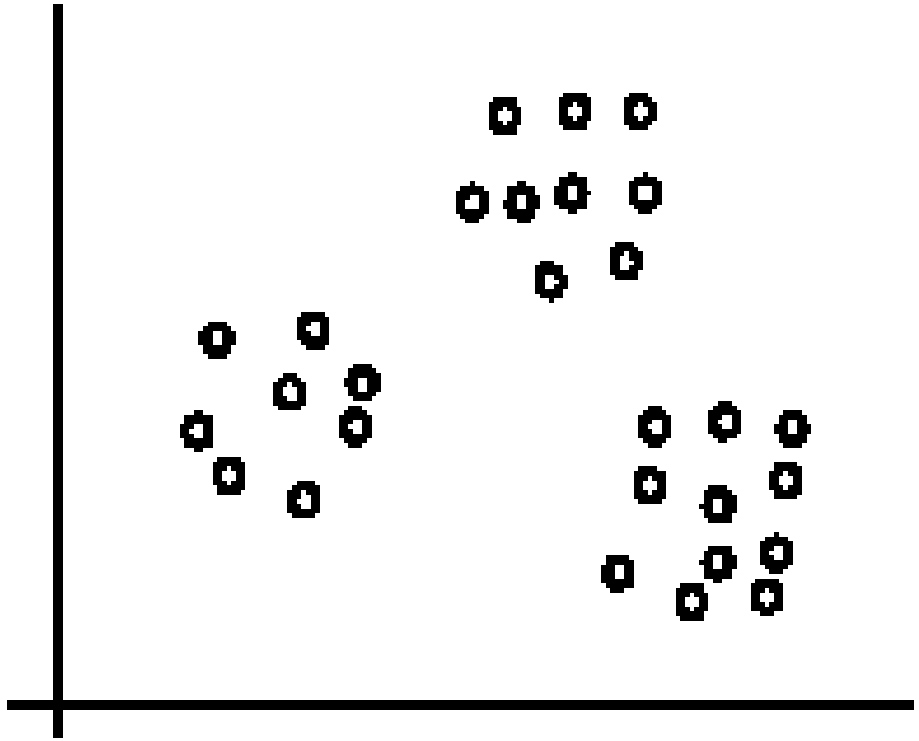
- Example 3: Given a collection of text documents, we want to organize them according to their content similarities,
  - To produce a topic hierarchy
- In fact, clustering is one of the most utilized data mining techniques.
  - It has a long history, and used in almost every field, e.g., medicine, psychology, botany, sociology, biology, archeology, marketing, insurance, libraries, etc.
  - In recent years, due to the rapid increase of online documents, text clustering becomes important.

Clustering

- Clustering is a technique for finding similarity groups in data, called **clusters**. I.e.,
  - it groups data instances that are similar to (near) each other in one cluster and data instances that are very different (far away) from each other into different clusters.
- Clustering is often called an **unsupervised learning** task as no class values denoting an *a priori* grouping of the data instances are given, which is the case in supervised learning.
- Due to historical reasons, clustering is often considered synonymous with unsupervised learning.
  - In fact, many other tasks are also unsupervised

An illustration

- The data set has three natural groups of data points, i.e., 3 natural clusters.



Aspects of clustering

- A clustering algorithm
  - Partitional clustering
  - Hierarchical clustering
  - …
- A distance (similarity, or dissimilarity) function
- Clustering quality
  - Inter-clusters distance $\Rightarrow$ maximized
  - Intra-clusters distance $\Rightarrow$ minimized
- The quality of a clustering result depends on the algorithm, the distance function, and the application.

Distance functions

- Key to clustering. “similarity” and “dissimilarity” can also commonly used terms.
- There are numerous distance functions for
  - Different types of data
    - Numeric data
    - Nominal data
  - Different specific applications

Distance functions for numeric attributes

- Most commonly used functions are
  - Euclidean distance and
  - Manhattan (city block) distance
- We denote distance with: *dist*($\mathbf{x}_i$, $\mathbf{x}_j$), where $\mathbf{x}_i$ and $\mathbf{x}_j$ are data points (vectors)
- They are special cases of Minkowski distance. h is positive integer.

$$dist(\mathrm{x_i}, \mathrm{x_j}) = ((x_{i1} - x_{j1})^h + (x_{i2} - x_{j2})^h + \ldots + (x_{ir} - x_{jr})^h)^{\frac{1}{h}}$$

Euclidean distance and Manhattan distance

- If h = 2, it is the Euclidean distance

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{ir} - x_{jr})^2}$$

- If h = 1, it is the Manhattan distance

$$dist(\mathbf{x}_i, \mathbf{x}_j) = | x_{i1} - x_{j1} | + | x_{i2} - x_{j2} | + \ldots + | x_{ir} - x_{jr} |$$

- Weighted Euclidean distance

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{w_1(x_{i1} - x_{j1})^2 + w_2(x_{i2} - x_{j2})^2 + \ldots + w_r(x_{ir} - x_{jr})^2}$$

Squared distance and Chebychev distance

- Squared Euclidean distance: to place progressively greater weight on data points that are further apart.

$$dist(\mathbf{x}_i, \mathbf{x}_j) = (x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{ir} - x_{jr})^2$$

- Chebychev distance: one wants to define two data points as "different" if they are different on any one of the attributes.

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \max(|x_{i1} - x_{j1}|, |x_{i2} - x_{j2}|, \ldots, |x_{ir} - x_{jr}|)$$

Clustering: K-means algorithm

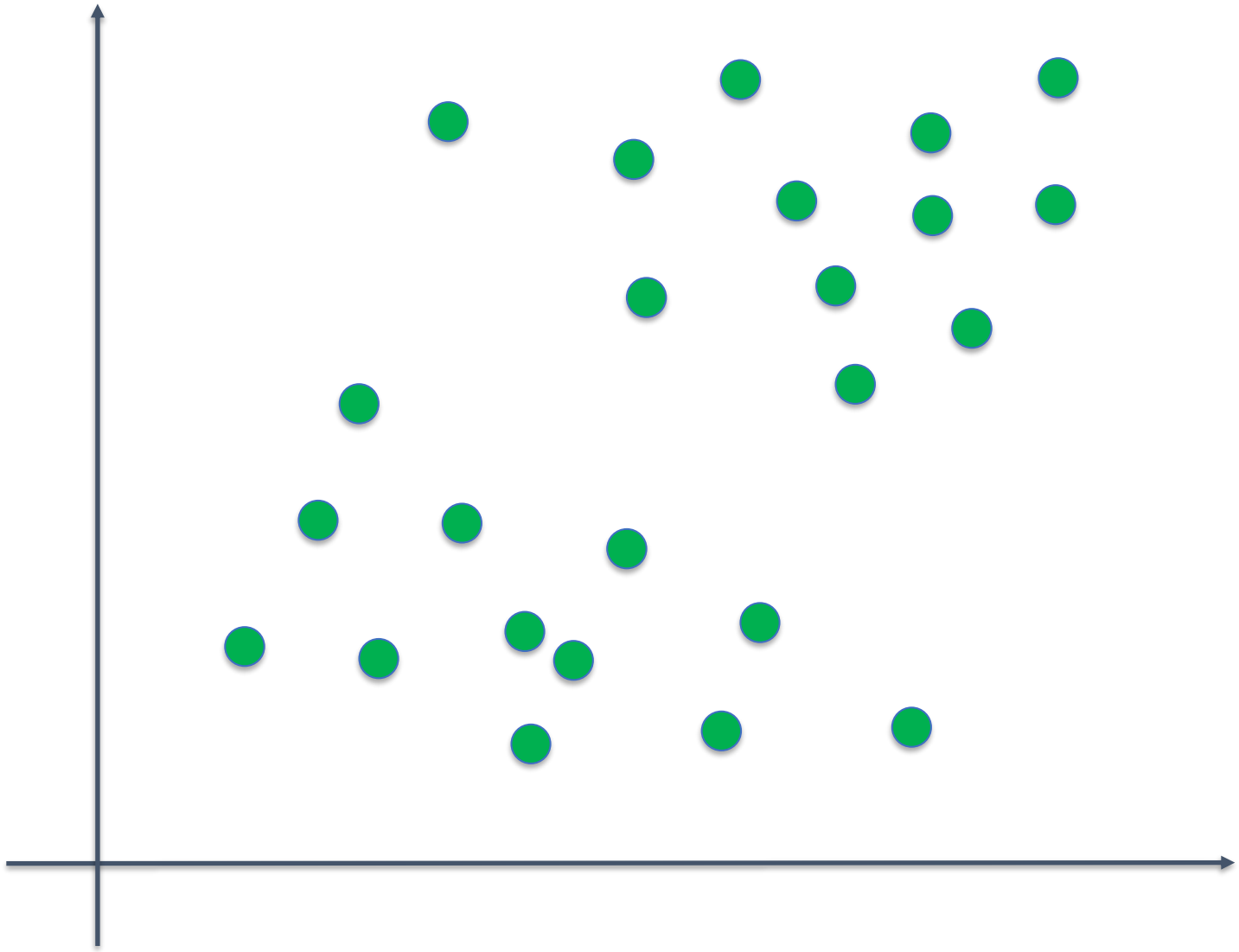
INPUT:

- K – number of clusters ← hyperparameter
- Initial locations of cluster centroids $\{\mu_1^0, \mu_2^0, \mu_3^0, \dots \mu_K^0\} \mu_i^0 \in \mathbb{R}^n$
- Training set: $\{x_1, x_2, x_3, \dots x_N\} x_i \in \mathbb{R}^n$

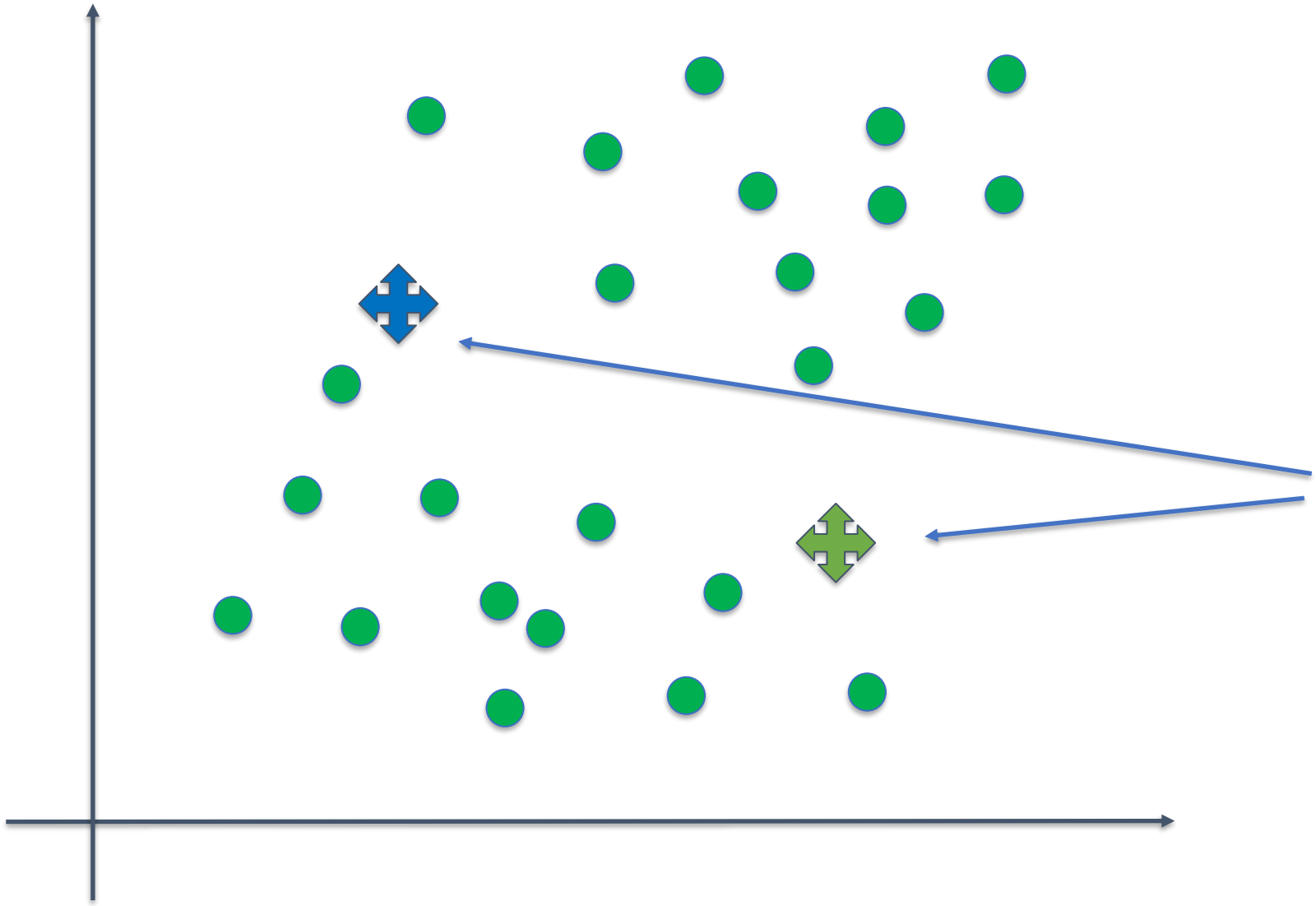
OUTPUT:

- Final locations of cluster centroids $\{\mu_1^*, \mu_2^*, \mu_3^*, \dots \mu_K^*\} \mu_i^* \in \mathbb{R}^n$

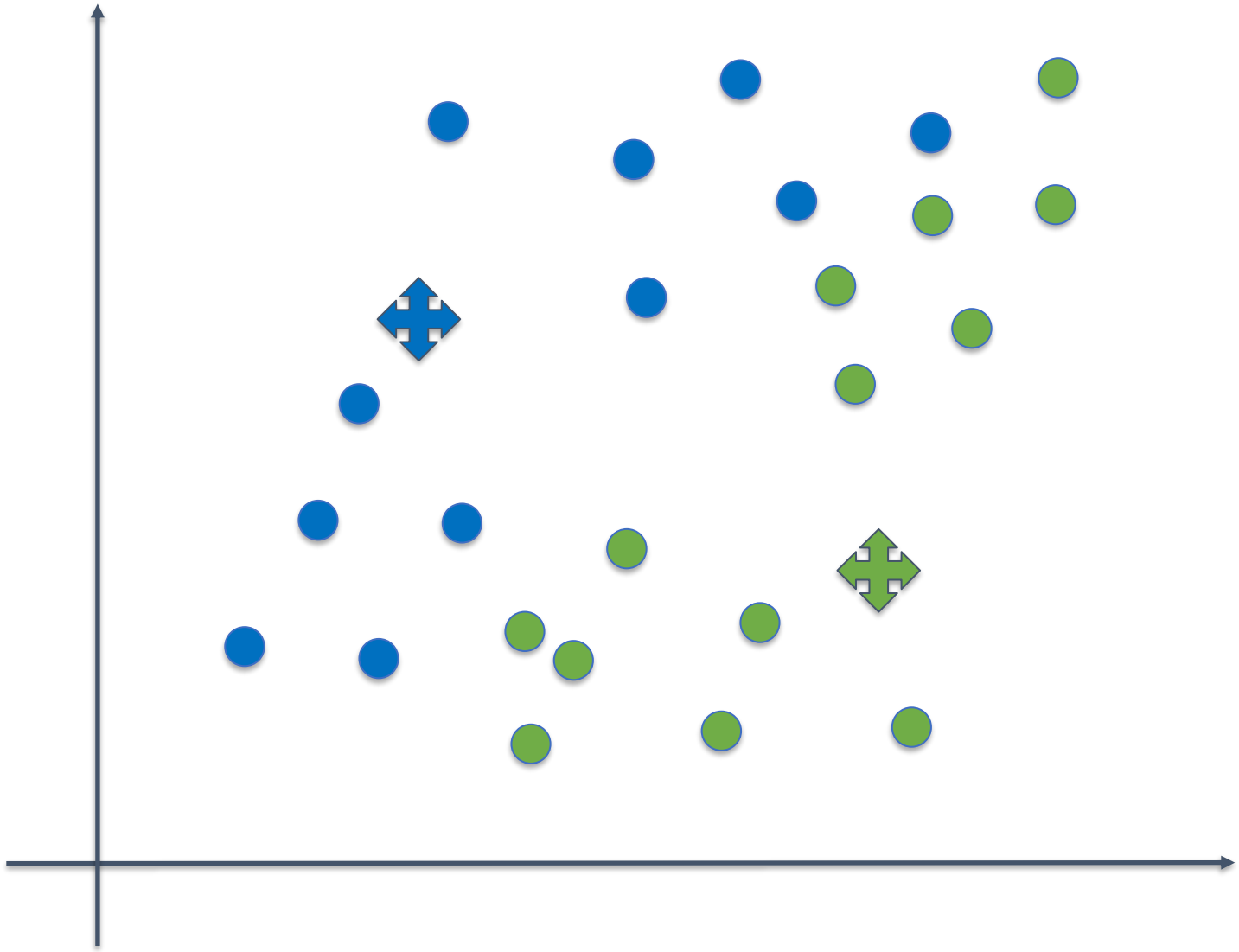
Clustering: K-means algorithm



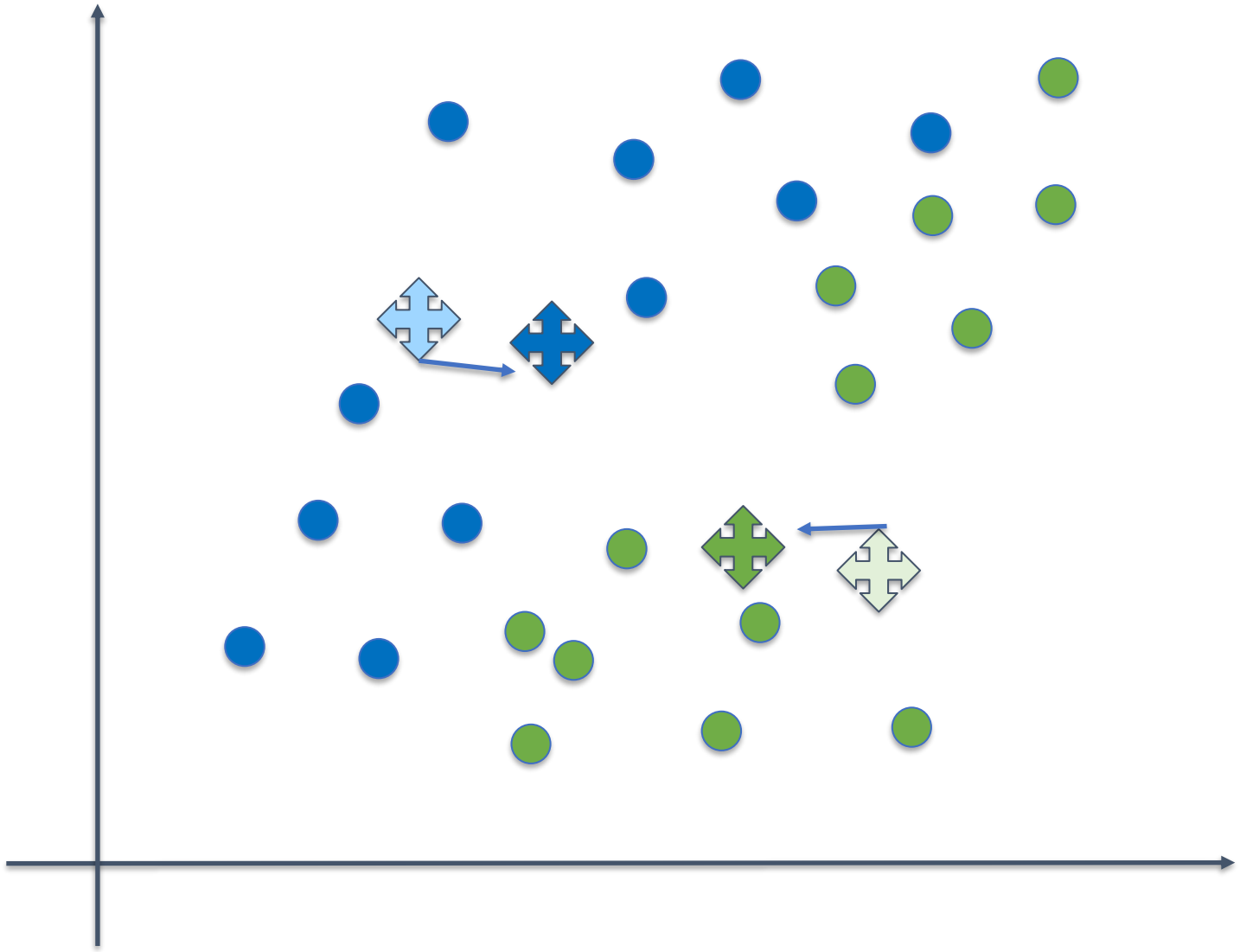
Clustering: K-means algorithm



Clustering: K-means algorithm

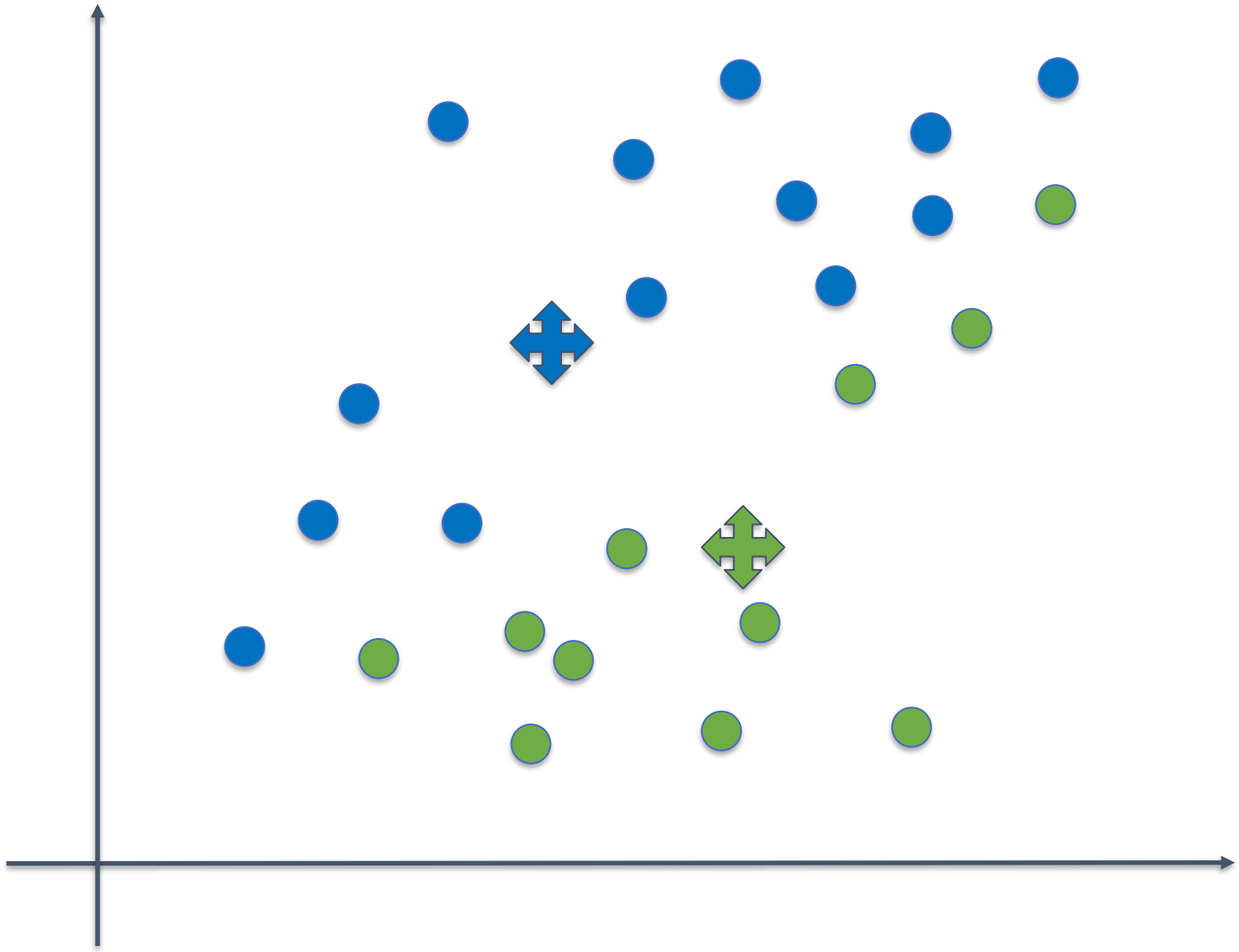


Clustering: K-means algorithm

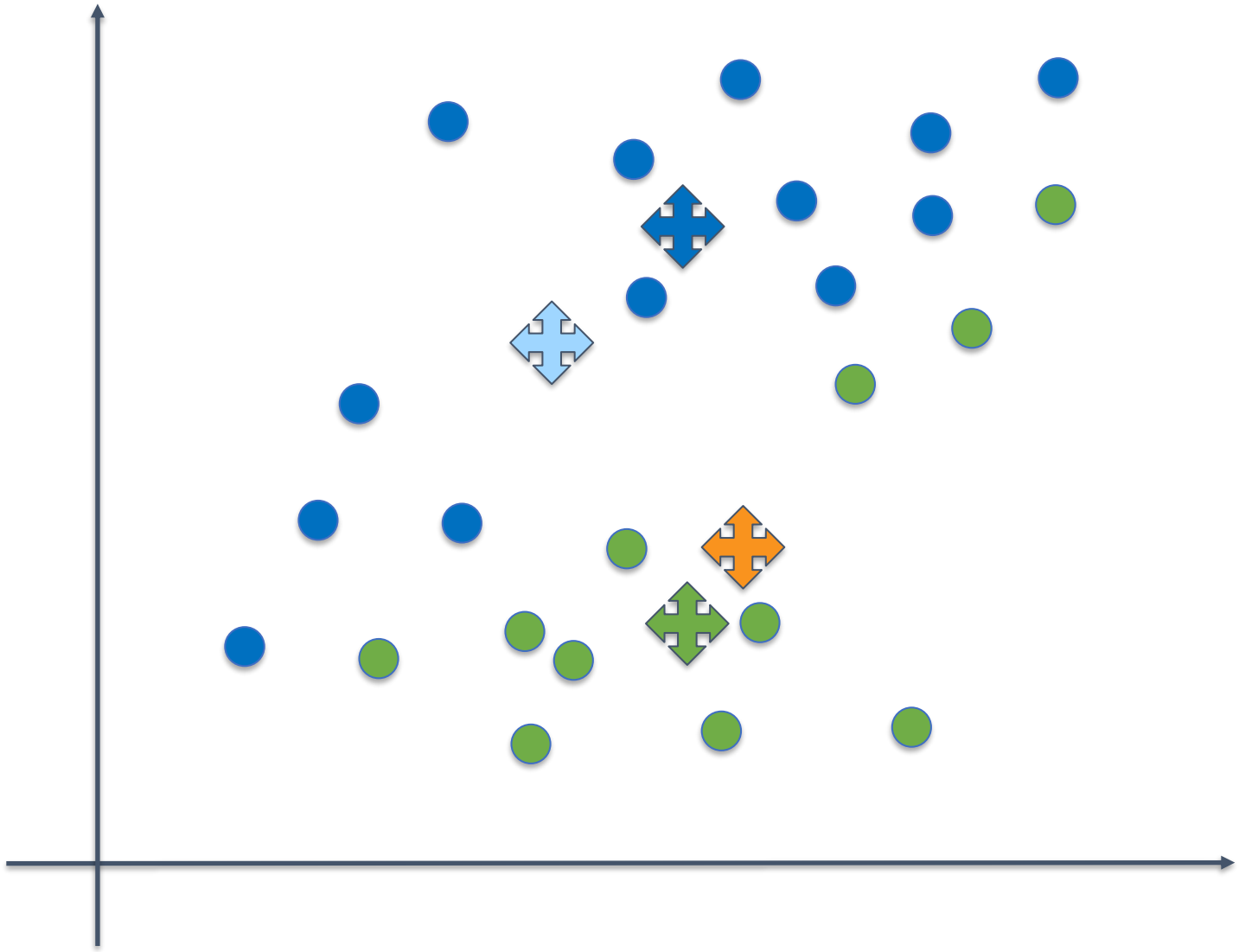


New cluster centroids

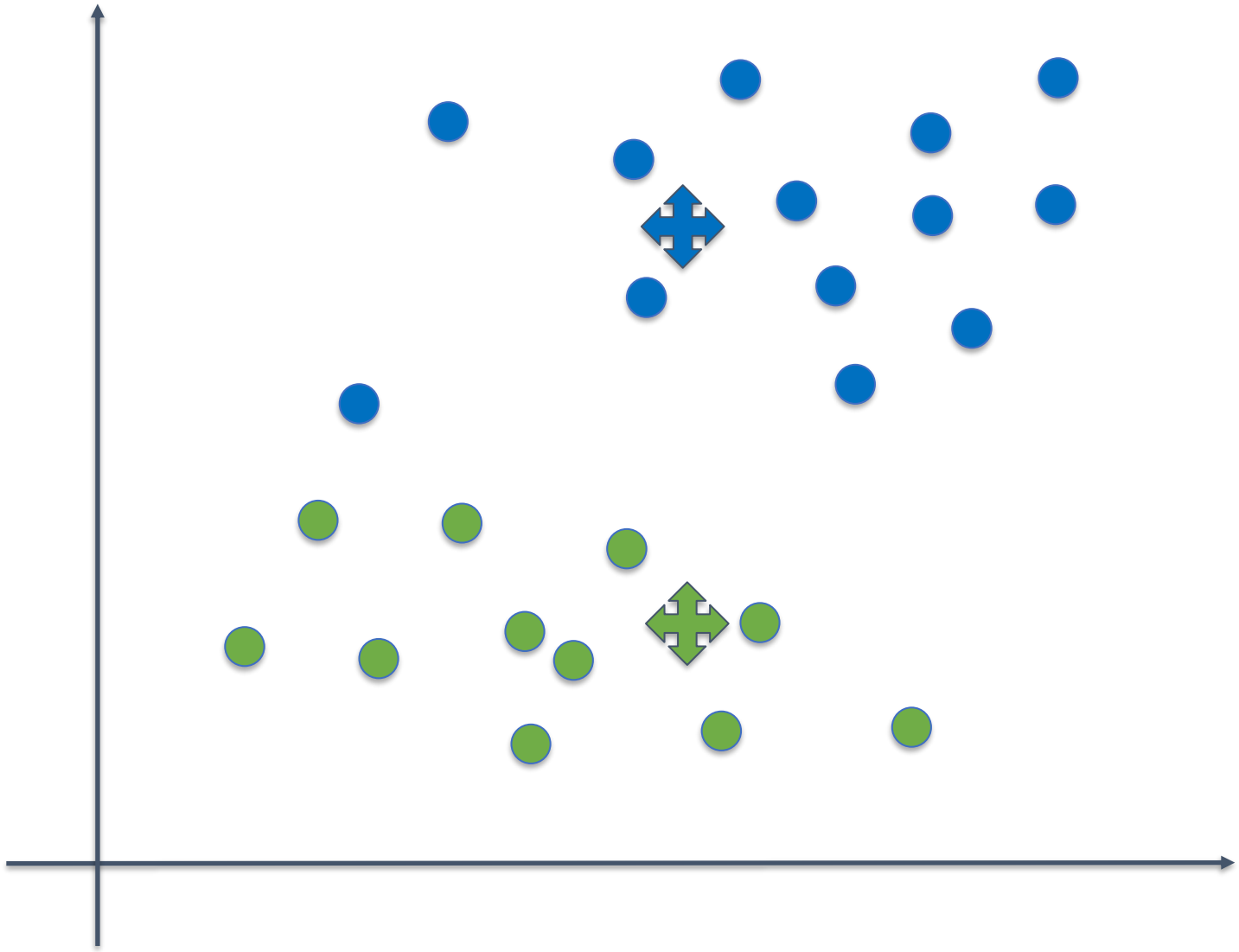
Clustering: K-means algorithm



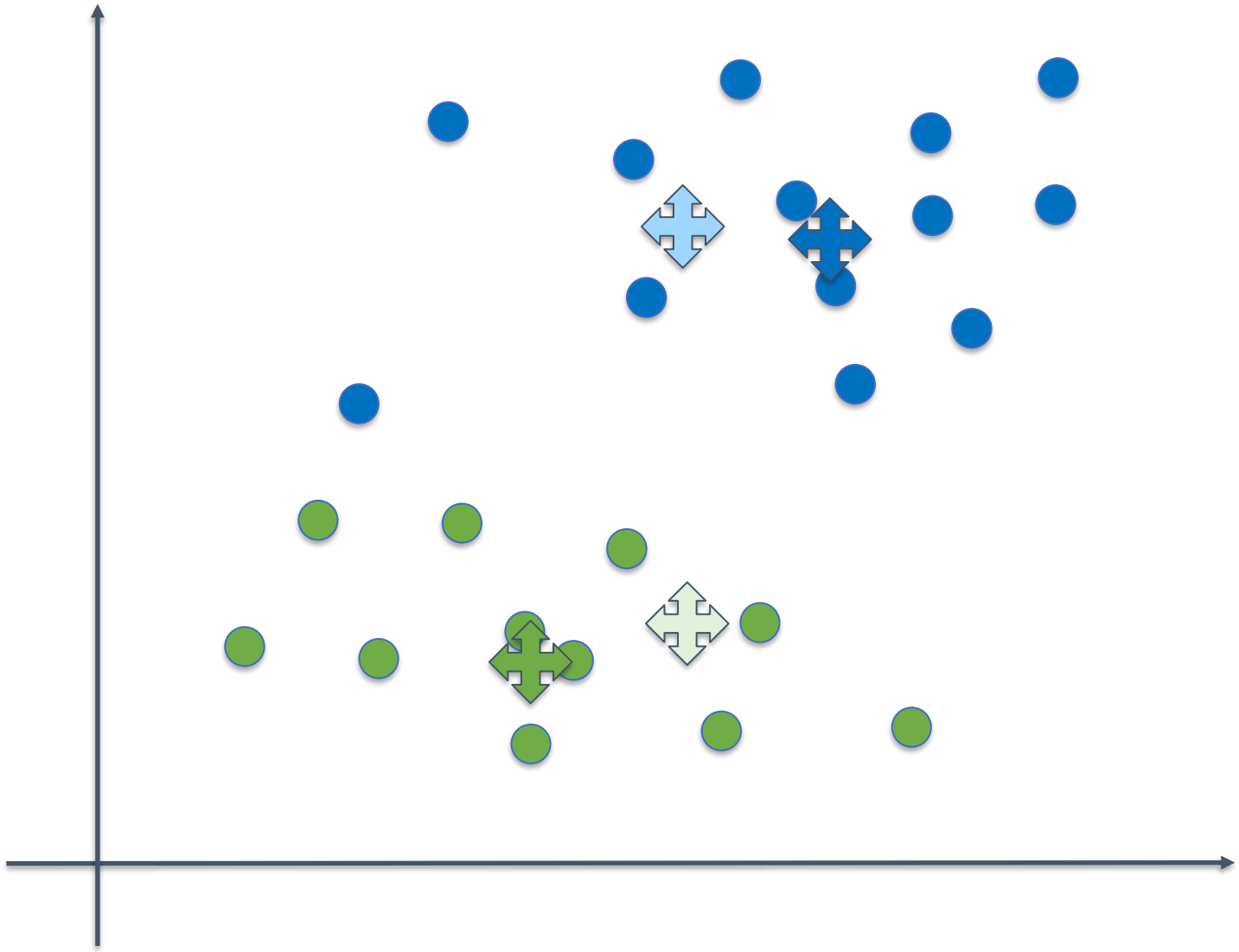
Clustering: K-means algorithm



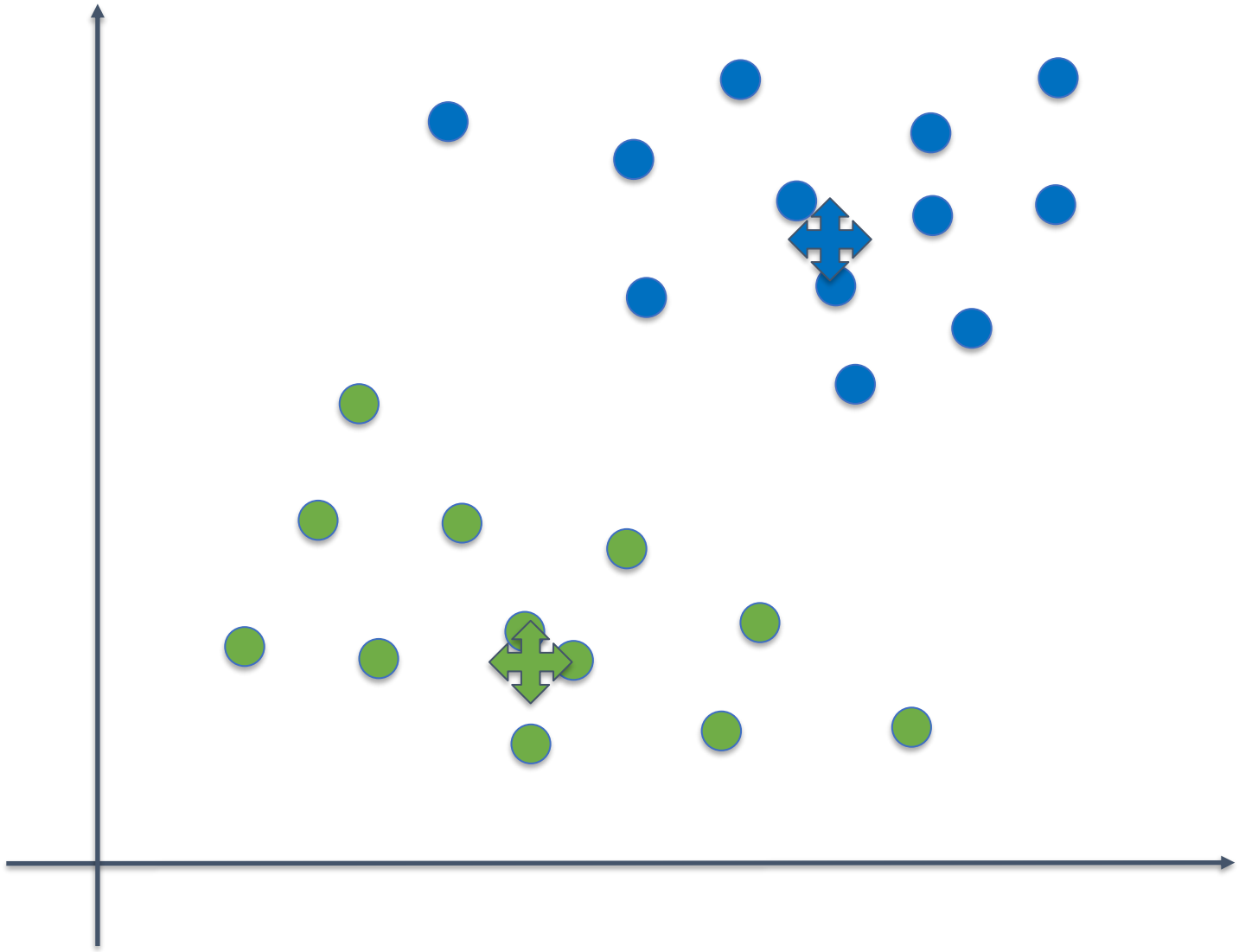
Clustering: K-means algorithm



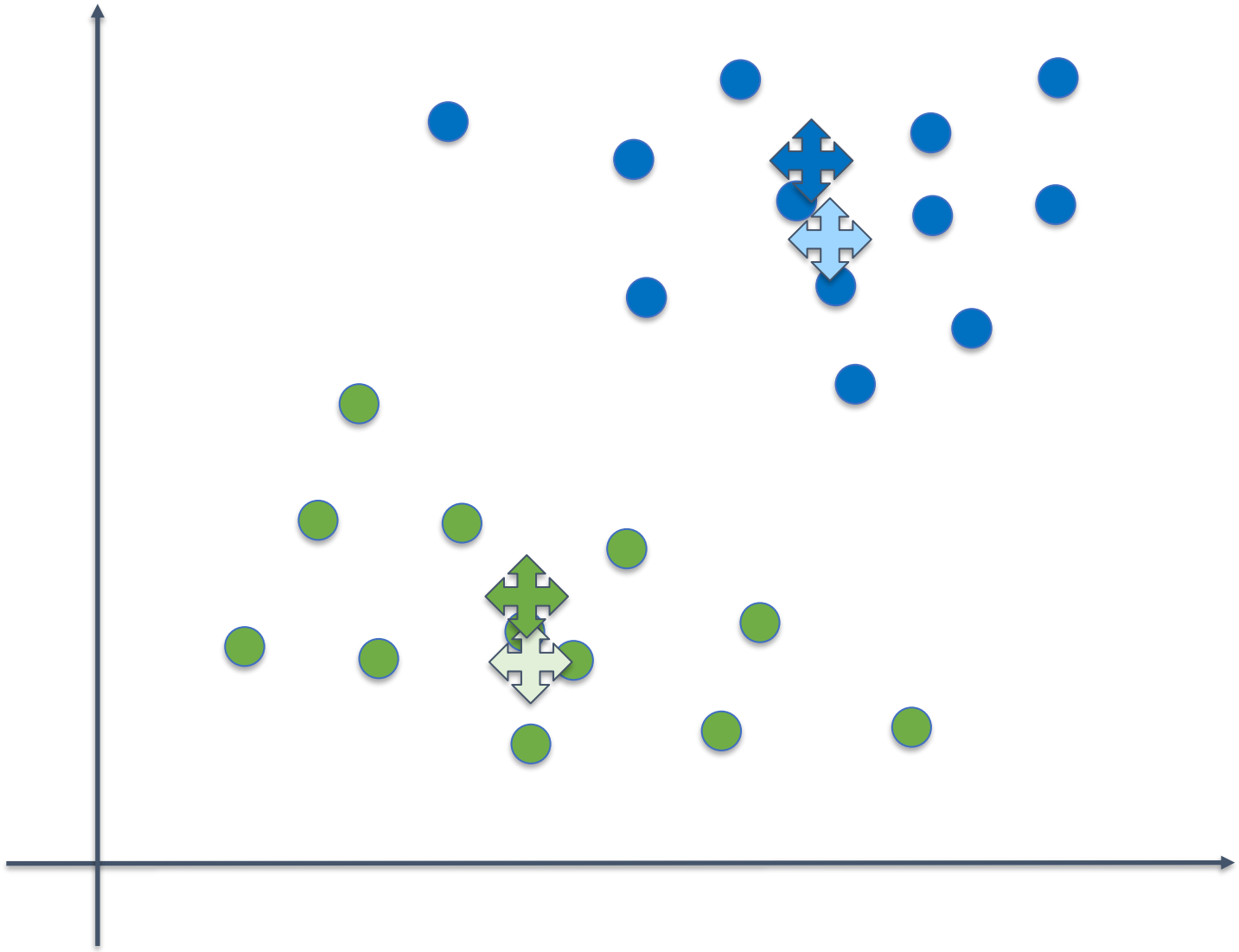
Clustering: K-means algorithm



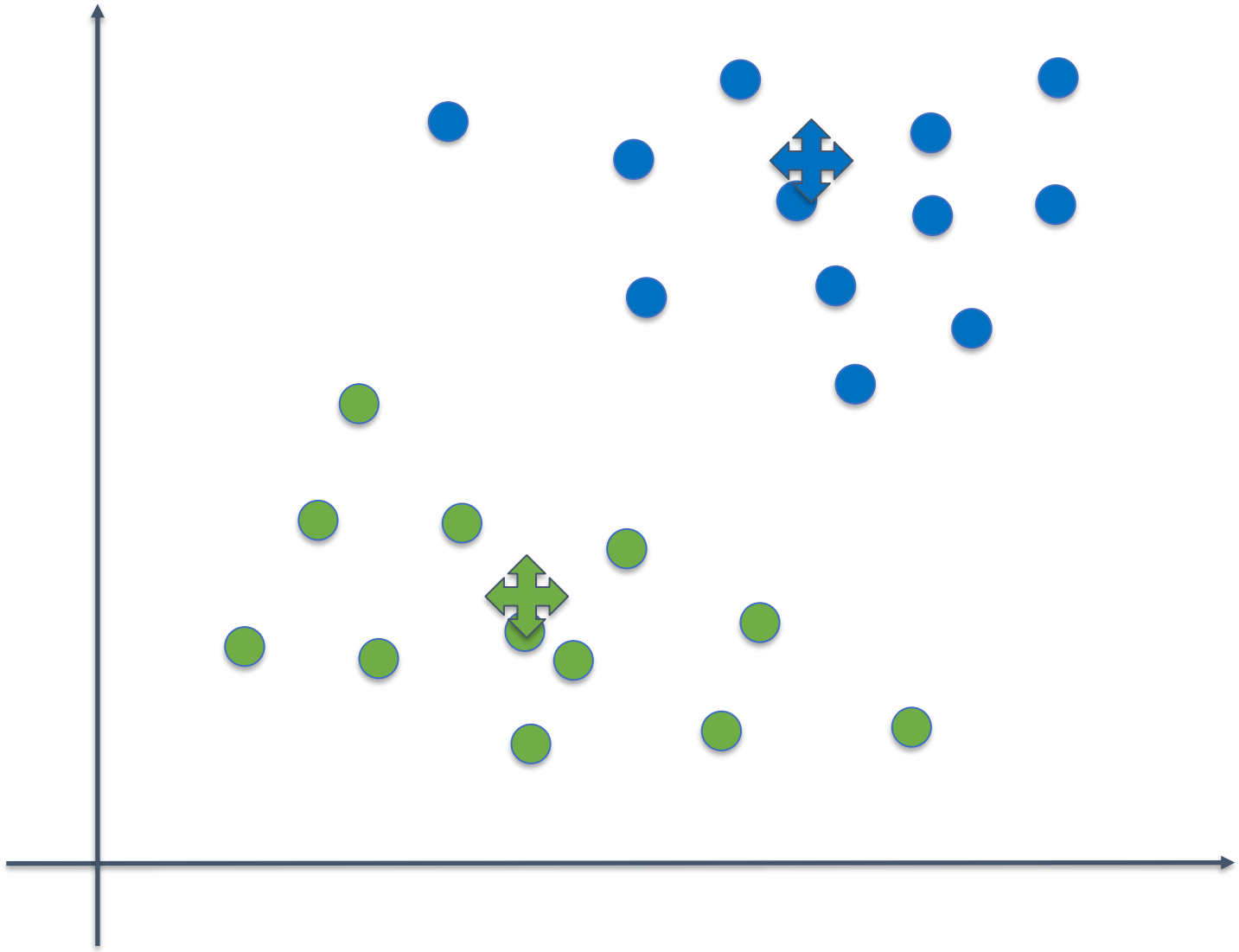
Clustering: K-means algorithm



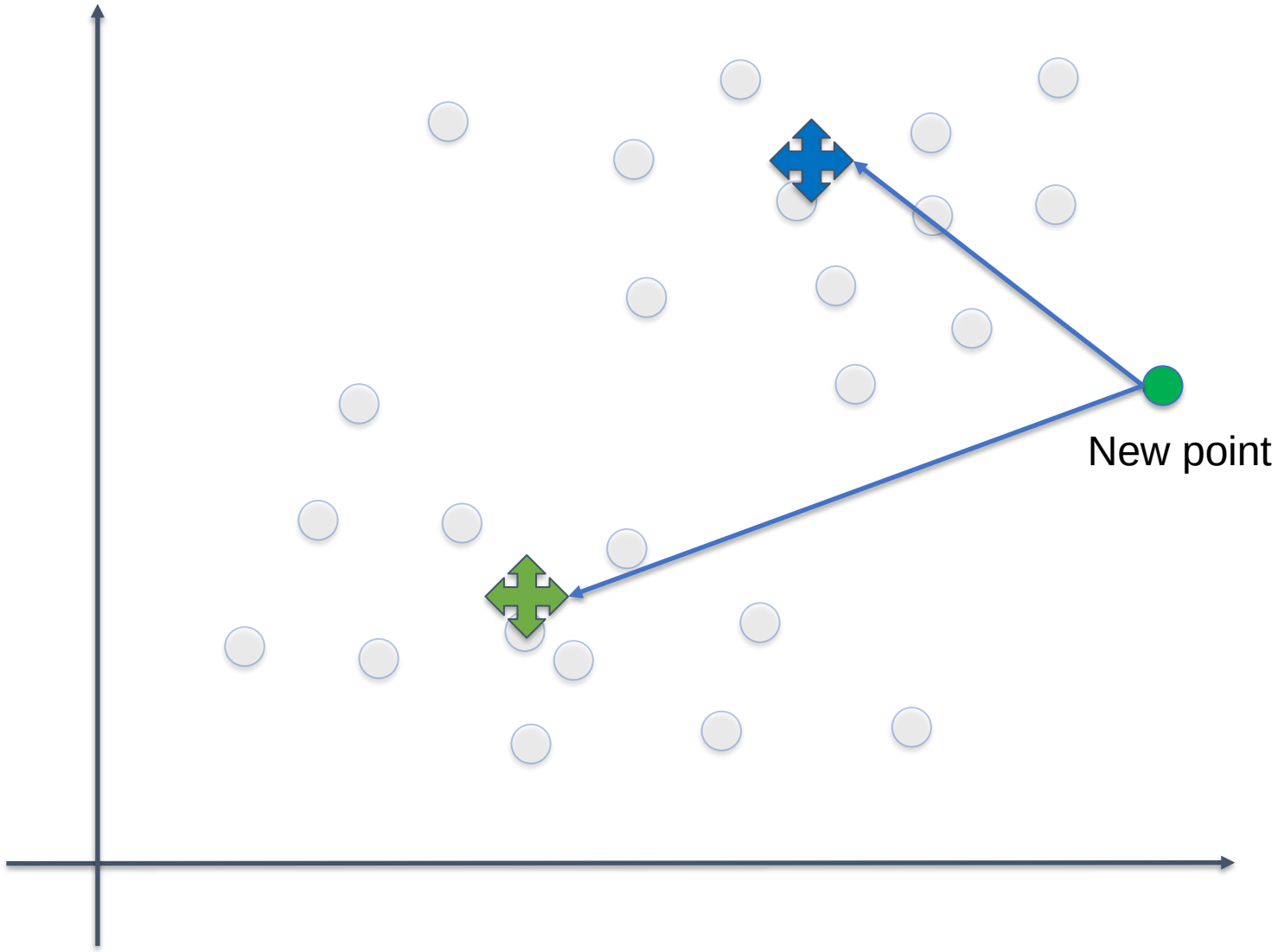
Clustering: K-means algorithm



Clustering: K-means algorithm



When we have a new data point



Assign to cluster with minimum distance

Clustering: K-means algorithm

Randomly initialize K cluster centroids $\mu_1^0, \mu_2^0, \mu_3^0, \ldots \mu_K^0 \in \mathbb{R}^n$

Repeat{

for i = 1 to N

$c^{(i)} \in \{1, 2, \ldots K\}$:= index of cluster centroid closest to x_i

for k = 1 to K

μ_k := average (mean) of points assigned to cluster k

}

Clustering: K-means optimization objective

$c^{(i)} \in \{1, 2, \ldots K\} :=$ index of cluster to which x_i is currently assigned

$\mu_k :=$ cluster centroid k

$\mu_{c^{(i)}} :=$ centroid of cluster to which example x_i has been assigned

OPTIMIZATION OBJECTIVE:

$$J(c^{(1)}, c^{(2)}, \ldots, c^{(N)}, \mu_1, \mu_2, \ldots, \mu_K) = \frac{1}{N} \sum_{i=1}^{N} \left\| x_i - \mu_{c^{(i)}} \right\|^2$$

$$\min_{\substack{c^{(1)},\ldots,c^{(N)} \\ \mu_1,\ldots,\mu_K}} J(c^{(1)}, c^{(2)}, \ldots, c^{(N)}, \mu_1, \mu_2, \ldots, \mu_K)$$

distortion

Clustering: K-means algorithm

Randomly initialize K cluster centroids $\mu_1^0, \mu_2^0, \mu_3^0, \ldots \mu_K^0 \in \mathbb{R}^n$

Repeat{

for i = 1 to N

$c^{(i)} \in \{1, 2, \ldots K\} :=$ index of cluster centroid closest to x_i

for k = 1 to K

$\mu_k :=$ average (mean) of points assigned to cluster k

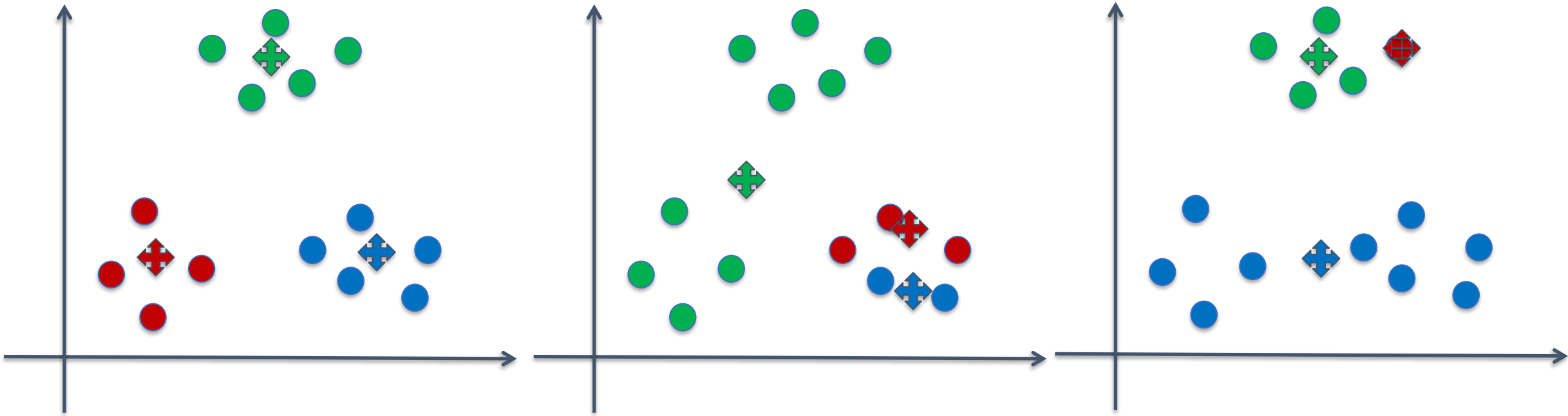
}

$J(c^{(1)}, c^{(2)}, \ldots, c^{(N)}, \mu_1, \mu_2, \ldots, \mu_K)$

Clustering: K-means algorithm initialization

- Randomly pick K training points and set $\mu_1^0, \mu_2^0, \mu_3^0, \ldots \mu_K^0$ equal to these points
- The initialization of the cluster centroids sometimes affects the final result (two runs of the K-means Algorithm may result in two different models)
- Some variants of the K-means Algorithm compute the initial positions of the centroids based on some properties of the dataset.

Clustering: K-means algorithm – local minima



$$J(c^{(1)}, c^{(2)}, \ldots, c^{(N)}, \mu_1, \mu_2, \ldots, \mu_K) = \frac{1}{N} \sum_{i=1}^{N} \left\| x_i - \mu_{c^{(i)}} \right\|^2$$

K-means Algorithm – Random Initialization

For j = 1 to 100 {

- Randomly initialize K-means (Number of clusters K)
- Run K-means. Obtain: $c^{(1)}, \ldots, c^{(N)}, \mu_1, \ldots, \mu_K$
- Computer cost function (distortion)

$$J(c^{(1)}, c^{(2)}, \ldots, c^{(N)}, \mu_1, \mu_2, \ldots, \mu_K)$$

}

Pick clustering solution that gave the lowest distortion

$$J(c^{(1)}, c^{(2)}, \ldots, c^{(N)}, \mu_1, \mu_2, \ldots, \mu_K)$$

K-means Algorithm – choosing the value of K

- Choosing the right number of clusters K is a difficult task. There are several methods, but none is optimal.
- One approach is to break up the dataset into training and test data and to use the concept of prediction strength to determine a suitable value of K.
- The cost function (distortion) can be used to find the relationship between the number of clusters and the distortion. However, as the number of clusters increases, the distortion may decrease, but there are trade-offs.

Strengths of k-means

- Strengths:
  - Simple: easy to understand and to implement
  - Efficient: Time complexity: $O(tKN)$,

    where N is the number of data points,

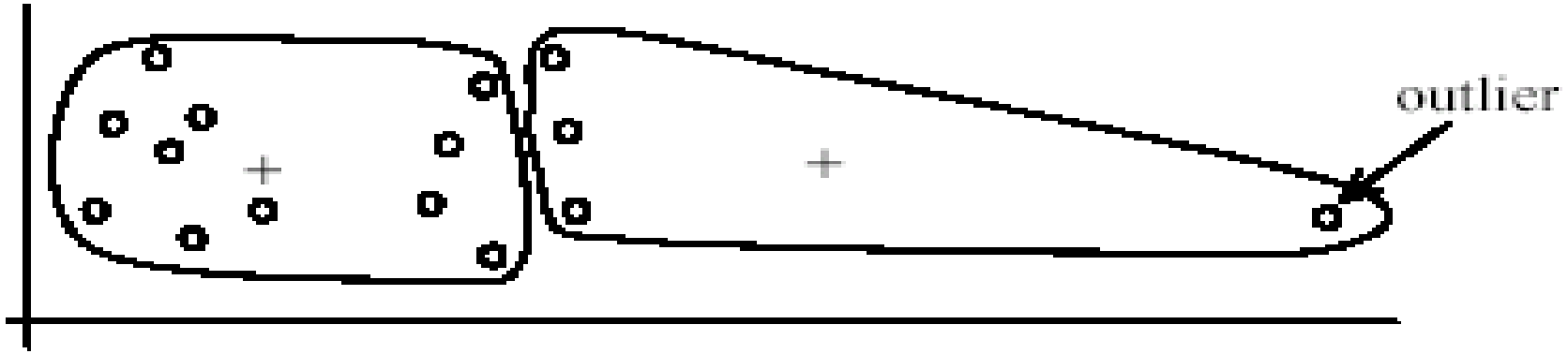
    K is the number of clusters, and

    t is the number of iterations.
  - Since both K and t are small. k-means is considered a linear algorithm.
- K-means is the most popular clustering algorithm.
- Note that: it terminates at a local optimum if sum of squared errors is used. The global optimum is hard to find due to complexity.

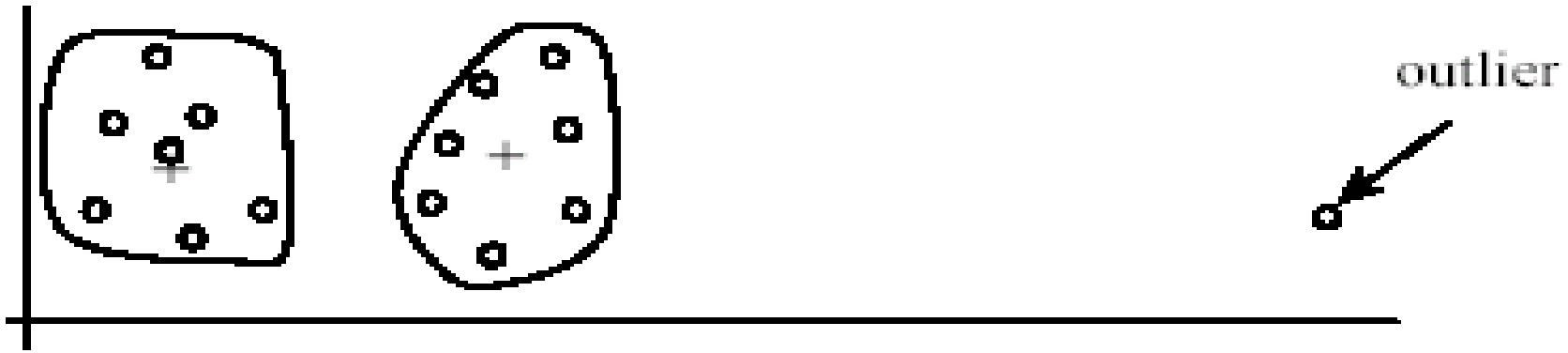
Weaknesses of k-means

- The algorithm is only applicable if the mean is defined.
  - For categorical data, *K*-mode - the centroid is represented by most frequent values.
- The user needs to specify *K*.
- The algorithm is sensitive to **outliers**
  - Outliers are data points that are very far away from other data points.
  - Outliers could be errors in the data recording or some special data points with very different values.

Weaknesses of k-means: Problems with outliers



(A): Undesirable clusters



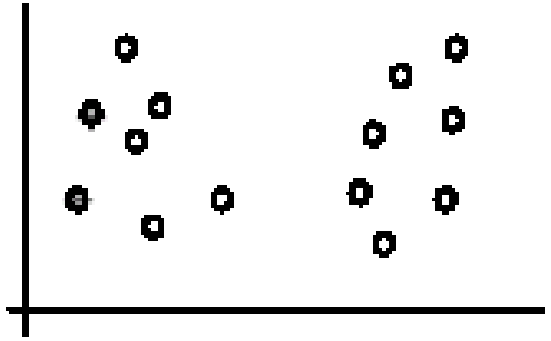
(B): Ideal clusters

Weaknesses of k-means: To deal with outliers

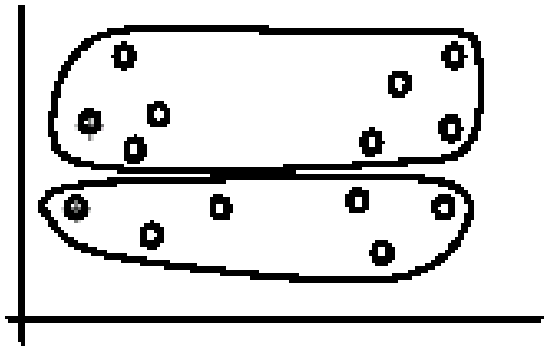
- One method is to remove some data points in the clustering process that are much further away from the centroids than other data points.
  - To be safe, we may want to monitor these possible outliers over a few iterations and then decide to remove them.
- Another method is to perform random sampling. Since in sampling we only choose a small subset of the data points, the chance of selecting an outlier is very small.
  - Assign the rest of the data points to the clusters by distance or similarity comparison, or classification

Weaknesses of k-means (cont …)

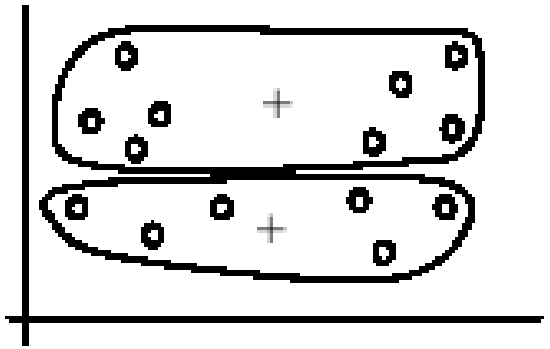
- The algorithm is sensitive to initial seeds.



(A). Random selection of seeds (centroids)



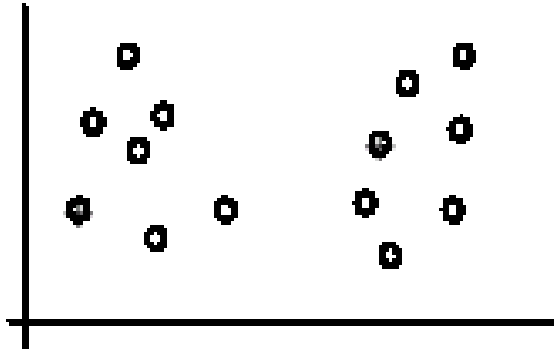
(B). Iteration 1



(C). Iteration 2

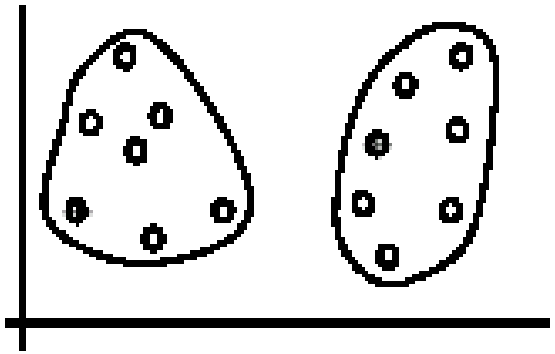
Weaknesses of k-means (cont …)

- If we use different seeds: good results

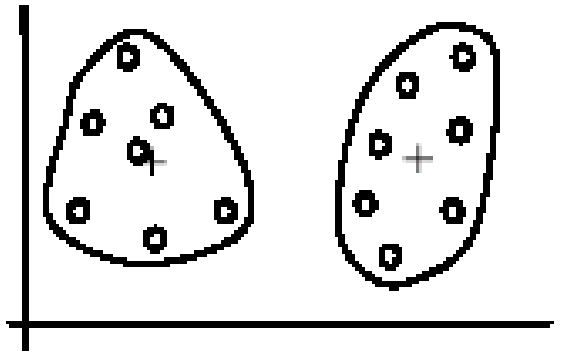


(A). Random selection of k seeds (centroids)

There are some methods to help choose good seeds



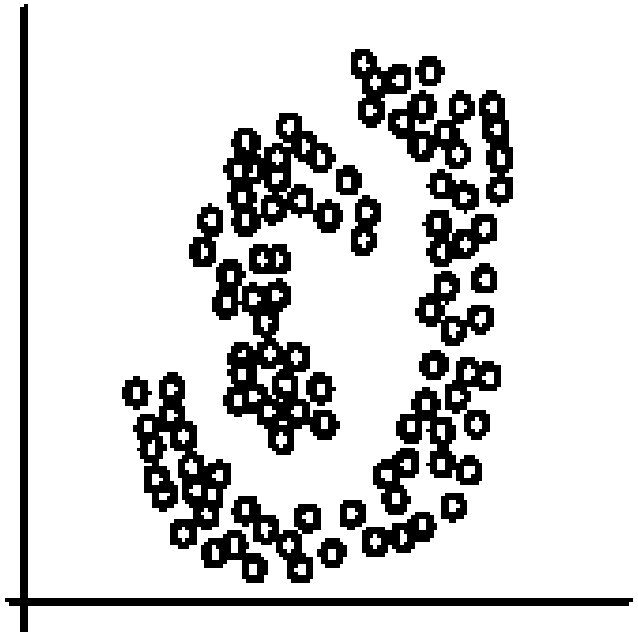
(B). Iteration 1



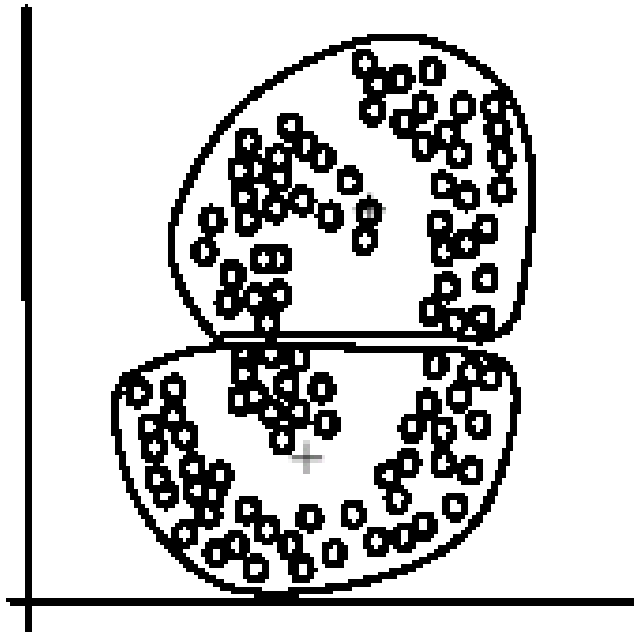
(C). Iteration 2

Weaknesses of k-means (cont …)

- The *k*-means algorithm is not suitable for discovering clusters that are not hyper-ellipsoids (or hyper-spheres).



(A): Two natural clusters



(B): *k*-means clusters

K-means summary

- Despite weaknesses, K-means is still the most popular algorithm due to its simplicity, efficiency
  - other clustering algorithms have their own lists of weaknesses.
- No clear evidence that any other clustering algorithm performs better in general
  - although they may be more suitable for some specific types of data or applications.
- Comparing different clustering algorithms is a difficult task. No one knows the correct clusters!

Other Clustering Algorithms

- **DBSCAN** (density-based spatial clustering of applications with noise). It is density based, as opposed to centroid-based (K-means). The advantage of DBSCAN is that it build clusters of arbitrary shape, while centroid-based algorithms create clusters that have the shape of a hypersphere. However, it has two hyperparameters to be selected.
- **HDBSCAN** (hierarchical DBSCAN). This is an advanced version of DBSCAN, with only one hyperparameter to be selected.
- **Gaussian Mixture Model** (GMM). An example (input point) may be a member of several clusters with different membership score.
- **Fuzzy Clustering**. Referred to as soft clustering or soft K-means, as compared to hard clustering (DBSCAN and K-means).

Dimensionality Reduction – Motivation

- Data compression
- Data visualization
- Interpretability of learning procedure

- Working in high-dimensional spaces can be undesirable:
  - raw data are often sparse as a consequence of the curse of dimensionality,
  - analyzing the data is usually computationally intractable (hard to control or deal with).
    - Dimensionality reduction is common in fields that deal with large numbers of observations and/or large numbers of variables

Dimensionality Reduction – Examples

- Principal Component Analysis (PCA)
  - Kernel PCA
- Linear discriminant analysis (LDA)
  - Find a linear combination of features that characterizes or separates two or more classes of objects or events.
- t-distributed Stochastic Neighbour Embedding (t-SNE)
  - a nonlinear dimensionality reduction technique useful for visualization of high-dimensional datasets.
- Autoencoders
  - Learn nonlinear dimension reduction functions and codings

Principal Component Analysis (PCA)

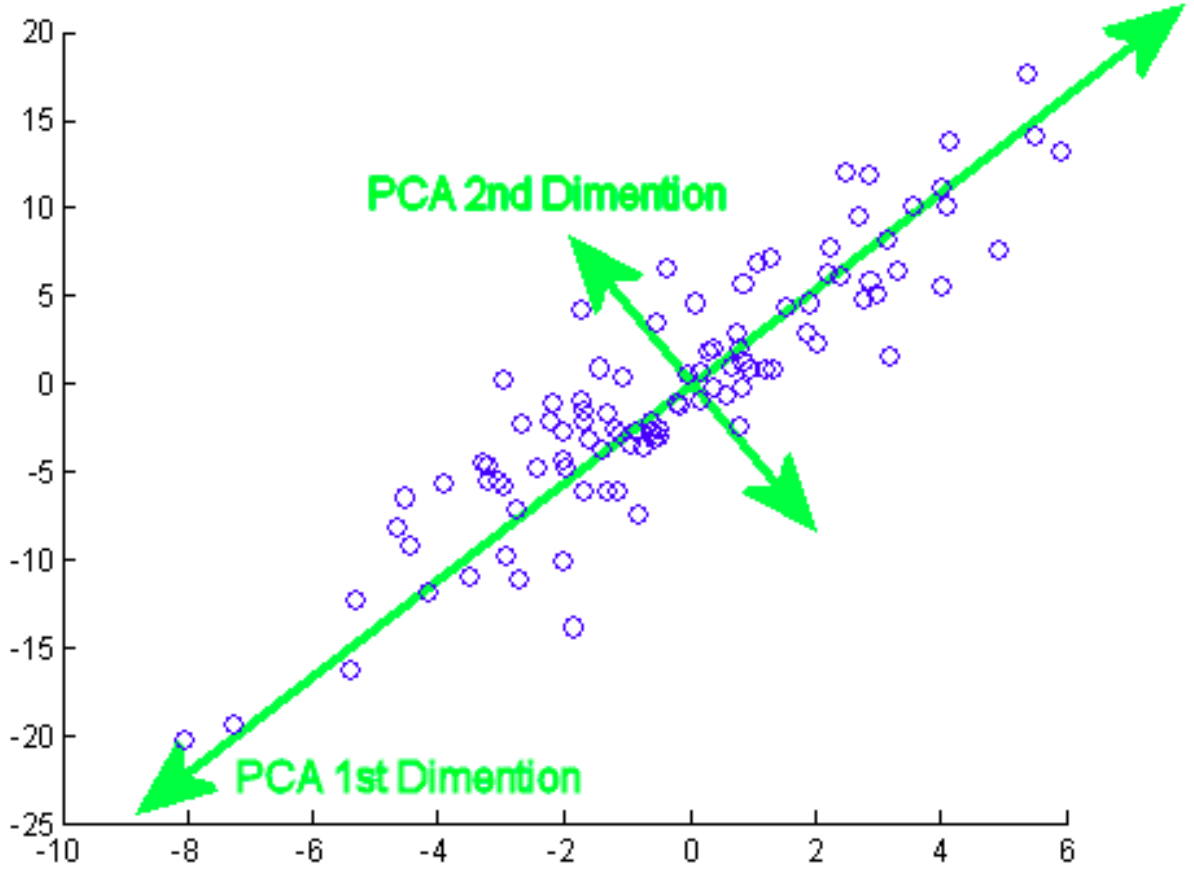
- Nonparametric method for obtaining relevant information from complex datasets.
  - It is one of the oldest dimensionality reduction methods.
- It is a technique used to emphasize variation and bring out strong patterns in a dataset by highlighting similarities and differences.
- PCA is mainly an exploratory technique that can be used to gain better understanding of the correlation between variables.

Principal Component Analysis (PCA) (2)

- **Principal Component Analysis** transforms a data set into a new orthogonal coordinate system in which data is centered and the features are completely uncorrelated
  - The mean of the new data is 0
  - The covariance of any pair of distinct features is 0
- The features are transformed through vectors called **principal components** that define a new coordinate system.
- Principal components are sorted in descending order by variance i.e., the first component has the largest variance
  - Components with low variance can be discarded, making PCA a method of dimensionality reduction.

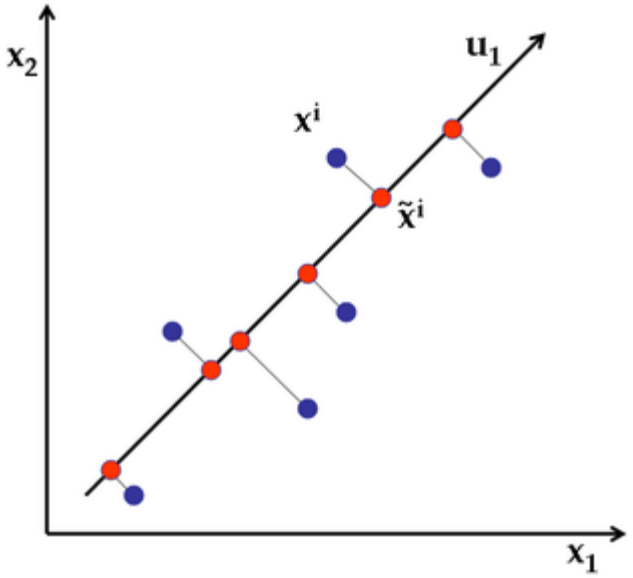
Principal Component Analysis (PCA)

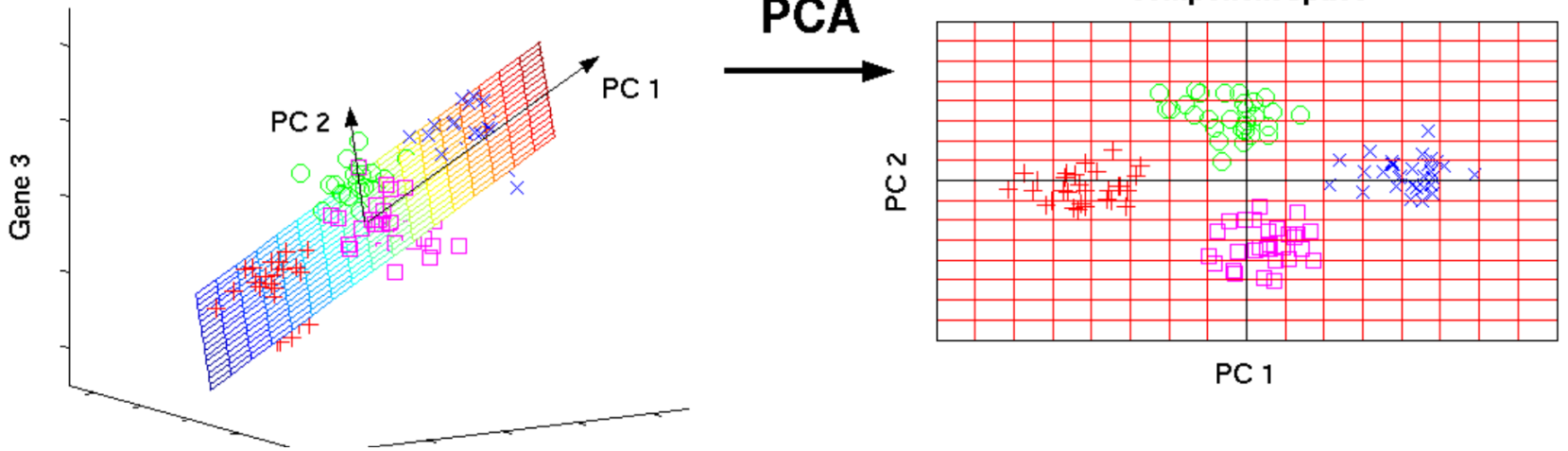
- The first axis (first principal component) is in the direction of largest variation.
- The second axis is orthogonal to the first axis and goes in the direction of second highest variance in the data.
- The third axis is orthogonal to both the first and second axis, and so on.



PCA problem formulation

- Reduce from 2-dimension to 1-dimension: Find a direction (a vector $u_1 \in \mathbb{R}^n$) onto which to project the data so as to minimize the projection error.
- Reduce from N-dimension to K-dimension: Find K vectors onto which to project the data, so as to minimize the projection error





PCA Process

- First, we need to center our data
  - $\hat{x}_{ij} = x_{ij} - \mu_i, \quad \mu_i = \frac{1}{N}\sum_{j=1}^{N} x_{ij},\ x \in \mathbb{R}^n$, N is number of data points
- Then we compute covariance matrix S for centered data and find its eigenvectors v and eigenvalues λ.
  - $S = \frac{1}{N} X X^T, S \in [n, n], X \in [n, N]$
  - $Sv = \lambda v, v = [n, 1], \lambda \in \mathbb{R}$

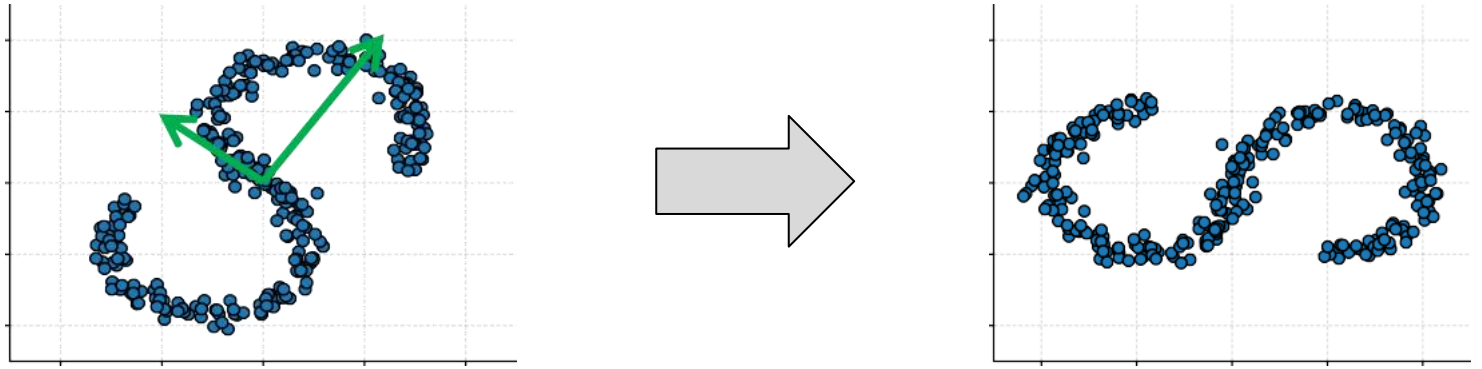
$$X = \begin{bmatrix} \hat{x}_{11} & \hat{x}_{21} & \cdots & \hat{x}_{N1} \\ \hat{x}_{12} & \hat{x}_{22} & \cdots & \hat{x}_{N2} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{x}_{1n} & \hat{x}_{2n} & \cdots & \hat{x}_{Nn} \end{bmatrix}$$

- Sort eigenvectors by corresponding eigenvalues in descending order
- Then we form matrix B from first M eigenvector and perform dimensionality reduction
  - $X^* = B^T X, \quad B \in [n, M], X^* \in [M, N]$

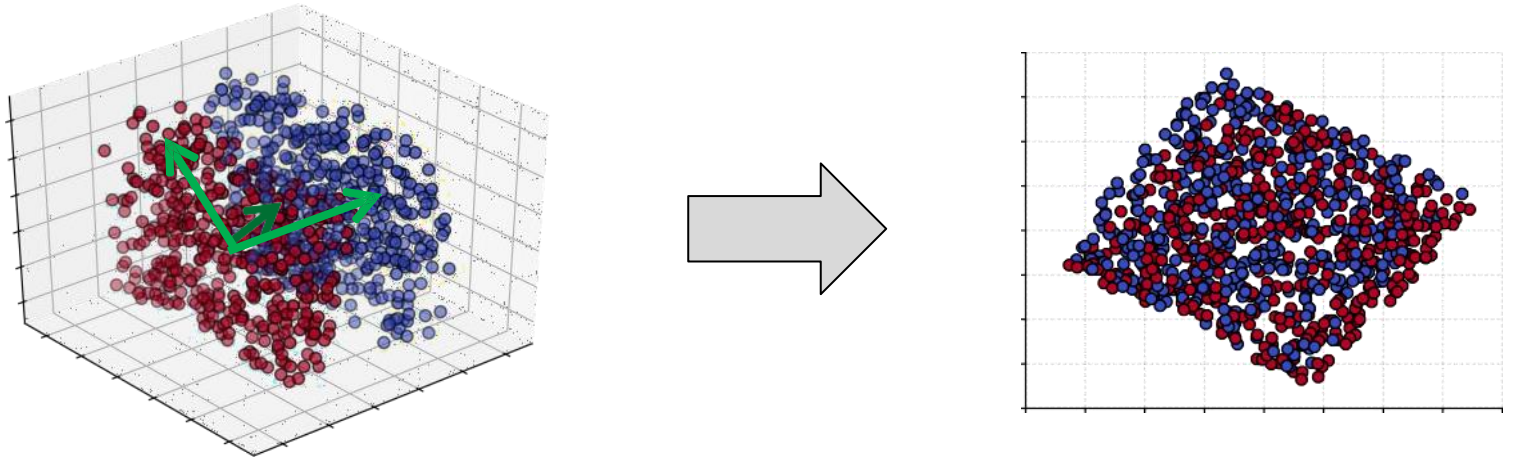
$$B = \begin{bmatrix} v_{11} & v_{21} & \cdots & v_{M1} \\ v_{12} & v_{22} & \cdots & v_{M2} \\ \vdots & \vdots & \ddots & \vdots \\ v_{1n} & v_{2n} & \cdots & v_{Mn} \end{bmatrix}$$

Limitations

- PCA does not consider non-linear correlations:



- PCA does not take labels into account, only variance of features:



Principal Component Analysis (PCA) – Comments

- PCA is not a linear regression
- Preprocessing of data (feature scaling, normalization) is required
- Singular Value Decomposition (SVD) is key tool for computing the principal components. [U, S, V] = svd(Sigma)
- Reconstruction from compressed representation
- Choosing the number of principal components
- Application of PCA is mainly:
  - To reduce memory needed to store data
  - To speed up learning algorithm
- To reduce overfitting, use regularization not PCA

Why not PCA for visualization?

- Recall the PCA objective: project data onto a lower dimensional subspace, such that the variance is maximized
- *Why the bad results?*

1. Linear projection
2. Mostly preserves distances between dissimilar points

➢ Is this really what we want for the purpose of visualization?

t-SNE

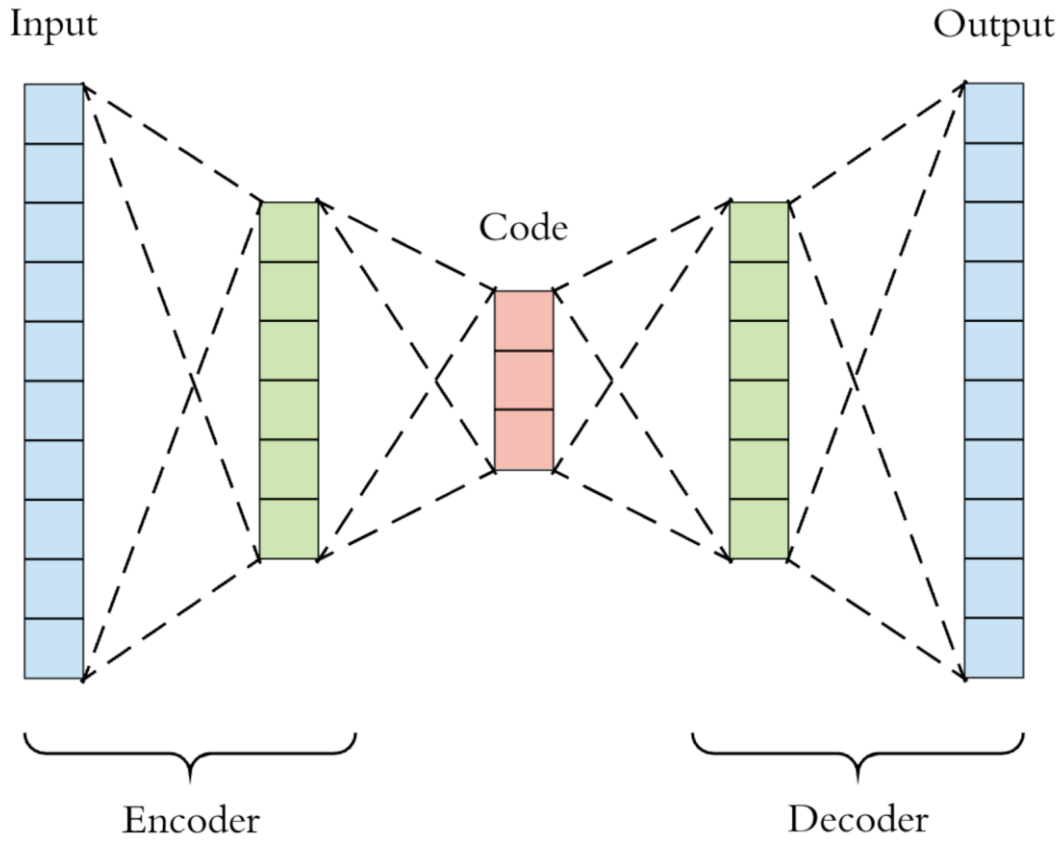
- **t-Distributed Stochastic Neighbor Embedding (t-SNE)** is a non-linear dimensionality reduction and visualization technique
- It maps the data points in a lower-dimensional space, such that points which were close in the original space remain close in the embedding space
  - It focuses on preserving local structure and does not preserve global structure
  - The results are non-deterministic
  - It is computationally expensive
- t-SNE does not produce a transformation of the space, it generates an embedding into a completely new space
- The features in the new space are difficult to interpret

Are we overfitting?

Machine Learning	versus	Visualization
Goal: Generalization Given a training set, do well on a test set. **Overfitting is undesirable**		Goal: Visualization We just want to “do well” on our data (“training set”) **“Overfitting” is desirable**

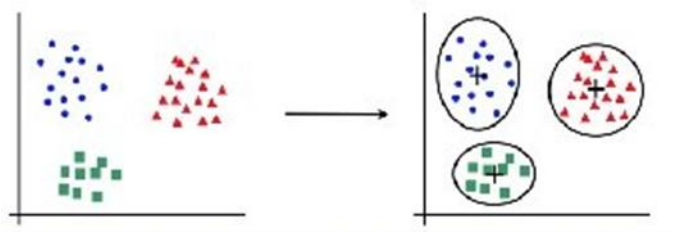
Autoencoder

- Type of neural network with encoder-decoder architecture.
- It is trained to reconstruct the input. We want the output of the autoencoder to be as similar to the input as possible.
- The Bottleneck Layer in the middle has much lower dimension than the input.
- A denoising autoencoder corrupts in input signal during the training by adding some random perturbation to the features.
- Effectively used in many applications: face recognition, acquiring the semantic meaning of words, anomaly detection, drug discovery, etc.

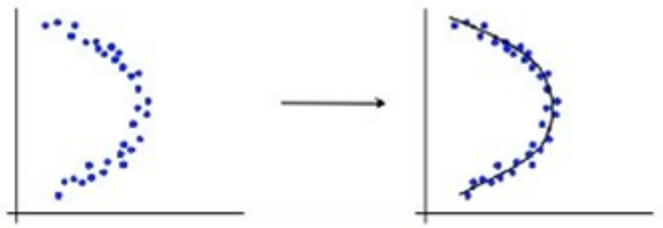


Summary

- Unsupervised Learning
  - No labels just the data
  - Grouping-Clustering
    - K-means



  - Dimensionality reduction
    - PCA



  - Autoencoders/Representation learning/Generative Models