Introduction to Statistical Machine Learning CSC/DSCC 265/465

Lecture 10: Unsupervised Learning – Part I

Cantay Caliskan

Plan for today

- ***Regularization***
- ***Kmeans***

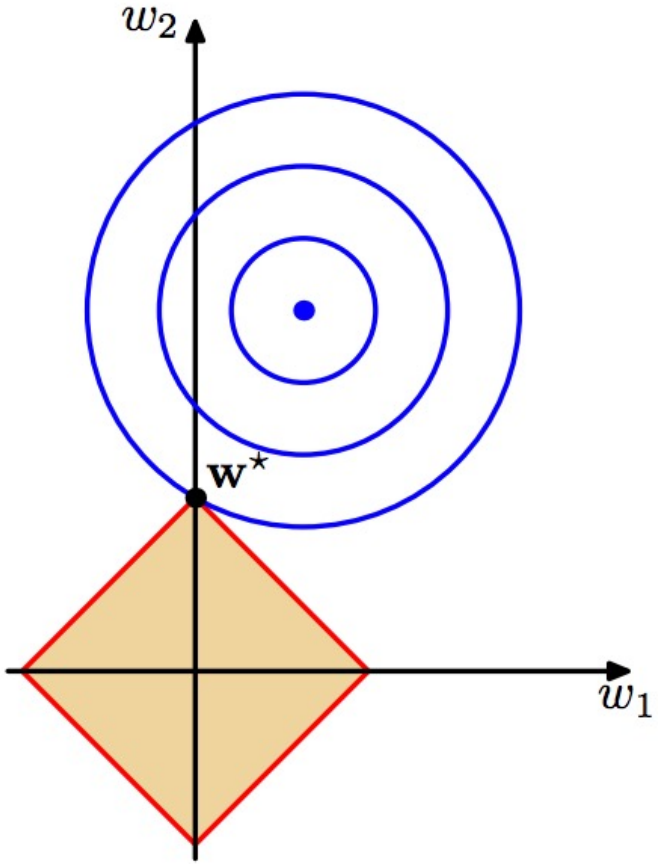
Plan for today

- ***Regularization***

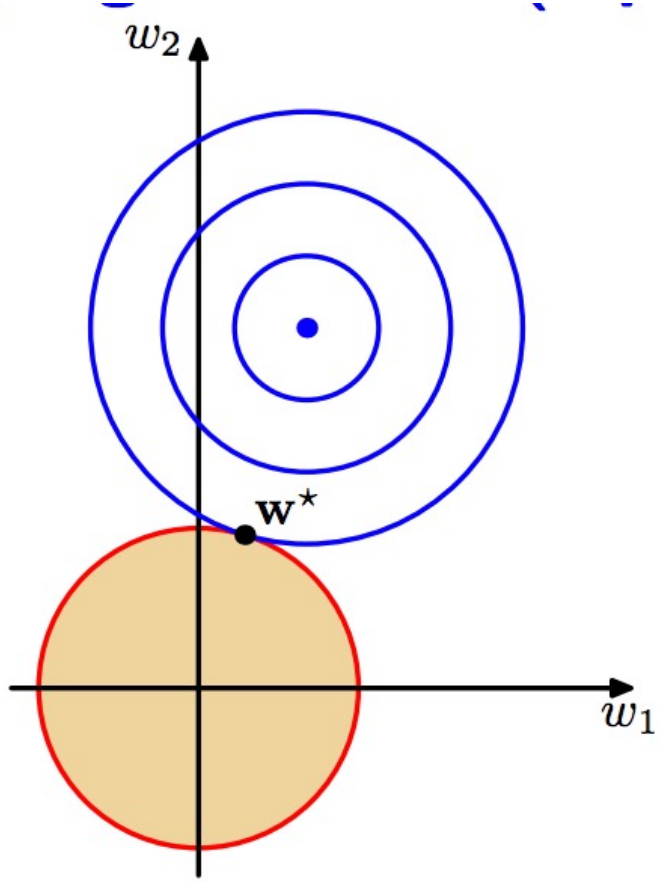
- *Kmeans*

What is left of: Regularization

Visualizing Regularization

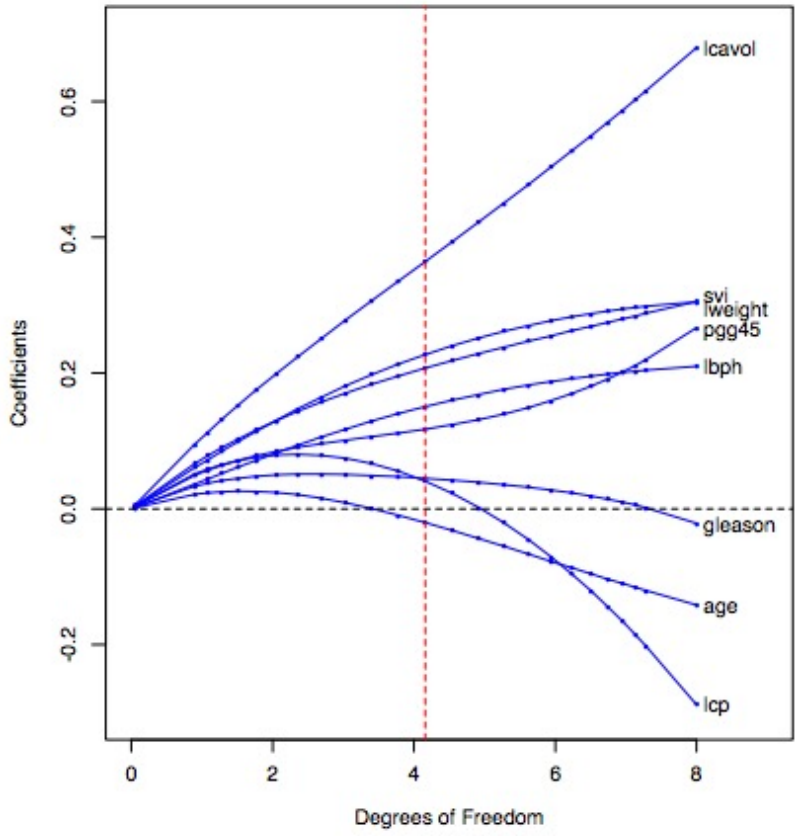


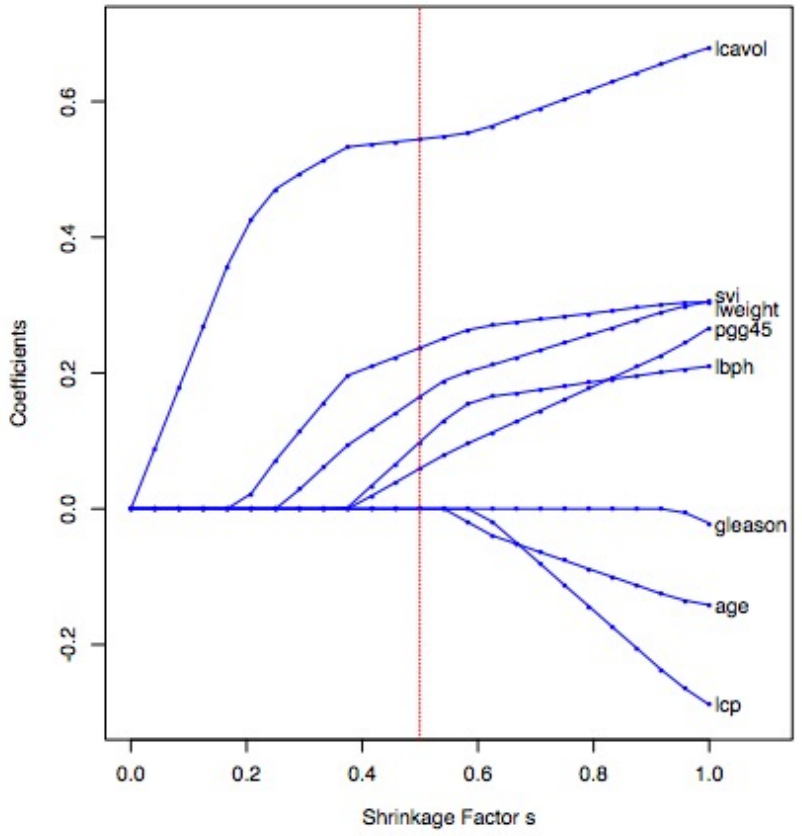
L1 Regularization



L2 Regularization

L1 vs. L2 effect





Question: Which one is L2 regularization?

L2 regularization is on the left!

So, which λ value should we pick?

- Two potential approaches:

1) ***Statistics approach***: Pick a value such that some information criterion, such as **AIC** or **BIC** is the smallest (highest goodness of fit):
   - **AIC**: Akaike Information Criterion
   - **BIC**: Bayesian Information Criterion

2) ***Machine learning approach***: Perform cross-validation and select the value λ that minimizes the cross-validated sum of squared residuals

Further approaches to pick the best λ

- **Manual tuning**: Finding the right combination of settings for regularization can be done via manual tuning (if you have the expertise)

- **Grid search:** Find two sets of parameters that are believed to contain the best set of parameters and converge toward the best set

Ridge vs. Lasso

- Often neither one is overall better

- ***Lasso*** can set some coefficients to zero
  - Thus, Lasso performs variable/feature selection
  - Ridge does not do that

- Both methods allow to use correlated predictors, but they solve multicollinearity issues differently:
  - ***Ridge*** regression: Coefficients of correlated predictors are similar
  - ***Lasso*** regression: One of the correlated predictors has a larger coefficient, while the rest are (nearly) zeroed

Ridge vs. Lasso

- ***Lasso*** tends to do well if there are a small number of significant parameters and the others are close to zero
  - When only a few predictors actually influence the response

- ***Ridge*** works well if there are many large parameters of about the same value
  - When most predictors impact the response

- In practice, we don't know the parameter values, so it is really hard to decide between two:
  - Answer: *ElasticNet* Regression

ElasticNet Regression

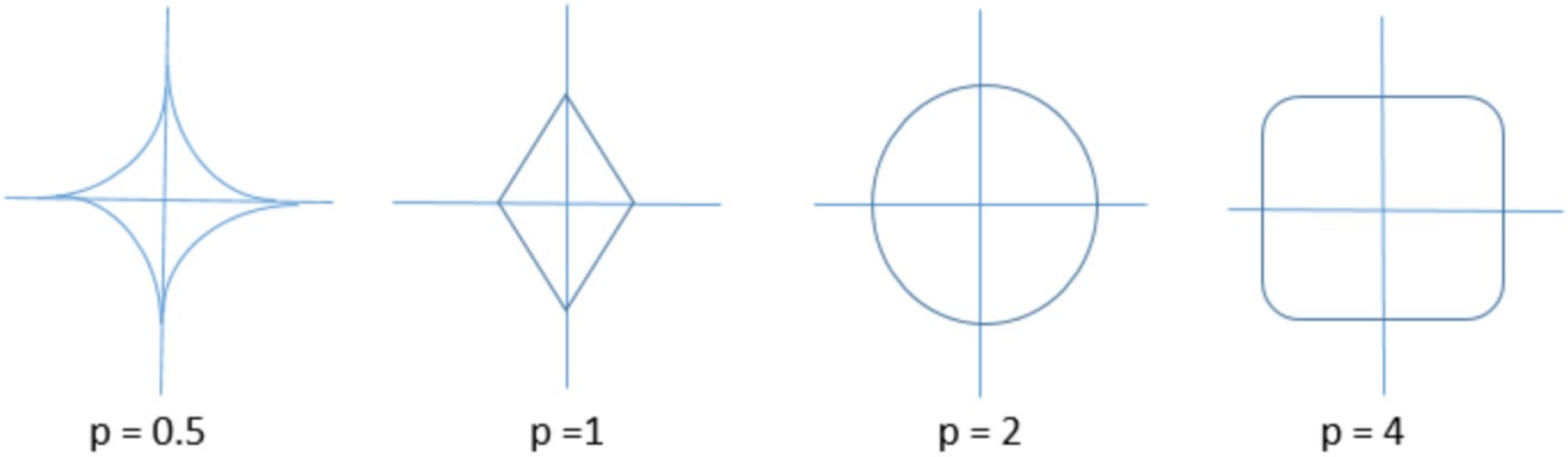
- First emerged as a result of critique on ***Lasso***
  - The variable selection for Lasso can be too dependent on data and thus unstable
  - Solution: Combine the penalties of Ridge and Lasso to get the best of both worlds.
  - The loss function for ElasticNet:

$$\mathbf{J(W)} = \frac{1}{2N}\sum_{i=1}^{N}((W_0 + W_1X_1^{(i)} + \ldots + W_PX_P^{(i)}) - Y_i)^2 + \frac{\lambda_1}{2N}\sum_{j=1}^{P}|W_j| + \frac{\lambda_2}{2N}\sum_{j=1}^{P}W_j^2$$

  - There are two parameters to tune: λ_1 and λ_2

L^p Regularizers

- No need to use only L1 and L2, or L1+L2
  - ***L^p regularization*** is also possible.



Question: What happens with the parameters here?

Hint: Axes represent your coefficient values

‘Other’ types of regularization

- ***Early stopping***
  - You can avoid overfitting if you are using an iterative optimization method, such as gradient descent
  - Conditions for early stopping: Amount of change in updates (i), number of iterations (ii)
  - Question: What are some disadvantages here?
  - A solution to some disadvantages: Validation-based early stopping

- ***Principal component regression (PCR)***
  - Instead of using the independent variables directly, the principal components of the explanatory variables are used as regressors

Plan for today

- *Regularization*
- ***Kmeans***

Unsupervised Learning

Unsupervised Learning: Motivation

- **Think about**
  - **Ways to 'group' some books**
  - Based on topic, color, size, author, alphabet, language etc.

- **How you can extract the new letters in an alphabet**

- **How you can extract an interesting style of clapping**
  - **Or hair**

Classification of Algorithms (in general)

1) **Pre-programmed algorithms**
   - Algorithms that work *deterministically*
   - Examples:
     - Sorting algorithms
     - (Most of the) graph algorithms

2) **Learning algorithms**
   - Algorithms that '*improve*' their behavior based on new input from the world

   - ***Machine learning***
   - And ***deep learning***

(Another) Classification of Learning Algorithms

1) Unsupervised Learning (=clustering)

- When the data does not include classification, this is called ***unsupervised learning***
- Example: Figuring out all different types of music

K-Means is here

2) Supervised Learning (=classification)

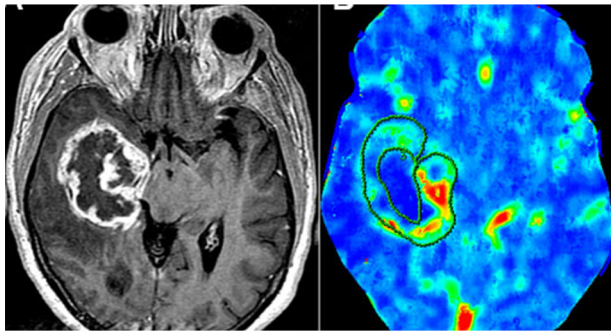
- Algorithms are given a ***training set***:
  - ***Training set***: Bunch of a data points and classification for those points
- New data is compared to the training set to make classifications
- Example: Figuring out the numbers written on a bank check

3) Semi-supervised Learning

- Label some data, and let the algorithm figure out the classification
- Example: Facebook's news feed algorithm which understands how an important topic looks like

The Clustering Problem

- Suppose we have ***N*** observations with ***p*** features
  - We may have little (to no) knowledge on those features…



- <u>Why would we do clustering?</u>
  - Stand-alone tool to gain insight into the **data visualization**
  - Usually much less costly -> **no labeling is necessary**
  - **Preprocessing** step for other algorithms
    - Indexing or compression often relies on clustering

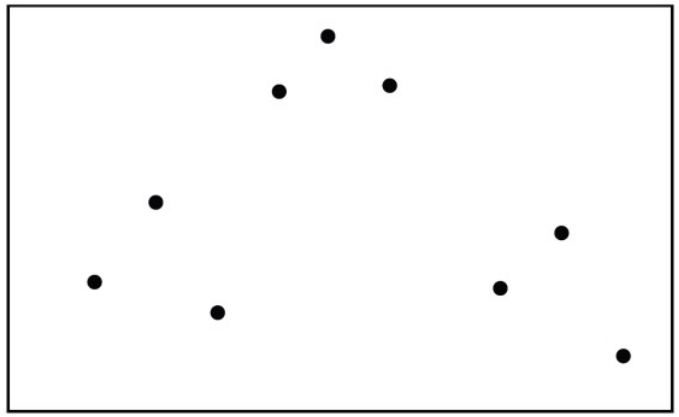
Machine Learning = A Lot of Human Labor

Some non-trivial issues

- Reminder: Our goal is…
  - Find a grouping so that
    - Similar objects are in the same cluster
    - Dissimilar objects are in different clusters

- Basic questions:
  - What does **similar** mean?
  - What is a **good partition** of the objects?
    - i.e. how is the quality of a solution measured?
  - How to find a good partition?

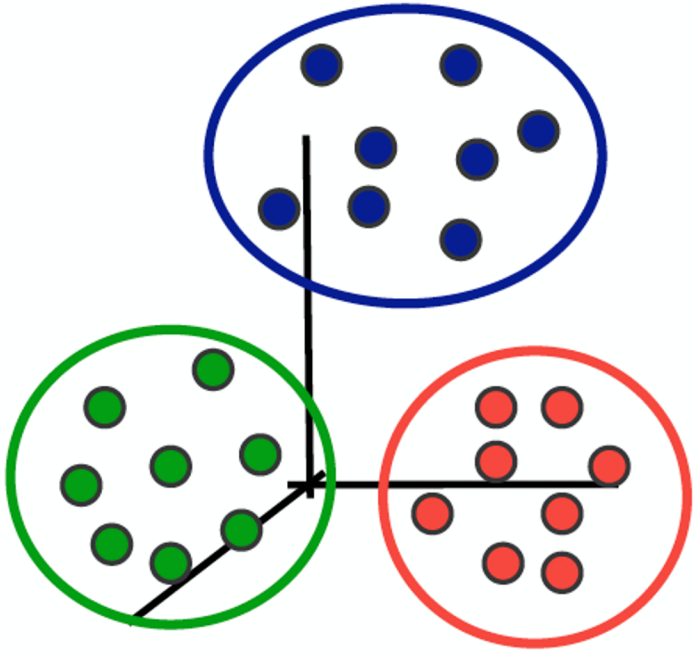
Clustering



- Assume the data $\{X_1, X_2, X_3, \ldots, X_D\}$ lives in a space defined by a distance metric (such as Euclidean space) where $X_d \in R^D$
- Assume the data belongs to **K** classes (clusterings)
- Assume the data points from same class are similar
- Question: How can we identify those classes?
- Question: What is the best grouping?

What is clustering?

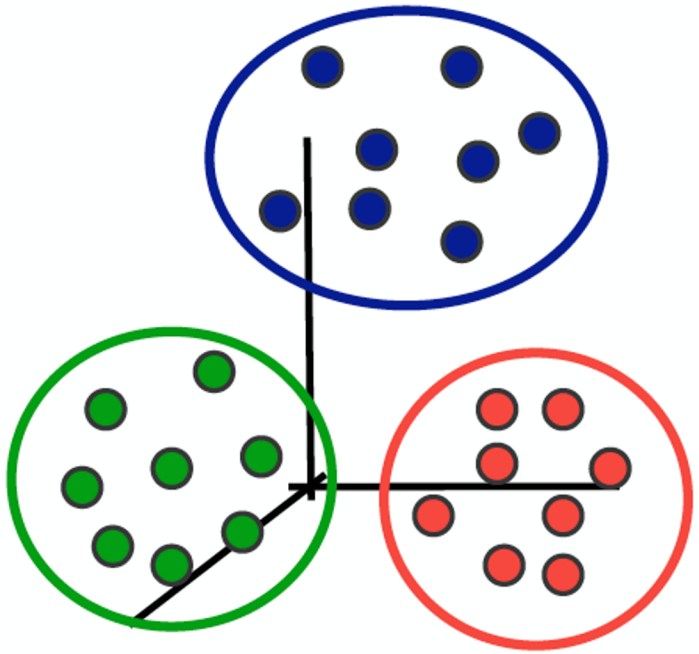
- A grouping of data objects, such that:
  - Objects within a group are similar (or near) to one another
  - And dissimilar (or far) from the objects in other groups



How to achieve this objective?

- A grouping of data objects, such that:
  - Objects within a group are similar (or near) to one another
  - And dissimilar (or far) from the objects in other groups

Minimize intra-cluster distances



Maximize inter-cluster distances

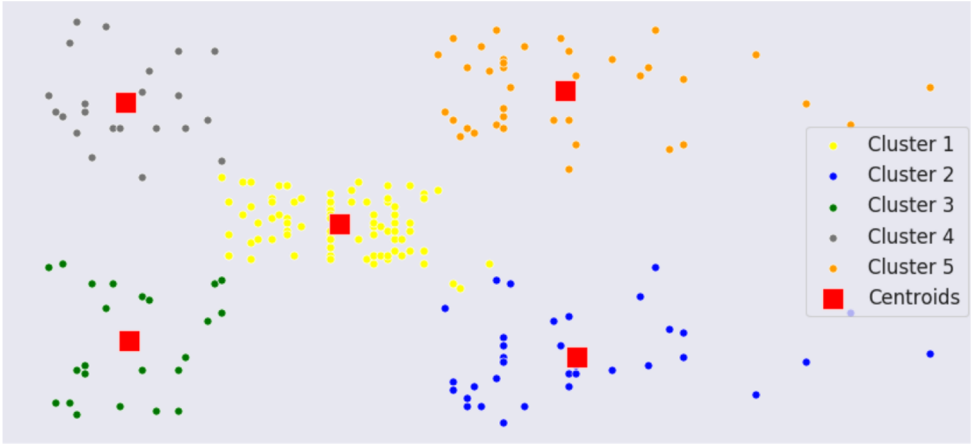
K-Means Clustering

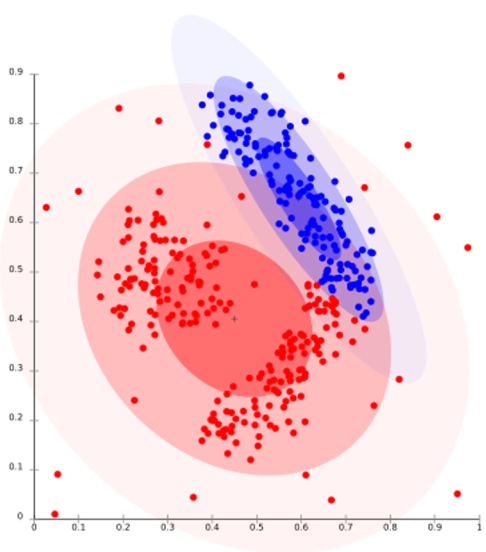
K-Means

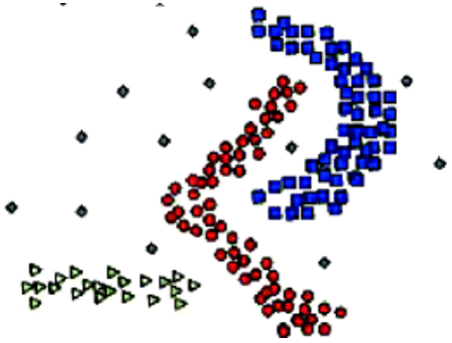
- **K-Means Clustering**
  - Clustering and Classification
  - Principles
  - Quality Control
  - Advantages and Drawbacks

Types of clustering

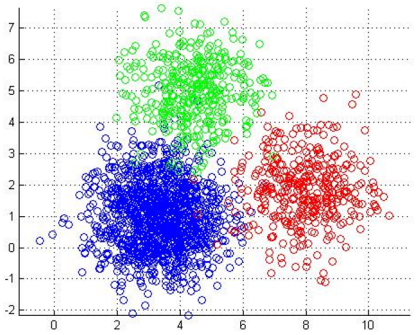
- **Centroid-based clustering**
  - Select the points associated with a center
- **Density-based clustering**
  - Closely located points put together
- **Distribution-based clustering**
  - Points from similar distribution put together
- **Hierarchical clustering**
  - Merge points bottom-up







K-Means Clustering

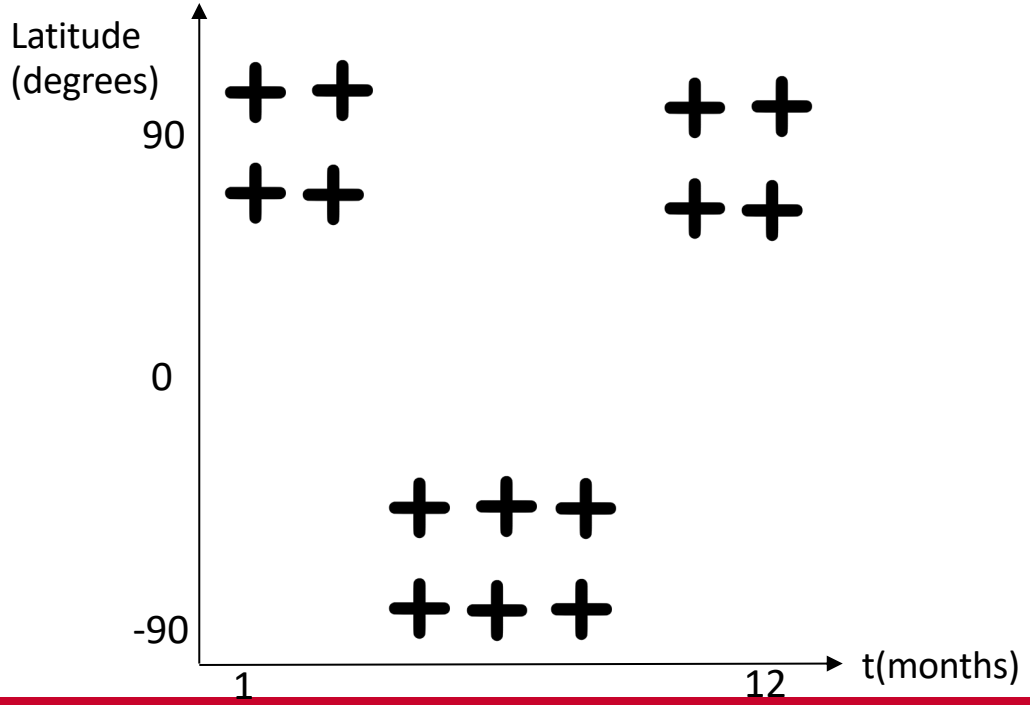


- A simple yet widely used clustering technique
- A ***randomized*** algorithm ← Some luck is involved!
- An ***iterative*** algorithm ← It takes turns to obtain the result!
- Some people consider it to be ***greedy*** ← Focus on short-term!
- Let's take a look at an example!

Example: Where can we ski?

- Question: Pretend that you have a dataset that shows the data points where you can **ski**

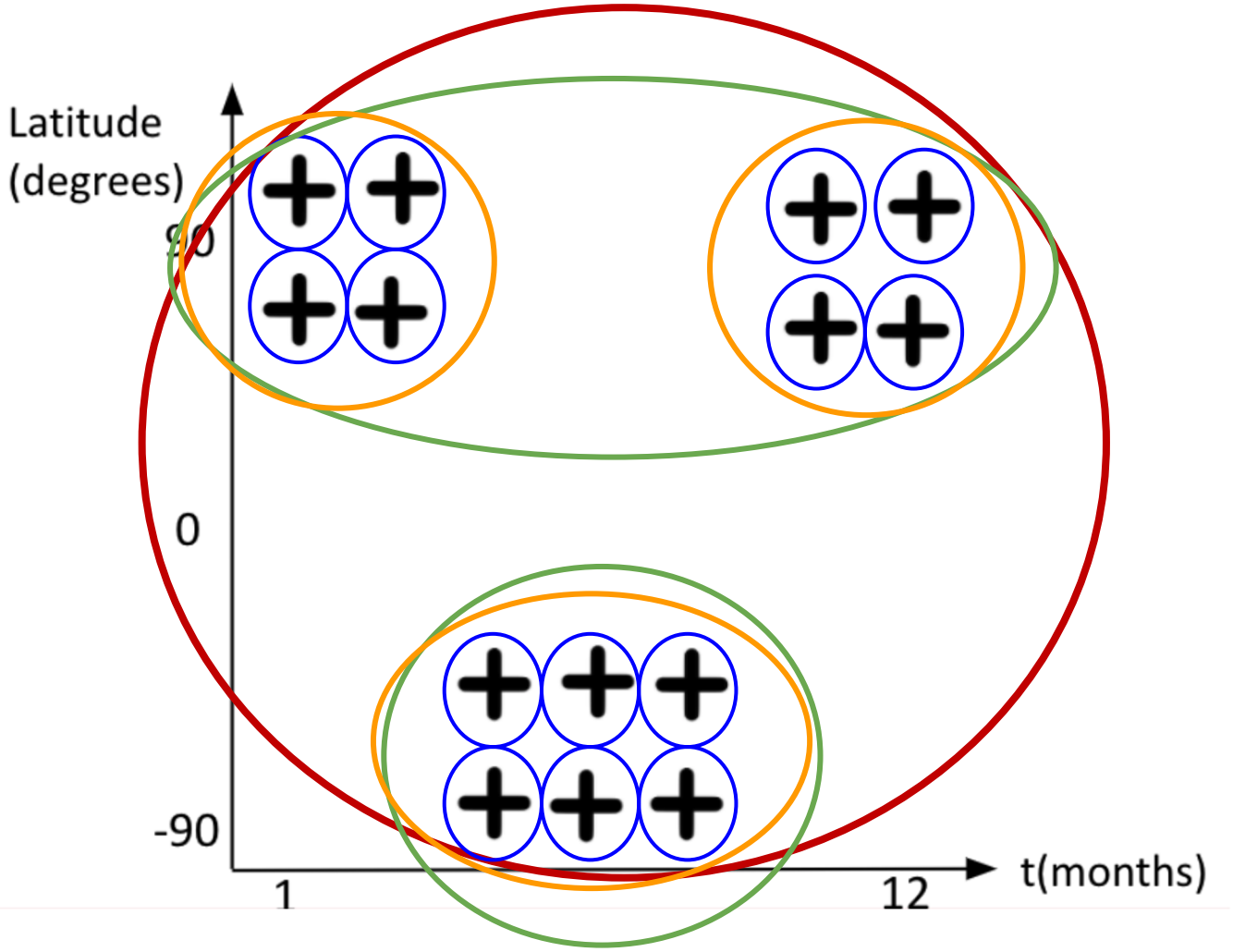
- Variables: *Latitude* and *season*



Goal: Grouping similar points together

Challenge: What is the optimal number of groups?

Notion of clustering...



How can we group all these points?

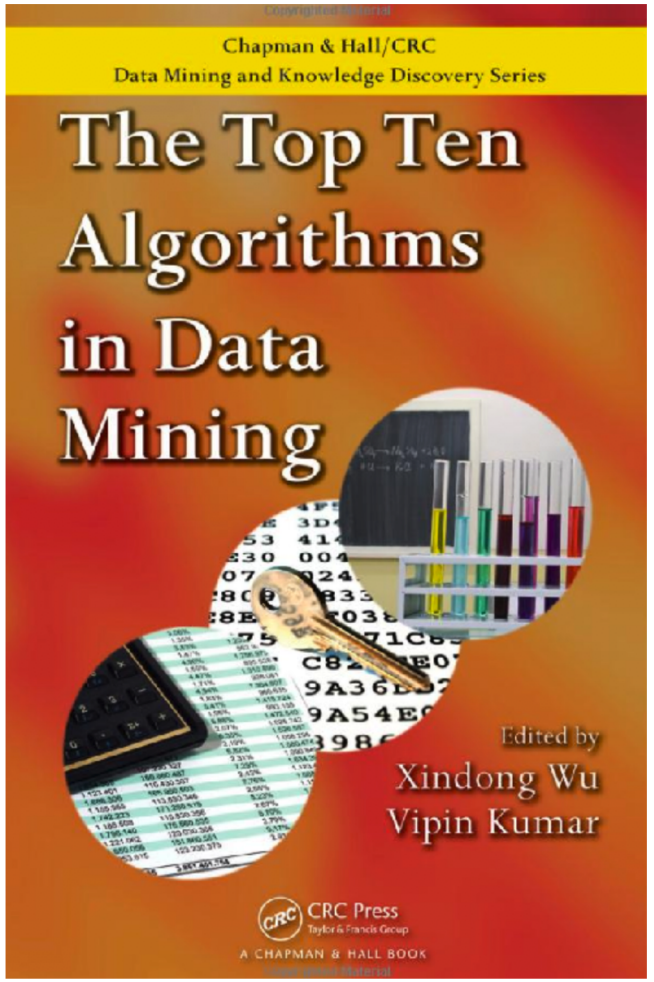
- *One* group for **all points**?
- *N* groups for **each point**?
- *Two* groups based on latitude?
- *Three* groups based on latitude and time?

Centroid Clustering

- Partition the n objects into k clusters
  - Each object belongs to exactly one cluster
  - The number of clusters k is given in advance

The K-Means Algorithm

- Voted among the top-10 algorithms in data mining
- Idea goes back to **Hugo Steinhaus** (1956) (‘*Sur la division des corps matériels en parties*’)
- First used by **Stuart Lloyd** at Bell Labs (1957)
- K-Means as a word was proposed by **James MacQueen** (1967)
- Published in an article by Edward Forgy in 1965
- One way of solving the K-Means problem



The K-Means Problem

- Consider set X = {x_1, x_2, …, x_n} of n points in R^d
- Assume that the number k is given
- Problem:
  - Find k points c_1, …, c_k (named centers or means)
  - And partition X into {X_1, …, X_k} by assigning each point x_i in X to its nearest cluster center
    - So that the cost

$$\sum_{i=1}^{n} \min_{j} \left\{ L_2^2(x_i, c_j) \right\} = \sum_{i=1}^{n} \min_{j} ||x_i - c_j||_2^2$$

is minimized

Standardization by cluster size also possible!

k = 1 and k = *N* are easy special cases (why?)

The K-Means Algorithm

1. Identify how many clusters (***k***) you want to have

2. Randomly (*or with another method*) pick ***k*** centroids $\{\mathbf{c_1}, \ldots, \mathbf{c_k}\}$

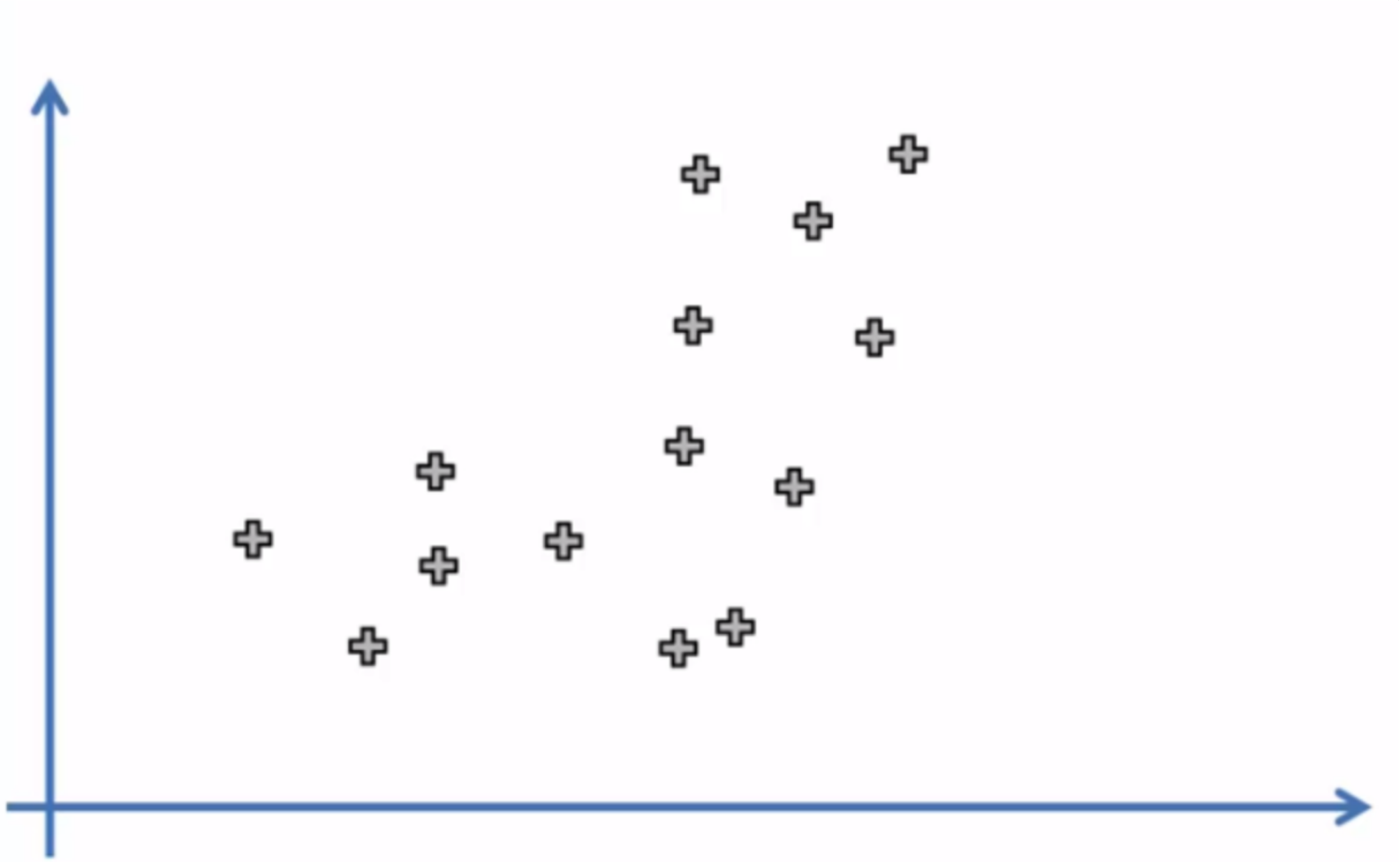
3. Assign each data point $\mathbf{x_i}$ in your feature set **X** to the closest centroid
    - Now we have the initial set of ***k*** clusters.

4. Re-calculate the centroid in each cluster by finding the point closest to every other point on average.
    - This is the mean of all points summed up in vector format.

5. Re-assign each data point $\mathbf{x_i}$ to the new closest centroid (cluster update!)

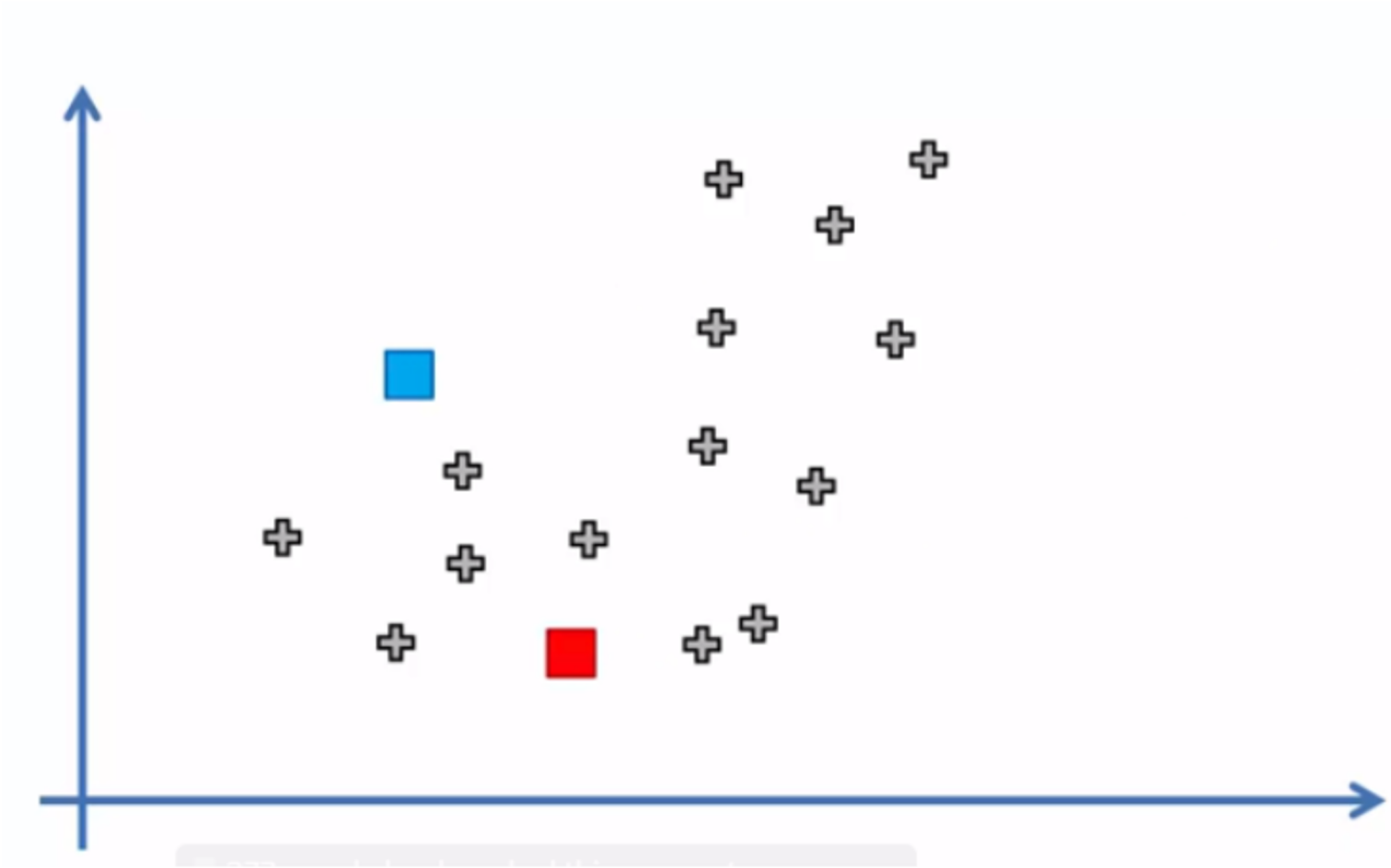
6. Perform **Step 4** -> until convergence.

Step 1: Choosing the Number of Clusters



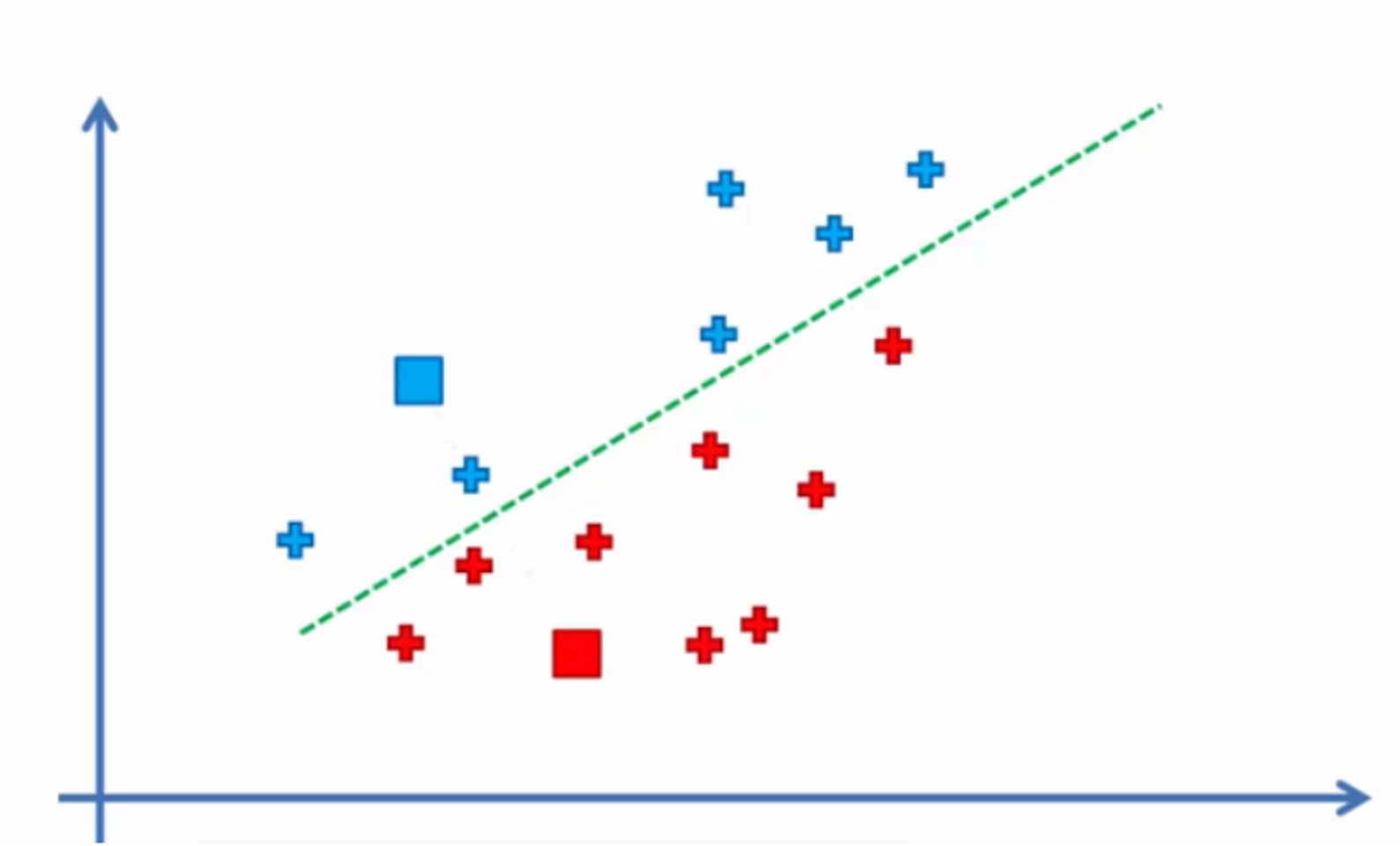
- What is the optimal number of clusters here?

Step 2: Select K centroids



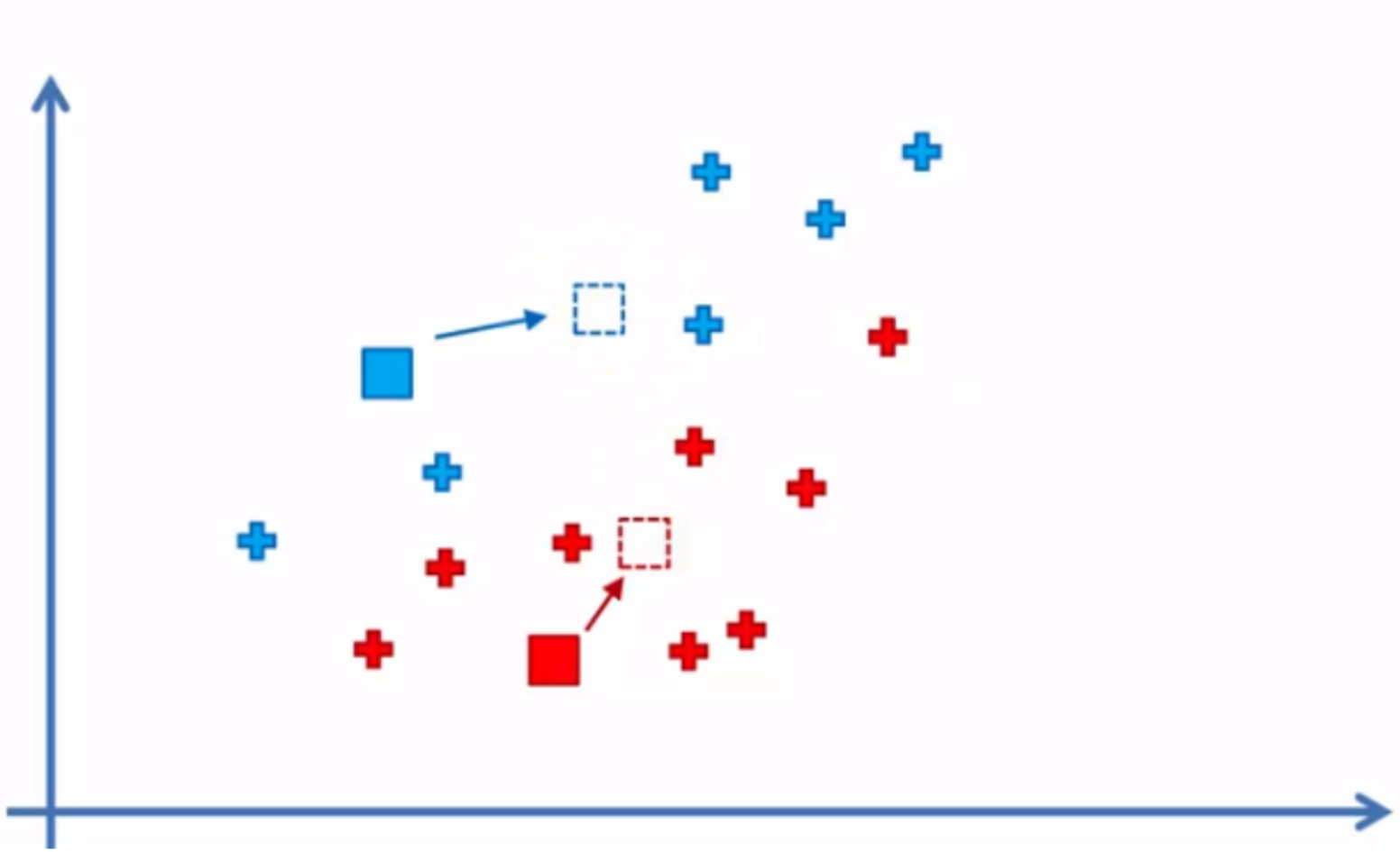
- Centroids *can be* **fictional** points in the vector space

Step 3: Point assignment



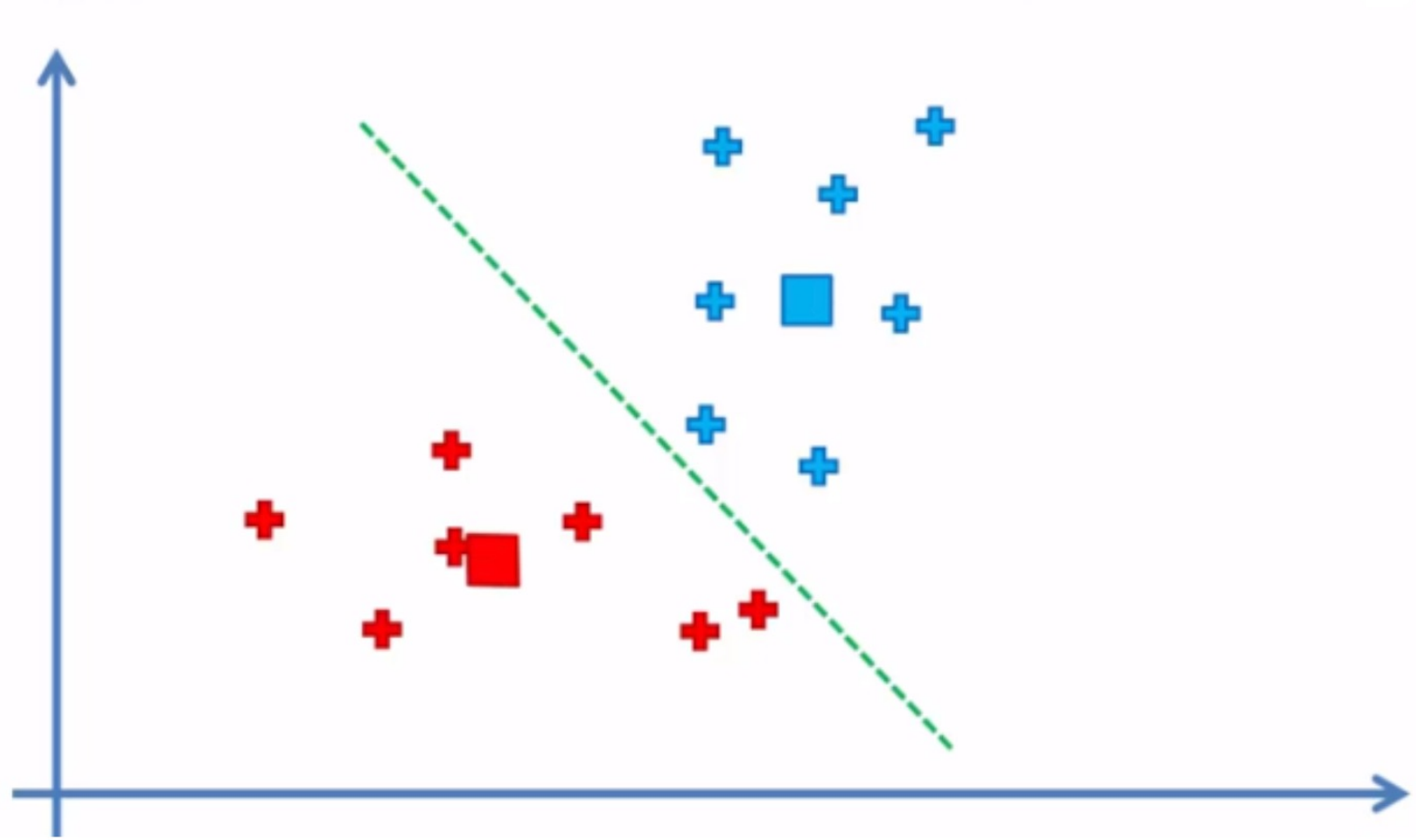
- Assign the data points to the closest ***centroids***

Step 4: Re-calculating the centroids



- Re-calculate the centroids for each cluster
- An easy way to calculate the centroid: find the ***mean*** of all data points in a cluster
- We completed our **first iteration**!

Step 5: Re-assignment of data points



- Re-assign each data point to the new closest ***centroid***
- Update the clusters
- So, we completed our **second iteration**!
- Check if there is a better centroid!

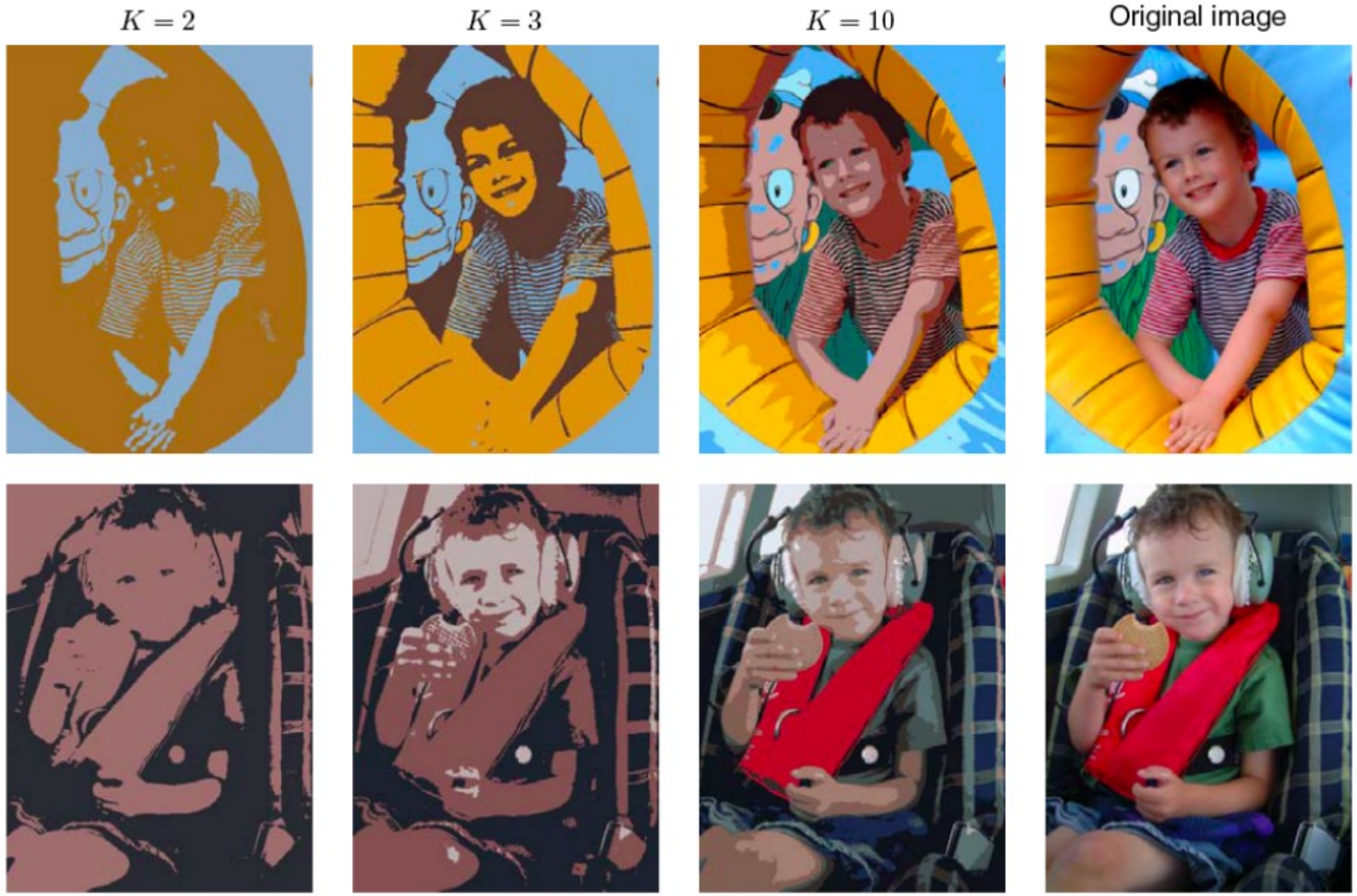
Video: K-Means clustering

https://www.youtube.com/watch?v=5I3Ei69I40s

Applications of K-Means in Real Life

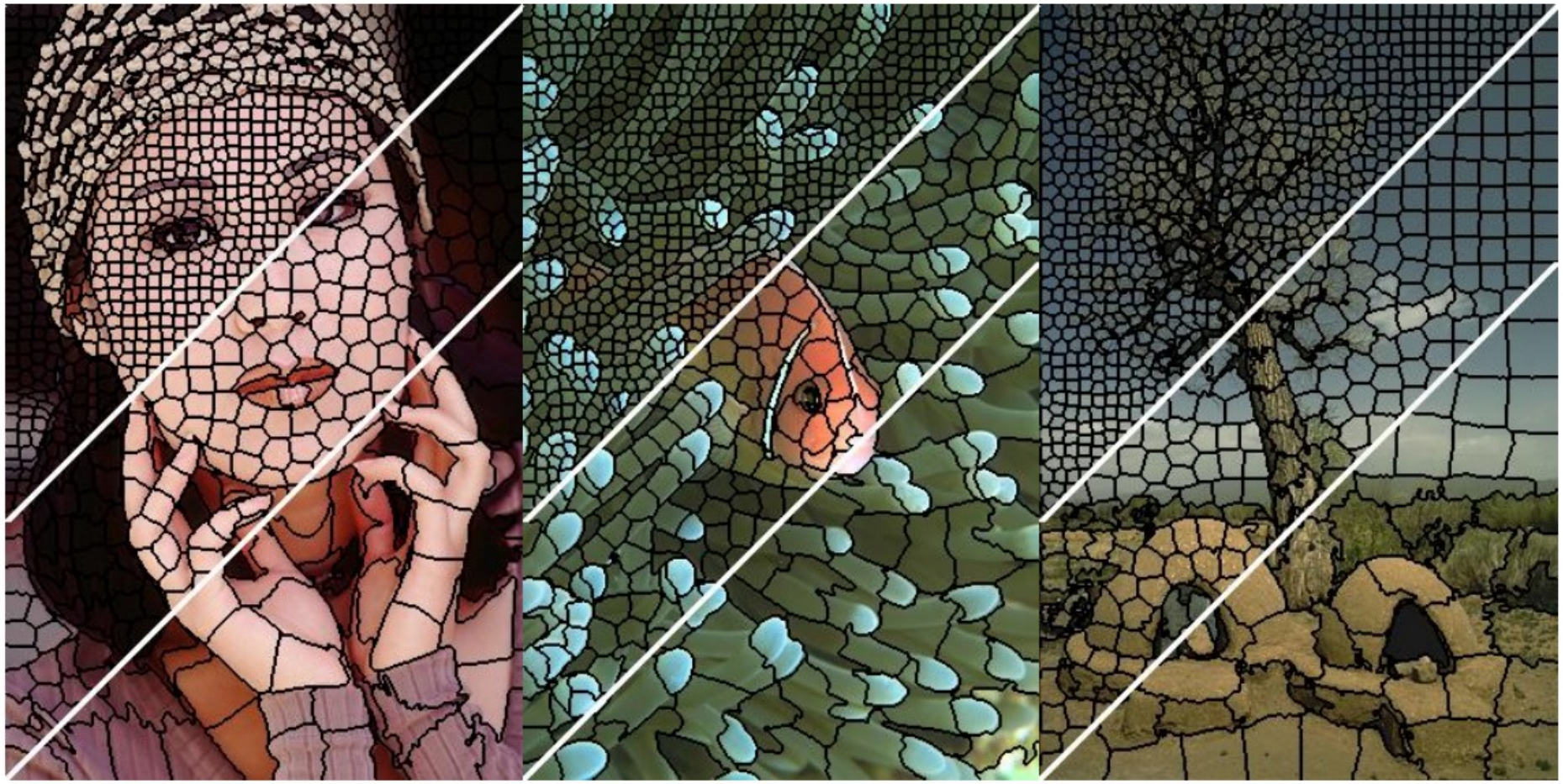
- **Behavioral segmentation**
  - Activities, purchases, profiles of people, user categories, voter categories
- **Categorization**
  - Grouping inventory, types of behavior
- **Sorting sensor measurements**
  - Detect activity types in motion sensors, group images, separate audio
- **Detecting anomalies**
  - Bots on Twitter, cancer cells etc.

K-Means for Vector Quantization



Source: Bishop

K-Means for Image Segmentation



Source: Bishop