HUGE

Hello

Clustering

2018

1. Introduction
2. Purpose.
3. Why and when?
4. Approaches
5. Validation

Agenda.

H

Who am I?

William Gómez
Web Engineer at HUGE

~~Ph.D. Student at Universidad de Antioquia~~

Adjunct professor at Universidad de Antioquia

1

Introduction

Artificial intelligence

Study of the design of intelligence agents to create machines that can mimic human intelligence.

Soft computing

Subdiscipline of AI that focuses on heuristics and imperfect solutions to complex problems.

Uncertainty.

Machine Learning

To make the machine learn by itself to solve problems using large quantity of data.

So what is ML actually?

Samples used to learn

ML System

Training

Model

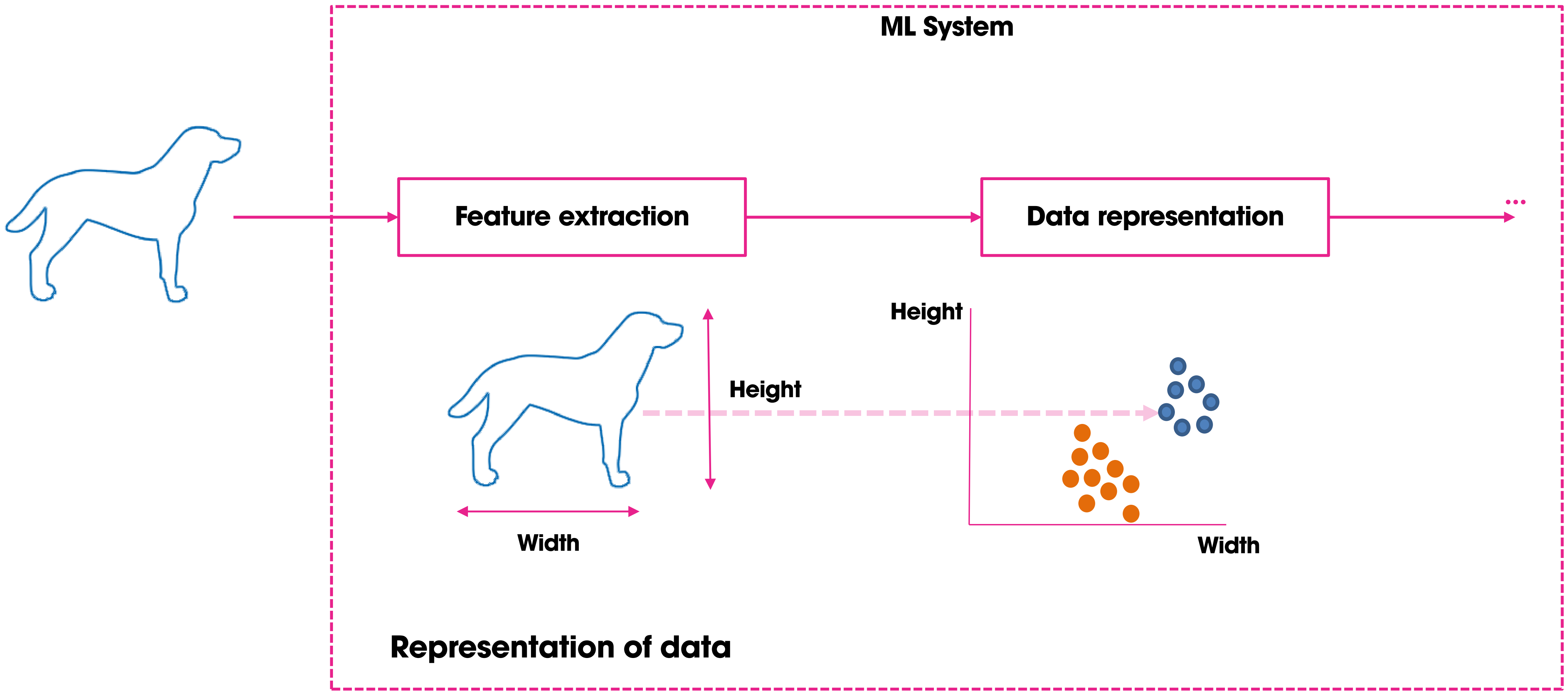
Prediction

Cat or Dog?

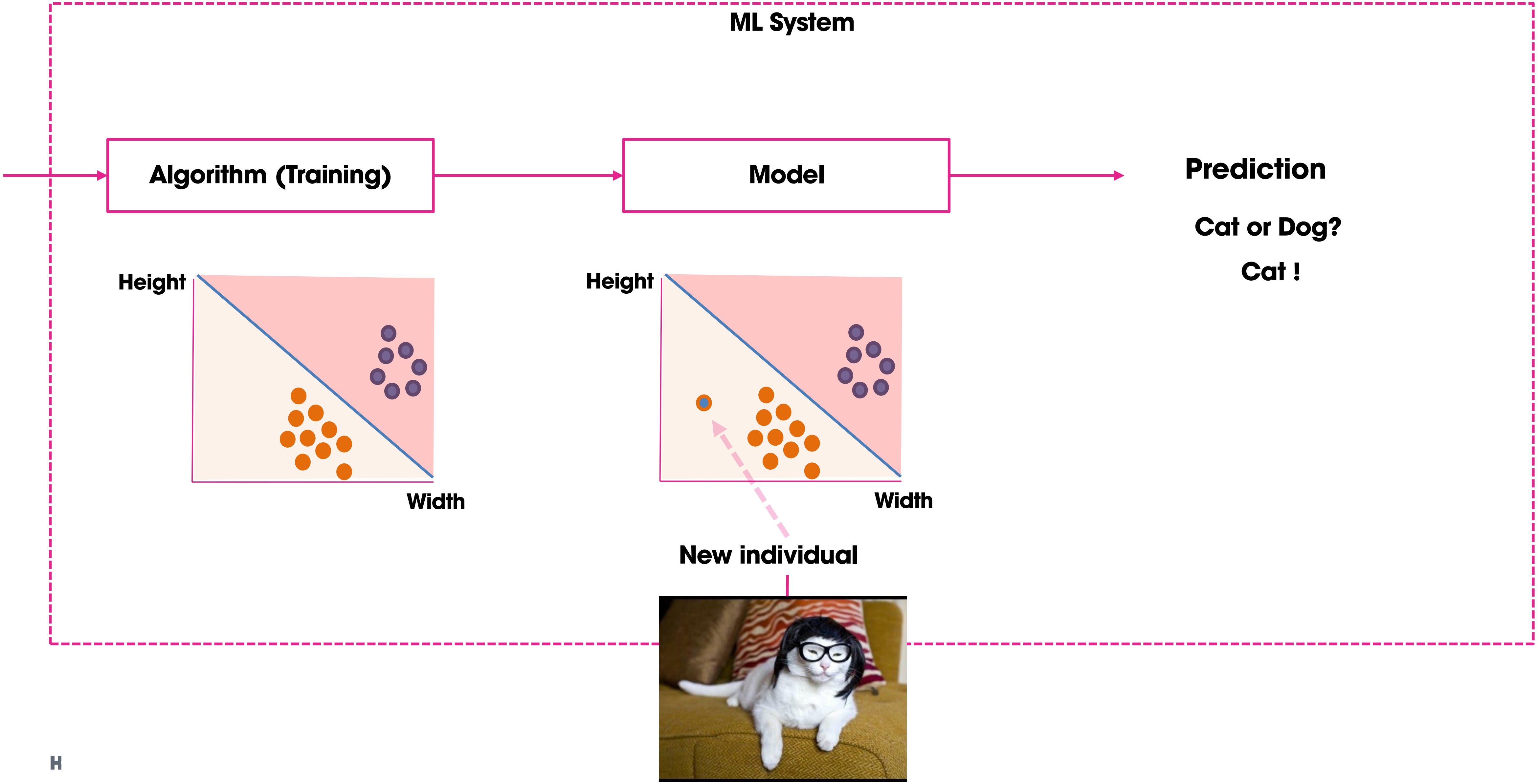
Cat !

New individual

How can ML be seen?



How can ML be seen?



Supervised

There is an expert knowledge that is desired to be reproduced.

Labels

$$f^*\left(\begin{bmatrix}\vec{X}_1\\ \vec{X}_2\\ \vdots\\ \vec{X}_k\end{bmatrix},\begin{bmatrix}y_1\\ y_2\\ \vdots\\ y_k\end{bmatrix}\right)=\begin{bmatrix}y_1^*\\ y_2^*\\ \vdots\\ y_k^*\end{bmatrix} \qquad \min\left(\begin{bmatrix}y_1^*\\ y_2^*\\ \vdots\\ y_k^*\end{bmatrix}-\begin{bmatrix}y_1\\ y_2\\ \vdots\\ y_k\end{bmatrix}\right)$$

Prediction

Unsupervised

There is no labeling or expected structure of the data.

$$f\begin{bmatrix}\vec{X}_1\\ \vec{X}_2\\ \vdots\\ \vec{X}_k\end{bmatrix}=\begin{bmatrix}y_1\\ y_2\\ \vdots\\ y_k\end{bmatrix}$$

~~Labels~~

Prediction, Hypothesis, Clustering, Patterns

2

Clustering and Purpose

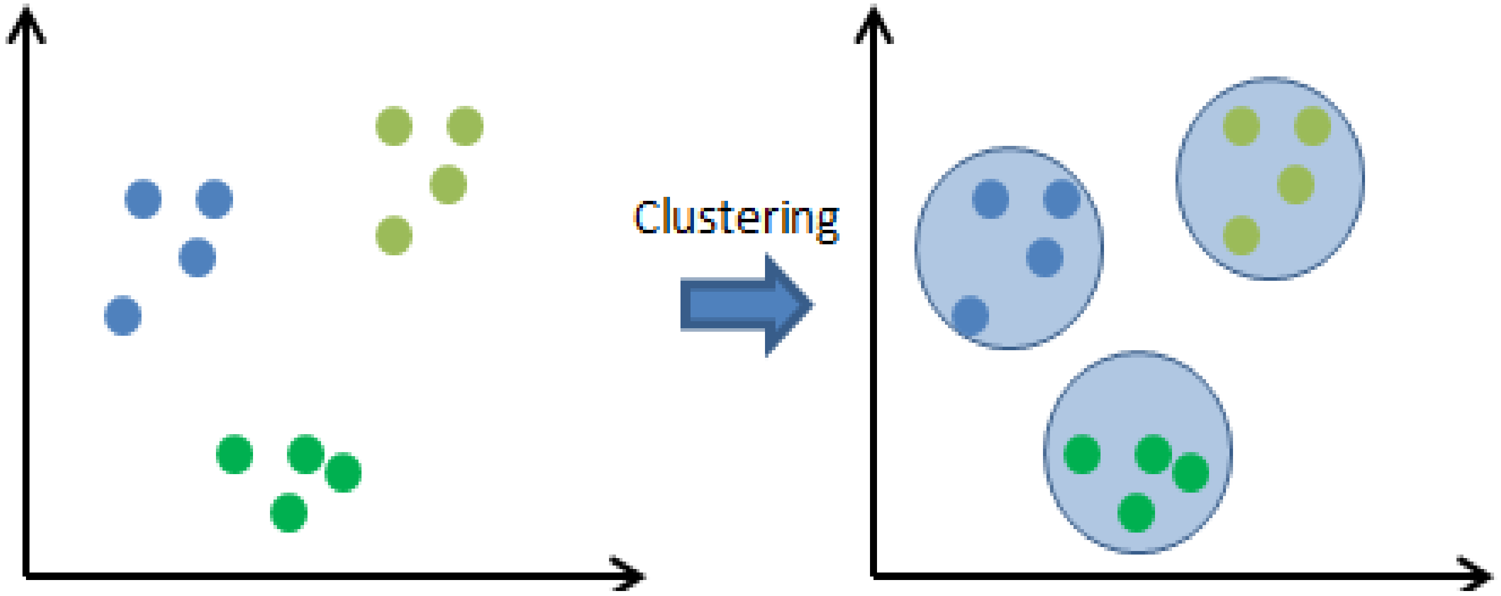
Clustering

Among the unsupervised learning, algorithms are clustering algorithms.

These focus on grouping objects according to their intrinsic characteristics or similarities.

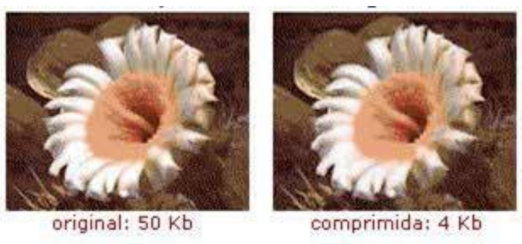
Clustering algorithms aim to find the structure of the data.

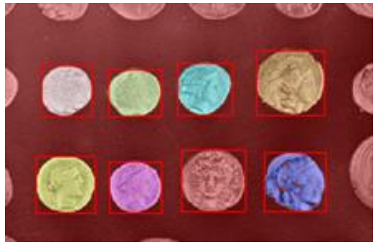
Clustering

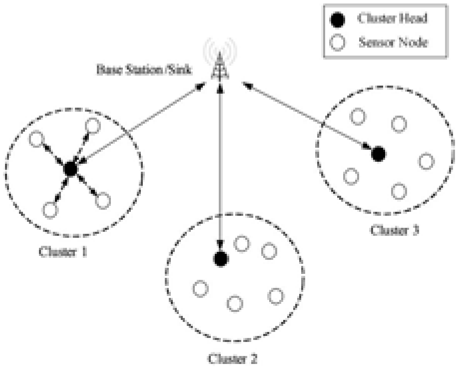


Applications

Data compression.
Image segmentation.
Controllers tunning.
Grouping of search engine results.
Speaker segmentation.
Multivariate statistics.







3

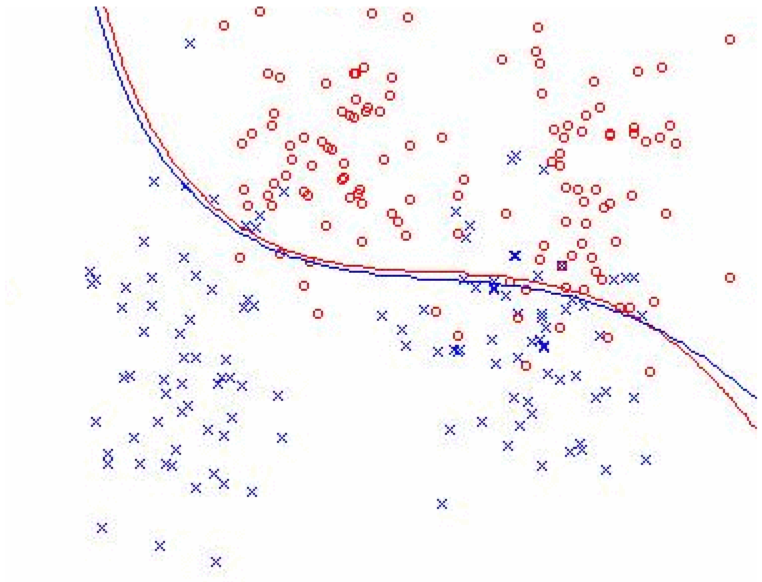
Why and when?

H

Clustering advantages

If the elements belonging to a cluster are similar, the prototype of the cluster allows us to make a synthesis of the elements in it.

Discovering groups even in predefined classes.



Why and when?

Data labeling doesn't always represent the actual structure of the data.

Why and when?

For example: We can have a dataset with dogs and cats, the labeling tells us that there are only two classes.
However, after applying a clustering algorithm, it is discovered that the cat class is composed of 4 groups of cats.
Not we can obtain these insights, but also which groups of cats are similar to which group of dogs and the reasons.

Why and when?

Data labeling is not always possible to be obtained.

Imagine a dataset with 10.000.000.000 records (not very difficult to obtain currently). How much time would be required to label this data?

Why and when?

Suppose you have a dictionary of 10000 words (it can be a dictionary of colors, features, keys, etc). It can be reduced using clustering, grouping words, and obtaining a representative word for each group.

Yes! You have done data compression.

Why and when?

Semi-supervised approach:

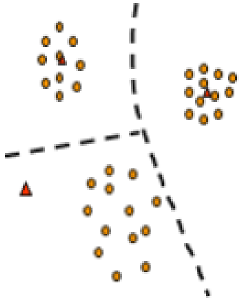
Suppose you have a dataset of Netflix movies. You have the ratings of only a few movies (labeling). How can you assign a label to the other movies? Apply a clustering algorithm to the movies and obtain a label for each of them.

4

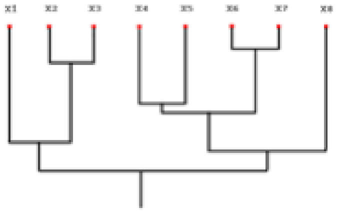
Approaches

Approaches

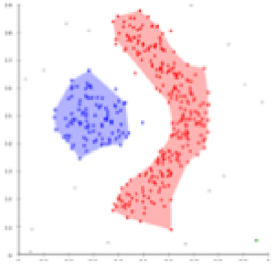
Partitioning-based



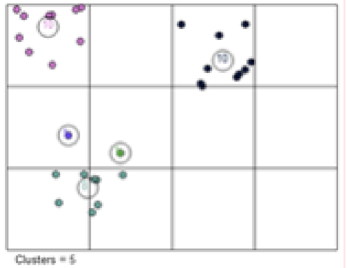
Hierarchical-based



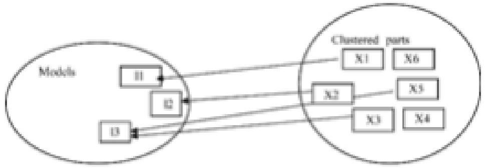
Density-based



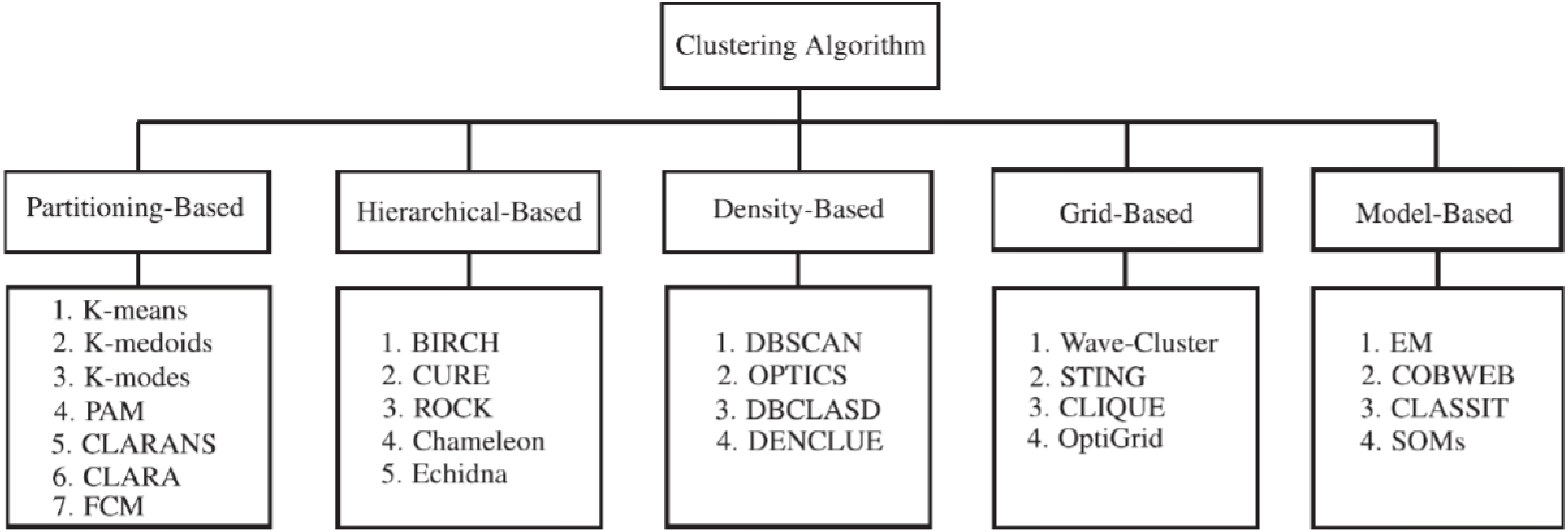
Grid-based



Model-based



Algorithms

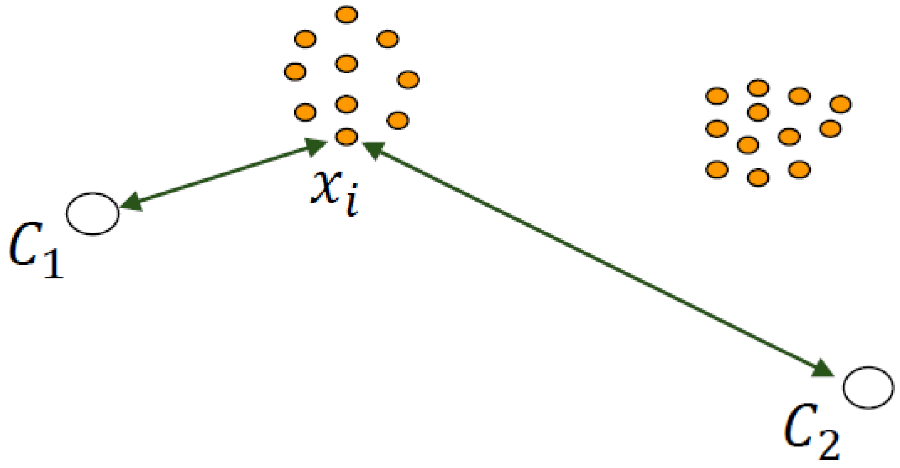

Clustering Algorithm
Partitioning-Based
Hierarchical-Based
Density-Based
Grid-Based
Model-Based
1. K-means
2. K-medoids
3. K-modes
4. PAM
5. CLARANS
6. CLARA
7. FCM
1. BIRCH
2. CURE
3. ROCK
4. Chameleon
5. Echidna
1. DBSCAN
2. OPTICS
3. DBCLASD
4. DENCLUE
1. Wave-Cluster
2. STING
3. CLIQUE
4. OptiGrid
1. EM
2. COBWEB
3. CLASSIT
4. SOMs

K-Means

This algorithm aims to partitioning a set of m objects into k groups/clusters.

Each observation belongs to its nearest cluster.

A similarity metric must be defined



K-Means: Steps

1. Create k random centroids.

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ x_{2,1} & x_{1,2} & \cdots & x_{2,p} \\ \vdots & \vdots & & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,p} \end{bmatrix}$$

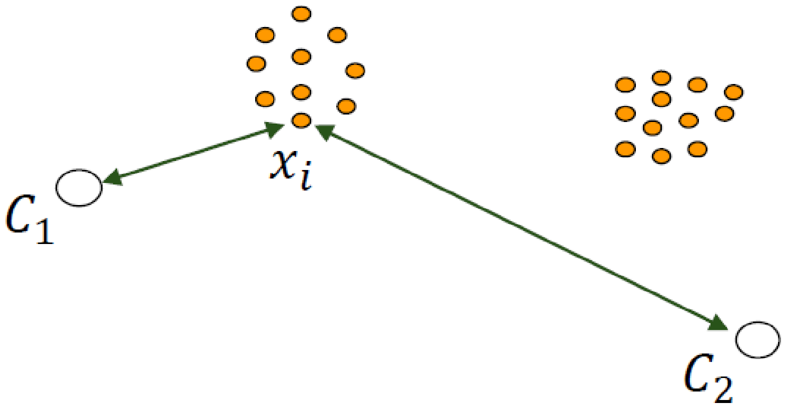
Each centroid is a vector of the shape (1,p).

$$C_1 = \begin{bmatrix} c_{1,1} & \cdots & c_{1,p} \end{bmatrix}$$
$$\vdots$$
$$C_k = \begin{bmatrix} c_{k,1} & \cdots & c_{k,p} \end{bmatrix}$$

p is the number of descriptors (features)

K-Means: Steps

2. Measure the distance of each sample to the cluster centroids.



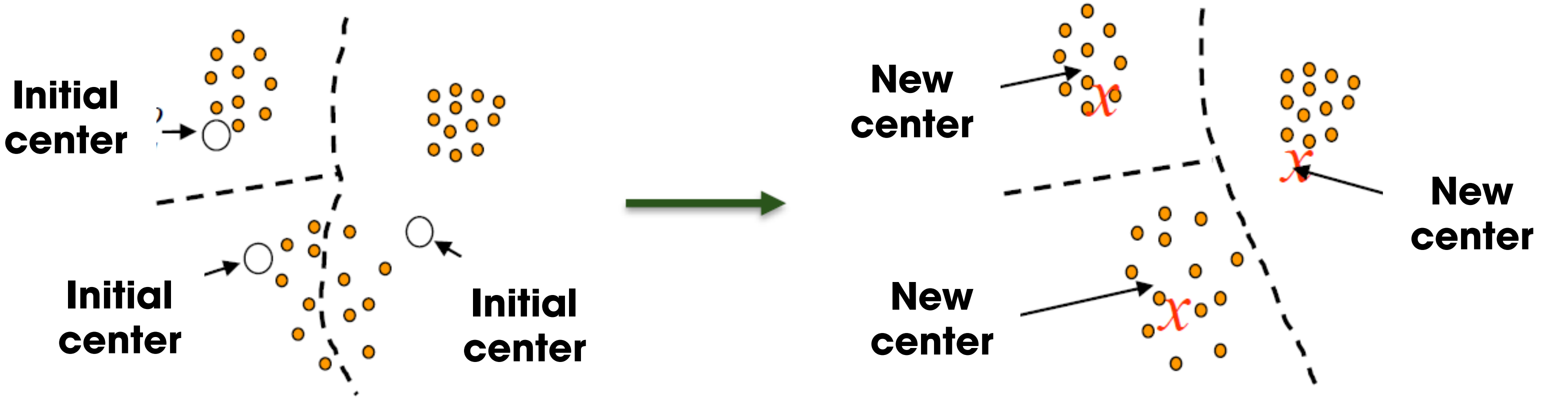
Let's use the Euclidean distance.

$$d_{i,j} = \|x_i - C_j\| = \sqrt{(x_{i,1} - c_{j,1})^2 + \dots + (x_{i,p} - c_{j,p})^2}$$

K-Means: Steps

3. Recompute the centroids using the mean of the samples assigned to that cluster.

$$C_j = \frac{1}{n_j} \sum_{x_i \in C_j} x_i$$

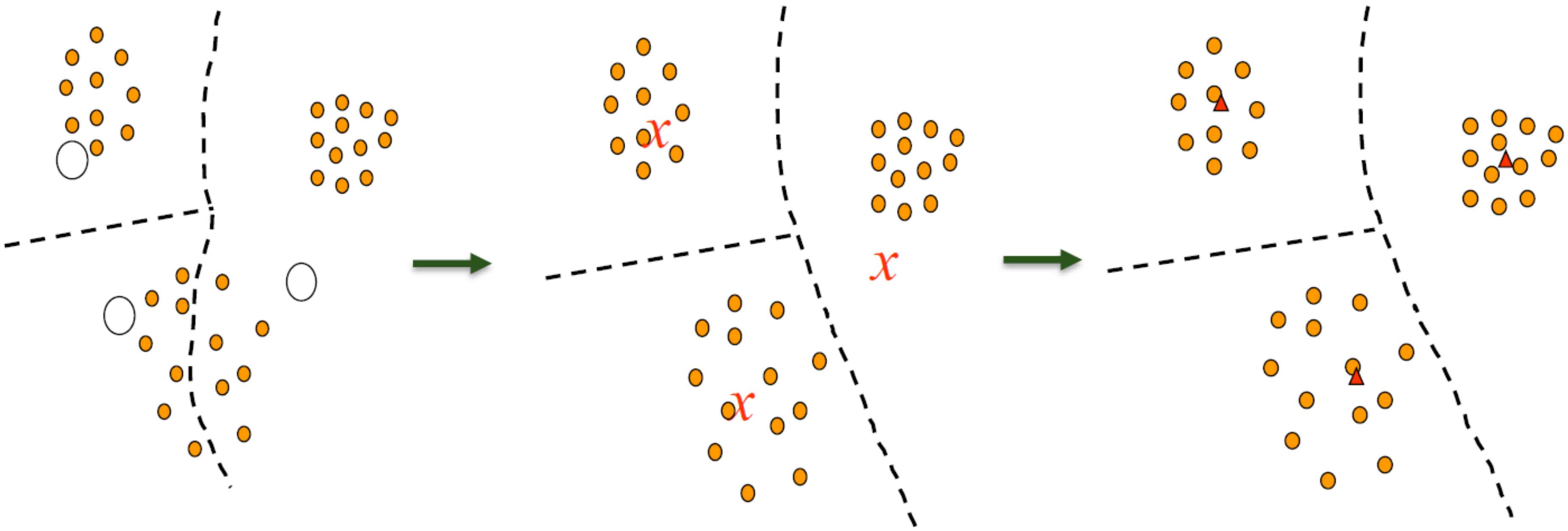


K-Means: Steps

$$S = \sum_{j} \sum_{x_i \in C_j} \|x_i - C_j\|$$

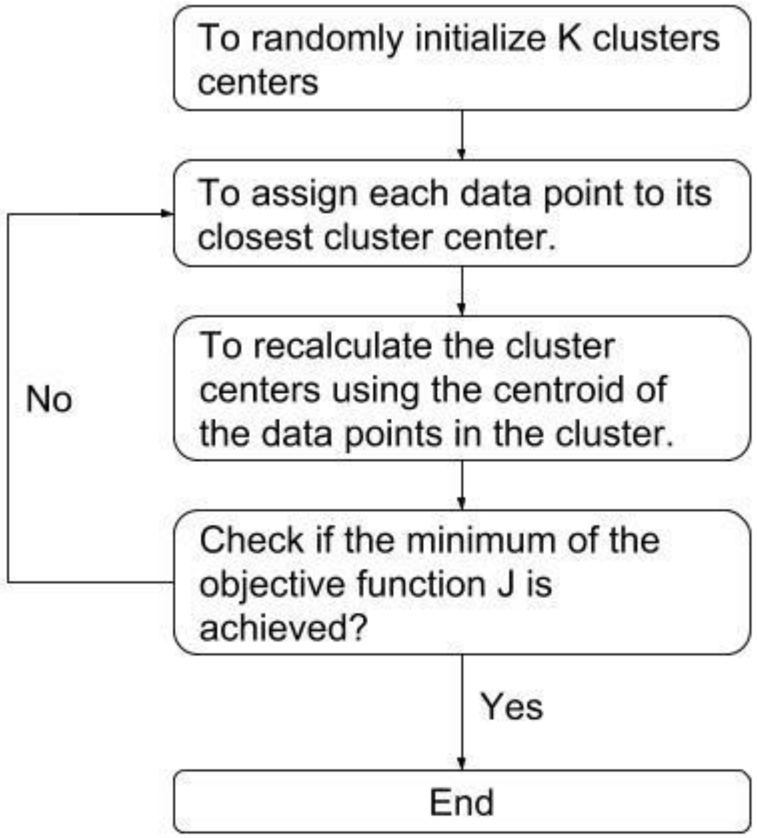
3. Check if the stopping rule is achieved (If S stops decreasing).

K-Means: Graphically



Partitioning-based

K-Means: Flow and Python




```python
Clusters = np.random.rand(nclusters, data.shape[1])
```

```python
for m in range(iterations):
 for i in range(data.shape[0]):
  distances =
    np.linalg.norm(Clusters - data[i,:], axis = 1)
  assign[i] = np.argmin(distances)
```

```python
for j in range(nclusters):
  ind = assign == j
  Clusters[j, :] = np.mean(data[ind, :], axis = 0)
```

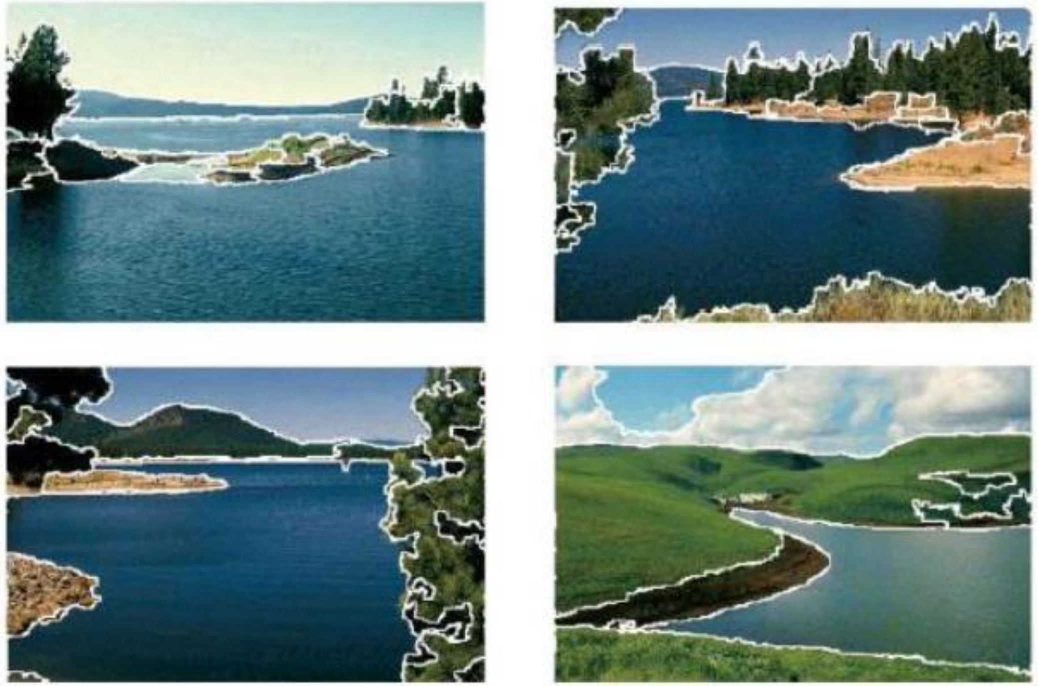
```python
for j in range(nclusters):
  ind = assign == j
  distances =
    np.linalg.norm(Clusters[j, :] - data[ind,:], axis = 1)
  J += np.sum(distances)

  if (abs(J - Jprev) < 0.01):
     return assign, Clusters
```

K-Means: Image segmentation example

Segmentation:

Divide the image in regions/sequences with coherent properties

Let's do Color Segmentation.

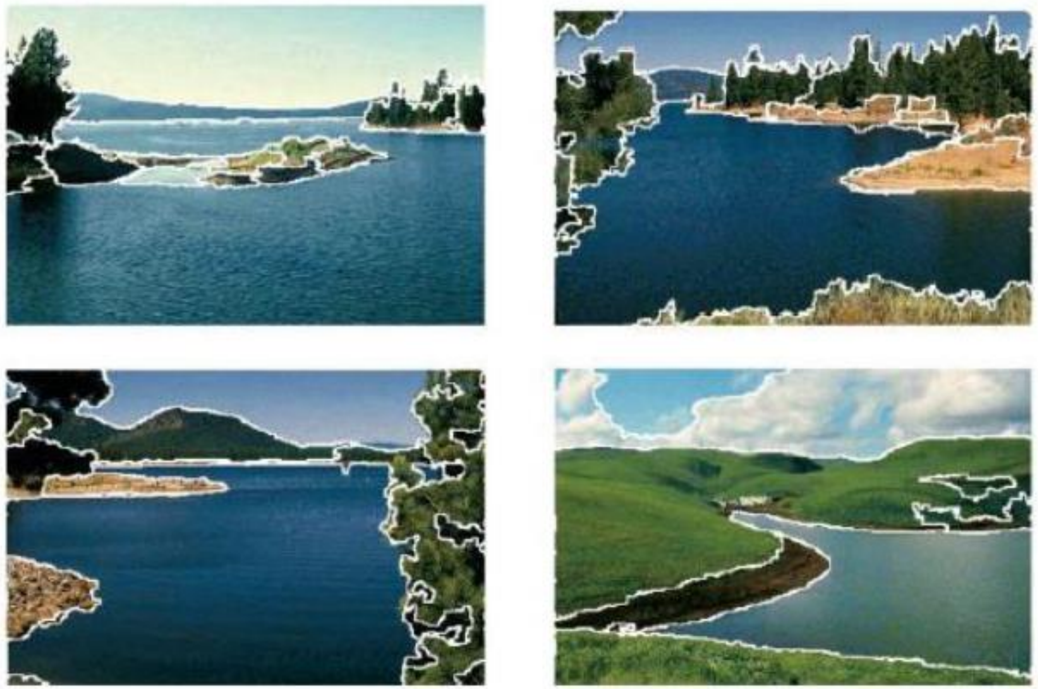


K-Means: Image segmentation example

Segmentation:

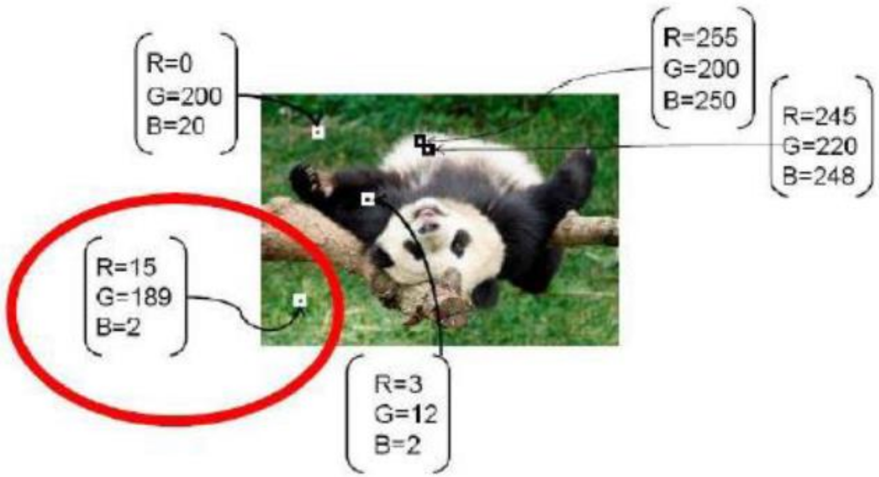
Divide the image in regions/sequences with coherent properties

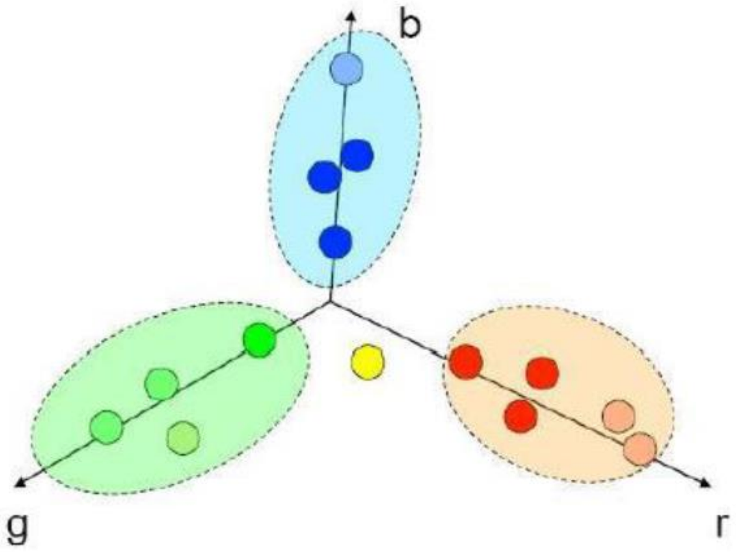
Let's do Color Segmentation.



K-Means: Image segmentation example

Each pixel becomes a sample of three values (Red, Green, Blue).





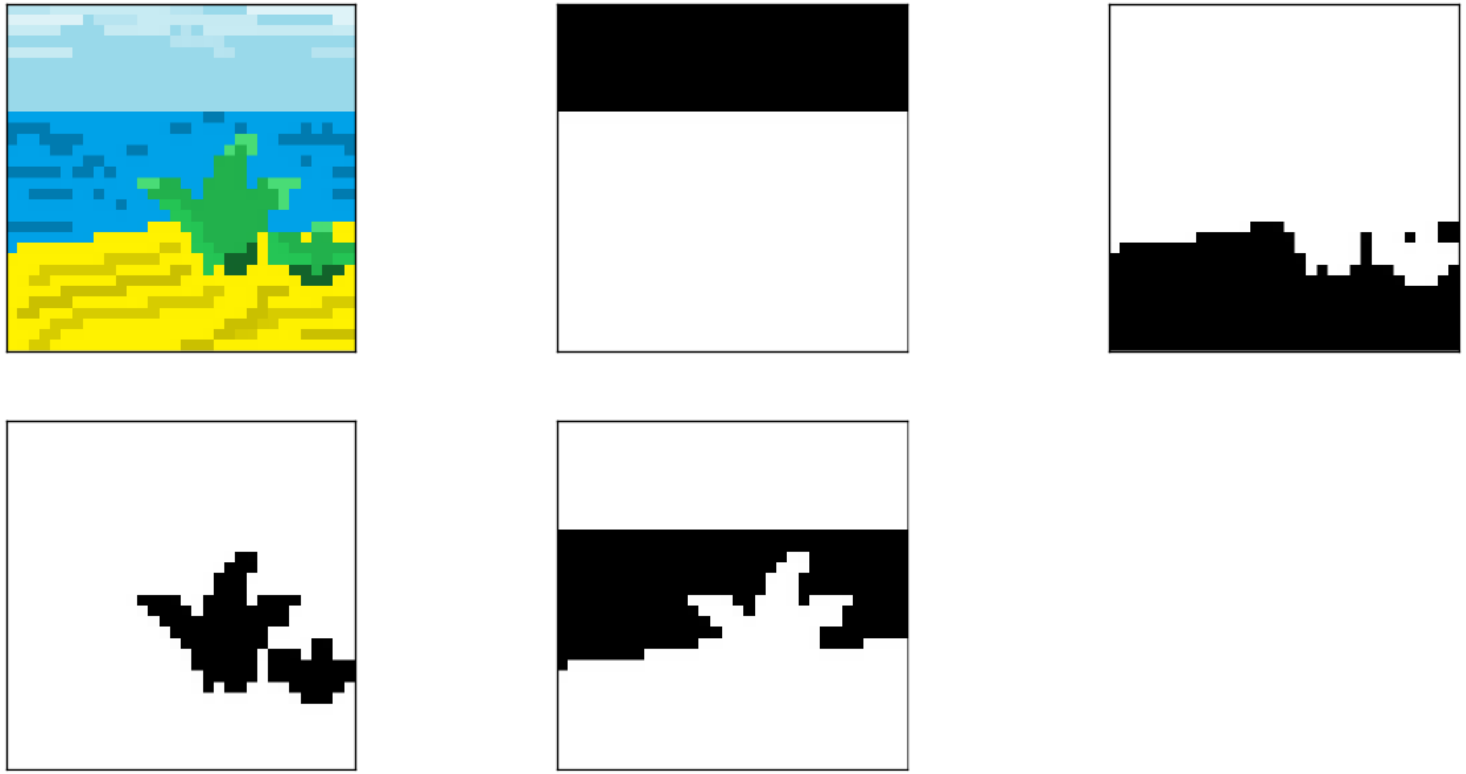
K-Means: Image segmentation example

To group regions is also important
to add pixel's position information.

$$X = \begin{bmatrix} p_1 & R_1 & G_1 & B_1 \\ p_2 & R_2 & G_2 & B_2 \\ \vdots & \vdots & \vdots & \vdots \\ p_N & R_N & G_N & B_N \end{bmatrix}$$

Partitioning-based

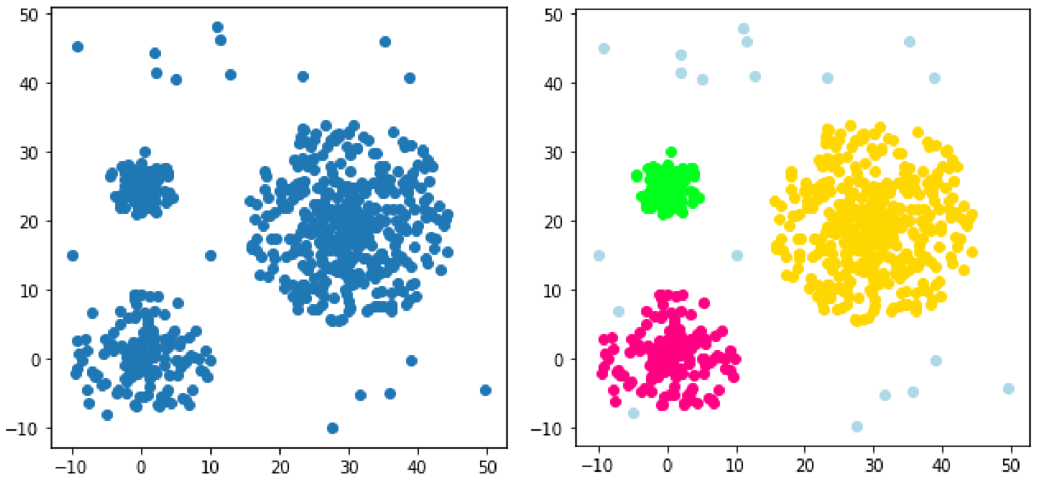
K-Means: Image segmentation example



DBSCAN

This algorithm estimates the density around each sample.

The idea is to create clusters once the density around a group of points exceeds a threshold.



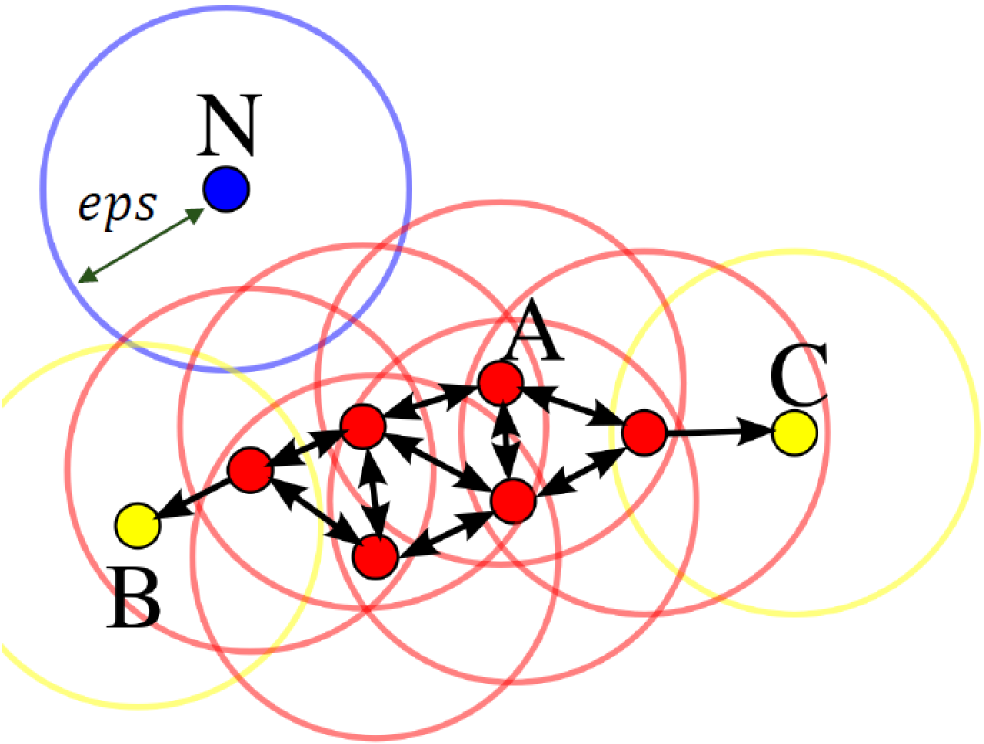
H

DBSCAN: Point classification

1. Nuclear points (A)
2. Boundary points (B, C)
3. Noisy points (N)

Two parameters:

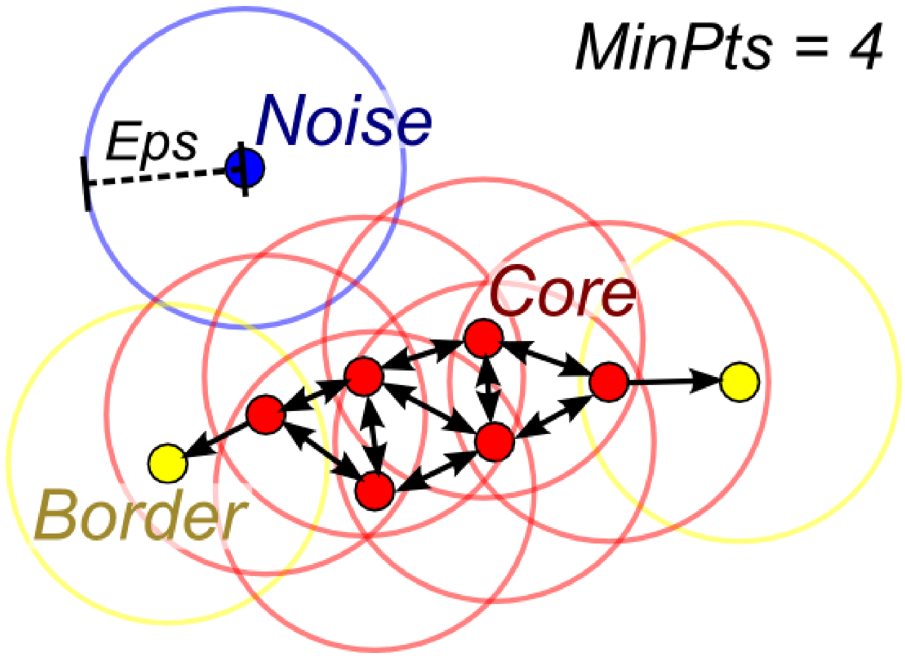
- eps: analysis radius.
- MinPts: Min number of points in the analysis radius.



DBSCAN: Steps

1. Select a random sample.

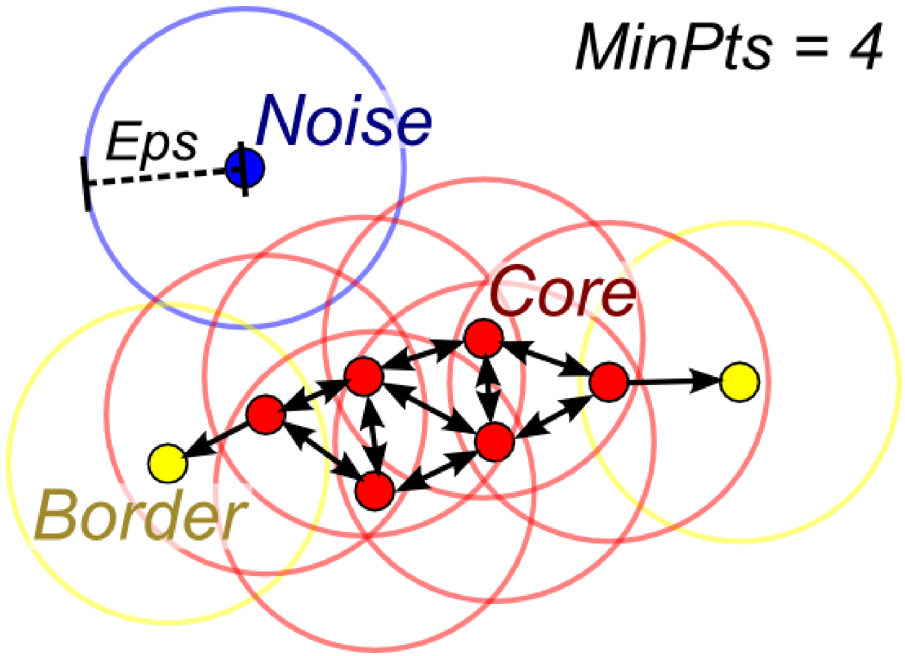
2. Check if there is at least a minimum number of points (MinPts) in the analysis radius (eps).



DBSCAN: Steps

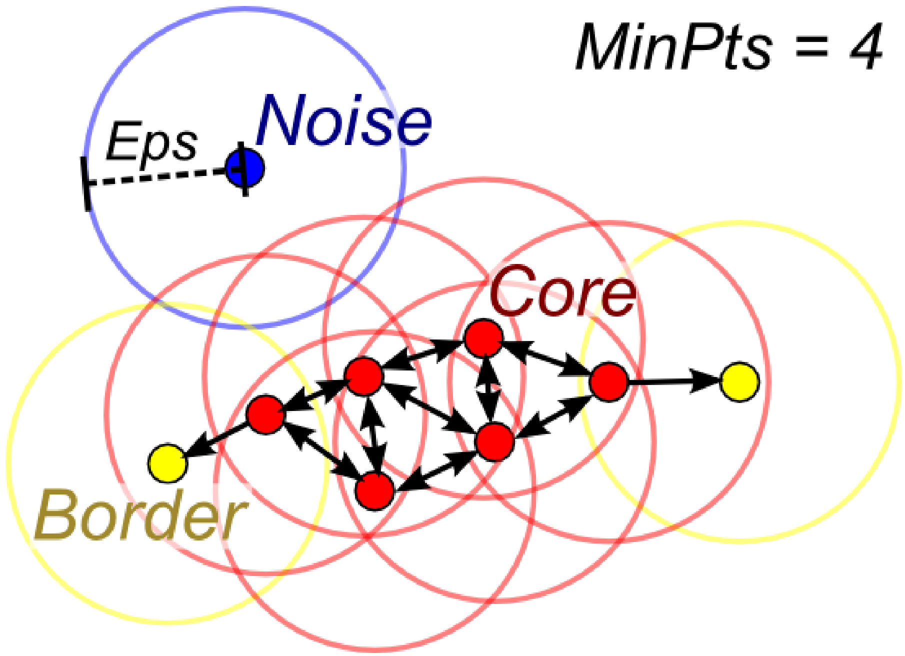
3. If the condition is fulfilled a new cluster is created from this random point.

4. Iteratively check the points in the analysis radius.

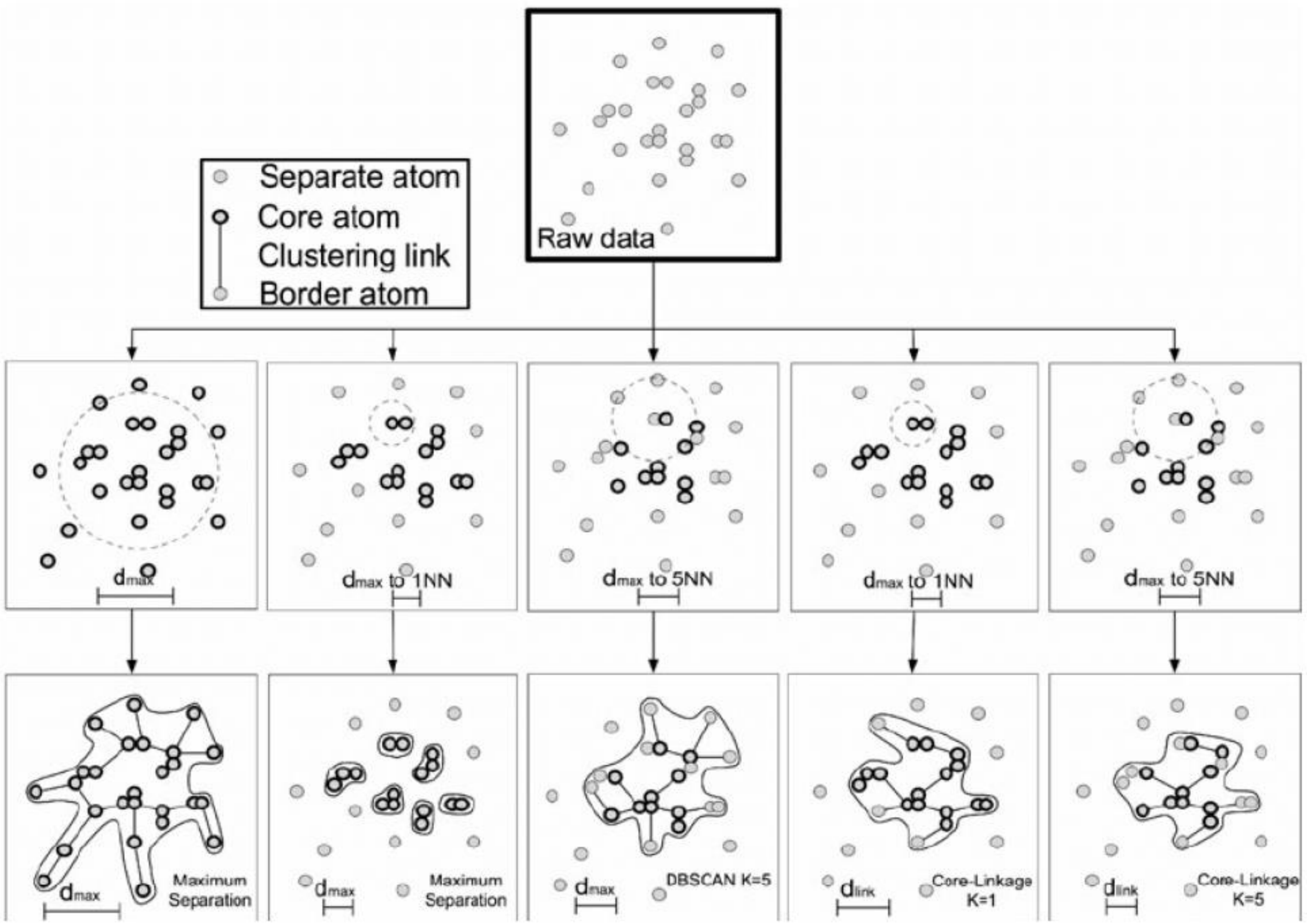


DBSCAN: Steps

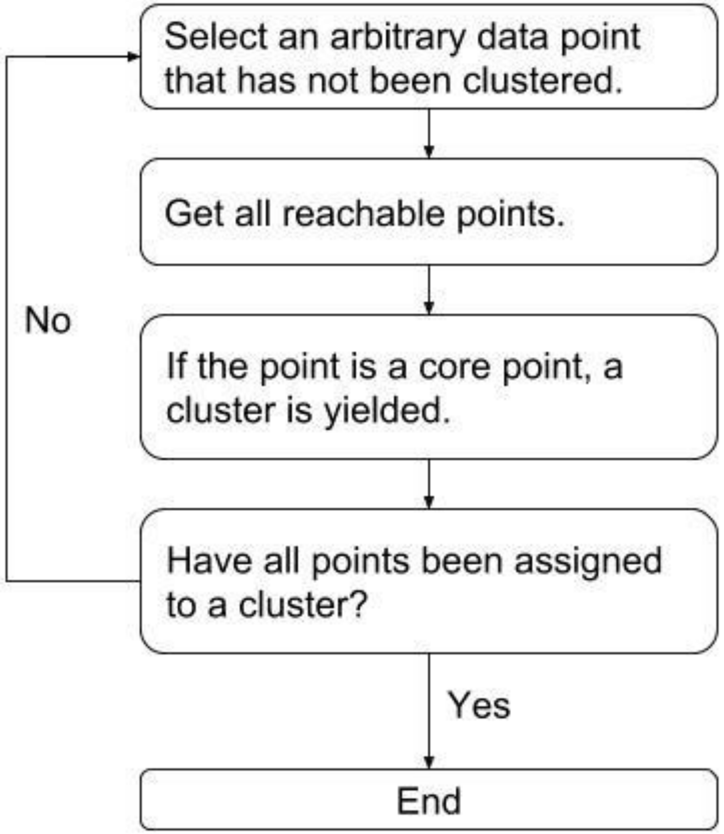
5. The new cluster is prolongated until no more points can be added to it.



DBSCAN: Graphically



DBSCAN: Flow and Python




```
while (len(clusteredPoints) != len(allPoints)):
 noClusteredPoints = np.setdiff1d(allPoints, clusteredPoints)
 index = np.random.choice(noClusteredPoints)
```

```
clusterIndices = np.array([index])
clusterIndices, alreadyChecked =
  self.CheckNeighbors(index, clusterIndices, alreadyChecked)
```

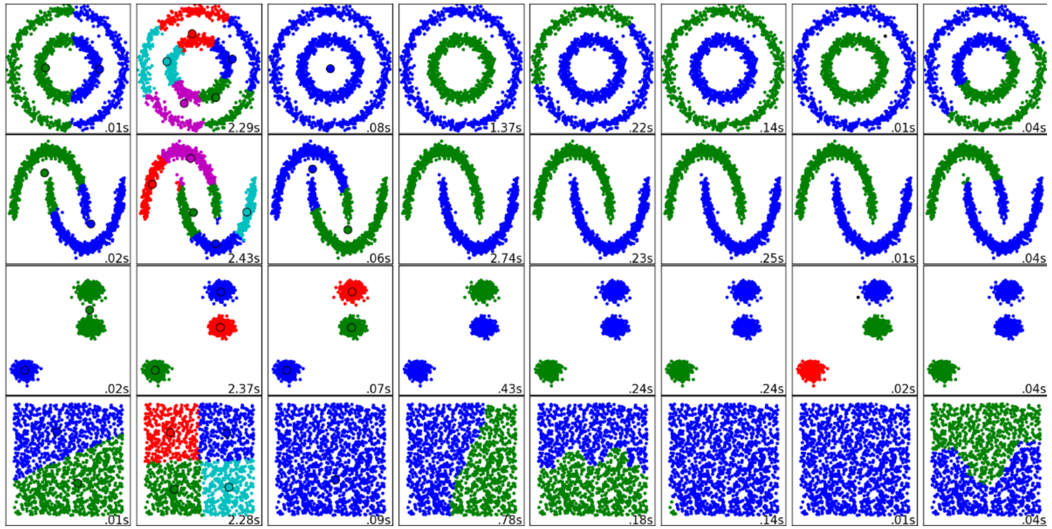
```
if (len(clusterIndices) >= self.minPts): #Nuclear Points
  self.assign[clusterIndices] = self.nclusters
  self.nclusters = self.nclusters + 1
else:
  self.assign[index] = -1 #Outlier, Noisy Point
```

5

Validation

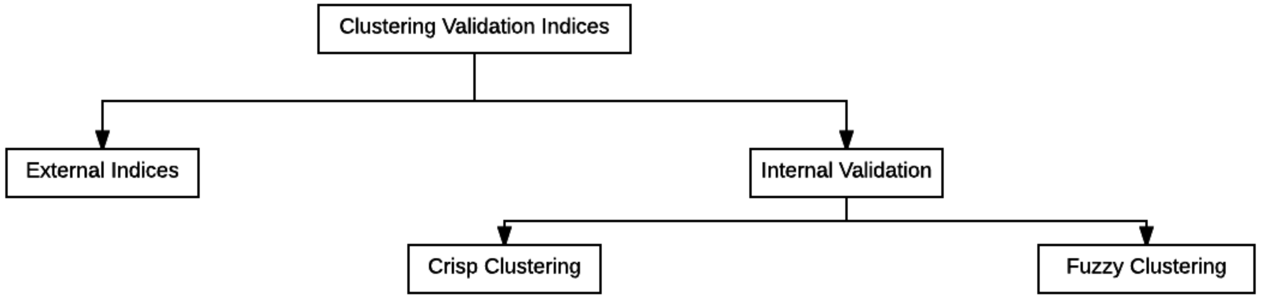
Validation problem

Since there is no expected structure of the data, the evaluation of the quality of the data partitions becomes complicated into a task.



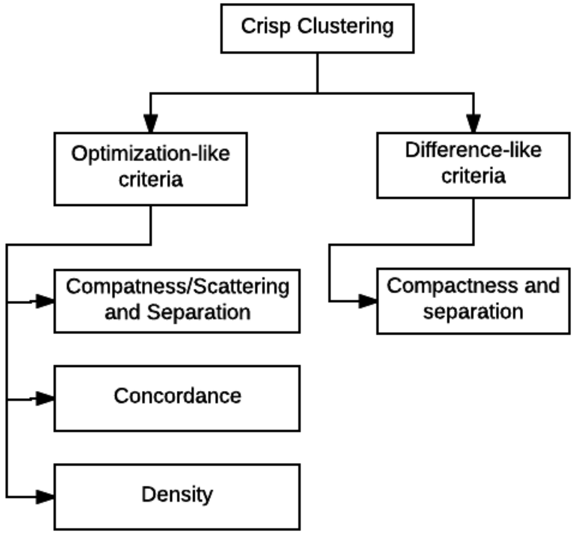
Validation indices

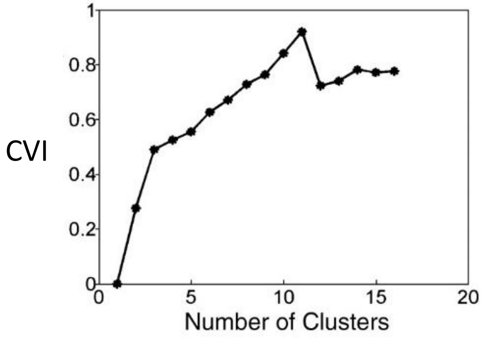
Validation indices are classified into two types: external criteria and internal criteria.

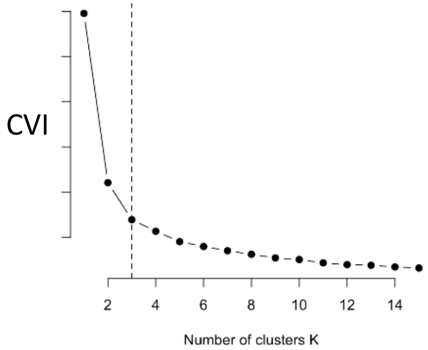


Validation indices

Several studies have compared clustering validation indices (CVIs) used in crisp clustering. None of these studies have been conclusive.

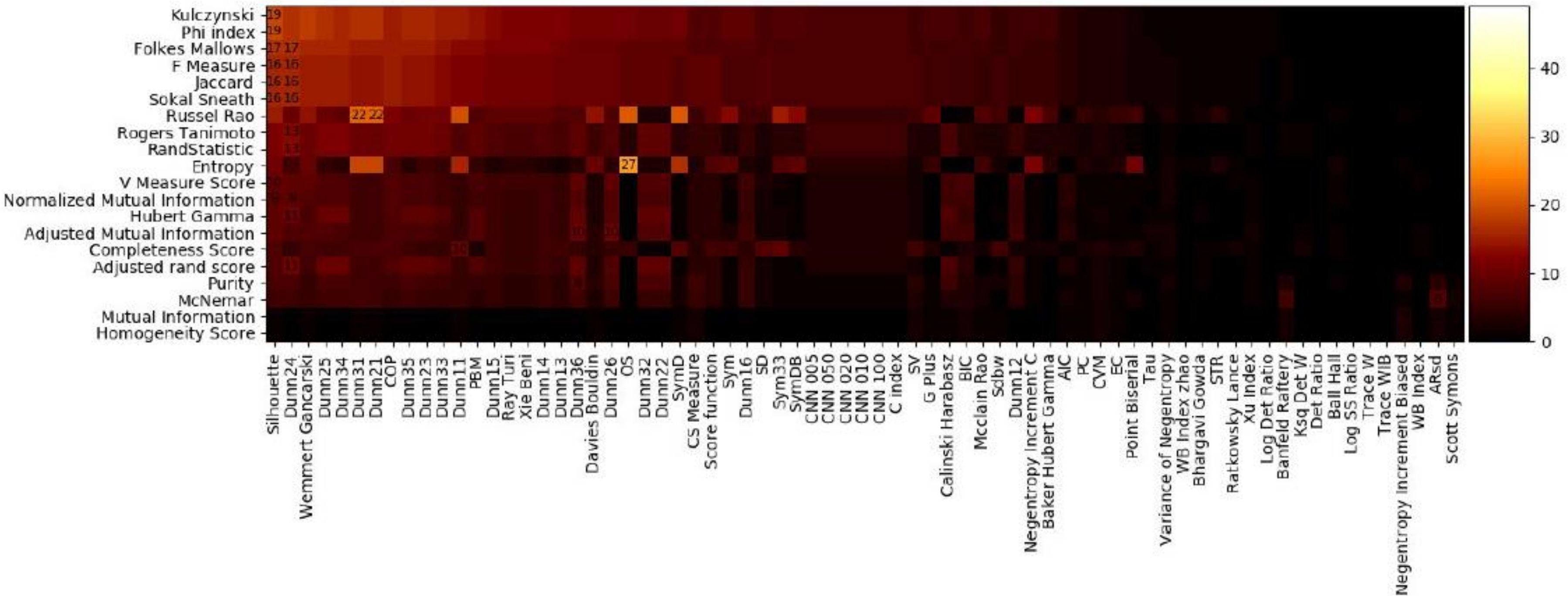






Work in progress

Validation library: 49 Datasets, 20 external indices and 72 external indices



Work in progress

Validation library:

https://github.com/williamegomez/Clustering-Validation-Indices