INTRODUCING CLUSTERING ALGORITHMS

Scott O'Hara

Metrowest Developers Machine Learning Group

03/03/2021

REFERENCES

The material for this talk is primarily drawn from the notes, slides and lectures of the courses and book below:

MOSTLY:

DS 5230 Unsupervised Machine Learning and Data Mining – Fall 2018

Northeastern University, Prof. Jan-Willem van de Meent

https://www.khoury.neu.edu/home/jwvdm/teaching/ds5230/fall2018/

A FEW THINGS:

Applied Machine Learning in Python

University of Michigan, Prof. Kevin Collins Thompson

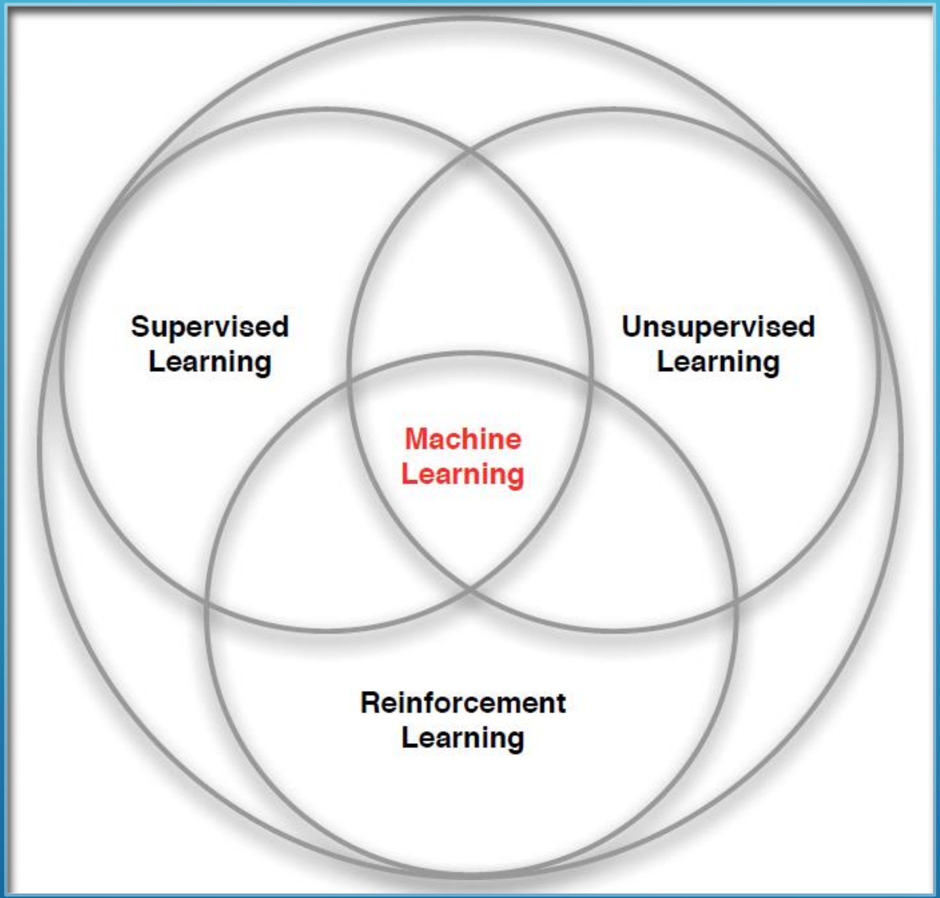
https://www.coursera.org/learn/python-machine-learning/home/welcome

The Hundred-Page Machine Learning Book (Ch. 9)

Andriy Burkov

http://themlbook.com/

3 TYPES OF MACHINE LEARNING



Supervised Learning – Learn a function from labeled data that maps input attributes to an output label e.g., linear regression, decision trees, SVMs.

Unsupervised Learning – Learn patterns in unlabeled data e.g., principle component analysis or clustering algorithms such as K-means, HAC, or Gaussian mixture models.

Reinforcement Learning – An agent learns to maximize rewards while acting in an uncertain environment.

Credit: image borrowed from lecture slides David Silver, DeepMind, "Introduction to Reinforcement Learning

WHAT IS UNSUPERVISED LEARNING?

- Unsupervised learning involves tasks that operate on datasets without labeled responses or target values.
- The goal is to discover interesting structure or information in the dataset.

Reference: *Applied Machine Learning in Python*. Coursera. University of Michigan, Prof. Kevin Collins Thompson

APPLICATIONS OF UNSUPERVISED LEARNING

- Visualize structure of a complex dataset.
- Density estimation to predict probabilities of events.
- Compress and summarize the data.
- Extract features for supervised learning.
- Discover important clusters or outliers.

Reference: *Applied Machine Learning in Python*. Coursera. University of Michigan, Prof. Kevin Collins Thompson

FOUR KINDS OF UNSUPERVISED LEARNING

1. **Density Estimation**
   - Model the probability density function of the unknown probability distribution from which the dataset has been drawn.
2. **Dimensionality Reduction**
   - Finds an approximate version of a dataset using fewer features while retaining some meaningful properties of the original data.
3. **Outlier Detection**
   - Detect the examples in the dataset that are very different from what a typical example in the dataset looks like.
4. **Clustering**
   - The task of grouping a set of objects in such a way that objects in the same group (called a **cluster**) are more like each other than to those in other groups (clusters).

References:

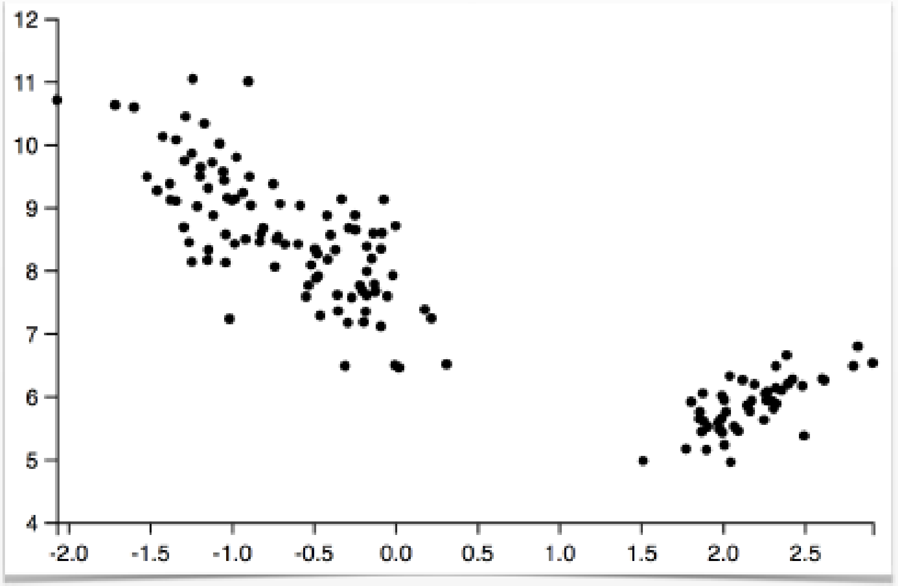
The Hundred-Page Machine Learning Book. Andriy Burkov.

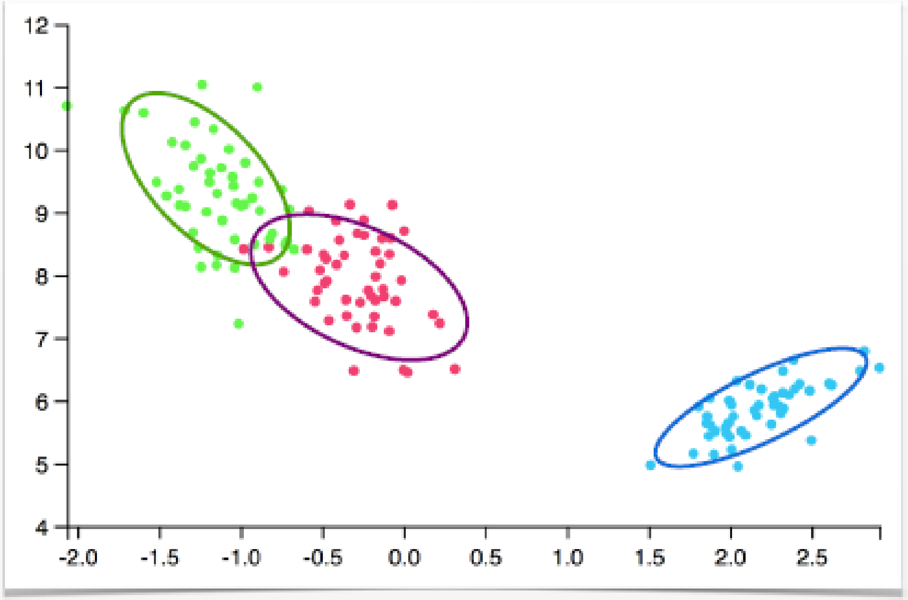
Applied Machine Learning in Python. Coursera. University of Michigan, Prof. Kevin Collins Thompson

Cluster analysis, https://en.wikipedia.org/w/index.php?title=Cluster_analysis&oldid=1002271612 (last visited Jan. 27, 2021).

Dimensionality reduction, https://en.wikipedia.org/w/index.php?title=Dimensionality_reduction&oldid=1002754996 (last visited Jan. 27, 2021).

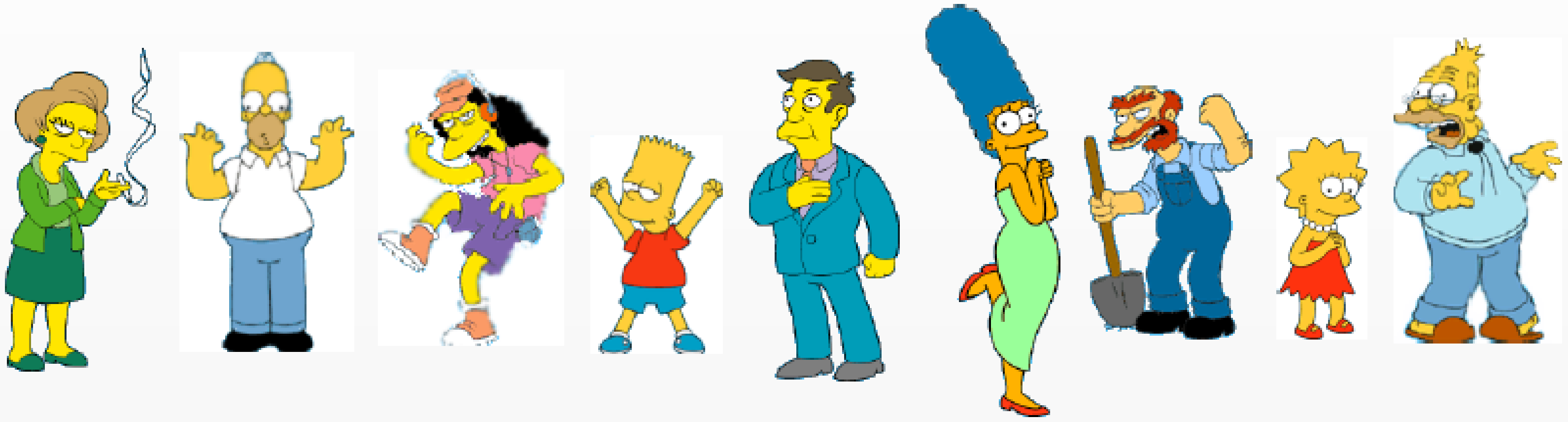
Clustering



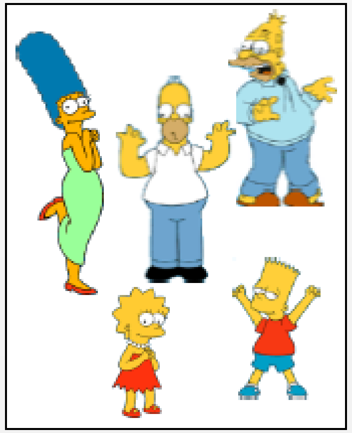


- Unsupervised learning (no labels for training)
- Group data into similar classes that
  - Maximize *inter-cluster* similarity
  - Minimize *intra-cluster* similarity

What is a natural grouping?

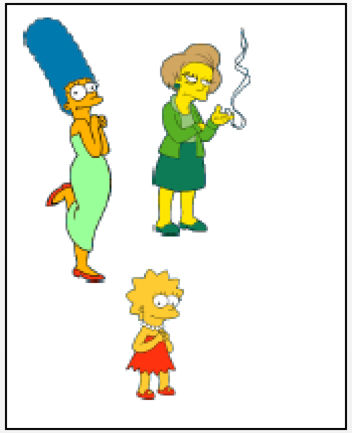


Choice of clustering criterion can be task-dependent

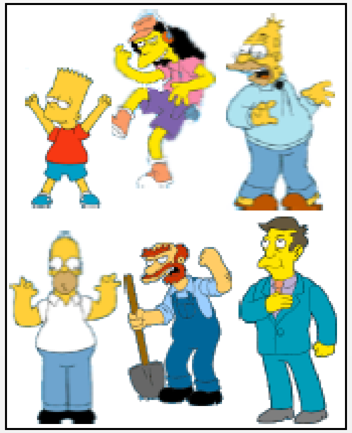


Simpson's Family

School Employees



Females

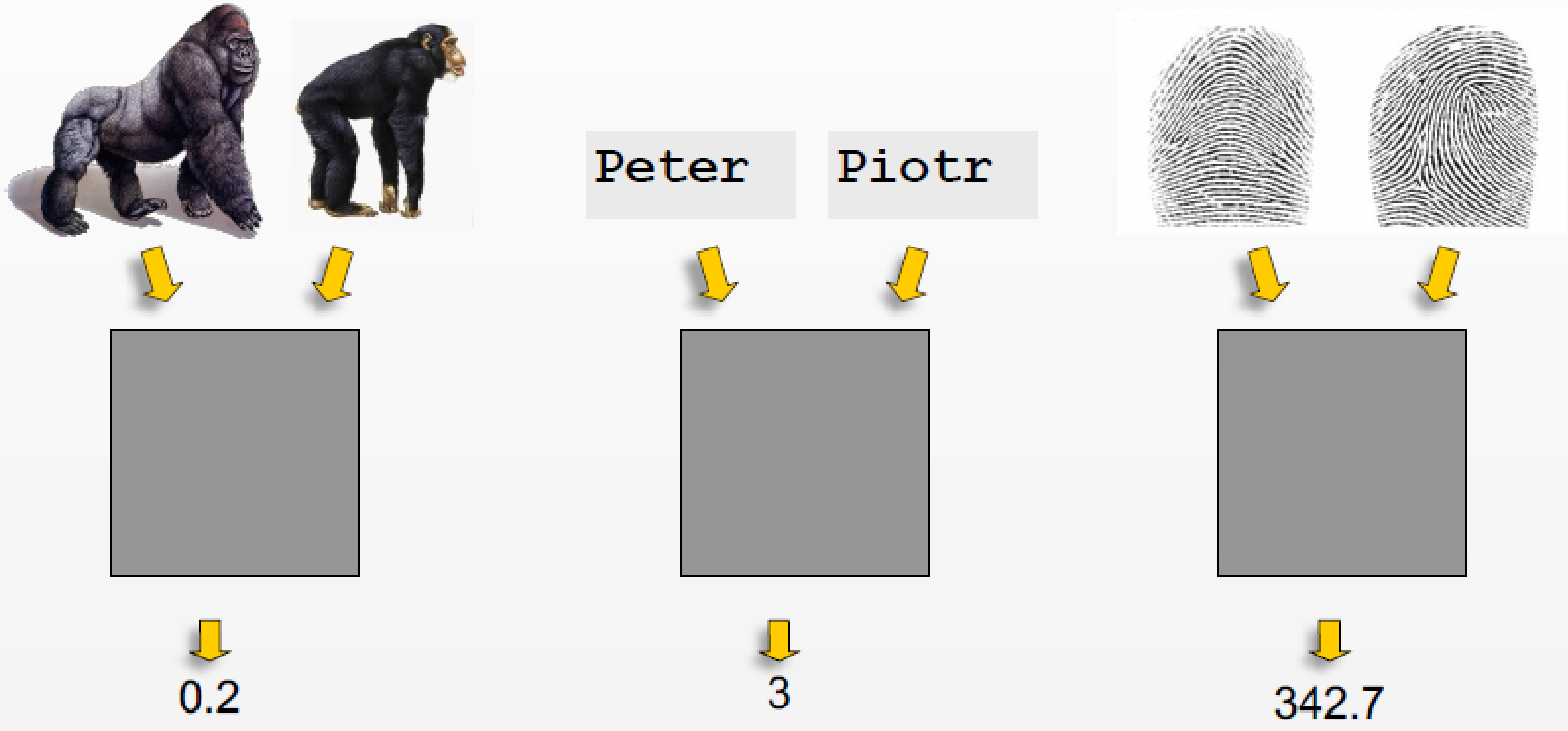


Males

What is Similarity?

Can be hard to define, but we know it when we see it.

Defining Distance Measures



Dissimilarity/distance: $d(\boldsymbol{x}_1, \boldsymbol{x}_2)$
Similarity: $s(\boldsymbol{x}_1, \boldsymbol{x}_2)$

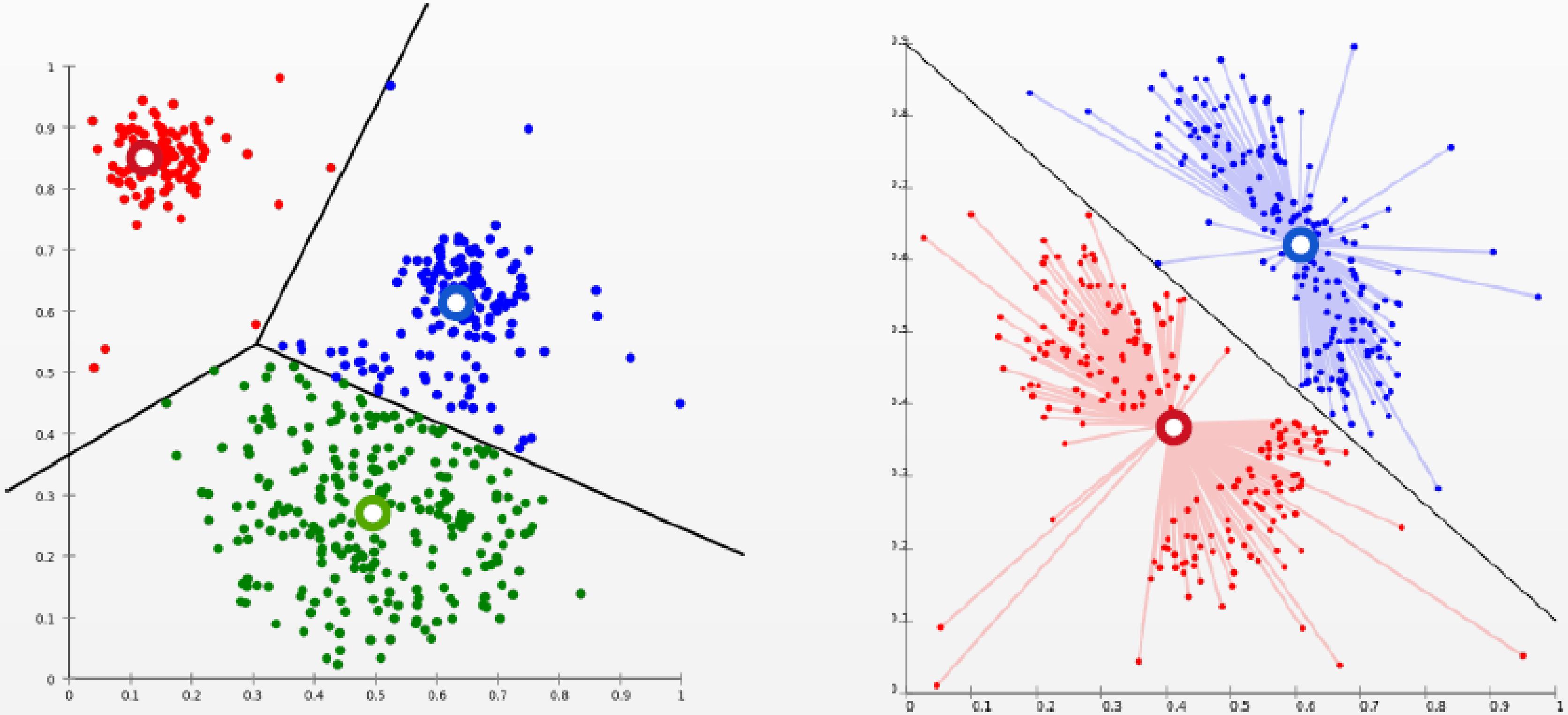
$\Big\}$ Proximity: $p(\boldsymbol{x}_1, \boldsymbol{x}_2)$

Distance Measures

- Euclidean Distance $$\sqrt{(\sum_{i=1}^{k}(x_i - y_i)^2)}$$
- Mahattan Distance $$\sum_{i=1}^{k}|x_i - y_i|$$
- Minkowski Distance $$\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{\frac{1}{q}}$$

Four Types of Clustering

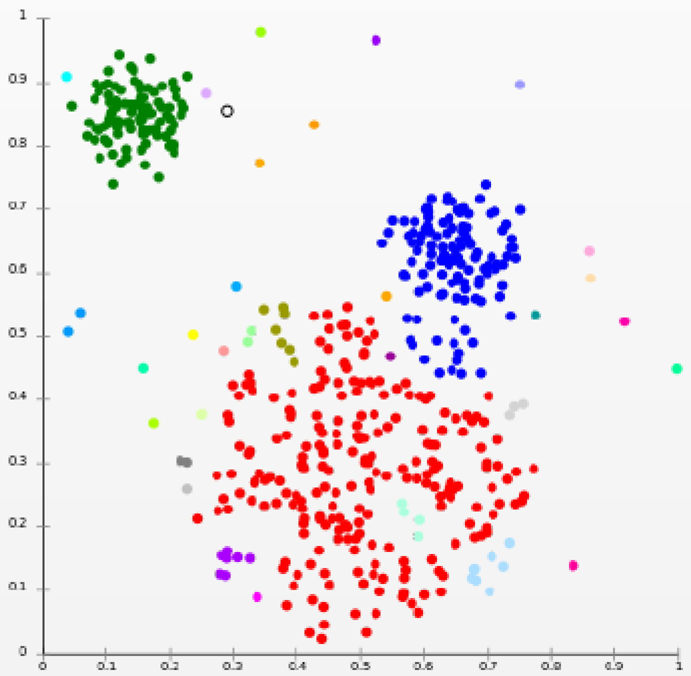
1. *Centroid-based (K-means, K-medoids)*

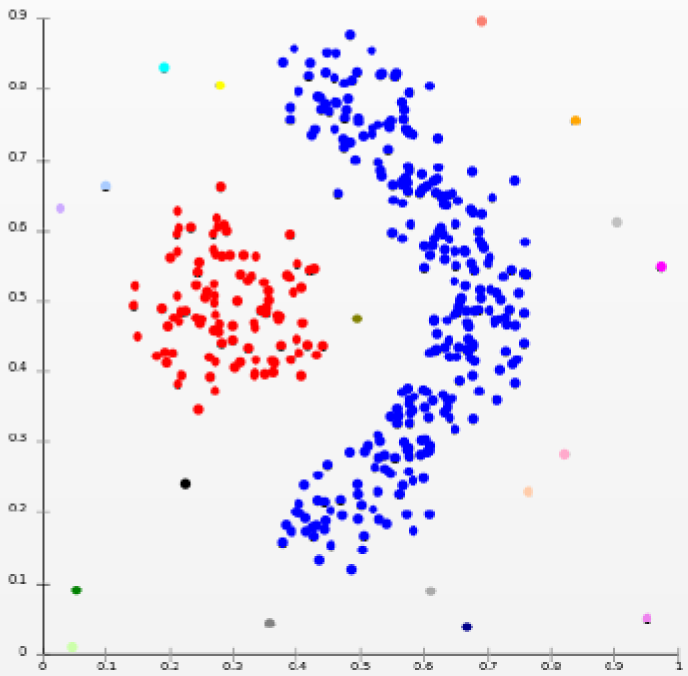


Notion of Clusters: Voronoi tesselation

Four Types of Clustering

2. *Connectivity-based (Hierarchical)*

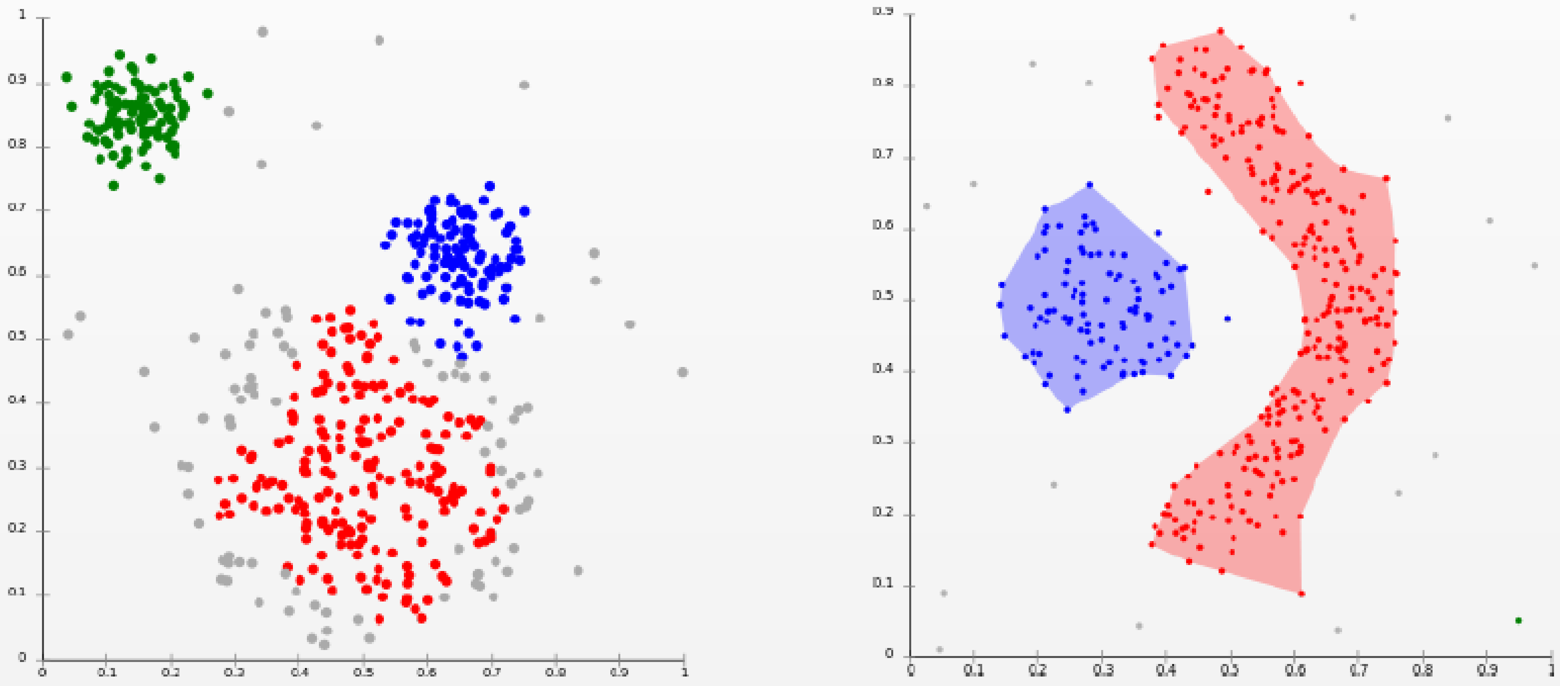




Notion of Clusters: Cut off dendrogram at some depth

Four Types of Clustering

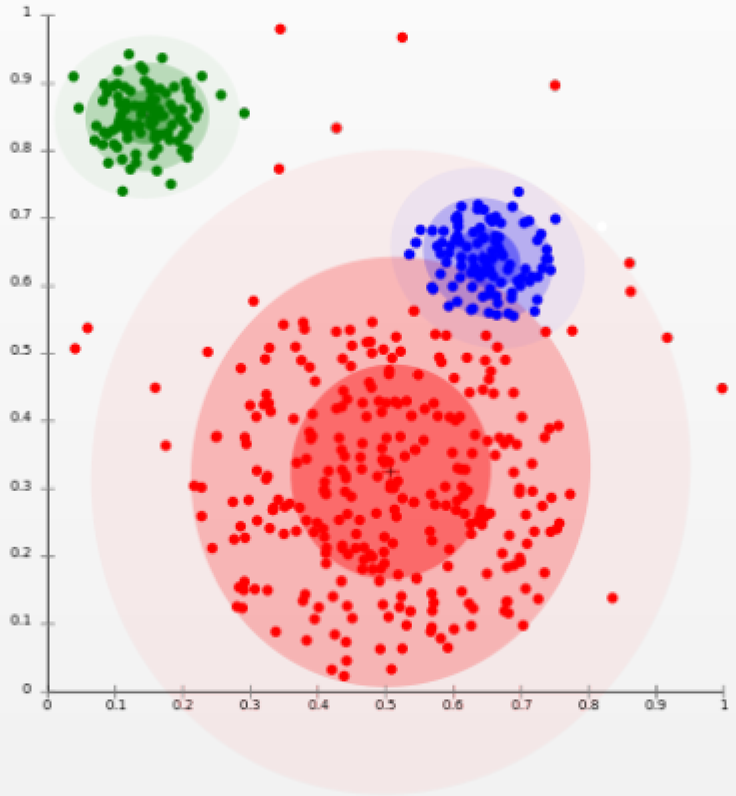
3. *Density-based (DBSCAN, OPTICS)*

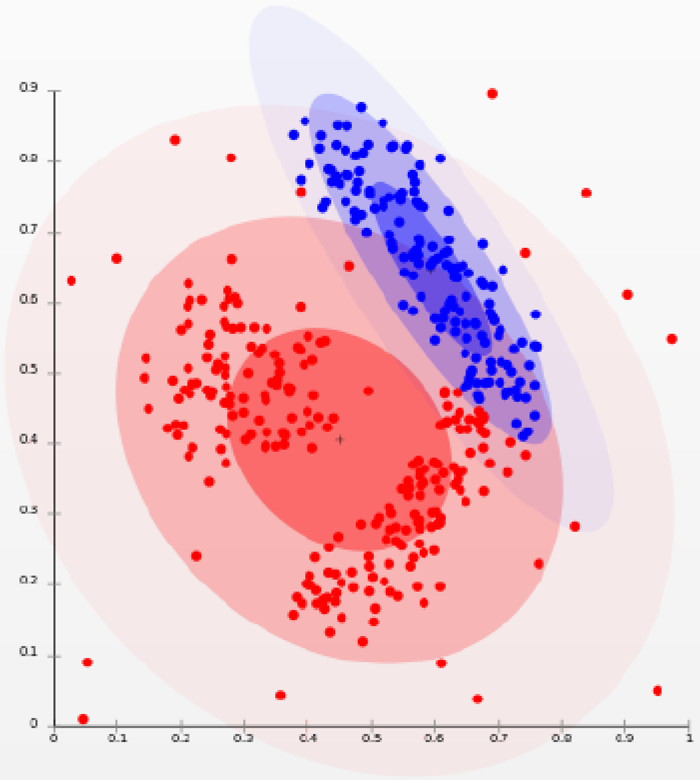


Notion of Clusters: Connected regions of high density

Four Types of Clustering

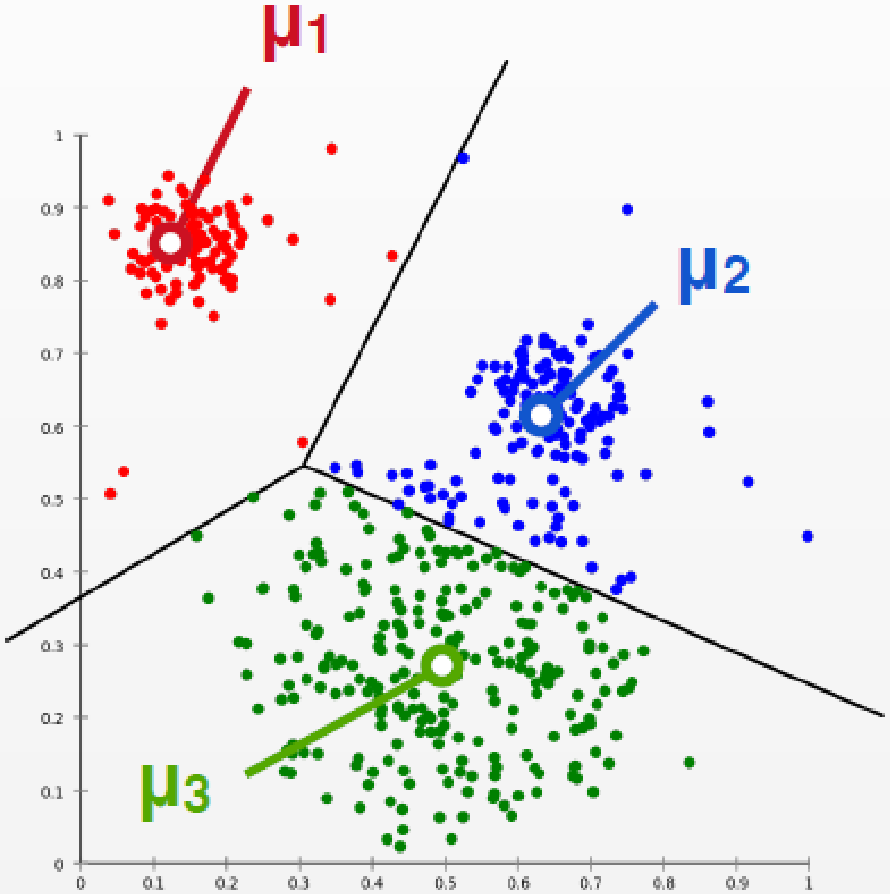
4. *Distribution-based (Mixture Models)*





Notion of Clusters: Distributions on features

K-means Clustering



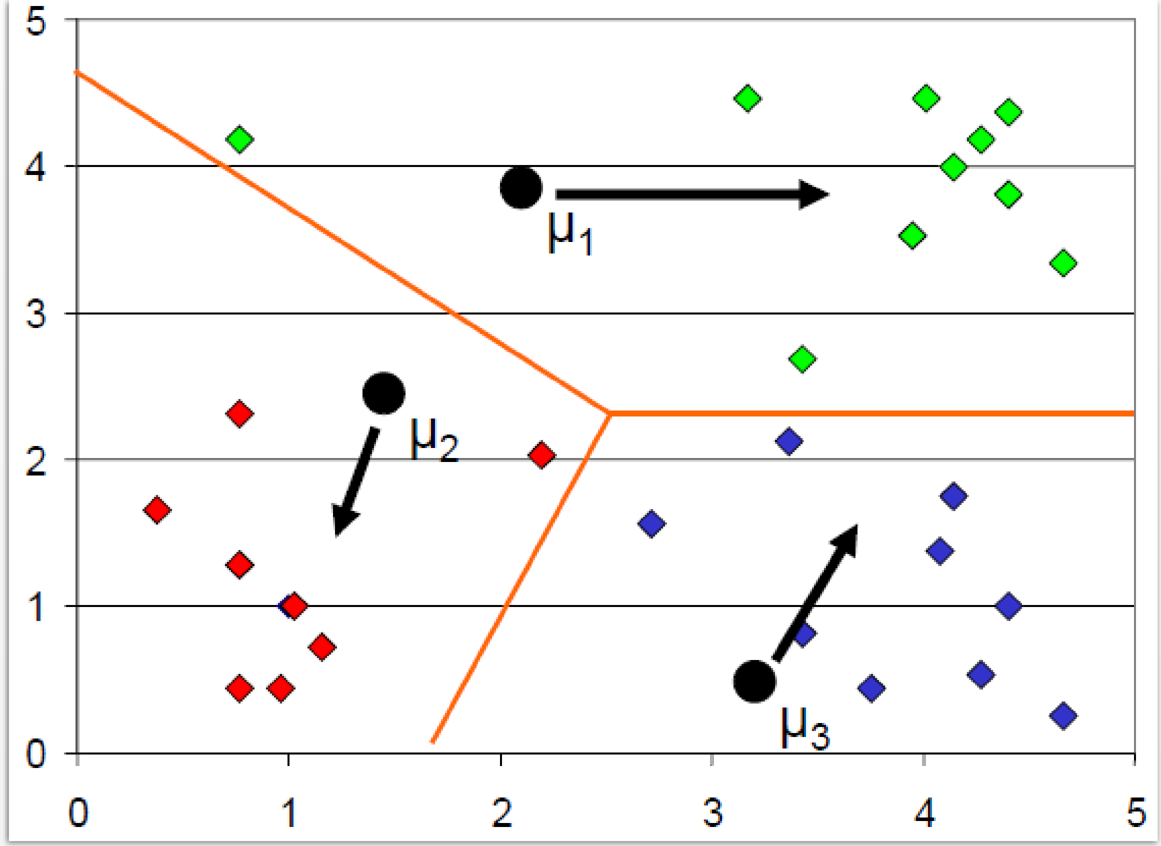
Idea: Minimize Sum of Squares

$$SSE_i = \sum_{\mathbf{x} \in C_i} \| \mathbf{x} - \mu_i \|^2$$

$$SSE = \sum_{j=1}^{K} SSE_j$$

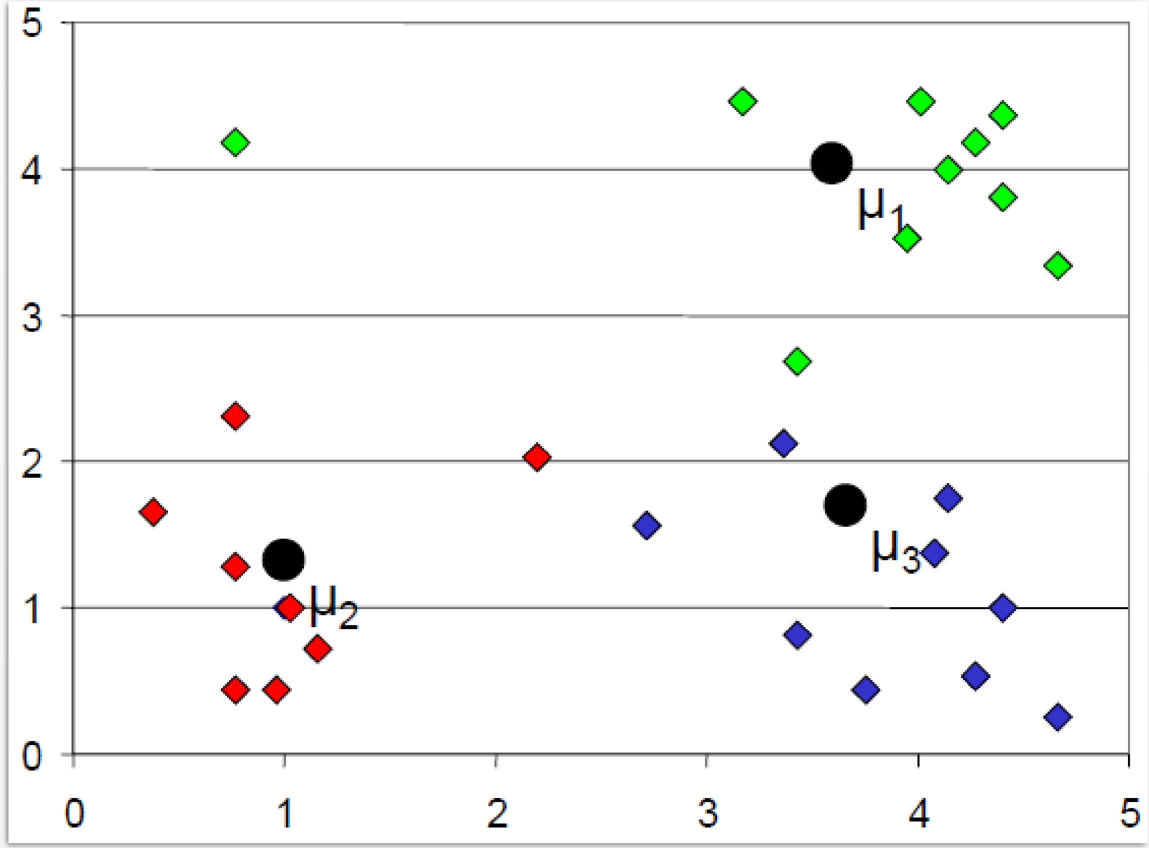
Use *heuristic* search
(as in hierarchical case)

K-means Clustering



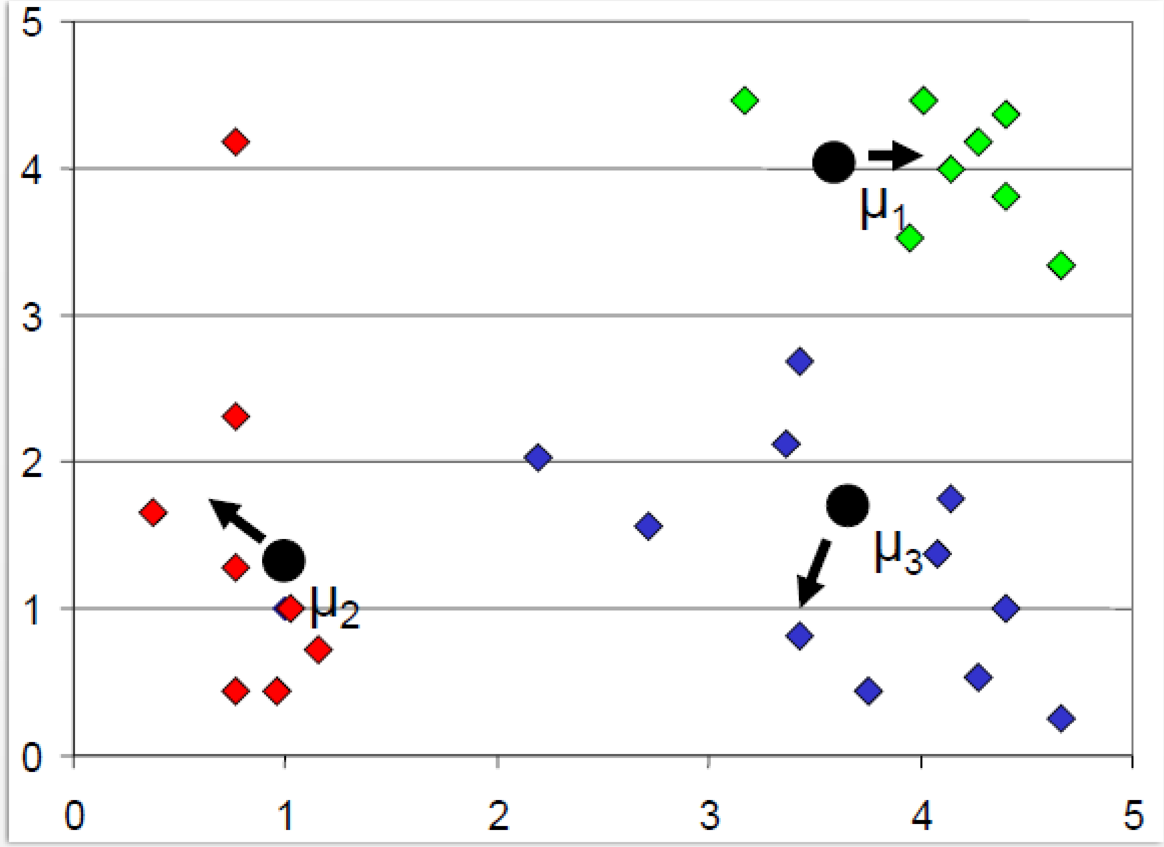
Assign each point to closest centroid,
then update centroids to average of points

K-means Clustering



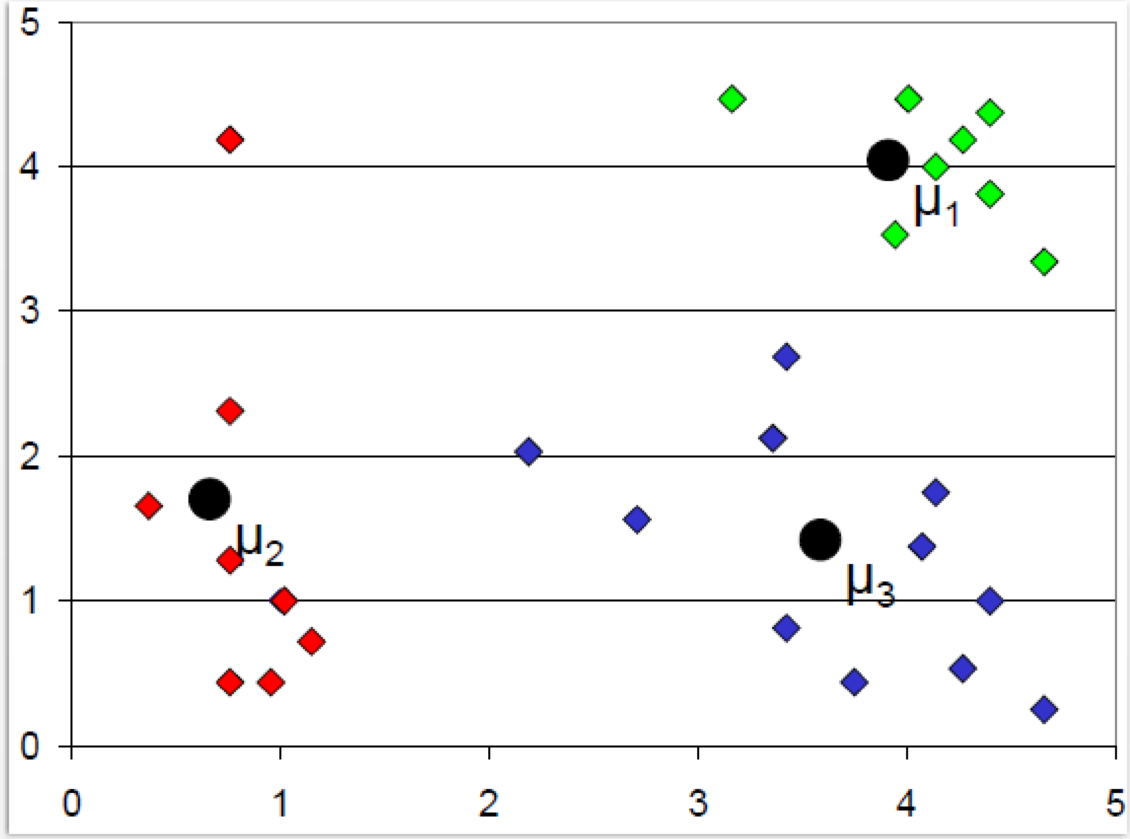
Assign each point to closest centroid, then update centroids to average of points

K-means Clustering



Repeat until convergence
(no points reassigned, means unchanged)

K-means Clustering



Repeat until convergence
(no points reassigned, means unchanged)

K-means Recap ...

- Randomly initialize *k* centers
  - $\mu^{(0)} = \mu_1^{(0)}, \ldots, \mu_k^{(0)}$

Iterate t = 0, 1, 2, ...

- **Classify**: Assign each point $j \in \{1, \ldots m\}$ to nearest center:
  - $C^{(t)}(j) \leftarrow \arg \min_{i=1,\ldots,k} \|\mu_i^{(t)} - x_j\|^2$
- **Recenter**: μ_i becomes centroid of its points:
  - $\mu_i^{(t+1)} \leftarrow \arg \min_{\mu} \sum_{j:C^{(t)}(j)=i} \|\mu - x_j\|^2 \qquad i \in \{1, \ldots, k\}$
  - Equivalent to $\mu_i \leftarrow$ average of its points!

What is K-means optimizing?

- Potential function F($\boldsymbol{\mu}$,C) of centers $\boldsymbol{\mu}$ and point allocations C:

$$F(\mu, C) = \sum_{j=1}^{m} ||\mu_{C(j)} - x_j||^2$$

$$= \sum_{i=1}^{k} \sum_{j:C(j)=i} ||\mu_i - x_j||^2$$

- Optimal K-means:
  - $\min_{\boldsymbol{\mu}} \min_C F(\boldsymbol{\mu}, C)$

K-means algorithm

- Optimize potential function:

$$\min_{\mu}\min_{C} F(\mu, C) = \min_{\mu}\min_{C} \sum_{i=1}^{k} \sum_{j:C(j)=i} ||\mu_i - x_j||^2$$

- **K-means algorithm:** (coordinate descent on F)

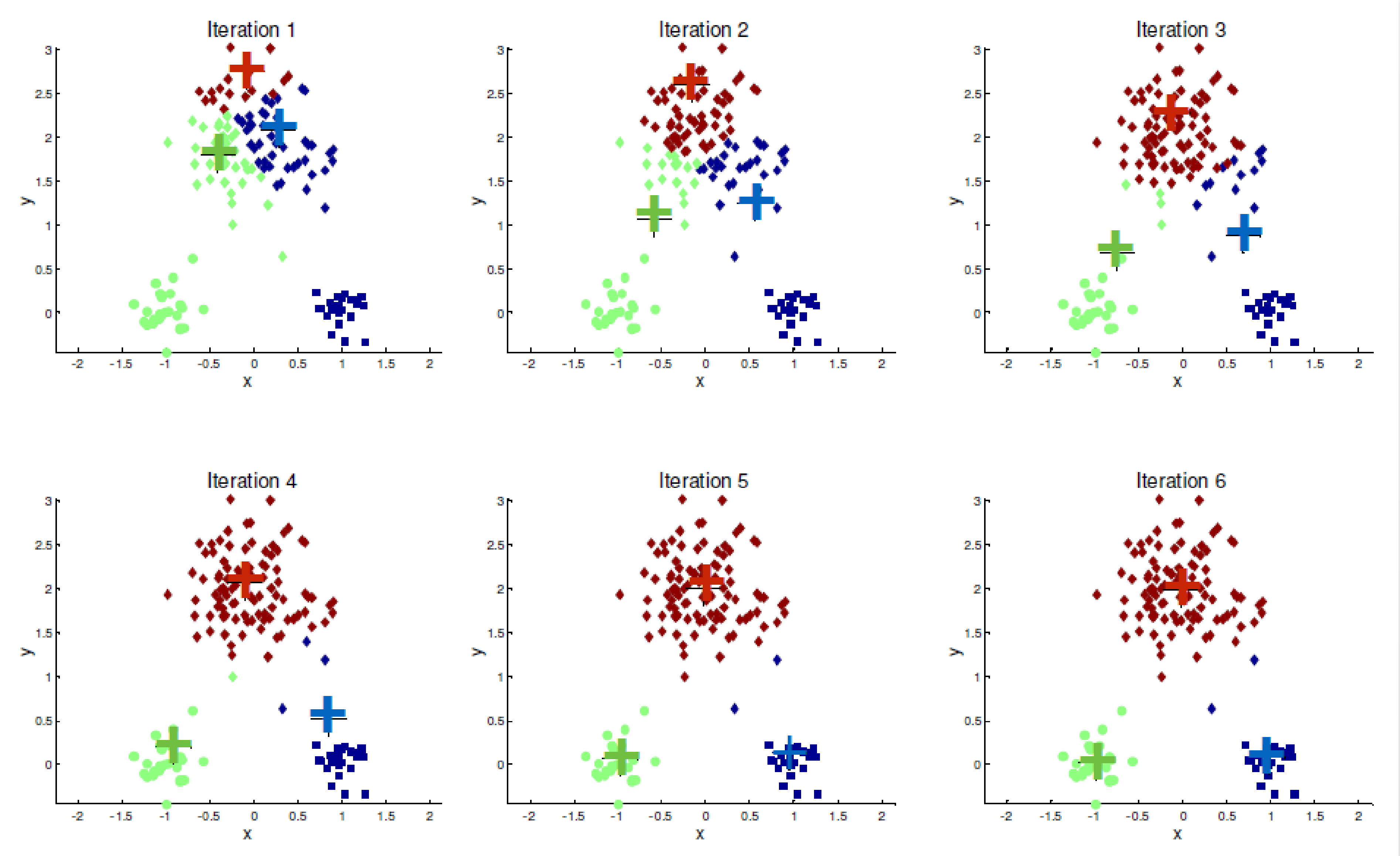
(1) Fix μ, optimize C — **Expected** cluster assignment

(2) Fix C, optimize μ — **Maximum** likelihood for center

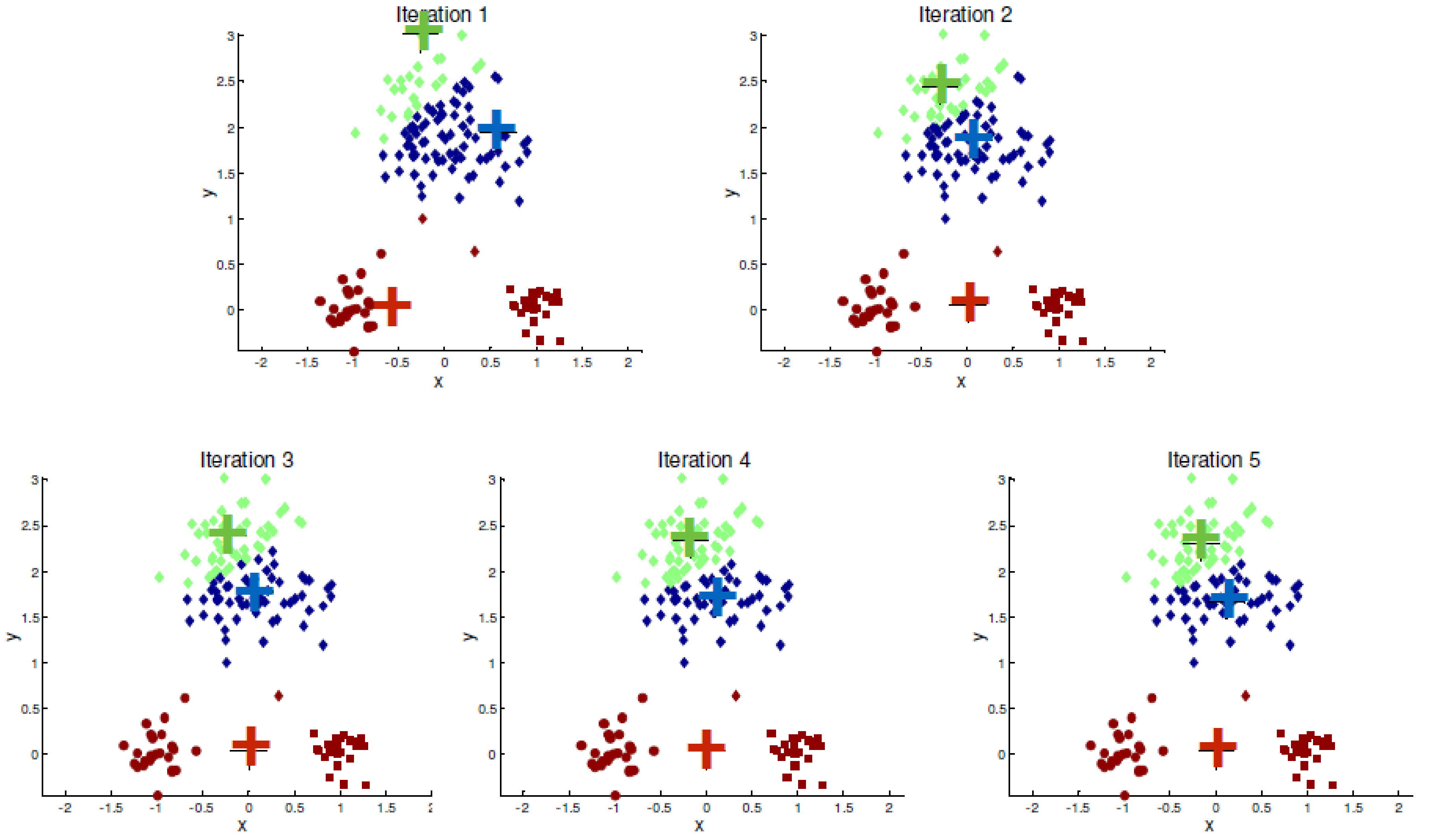
Soon we will see a generalization of this approach:

EM algorithm

“Good” Initialization of Centroids



“Bad” Initialization of Centroids



Importance of Initial Centroids

What is the chance of randomly selecting one point from each of *K* clusters?

(assume each cluster has size n = N/K)

$$\frac{\text{ways to select one from each cluster}}{\text{ways to select K centroids}} = \frac{K!n^K}{(Kn)^K} = \frac{K!}{K^K}$$

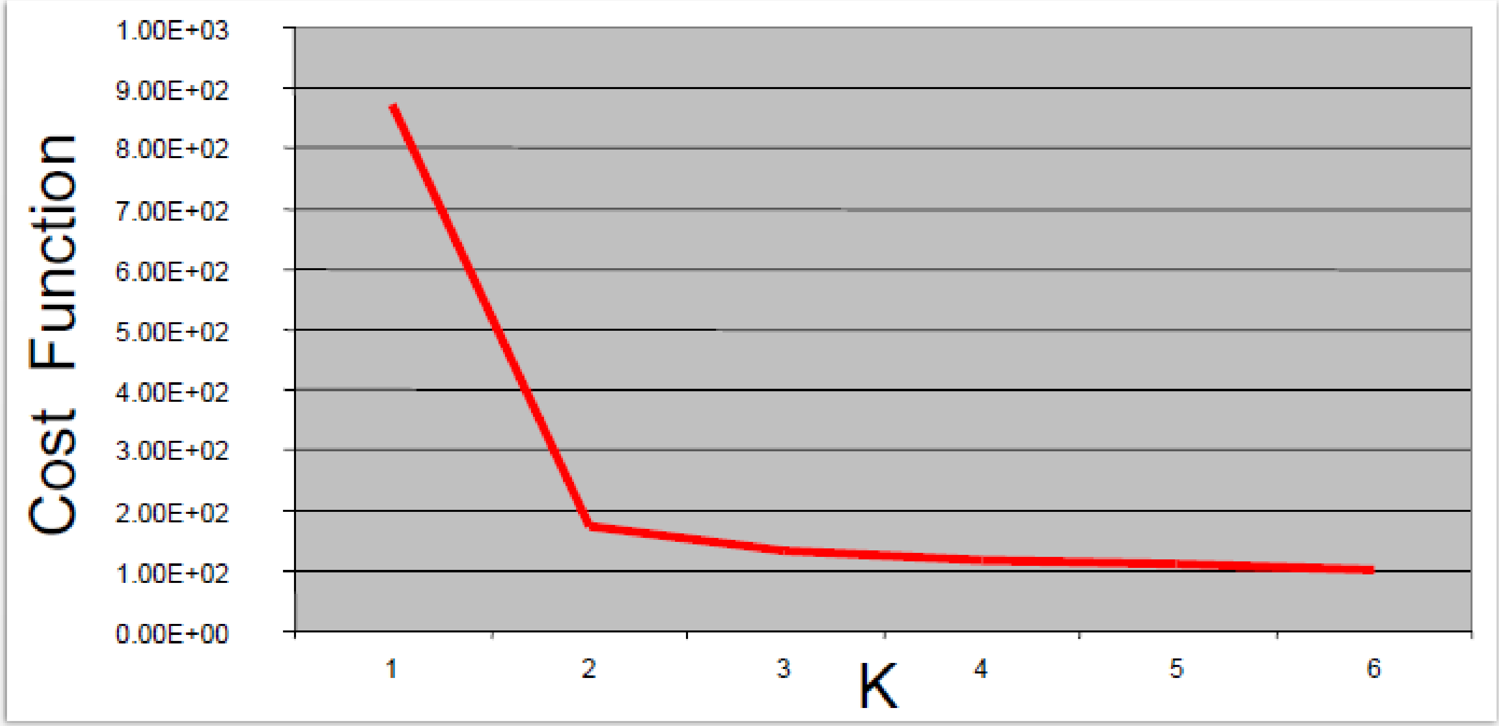
Implication: We will almost always have multiple initial centroids in same cluster.

Importance of Initial Centroids

Initialization tricks

- Use multiple restarts
- Initialize with hierarchical clustering
- Select more than K points,
  keep most widely separated points

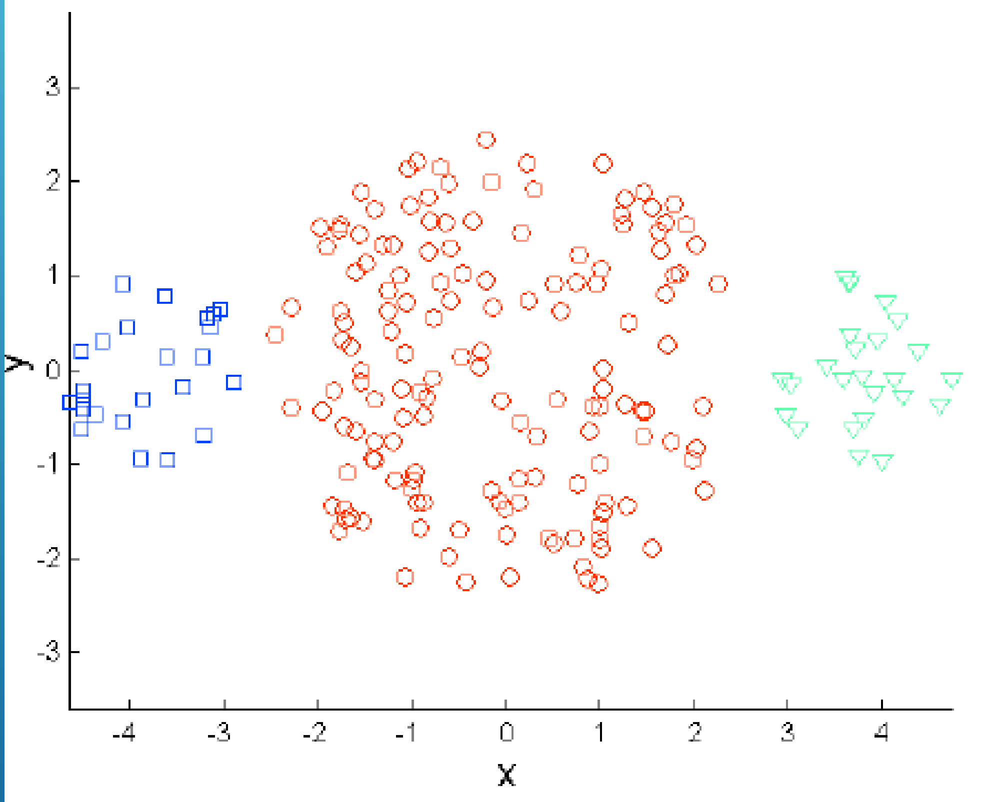
Choosing K



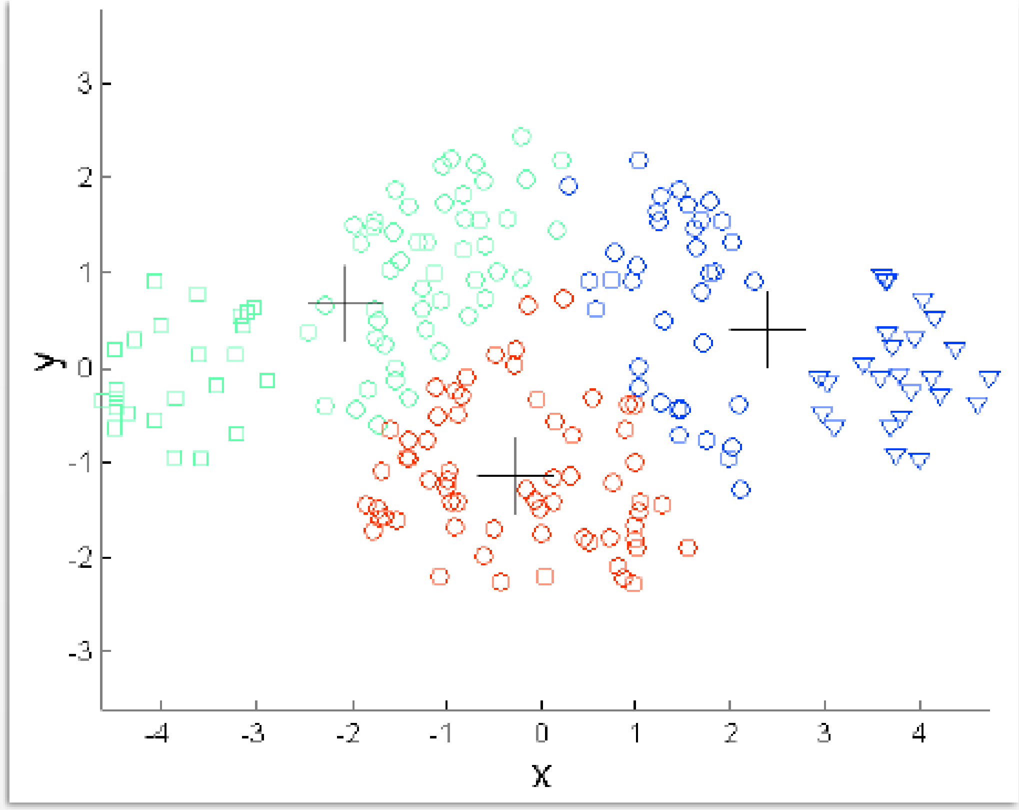
"Elbow finding" (a.k.a. "knee finding")

Set K to value just above "abrupt" increase

K-means Limitations: Differing Sizes

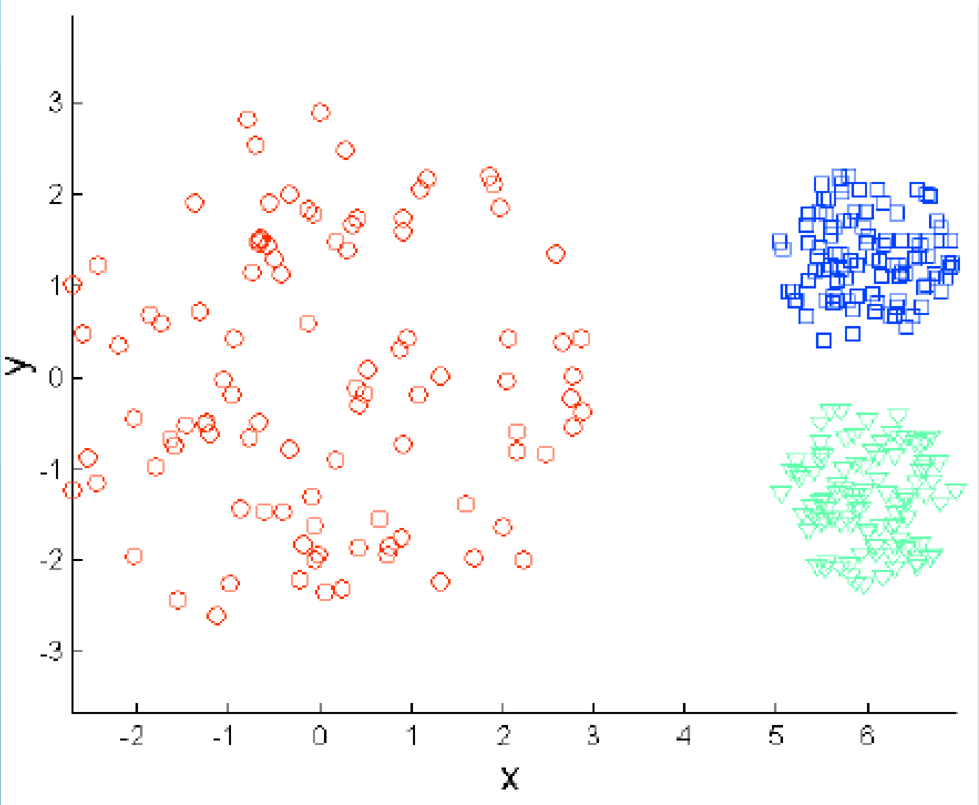


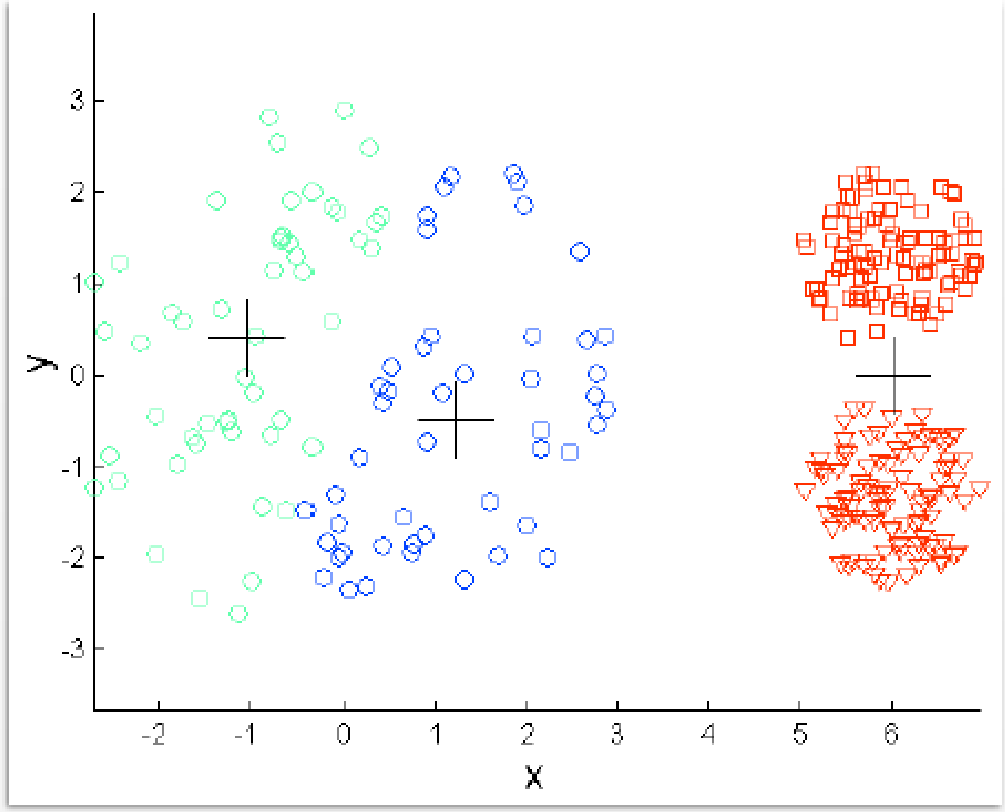
Original Points



K-means (3 clusters)

K-means Limitations: Different Densities

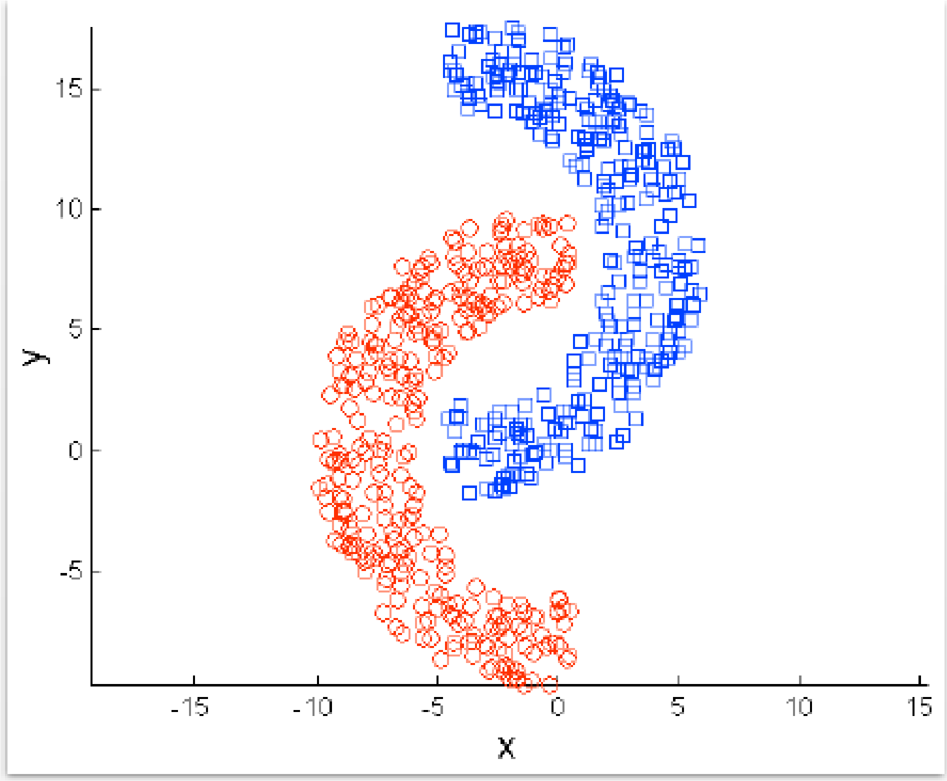




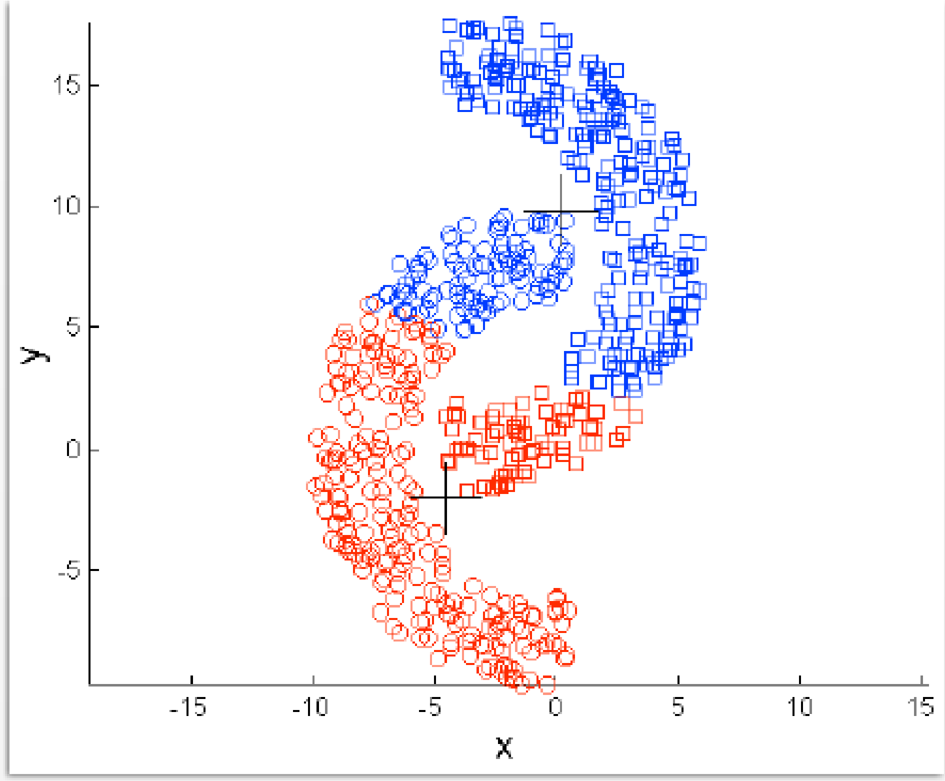
Original Points

K-means (3 clusters)

K-means Limitations: Non-globular Shapes

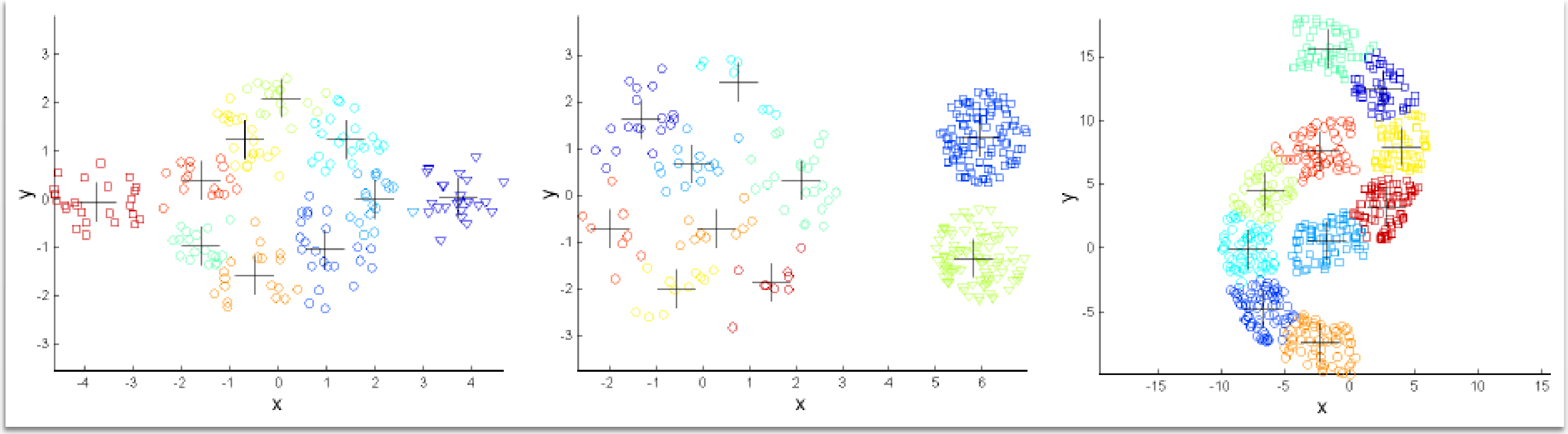


Original Points



K-means (2 clusters)

Overcoming K-means Limitations

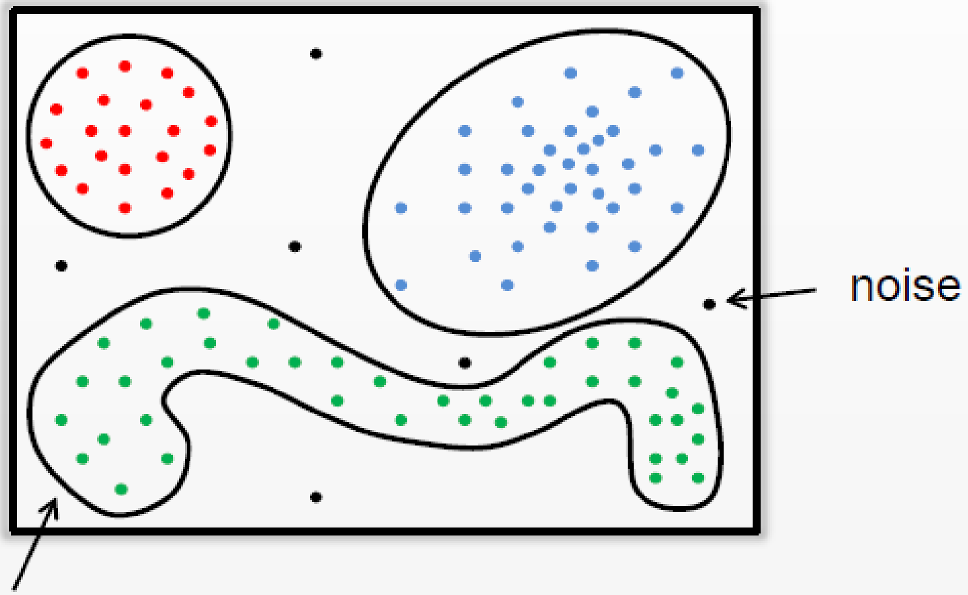


Intuition: “Combine” smaller clusters into larger clusters

- *One Solution:* Hierarchical Clustering
- *Another Solution:* Density-based Clustering

Density-based Clustering

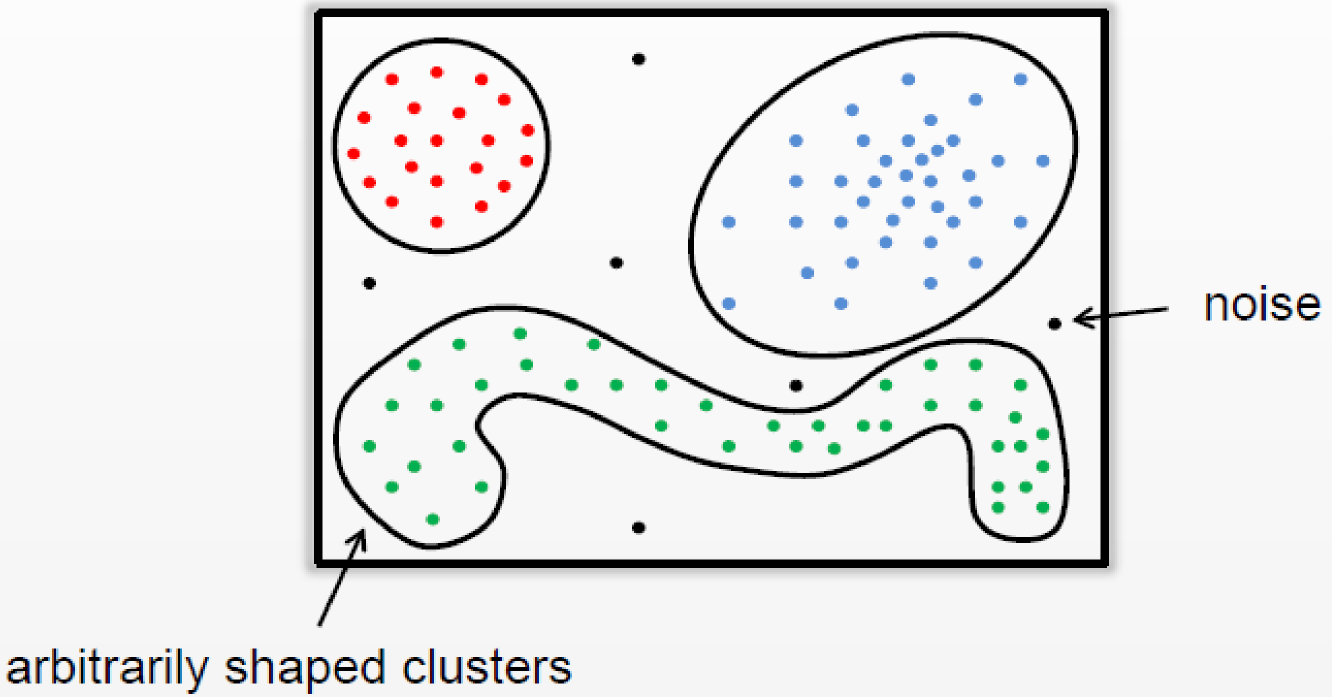
DBSCAN



[PDF] A density-based algorithm for discovering clusters in large spatial databases with noise.
M Ester, HP Kriegel, J Sander, X Xu - Kdd, 1996 - aaai.org
Abstract Clustering algorithms are attractive for the task of class identification in spatial databases. However, the application to large spatial databases rises the following requirements for clustering algorithms: minimal requirements of domain knowledge to ...
Cited by 8901 Related articles All 70 versions Cite Save More

(one of the most-cited clustering methods)

DBSCAN



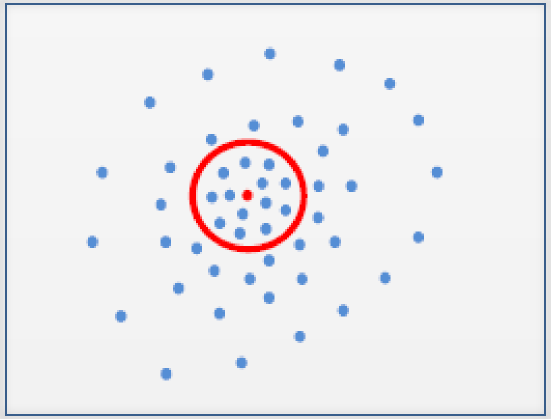
Intuition

- A *cluster* is a region of *high* density
- *Noise* points lie in regions of *low* density

Defining "High Density"

Naïve approach

For each point in a cluster there are at least a minimum number (MinPts) of points in an Eps-neighborhood of that point.

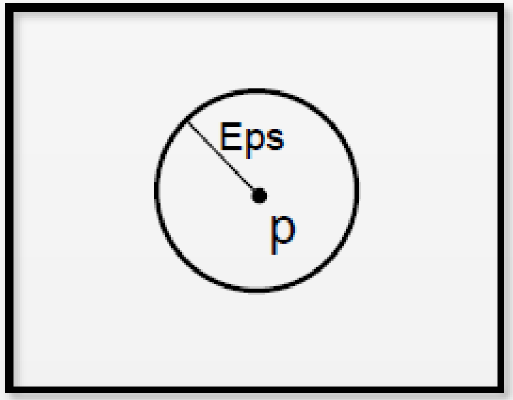


cluster

Defining "High Density"

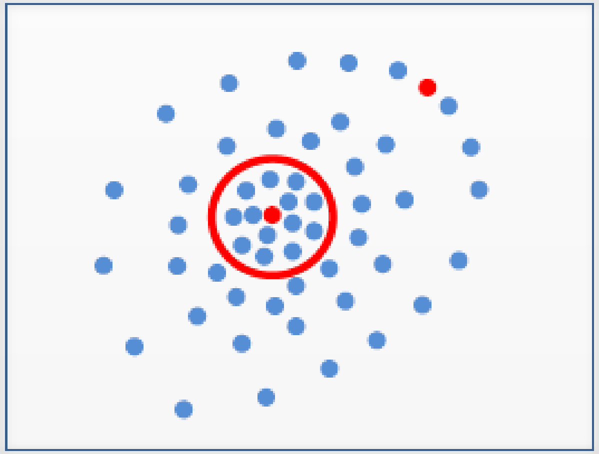
Eps-neighborhood of a point p

$$N_{Eps}(p) = \{ q \in D \mid dist(p, q) \leq Eps \}$$



Defining "High Density"

- In each cluster there are two kinds of points:
  - points inside the cluster (core points)
  - points on the border (border points)



cluster

An Eps-neighborhood of a border point contains significantly less points than an Eps-neighborhood of a core point.

Defining "High Density"

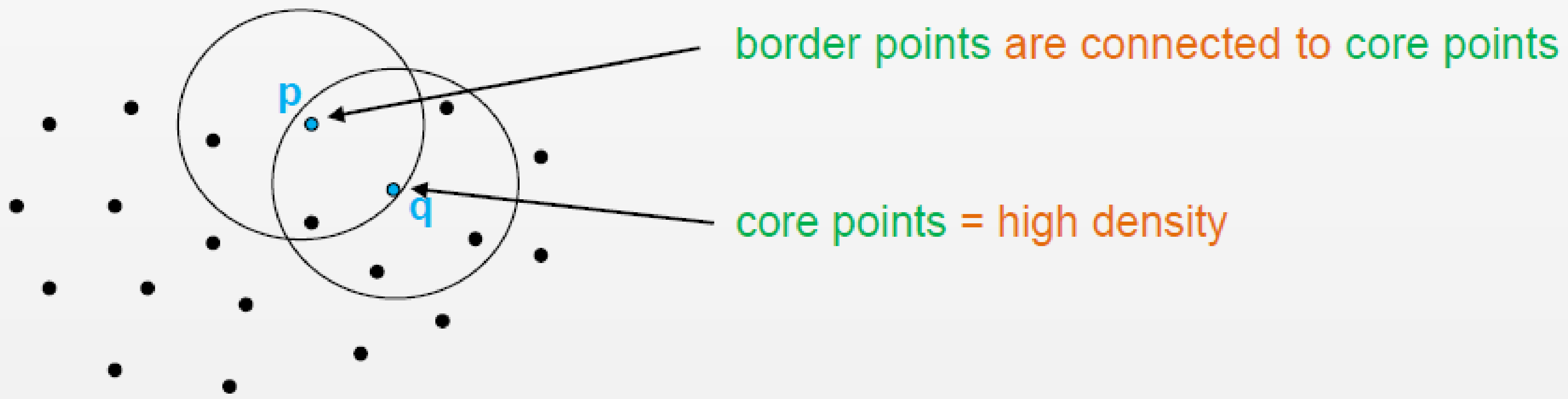
Better notion of cluster

For every point p in a cluster C there is a point $q \in C$, so that

(1) p is inside of the Eps-neighborhood of q

and

(2) $N_{Eps}(q)$ contains at least MinPts points.



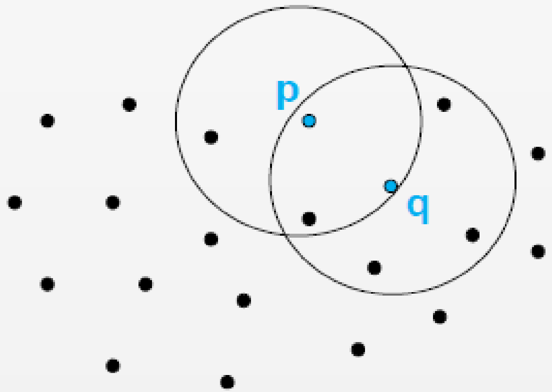
Density Reachability

Definition

A point p is directly density-reachable from a point q with regard to the parameters Eps and MinPts, if

1) $p \in N_{Eps}(q)$ (reachability)

2) $| N_{Eps}(q) | \geq MinPts$ (core point condition)



Parameter: MinPts = 5

p directly density reachable from q

$p \in N_{Eps}(q)$

$| N_{Eps}(q) | = 6 \geq 5 = MinPts$ (core point condition)

q **not** directly density reachable from p

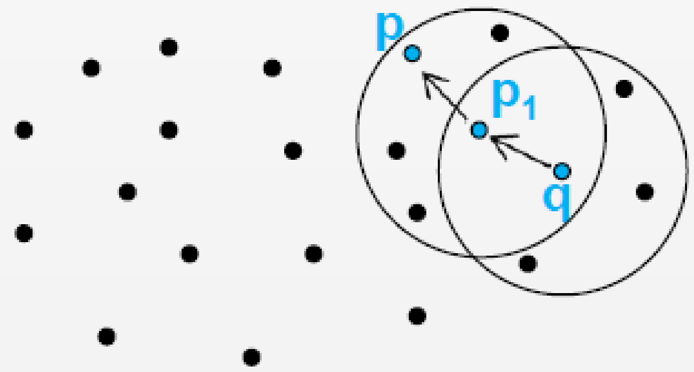
$| N_{Eps}(p) | = 4 < 5 = MinPts$ (core point condition)

Note: This is an asymmetric relationship

Density Reachability

Definition

A point p is density-reachable from a point q
with regard to the parameters Eps and MinPts
if there is a chain of points $p_1, p_2, \ldots, p_s$ with $p_1 = q$ and $p_s = p$
such that p_{i+1} is directly density-reachable from p_i for all $1 < i < s-1$.



MinPts = 5

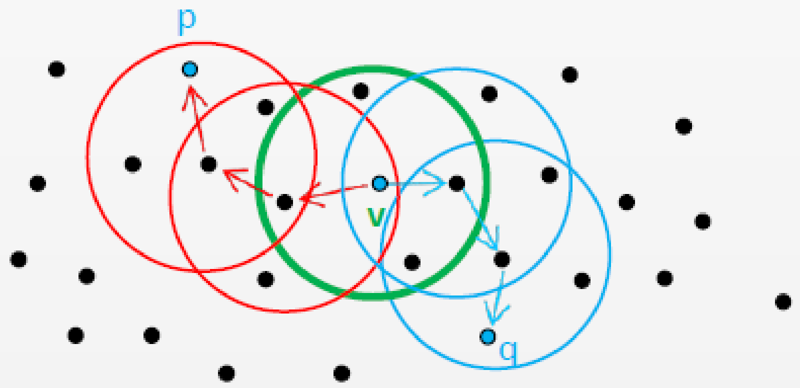
$| N_{Eps}(q) | = 5 = \text{MinPts}$ (core point condition)

$| N_{Eps}(p_1) | = 6 \geq 5 = \text{MinPts}$ (core point condition)

Density Connectivity

Definition (density-connected)

A point p is density-connected to a point q with regard to the parameters Eps and MinPts if there is a point v such that both p and q are density-reachable from v.



MinPts = 5

Note: This is a symmetric relationship

Definition of a Cluster

A cluster with regard to the parameters Eps and MinPts is a non-empty subset C of the database D with

1) For all p, q $\in$ D: (Maximality)
   If p $\in$ C and q is density-reachable from p with regard to the parameters Eps and MinPts, then q $\in$ C.

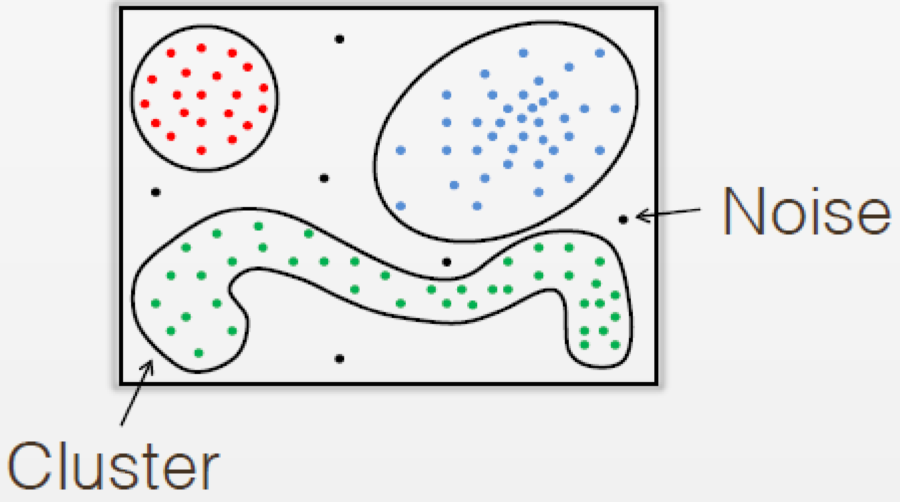
2) For all p, q $\in$ C: (Connectivity)
   The point p is density-connected to q with regard to the parameters Eps and MinPts.

Definition of Noise

Let $C_1,\ldots,C_k$ be the clusters of the database D with regard to the parameters Eps_i and $MinPts_i$ $(i=1,\ldots,k)$.

The set of points in the database D not belonging to any cluster $C_1,\ldots,C_k$ is called noise:

$$Noise = \{\, p \in D \mid p \notin C_i \text{ for all } i = 1,\ldots,k \}$$



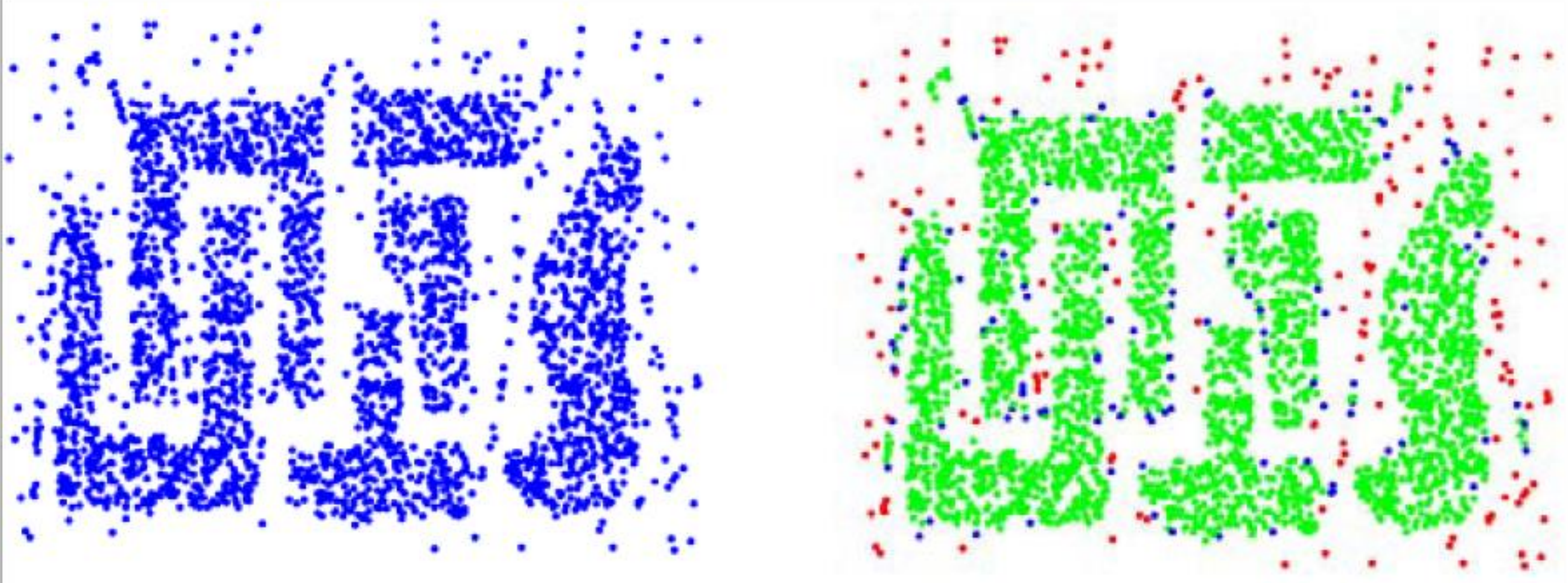
DBSCAN Algorithm

(1) Start with an arbitrary point p from the database and
retrieve all points density-reachable from p
with regard to Eps and MinPts.

(2) If p is a core point, the procedure yields a cluster
with regard to Eps and MinPts
and all points in the cluster are classified.

(3) If p is a border point, no points are density-reachable from p
and DBSCAN visits the next unclassified point in the database.

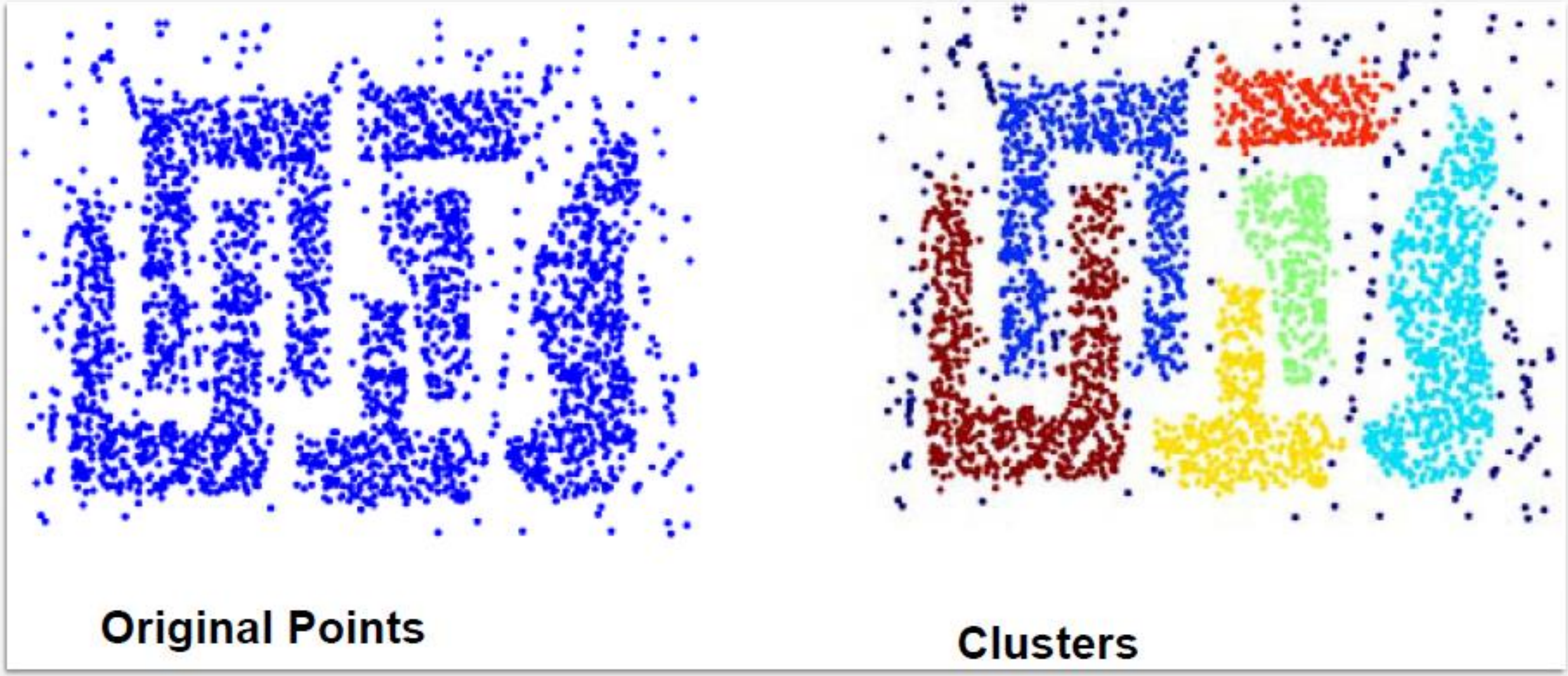
DBSCAN Algorithm



Original Points

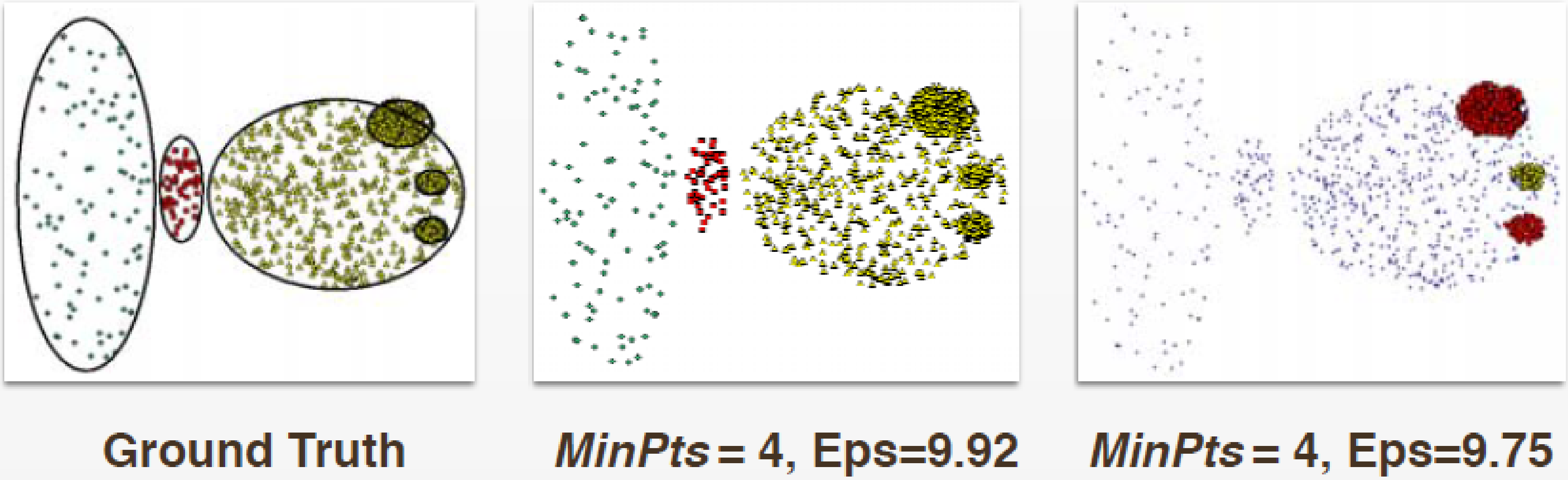
Point types: core, border and noise

DBSCAN strengths



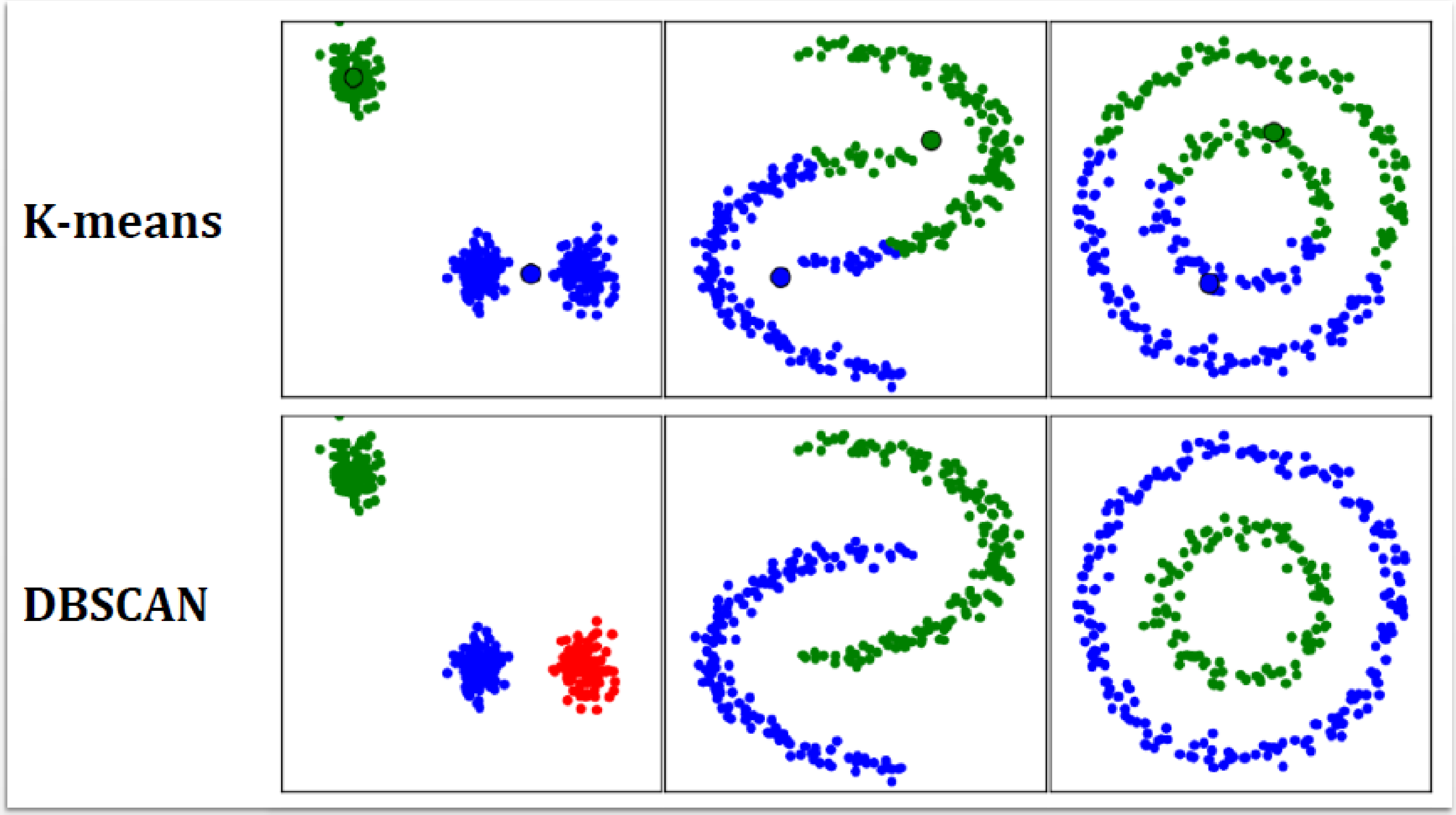
+ Resistant to noise
+ Can handle arbitrary shapes

DBSCAN Weaknesses



- Varying densities
- High dimensional data
- Overlapping clusters

K-means vs DBSCAN



VISUALIZING K-MEANS AND DBSCAN

- https://www.naftaliharris.com/blog/visualizing-k-means-clustering/
- https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/

FUTURE TOPICS: MORE CLUSTERING ALGORITHMS

1. **Centroid-based**
   - *K-means*
   - K-medoids
2. **Connectivity-based (Hierarchical)**
   - Hierarchical Agglomerative Clustering (HAC, bottom-up)
   - Hierarchical K-means (top-down)
   - Spectral Clustering (graph-based)
3. **Density-based**
   - *DBSCAN*
   - OPTICS
4. **Distribution-based (Mixture Models)**
   - Mixture of Gaussians
   - Expectation-Maximization (EM)

FUTURE UNSUPERVISED LEARNING TOPICS

1. **Density Estimation Topics**
   - Histograms
   - Kernel Density Estimation
2. **Dimensionality Reduction**
   - Principal Component Analysis (PCA)
   - t-SNE
   - UMAP
   - Autoencoders
3. **Outlier Detection Topics**
   - One-Class Classifier Learning
   - Autoencoders
4. **Other Clustering Topics**
   - HDBSCAN*
   - Cross-validation
5. **Deep Neural Network Approaches**

References:

The Hundred-Page Machine Learning Book. Andriy Burkov.

Applied Machine Learning in Python. Coursera. University of Michigan, Prof. Kevin Collins Thompson

Cluster analysis, https://en.wikipedia.org/w/index.php?title=Cluster_analysis&oldid=1002271612 (last visited Jan. 27, 2021).

Dimensionality reduction, https://en.wikipedia.org/w/index.php?title=Dimensionality_reduction&oldid=1002754996 (last visited Jan. 27, 2021).

INTRODUCING CLUSTERING ALGORITHMS

Scott O'Hara
Metrowest Developers Machine Learning Group
03/03/2021

NEW TALK

Reference: Machine Learning: Clustering & Retrieval
University of Washington, Profs. Emily Fox & Carlos Guestrin

Reference: The Hundred-Page Machine Learning Book. Andriy Burkov.

Reference: *Applied Machine Learning in Python*. Coursera.
University of Michigan, Prof. Kevin Collins Thompson