# Clustering*

*** with a focus on Density-based Clustering**

A brief overview

Matt Piekenbrock

Graduate CS Student | B.S. in Computer Science, Statistics minor

# Why should anyone care about clustering

- Clustering has been successfully across many disciplines, including:
  - Image segmentation / recognition [16]
  - Character Recognition [10]
  - Spatial Data mining [20]
  - Epidemiology [15, 35]
  - Healthcare [18, 28]
- In *many* contexts
  - Finding ‘Optimal’ protein structure alignments [5]
  - Improving the performance of neural networks [9]
  - Modeling earthquakes [12]
  - Learning from MRI images (fuzzy clustering) [18]
  - Strategic Group Analysis [21]
- Perhaps a better question: what has clustering *not* been used for?

# The Definition Slide: Clustering

- A (generally) unsupervised machine learning task that attempts to group objects based on some notion of "[dis]similarity"

- Alternative Def. 1: "The aim is to assign instances to classes that are not defined a priori and that are (usually) supposed to somehow reflect the "underlying structure" of the entities that the data represents." [19]

- Alternative Def. 2: "Cluster analysis encompasses many diverse techniques for discovering structure in complex bodies of data"[43]

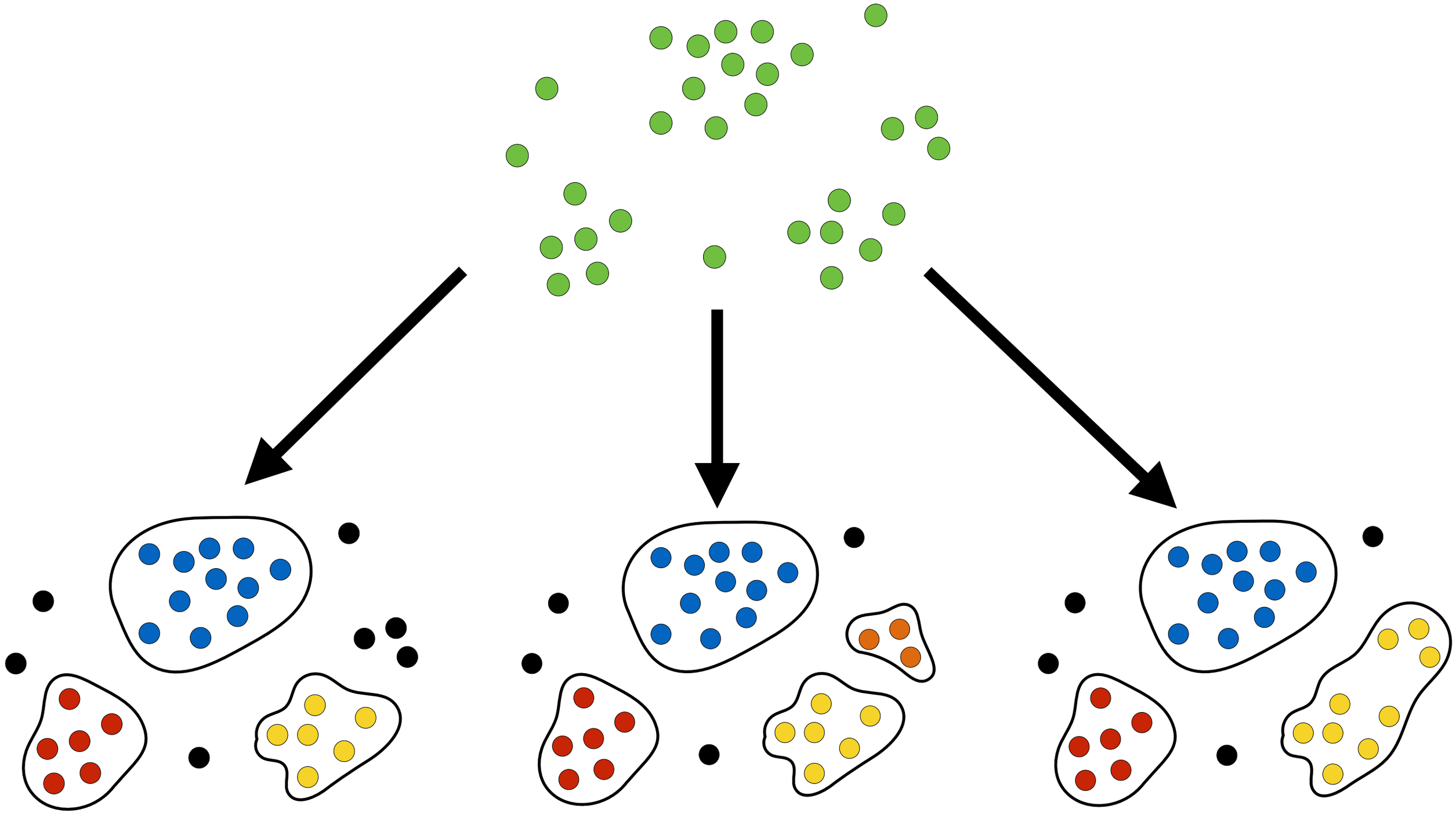
# What is a Cluster?

- Knuth on Computer Programming: “It is clearly an art, but many feel that a science is possible and desirable”… Many feel the same applies to clustering [1]
- Motivated publications like: “Clustering: Art or Science?”[19]
  - Making clustering *more scientific* is ‘certainly desirable’, but it’s not obvious how to do so [Raftery et. all]
    - Can there be a domain-independent, ‘optimal’ clustering solution? Most recent comparative studies argue **no** [3],
- “Supervised clustering can be easily made into a well defined problem with loss functions which precisely formalize what one is trying to do… ” [19]
- “The difficulty with unsupervised clustering is that there are a *huge* number of possibilities regarding what will be done with it and (as yet) no abstraction akin to a loss function which distills the end-user intent.”[19]

[1] R. B. Zadeh. Towards a principled theory of clustering. In Thirteenth International Conference on Artificial Intelligence and Statistics, 2010
[2] O. Arb elaitz, I. Gurrutxaga, J. Muguerza, J. M. PeRez, and I. Perona. An extensive comparative study of cluster validity indices. Pattern Recognition, 46(1):243{256, 2013

# Identifying Clusters:

*How many "clusters" are there?*



# Types of Clustering

- Generally can be broken into the following categories:
  - Partitioning Methods
  - Subspace
  - Graph-based
  - Density-based
  - Model-based (distribution models)
  - Hierarchical
- … even the output often varies
  - Flat or ‘crisp’ clustering - hard assignments given to each object
  - Soft (fuzzy) clustering - objects may be given multiple values
  - Hierarchical - points are assigned labels based on a hierarchy
- “Most clustering done in practice is based largely on heuristic but intuitively reasonable procedures” [Raftery et. all]

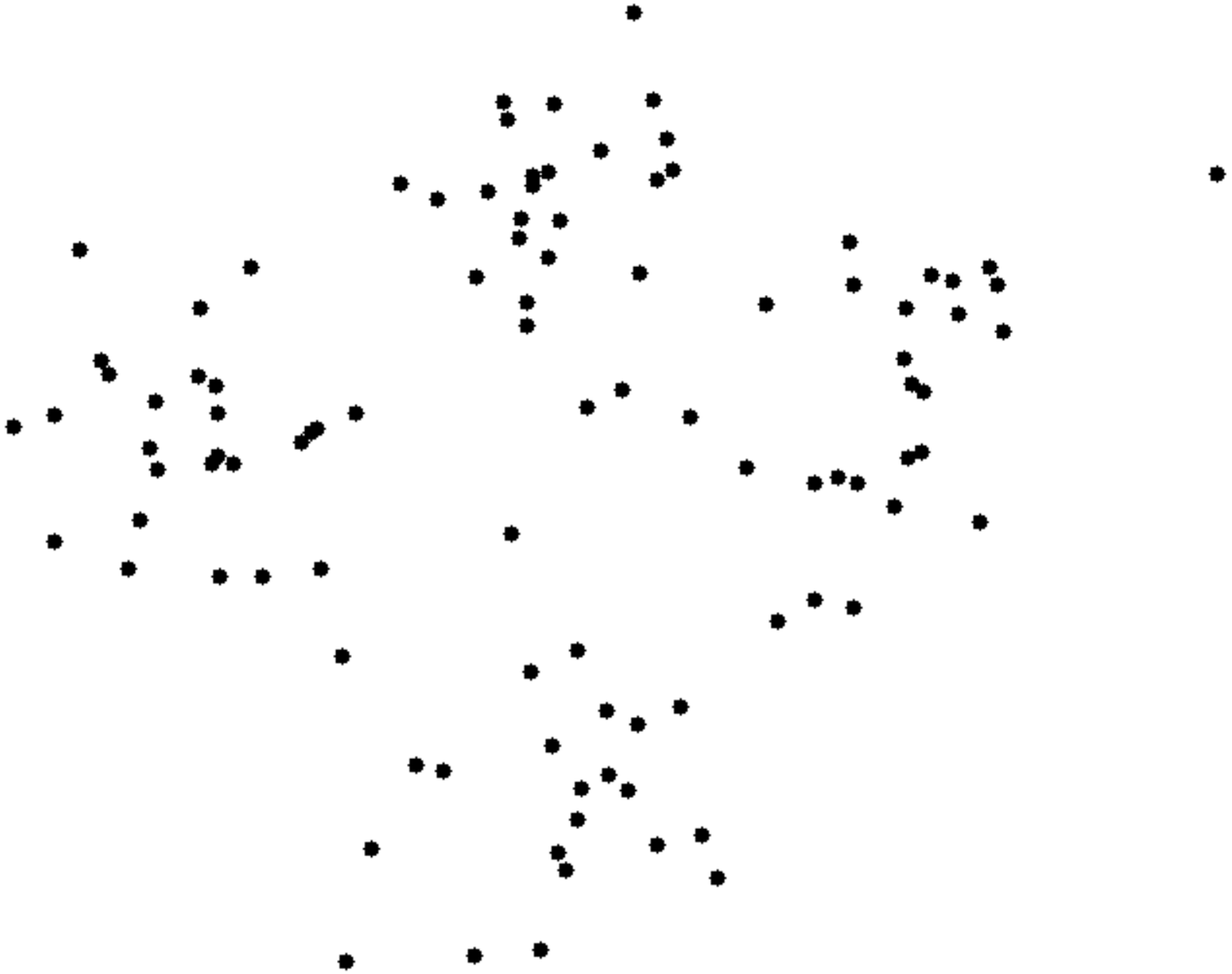
# K-Means Clustering

- Given a set of observations $X = (x_1, x_2, \ldots, x_n)$ , partition the $n$ observations into $k$ partitions $S = \{S_1, S_2, \ldots, S_k\}$ , such that

$$\underset{S}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2$$

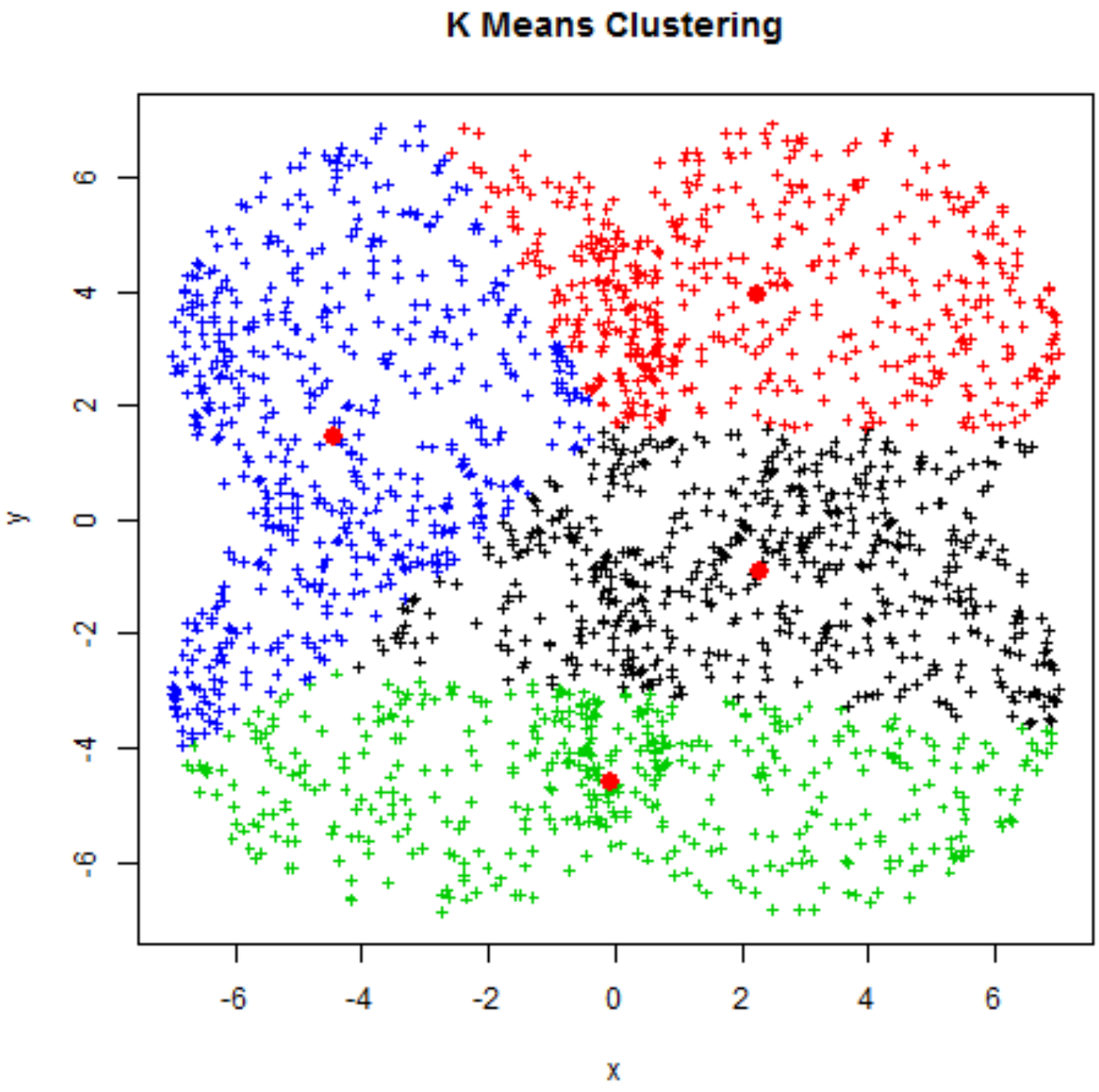
- The algorithm now known today as K-Means has been formulated by many authors; used to be called [26]:
  - Dynamic Clusters methods
  - Iterated minimum-distance partition method
  - Nearest centroid sorting
  - Iterative Relocation [Raftery et. all]
  - Vector Quantization
- Wasn't referred to as "k-means" until 1967 [McQueen 1967]
- Original foundational theory was based on the idea that K-means asymptotically minimizes the sum-of-squares criterion (SSQ)

# K Means: Animation



Animation from: http://shabal.in/visuals/kmeans/right.gif

# K-Means: Another Animation



# From the top: Statistical Learning

- Statistical learning theory deals with the problem of finding a predictive function based on some finite data source
- Statistical inference may divided into two major areas [1]:
    1. Estimation
    2. Tests of hypothesis

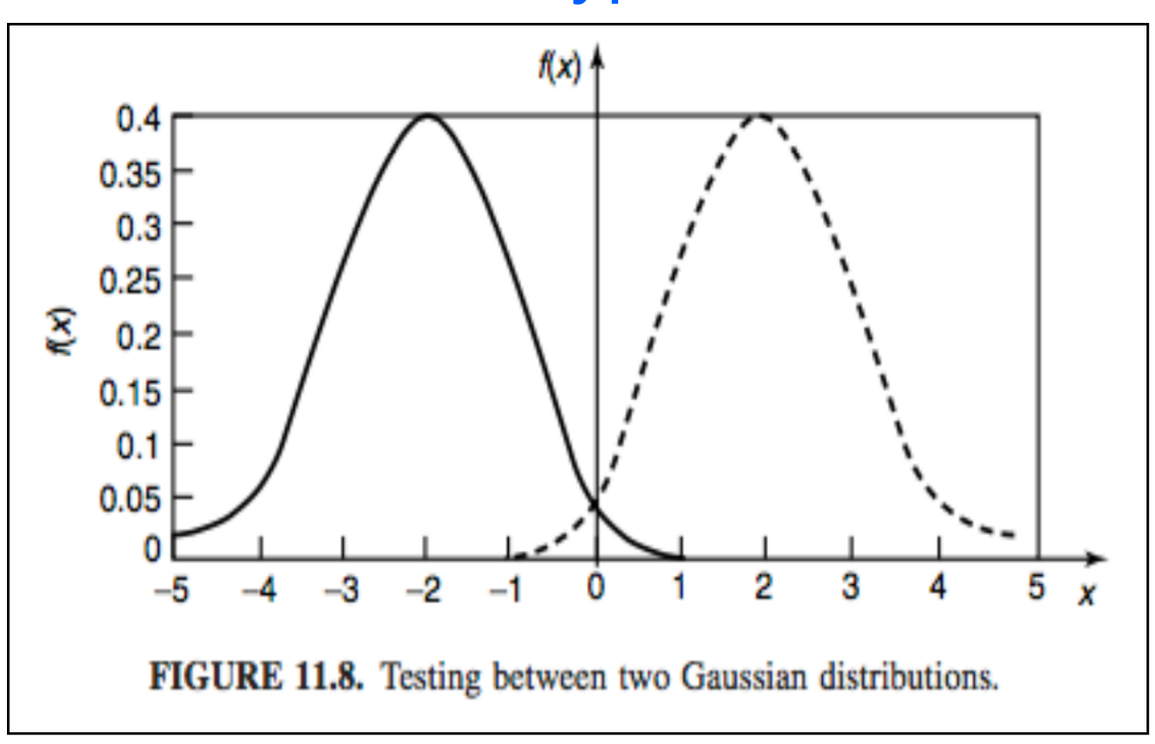
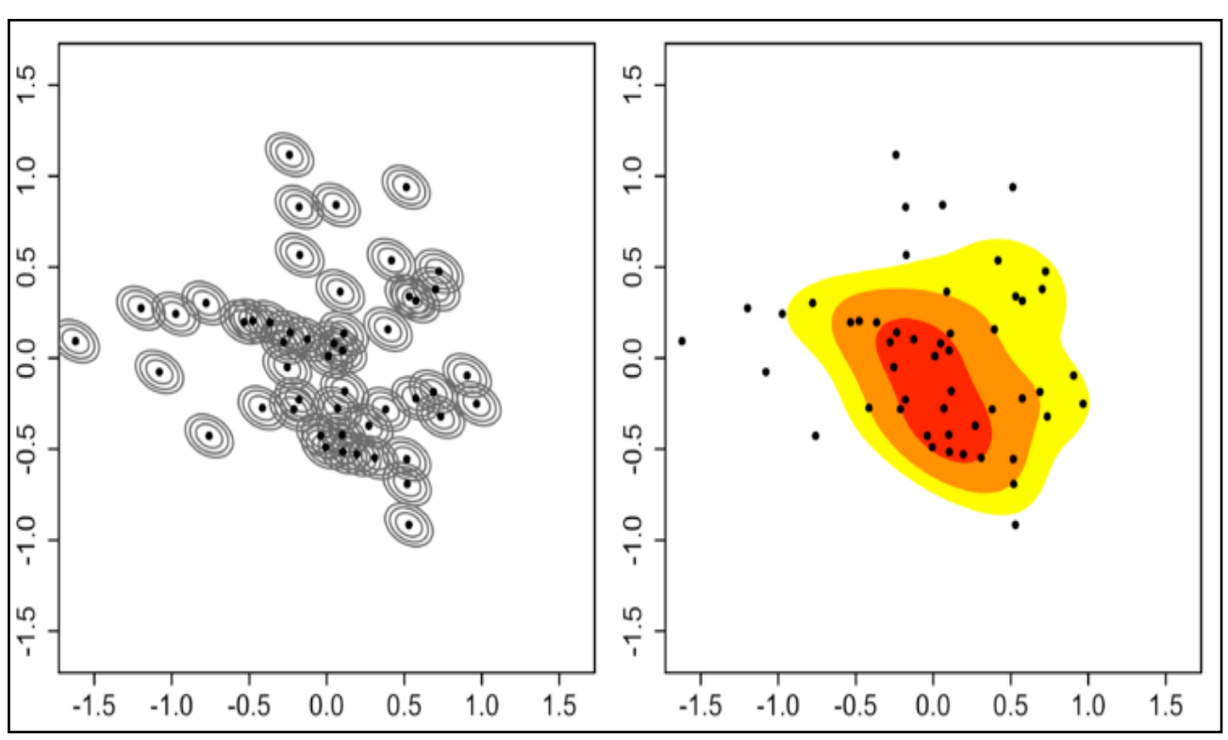



**FIGURE 11.8.** Testing between two Gaussian distributions.





1. Image from Garber, Fred. “Information Theory.” Wright State University, Ohio. 2016. Lecture.
2. Image from wikipedia: https://en.wikipedia.org/wiki/Multivariate_kernel_density_estimation
3. Walpole, Ronald E., et al. Probability and statistics for engineers and scientists. Vol. 5. New York: Macmillan, 1993.

# Parameter Estimation

- Research involving the improvement of point estimators seems foundationally useful

- From [1], a statistic $\hat{\Theta}$ is said to be an **unbiased estimator** of the population parameter $\theta$ if:

$$\mu_{\Theta} = E(\hat{\Theta}) = \theta$$

- Similarly, if $\hat{\Theta}_1$ and $\hat{\Theta}_2$ are both estimators of the parameter $\theta$, and $\sigma^2_{\theta_1} < \sigma^2_{\theta_2}$, then $\hat{\Theta}_1$ is said to be a **more efficient estimator** of $\theta$

1. Walpole, Ronald E., et al. Probability and statistics for engineers and scientists. Vol. 5. New York: Macmillan, 1993.

# K Means: The original formulation (or one of)

Consider a the problem of estimating $\mu = E[X]$ from a real-valued random variable $X$ with a distribution density $f(x)$, where $X$ contains $k$ contiguous intervals

$$B_i = (\mu_{i-1}, \mu_i](i = 1, \ldots, k+1)$$

such that, when $n_i$ observations are sampled from each interval $B_i$, is proportional to the probability mass of $n_i = n \cdot P(B_i)$. Each $x$ in $B_i$ 'builds' a class $C_i$ with class average $\bar{x}_{C_i}(i = 1, \ldots, k)$ such that the linear combination

$$\hat{\mu} := \sum_{i=1}^{k} \frac{n_i}{n} \bar{x}_{C_i}$$

provides an **unbiased estimator** of $\mu$ with variance given by the SSQ criterion,

$$g(\mathcal{B}) := \sum_{i=1}^{k} \int_{B_i} ||x - E[X|X \in B_i]||^2 \, dP(x) \rightarrow \min_{\mathcal{B}}.$$

# Classification

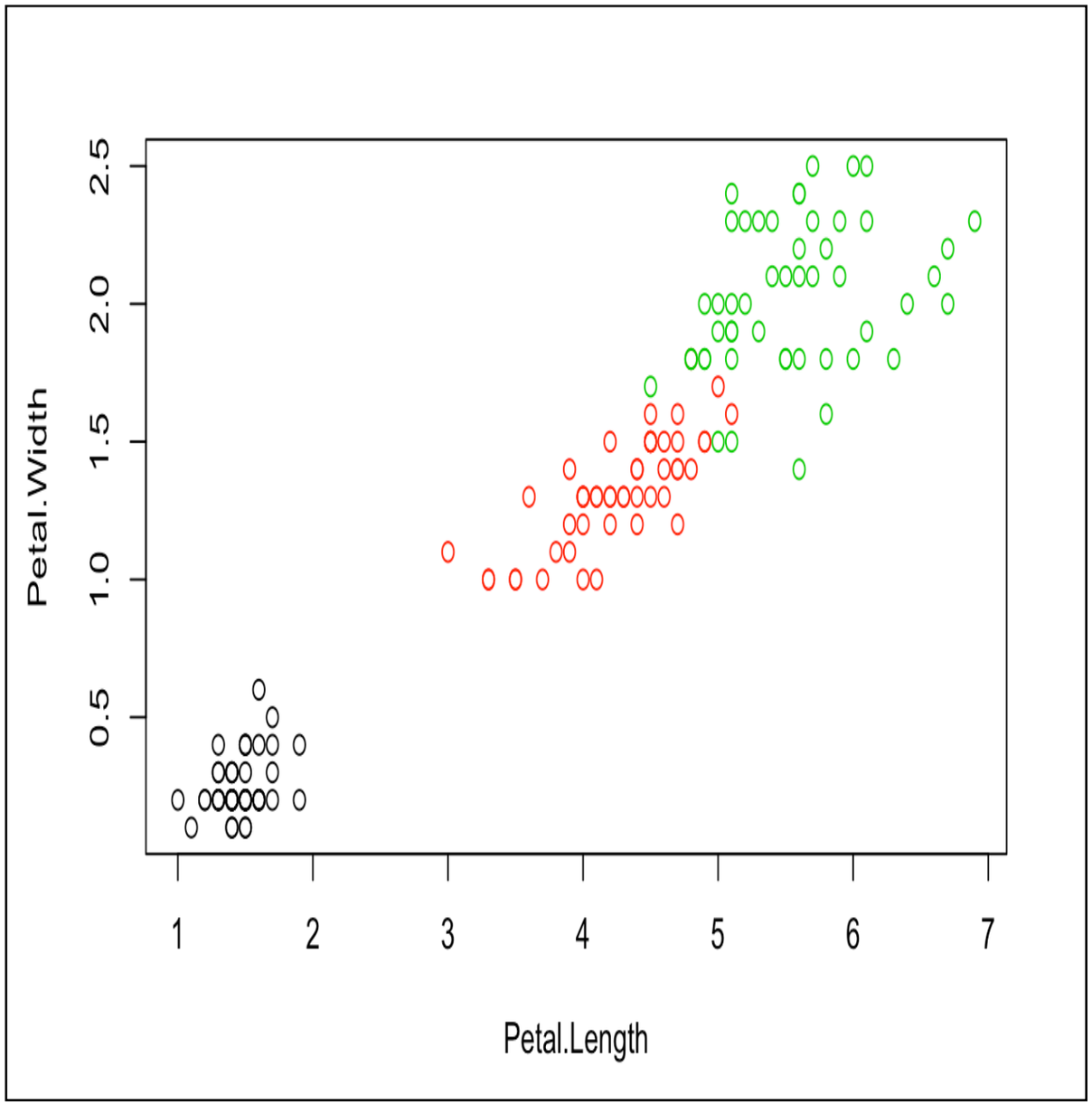
*We all know about [supervised] classification*

Given a set of $N$ training observations of the form

$$\{(x_1, y_1), \ldots, (x_N, y_N)\}$$

such that $x_i$ is a $d$-dimensional vector of input features corresponding to grouped observations about $y_i$ , and $x_i$ is its known label/class, find the *best* learning algorithm:

$$g : X \to Y$$



# Classification: Discriminative Models

Can the classes be linearly separated by optimizing some coefficients $\beta$ , s.t.

$$y_i = \begin{cases} 1 & x_i^T \beta + \epsilon_i > 0 \\ 0 & \text{else} \end{cases}$$

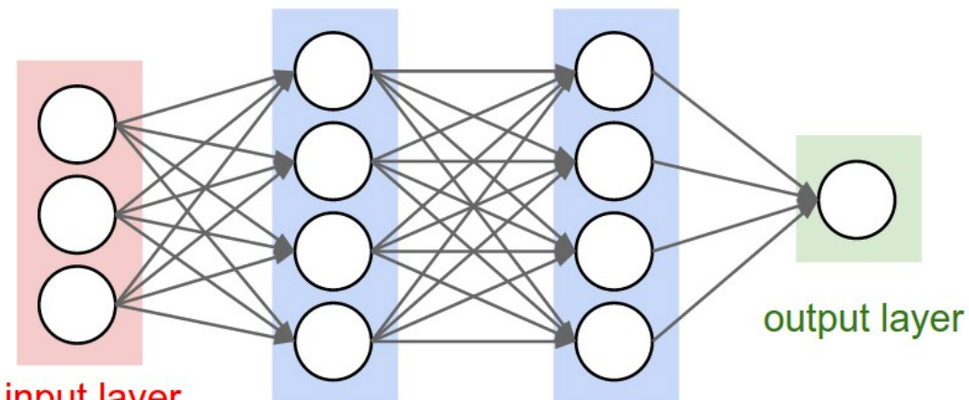
Where the imposed decision function often interpreted to ideally model the conditional probability $P(Y \mid X)$ modeled through the logistic function

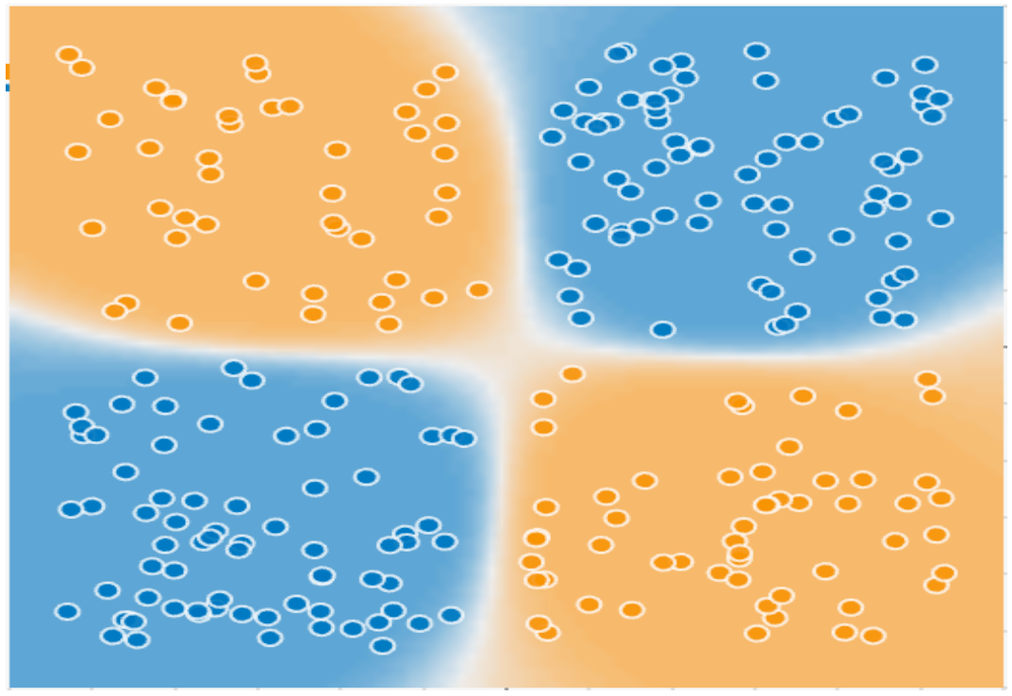
$$f(x) = \frac{1}{1 + e^{-x}}$$

… which the evaluation of is often referred to as the ‘degree of membership’, or the ‘probability of membership

$$g : X \to Y$$

*Neural Networks*



Plots from: http://cs231n.github.io/ and http://playground.tensorflow.org/

# Consistency of estimators

- These models generally require training in the form of some loss/cost/objective function

$$L(\theta, \hat{\theta})$$

representing some function that quantifies the difference between estimated and true values for an instance of data, or more expressively:

$$L(g(X;\theta), g(X;\hat{\theta}))$$

- Ideally, shouldn't a statistical learning model attempt to be ***consistent*** and thus ***asymptotically unbiased****?*

- What if biased estimation improves the power of the model?

# Statistical Estimation

- Perhaps one the most central preoccupations of statistic learning theory is to understand **what statistical estimation based on a finite data set reveals about the underlying distribution from which the data were sampled** [7]

- In machine learning, point estimators are responsible for creating these decision boundaries

- K-means algorithms has been proven to have strong consistency guarantees w.r.t the SSQ [Pollard et. all]

- The question in clustering remains:

  - "Is the best $k$-means solution to $f$ and interesting and desirable quantity, in settings outside of vector quantization?"[8]

1. Pollard, David. "Strong consistency of $ k $-means clustering." The Annals of Statistics 9.1 (1981): 135-140.

# Model-based Clustering

- "Model-based clustering assumes the data were generated from a model, and tries to **recover the original model from the data**"[1]

- In general, model-based clustering uses the **information assumed regarding the structure of the data** to measure the "goodness" of the clustering, i.e. (for ex. the likelihood)

$$\arg\max_{\Theta} L(D \mid \Theta)$$

- Where, generally speaking, the goal is to find the optimal parameter settings $\Theta$ which maximizes the (log)-likelihood $L$ of generating the data $D$

1. http://nlp.stanford.edu/IR-book/html/htmledition/model-based-clustering-1.html

# Model-based Clustering: Example

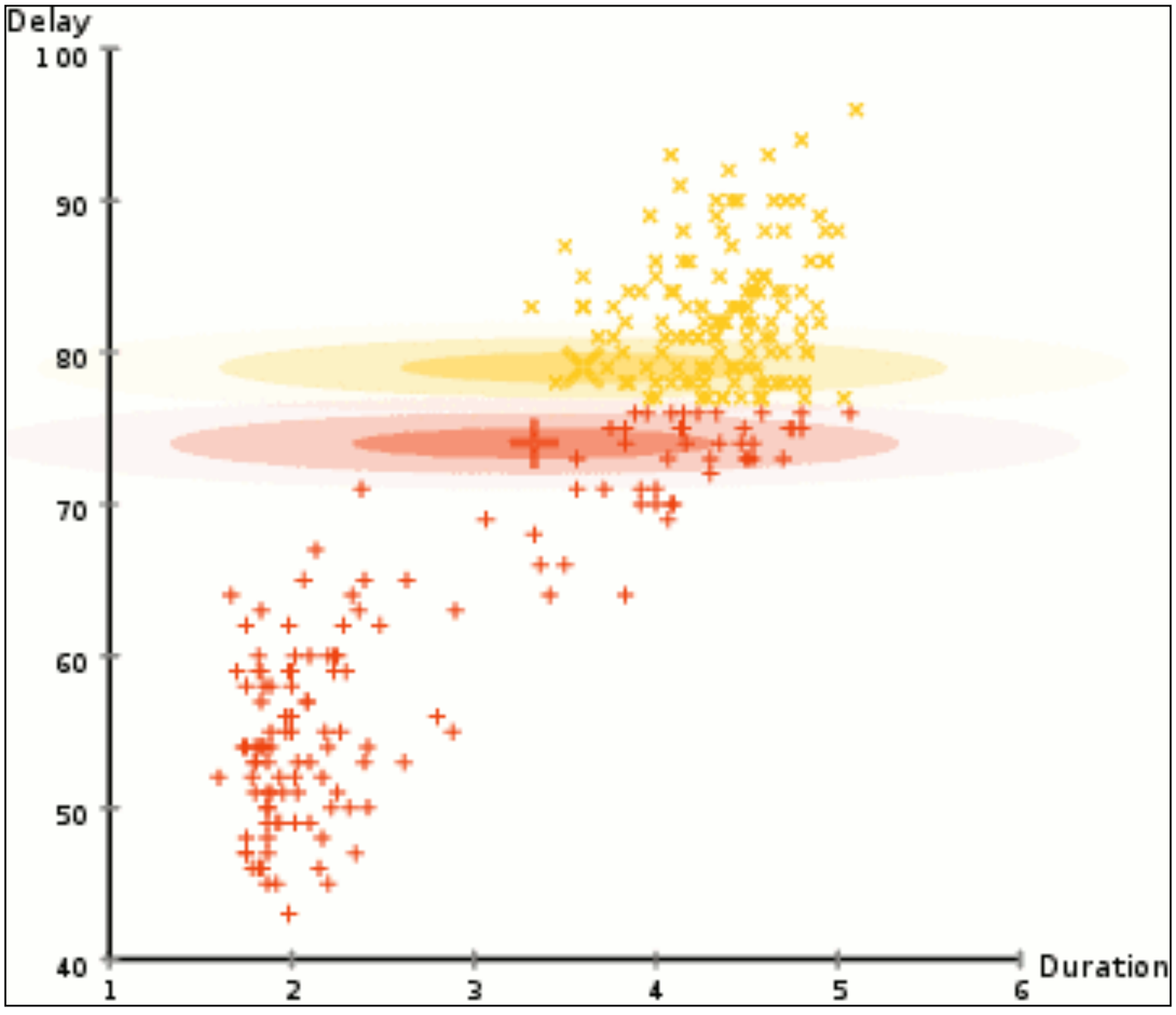
- For example, in the K-Means algorithm,

$$\Theta = \{\mu_1, \mu_2, \ldots, \mu_k\}$$

- Another common example includes Gaussian Mixture Modeling (GMM)

- One of the most often chosen optimization functions is the **Expectation-Maximization** (EM) algorithm which iteratively updates the parameters $\Theta$ based on maximum likelihood estimate

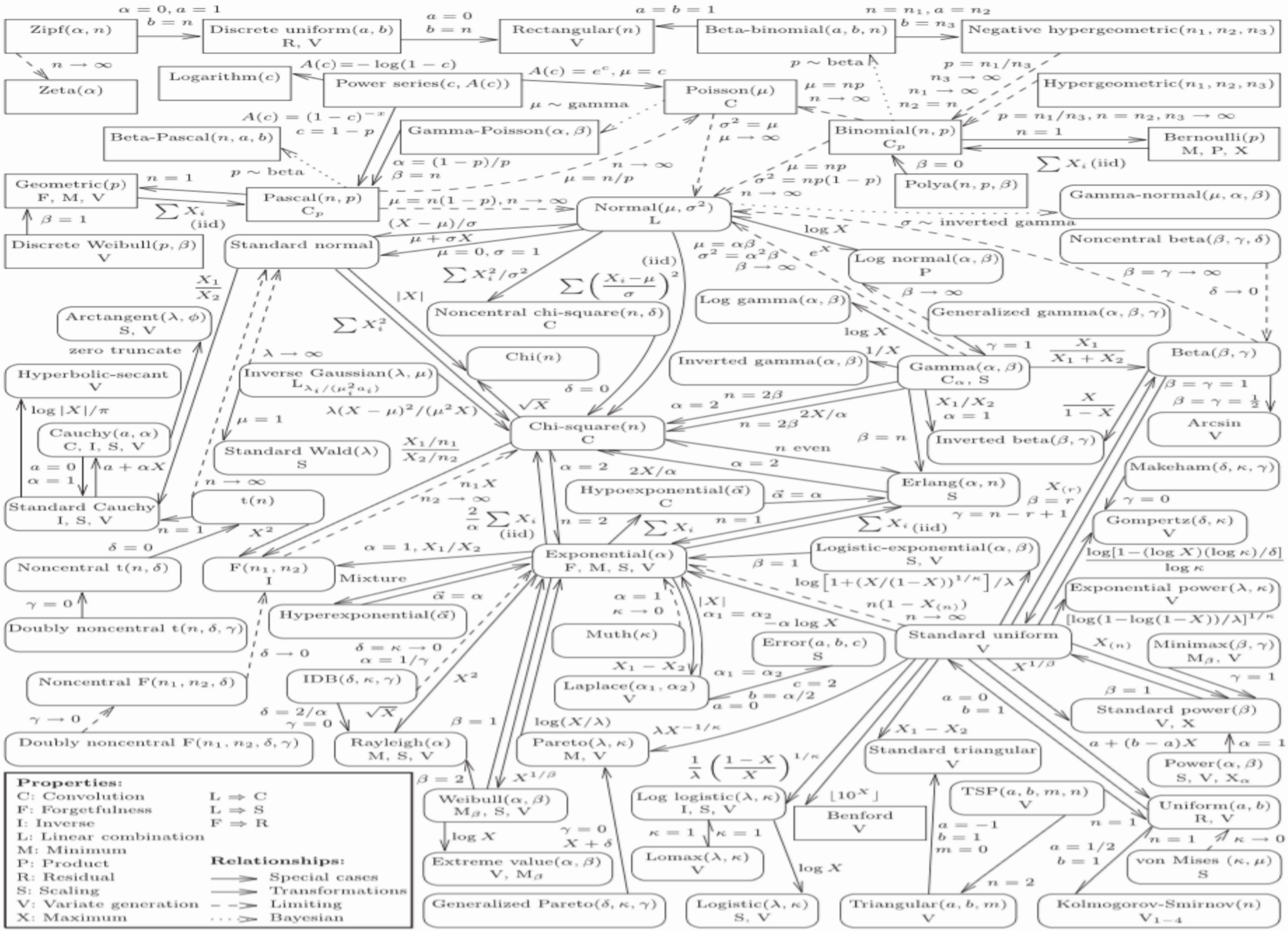
# Gaussian Mixture Modeling



# Nonparametric Model-based Clustering

- Consider the following game played with a predefined, finite class of distributions $\Theta = \{\theta_1, \ldots, \theta_l\}$ defined on some common space $\chi$
  - Nature picks $I \in \{1, \ldots, l\}$
  - You're given $n$ i.i.d samples $X_1, \ldots, X_n$ from $\theta_I$
  - Guess the identity of $I$ for each $X_i$ , and estimate $\theta_I$

Example from: K. Chaudhuri and S. Dasgupta. Rates of convergence for the cluster tree. In Advances in Neural Information Processing Systems, pages 343{351, 2010.}




**Univariate distribution relationships, courtesy Leemis and McQueston [2].**

# The Definition Slide: *Density-based* Clustering

- "Density-based clusters are separated from each other by contiguous regions of low density of objects. Data objects located in low-density regions are typically considered noise or outliers." [Sanders et. all '14]

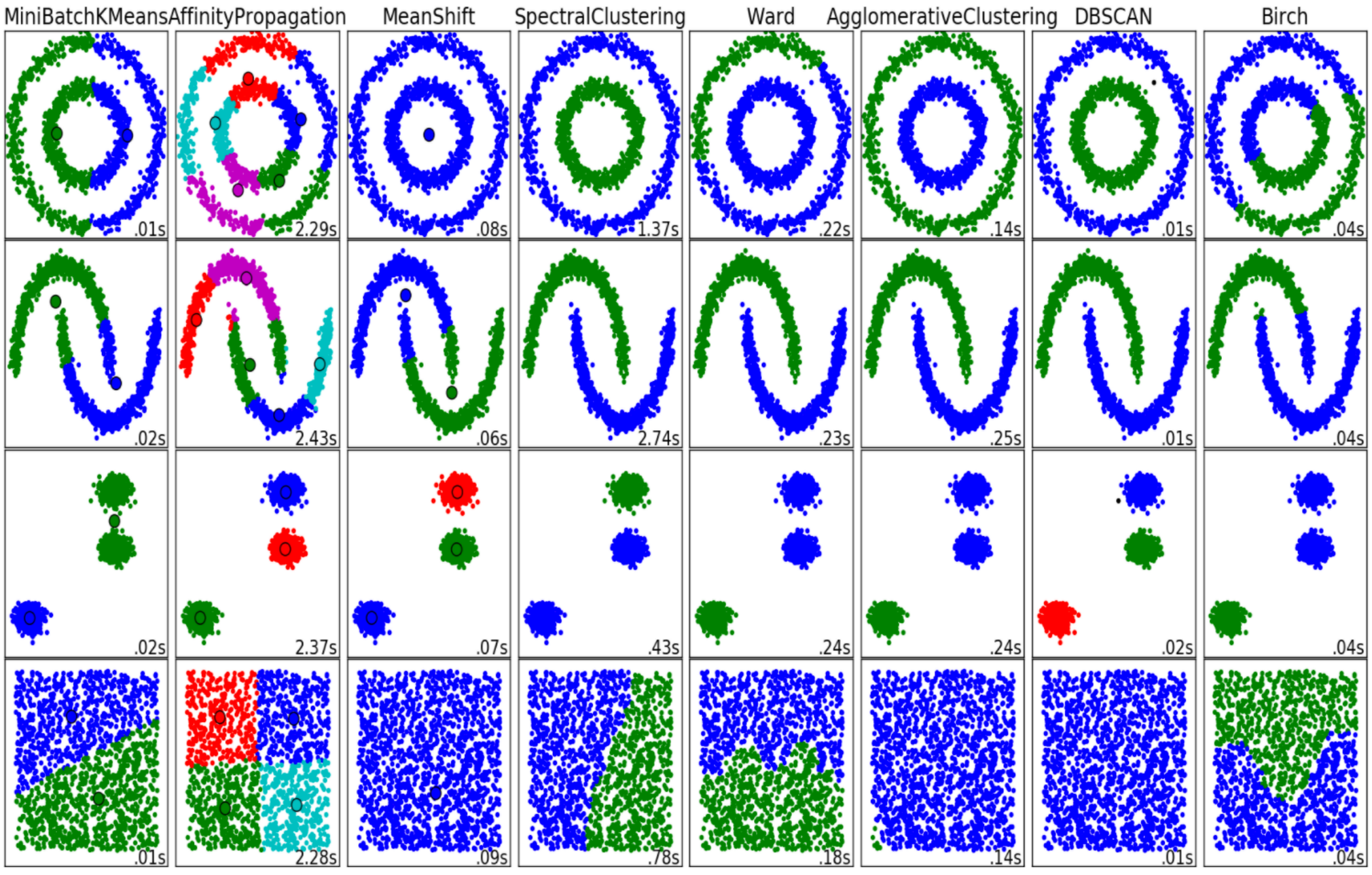
- The goal remains the same: create an algorithm to model the underlying ***spatial density*** of the data

- Murphy on Unsupervised Learning: "Unsupervised learning can be reformulated as the problem of density estimation" [Murphy, see below]

- Density-based methods "Let the data talk for itself" [Raftery et. all]

1. Murphy, Kevin P. Machine learning: a probabilistic perspective. MIT press, 2012.

# Comparison of common Clustering Algorithms



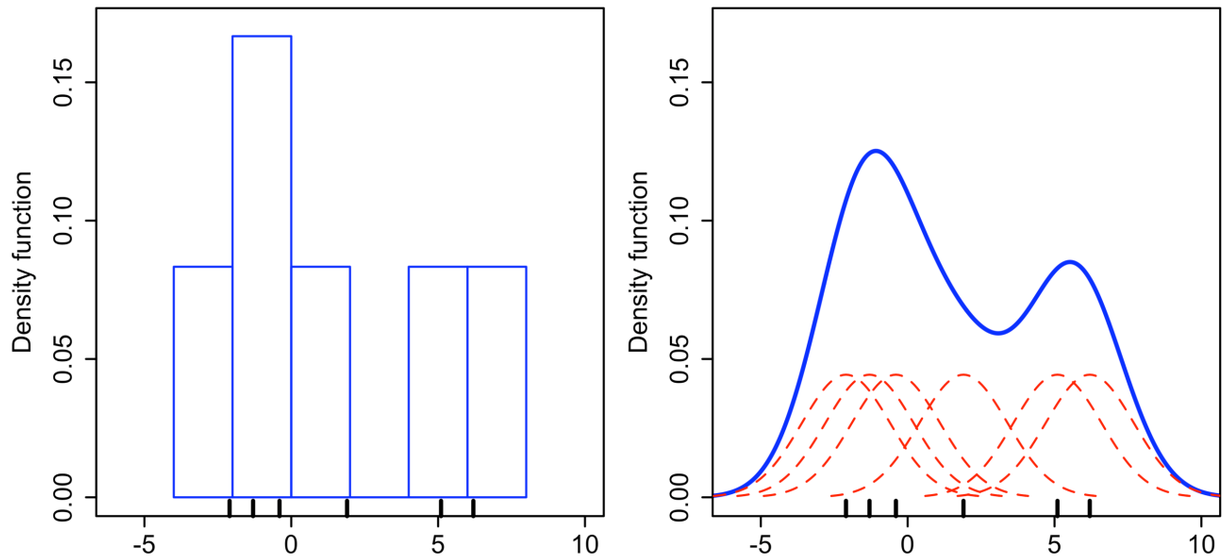

A comparison of the clustering algorithms in scikit-learn

Image from: http://scikit-learn.org/stable/modules/clustering.html

# DENCLUE

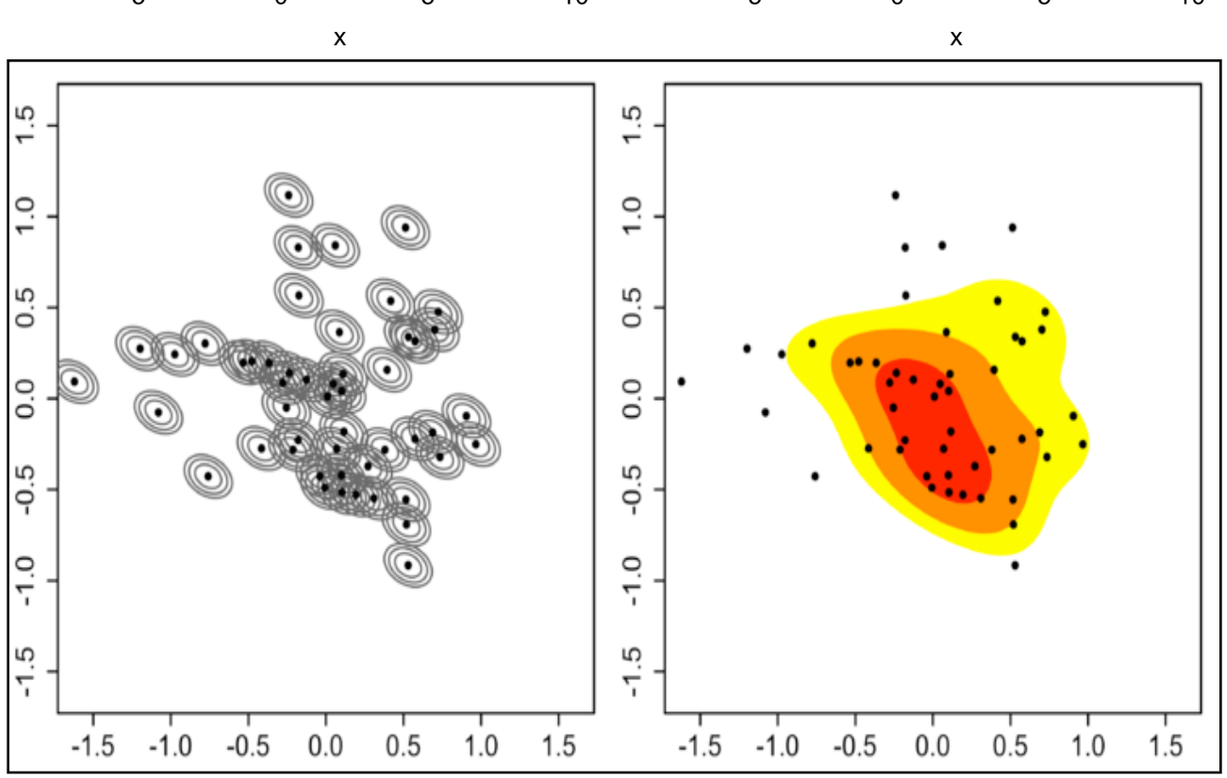
- DENsity CLUstering (DENCLUE) proposed in 1998, based on Kernel Density Estimation (KDE)

- Basic Idea: Use KDE to **approximate the density**, then **cut the density at some value.**




- The remaining **modes** distinguish which points are defined for each cluster

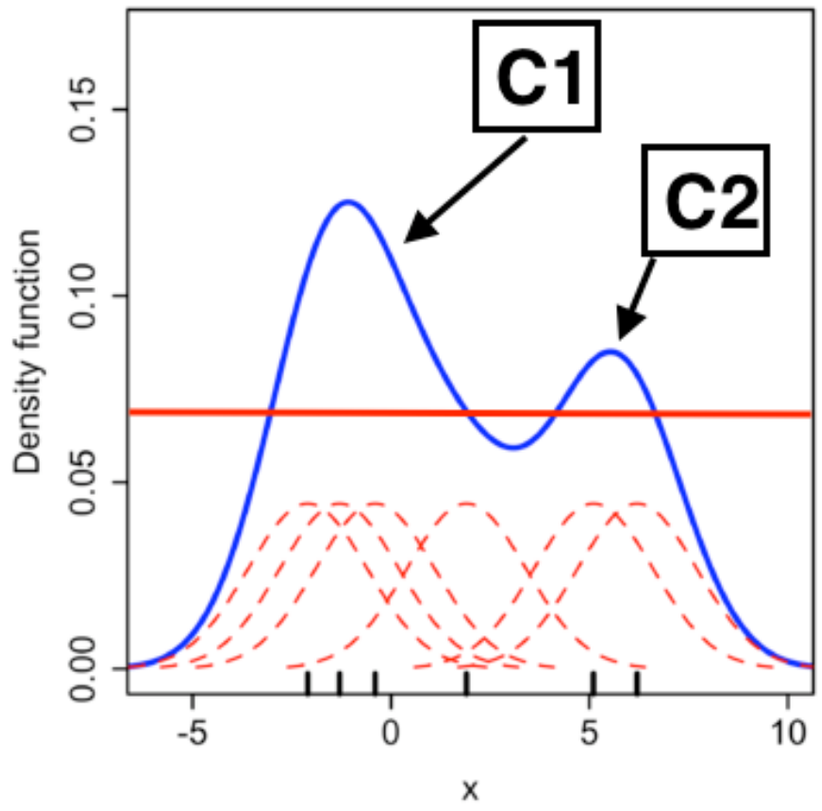
- Use of “Square-wave” kernel can be shown to match Single linkage and DBSCAN with right parameter settings

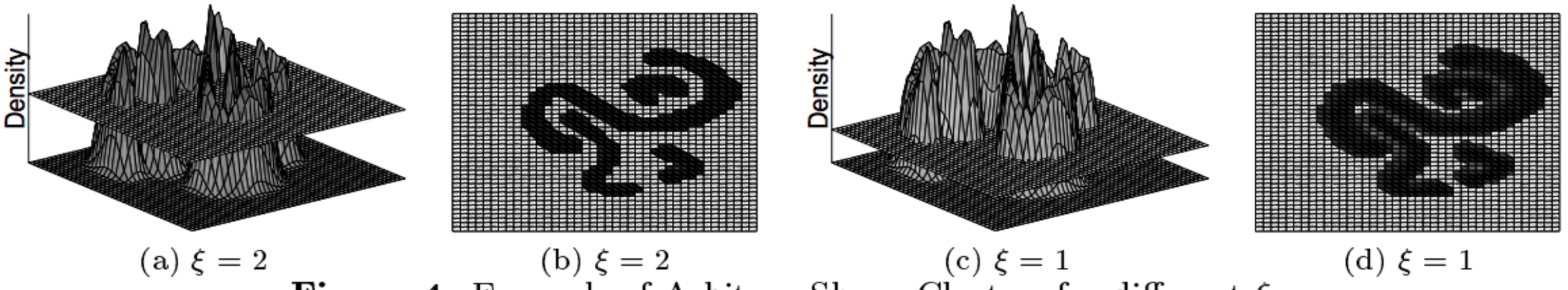



(top and bottom) Image from wikipedia: https://en.wikipedia.org/wiki/Multivariate_kernel_density_estimation

# DENCLUE

- Requires two crucial parameters:
  - $\sigma$: Which determines the shape ("smoothness") of the KDE
  - $\delta$: Density level which determines at what density above which modes distinguish the cluster grouping




(a) $\xi = 2$ (b) $\xi = 2$ (c) $\xi = 1$ (d) $\xi = 1$

**Figure 4:** Example of Arbitray-Shape Clusters for different $\xi$

1. (top) from [Consistent procedures for cluster tree estimation and pruning]
2. (bottom) from DENCLUE paper

# Density-based Clustering: DBSCAN

- Density-based Spatial Clustering of Applications with Noise (DBSCAN) Introduced in 1996 [DBSCAN]
- The idea: Given a distance threshold $\epsilon$ and minimum cluster size $minPts$
  - For a given point $p$, find all the neighbors of $p$ s.t.

$$N_\epsilon(p) = \{q \mid d(p,q) < \epsilon\}$$

- Points are then classified as follows:

**Definition 2. Point classes.** *A point $p$ is classified as*

- *a* **core point** *if $N_\epsilon(p)$ has high density, i.e., $|N_\epsilon(p)| \geq minPts$ where $minPts \in \mathbb{Z}^+$ is a user specified density threshold,*
- *a* **border point** *if $p$ is not a core point, but it is in the neighborhood of a core point $\hat{p}$, i.e., $p \in N_\epsilon(\hat{p})$, or*
- *a* **noise point**, *otherwise.*

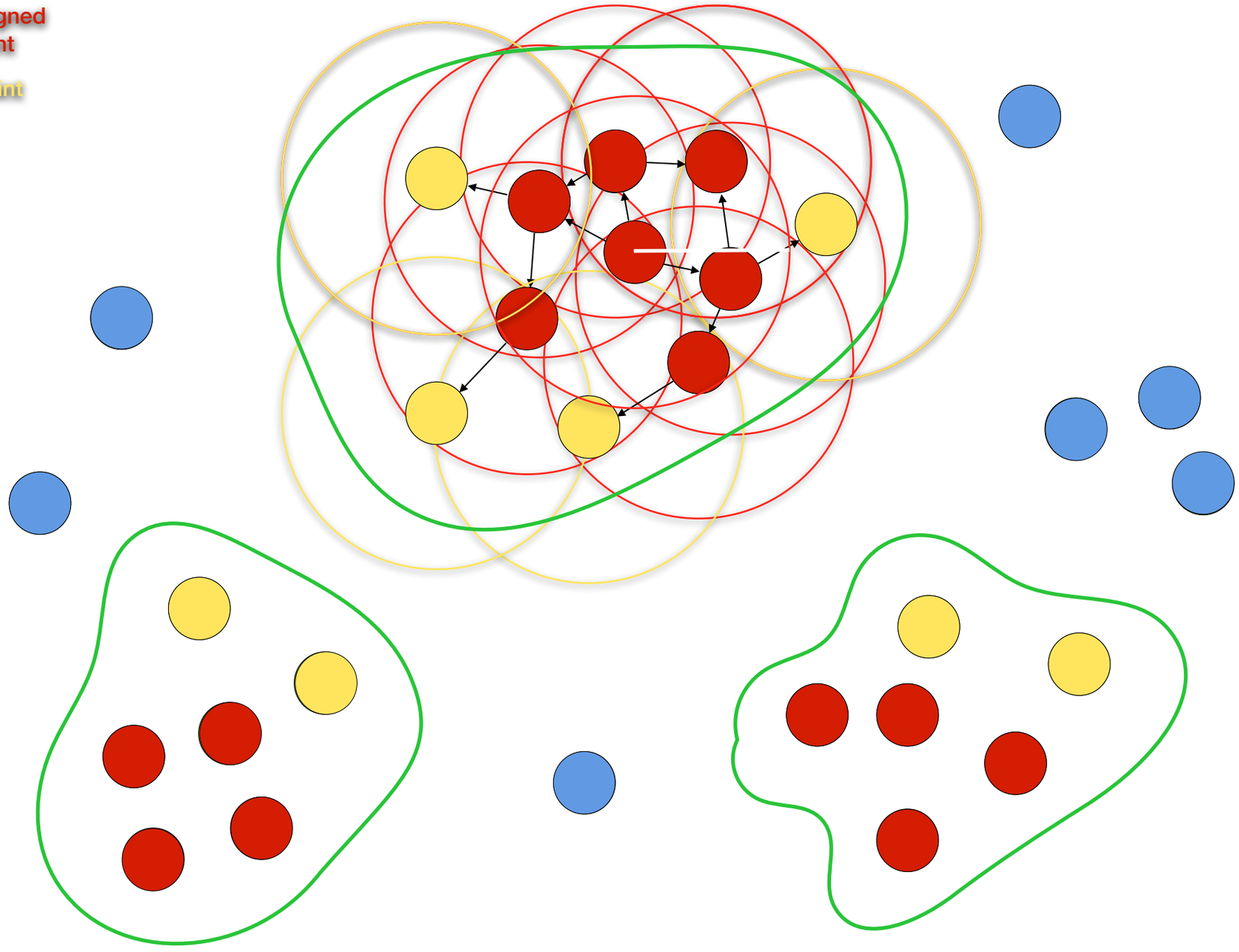
# DBSCAN

***minPts*** = 4 $\varepsilon$ = 1

= Point has been assigned a cluster assignment

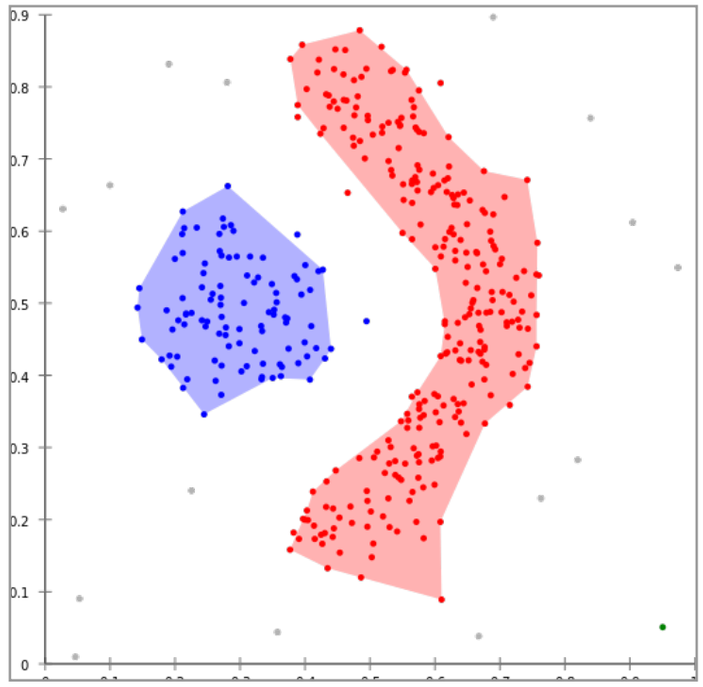
= Point is a border point

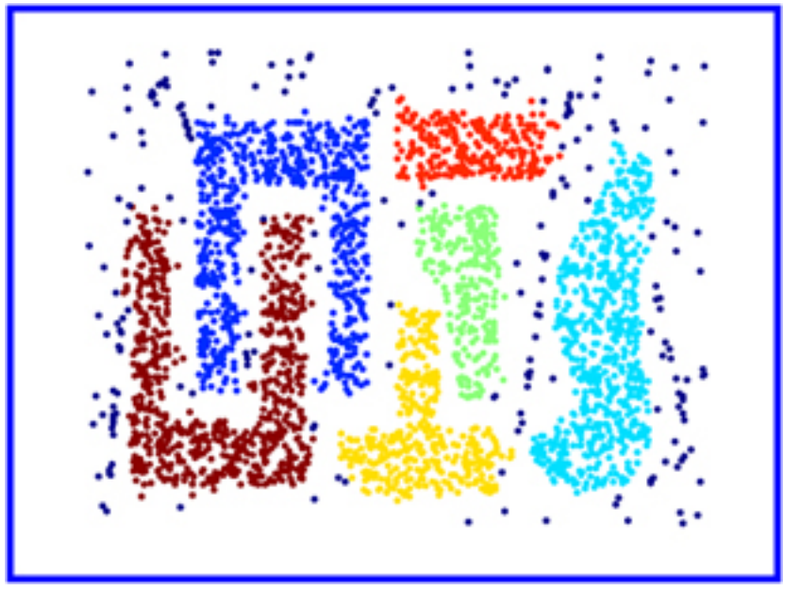
= Point is just noise



# DBSCAN Cont.

- Pros:
  - Ability to identify arbitrarily-shaped clusters
  - Has some configurable notion of noise
  - Can be made fairly efficient [see BA]
  - Intuitively easy to understand
- Cons:
  - Not *entirely* deterministic (border points)
  - **Cannot find clusters of varying density**
  - Not too much advice on how to set parameters
    - "In practice, we found setting *minPts* = around 10 or 15 to be useful for most applications, or much higher for higher dimensional datasets" [HDBSCAN 15]



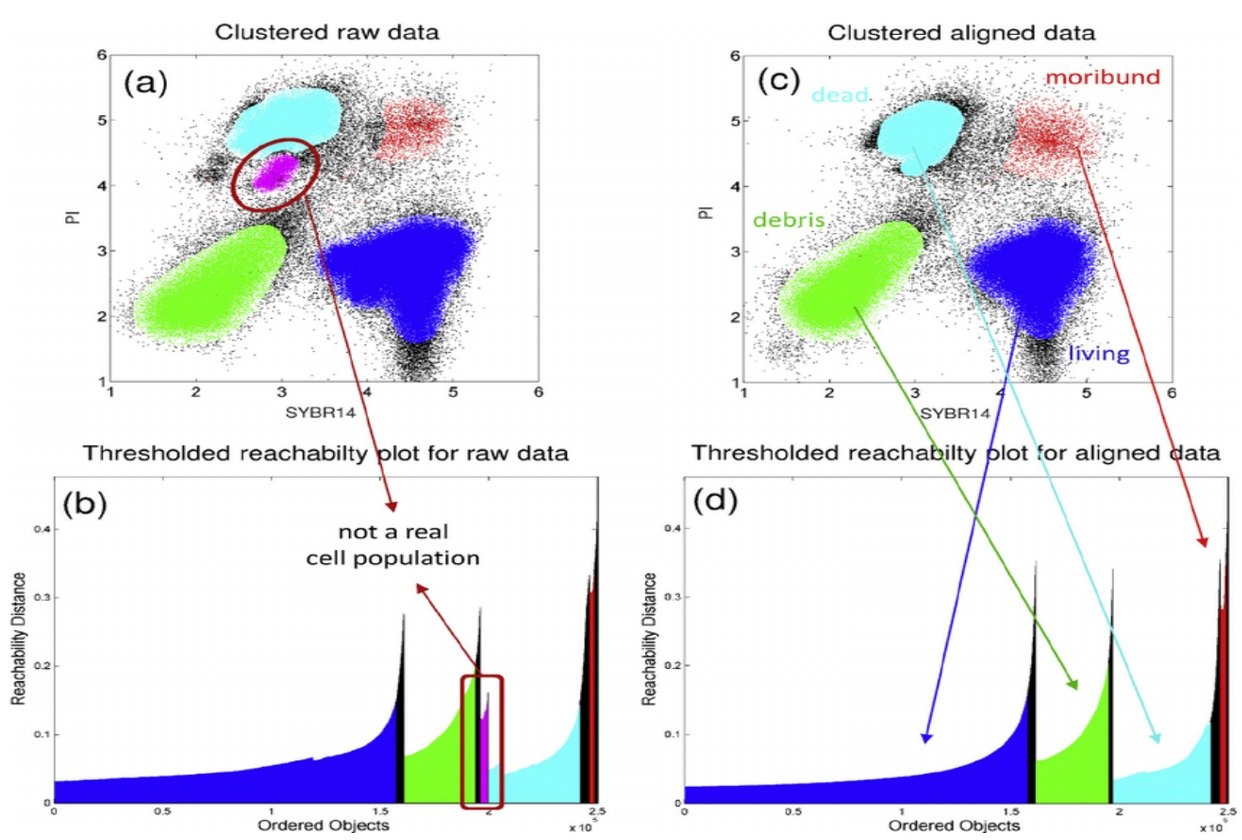
1. (top) from wikipedia https://en.wikipedia.org/wiki/DBSCAN
2. (bottom) from http://www.hypertextbookshop.com/dataminingbook/public_version/contents/chapters/chapter004/section004/blue/page003.html

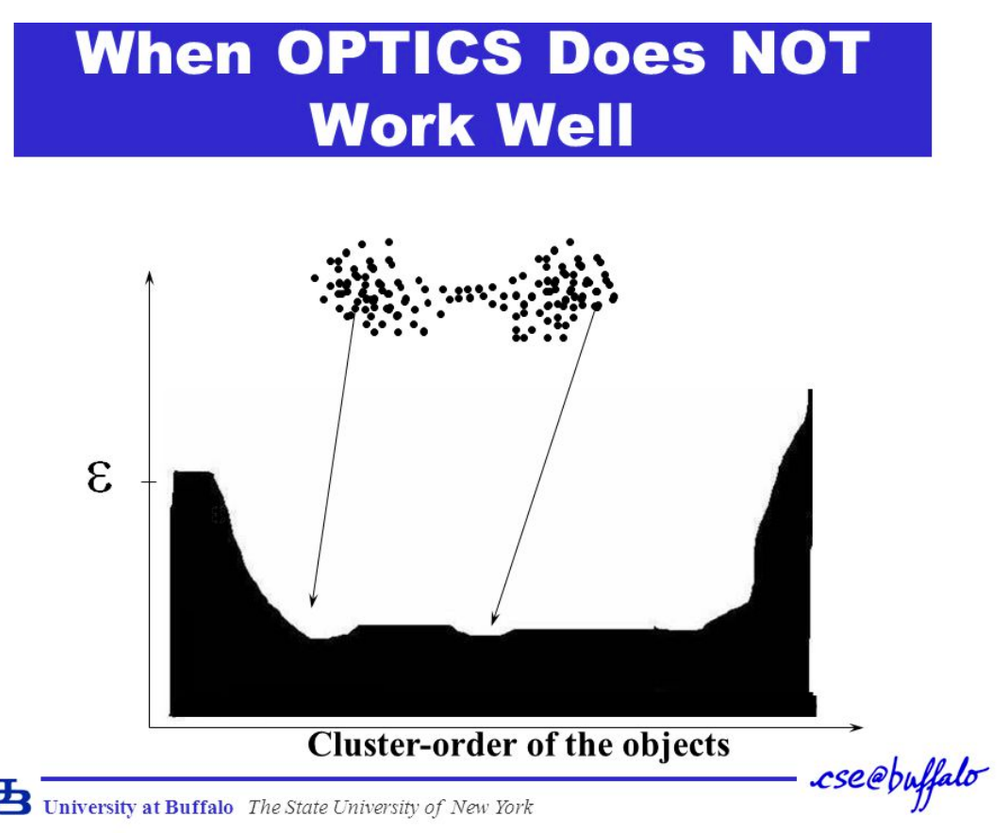
# Extensions to DBSCAN (or a small subset thereof)

- DBSCAN was remarkably successful, winning the KDD 2014 Test of Time award
- Extensions proposed for DBSCAN:
  - OPTICS (1999) [Ankerst et. all]
  - V-DBSCAN for data with varying density clusters (2007) [22]
  - ST-DBSCAN for spatiotemporal data (2007) [4]
  - P-DBSCAN for geotagged images (2010) [29]
  - C-DBSCAN for constraint-based DBSCAN (2007) [40]
  - TI-DBSCAN - DBSCAN that respects the triangle inequality (2010) [31]
  - Revised DBSCAN - Finds ‘dense adjacent clusters’ (2013) [38]
  - GMDBSCAN - Grid-based multi-density DBSCAN  (2008)
  - DBSCAN with adaptive parameter settings (2012) [42]
  - DBSCAN-GM - DBSCAN on GMM means (2012)
  - Soft-DBSCAN - DBSCAN + fuzzy learning (2013) [36]
  - GB-DBSCAN Grid-based DBSCAN (2012) [11]
  - AutoEPS-DBSCAN - DBSCAN that calculates eps automatically (2013) [17, 44]
  - $\gamma$-ray DBSCAN - DBSCAN specialized for gamma rays (2013) [37]
  - MSDBSCAN - Multi-density Scale-independent DBSCAN (2012) [14]
  - Numerous variants improving performance [6, 11, 27, 34, 39]
  - … + At least 50 more DBSCAN variants

# OPTICS

- OPTICS (Ordering of Points To Identify Clustering Structure) is an ***ordering algorithm*** that **linearly orders points** such that points that are ***spatially closest*** become neighbors in the ordering
  - Produces a dendrogram-like structure that can be used to:
    - Extract a **DBSCAN-like solution**
    - Extract a **condensed hierarchical clustering**



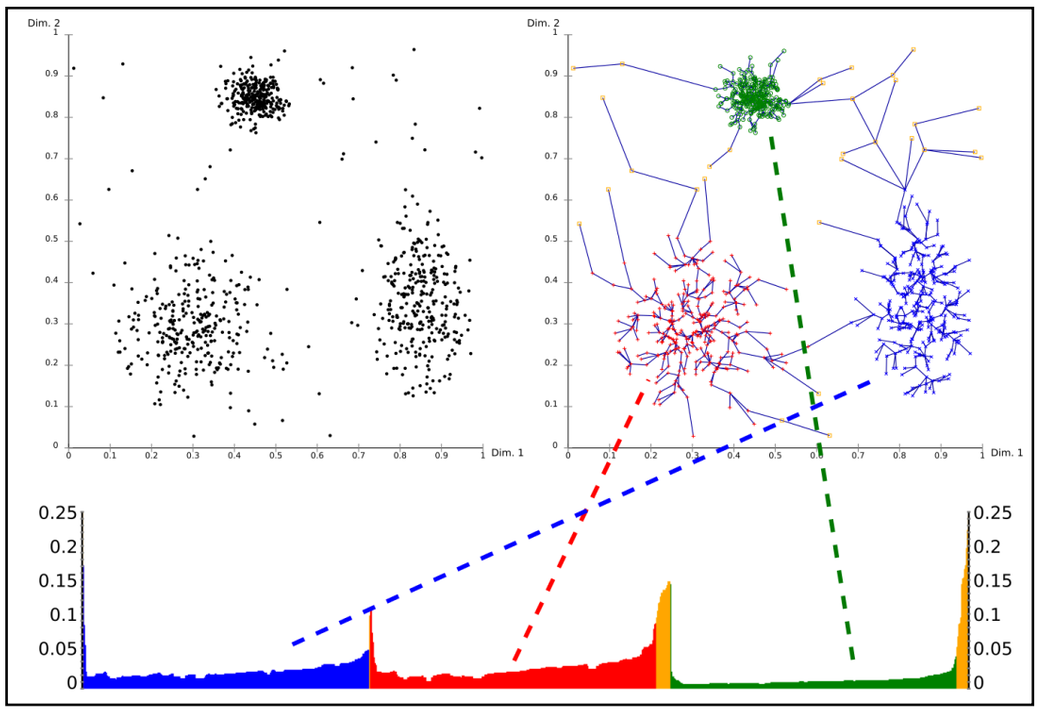

Left from: https://www.researchgate.net/figure/272240022_fig6_Fig-6-OPTICS-clustering-of-raw-and-normalized-fl-uorescence-data-for-the-viability
Right from: http://slideplayer.com/slide/4238890/

# OPTICS cont.

- Proposed by some of the original authors of DBSCAN
- Orders points by minimum reachability distance
- Produces visualization called ‘reachability-plot’, which was shown to be equivalent to a dendrogram built from reachability distance in 2003
- Pros:
  - *Slightly* more robust to **variations in density**
  - Visualization arguably easier for large datasets

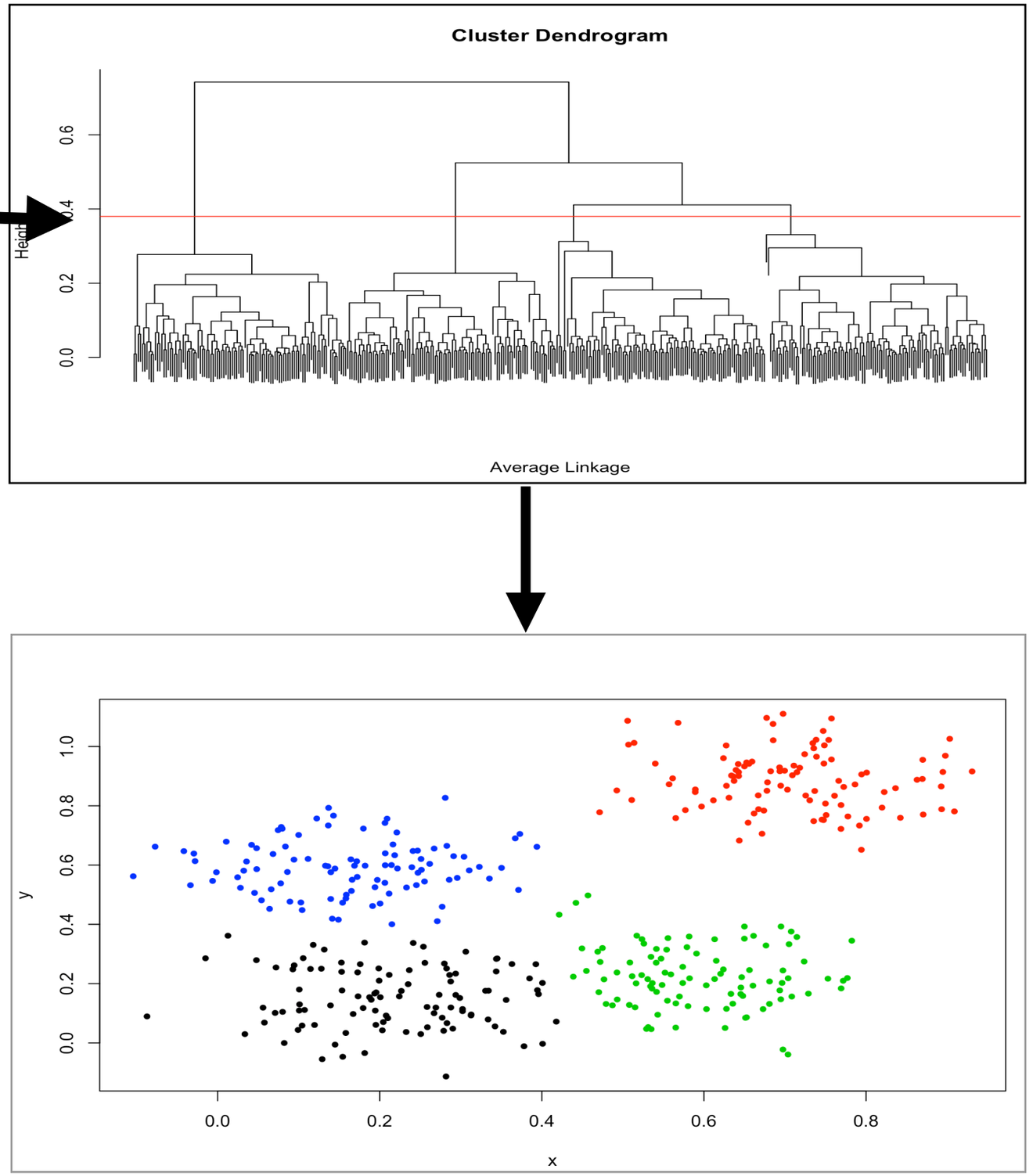
$$\text{core-dist}_{\varepsilon,MinPts}(p) = \begin{cases} \text{UNDEFINED} & \text{if } |N_\varepsilon(p)| < MinPts \\ MinPts\text{-th smallest distance to } N_\varepsilon(p) & \text{otherwise} \end{cases}$$

$$\text{reachability-dist}_{\varepsilon,MinPts}(o,p) = \begin{cases} \text{UNDEFINED} & \text{if } |N_\varepsilon(p)| < MinPts \\ \max(\text{core-dist}_{\varepsilon,MinPts}(p), \text{dist}(p,o)) & \text{otherwise} \end{cases}$$



# Hierarchical Clustering

- Hierarchical cluster involves building a hierarchy of (potentially overlapping) clusters.
  - Agglomerative (“bottom-up”)
  - Divisive (“top-down”)
- Once the hierarchy is built, cluster extraction is done by selecting some height to ‘cut’



# Hierarchical Clustering

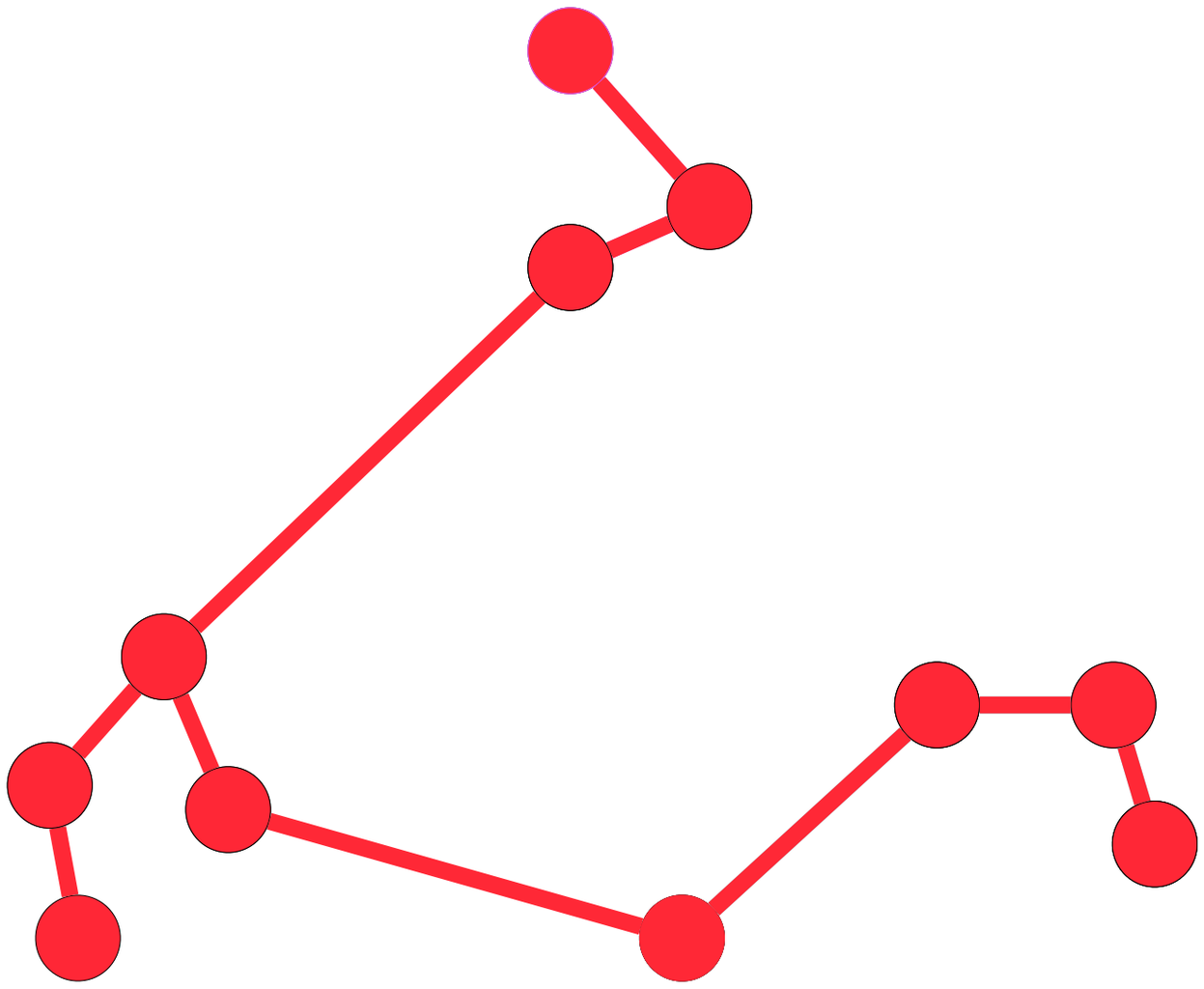
- Deciding how to *merge* or *split* clusters depends on 1) the **similarity measure/metric** and 2) the **linkage criterion**
    1. Euclidean Distance, Mahalanobis distance, etc.
    2. Could be:
        - **Distance based**: Max/min/mean distance (complete/single/average linkage)
        - **Probability based**: the probability that the candidate clusters come from the same distribution
        - The product of the **in-and-out degree** of a node (graph degree linkage)
        - **Instance-based constraints** (see Wagstaff 2002)

# Hierarchical Clustering: Single Linkage

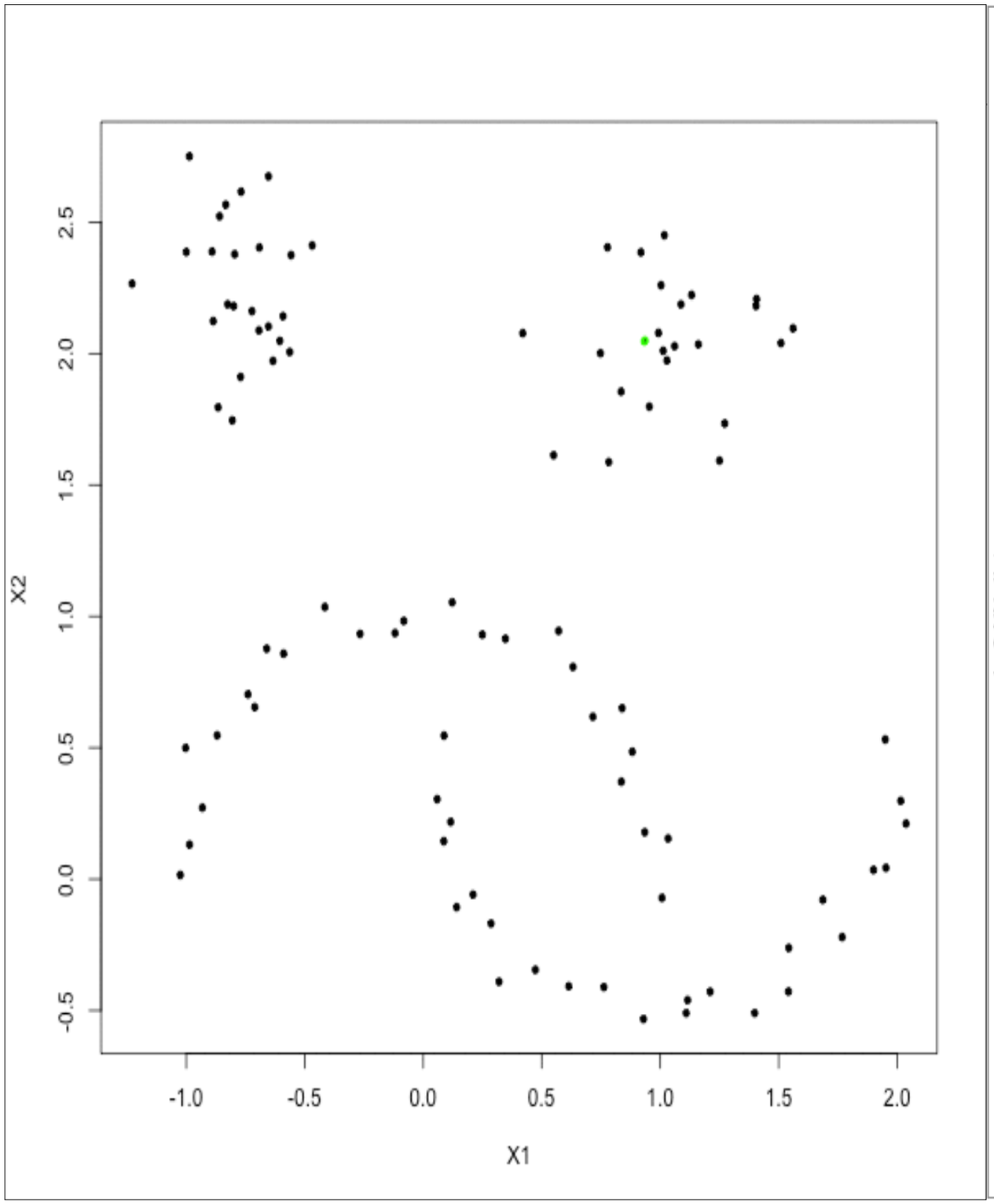
- Perhaps the most well-known hierarchical clustering algorithm is called the **Single-Linkage algorithm**
  - **The idea:**
    - Start with every single point in it's own cluster (*n* clusters)
    - Merge the *two closest points* to form a new cluster
    - Repeat: each time *merging the two closest pair of points*
- **The configuration:**
  - The **similarity metric** is **Euclidean distance**
  - The **linkage criterion** is the [pairwise] minimum distance

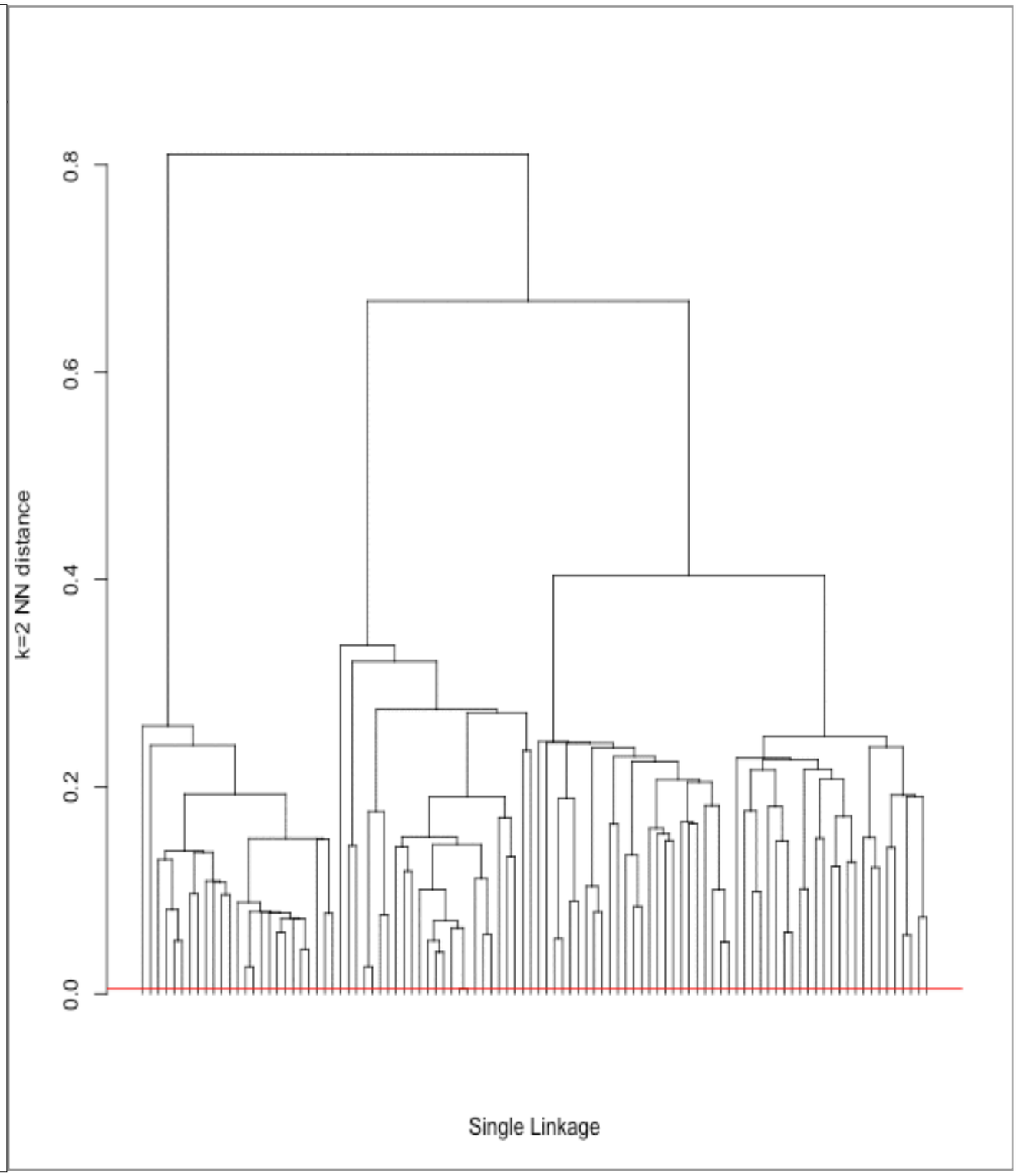
# Hierarchical Clustering: Single Linkage

19 Clusters



# Hierarchical Clustering: Single Linkage Example



# Hierarchical Clustering

- Definitions of how to define "structure" and "similarity" vary quite a bit. Should structure be identified by:
  - Euclidean or other vector-based distance (SSQ)?
  - Some form of semantic similarity?
  - Some notion of density?
  - …
- Even if distance metric and linkage criterion are known…
  - Each method generally has **one or more** parameter settings
  - Some are extremely unintuitive
- There could exist **numerous configurations**, each of which may be better-or-worse depending on the application
  - Is choice of the **proper configuration *arbitrary***?
  - How to determine the '**optimal**' configuration?

# Hierarchical Clustering: Wards Criterion

- Rather than use distance metrics as measures of similarity, why not merge based on a more statistically intuitive notion

- Perhaps its fair to assume observations come from an approximately elliptically-shaped distribution

- Ward proposed agglomeration where observations are merged in such as way that minimizes the total squared error (maximizes $r^2$ at each merge)

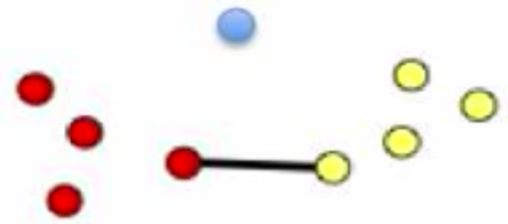
  - "…Interpreted as the **proportion of variation explained by a particular clustering of the observations.**"

- He also proposed that really any objective function could be used, "any function that reflects the investigator's purpose."

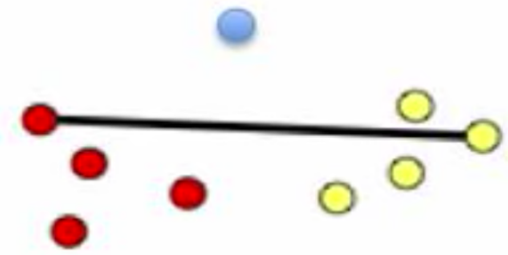
Nice article on ANOVA and Wards Criterion:
http://sites.stat.psu.edu/~ajw13/stat505/fa06/19_cluster/09_cluster_wards.html

# Cluster distance measures

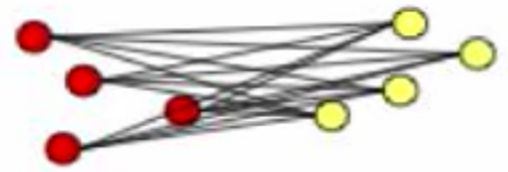
- Single link: $D(c_1, c_2) = \min_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$
  - distance between closest elements in clusters
  - produces long chains a→b→c→…→z



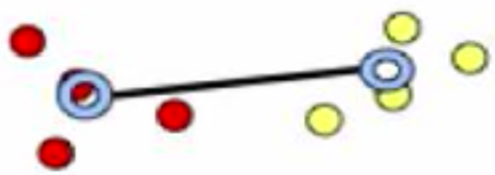
- Complete link: $D(c_1, c_2) = \max_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$
  - distance between farthest elements in clusters
  - forces "spherical" clusters with consistent "diameter"



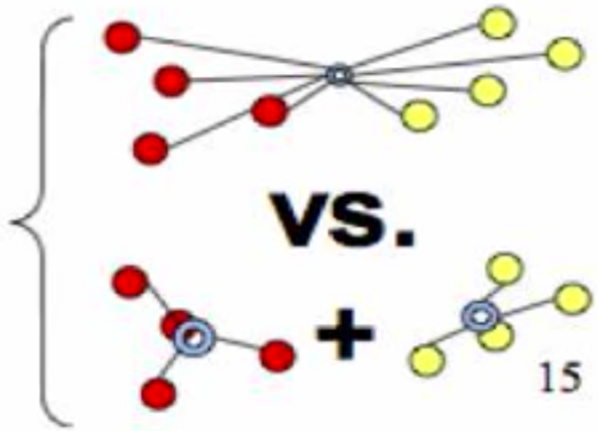
- Average link: $D(c_1, c_2) = \frac{1}{|c_1|}\frac{1}{|c_2|}\sum_{x_1 \in c_1}\sum_{x_2 \in c_2} D(x_1, x_2)$
  - average of all pairwise distances
  - less affected by outliers



- Centroids: $D(c_1, c_2) = D\left(\left(\frac{1}{|c_1|}\sum_{x \in c_1}\vec{x}\right), \left(\frac{1}{|c_2|}\sum_{x \in c_2}\vec{x}\right)\right)$
  - distance between centroids (means) of two clusters



- Ward's method: $TD_{c_1 \cup c_2} = \sum_{x \in c_1 \cup c_2} D(x, \mu_{c_1 \cup c_2})^2$
  - consider joining two clusters, how does it change the total distance (TD) from centroids?






Above from: https://www.youtube.com/watch?v=VMyXc3SiEqs

# Examples

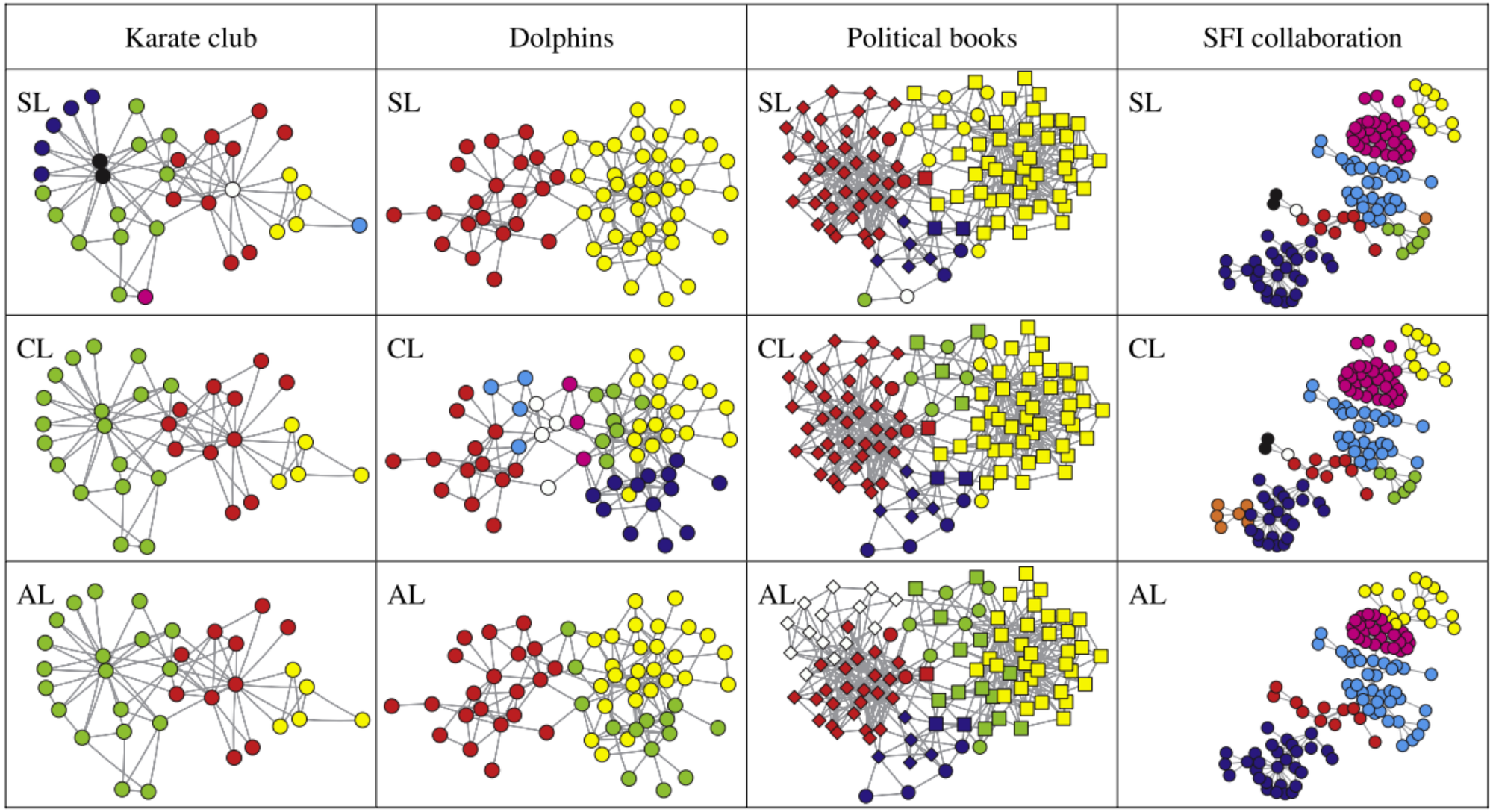



**Figure 6.** The partitioning results detected by single linkage (SL), complete linkage (CL) and average linkage (AL) for the real-world networks.

Image from: J. Liu and T. Liu. Coarse-grained diffusion distance for community structure detection in complex networks. Journal of Statistical Mechanics: Theory and Experiment, 2010(12):P12030, 2010

# Clustering Validation

- For many, Ward's criterion does not give satisfactory results, especially when the "optimal" clustering **is a "natural" cluster of arbitrary shape**
- An active research field attempts to design "validation measures" that attempt to assess the quality of a clustering (see Jain et. all)
- Different types of validation measures have been proposed
  - External: Given supervised "truth" data, find the best clustering algorithms prior to working on unsupervised data
  - Internal: Intrinsic to the data itself. Assesses how well the resulting clustering "fits" a predefined cluster structure
  - Stability-based:  Attempts to measure how consistent the clustering algorithm performs using different parameter settings

# Examples of Internal Validation Measures

Given a dataset $X$ of $N$ objects in $d$-dimensional space,

$$X = \{x_1, x_2, \ldots, x_N\} \subseteq \mathbb{R}^d$$

... a clustering or partitioning in $X$ is a set of disjoint clusters that partition $X$ into $K$ groups:

$$C = \{c_1, c_2, \ldots, c_K\}$$

where $\overline{X} = 1/N \sum_{x_i \in X} x_i$ and $\overline{C_K} = 1/\mid C_K \mid \sum_{x_i \in C_K} x_i$

$$d_{euc}(q,p) = \text{sqrt}\Big(\sum_{i=1}^{d}(q_i - p_i)^2\Big)$$

$$Diam(C_K) = \max_{x_i, x_j \in C_K}(d_{euc}(x_i, x_j))$$

$$NN(C_K, C_l) = \min_{x_i \in C_K} \min_{x_j \in C_l} d_{euc}(x_i, x_j)$$

$$AvgIntraDist(x_i, C_K) = 1/\mid C_K \mid \sum_{x_j \in C_K} d_{euc}(x_i, x_j)$$

1. **Dunn Index**
2. **Calinski-Harabasz**
3. **Davies-Bouldin**
4. **Silhouette Index**

$$Dunn(C) = \frac{\min_{C_K \in C} \min_{C_l \in C \setminus C_K} NN(C_K, C_l)}{\max_{C_K \in C}(Diam(C_K))}$$

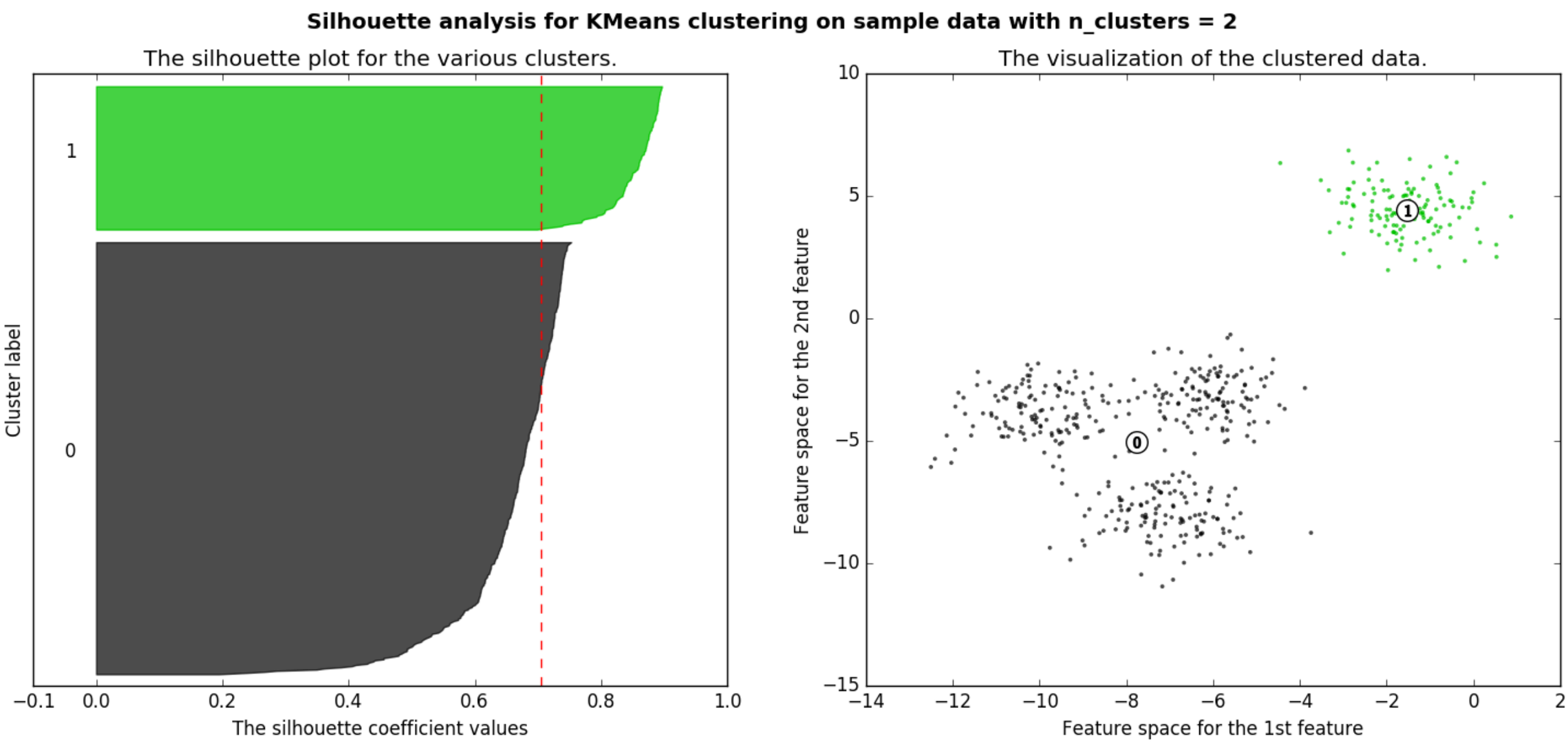
$$CH(C) = \frac{N-K}{K-1} \frac{\sum_{C_K \in C} \mid C_K \mid d_{euc}(\overline{C_K}, \overline{X})}{\sum C_K \in C \sum_{x_i \in C_K} d_{euc}(x_i, \overline{C_K})}$$

$$DB(C) = \frac{1}{K} \sum_{C_K \in C} \max_{C_l \in C \setminus C_K} \left(\frac{S(C_K) + S(C_l)}{d_{euc}(\overline{C_K}, \overline{C_l})}\right)$$

where $S(C_K) = 1/\mid C_K \mid \sum_{x_i \in C_K} d_{euc}(x_i, \overline{C_K})$

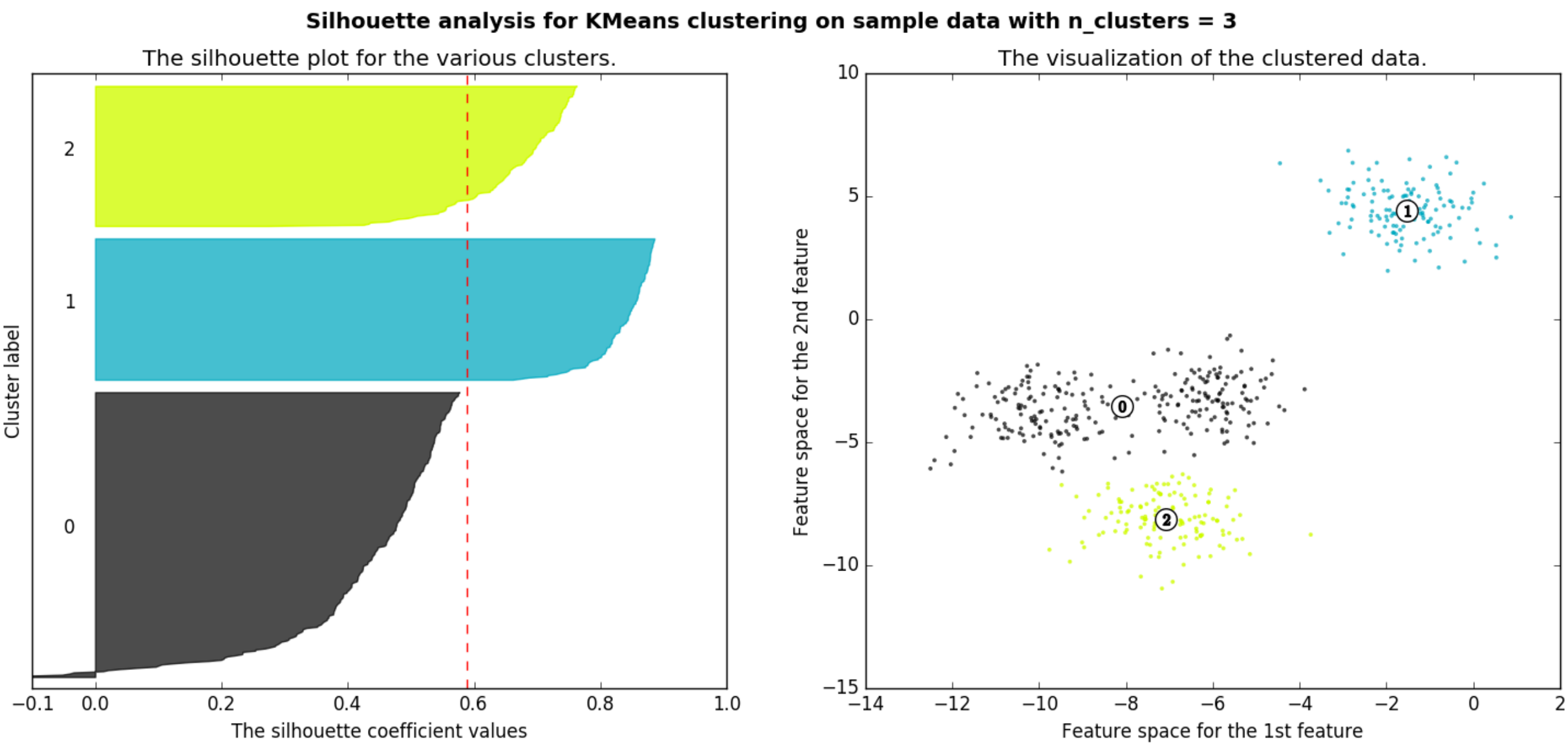
$$Sil(C) = 1/N \sum_{C_K \in C} \sum_{x_i \in C_K} \frac{NN(x_i, C_K) - AvgIntraDist(x_i, C_K)}{\max(AvgIntraDist(x_i, C_K), NN(x_i, C_K))}$$

# Clustering Validation Example: Silhouette Index



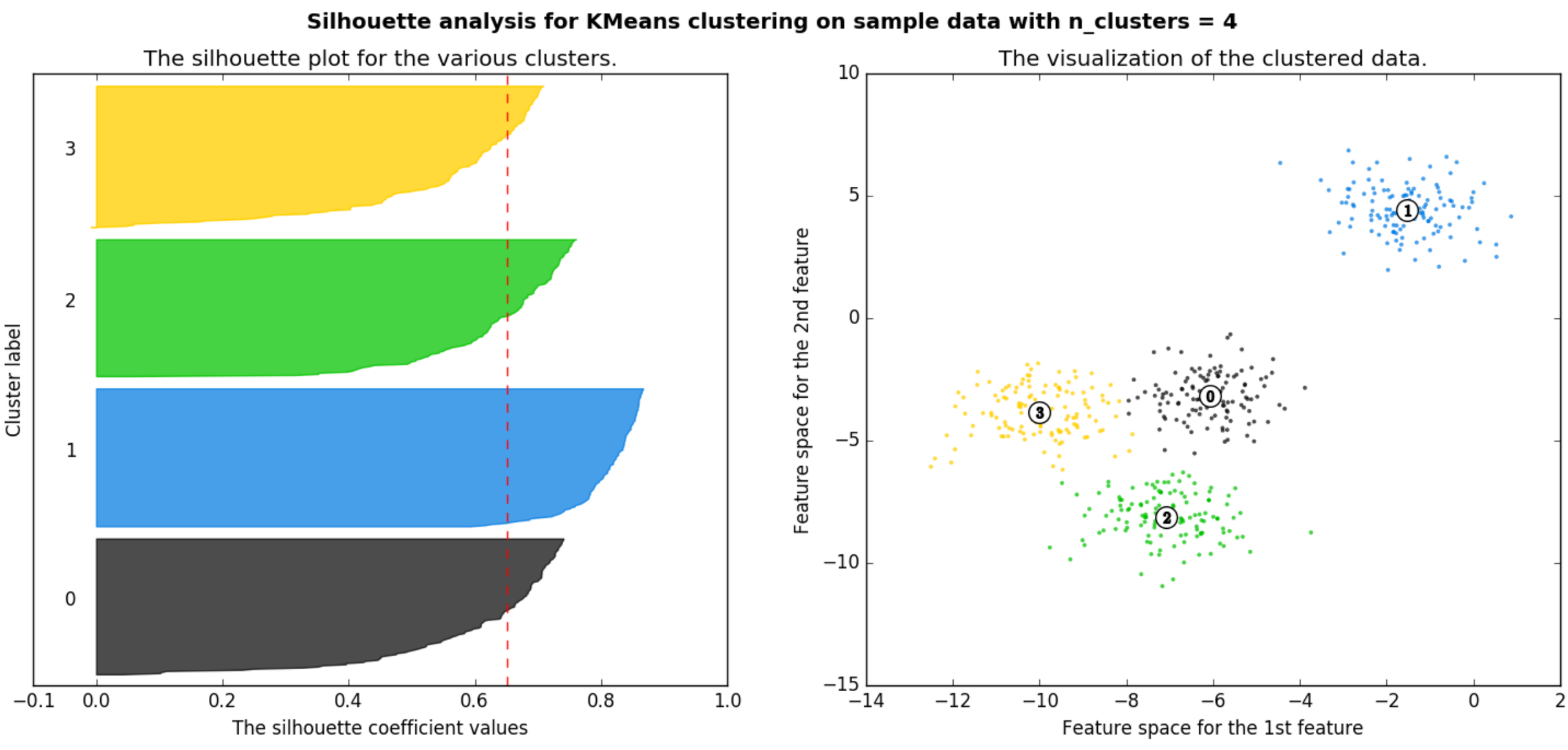

Example from: http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

# Clustering Validation Example: Silhouette Index



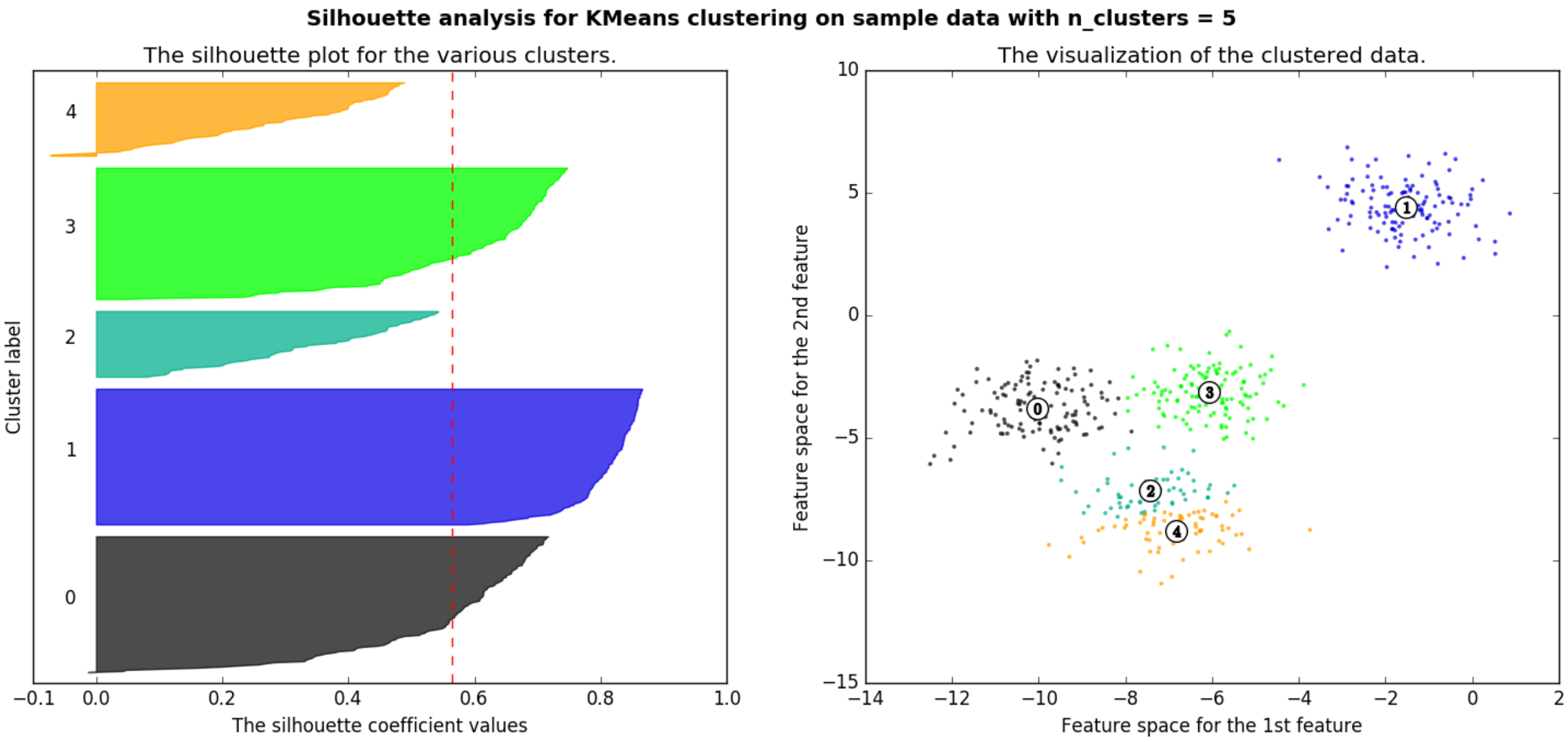

Example from: http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

# Clustering Validation Example: Silhouette Index




Example from: http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

# Clustering Validation Example: Silhouette Index




Example from: http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

# Clustering Validation Example: Silhouette Index

- For n_clusters = 2 The average silhouette_score is : 0.704978749608

- For n_clusters = 3 The average silhouette_score is : 0.588200401213

- For n_clusters = 4 The average silhouette_score is : 0.650518663273

- For n_clusters = 5 The average silhouette_score is : 0.563764690262

Example from: http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

# Clustering Validation: Internal Validation Indices

- Which validation measure performs uniformly the '**best**'?
- Attempts to comprehensively generate data "representative" of most data with a "fixed hyper-cubic sampling window" [3]:
  - Fixed number of clusters
  - Dimensionality
  - Amount of cluster overlap
  - Cluster density
  - Noise level
- 6480 configurations later, after testing 30 validation indices

- The question remains: if synthetic data are generated using known distributions, even with parameter settings, isn't the 'best' validation index going to correspond with how similar the validation measure is with the generative model?

# Internal Validation Measures

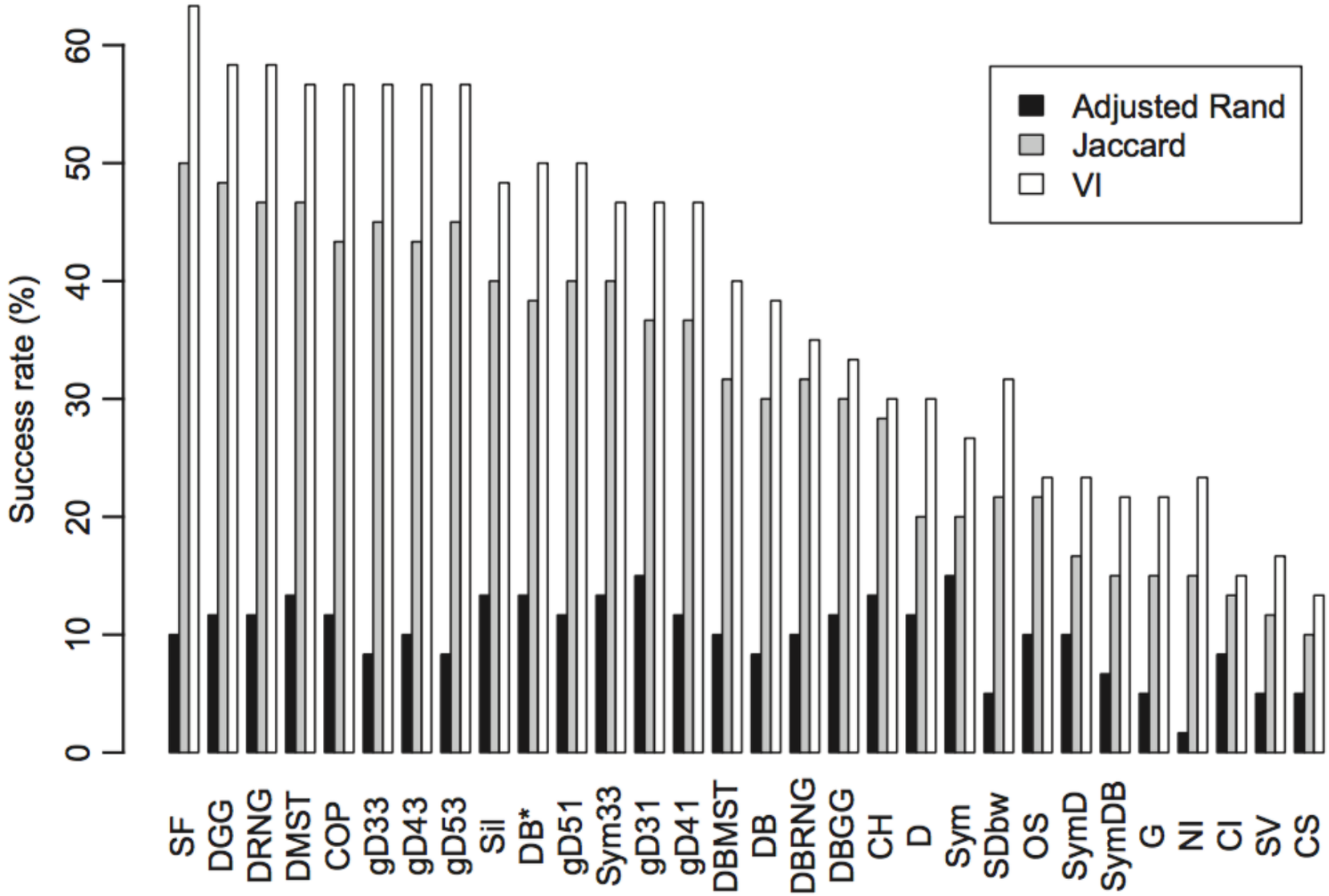
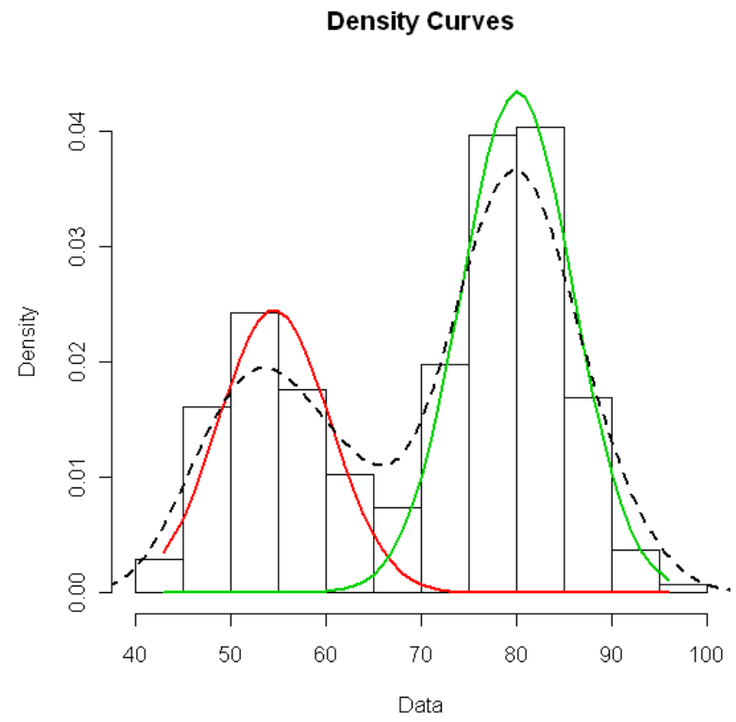


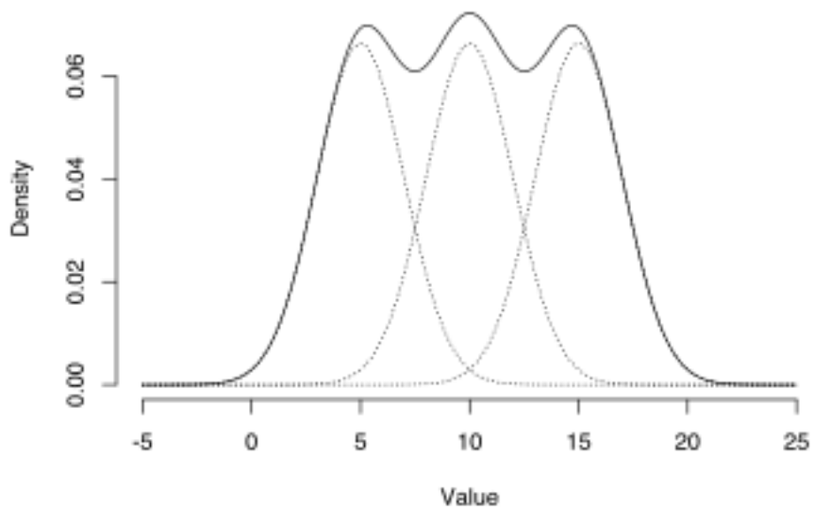

**Fig. 11.** Results for real datasets broken down by partition similarity measure.

Time Permitting

# Detour 1: Mixture Modeling

- Mixture modeling is used to represent the presence of subpopulations within an overall population
- Represents a "mixture" of probability distributions
- Represents a particular difficulty in density-based clustering
  - What if no assumptions are made on the modality of the underlying distribution
  - Nor on the form/shape of the distribution
- The question remains: How to determine whether a particular set of "group" of data came from a distribution?




Picture above from: https://www.r-bloggers.com/fitting-mixture-distributions-with-the-r-package-mixtools/
(below) from wikipedia: https://en.wikipedia.org/wiki/Mixture_distribution

# John Hartigan

- Published a number of seminal papers on the statistical nature of clustering

## John Hartigan

Eugene Higgins Professor of Statistics (Emeritus)

**Address:**
24 Hillhouse Ave, New Haven, CT 06511-6814
203-432-0666
john.hartigan@yale.edu
Website

**Research Interests:**
Foundations of probability and statistics, Bayes theory, classification, statistical computing, graphical methods.


- [22] J. Hartigan. Estimation of a convex density contour in two dimensions. Journal of the American Statistical Association, 82(397):267{270, 1987.
- [23] J. A. Hartigan. Direct clustering of a data matrix. Journal of the american statistical association, 67(337):123{129, 1972.
- [24] J. A. Hartigan. Consistency of single linkage for high-density clusters. Journal of the American Statistical Association, 76(374):388{394, 1981.
- [25] J. A. Hartigan. Statistical theory in clustering. Journal of classication, 2(1):63{76, 1985.
- [26] J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1):100{108, 1979

# Hartigan: Consistency of Single Linkage

- A "High Density Cluster" is defined as the set of all $x \in X$ defined on a population density $f$ in $\mathbb{R}^d$s.t.

$$\{x : f(X) \geq \lambda\} \text{ for all } \lambda > 0$$

- The hierarchy of all maximally connected density-level sets represents the what is now referred to as the *cluster tree*

- Hartigan Posed with the question of is single-linkage consistent in identifying disjoint, high density clusters asymptotically?

- Proof of asymptotic consistency related problem from percolation theory proposed by Broadbent and Hammersley:
  - "…if small spheres are removed at random from a solid, at which density of spheres will water begin to flow through the solid" [Hartigan '81]

# Hartigan: Consistency of Single Linkage

- “It is important to notice that the inconsistency of density estimation by "nearest neighbor" does not translate immediately into inconsistency of clustering” … [Hartigan]

- “If our task is to identify high-density clusters, it is necessary to estimate the density *f* **or at least to estimate order relationships between the density at different points.**” [Hartigan]

- “The vast literature on density estimation (for example, Wegman 1972a,b) now becomes relevant. **The difficulties with determining a suitably shaped kernel in density estimation are analogous to the difficulties of determining a suitable distance measure in clustering.**”[Hartigan]

# Hartigan: Consistency of Single Linkage

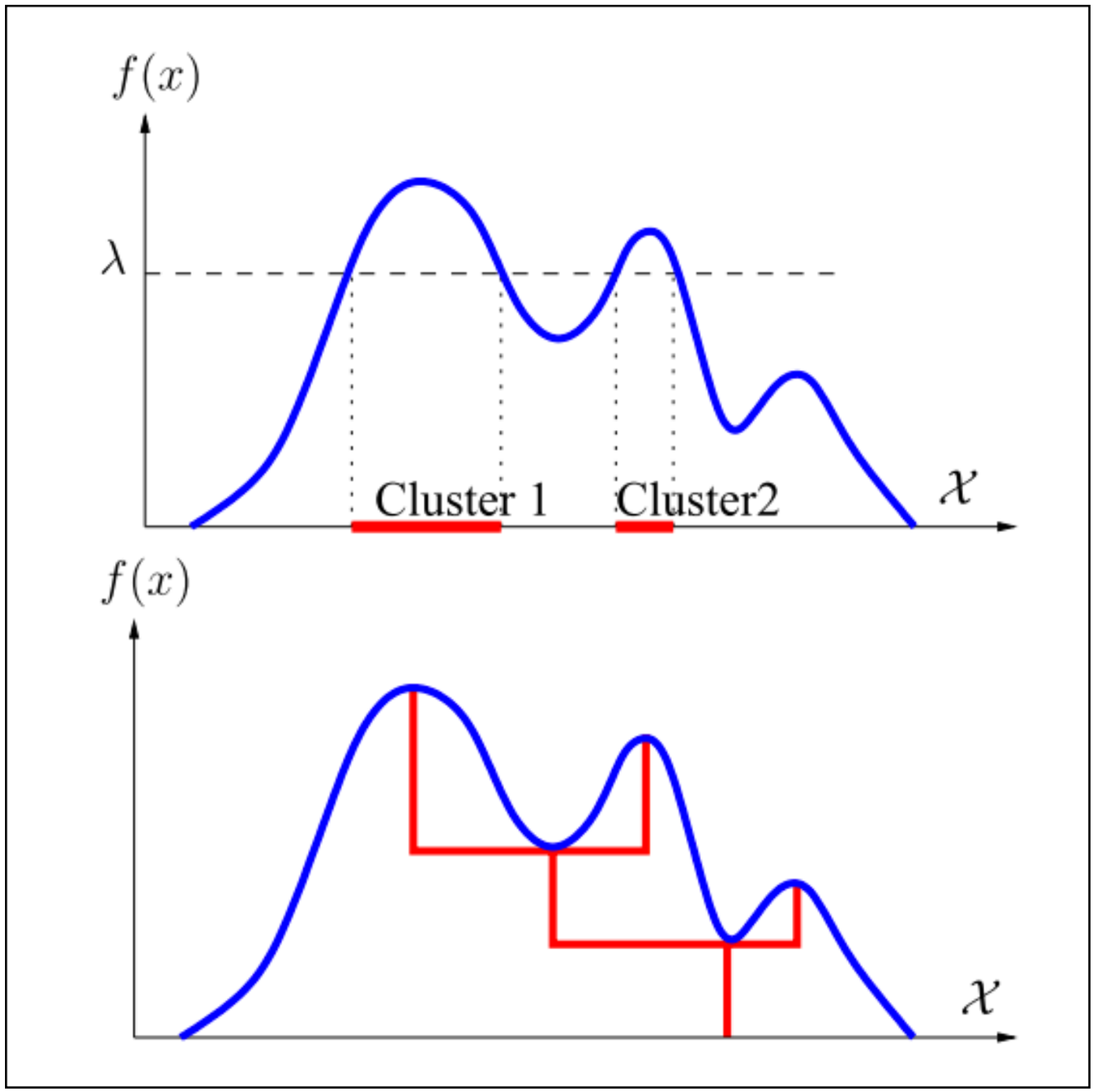
- Given some estimator $\Theta_n$ of the cluster tree $f$, said estimator is consistent iff:
- For any sets $A$, $A' \subset \mathbb{R}^d$ let $A_n$ (respectively $A'_n$) denote the smallest cluster of $\Theta_n$ containing samples in $A$ (respectively $A'$)
- $\Theta_n$ is consistent if, whenever $A$ and $A'$ are different connected components of $\{x : f(X) \geq \lambda\}$ for some $\lambda > 0$,

$$P(A_n \text{ is disjoint from } A') \to 1 \text{ as } n \to \infty$$

- Simply: if there exists high density clusters, is said to be consistent if asymptotically said estimator can distinguish such clusters, or at least a positive fraction [ largest possible ], passing arbitrarily close to all points in respective clusters

Much of the material here borrowed from: https://normaldeviate.wordpress.com/2012/11/23/the-density-cluster-tree-a-guest-post/

# Consistency of Single Linkage Clustering



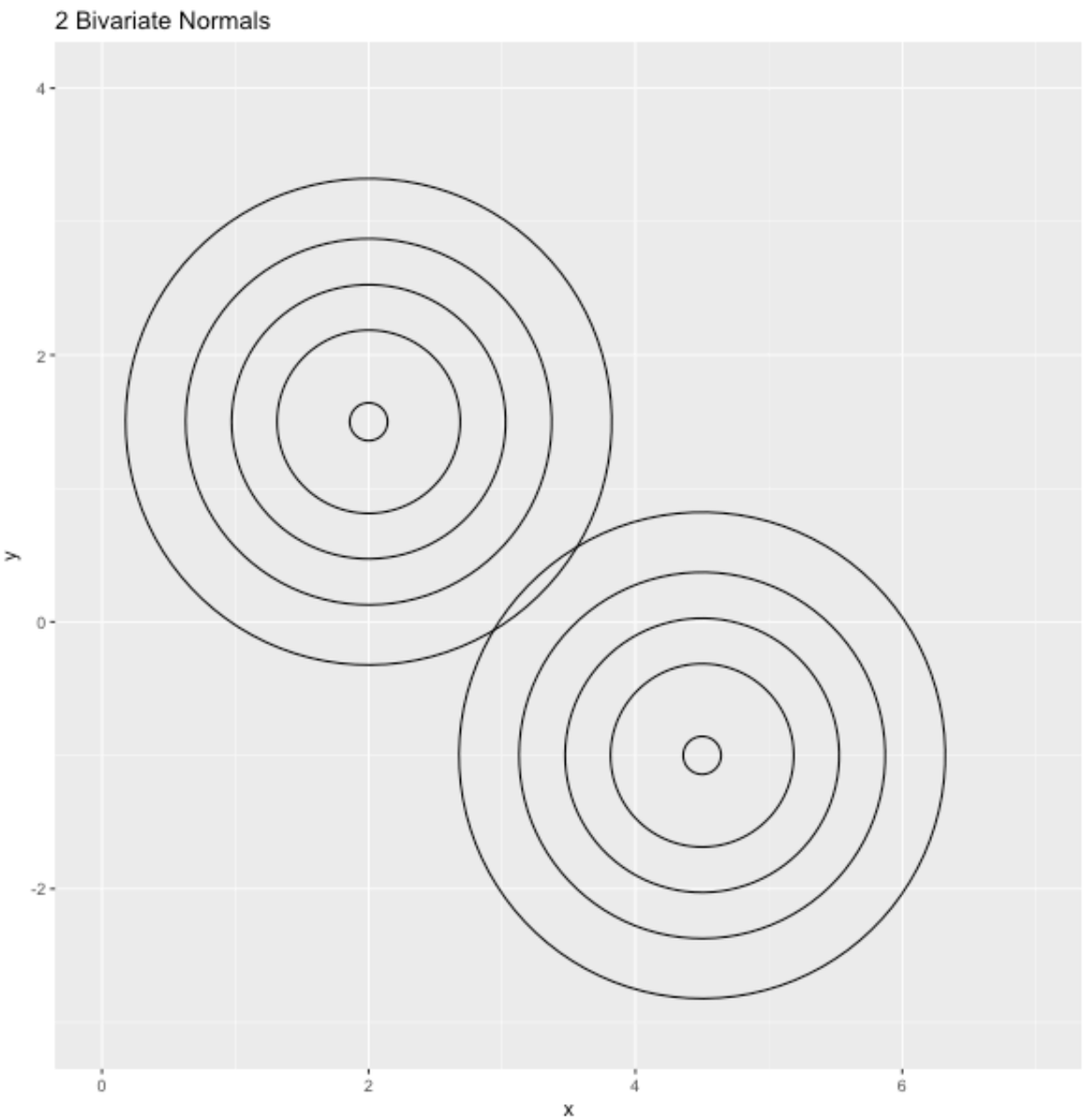
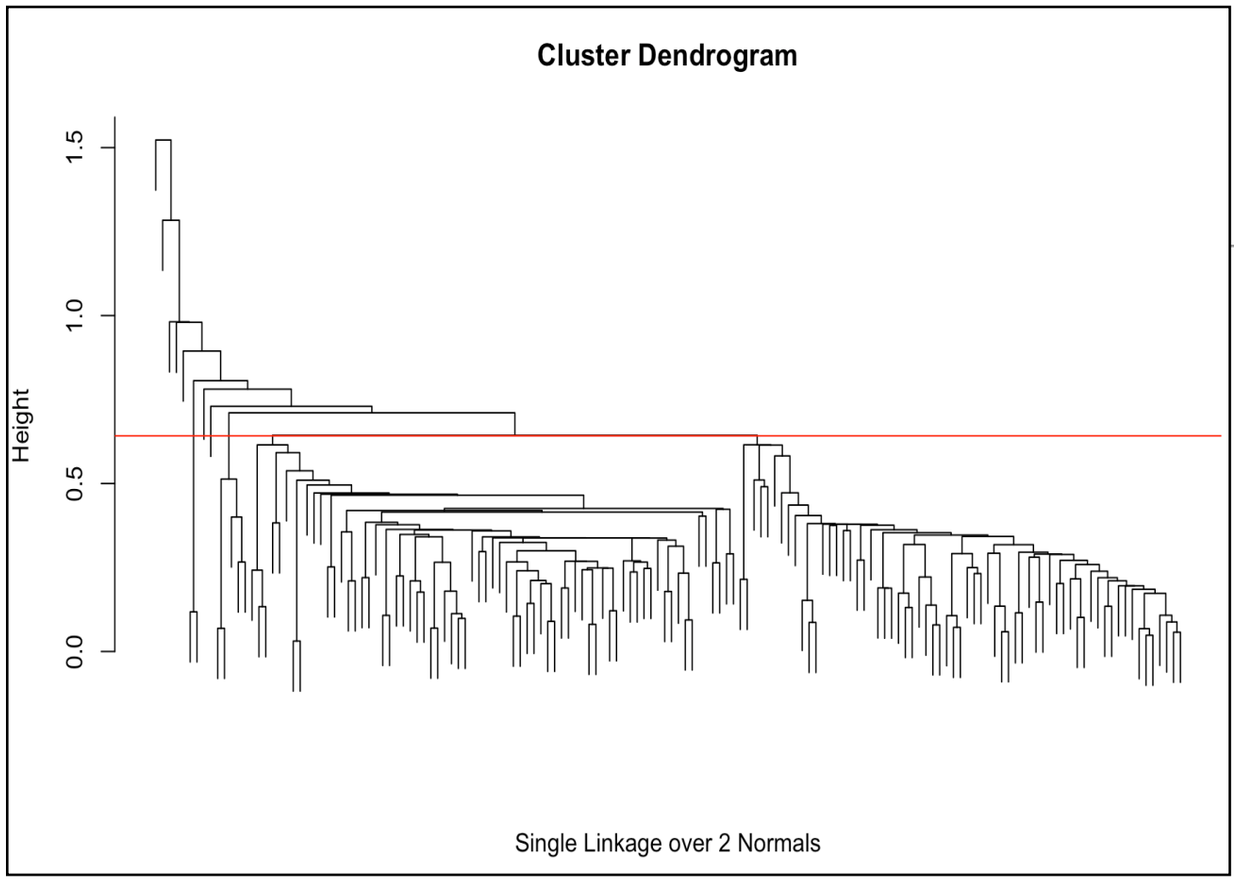
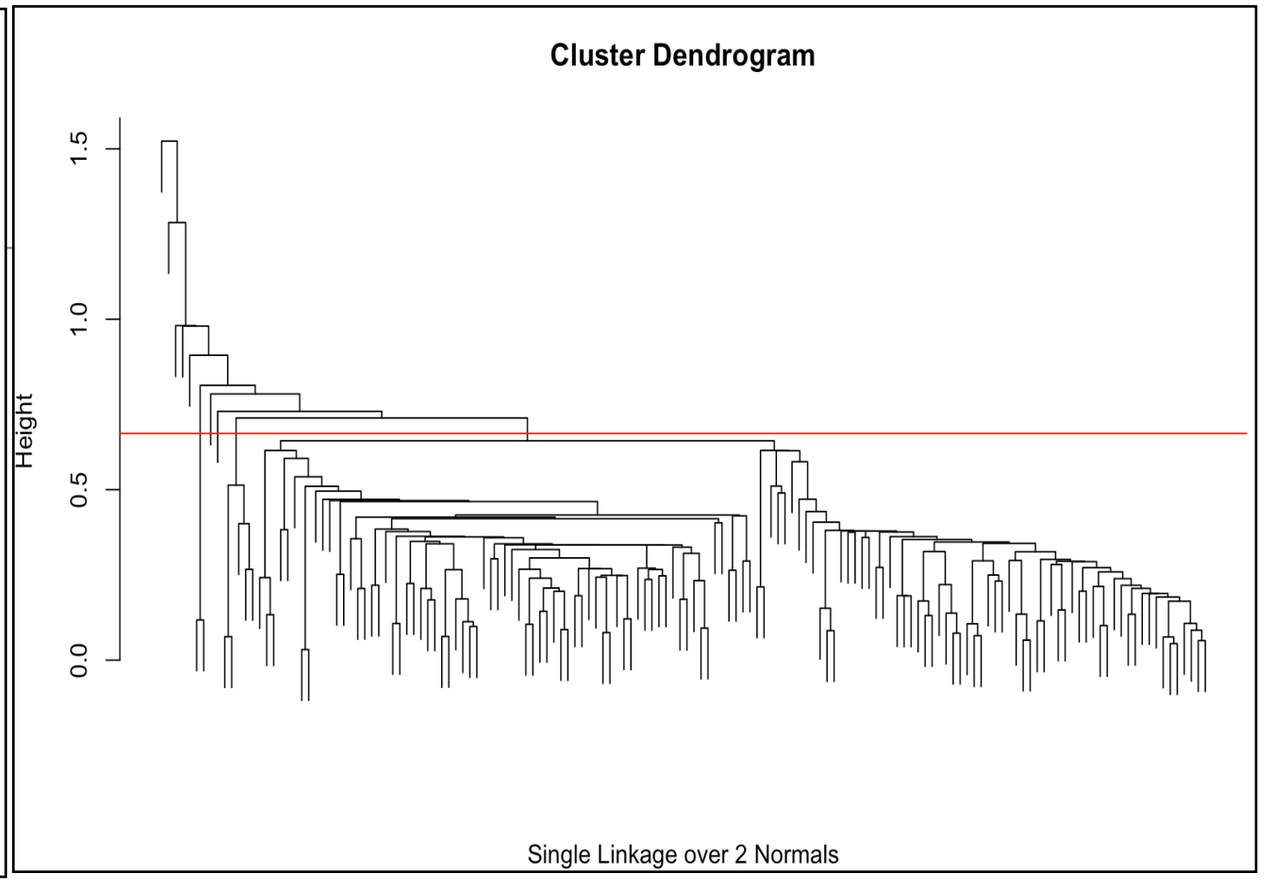
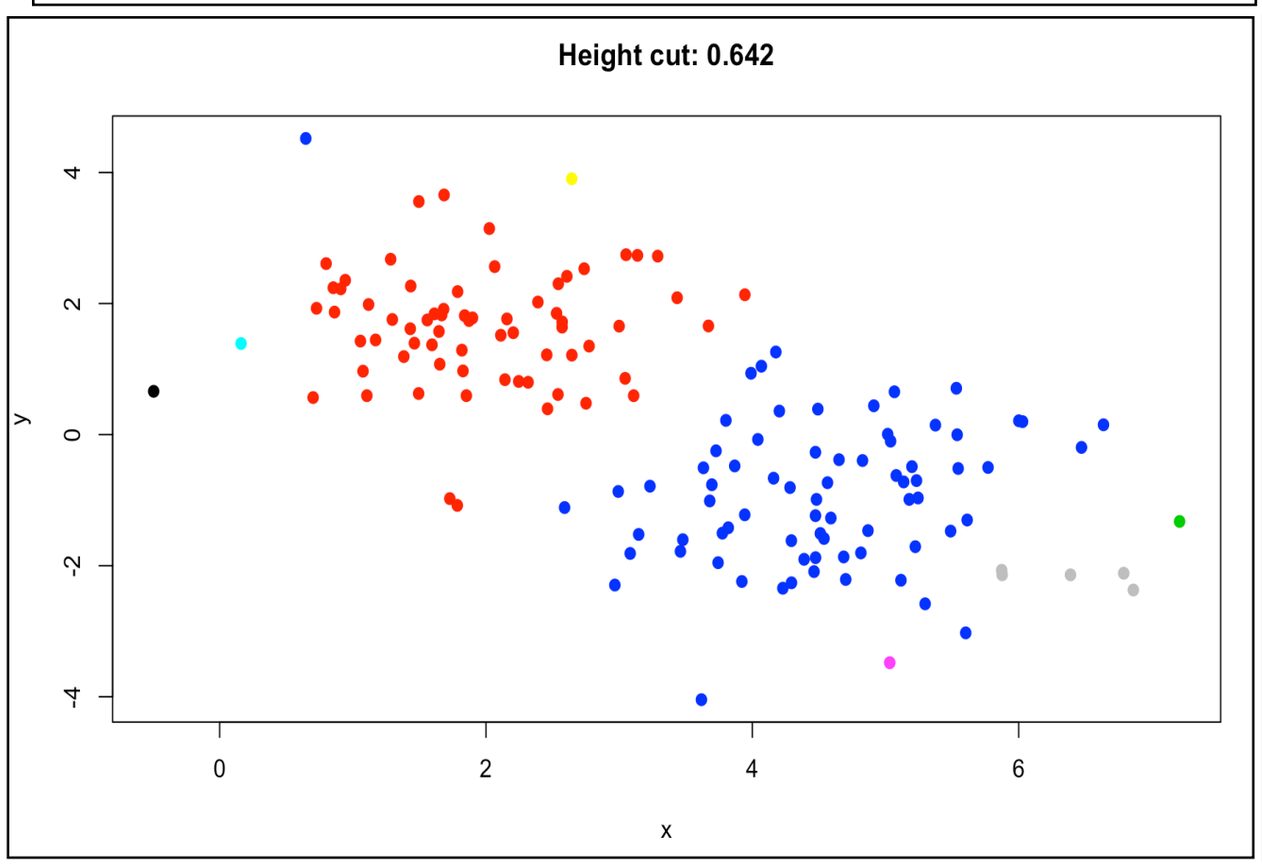
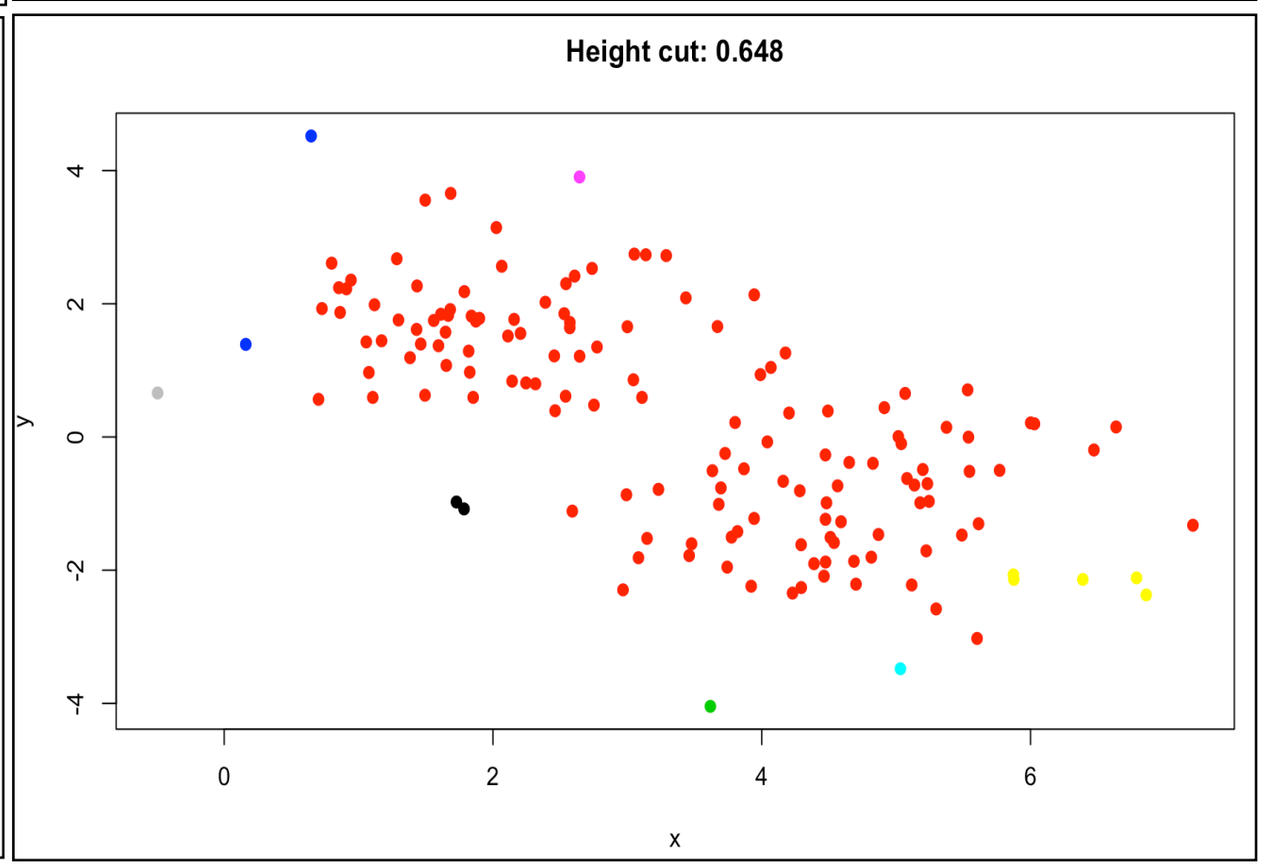




Image (left) from [Consistent]


Cluster Dendrogram
Height
1.5
1.0
0.5
0.0
Single Linkage over 2 Normals


Cluster Dendrogram
Height
1.5
1.0
0.5
0.0
Single Linkage over 2 Normals


Height cut: 0.642
y
4
2
0
-2
-4
0
2
4
6
x


Height cut: 0.648
y
4
2
0
-2
-4
0
2
4
6
x

# What does it mean to be consistent?

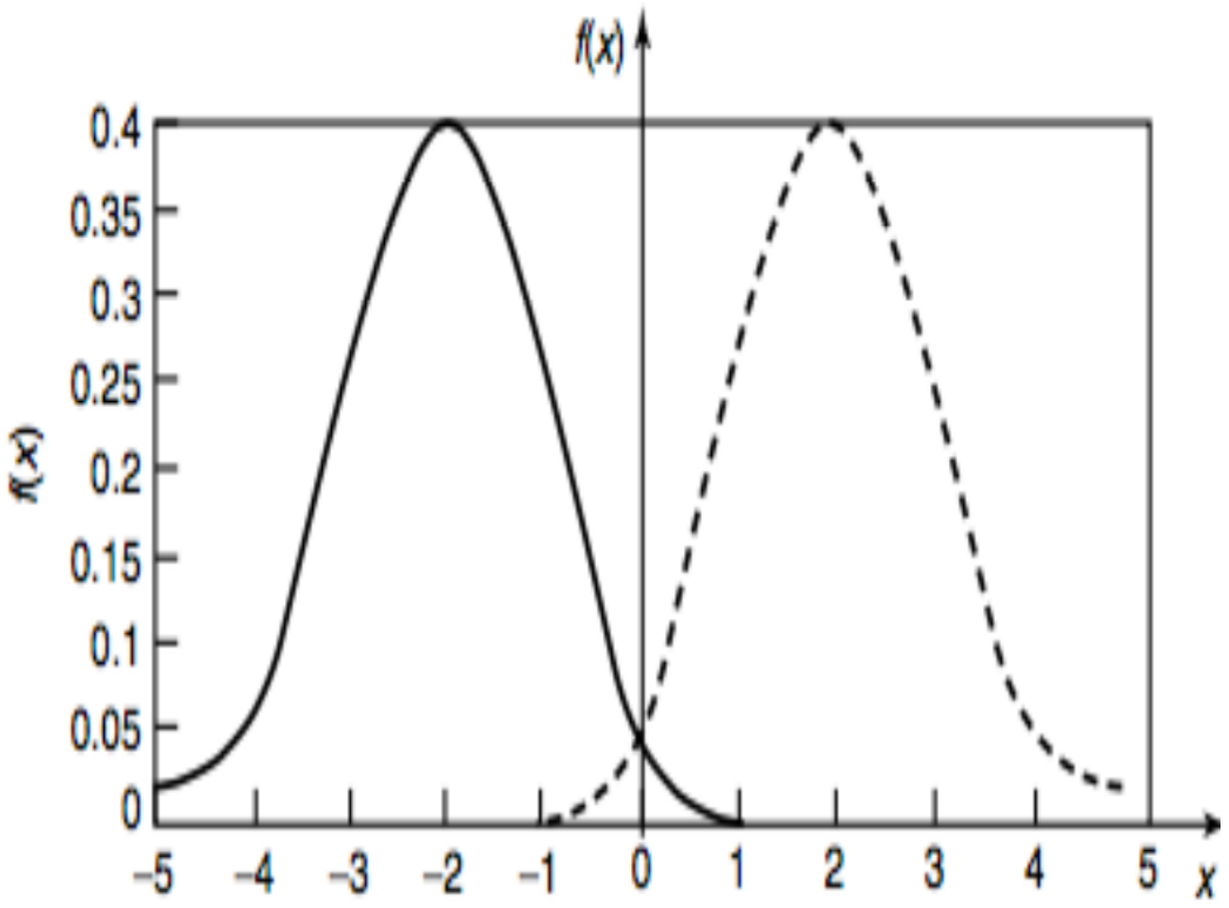



**FIGURE 11.8.** Testing between two Gaussian distributions.


1. Image (left) from Garber, Fred. “Information Theory.” Wright State University, Ohio. 2016. Lecture.

# Rates of convergence for the cluster tree

- The result was that single-linkage is inconsistent for $d > 2$
- Work picked up by SD and KC in 2010
- Resulting dendrogram/tree called “Robust Single Linkage”

This procedure for building a hierarchical clustering takes as input a data set $x_1, \ldots, x_n \in \mathbb{R}^d$.

1. For each data point $x_i$, set $r_2(x_i) =$ distance from $x_i$ to its nearest neighbor.
2. As $r$ grows from 0 to $\infty$:
   (a) Construct a graph $G_r$ with nodes $\{x_i : r_2(x_i) \leq r\}$.
   Include edge $(x_i, x_j)$ if $\|x_i - x_j\| \leq r$.
   (b) Let $\widehat{\mathbb{C}}(r)$ be the connected components of $G_r$.

Above from: Chaudhuri, K. and DasGupta, S. (2010). Rates of convergence for the cluster tree. Advances in Neural Information Processing Systems, 23, 343–351.

# Rates of convergence for the cluster tree

- The results:
  - Begs the question “What clusters can be identified from a *finite* sample?”
  - Finite sample-rate convergence are proven, achieves*:

$$k \sim d \log n$$

  - “Which we conjecture to be the best possible”  possible”[2]
  - Curse of dimensionality increases $k$ to be exponential in $d$

*1. Depends on value of $\alpha$, a variance-related parameter dependent on the data

2. From Chaudhuri, K. and DasGupta, S. (2010). Rates of convergence for the cluster tree. Advances in Neural Information Processing Systems, 23, 343–351.

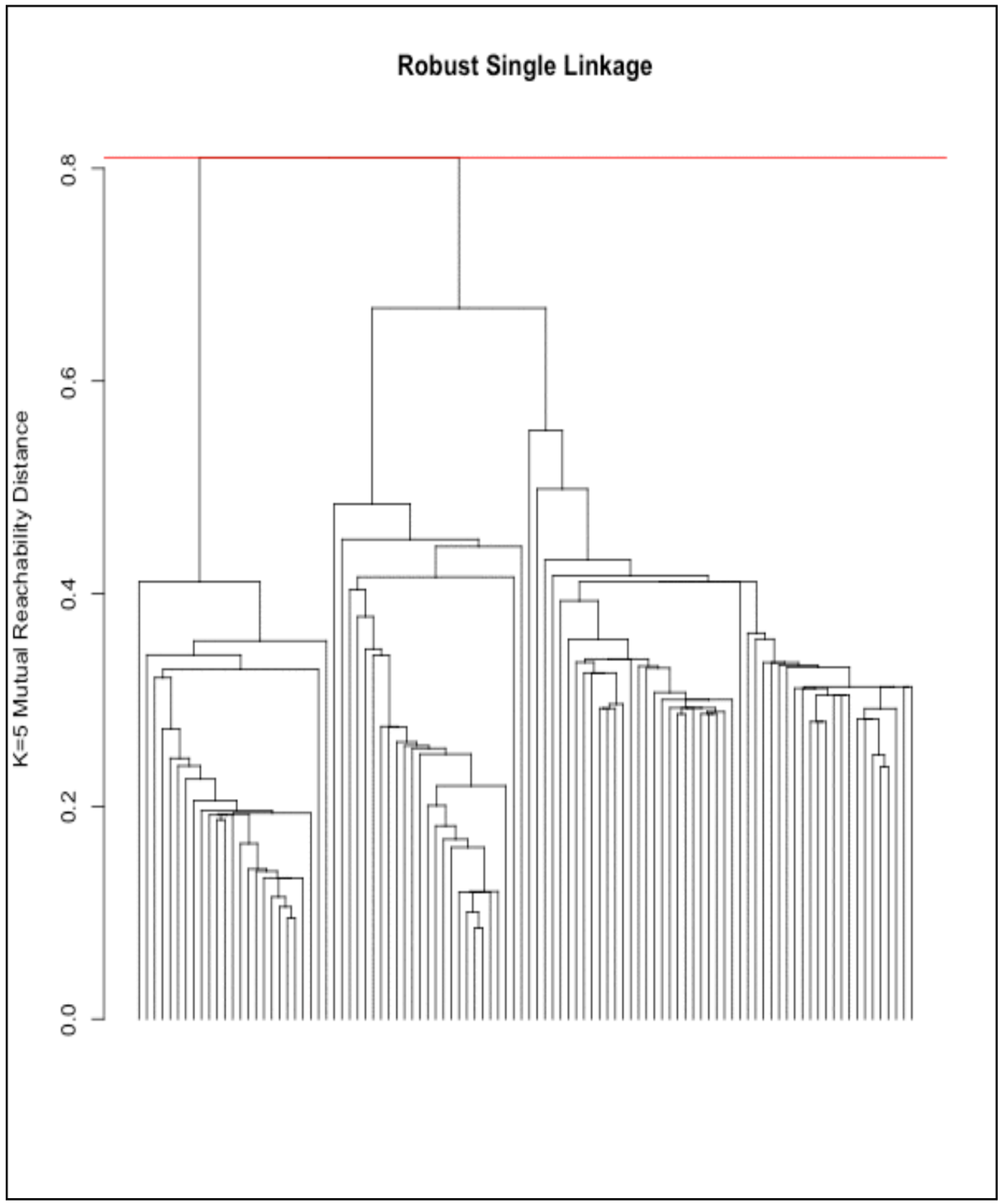
# Hierarchical DBSCAN

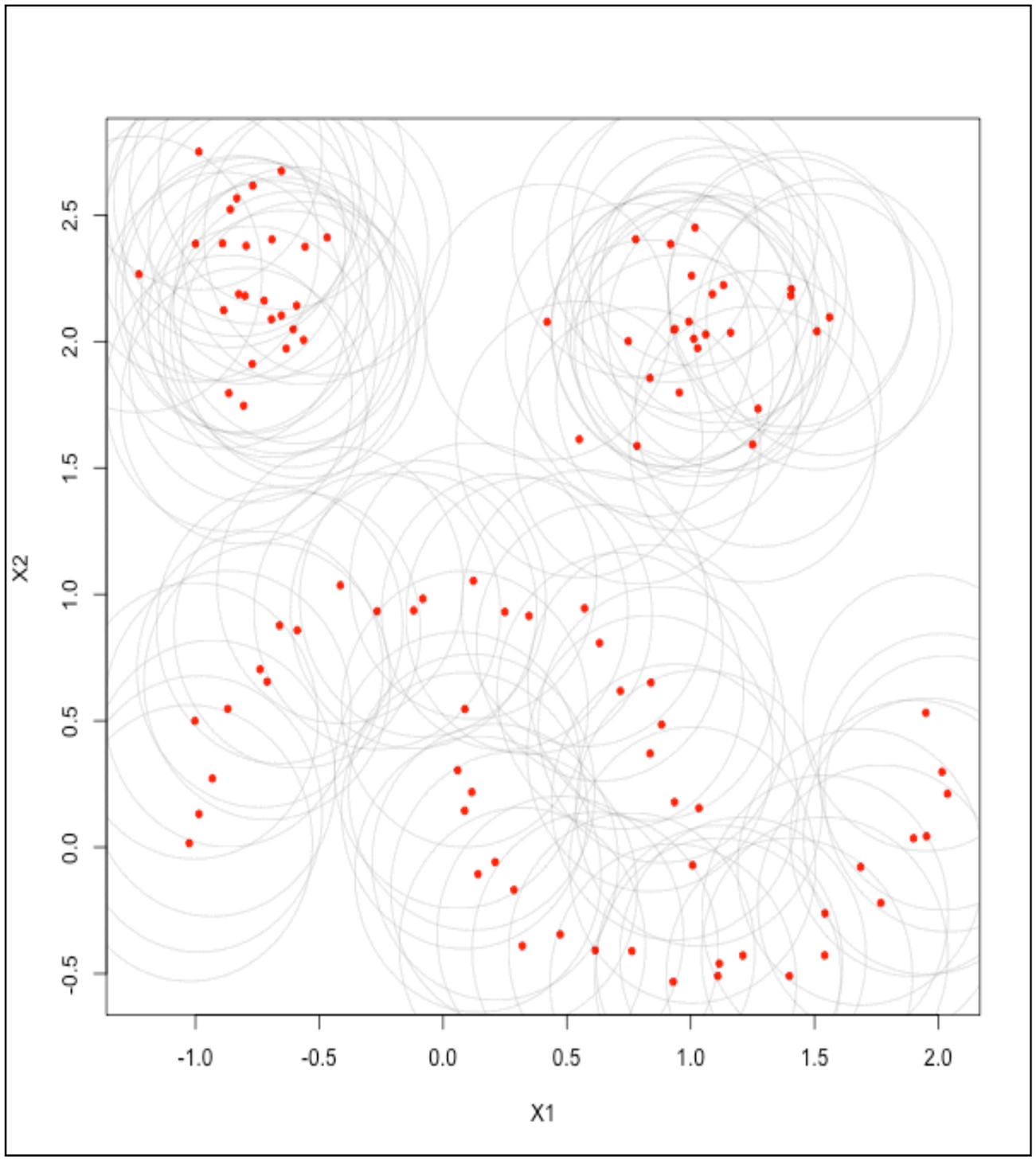
- In 2013 (and 2015) Hierarchical DBSCAN proposed
- Shown that DBSCAN corresponds *exactly* with cutting the Robust Single Linkage tree at height $\epsilon$
- Conforms to Hartigans model of the "***cluster tree***"

$$\{x : f(X) \geq \lambda\} \text{ for all } \lambda > 0$$

- 'Flat' extraction method proposed based on newest stability-based cluster validation methods

# Robust Single Linkage (left)  vs. HDBSCAN (right)



# Hierarchical DBSCAN: Stability-based Extraction

- Following Hartigan's original definition of *fractional consistency,* if a saliently-separable high-density cluster exists, then it will be **distinguished somewhere in hierarchy**
- Framework proposed by Campello, Sanders, et. all to heuristically extract **"best guess"** how to locally cut the hierarchy
- Builds off of idea of "*relative excess of mass*"
- OPTICS paper, Jain et. all from 1999(!)
  - "An important property of many real-data sets is that their **intrinsic cluster structure** cannot be characterized by global density parameters…."
  - "Very different local densities may be needed to reveal clusters in different regions of the data space"

1. Campello, Ricardo JGB, et al. "Hierarchical density estimates for data clustering, visualization, and outlier detection." ACM Transactions on Knowledge Discovery from Data (TKDD) 10.1 (2015.

# Flat cluster extraction

- Excess of Mass: $$E(\mathbf{C}_i) = \int_{x \in \mathbf{C}_i} (f(x) - \lambda_{\min}(\mathbf{C}_i))\, dx$$
- *Relative* Excess of Mass: $$E_R(\mathbf{C}_i) = \int_{x \in \mathbf{C}_i} (\lambda_{\max}(x, \mathbf{C}_i) - \lambda_{\min}(\mathbf{C}_i))\, dx$$
- Benefit: Doesn't actually require integration, nor a continuous approximation of the underlying density
- **"at least to estimate order relationships between the density at different points."**

$$S(\mathbf{C}_i) = \sum_{\mathbf{x}_j \in \mathbf{C}_i} \left(\lambda_{\max}(\mathbf{x}_j, \mathbf{C}_i) - \lambda_{\min}(\mathbf{C}_i)\right) = \sum_{\mathbf{x}_j \in \mathbf{C}_i} \left(\frac{1}{\varepsilon_{\min}(\mathbf{x}_j, \mathbf{C}_i)} - \frac{1}{\varepsilon_{\max}(\mathbf{C}_i)}\right)$$

$$\hat{S}(\mathbf{C}_i) = \begin{cases} S(\mathbf{C}_i), & \text{if } \mathbf{C}_i \text{ is a leaf node} \\ \max\{S(\mathbf{C}_i), \hat{S}(\mathbf{C}_{i_l}) + \hat{S}(\mathbf{C}_{i_r})\} & \text{if } \mathbf{C}_i \text{ is an internal node} \end{cases}$$

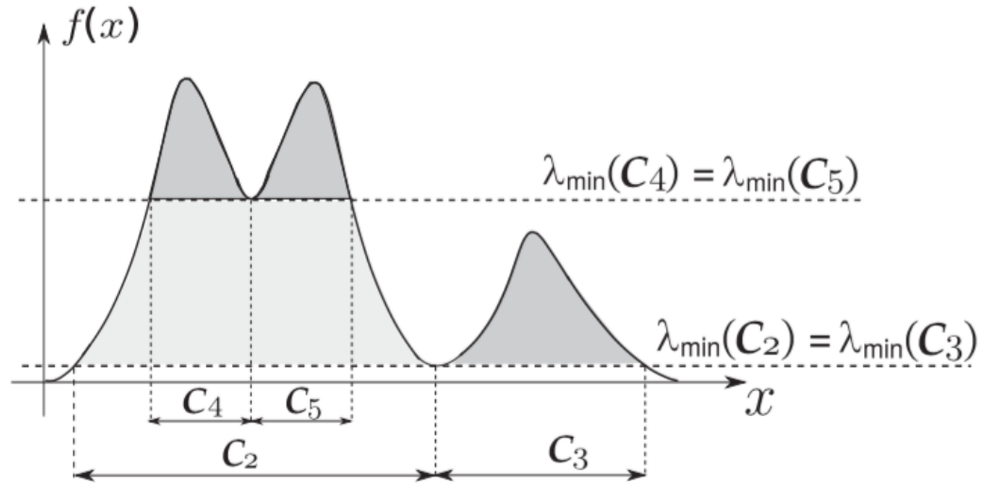
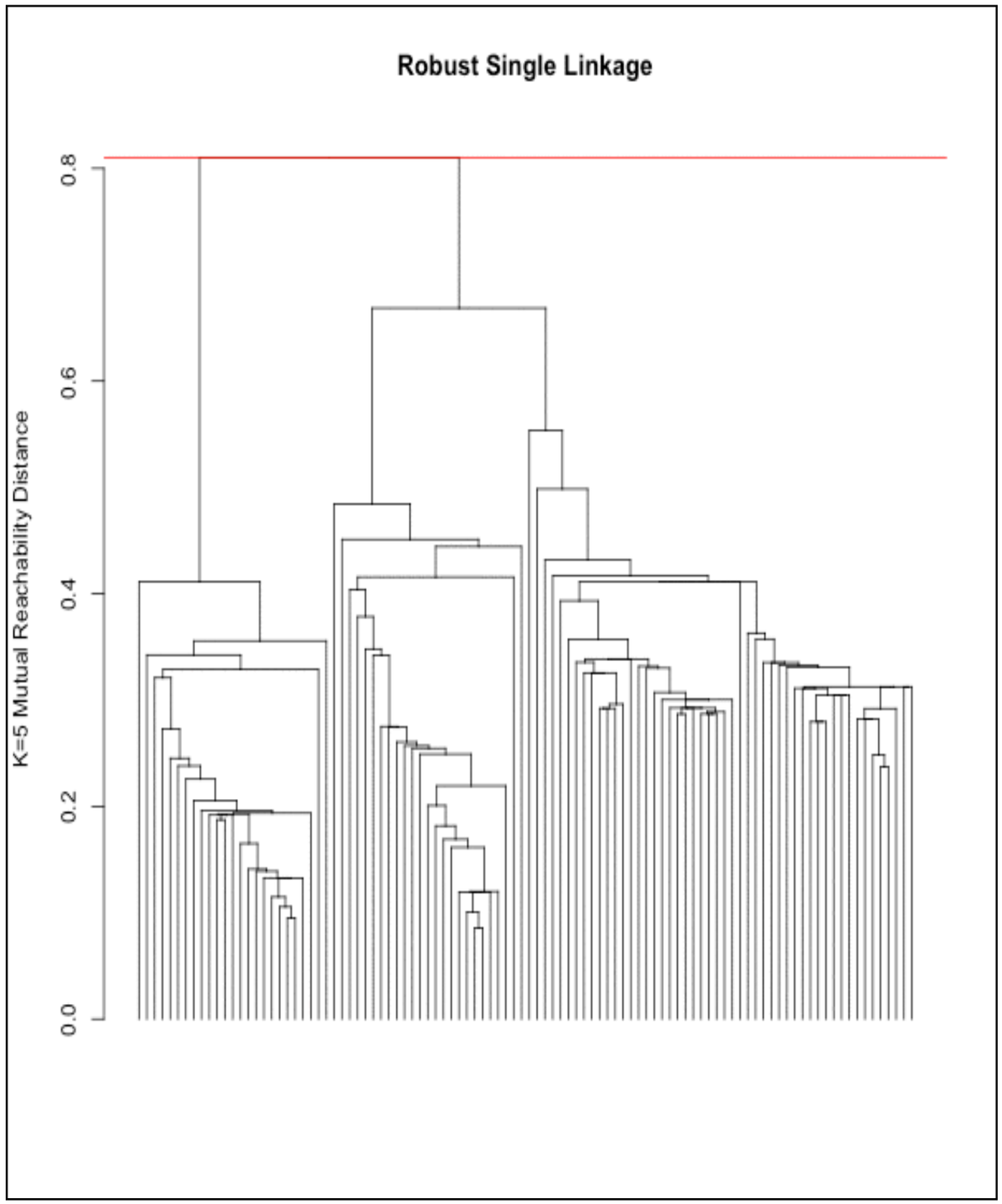
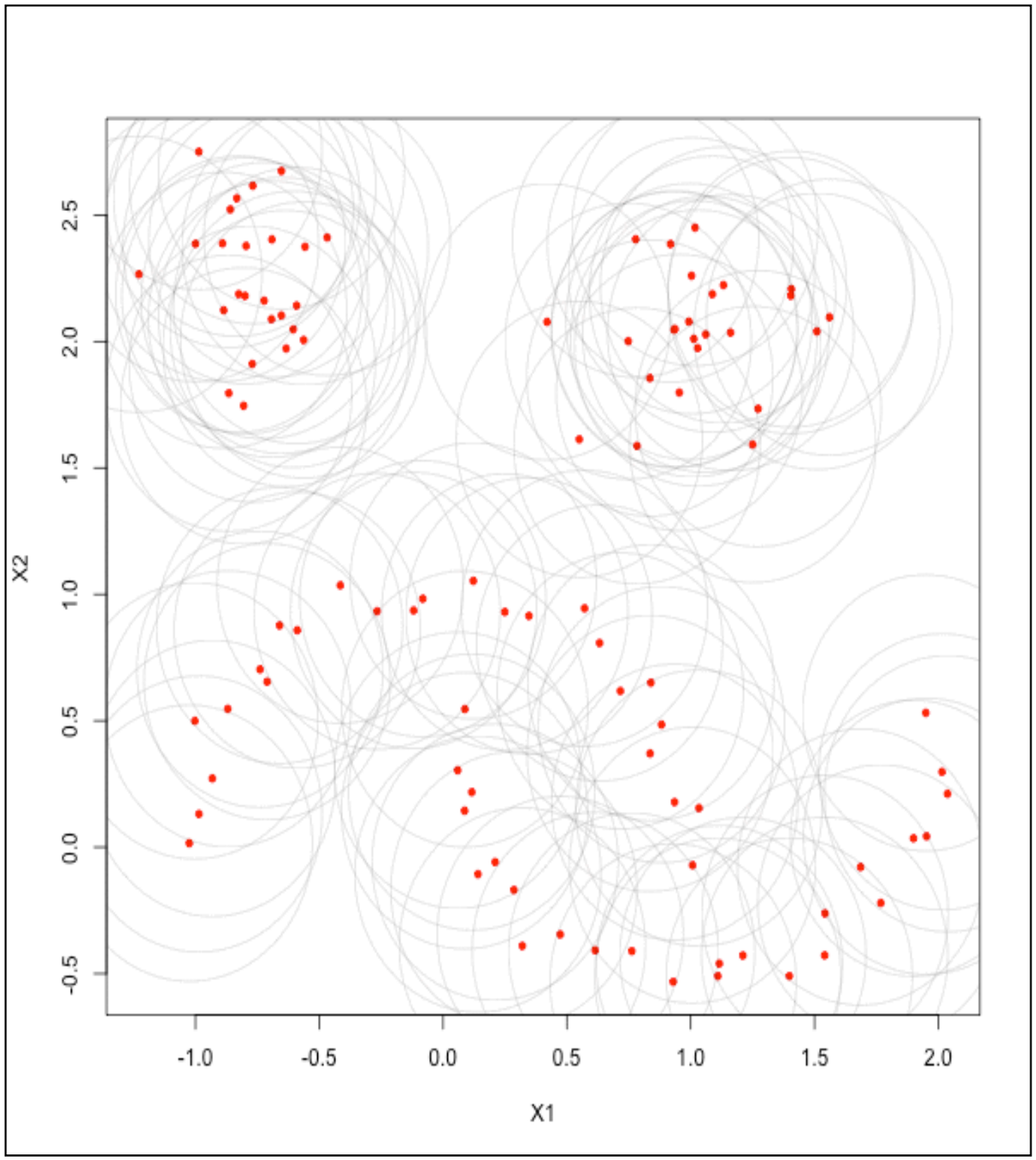


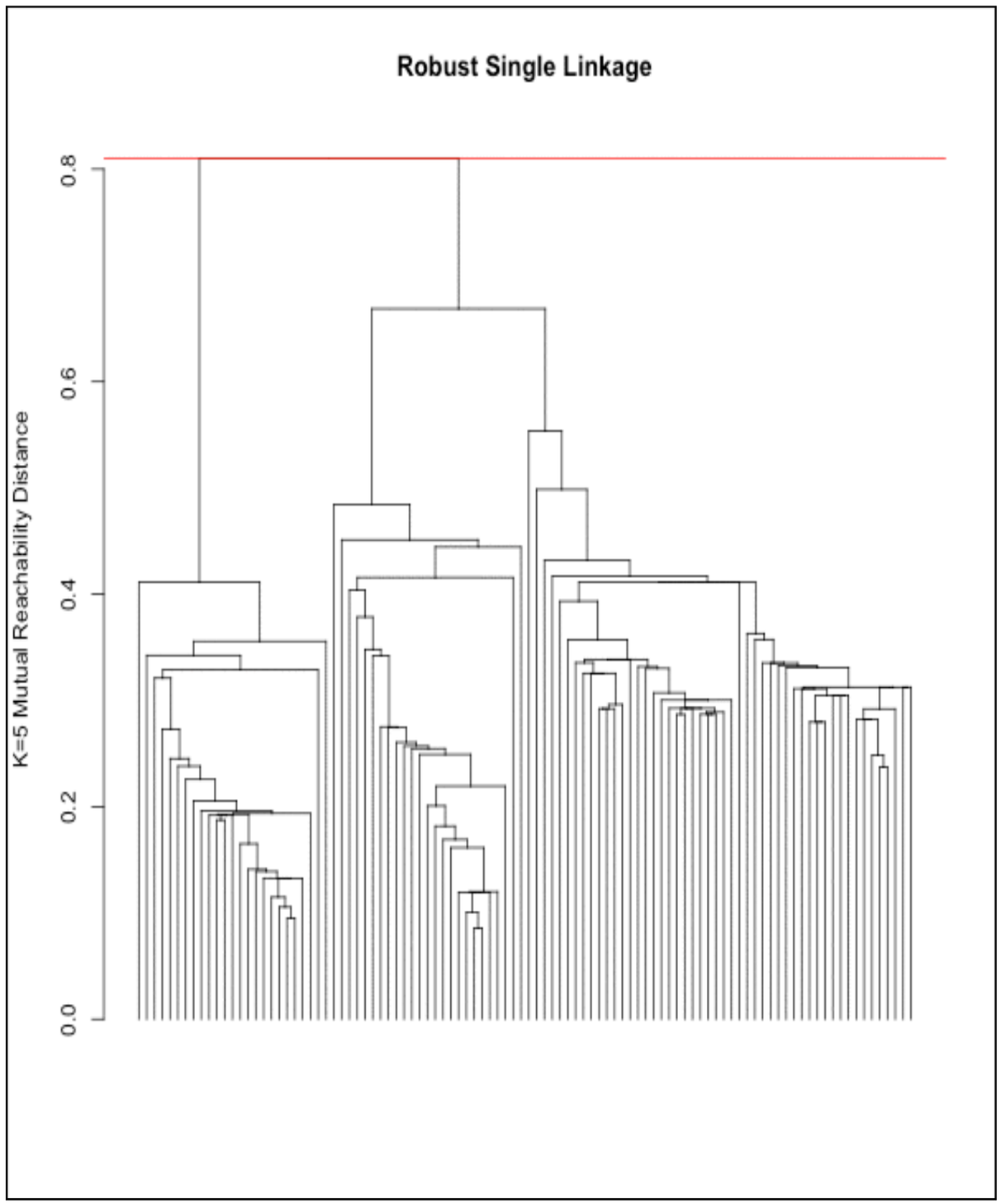

Fig. 7. Illustration of a density function, clusters, and excesses of mass.

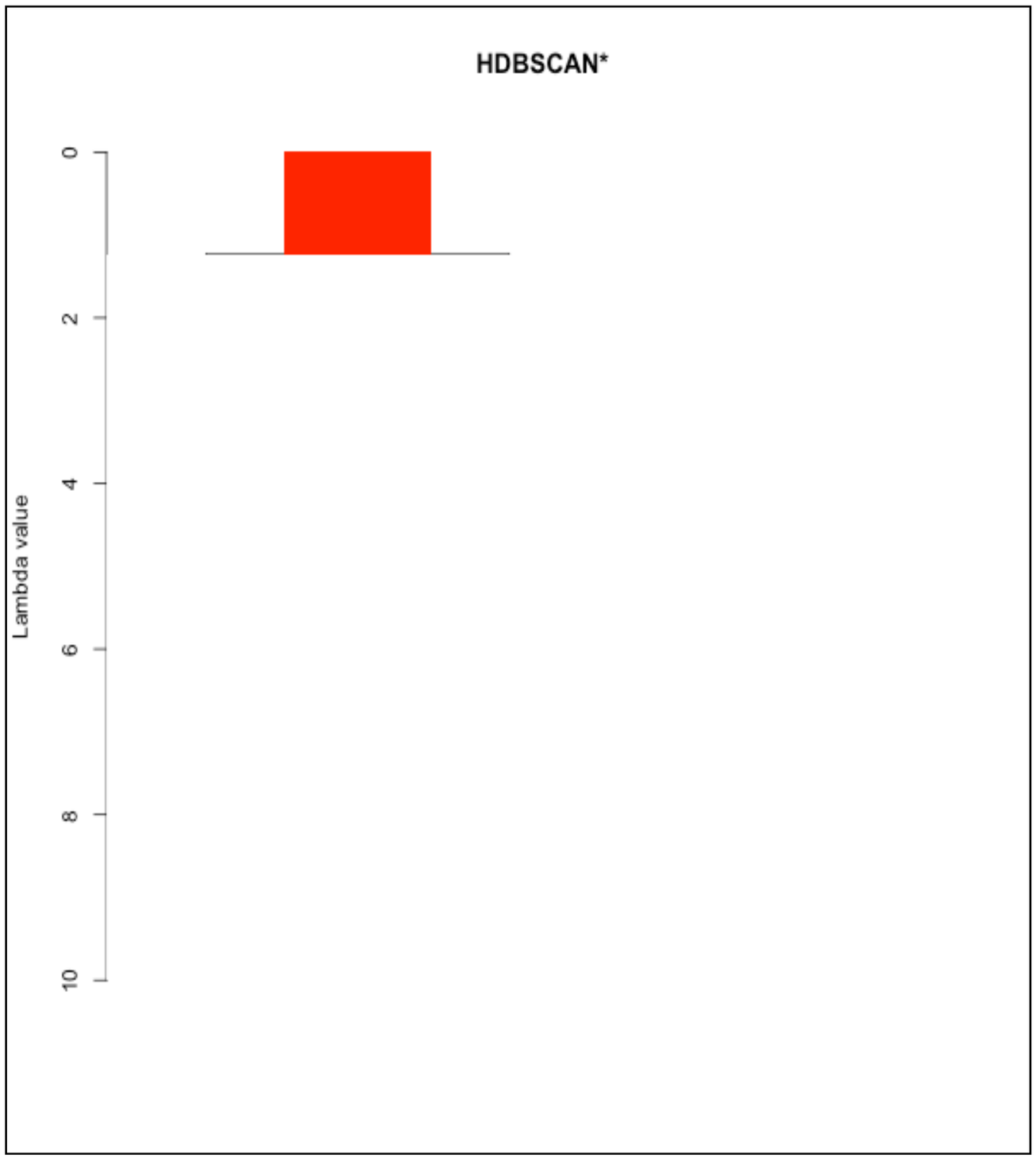
# Robust Single Linkage (left)  vs. HDBSCAN (right)

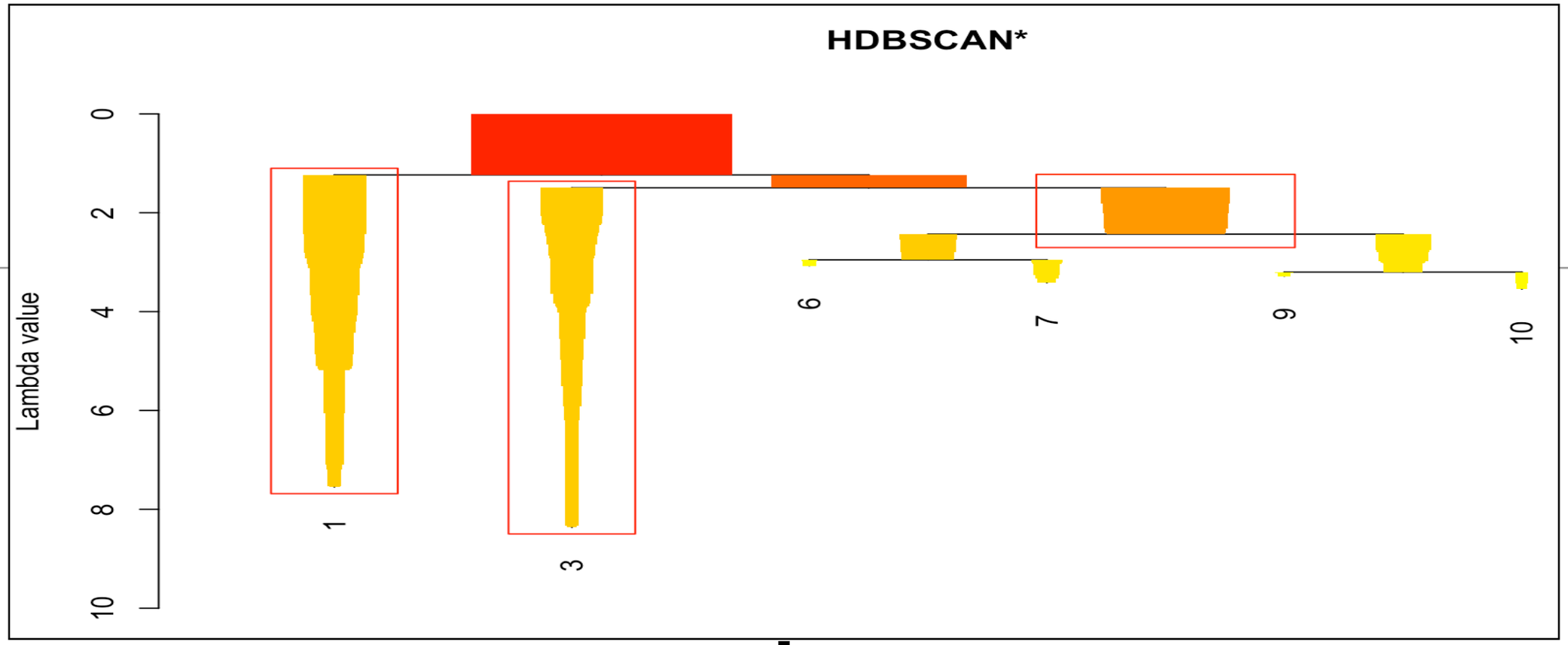


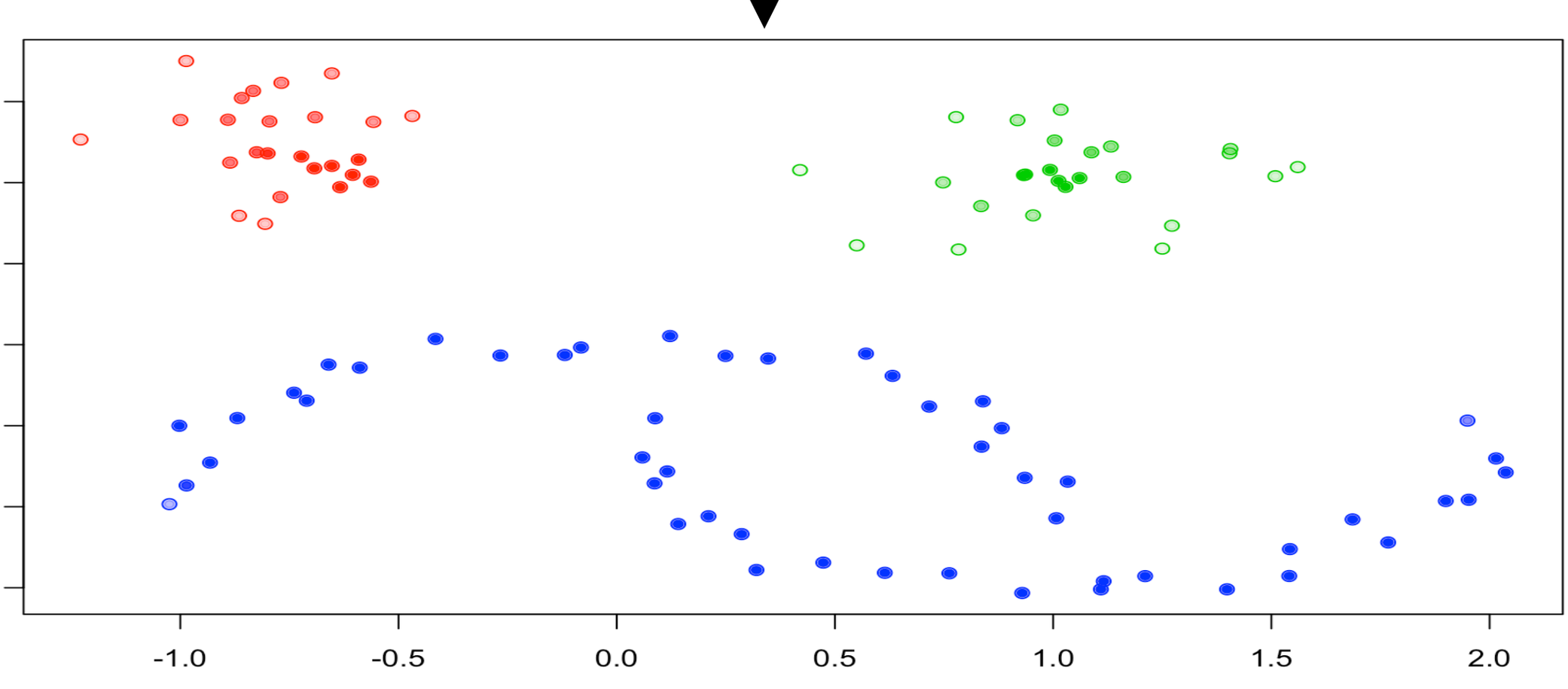



# Hierarchical DBSCAN



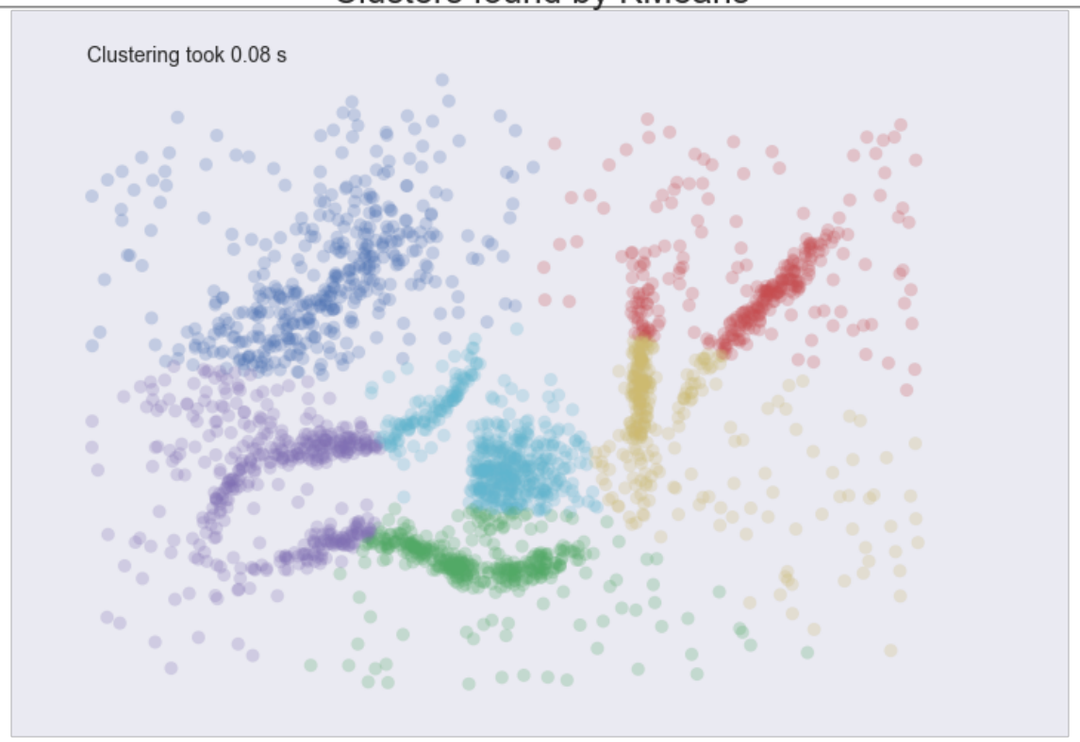

HDBSCAN*
Lambda value
0
2
4
6
8
10
1
3
6
7
9
10


2.5
2.0
1.5
1.0
0.5
0.0
-0.5
-1.0
-0.5
0.0
0.5
1.0
1.5
2.0

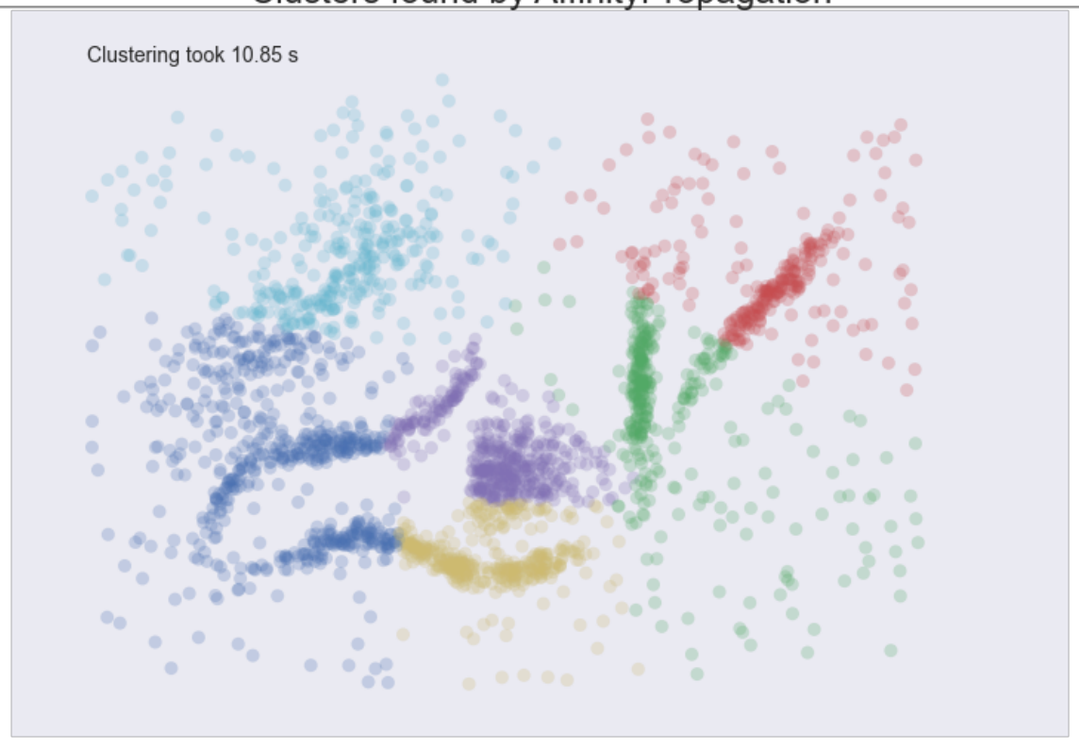
# Consider the following data set

# Cluster Examples

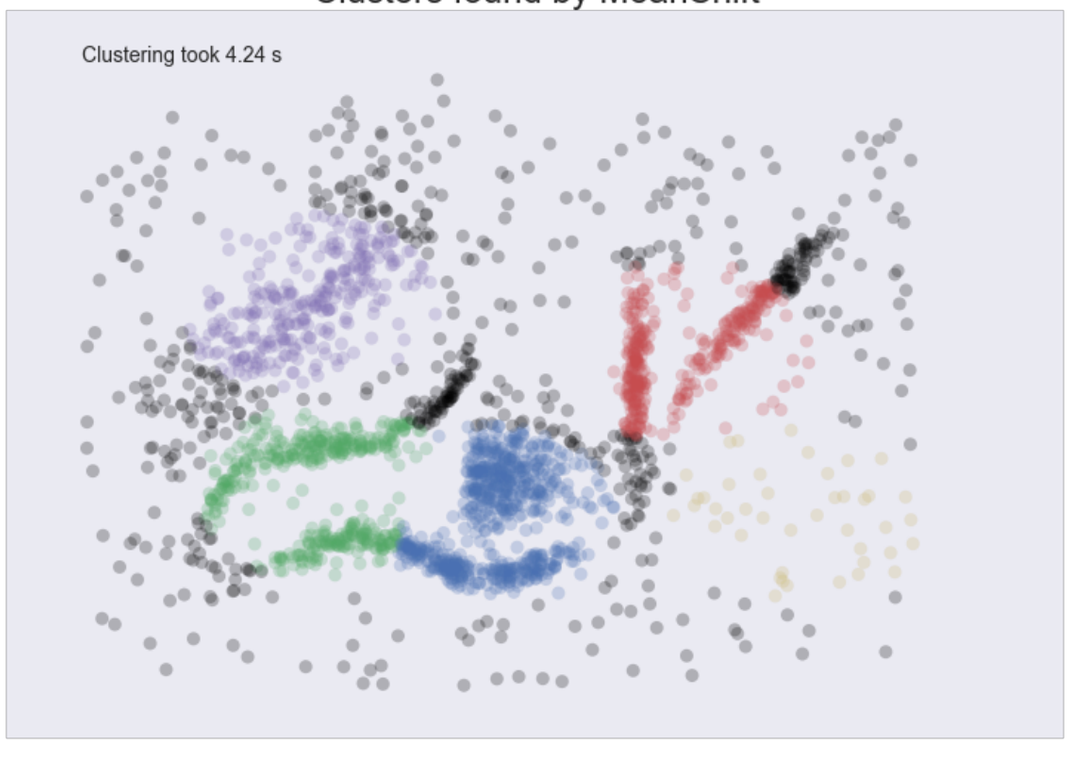
Clusters found by KMeans



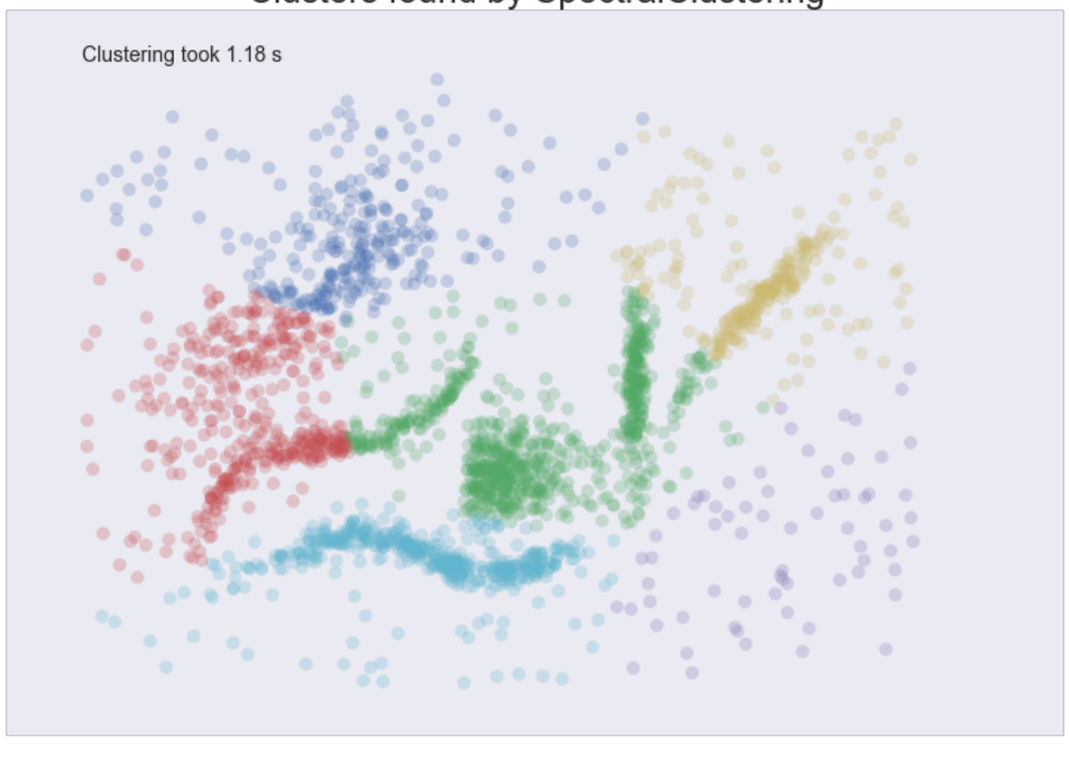

Clusters found by AffinityPropagation



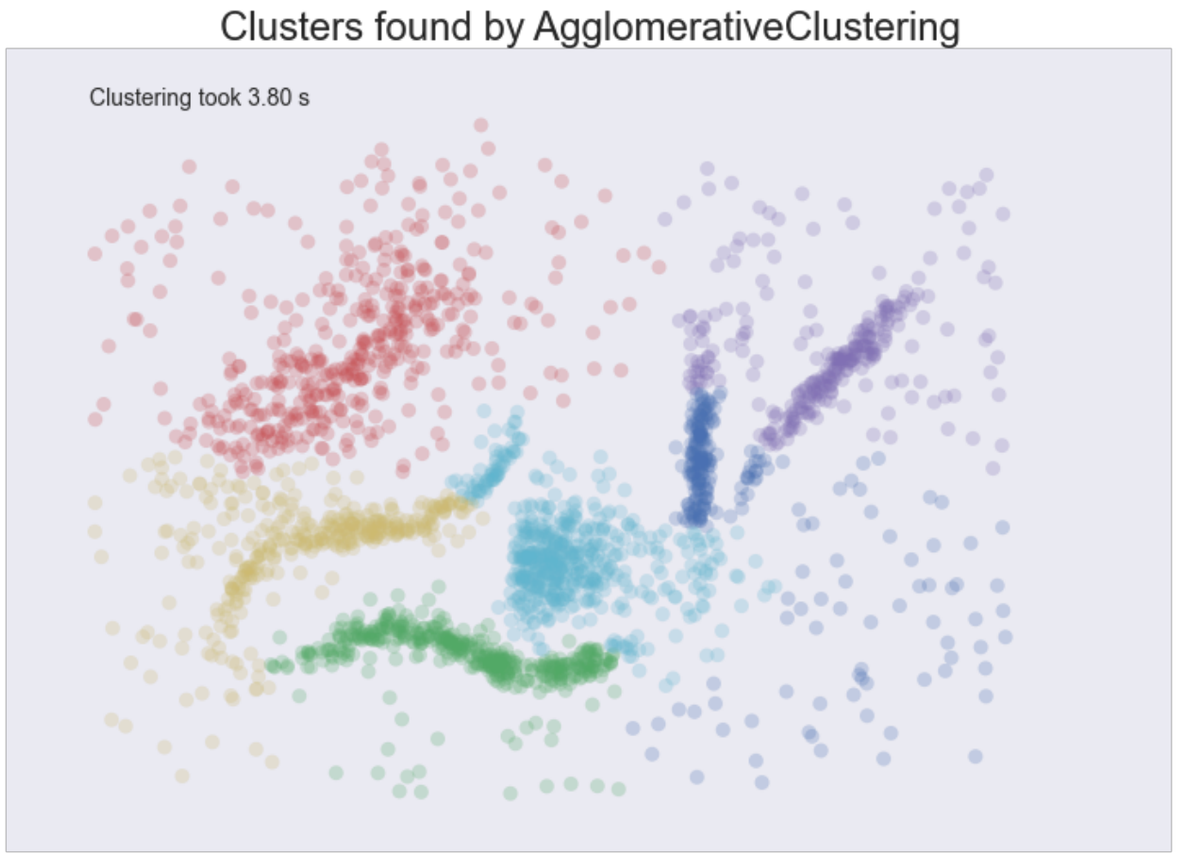

Clusters found by MeanShift

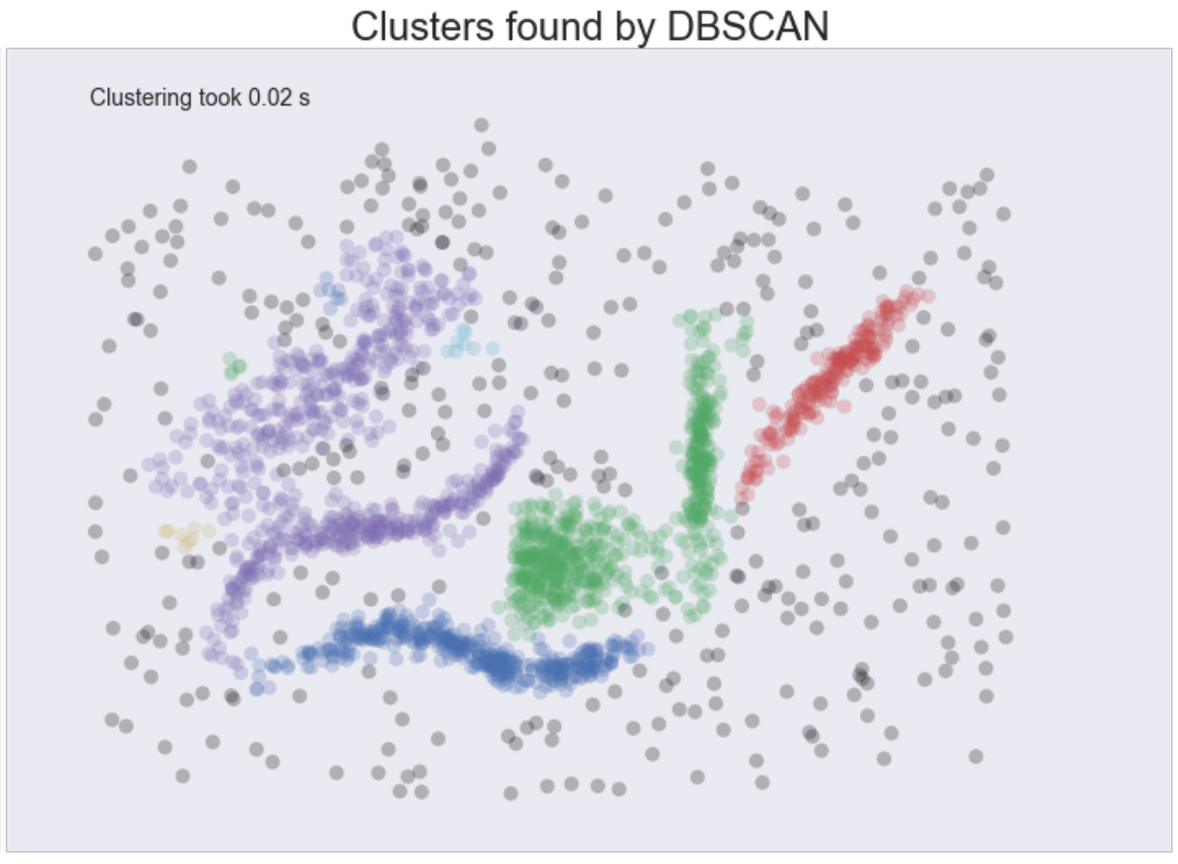



Clusters found by SpectralClustering

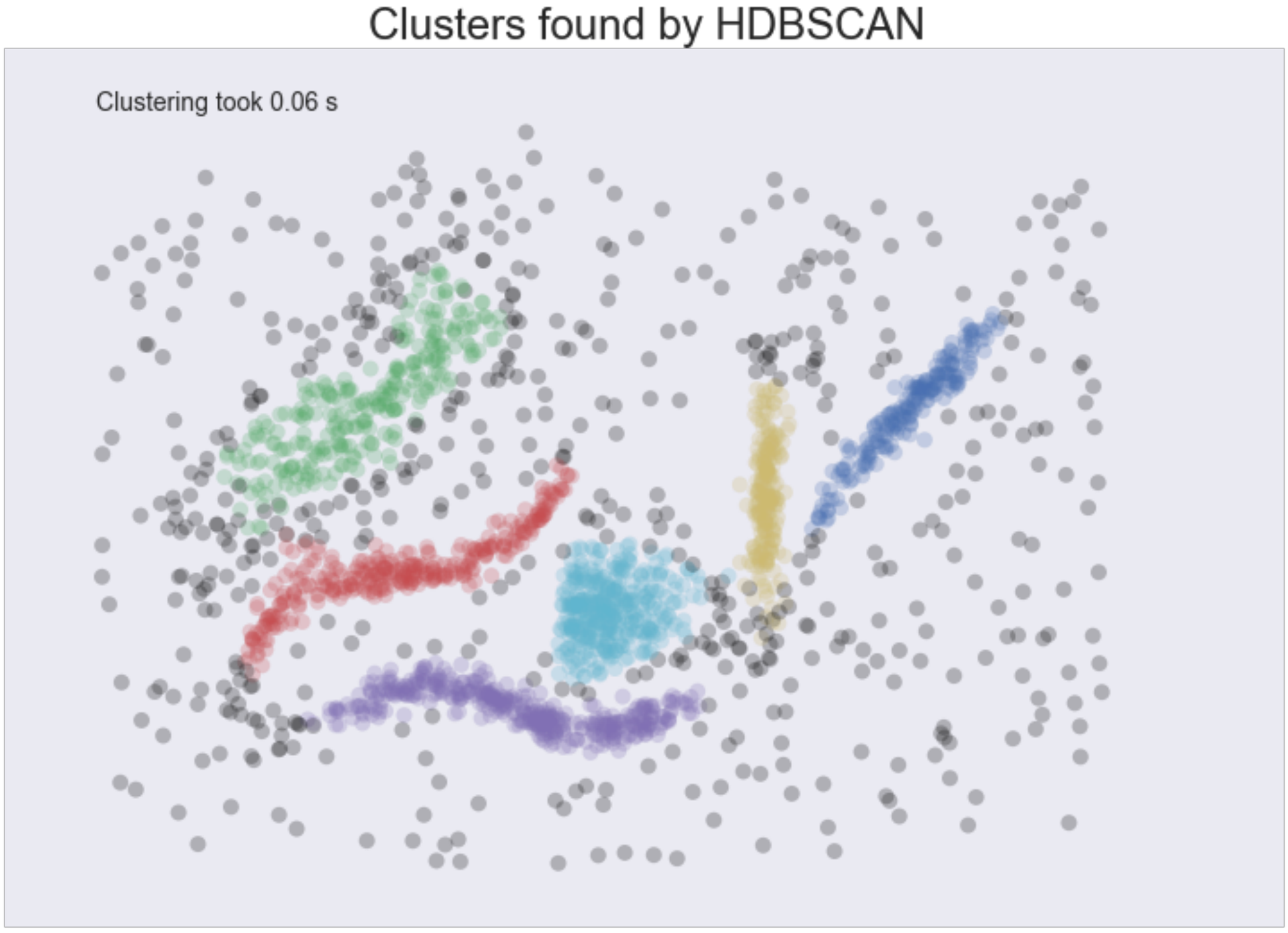


# Wards criterion and DBSCAN



# The result



# Conclusion

- Clustering can be widely useful for knowledge discovery and for learning the structure of a complex system

- Powerful discriminators may be very exciting, but are often built on ***several***, albeit perhaps reasonable assumptions

- Nonetheless, having a model that **makes minimal assumptions** about the data is useful

- Perhaps extensions that allow structured, reasonable bias to be 'injected' into the model may be worthwhile

# References


- 1] M. R. Anderb erg. Cluster analysis for applications: probability and math- ematical statistics: a series of monographs and textbooks, volume 19. Aca- demic press, 2014.
- [2] G. Andrade, G. Ramos, D. Madeira, R. Sachetto, R. Ferreira, and L. Ro cha. G-dbscan: A gpu accelerated algorithm for density-based clustering. Pro- cedia Computer Science, 18:369{378, 2013
- [3] O. Arb elaitz, I. Gurrutxaga, J. Muguerza, J. M. PeRez, and I. Perona. An extensive comparative study of cluster validity indices. Pattern Recognition, 46(1):243{256, 2013.
- [4] D. Birant and A. Kut. St-dbscan: An algorithm for clustering spatial{ temp oral data. Data & Know ledge Engineering, 60(1):208{221, 2007. [
- 5] N. S. Boutonnet, M. J. Ro oman, M.-E. Ochagavia, J. Richelle, and S. J. Wo dak. Optimal protein structure alignments by multiple linkage clus- tering: application to distantly related proteins. Protein engineering, 8(7):647{662, 1995.
- [6] S. Chakrab orty and N. K. Nagwani. Analysis and study of incremental dbscan clustering algorithm. arXiv preprint arXiv:1406.4754, 2014.
- [7] K. Chaudhuri and S. Dasgupta. Rates of convergence for the cluster tree. In Advances in Neural Information Processing Systems, pages 343{351, 2010.
- [8] K. Chaudhuri, S. Dasgupta, S. Kp otufe, and U. von Luxburg. Consistent pro cedures for cluster tree estimation and pruning. IEEE Transactions on Information Theory, 60(12):7900{7912, 2014.
- [9] Y. Chtioui, D. Bertrand, and D. Barba. Reduction of the size of the learning data in a probabilistic neural network by hierarchical clustering. applica- tion to the discrimination of seeds by articial vision. Chemometrics and Intel ligent Laboratory Systems, 35(2):175{186, 1996.
- [10] S. D. Connell and A. K. Jain. Template-based online character recognition. Pattern Recognition, 34(1):1{14, 2001.
- [11] H. Darong and W. Peng. Grid-based dbscan algorithm with referential parameters. Physics Procedia, 24:1166{1170, 2012.
- [12] J. Dieterich. A constitutive law for rate of earthquake pro duction and its application to earthquake clustering. Journal of Geophysical Research: Solid Earth, 99(B2):2601{2618, 1994.

# References


- [13] L. Engelman and J. A. Hartigan. Percentage p oints of a test for clusters. Journal of the American Statistical Association, 64(328): 1647{1648, 1969.
- [14] G. Esfandani and H. Ab olhassani. Msdbscan: multi-density scale- indep endent clustering algorithm based on dbscan. In International Confer- ence on Advanced Data Mining and Applications, pages 202{213. Springer, 2010.
- [15] R. A. Fisher. The use of multiple measurements in taxonomic problems. Annals of eugenics, 7(2):179{188, 1936.
- [16] W. B. Frakes and R. Baeza-Yates. Information retrieval: data structures and algorithms. 1992
- [17] M. N. Gaonkar and K. Sawant. Auto epsdbscan: Dbscan with eps automatic for large dataset. International Journal on Advanced Computer Theory and Engineering, 2(2):11{16, 2013.
- [18] I. Gath and D. Ho ory. Detection of elliptic shells using fuzzy clustering: application to mri images. In Pattern Recognition, 1994. Vol. 2-Conference B: Computer Vision & Image Processing., Proceedings of the 12th IAPR International. Conference on, volume 2, pages 251{255. IEEE, 1994.
- [19] I. Guyon, U. Von Luxburg, and R. C. Williamson. Clustering: Science or art. In NIPS 2009 workshop on clustering theory, pages 1{11, 2009.
- [20] J. Han, K. Kop erski, and N. Stefanovic. Geominer: a system prototyp e for spatial data mining. In AcM sIGMoD Record, volume 26, pages 553{556. ACM, 1997.
- [21] K. R. Harrigan. An application of clustering for strategic group analysis. Strategic Management Journal, 6(1):55{73, 1985.
- [22] J. Hartigan. Estimation of a convex density contour in two dimensions. Journal of the American Statistical Association, 82(397): 267{270, 1987.
- [23] J. A. Hartigan. Direct clustering of a data matrix. Journal of the american statistical association, 67(337):123{129, 1972.
- [24] J. A. Hartigan. Consistency of single linkage for high-density clusters. Journal of the American Statistical Association, 76(374): 388{394, 1981.

# References


- [25] J. A. Hartigan. Statistical theory in clustering. Journal of classication, 2(1):63{76, 1985.
- [26] J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means cluster- ing algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1):100{108, 1979.
- [27] Y. He, H. Tan, W. Luo, H. Mao, D. Ma, S. Feng, and J. Fan. Mr-dbscan: an ecient parallel density-based clustering algorithm using mapreduce. In Paral lel and Distributed Systems (ICPADS), 2011 IEEE 17th International Conference on, pages 473{480. IEEE, 2011.
- [28] E. C. Kenley and Y.-R. Cho. Entropy-based graph clustering: Application to biological and so cial networks. In Data Mining (ICDM), 2011 IEEE 11th International Conference on, pages 1116{1121. IEEE, 2011.
- [29] S. Kisilevich, F. Mansmann, and D. Keim. P-dbscan: a density based clustering algorithm for exploration and analysis of attractive areas using collections of geo-tagged photos. In Proceedings of the 1st international conference and exhibition on computing for geospatial research & applica- tion, page 38. ACM, 2010
- [30] H.-P. Kriegel, E. Schub ert, and A. Zimek. The (black) art of runtime evaluation: Are we comparing algorithms or implementations? Know ledge and Information Systems, pages 1{38, 2016.
- [31] M. Kryszkiewicz and P. Lasek. Ti-dbscan: Clustering with dbscan by means of the triangle inequality. In International Conference on Rough Sets and Current Trends in Computing, pages 60{69. Springer, 2010.
- [32] J. Liu and T. Liu. Coarse-grained diffusion distance for community struc- ture detection in complex networks. Journal of Statistical Mechanics: The- ory and Experiment, 2010(12):P12030, 2010

# References


- 33] P. Liu, D. Zhou, and N. Wu. Vdbscan: varied density based spatial cluster- ing of applications with noise. In Service Systems and Service Management, 2007 International Conference on, pages 1{4. IEEE, 2007.
- [34] M. A. Patwary, D. Palsetia, A. Agrawal, W.-k. Liao, F. Manne, and A. Choudhary. A new scalable parallel dbscan algorithm using the disjoint- set data structure. In Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, page 62. IEEE Computer So ciety Press, 2012.
- [35] D. Pfeier, T. P. Robinson, M. Stevenson, K. B. Stevens, D. J. Rogers, A. C. Clements, et al. Spatial analysis in epidemiology, volume 557976890. Oxford University Press New York, 2008.
- [36] A. Smiti and Z. Eloudi. Soft dbscan: Improving dbscan clustering metho d using fuzzy set theory. In Human System Interaction (HSI), 2013 The 6th International Conference on, pages 380{385. IEEE, 2013.
- [37] A. Tramacere and C. Vecchio.
- -ray dbscan: a clustering algorithm applied to fermi-lat
- -ray data-i. detection p erformances with real and simulated data. Astronomy & Astrophysics, 549:A138, 2013.
- [38] T. N. Tran, K. Drab, and M. Daszykowski. Revised dbscan algorithm to cluster data with dense adjacent clusters. Chemometrics and Intel ligent Laboratory Systems, 120:92{96, 2013.
- [39] P. Viswanath and R. Pinkesh. l-dbscan: A fast hybrid density based cluster- ing metho d. In Pattern Recognition, 2006. ICPR 2006. 18th International Conference on, volume 1, pages 912{915. IEEE, 2006.
- [40] K. L. Wagstaff, Intel ligent clustering with instance-level constraints. PhD thesis, Cornell University, 2002

# References


- [41] J. H. Ward Jr. Hierarchical grouping to optimize an ob jective function. Journal of the American statistical association, 58(301): 236{244, 1963.

- [42] C. Xiaoyun, M. Yufang, Z. Yan, and W. Ping. Gmdbscan: multi-density dbscan cluster based on grid. In e-Business Engineering, 2008. ICEBE'08. IEEE International Conference on, pages 780{783. IEEE, 2008

- [43] R. B. Zadeh. Towards a principled theory of clustering. In Thirteenth International Conference on Articial Intel ligence and Statistics, 2010.

- [44] H. Zhou, P. Wang, and H. Li. Research on adaptive parameters determina- tion in dbscan algorithm. Journal of information & computational science, 9(7):1967{1973, 2012