COMPSCI762: Introduction to Machine Learning

Clustering

Jörg Simon Wicker and Katerina Taškova

The University of Auckland

SCIENCE
SCHOOL OF COMPUTER SCIENCE

Clustering (cont.)

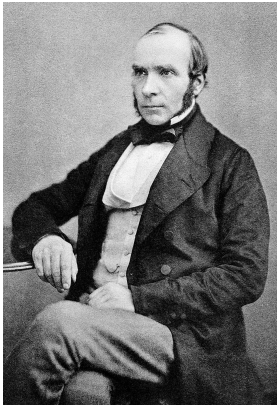
- Density-Based Clustering
- Hierarchical Clustering
  - Agglomerative Clustering
- Cluster Quality

Summary

Partly based on the lecture slides from University of British Columbia CPSC340

Density-Based Clustering

Motivation: Cholera Outbreak

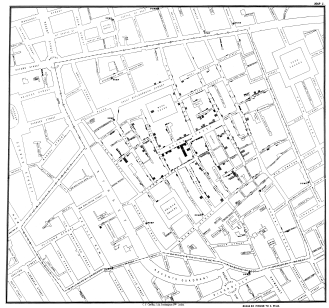


John Snow

en.wikipedia.org/wiki/John_Snow

Motivation: Cholera Outbreak

- John Snow's 1854 spatial histogram of deaths from cholera in Soho, London, UK



en.wikipedia.org/wiki/1854_Broad_Street_cholera_outbreak

- Found cluster of cholera deaths around a particular water pump
  - Went against airborne theory, but pump later found to be contaminated
  - "Father" of epidemiology

Density-Based Clustering

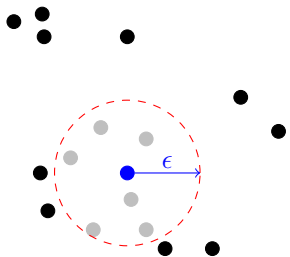
- Density-based clustering
  - Clusters are defined by "dense" regions
  - Instances in non-dense regions don't get clustered
    - Not trying to "partition" the space
- Clusters can be non-convex
  - Elephant clusters affected by vegetation, mountains, rivers, water access, etc
- It's (almost) non-parametric clustering method
  - No fixed number of clusters k
  - Clusters can become more complicated with more data

Other Potential Applications

- Where are high crime regions of a city?
- Where should taxis patrol?
- Which products are similar to this one?
- Which pictures are in the same place?
- Where can proteins 'dock'?
- Where are people tweeting?

Density-Based Clustering

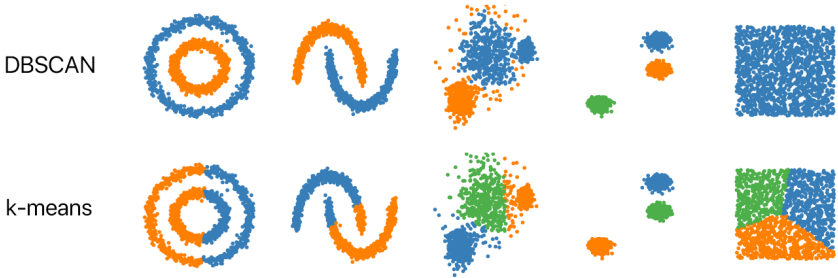
- Density-Based Spatial Clustering with applications with Noise algorithm (DBSCAN) has two hyperparameters
  - ϵ: distance we use to decide if another point is a "neighbour"
  - MinNeighbours: number of neighbours needed to say a region is "dense"
    - If you have at least *minNeighbours* neighbours, you are called a core point



If *minNeighbours* ≤ 6, then • is a core point, since there are 6 • neighbours

- Main idea: merge all neighbouring core points to form clusters, i.e implements a "chain reaction" throughout the dense areas

DBSCAN

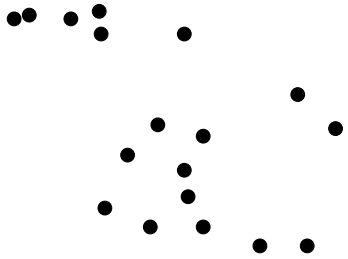


- Core points are points that have many points nearby
- Clusters contain
  - All core points that can be reached by following a sequence of core points
  - All non-core neighbours of core points (boundary points)

Density-Based Clustering Pseudo-Code

- Initially, no instance (point) has been visited.
- For each point x_i that has not been visited
  - If x_i is already assigned to a cluster, do nothing
  - Test whether x_i is a core point ($\geq$ *minNeighbours* instances within ϵ)
    - If x_i is not a core point, do nothing (this could be an outlier)
    - If x_i is a core point, make a new cluster, and call the *expand cluster* function (this spreads the "reaction" to nearby points)
- Function *expand cluster*
  - Assign to this cluster all x_j within distance ϵ of core point x_i to this cluster (including the core point)
  - For each new core point found, call *expand cluster* (recursively)
- Complexity: $O(n^2 d)$, for n instances in d-dimensional space

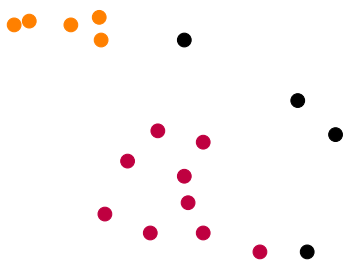
Density-Based Clustering – Example



$minNeighbours = 4$

Output: 2 clusters and 4 unassigned points

Density-Based Clustering – Example



$minNeighbours = 4$

Output: 2 clusters and 4 unassigned points

Interactive Demo!

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

Density-Based Clustering Issues

- What are the problems with Density-Based Clustering?
- Some points are **not assigned** to a cluster
  - Good or bad, depending on the application
- Ambiguity of non-core (boundary) points
- **Sensitive to the choice of ϵ and** *minNeighbours*
  - Original paper proposed an *elbow* method
  - Otherwise, not sensitive to initialization (except for boundary points)
- If you get a new example, **finding cluster is expensive**
  - Need to compute distances to m core points $O(md)$, or in worst case to all training points $O(nd)$.
- In high-dimensions, need a lot of points to *fill* the space

Ensemble Clustering

- We can consider ensemble methods for clustering
  - "consensus" clustering
- It's a good/important idea
  - Bootstrapping is widely-used
  - **Do clusters change if the data was slightly different?**
- But we need to be careful about how we combine models
- E.g., run k-means 20 times and then cluster using the mode of the assigned cluster labels $\{\hat{y}_i^m | 1 \leq m \leq 20\}$ for the i-th instance
- Normally, averaging across models doing different things is good
- But this is a bad ensemble method: **worse than k-means on its own**

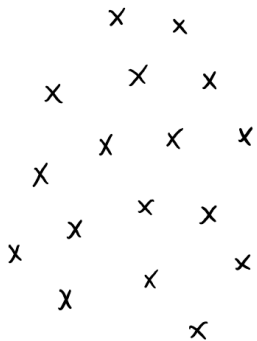
Label Switching Problem

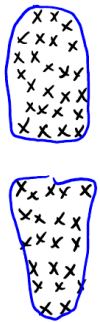
- This doesn't work because of the label switching problem
  - The cluster labels $\hat{y}_i$ are meaningless
  - We could get **same clustering with permuted labels** ("exchangeable" labels)
  - All $\hat{y}_i$ become **equally likely as number of initializations increases**
- Ensembles can't depend on label "meaning"
  - Don't ask: *Is point x_i in red square cluster?*, which is meaningless
  - Ask: *Is point x_i in the same cluster as x_j?*, which is meaningful

Hierarchical Clustering

Motivation: Differing Densities

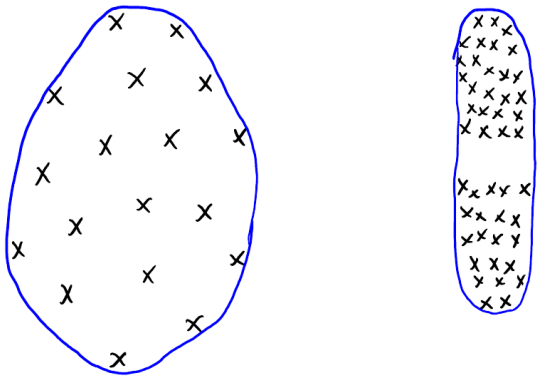
Consider density-based clustering on this data:





Motivation: Differing Densities

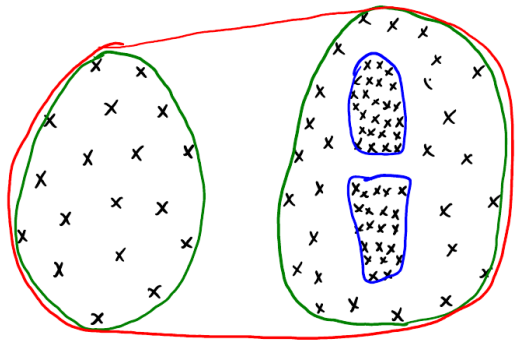
Increase epsilon and run it again:



There may be no density-level that gives you 3 clusters.

Motivation: Differing Densities

More complicated case:



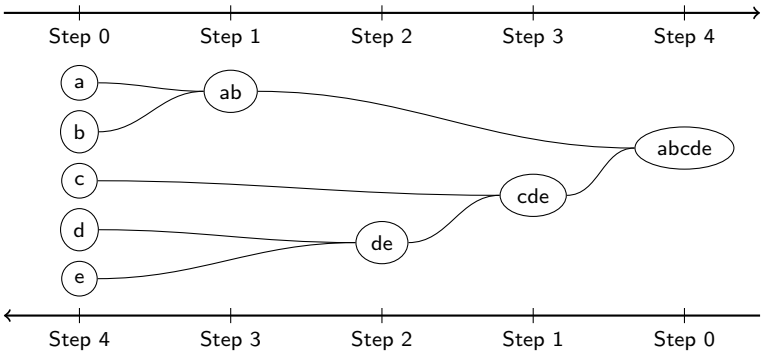
- We need to choose between **coarse/fine** clusters.
- Instead of fixed clustering, we often want hierarchical clustering.

Hierarchical Clustering

- Hierarchical clustering produces a **tree of clusters**
  - Each node in the tree splits the data into 2 or more cluster
  - Much more information than using a fixed clustering
  - Often have **individual data points as leaves**

Hierarchical Clustering

- Need *similarity/distance matrix* to guide the clustering process.
- Two approaches:



agglomerative

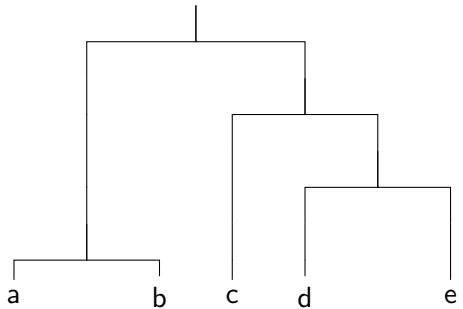
AGNES

divisive

DIANA

Dendrogram

- Result of all hierarchical clustering methods
- Gives all clusterings for all number of clusters



https://miro.medium.com/v2/resize:fit:720/1*ET8kCcPpr893vNZFs8j4xg.gif

Note

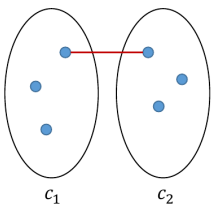
- The hierarchical structure is returned whether there exists one in the data or not
  - How would the dendrogram look if there is no structure in the data?
- A dendrogram is the description of the result of the algorithm, not a graphical summary of the data

AGNES – Aglomerative Nesting

Input: Instances $x_i \in X$, distance metric *dist*
$C \leftarrow \{\}$ //empty clustering
//Make each instance a single (singleton) cluster
foreach $x_i \in X$ **do**
    $C_i \leftarrow \{x_i\}$ Create a leaf at level 0 for every singleton cluster
    $C \leftarrow C \cup C_i$
end
//Merge closest clusters until single cluster left
while $|C| > 1$ **do**
    Let C_i and C_j be the clusters that minimize distance $dist(C_k, C_h)$ between any two clusters
    $C_i \leftarrow C_i \cup C_j$
    Create a parent of C_i and C_j at level $d(C_i, C_j)$
    $C \leftarrow C \setminus C_j$
end
return *Dendrogram of C*

Distances between Clusters

- How do you measure the distance $dist(C_1, C_2)$ between cluster C_1 and cluster C_2 ?

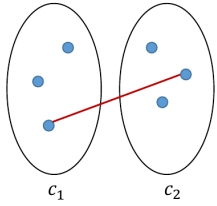


Single linkage

$$\min_{x_i \in C_1, x_j \in C_2} dist(x_i, x_j)$$

Issues: chaining/noise

Clusters: locally cohesive

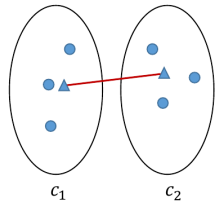


Complete linkage

$$\max_{x_i \in C_1, x_j \in C_2} dist(x_i, x_j)$$

Issues: outliers/noise

Clusters: compact (small diameter)



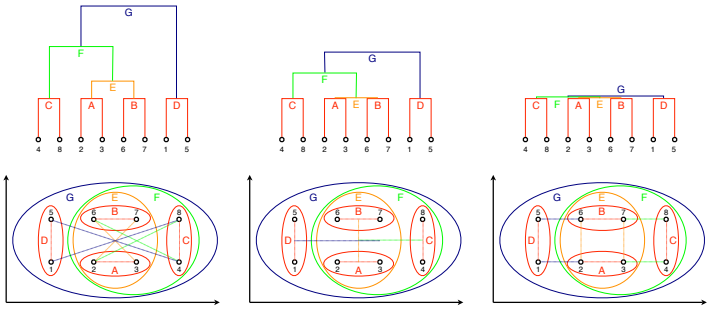
Centroid linkage

$$\frac{1}{|C_1|} \sum_{x_i \in C_1} x_i - \frac{1}{|C_2|} \sum_{x_j \in C_2} x_j$$

Issues: inversions

Clusters: spherical

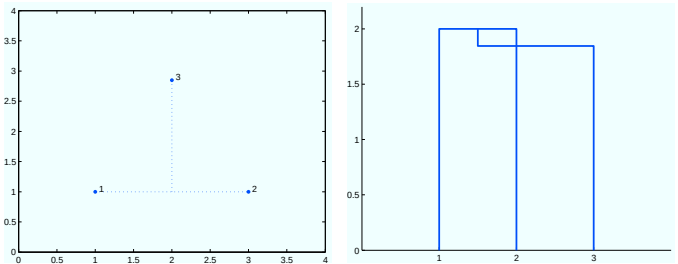
Distances between Clusters



source: cs.bris.ac.uk/ flach/mlbook/

We consider a regular grid of 8 points (2 rows by 4 columns). (Left) Complete linkage gives the impression that D is far removed from the rest. (Middle) With centroid linkage we see that A and B are not really discernible as separate clusters, even though they are found first. (Right) Single linkage most clearly demonstrates that there is no meaningful cluster structure in this set of points.

Centroid distance: inversion problem



source: cs.bris.ac.uk/flach/mlbook/

Points 1 and 2 are closer to each other than to point 3. However, the distance between point 3 to the centroid of the other two points is less than any of the pairwise distances.This results in a decrease in centroid linkage when adding point 3 to cluster $\{1, 2\}$, and hence a non-monotonic dendrogram.

Complexity of Agglomerative Clustering

- Computational complexity of basic implementation is $O(n^3)$.
  - One step costs $O(n^2 d = O(d)$ (standard) distance calculation between up to $O(n^2)$ points.
  - At most O(n) steps
  - Starting with n singleton clusters and merging 2 clusters in each step, after $O(n)$ steps only 1 cluster will be left.
- This can be reduced to $O(n^2 d \log n)$ with a priority queue: store distances in a sorted order, only update the distances that change.

Cluster Quality

Cluster Quality

- Two methods: extrinsic vs. intrinsic
- Extrinsic: supervised, i.e., the ground truth is available
  - Compare a clustering against the ground truth using certain clustering quality measure
    For example: Purity, Normalized Mutual Information and Rand Index
- Intrinsic: unsupervised, i.e., the ground truth is unavailable
  - Maximize the within-cluster similarity, and minimize the between-cluster similarity
  - Evaluate the goodness of a clustering by considering how well **the clusters are separated, and how compact the clusters are**
    For example: Silhouette coefficient

Extrinsic Methods

- A good clustering quality measure $Q(C, G)$ for a clustering C given the ground truth G, should satisfy following 4 essential criteria
  - Cluster homogeneity: the purer, the better
  - Cluster completeness: should assign objects belonging to the same category in the ground truth to the same cluster
  - Rag bag: putting a heterogeneous object into a pure cluster should be penalized more than putting it into a rag bag (i.e., "miscellaneous" or "other" category)
  - Small cluster preservation: splitting a small category into pieces is more harmful than splitting a large category into pieces

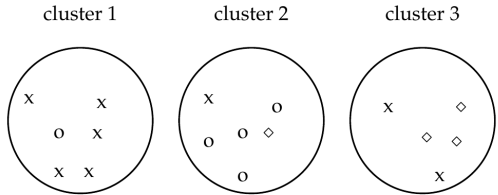
Purity

- Given a set of points $D = \{x_1, \ldots, x_n\}$, and two partitions of D, $C = \{C_1, \ldots, C_k\}$ and $G = \{G_1, \ldots, G_q\}$, with C being the clustering and G being the ground truth categorization of the points, the purity is defined as:

$$Purity(C, G) = \frac{1}{n} \sum_{i=1}^{k} max_j |C_i \cap G_j|$$

  Assign each cluster to the category which is most frequent in the cluster, and take the accuracy of this assignment (i.e. number of correctly assigned points over n).
- The **perfect clustering** result has a **purity of 1**
- When $k = n$, each point is a cluster, what is the purity?

Example: Purity



	Cluster1	Cluster 2	Cluster 3	Sum
Cluster size	6	6	5	$n = 17$
Majority category	x	○	◇	
Majority category size	5	4	3	$m = 12$

Purity is $m/n \approx 0.71$

Rand Index

- Given the a set of points D and two partitions of D, C and G as defined previously, the Rand Index measures the overall accuracy of the clustering on the level of pairs of data points:

$$RI(C, G) = \frac{TP + TN}{TP + FP + TN + FN} = \frac{TP + TN}{\binom{n}{2}} = \frac{TP + TN}{n \cdot (n-1)/2}$$

  - TP - the number of **correctly clustered pairs** of points (in the same cluster and in the same category)
  - TN - the number of **correctly separated pairs** of points (belonging to different categories are also assigned in different clusters)
  - FN and FP - numbers of incorrectly clustered and separated pairs, respectively
- Issue: **equal weight to false positives and false negatives**, better use F-measure.

Example: Rand Index

Assume D has five data points belonging to two categories, i.e.
$G = \{G_1 = \{x_1, x_2\}, G_2 = \{x_3, x_4, x_5\}\}$. We have $5 \cdot 4/2 = 10$ possible data point pairs.

- 4 pairs are in the same categories, i.e. $(x_1, x_2), (x_3, x_4), (x_3, x_5), (x_4, x_5)$
- 6 pairs have points in different categories

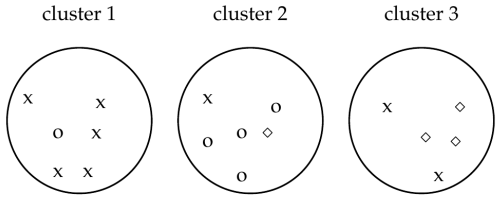
Assume we have the following clustering $C = \{C_1 = \{x1, x2, x3\}, C_2 = \{x4, x5\}\}$

- Two (out of the four) pairs are incorrectly separated (x_3, x_4), $(x_3 x_5)$, and two (out of the six) pairs are incorrectly clustered together.
- We can tabulate this as a two-by-two contingency table:

	Same category	Different category	Sum
Same cluster	$TP = 2$	$FP = 2$	4
Different cluster	$FN = 2$	$TN = 4$	6
Sum	4	6	10

$RI = 6/10 = 0.6$

Example: Rand Index



	Cluster 1	Cluster 2	Cluster 3	Sum
Cluster size	6	6	5	$n = 17$
$TP + FP$	$6 \cdot 5/2$	$6 \cdot 5/2$	$5 \cdot 4/2$	40 pairs
TP	$5 \cdot 4/2$ (x)	$4 \cdot 3/2$ ($\circ$)	$3 \cdot 2/2(\diamond) + 1(x)$	20 pairs

	Category x	Category $\circ$	Category $\diamond$	Sum
Category size	8	5	4	$n = 17$
$TP + FN$	$8 \cdot 7/2$	$5 \cdot 4/2$	$4 \cdot 3/2$	44 pairs

$FN = 44 - TP = 24$ $\quad$ $TN = 17 \cdot 16/2 - (TP + FP + FN) = 72$ pairs

$$RI = (TP + TN)/136 = 92/136 \approx 0.68$$

Mutual Information

- Mutual information, **an information-theoretic measure** about the amount of information by which our knowledge about the categories increases when we are told what the clusters are.
- It is **0 if the clustering is random** with respect to category membership.

$$I(C, G) = \sum_i \sum_j p(C_i, G_j) \log \left(\frac{p(C_i, G_j)}{p(C_i) \cdot p(G_j)} \right)$$
$$= \sum_i \sum_j \frac{|C_i \cap G_j|}{n} \log \left(\frac{n \cdot |C_i \cap G_j|}{|C_i| \cdot |G_j|} \right)$$

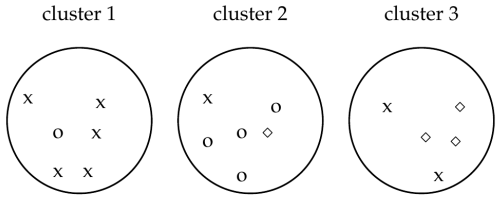
Normalized Mutual Information

- Mutual information is **1 for prefect clustering**, but also 1 for k=n (same problem as with purity).
- Normalized mutual information(NMI) in range[0,1], and penalizes for too many clusters:

$$NMI(C, G) = \frac{I(C, G) \cdot 2}{H(C) + H(G)}$$

- where H is the entropy $H(C) = -\sum_i p(C_i) \log p(C_i) = -\sum_i \frac{|C_i|}{n} \log \frac{|C_i|}{n}$

Example: NMI (for homework)



	Cluster 1 (C_1)	Cluster 2 (C_2)	Cluster 3 (C_3)	Sum
Category x (G_1)	$\lvert C_1 \cap G_1 \rvert = 5$	1	2	$\lvert G_1 \rvert = 8$
Category $\circ$ (G_2)	1			
Category $\diamond$ (G_3)	0			
Sum	$\lvert C_1 \rvert = 6$			n = 17

Fill the contingency table with the missing counts and use them to verify that NMI = 0.36.

Silhouette Coefficient

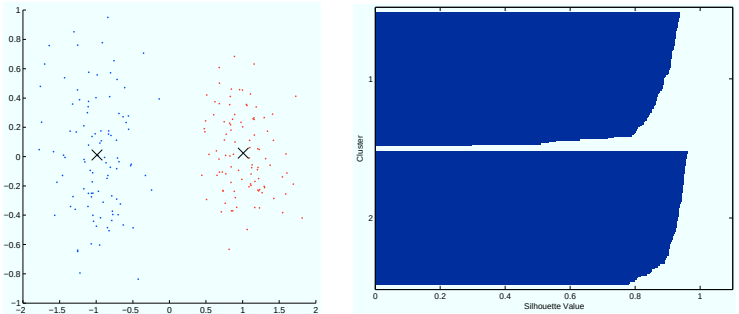
- A measure for the distance within a cluster and between clusters
- Range [-1,1]. More positive, better cluster

$$s(x) = \frac{b(x) - a(x)}{max(a(x), b(x))}$$

- $a(x)$ is the mean distance between a point x and all other points in the same cluster (how **compact is the cluster** C_i to which x belongs).
- $b(x)$ is the mean distance between a point x and all other points in the nearest cluster (how much **separated is the cluster** C_i to which x belongs)
- Fitness of the cluster C_i in given clustering can be measured by the Average Silhouette Coefficient value of all points in the cluster.

Visualized Silhouette Coefficient

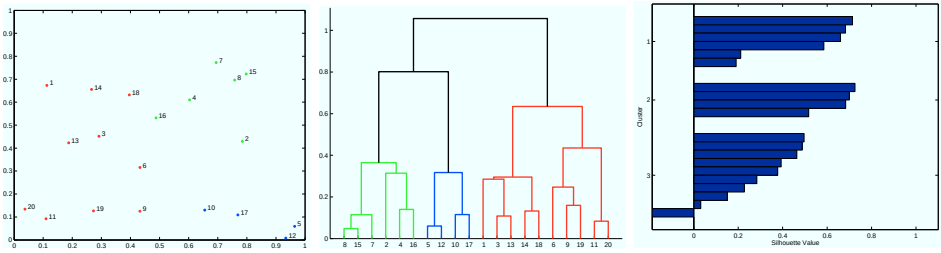
Typically we sort and plot $s(x)$ for each instance, grouped by cluster.



source: cs.bris.ac.uk/flach/mlbook/

Almost all points have high $s(x)$, which means that they are much closer, on average, to the other members of their cluster than to the members of the neighbouring cluster.

Visualized Silhouette Coefficient



source: cs.bris.ac.uk/flach/mlbook/

(Left) 20 data points, generated by uniform random sampling.
(Middle) The dendrogram generated from complete linkage. The three clusters suggested by the dendrogram are spurious as they cannot be observed in the data.
(Right) The rapidly decreasing $s(x)$ in each cluster confirm the absence of a strong cluster structure. Point 18 has a negative s(x) - on average closer to the green points (8,15,7,2,4,16) than to the other red points(1,3,13,14,6,9,19,11,20).

Summary

Summary

- Clustering: finding 'groups' of related data points
- K-means clustering
  - Simple iterative strategy that partitions space into convex spaces.
  - Fast but sensitive to initialization.
- Density-based clustering
  - *Expand* and *merge* dense regions of points to find clusters
  - Not sensitive to initialization or outliers
  - Useful for finding non-convex connected clusters
- Ensemble clustering combines multiple clusterings
  - Can work well but need to account for label switching
- Hierarchical clustering: more informative than fixed clustering
  - Agglomerative: Each point starts as a cluster, sequentially merge clusters.
  - Divisive: All points form one initial cluster, sequentially split into clusters.

Literature and other resources

- Chapter 10 of Han's *Data Mining: Concepts and Techniques*
- Chapter 16 and 17 in the online book `https://www-nlp.stanford.edu/IR-book/`
- Clustering methods in Python package scikit-learn
  `https://scikit-learn.org/stable/modules/clustering.html`

SCIENCE
SCHOOL OF COMPUTER SCIENCE

Thank you for your attention!

https://ml.auckland.ac.nz