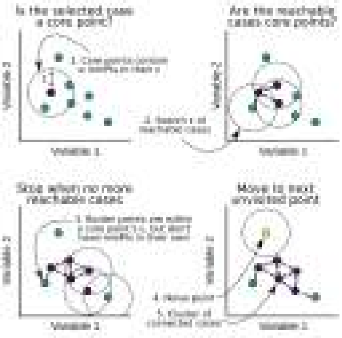
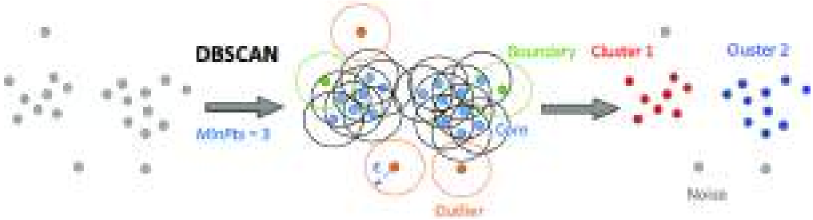
DBSCAN

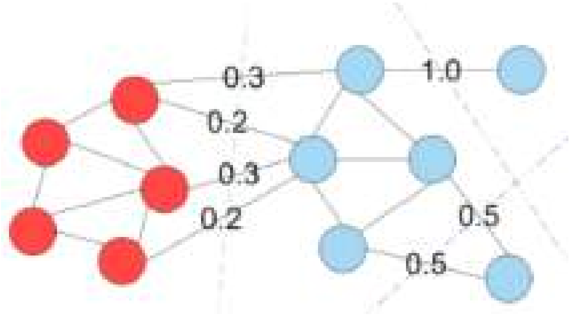
Density-based spatial clustering of applications with noise



- $\rightarrow$ Divide points into 3 categories (core, boundary, outliers) whether there are at least *minPts* in their ϵ-neighborhood or not
- $\rightarrow$ Find the connected components of core points (ignoring all non-core points)
- $\rightarrow$ Assign non-core points to nearby cluster if it is less than ϵ away, otherwise assign to noise.

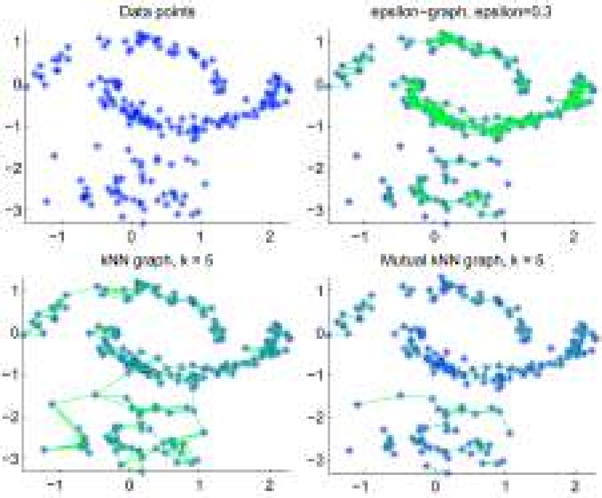


Overall idea: View clustering task as a min-cut operation in a graph



Overall idea: View clustering task as a min-cut operation in a graph

→ Compute similarity graph (but which one?) of data $\mathbf{x}_1, \ldots, \mathbf{x}_n$



Overall idea: View clustering task as a min-cut operation in a graph

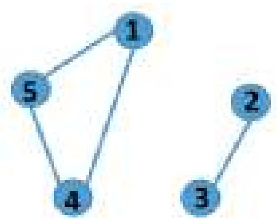
- $\rightarrow$ Compute similarity graph (but which one?) of data $\mathbf{x}_1, \ldots, \mathbf{x}_n$
- $\rightarrow$ Compute (weighted) adjacency matrix $\mathbf{W}$, degree matrix $\mathbf{D}$ and Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$
- $\rightarrow$ Perform eigendecomposition of $\mathbf{L} = (\mathbf{E}, \mathbf{\Lambda})$

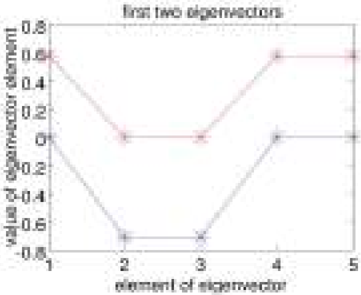
Spectral clustering

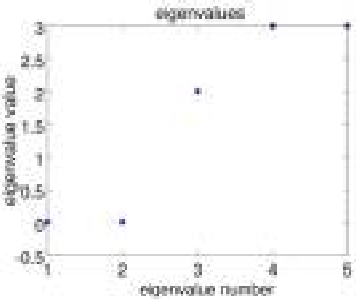
Overall idea: View clustering task as a min-cut operation in a graph

- → Compute similarity graph (but which one?) of data $\mathbf{x}_1, \dots, \mathbf{x}_n$
- → Compute (weighted) adjacency matrix $\mathbf{W}$, degree matrix $\mathbf{D}$ and Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$
- → Perform eigendecomposition of $\mathbf{L} = (\mathbf{E}, \mathbf{\Lambda})$

Fact #1 0 is an eigenvalue of $\mathbf{L}$ with multiplicity $\equiv$ # connected components in graph, its eigenvectors are identity vectors of those connected components





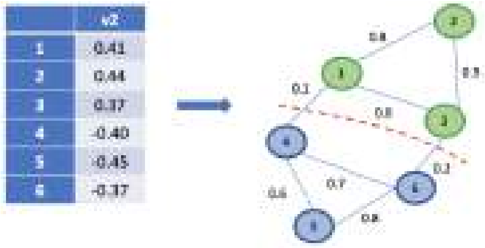


Overall idea: View clustering task as a min-cut operation in a graph

- → Compute similarity graph (but which one?) of data $\mathbf{x}_1, \dots, \mathbf{x}_n$
- → Compute (weighted) adjacency matrix $\mathbf{W}$, degree matrix $\mathbf{D}$ and Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$
- → Perform eigendecomposition of $\mathbf{L} = (\mathbf{E}, \mathbf{\Lambda})$

Fact #1 0 is an eigenvalue of $\mathbf{L}$ with multiplicity $\equiv$ # connected components in graph, its eigenvectors are identity vectors of those connected components

Fact #2 Eigenvector of smallest non-zero eigenvalue (Fiedler vector) gives the normalized min-cut of graph



Overall idea: View clustering task as a min-cut operation in a graph

→ Compute similarity graph (but which one?) of data $\mathbf{x}_1, \ldots, \mathbf{x}_n$

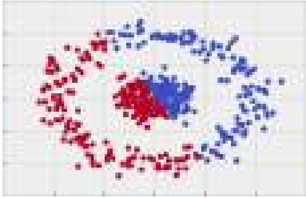
→ Compute (weighted) adjacency matrix $\mathbf{W}$, degree matrix $\mathbf{D}$ and Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$

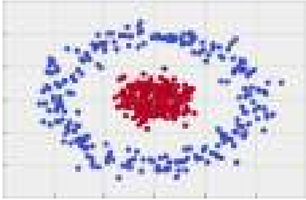
→ Perform eigendecomposition of $\mathbf{L} = (\mathbf{E}, \mathbf{\Lambda})$

Fact #1 0 is an eigenvalue of $\mathbf{L}$ with multiplicity $\equiv$ # connected components in graph, its eigenvectors are identity vectors of those connected components

Fact #2 Eigenvector of smallest non-zero eigenvalue (Fiedler vector) gives the normalized min-cut of graph

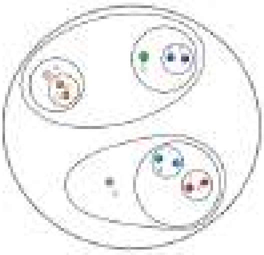
→ Performs k-means clustering on the k smallest eigenvectors $[\mathbf{e}_1, \ldots, \mathbf{e}_k]_{n \times k}$





Goal: Generate a sequence of nested clusters and order them in a hierarchy, represented by a dendogram:

- $\rightarrow$ Leaves of the dendogram $=$ initial data
- $\rightarrow$ Inner nodes of the dendogram $=$ clusters

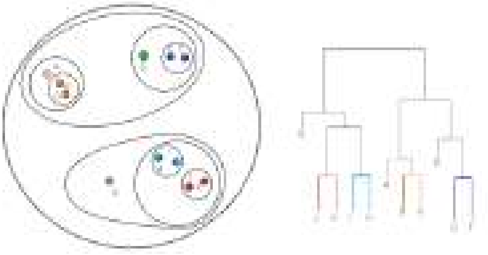


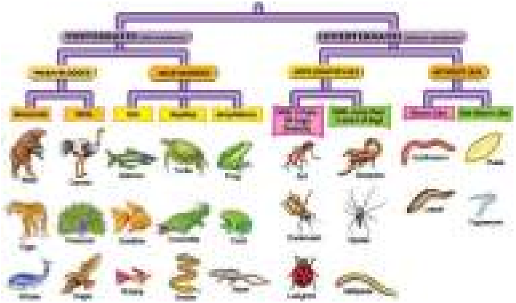
Hierarchical clustering

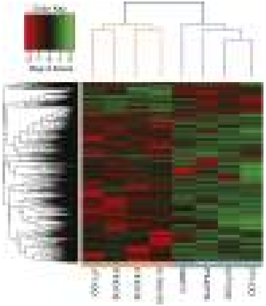
A very natural way of handling data

Goal: Generate a sequence of nested clusters and order them in a hierarchy, represented by a dendogram:

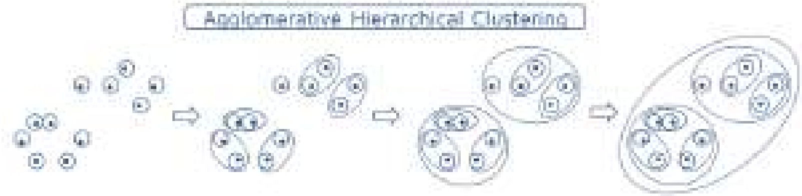
- $\rightarrow$ Leaves of the dendogram $=$ initial data
- $\rightarrow$ Inner nodes of the dendogram $=$ clusters



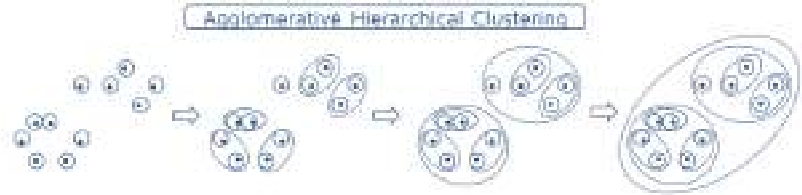




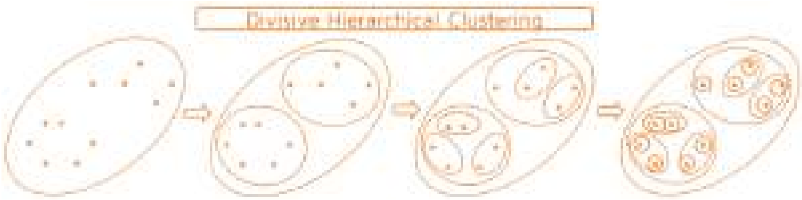
Agglomerative clustering $\rightarrow$ merge clusters from fine to coarse (bottom-up approach)



Agglomerative clustering → merge clusters from fine to coarse (bottom-up approach)



Divisive clustering → split clusters (top-down approach)



→ Needs some heuristics to avoid the $\mathcal{O}(2^n)$ ways of splitting each cluster...
→ Mainly used to index vector data

Hierarchical clustering

A bestiary on how to compute the distance between clusters

Single linkage: distance between closest elements in clusters
→ $d(\mathcal{C}_1, \mathcal{C}_2) = \min_{\mathbf{x}_i \in \mathcal{C}_1, \mathbf{x}_j \in \mathcal{C}_2} d(\mathbf{x}_i, \mathbf{x}_j)$
→ produces long chains

Complete linkage: distance between farthest elements in clusters
→ $d(\mathcal{C}_1, \mathcal{C}_2) = \max_{\mathbf{x}_i \in \mathcal{C}_1, \mathbf{x}_j \in \mathcal{C}_2} d(\mathbf{x}_i, \mathbf{x}_j)$
→ forces spherical clusters with consistent diameters

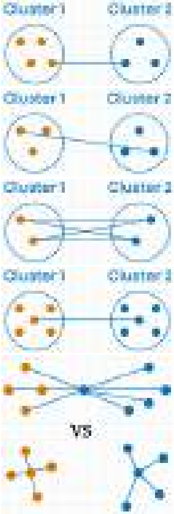
Average linkage: average of all pairwise distances
→ $d(\mathcal{C}_1, \mathcal{C}_2) = \frac{1}{|\mathcal{C}_1|} \frac{1}{|\mathcal{C}_2|} \sum_{\mathbf{x}_i \in \mathcal{C}_1} \sum_{\mathbf{x}_j \in \mathcal{C}_2} d(\mathbf{x}_i, \mathbf{x}_j)$
→ more robust to outliers

Centroid linkage: distance between centroids of clusters
→ $d(\mathcal{C}_1, \mathcal{C}_2) = d(\boldsymbol{\mu}_{\mathcal{C}_1}, \boldsymbol{\mu}_{\mathcal{C}_2})$ with $\boldsymbol{\mu}_{\mathcal{C}_k}$ centroid of $\mathcal{C}_k$
→ "hierarchical k-means"

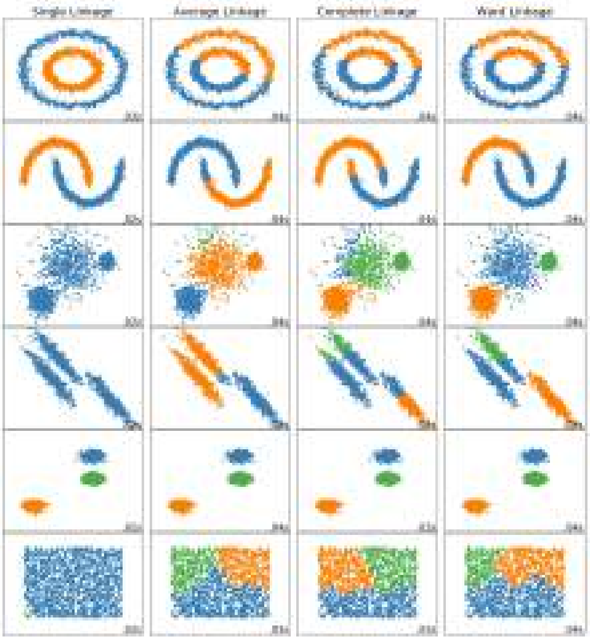
Ward's linkage: increase in variance when merging the two clusters

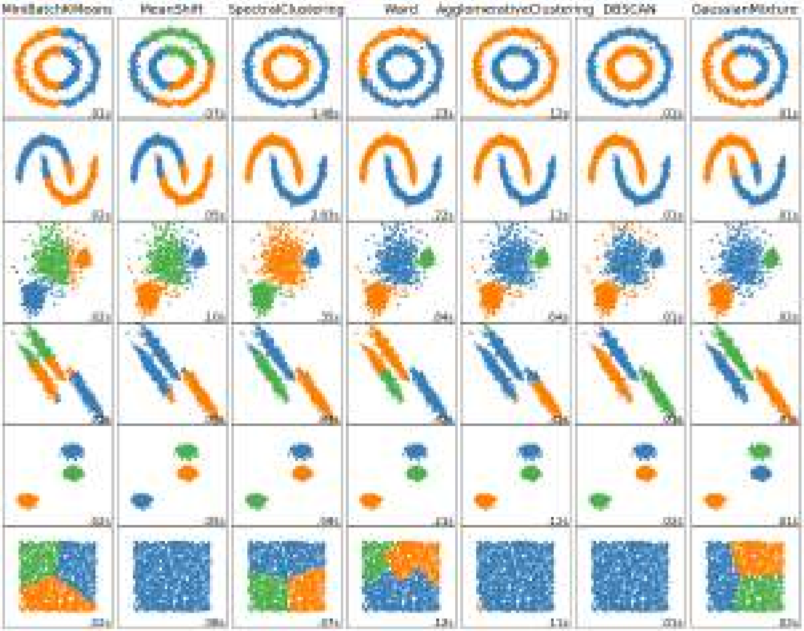
$$\rightarrow d(\mathcal{C}_1, \mathcal{C}_2) = \sum_{\mathbf{x} \in \mathcal{C}_1 \cup \mathcal{C}_2} \|\mathbf{x} - \boldsymbol{\mu}_{\mathcal{C}_1 \cup \mathcal{C}_2}\|^2 - \sum_{\mathbf{x}_i \in \mathcal{C}_1} \|\mathbf{x}_i - \boldsymbol{\mu}_{\mathcal{C}_1}\|^2 - \sum_{\mathbf{x}_j \in \mathcal{C}_2} \|\mathbf{x}_j - \boldsymbol{\mu}_{\mathcal{C}_2}\|^2$$

→ can be rewritten as $d(\mathcal{C}_1, \mathcal{C}_2) = \frac{|\mathcal{C}_i||\mathcal{C}_j|}{|\mathcal{C}_i| + |\mathcal{C}_j|} \|\boldsymbol{\mu}_{\mathcal{C}_i} - \boldsymbol{\mu}_{\mathcal{C}_j}\|^2$



Comparison of hierarchical clustering strategies




MiniBatchKMeans
MeanShift
SpectralClustering
Ward
AgglomerativeClustering
DBSCAN
GaussianMixture

Summary of the presented clusering methods (with $n =$ number of samples in the dataset).

Technique	Type	Parametric	# clusters	Computational
K-Means	Centroid	yes	fixed a priori	$\mathcal{O}(n)$
GMM	Centroid	yes	fixed a priori	$\mathcal{O}(n)$
Mean Shift	Centroid/Density	no	no control	$\mathcal{O}(n^2)$
DBSCAN	Density	no	no control	$\mathcal{O}(n \log n)$
Spectral	Centroid/Density	no	a posteriori	$\mathcal{O}(n^3)$
Hierarchical Agglomerative	Hierarchical	no	a posteriori	$\mathcal{O}(n^3)$, fast: $\mathcal{O}(n^2 \log n)$
Hierarchical Divisive	Hierarchical	no	a posteriori	naive: $\mathcal{O}(2^n)$, real: $\mathcal{O}(kn)$

Non-contractual.

Some further reading

SLIC: Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Süsstrunk, S. (2012). SLIC superpixels compared to state-of-the-art superpixel methods. IEEE transactions on pattern analysis and machine intelligence.

Partitioning around medoids: Kaufman, L., & Rousseeuw, P. J. (2009). Finding groups in data: an introduction to cluster analysis (Vol. 344). John Wiley & Sons.

FCM: Bezdek, J. C., Ehrlich, R., & Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. Computers & geosciences.

GMM clustering: McLachlan, G. J., & Basford, K. E. (1988). Mixture models: Inference and applications to clustering (Vol. 38). New York: M. Dekker.

EM algorithm: Gupta, M. R., & Chen, Y. (2011). Theory and use of the EM algorithm. Now Publishers Inc.

Kernel density estimation: Parzen, E. (1962). On estimation of a probability density function and mode. The annals of mathematical statistics.

Mean shift clustering: Comaniciu, D., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. IEEE Transactions on pattern analysis and machine intelligence.

Spectral clustering: Von Luxburg, U. (2007). A tutorial on spectral clustering. Statistics and computing.

DBSCAN: Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the 2nd International Conference on Knowledge Discovery and Data mining.

Hiearchical clustering: Murtagh, F., & Contreras, P. (2012). Algorithms for hierarchical clustering: an overview. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery.

Overview: A Comprehensive Survey of Clustering Algorithms Xu, D. & Tian, Y. Ann. Data. Sci. (2015) 2: 165.