Mathematical techniques in data science

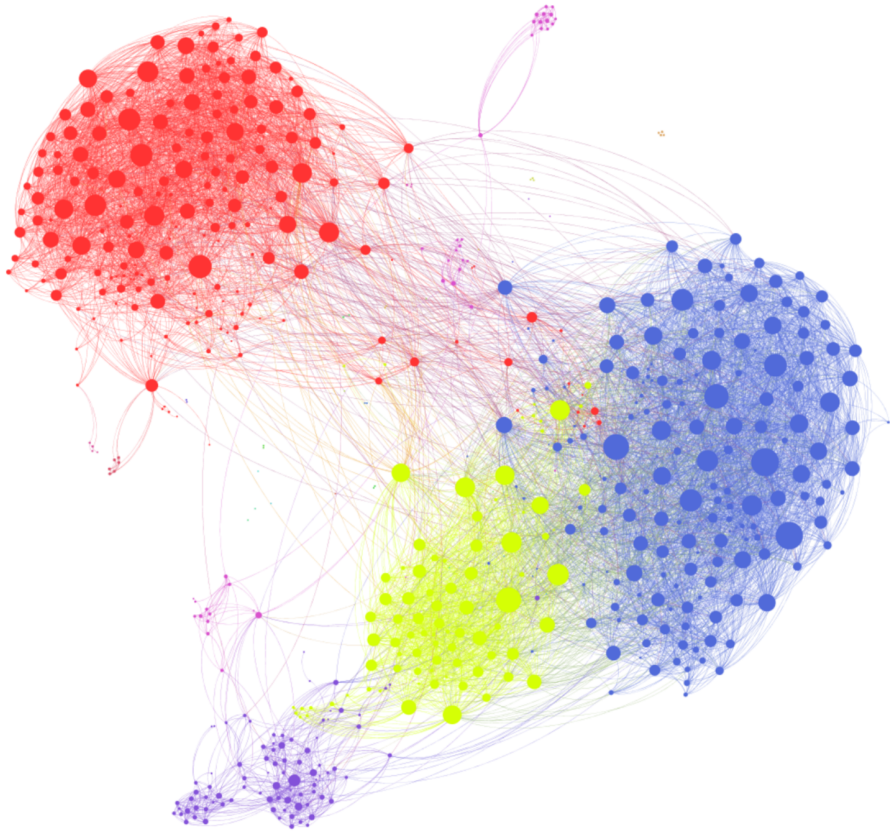
Lecture 27: Clustering: Random topics

April 26th, 2019

Schedule

Week	Chapter
1	Chapter 2: Intro to statistical learning
3	Chapter 4: Classification
4	Chapter 9: Support vector machine and kernels
5, 6	Chapter 3: Linear regression
7	Chapter 8: Tree-based methods + Random forest
8	
9	Neural networks
12	PCA $\rightarrow$ Manifold learning
11	**Clustering: K-means $\rightarrow$ Spectral Clustering**
10	Bayesian methods + UQ
13	Reinforcement learning/Online learning/Active learning
14	Project presentation

Clustering

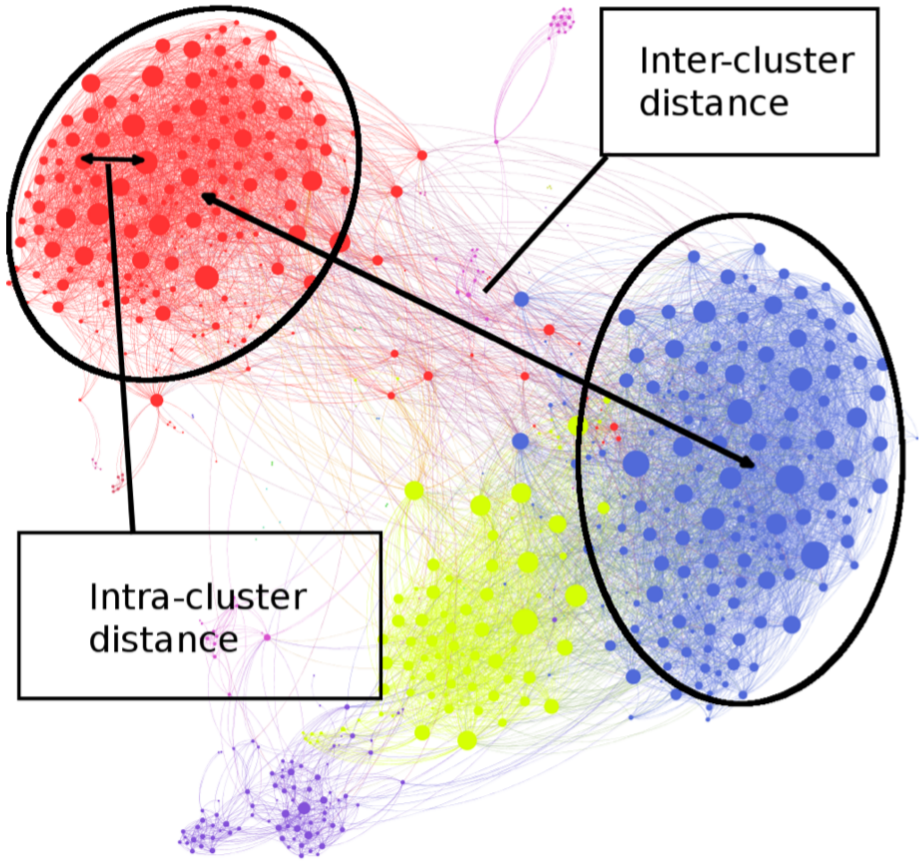


- Unsupervised problem
- Want to label points according to a measure of their similarity

Clustering

We try to partition observations into "clusters" such that:

- Intra-cluster distance is minimized.
- Inter-cluster distance is maximized.



K-means clustering

The K-means algorithm is a popular algorithm to cluster a set of points in $\mathbb{R}^p$.

- We are given n observations $x_1, x_2, \ldots, x_n \in \mathbb{R}^p$.
- We are given a number of clusters K.
- We want a partition $\hat{S} = \{S_1, \ldots, S_K\}$ of $\{x_1, \ldots, x_n\}$ such that

$$\hat{S} = \operatorname*{argmin}_S \sum_{i=1}^{K} \sum_{x_j \in S_i} \|x_j - \mu_i\|^2,$$

where $\mu_i = \frac{1}{|S_i|} \sum_{x_j \in S_i} x_j$ is the mean of the points in S_i (the "center" of S_i).

Lloyd's algorithm

Lloyds's algorithm for K-means clustering

- Denote by $C(i)$ the cluster assigned to x_i.
- Lloyds's algorithm provides a heuristic method for optimizing the K-means objective function.

Start with a "cluster centers" assignment $m_1^{(0)}, \ldots, m_K^{(0)}$. Set $t := 0$. Repeat:

1. Assign each point x_j to the cluster whose mean is closest to x_j:

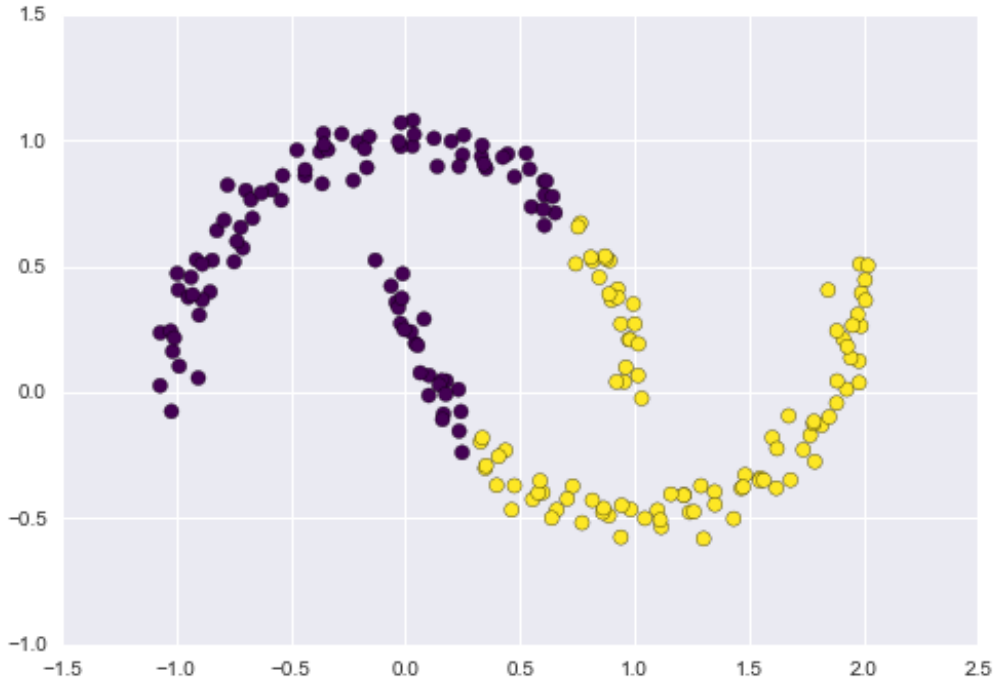
$$S_i^{(t)} := \{x_j : \|x_j - m_i^{(t)}\|^2 \leq \|x_j - m_k^{(t)}\|^2 \ \forall k = 1, \ldots, K\}.$$

2. Compute the average $m_i^{(t+1)}$ of the observations in cluster i:

$$m_i^{(t+1)} := \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j.$$

Issues with k-means

- k-means is limited to linear cluster boundaries



- Solution: adding non-linearities to the model
  - kernel k-means
  - spectral clustering

Kernel k-means = kernel trick + k-means

- Ideas:
  - maps the data to a high-dimensional space (called feature space) by a non-linear function ϕ to separate the clusters linearly
  - Using this high-dimensional representation to run k-means
  - Project the data back to the original space to identify the clusters
- Note: the kernel trick works best if we don't have to construct $\phi(x)$ explicitly, but can compute

$$K(x, y) = \langle \phi(x), \phi(y) \rangle$$

- For k-means, we need to compute

$$\|\phi(x_i) - m_j\|^2$$

Kernel k-means

Input: K: kernel k: number of clusters

Output: $C_1, ..., C_k$: partitioning of the points

1. Initialize the k clusters: $C_1^{(0)}, ..., C_k^{(0)}$.
2. Set $t = 0$.
3. For each point $\mathbf{a}$, find its new cluster index as

$$j^*(\mathbf{a}) = \text{argmin}_j \|\phi(\mathbf{a}) - \mathbf{m}_j\|^2, \text{ using (2).}$$

4. Compute the updated clusters as

$$C_j^{t+1} = \{\mathbf{a} \ : \ j^*(\mathbf{a}) = j\}.$$

5. If not converged, set $t = t + 1$ and go to Step 3; Otherwise, stop.

Spectral clustering: overview

1. Construct a *similarity matrix* measuring the similarity of pairs of objects.
2. Use the similarity matrix to construct a (weighted or unweighted) graph.
3. Compute eigenvectors of the *graph Laplacian* (builds an embedding of the graph into $\mathbb{R}^p$).
4. Cluster the graph

Neighbor graph

Step 1: Construct the neighbor graph

- For each point, determine either
  - K nearest neighbors
  - all points in a fixed radius
- each point is connected to its neighbours
- edge length equal to ~~Euclidean distance between the points~~

$$W_{ij} = e^{-\frac{|x_i - x_j|^2}{4t}}$$

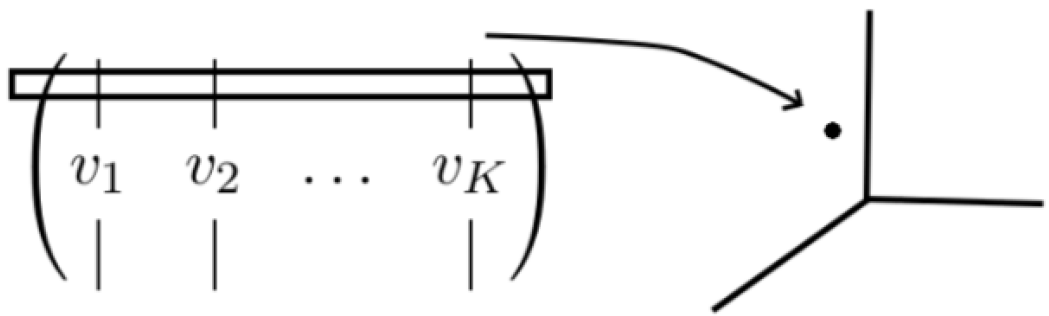
Graph embedding

Step 2:

- Compute eigenvectors of the (normalized or unnormalized) graph Laplacian

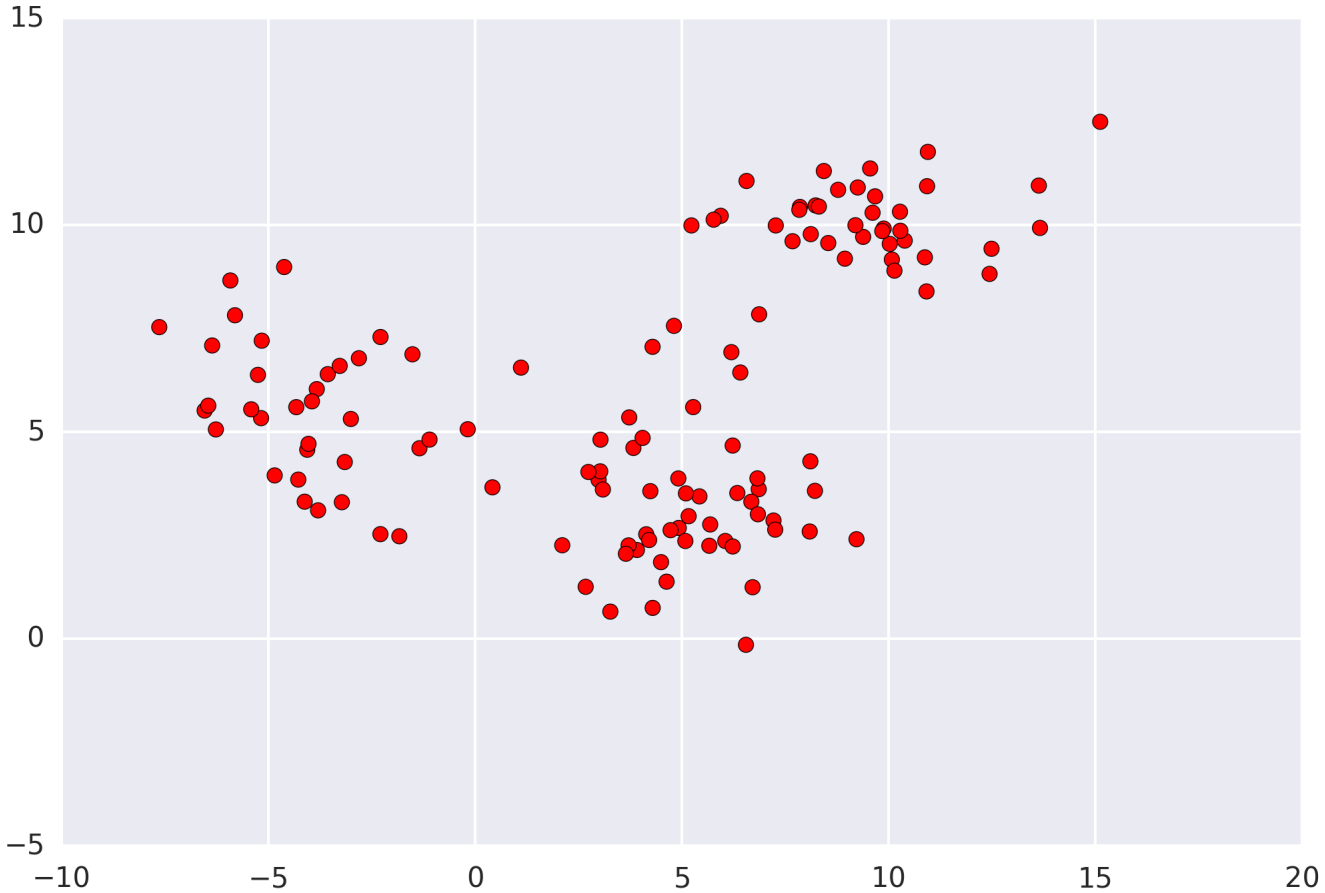
$$L = D - W, \quad L_{sym} = D^{-1/2} L D^{-1/2}$$

- Construct a matrix containing the smallest K eigenvectors of L or L_{sym} as columns
- Normalize the rows to have norm 1
- Each row identifies a vertex of the graph to a point in $\mathbb{R}^K$



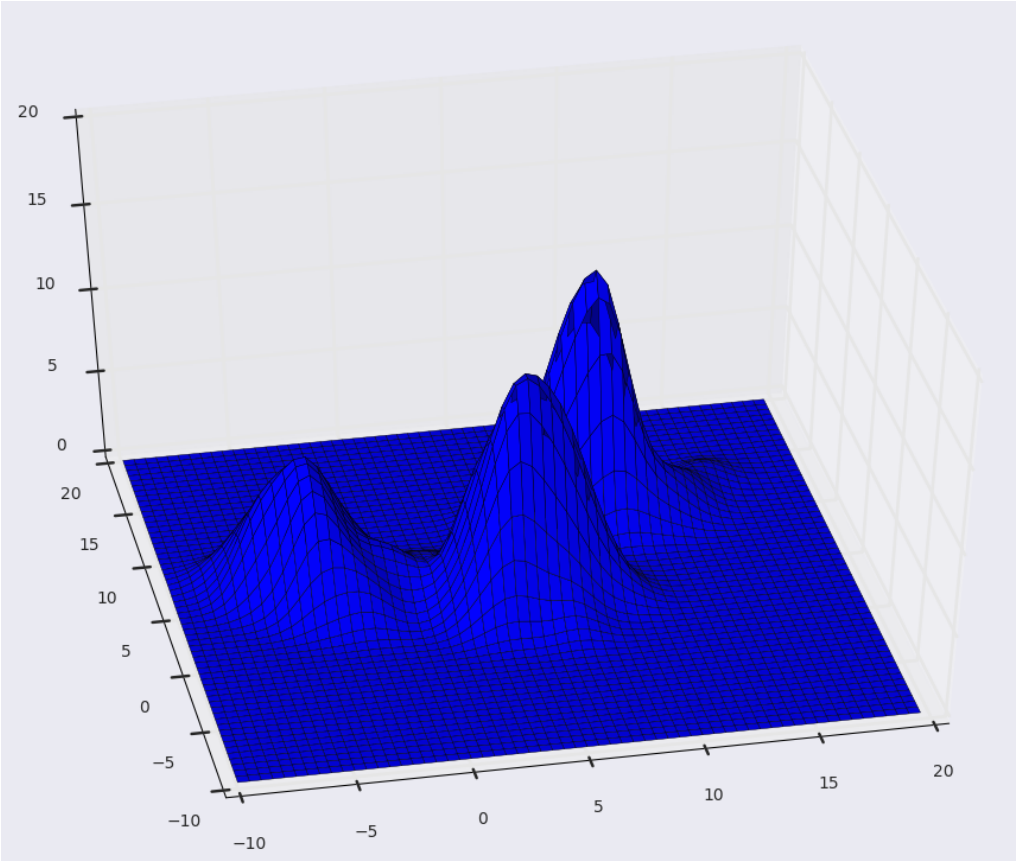
Other clustering methods

Mean shift clustering



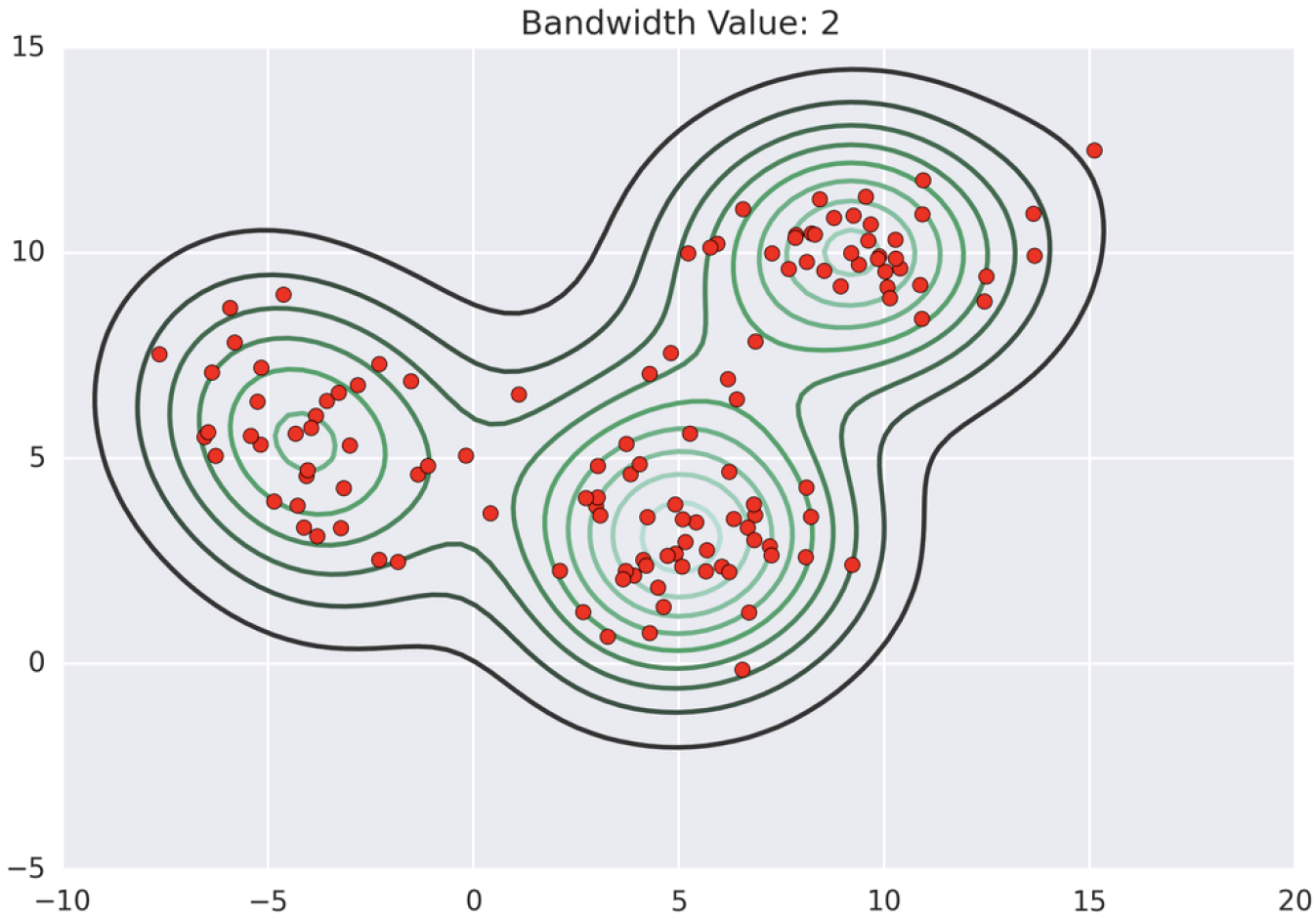
Idea #1: density estimation

Mean shift clustering



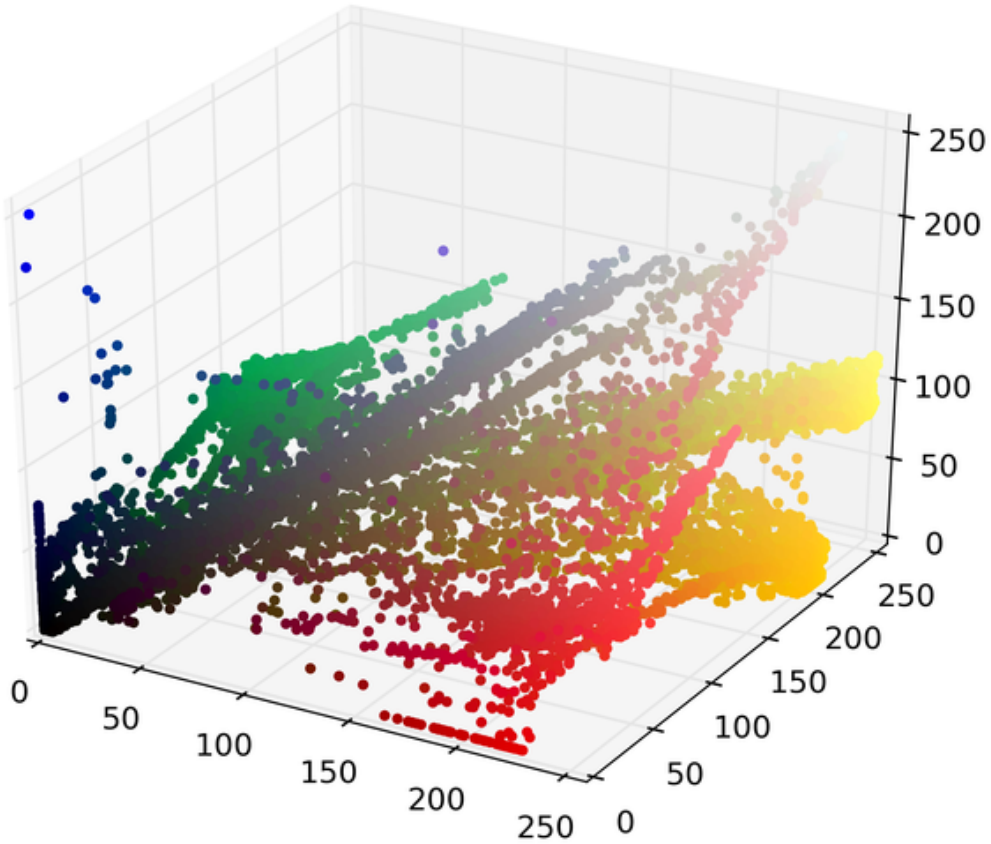
Idea #2: searching for the modes by gradient ascend

Mean shift clustering



Mean shift

Mean shift clustering



Mean shift

Kernel density estimator (KDE)

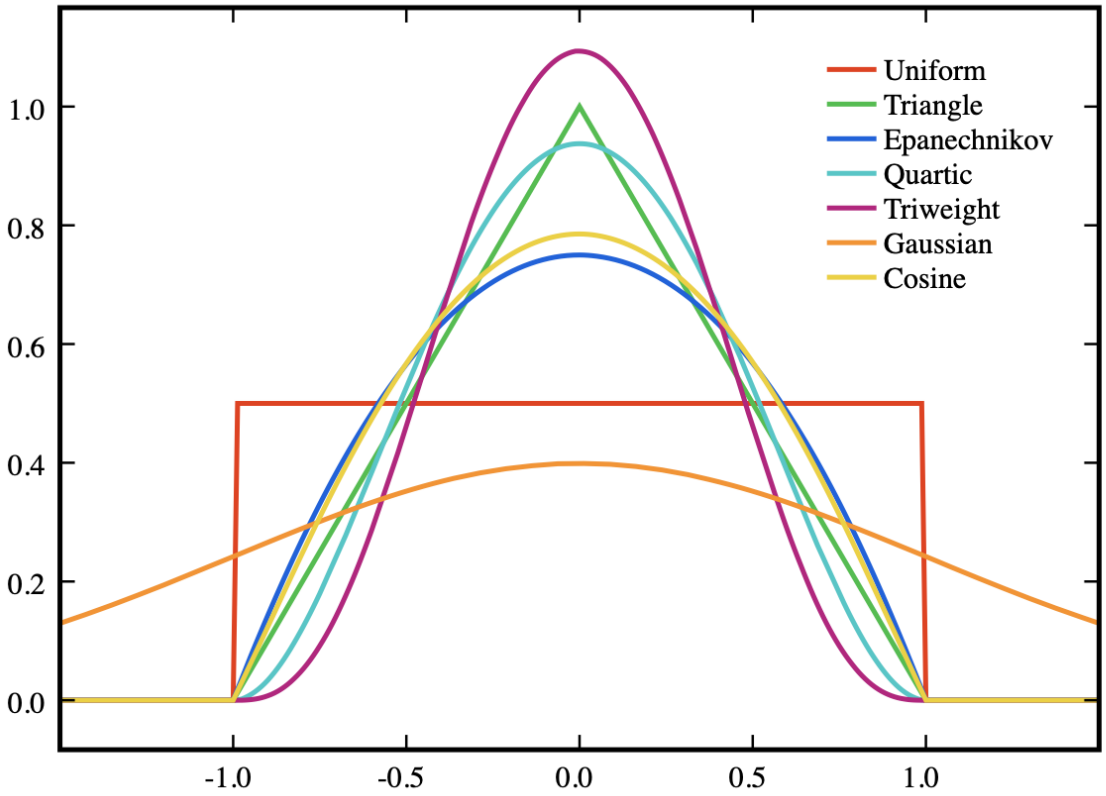
- Given data set $\{x_1, x_2, \ldots, x_n\}$, the density function is estimated by

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

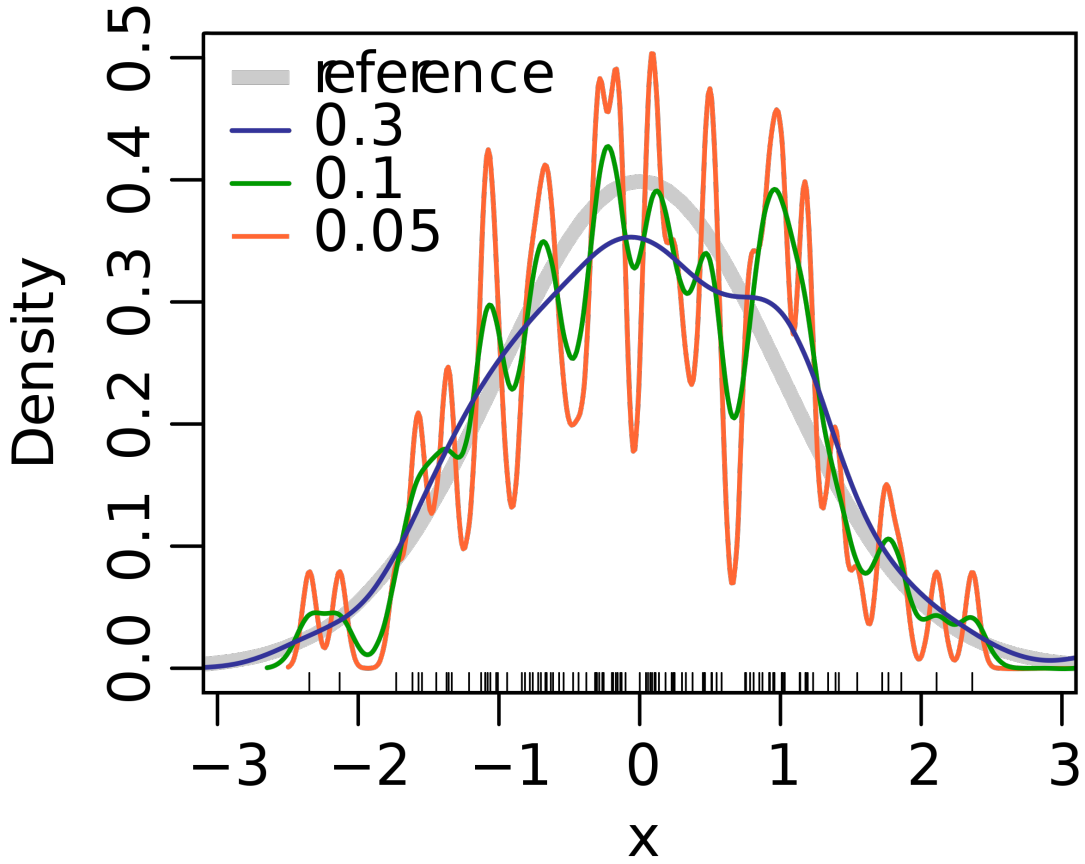
where

- h is the bandwidth
- K is a kernel: symmetric function around 0, integrate to 1

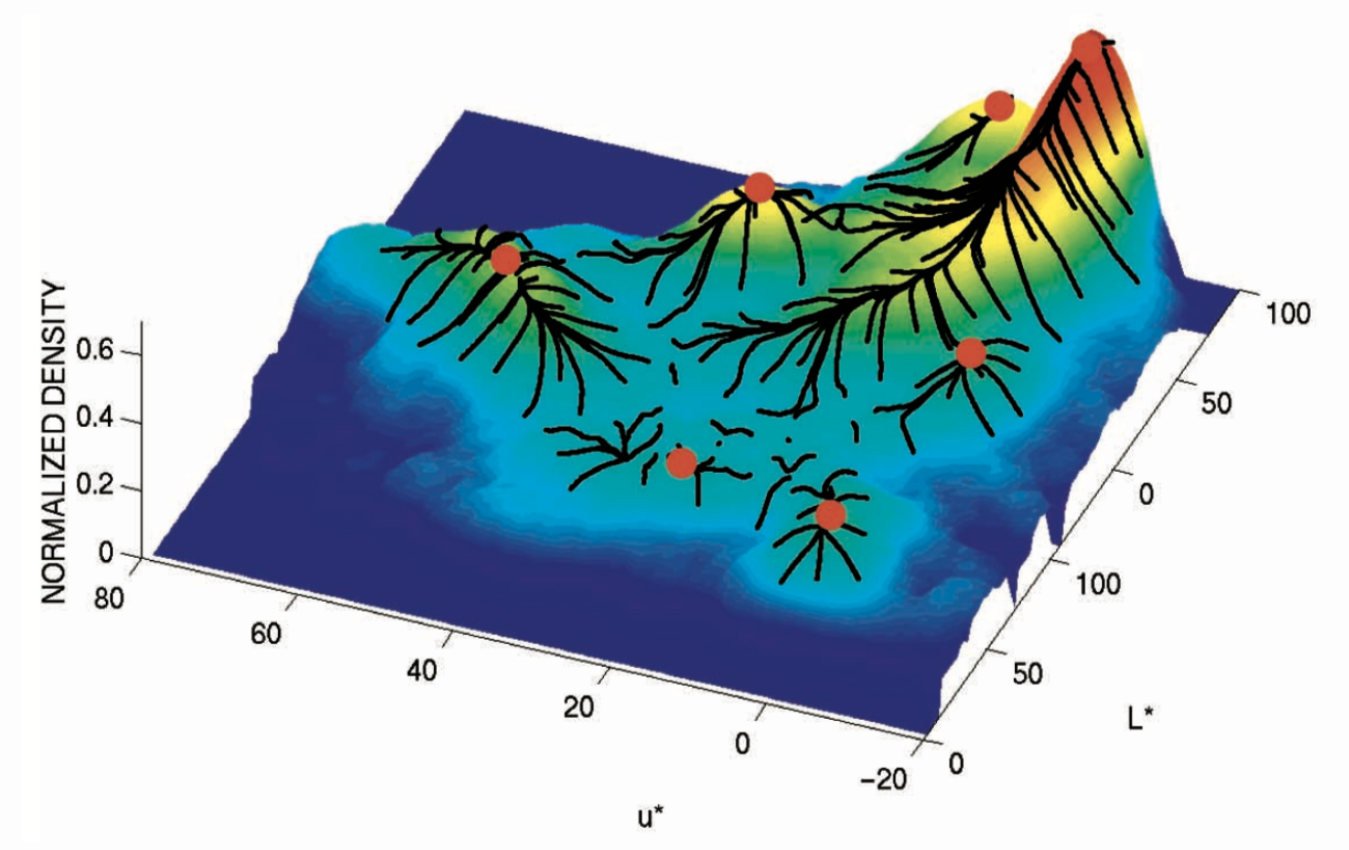
1D kernels


Uniform
Triangle
Epanechnikov
Quartic
Triweight
Gaussian
Cosine
1.0
0.8
0.6
0.4
0.2
0.0
-1.0
-0.5
0.0
0.5
1.0

Kernel density estimator (KDE)

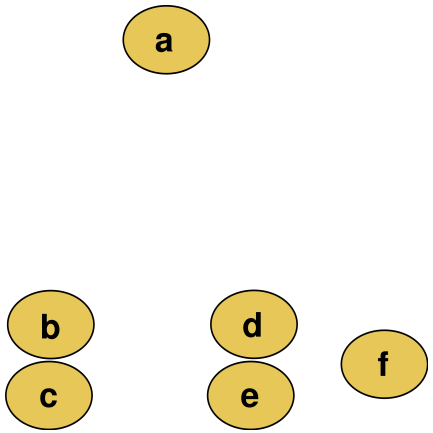


Mean shift clustering

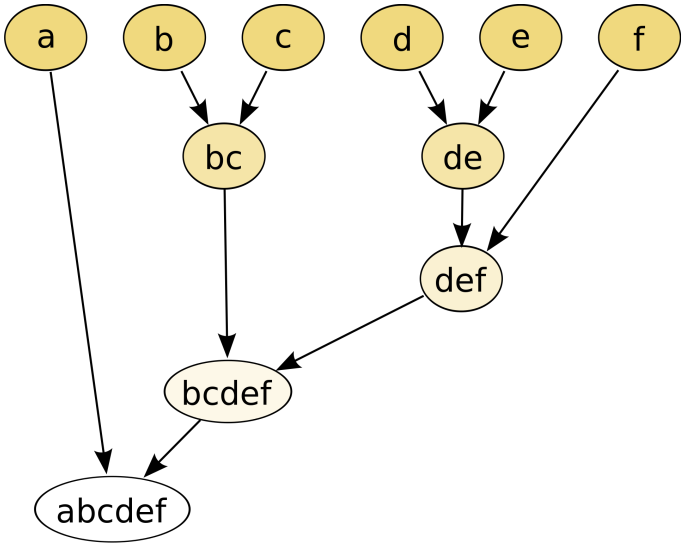


Hierarchical clustering

Data:



Cluster:

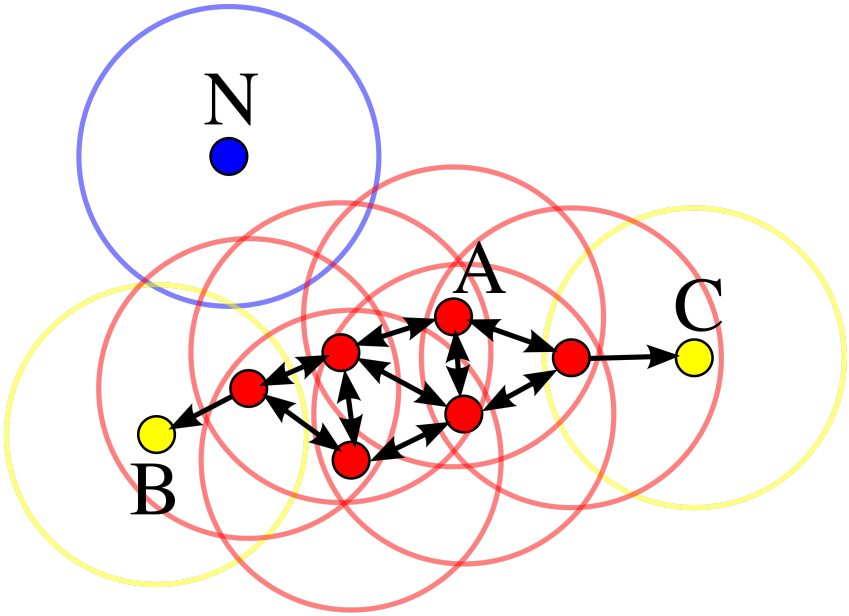


Hierarchical clustering

Distance between clusters:

- The maximum distance between elements of each cluster (also called complete-linkage clustering):
$$\max\{\, d(x,y) : x \in \mathcal{A},\, y \in \mathcal{B}\,\}.$$
- The minimum distance between elements of each cluster (also called single-linkage clustering):
$$\min\{\, d(x,y) : x \in \mathcal{A},\, y \in \mathcal{B}\,\}.$$
- The mean distance between elements of each cluster (also called average linkage clustering, used e.g. in UPGMA):
$$\frac{1}{|\mathcal{A}| \cdot |\mathcal{B}|} \sum_{x \in \mathcal{A}} \sum_{y \in \mathcal{B}} d(x,y).$$
- The sum of all intra-cluster variance.
- The increase in variance for the cluster being merged (Ward's method[7])
- The probability that candidate clusters spawn from the same distribution function (V-linkage).

DBSCAN



- parameter: k and ϵ
- core point: the ball of the radius ϵ around x contains at least k points

High-level view of clustering

Axiomatic definition of clustering (attempt)

Scale Invariance (SI) For any domain set $\mathcal{X}$, dissimilarity function d, and any $\alpha > 0$, the following should hold: $F(\mathcal{X}, d) = F(\mathcal{X}, \alpha d)$ (where $(\alpha d)(x, y) \stackrel{\text{def}}{=} \alpha\, d(x, y)$).

Richness (Ri) For any finite $\mathcal{X}$ and every partition $C = (C_1, \ldots C_k)$ of X (into nonempty subsets) there exists some dissimilarity function d over $\mathcal{X}$ such that $F(\mathcal{X}, d) = C$.

Consistency (Co) If d and d' are dissimilarity functions over $\mathcal{X}$, such that for every $x, y \in \mathcal{X}$, if x, y belong to the same cluster in $F(\mathcal{X}, d)$ then $d'(x, y) \leq d(x, y)$ and if x, y belong to different clusters in $F(\mathcal{X}, d)$ then $d'(x, y) \geq d(x, y)$, then $F(\mathcal{X}, d) = F(\mathcal{X}, d')$.

Axiomatic definition of clustering (attempt)

Consider a clustering function, F, that takes as input any finite domain X with a dissimilarity function d over its pairs and returns a partition of X

Scale Invariance (SI) For any domain set $\mathcal{X}$, dissimilarity function d, and any $\alpha > 0$, the following should hold: $F(\mathcal{X}, d) = F(\mathcal{X}, \alpha d)$ (where $(\alpha d)(x, y) \stackrel{\text{def}}{=} \alpha\, d(x, y)$).

Richness (Ri) For any finite $\mathcal{X}$ and every partition $C = (C_1, \ldots C_k)$ of X (into nonempty subsets) there exists some dissimilarity function d over $\mathcal{X}$ such that $F(\mathcal{X}, d) = C$.

Consistency (Co) If d and d' are dissimilarity functions over $\mathcal{X}$, such that for every $x, y \in \mathcal{X}$, if x, y belong to the same cluster in $F(\mathcal{X}, d)$ then $d'(x, y) \leq d(x, y)$ and if x, y belong to different clusters in $F(\mathcal{X}, d)$ then $d'(x, y) \geq d(x, y)$, then $F(\mathcal{X}, d) = F(\mathcal{X}, d')$.

Axiomatic definition of clustering (attempt)

Theorem

There exists no function, F, that satisfies all the three properties: Scale Invariance, Richness, and Consistency.

Axiomatic definition of clustering (attempt)

k-Richness: every partition of the domain into k subsets is attainable by the clustering function)

Theorem

k-Richness, Scale Invariance and Consistency all hold for the k-means clustering algorithm