Clustering based on density estimation: a proposal

Adelchi Azzalini
Università di Padova

summarizes a series of papers
joint work, partly with N. Torelli, partly with G. Menardi

October 2016

Some approaches to clustering

Clustering problem (loosely)

d characteristics are observed on each of n objects
$\rightarrow$ identify sets of homogeneous groups of objects.

Teminology: called 'unsupervised classification' in machine learning

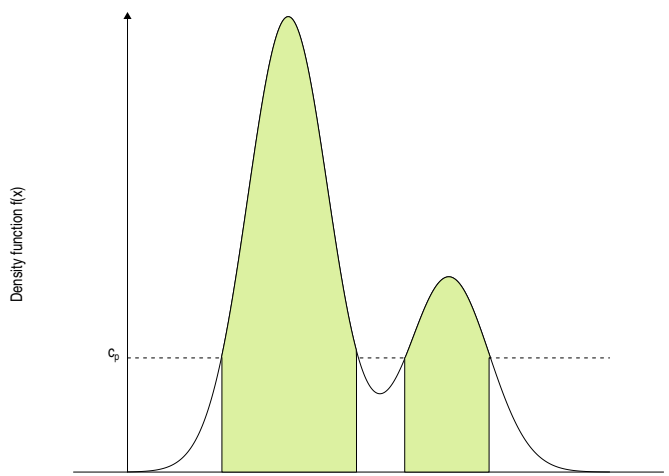
Some approaches:

- methods based on distance/dissimilarity
  (hierarchical and non-hierarchical)
- finite mixtures of parametric probability distributions,
  typically for continuous variables
- methods based on a non-parametric density estimate
  $\rightarrow$ discuss a specific proposal
  (proposals of similar logic exist, no attempt of a full discussion)

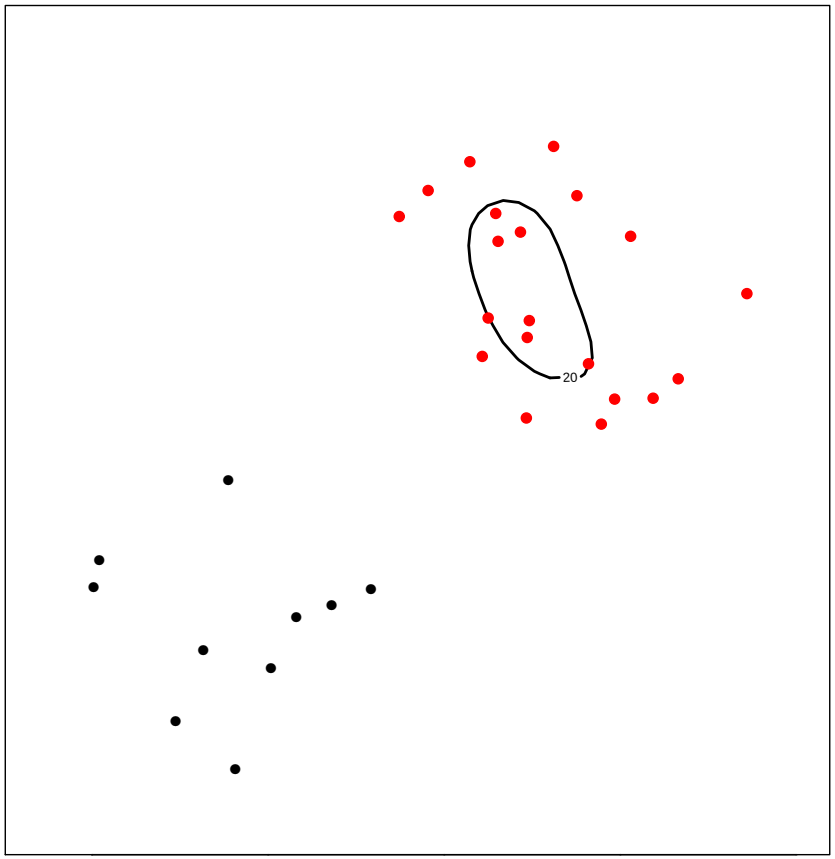
Clusters as regions of high density

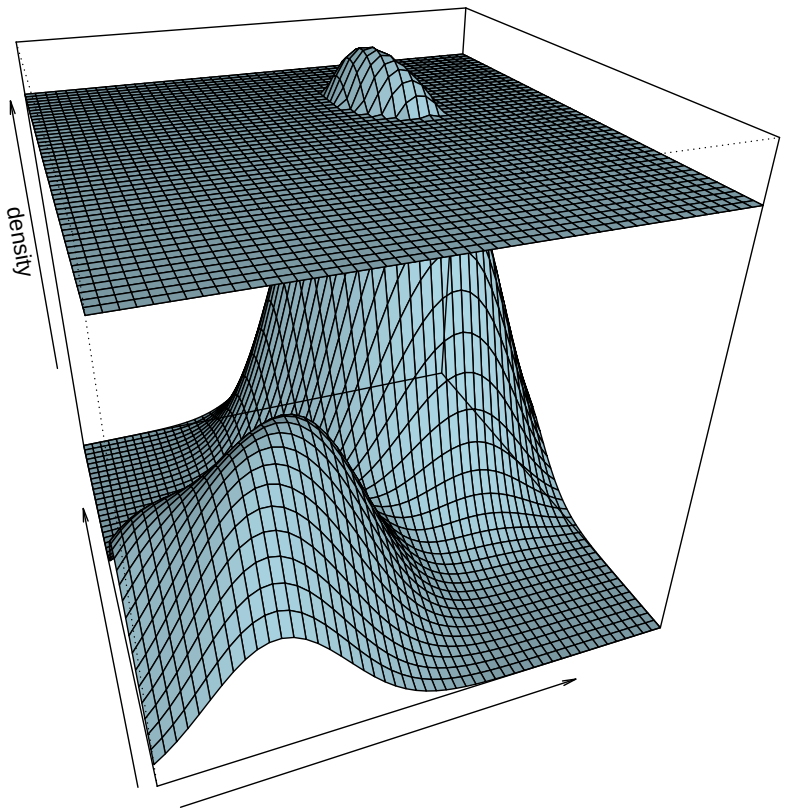
- The idea goes back to Wishart (1969) and Hartigan (1975)
- "Clusters may be thought of as regions of high density separated from [...] other regions of low density"
- "it is easy to show that such clusters form a tree"
- but the computational burden prevented its exploration

- an appealing feature is the explicitly-stated notion of cluster
  [holds for model-based approach too, not for dissimilarity-based methods]

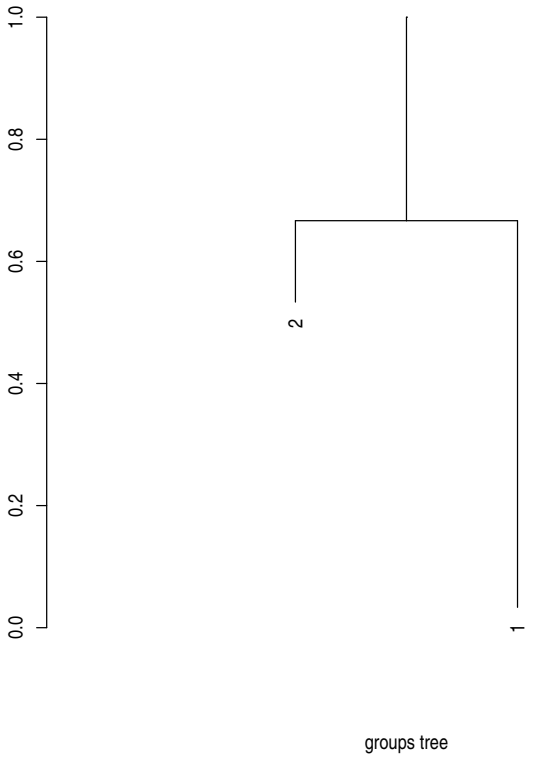


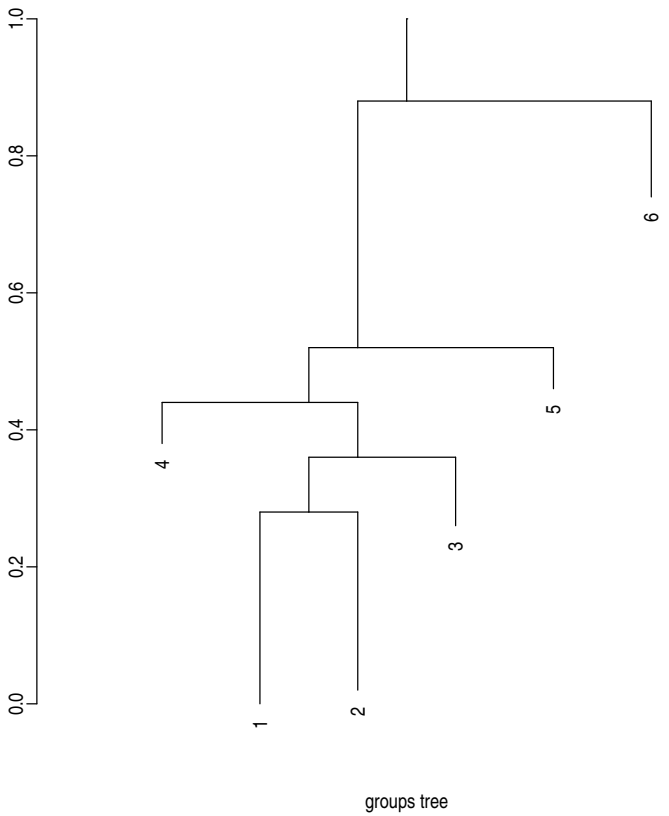
Clusters at various density levels





Clusters tree

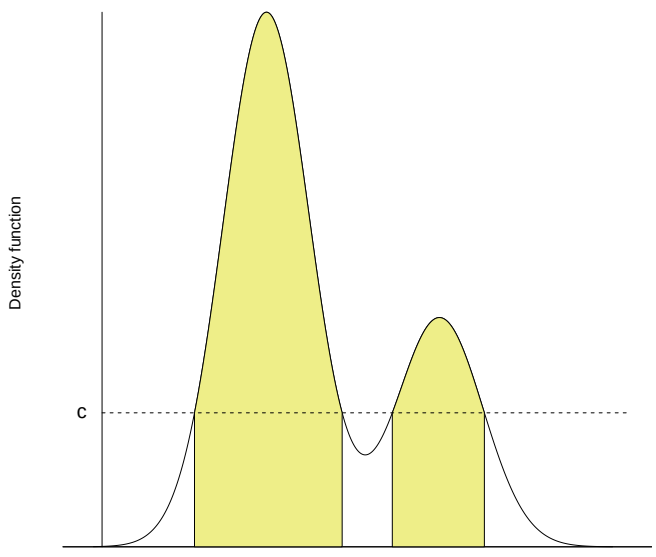


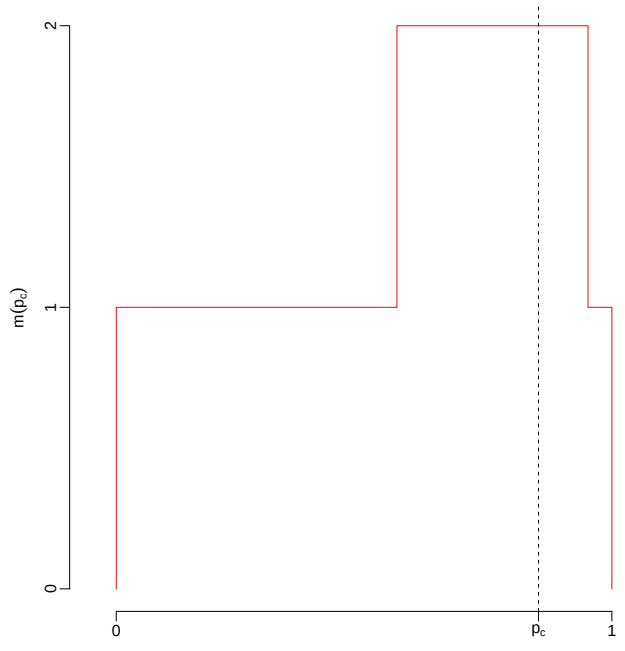


Clusters as regions of high density

$$R(c) = \{x : x \in \mathbb{R}^d, f(x) \geq c\}, \qquad \mathbb{P}\{R(c_p)\} = p$$

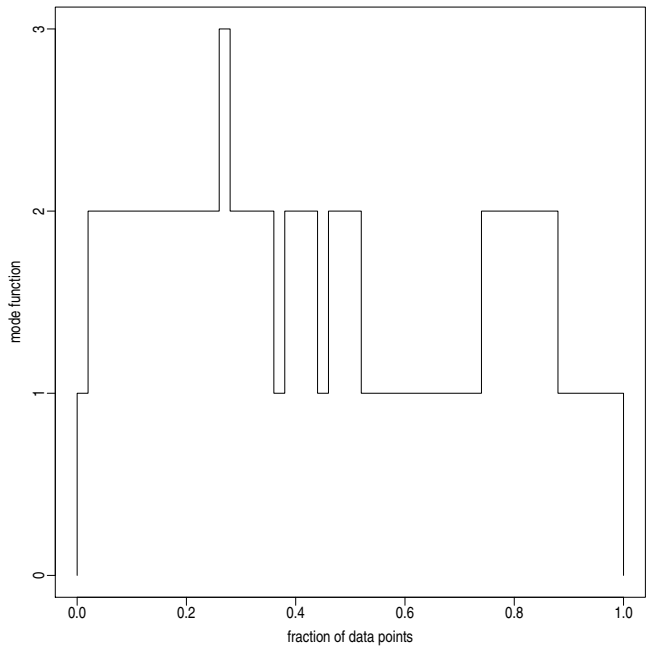
$$m(p) = \text{number of modes associated to level } p$$





Mode function $m(p)$

- as p ranges in $(0, 1)$, produce a step function $m(p)$



- put $m(0) = m(1) = 0$
- total number of increments of $m(p)$ is the number of modes of $f(x)$, it coincides with the number of decrements
- other features of $f(x)$ can be inferred from $m(p)$

Sample version of $R(c_p)$

- Observe n data points in $\mathbb{R}^d$
- Obtain $\hat{f}$ of f, e. g. by kernel method
- Estimate:

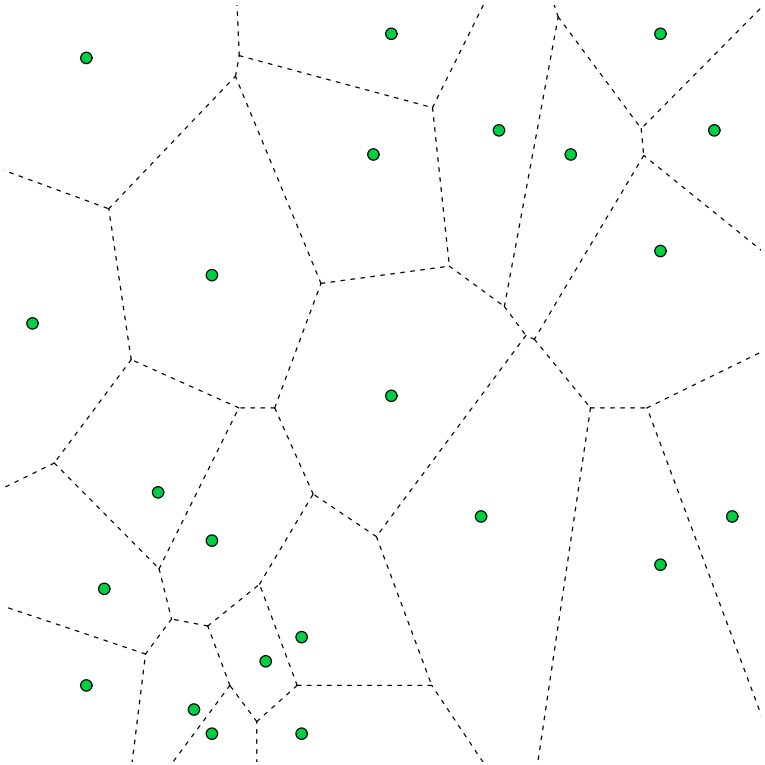
$$\hat{R}(c_p) = \{x : x \in \mathbb{R}^d, \hat{f}(x) \geq c_p\}, \qquad p \in (0, 1)$$

- 'Sample only' version:

$$S(c_p) = \{x_i : x_i \in S, \hat{f}(x_i) \geq c_p\}, \qquad p \in (0, 1)$$

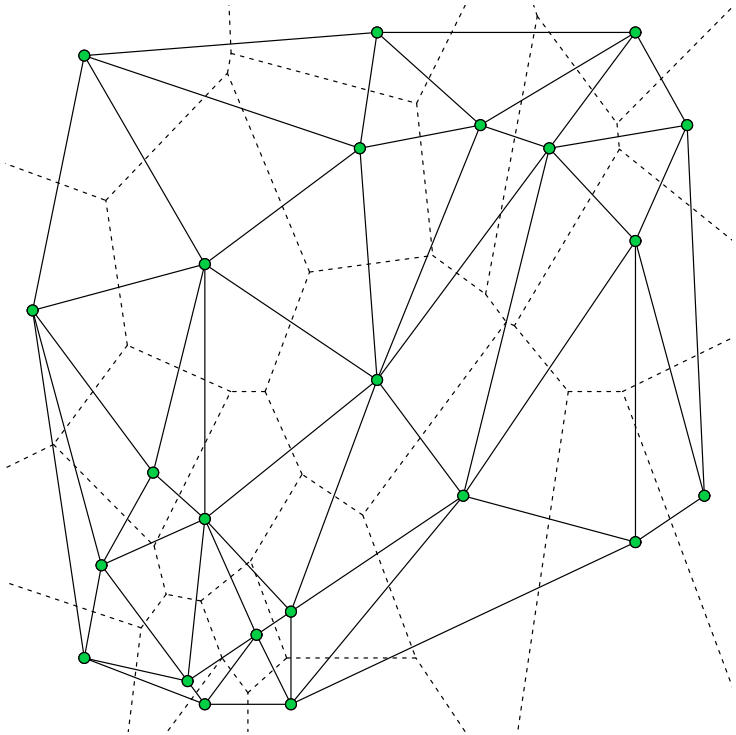
- For given p, identification of the connected components of $S(c_p)$ and of the tree structure of the modes is not trivial

How to find connected components



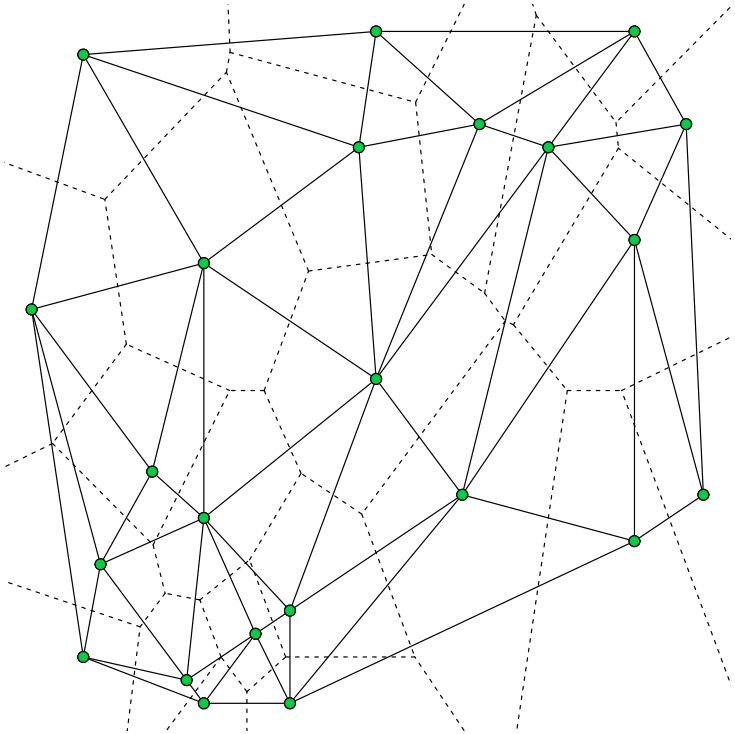
Voronoi tessellation

How to find connected components

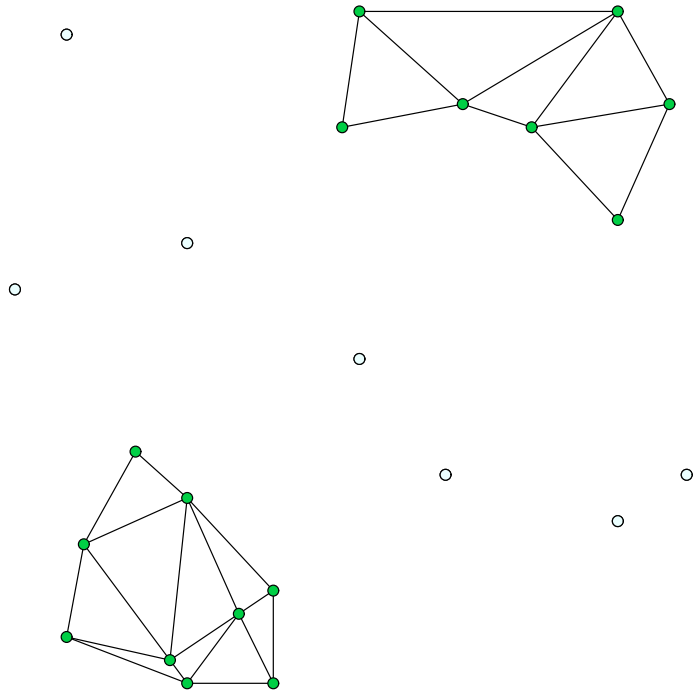


Voronoi tessellation
and Delaunay triangulation

How to find connected components



Voronoi tessellation
and Delaunay triangulation



after removing edges of points
with low density

Method

- compute $\hat{f}$
- compute Delaunay Triangulation (DT)
- for 'all' $p \in (0, 1)$
  (a) remove points of low density ($\hat{f} < c_p$) from DT
  (b) find members of connected sets
  (c) compute $m(p)$
- build tree of modes
  and form initial 'cluster cores'
- allocate remaining points

Allocation of points outside initial groupings

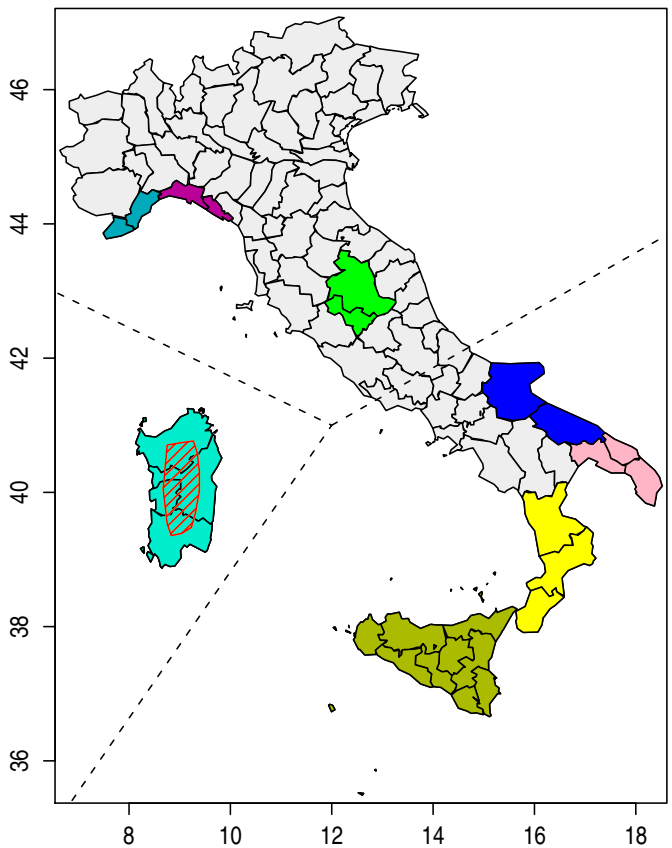
- classification problem:
  for any given x_0, allocate it among identified 'cluster cores',
  having estimated densities $\hat{f}_1, \ldots, \hat{f}_M$
- peculiar aspect: x_0 is not a randomly selected point
- allocate x_0 according to highest density ratio

$$r_j(x_0) = \frac{\hat{f}_j(x_0)}{\max_{k \neq j} \hat{f}_k(x_0)}, \qquad j = 1, \ldots, M$$

- repeat for all x_0's to be allocated
- Possible variants:
  - keep $\hat{f}_1, \ldots, \hat{f}_M$ fixed for the whole process
  - update $\hat{f}_j$'s sequentially after each point allocation
    (hence start with the x_0 having highest $\max_j r_j(x_0)$)
  - intermediate policy: update $\hat{f}_j$'s in block-sequential manner

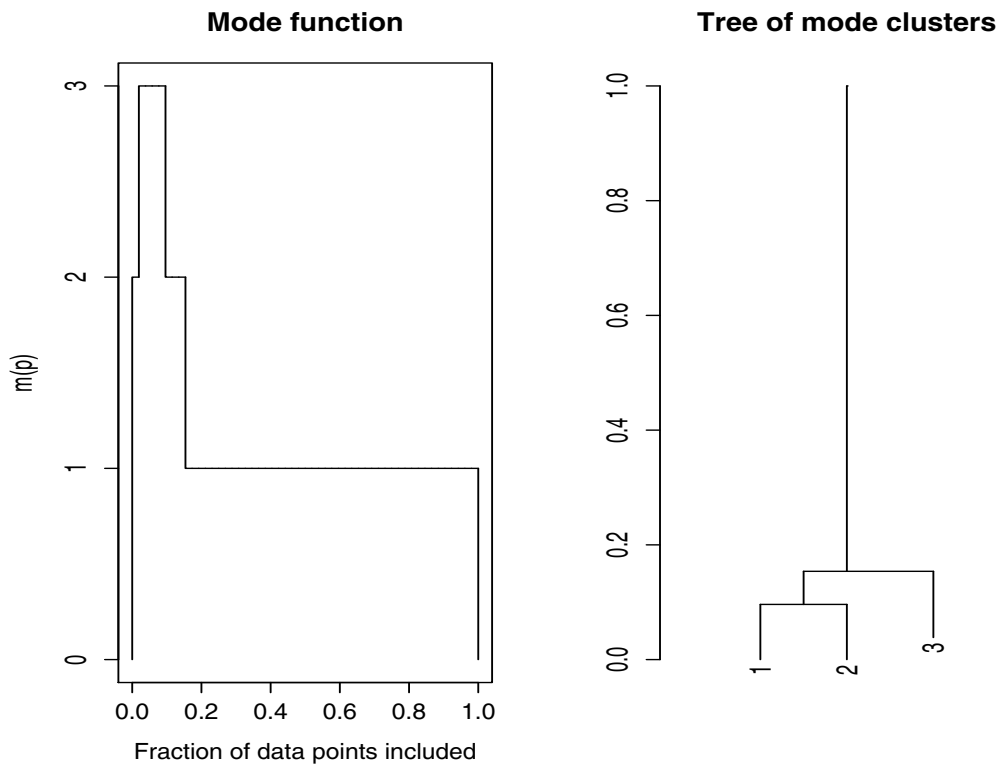
Olive oil data – I

Fatty acid composition:
8 components of olive oil
on $n = 572$ oil specimens,
from 9 areas
belonging to three macro-areas

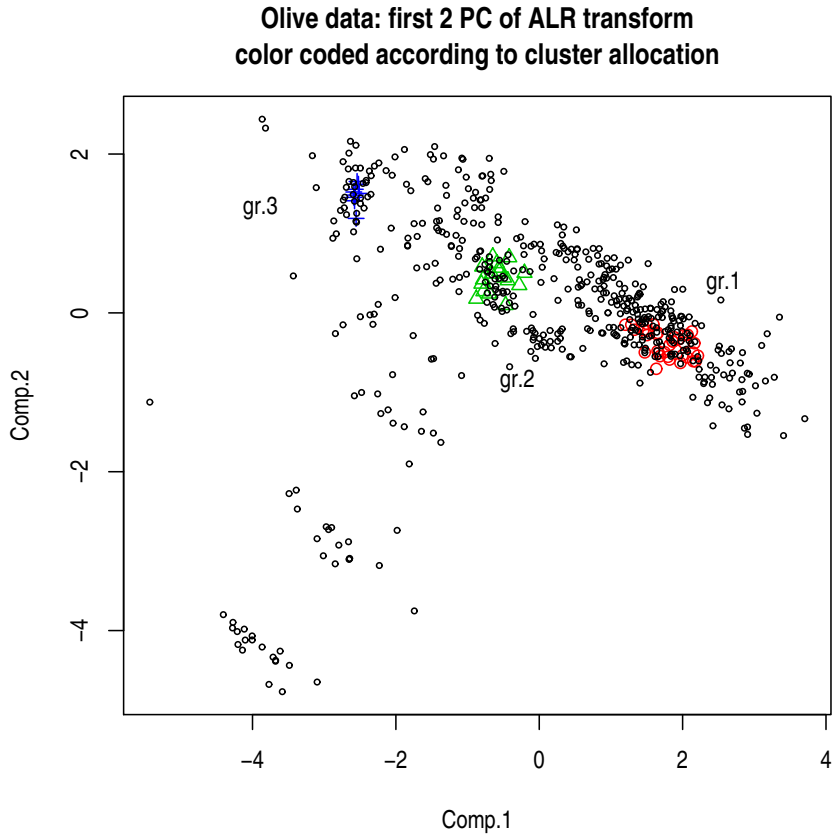


Olive oil data – II

Consider ALR transform ($x_j = \log(z_j/z_k)$, $j \neq k$), with $k = 4$
use 5 principal components (96% of total variability)
$\hat{f}$ from kernel method,



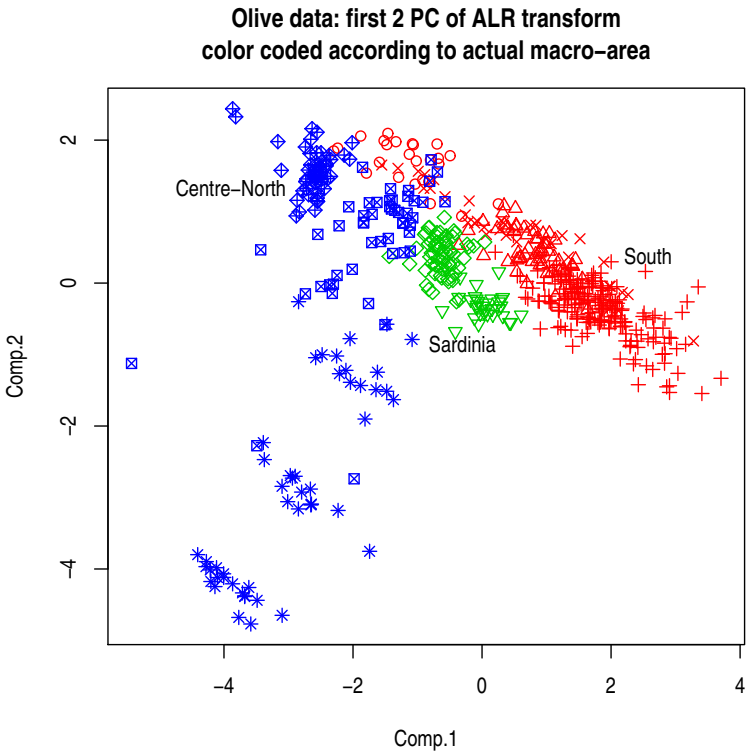
Olive oil data – III



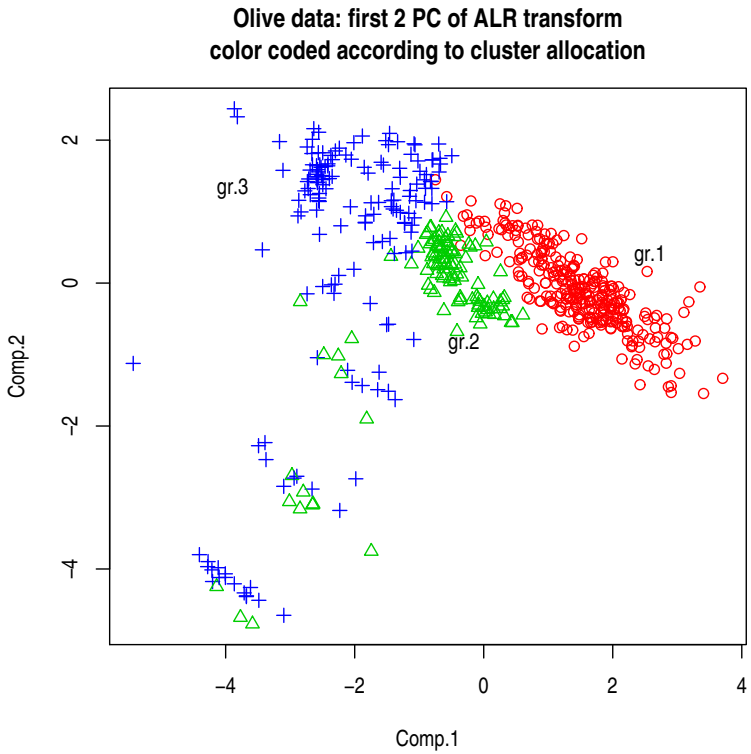
Next step: classification of unallocated points using density ratio

Olive oil data – IV

Truth



Clusters



Olive oil data – V

k-means (best out of 20)

macro.area	1	2	3
South	282	0	41
Sardinia	0	97	1
Centre-North	0	55	96

ARI=0.66

Hierarchical (complete linkage)

macro.area	1	2	3
South	107	216	0
Sardinia	98	0	0
Centre-North	88	0	63

ARI=0.28

Mclust (forcing 3 groups)

macro.area	1	2	3
South	0	323	0
Sardinia	1	0	97
Centre-North	151	0	0

ARI ≈ 1

(ARI=0.60 with 5 groups, that is, Mclust's own choice)

this method

macro.area	1	2	3
South	321	0	2
Sardinia	0	98	0
Centre-North	0	45	106

ARI=0.87

Density-based 'silhouette' diagnostics

- 'silhouette' plots are diagnostics for quality of clustering
- classical silhouette (Rousseeuw, 1987) based on distances
- in the present context, introduce a density-based silhouette
- start from posterior probability of cluster m for object x_i:

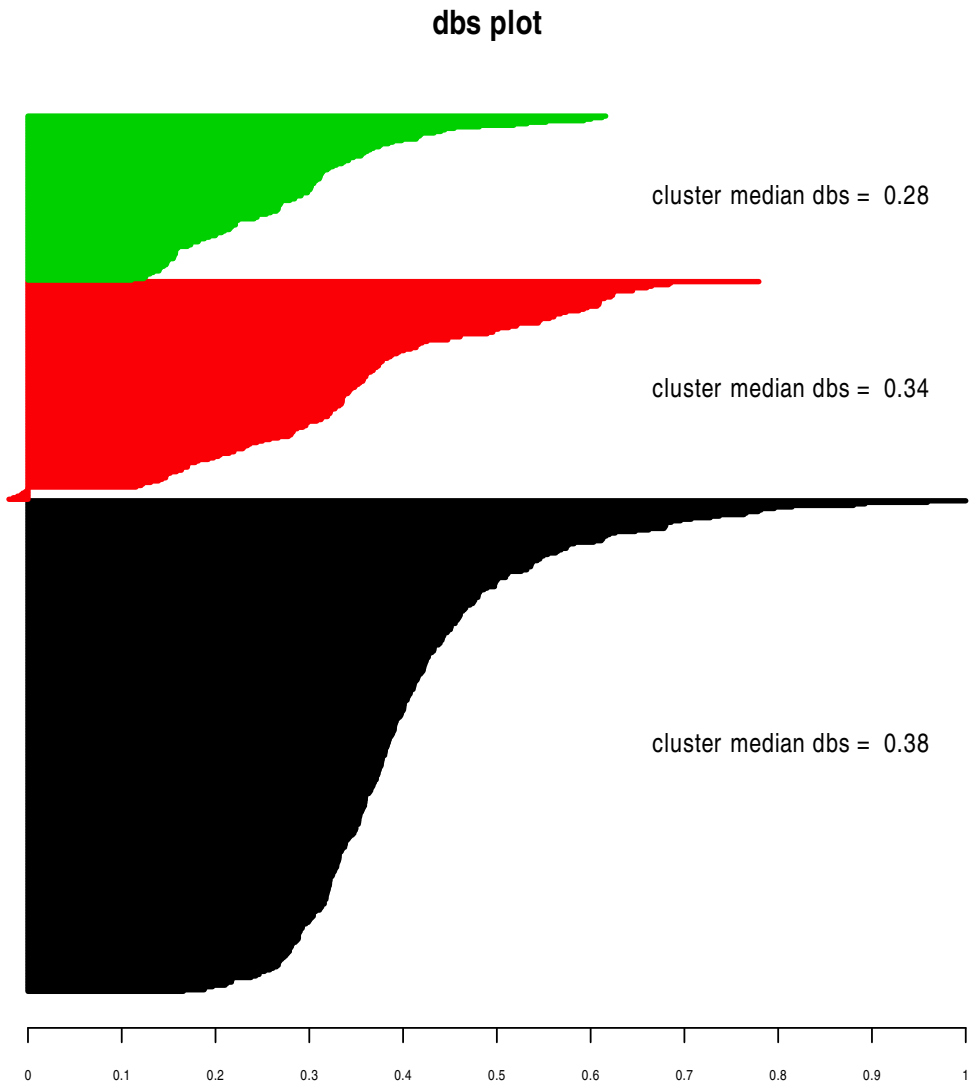
$$\pi'_m(x_i) = \frac{\pi_m f_m(x_i)}{\sum_j \pi_j f_j(x_i)}, \qquad m \in \{1, \ldots, M\}$$

- then define

$$\mathrm{dbs}_i = \frac{\log[\pi'_{m_0}(x_i)/\pi'_{m_1}(x_i)]}{\max_k |\log[\pi'_{m_0}(x_k)/\pi'_{m_1}(x_k)]|}$$

where m_0 is selected cluster for x_i and m_1 is best alternative

Olive oil data – VI: silhouette plot



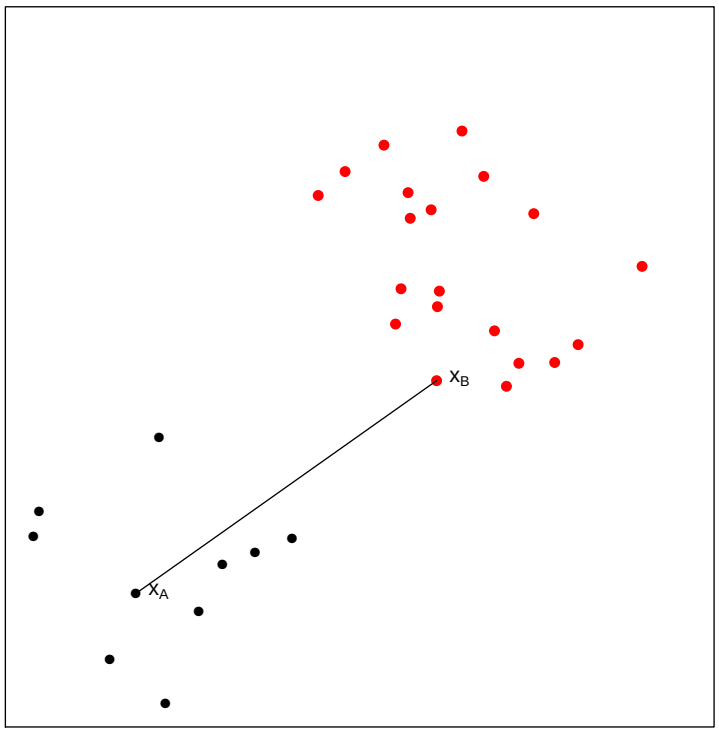
Discussion I

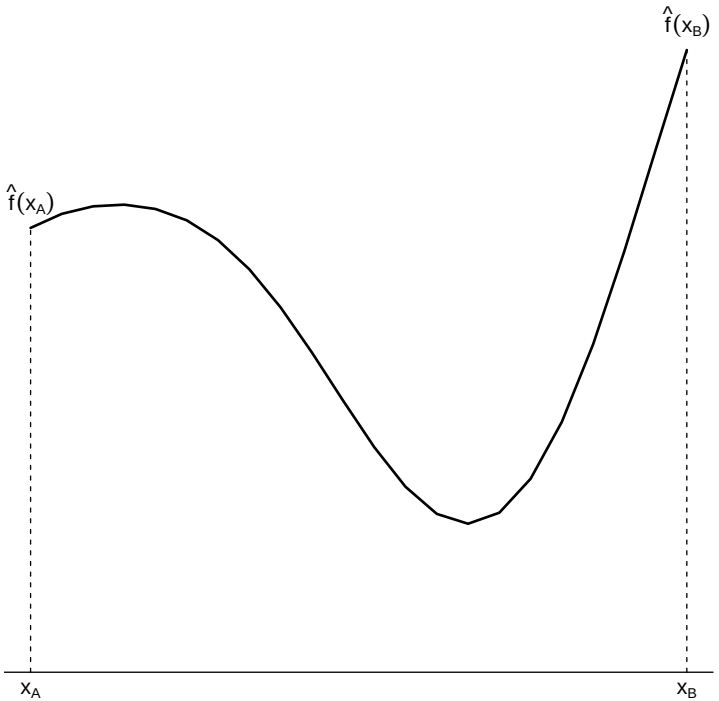
- the method is not (explicitly) based on distance
- hierarchical structure, but pruning is not required
- provides an estimate of the number of groups
- allows scoring confidence of allocation
  or non-allocation of doubtful points
- storage allocation is $n \times d$
  (instead of n^2 for dissimilarity matrix)
- since we focus on mode detection, not on fine aspects of $\hat{f}$,
  the outcome is not so sensitive to choice of the bandwidth
- computation issue for large d:
  DT time grows as $O(n^{\lfloor d/2 \rfloor})$ when $d > 3$
- 'silhouette'-type diagnostics are available

Variant method to avoid Delaunay triangulation (DT)

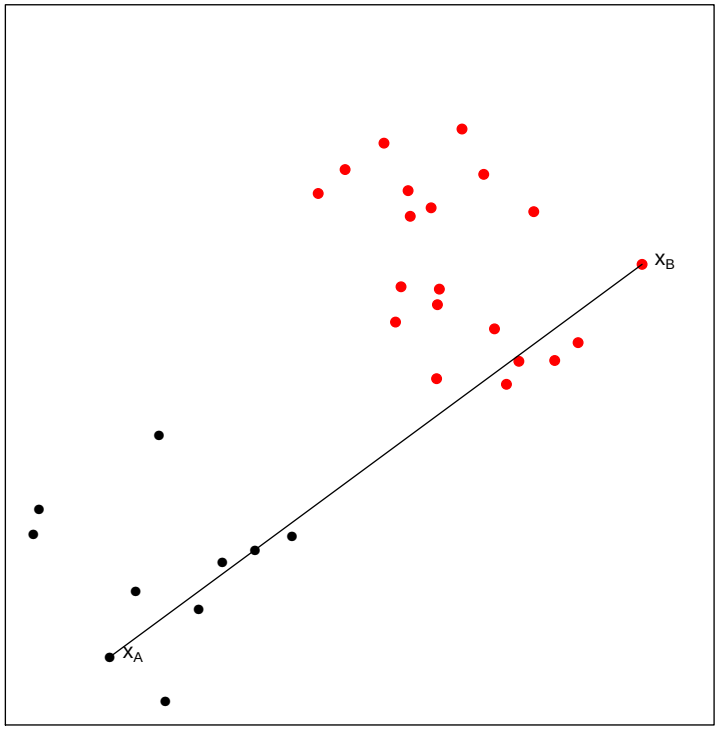
- DT step prevents to handle large d ($d > 6$, say)
- look for alternative to build sets of connected points
- idea: for any pair of points (x_i, x_j) examine $\hat{f}$ along the segment joining them
- this search is one-dimensional for any d
- the presence of a 'valley' indicates separated groups
- in principle, points must be connected if and only if there is no 'valley' between them
- in practice, allow for sampling error and discard 'small valleys'
- requires to store matrix of size $n \times n$

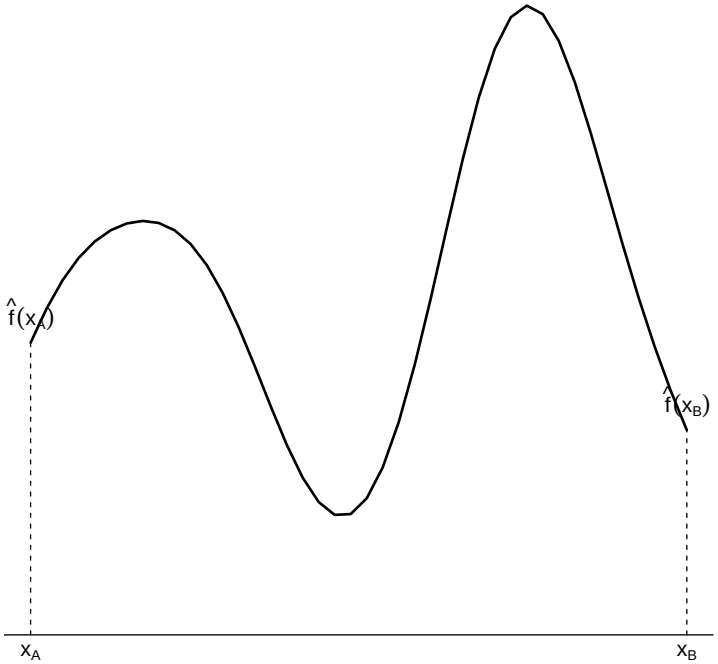
Search along segments (1)



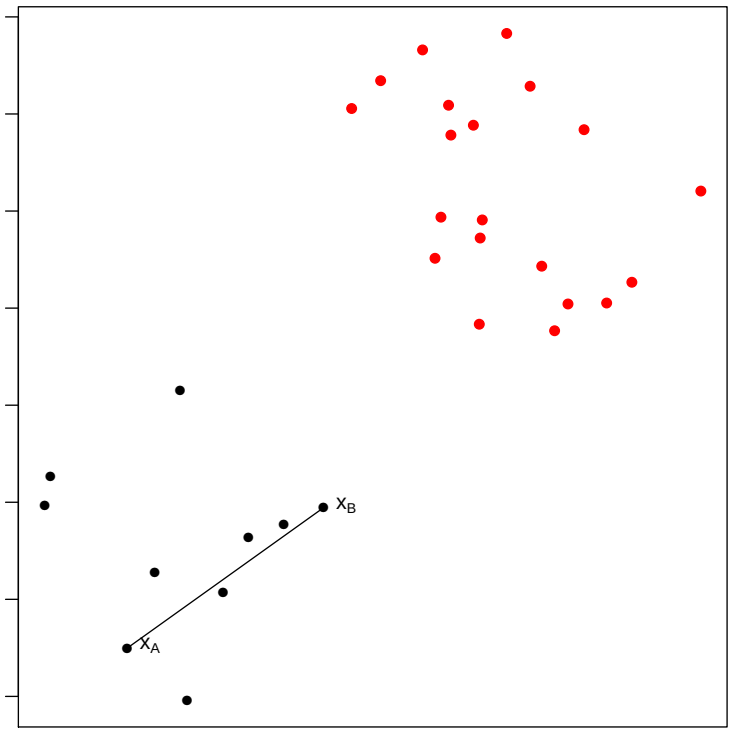


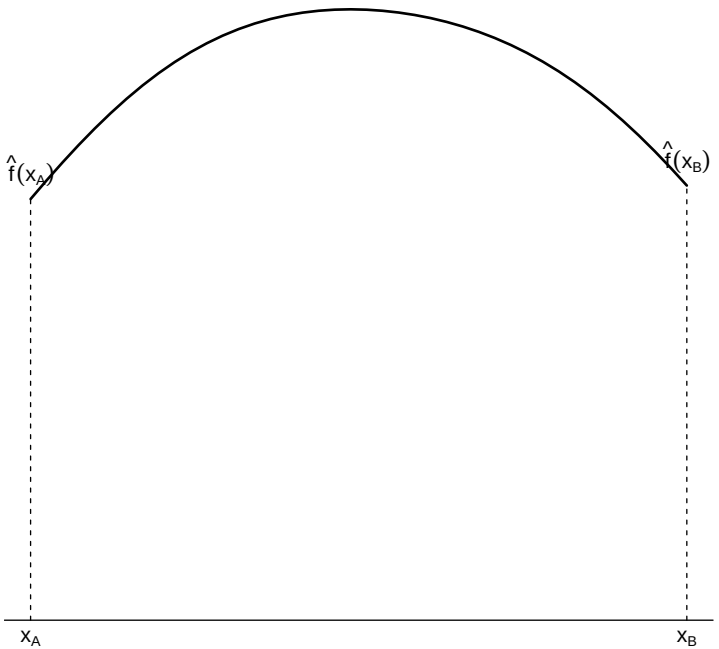
Search along segments (2)



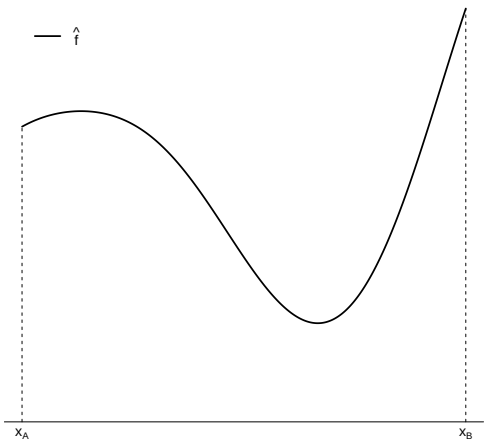


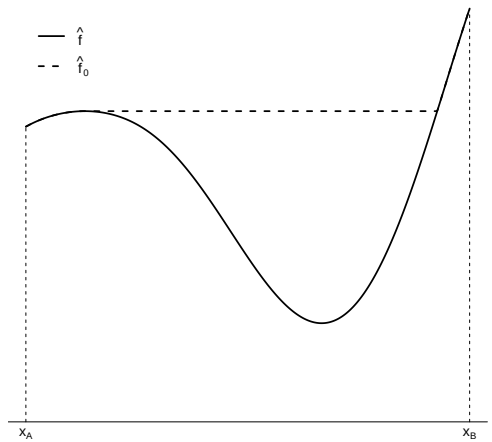
Search along segments (3)

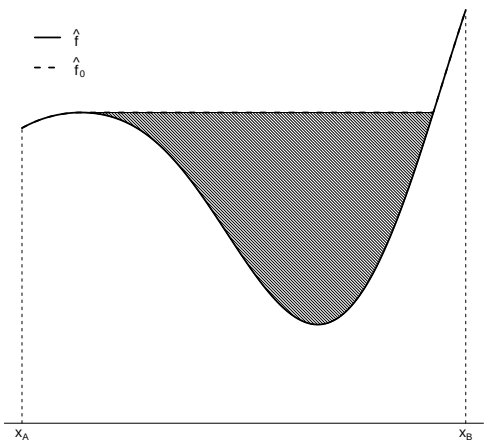


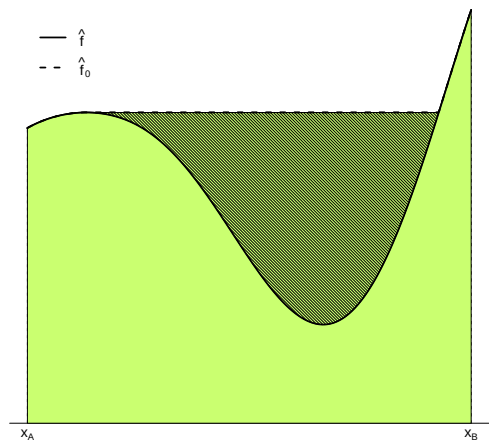


Measuring the extension of a valley









If dark-green area is small compared to light-green, ignore this valley

Dealing with non-continuous variables

- clearly categorical and discrete data cannot be handled directly
- the same problem exists for model-based clustering, in practice
- a simple practical solution:
  - obtain matrix of dissimilarity between objects
  - use multidimentional scaling to construct continuous variables
  - apply clustering method to these variables
- how many MDS variables? use background information
- useful to jitter constructed variables to avoid replicated values
- In case of mixed variables, two variants:
  - we can start from dissimilarity matrix computed on all variables
  - use dissimilarity of non-continuous variables and then merge MDS variables with original continuous variables

Discussion II: final remarks

- depending on magnitude of d, use either DT or the 'valleys' technique to build graph of connections among objects
- the method appears to work well as for quality of outcome for small to medium-sized d
- for large d, 'curse of dimensionality' is there, but not as serious as when focus is on estimation of f
- use of variable bandwidth for $\hat{f}$ helps in high dimensions but still room for improvement

Articles and software

- Azzalini & Torelli (2007), *Stat. & Comp.*, 17, 71–80
- Menardi (2011), *Stat. & Comp.*, 21, 295–308
- Menardi & Azzalini (2014), *Stat. & Comp.*, 24, 753–767
- Azzalini & Menardi (2014), *J. Stat. Software*, 57 (11)
  provides an overview of the formulation and use of software
- Azzalini (2015), chapter in *Handbook of Cluster Analysis*
  an overview more on the theoretical side
- Azzalini & Menardi (2016), *Comp. Stat.*, 31, 771–798

- R package `pdfCluster` on CRAN