BIOSTAT527

4/12/23

Lecture 6

Non-parametric clustering

Lecture IV – Non-parametric clustering

Marina Meilă
mmp@stat.washington.edu

Department of Statistics
University of Washington

soft / hard

vectors / graphs

1. Paradigms for clustering ✔ — param (K-means, EM) / NP — density estimation
2. Methods based on non-parametric density estimation ←
3. Model-based: [Dirichlet process mixture models]

Reading AoNPS Ch.: –, HTF Ch.: 14.3 Murphy Ch.: 11.[1], 11.2.1-3, 11.3, Ch 25

Clustering Paradigms

Depend on type of data, type of clustering, type of cost (probabilistic or not), and constraints (about K, shape of clusters)

- Data = vectors $\{x_i\}$ in $\mathbb{R}^d$

Parametric	Cost based [hard]
(K known)	Model based [soft]
Non-parametric	Dirichlet process mixtures [soft]
(K determined	Information bottleneck [soft]
by algorithm)	Modes of distribution [hard]
	Gaussian blurring mean shift? [hard]
	Level sets of distribution [hard]

- Data = similarities between pairs of points $[S_{ij}]_{i,j=1:n}$, $S_{ij} = S_{ji} \geq 0$

Similarity based clustering

Graph partitioning	spectral clustering [hard, K fixed, cost based]
	typical cuts [hard non-parametric, cost based]
Affinity propagation	[hard/soft non-parametric]

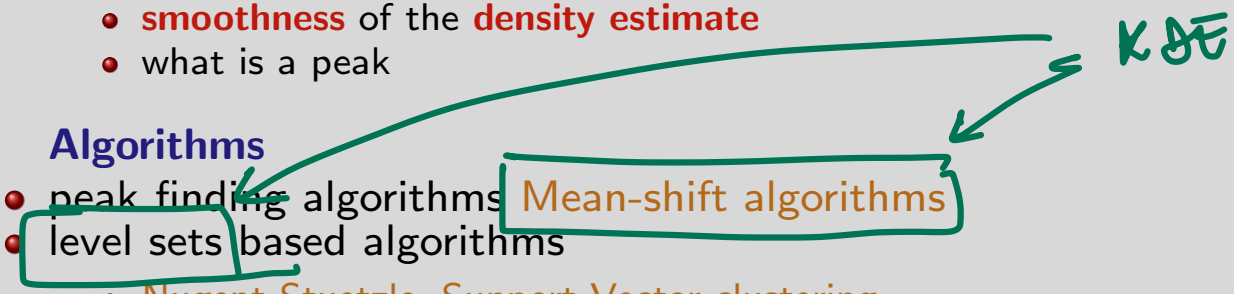
Methods based on non-parametric density estimation

Idea The clusters are the isolated peaks in the (empirical) data density

- group points by the peak they are under
- some outliers possible
- $K = 1$ possible(no clusters)
- shape and number of clusters K determined by algorithm
- **structural parameters**
  - **smoothness** of the **density estimate**
  - what is a peak

Algorithms

- peak finding algorithms
- level sets based algorithms
  - Nugent-Stuetzle, Support Vector clustering
- Information Bottleneck ?



Mean shift algorithms

$f(x) = KDE\ h$

Idea find points with $\nabla f(x) = 0$

Assume $K(z) = e^{-||z||^2/2}/\sqrt{2\pi}$ Gaussian kernel $\rightarrow$ $k'(z) = -z\,k(z)$, $z \in \mathbb{R}$

$\nabla k(z)$, $z \in \mathbb{R}^d$

$= b(z)$

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

Local max of f is solution of implicit equation

Mean Shift

$m(x) - x$

$$x = \frac{\sum_{i=1}^n x_i K(\frac{x-x_i}{h})}{\sum_{i=1}^n K(\frac{x-x_i}{h})} = \sum_i w_i x^i = m(x)$$

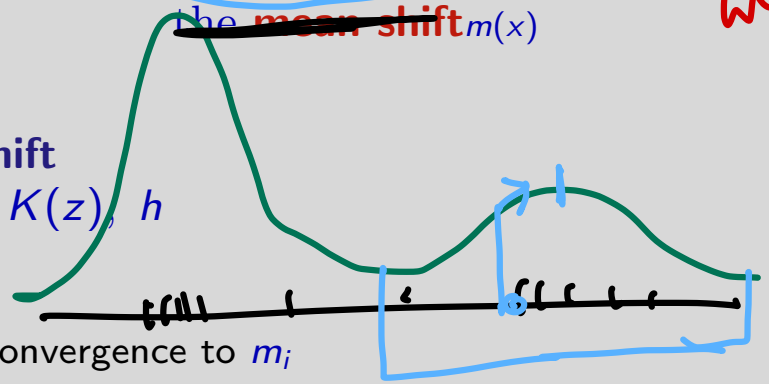
the mean shift $m(x)$

weighted mean around x = a peak μ_1

Algorithm Simple Mean Shift

Input Data $\mathcal{D} = \{x_i\}_{i=1:n}$, kernel $K(z)$, h

1. for $i = 1 : n$
   1. $x \leftarrow x_i$
   2. iterate $x \leftarrow m(x)$ until convergence to m_i
2. group points with same m_i in a cluster



C_1

Ex: $m(x) - x \propto \nabla f(x)$

Computation

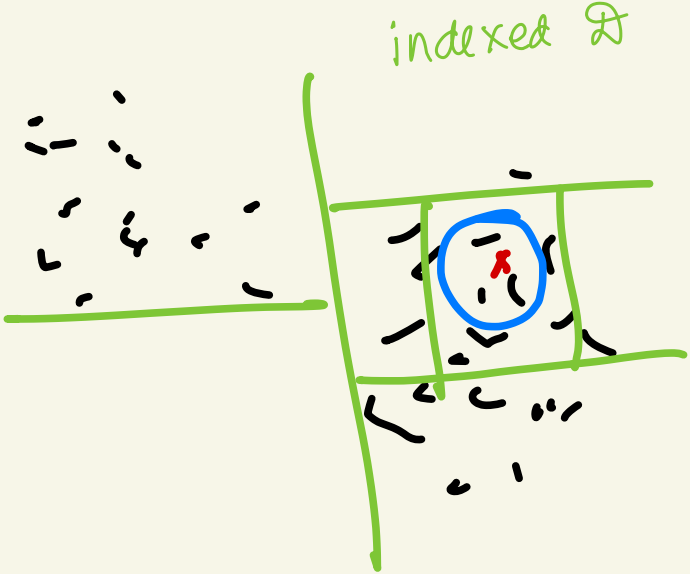
$m(x)$ n $K()$ evaluations

$|\{$neighbors of $x\}|$ $K()$ evaluations

if method to find neighbors used

x iterations to local max μ_k

x n μ_k for every $x^i \in D$



$$\nabla f(x) = 0$$

$$\Updownarrow$$

$$x \cdot \sum K\left(\frac{x - x^i}{h}\right) = \sum K\left(\frac{x - x^i}{h}\right) \cdot x^i$$

KDE $\approx f(x)$

weighted avg of x^i around x

$$x = \frac{\sum_i K(\,)\, x^i}{\sum_i K(\,)}$$

$w_i \equiv$ Nadaraya-Watson

Remarks

- mean shift iteration guaranteed to converge to a max of f
- computationally expensive
- a faster variant...

heuristic acceleration

Algorithm Mean Shift (Comaniciu-Meer)

Input Data $\mathcal{D} = \{x_i\}_{i=1:n}$, kernel $K(z)$, h

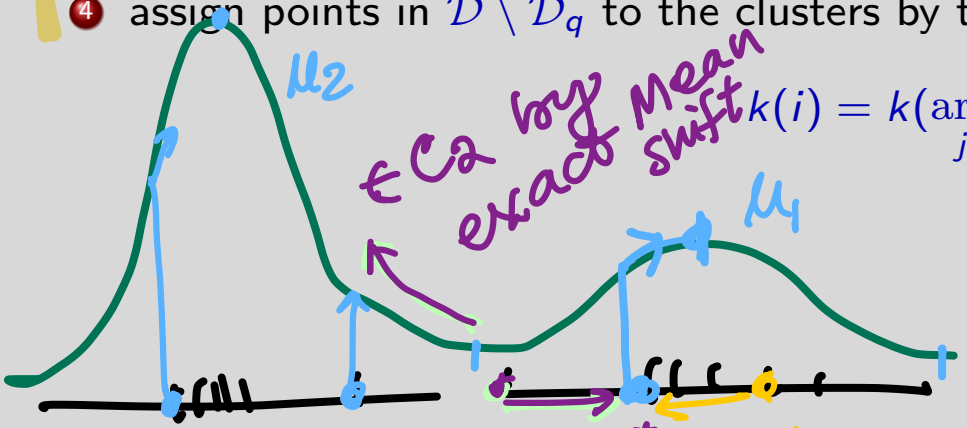
1. select q points $\{x_j\}_{j=1:q} = \mathcal{D}_q \subseteq \mathcal{D}$ that cover the data well ← land marks
2. for $j \in \mathcal{D}_q$
   1. $x \leftarrow x_j$
   2. iterate $x \leftarrow m(x)$ until convergence to m_j

Mean Shift

f(x) from all data

3. group points in $\mathcal{D}_q$ with same m_j in a cluster
4. assign points in $\mathcal{D} \setminus \mathcal{D}_q$ to the clusters by the **nearest-neighbor** method

$$k(i) = k(\underset{j \in \mathcal{D}_q}{\operatorname{argmin}} ||x_i - x_j||)$$



$x^i \notin \mathcal{D}_q$ nearest neighbor $\in \mathcal{D}_0$

$n_q = |\mathcal{D}_q|$

$n_q \times n \times$ iteration

KDE

$(n - n_q) \times n_q$

Furthest points heuristic (Anchor algorithm)

Given $\mathcal{D} = \{x^1, \dots x^n\}$

Want $\mathcal{D}' \subset \mathcal{D}$ "that covers $\mathcal{D}$ well"

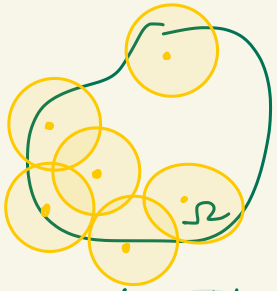
$n' = |\mathcal{D}'|$ $\quad \mu^{1:n'}$

$\mu_1 \sim \text{uniform}(\mathcal{D})$

for $j = 2, \dots n'$

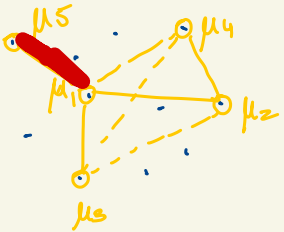
$$\mu_j = \underset{x^i \notin \mathcal{D}'}{\text{argmax}} \left(\min_{i'=1:j-1} \|x^i - \mu_{i'}\| \right)$$

furthest closest neighbor in $\mathcal{D}'$



ε-net covers Ω

$\Rightarrow \forall x \in \Omega$ has $\|x - \mu_i\| < \varepsilon$ for some μ_i



Problems

- selects outliers !!
- $n' = ?$

[Supplement: Gaussian blurring mean shift]

Idea

- like **Simple Mean Shift** but points are shifted to new locations
- the density estimate f changes
- becomes concentrated around peaks very fast

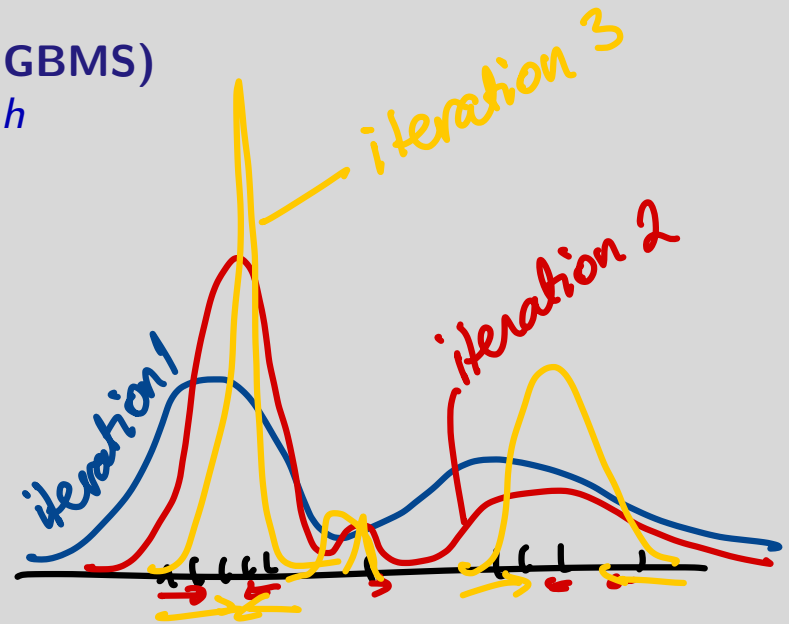
Algorithm Gaussian Blurring Mean Shift (GBMS)

Input Data $\mathcal{D} = \{x_i\}_{i=1:n}$, Gaussian kernel $K(z)$, h

1. Iterate until STOP
   1. for $i = 1 : n$ compute $m(x_i)$
   2. for $i = 1 : n$, $x_i \leftarrow m(x_i)$

Remarks

- all x_i converge to a single point
  $\Rightarrow$ need to stop before convergence



- very fast
- — " — heuristic
- clustering alg

Empirical stopping criterion ?

- define $e_i^t = ||x_i^t - x_i^{t-1}||$ the change in x_i at t
- define $H(e^t)$ the **entropy** of the **histogram** of $\{e_i^t\}$
- STOP when $\sum_{i=1}^n e_i^t/n <$`tol` OR $|H(e^t) - H(e^{t-1})| <$`tol'`

Convergence rate If true f Gaussian, convergence is **cubic**

$$||x_i^t - x^*|| \leq C||x_i^{t-1} - x^*||^3$$

very fast!!

The Nugent-Stuetzle algorithm

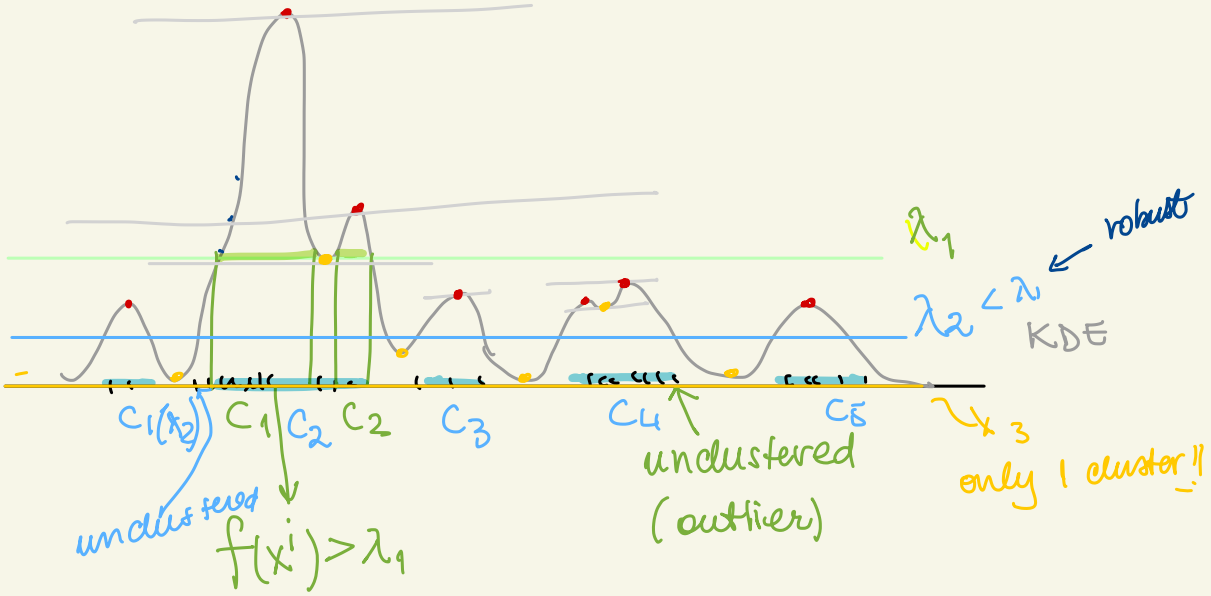
Algorithm Nugent-Stuetzle

Input Data $\mathcal{D} = \{x_i\}_{i=1:n}$, kernel $K(z)$

1. Compute KDE $f(x)$ for chosen h
2. for levels $0 < l_1 < l_2 < \ldots < l_r < \ldots < l_R \geq \sup_x f(x)$
   1. find **level set** $L_r = \{x \,|\, f(x) \geq l_r\}$ of f
   2. if L_r **disconnected** then each connected component is a cluster $\rightarrow (C_{r,1}, C_{r,2}, \ldots C_{r,K_r})$

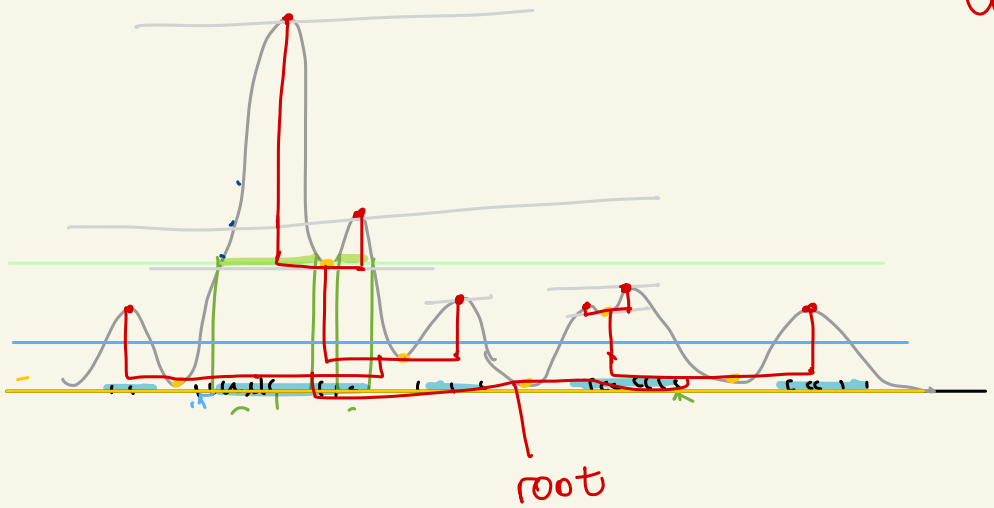
Output clusters $\{(C_{r,1}, C_{r,2}, \ldots C_{r,K_r})\}_{r=1:R}$

1. Compute $f(x^i)$ $i=1:n$ n^2
2. ...



- How to choose λ ? — not too many unclustered
  — robust $\lambda \pm \Delta\lambda \approx$ same clusters

Cluster tree



Remarks

- every cluster $C_{r,k} \subseteq$ some cluster $C_{r-1,k'}$
- therefore output is hierarchical clustering
- some levels can be pruned (if no change, i.e. $K_r = K_{r-1}$)
- algorithm can be made recursive, i.e. efficient
- finding level sets of f tractable only for $d = 1, 2$
- for larger d, $L_r = \{x_i \in \mathcal{D} \,|\, f(x_i) \geq l_r\}$
- to find connected components
  - for $i \neq j \in L_r$
    if $f(tx_i + (1-t)x_j) \geq l_r$ for $t \in [0, 1]$
    then $k(i) = k(j)$
- confidence intervals possile by resampling