CSE 547/
STAT 548

4/11/25

Lecture 4

KDE - selecting h
NP clustering
Mean Shift

Project info
HW 1 posted

Lecture II – Clustering – Part II: Non-parametric clustering

Marina Meilă
mmp@stat.washington.edu

Department of Statistics
University of Washington

CSE 547/STAT 548
Winter 2022

1. Paradigms for clustering
2. Methods based on non-parametric density estimation
3. Model-based: Dirichlet process mixture models

Kernel density estimation

- h = smoothing parameter
- $h \nearrow$ Bias $\uparrow$, Variance $\searrow$

Input
- data $\mathcal{D} \subseteq \mathbb{R}^d$
- **Kernel** function $K(z)$
- parameter **kernel width** h (is a **smoothness parameter**)

Output $f(x)$ a **probability density** over $\mathbb{R}^d$

"parameters"

fction call $\rightarrow$
$$f(x) = \boxed{\frac{1}{nh^d}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)}$$

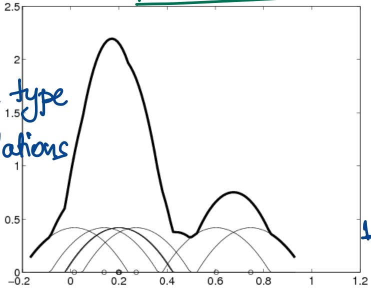
- $h, K()$ not learned from data
- $\{x_{1:n}\}$ grows with n

KDE

Train: store $x_{1:n}$, h, kernel type

"Testing": all calculations

Use



h : Model Selection

1. Choose "grid" of h values

Ex: $\{0.05, 0.25, 0.5, 1, 2, \dots\}$ logarithmic grid

2. CV

- f is sum of Gaussians centered on each x_i
- f is smoother (less variation) if h larger
- **caveat:** dimension d can't be too large

CV the idea

Given $\mathcal{D}=\{x_{1:n}\}$, $K()$

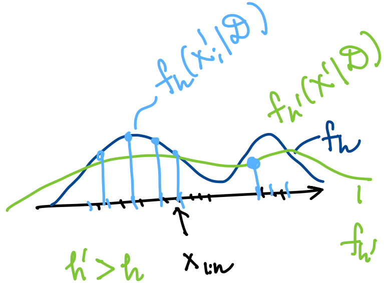
$H=\{h_1,h_2,\dots h_k\}$ set of h values ("grid")

1. get sample $\mathcal{D}'=\{x'_{1:n'}\}$ ~ same P
2. Compute validation likelihood

   for $h\in H$

   $$\ell^V(h)=\ln f_h(\mathcal{D}'|\mathcal{D})$$

3. Choose $h^*=\underset{h\in H}{\arg\max}\ \ell^V(h)$



The kernel function

- Example $K(z) = \frac{1}{(2\pi)^{d/2}} e^{-||z||^2/2}$, $z \in \mathbb{R}^d$ is the Gaussian kernel
- In general
  - $K()$ should represent a density on $\mathbb{R}^d$, i.e $K(z) \geq 0$ for all z and $\int K(z)dz = 1$
  - $K()$ symmetric around 0, decreasing with $||z||$
- In our case, K must be differentiable

I Mean Shift Clustering

(II Level Set Clustering)

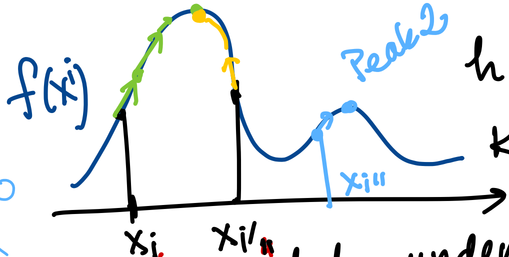
Peak(x_i) = Peak 1

(I) Idea: the mean-shift

$$f(x) = \frac{1}{n h^d} \sum_{i=1}^{n} e^{-\frac{\|x - x_i\|^2}{2h^2}}$$

$$\nabla f(x) = \frac{1}{\cancel{n h^d (2\pi)^{d/2}}} \sum_{i=1}^{n} \cancel{-} \frac{\cancel{2}(x - x_i)}{\cancel{2} h^2} \cdot \underbrace{e^{-\frac{\|x - x_i\|^2}{2h^2}}}_{K(\ )} = 0$$

peak



h fixed

KDE = $f(x)$

Cluster = "data under peak"

$x_i \in$ Cluster 1

$x_{i''} \in$ Cluster 2

$$\sum_{i=1}^{n} (x - x_i) K_h(x - x_i) = 0$$

$$\sum_{i=1}^{n} x K_h(x - x_i) = \sum_{i=1}^{n} x_i K_h(x - x_i)$$

$$x \cdot \underbrace{\boxed{\sum_{i=1}^{n} K_h(\ )}}_{Z(x)} = \sum_{i=1}^{n} x_i K_h(\ )$$

$$\boxed{x} = \sum_{i=1}^{n} \underbrace{\frac{K_h(x - x_i)}{Z(x)}}_{w_i(x) > 0} \cdot x_i = \boxed{\sum_{i=1}^{n} w_i x_i} = m(x)$$

weighted sum of $x_{1:n}$

when $\nabla f(x) = 0$

$$\sum_{i=1}^{n} w_i = 1$$

Mean-shift $\quad m(x) - x = 0 \Leftrightarrow \nabla f(x) = 0$

Alg: iterate $x \leftarrow m(x)$ until convergence

Mean shift algorithms

Idea find points with $\nabla f(x) = 0$
Assume $K(z) = e^{-||z||^2/2}/\sqrt{2\pi}$ Gaussian kernel

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

Local max of f is solution of implicit equation

$$x = \underbrace{\frac{\sum_{i=1}^n x_i K(\frac{x-x_i}{h})}{\sum_{i=1}^n K(\frac{x-x_i}{h})}}_{\text{the } \textbf{mean shift} \, m(x)}$$

Algorithm Simple Mean Shift

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

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

Alg properties

1. #peaks K depends on h

2. Runtime $\boxed{\sum_{i=1}^{n}} \underbrace{\frac{K_h(x-x_i)}{Z(x)}}_{w_i(x)>0} \cdot x_i$ × each step until convergence × all $i=1:n$ $\Theta(n^2)$

$\Theta(n)$

$\sum_{i \in \text{Neighbors}(x)}$

1. subsample $n' << n$
   $\Rightarrow$ find all peaks $m_{1:K}$

2. assign remaining to nearest m_k

Remarks

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

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
2. for $j \in \mathcal{D}_q$
   1. $x \leftarrow x_j$
   2. iterate $x \leftarrow m(x)$ until convergence to m_j
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

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