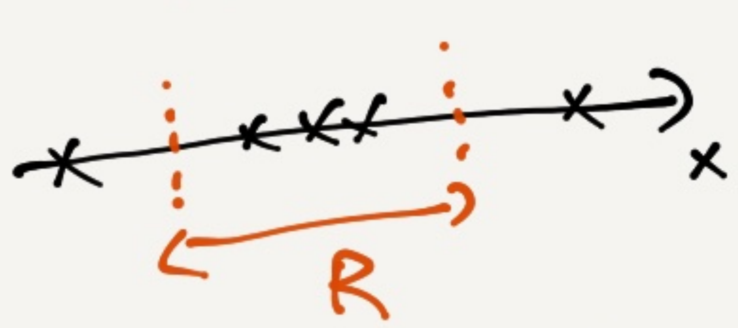
Non-parametric Density Estimation

- So far we have looked at parametric models
  $\Rightarrow$ make assumptions about the form of the density (Gaussian, Expo, GMM, etc)
- Non parametric Estimation – estimate $p(x)$ w/o assuming a form (has some parameters)

Histogram

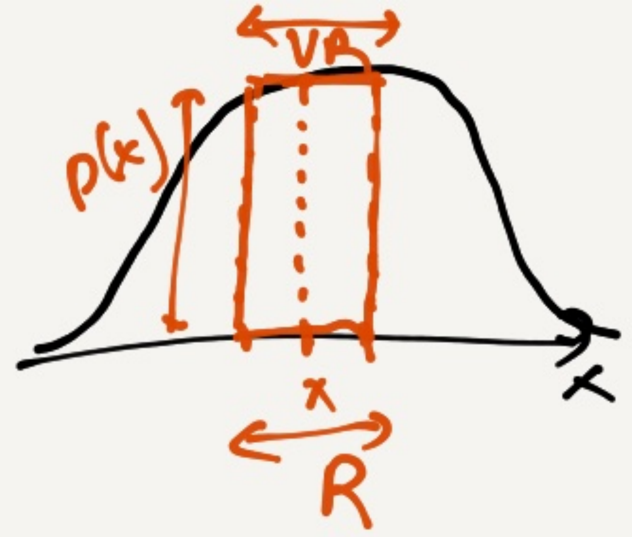
- Samples $\{x_1, \ldots, x_N\} = D$
- consider a region R
  - define $P = p(x \in R) = \int_R p(x)dx$
  - define K_R = # of points in D that are inside R.
  - ML estimate of P:
    $\hat{P} = \frac{K_R}{N}$

$K_R = 3$, $\hat{P} = 3/5$

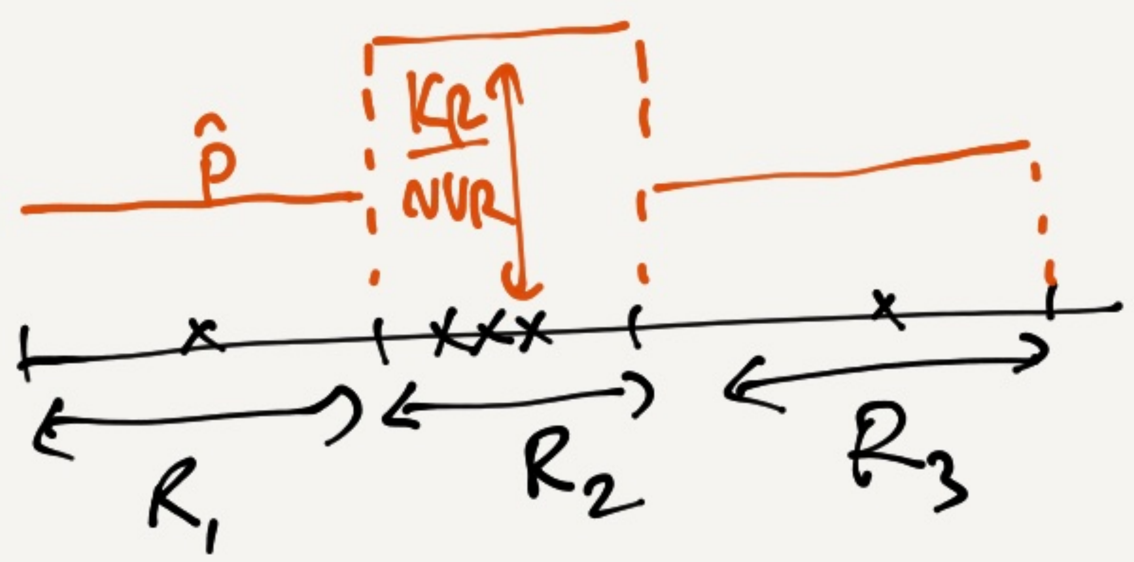


(this is just a bin in a histogram)

- Assume R is small enough, then
  $\hat{P} \approx p(x) V_R \approx \int_R p(x)dx$
  V_R: volume of R



- Solve for $p(x)$:
  $$\hat{p}(x) = \frac{\hat{P}}{V_R} = \frac{K_R}{N V_R}$$
  K_R: # points in R
  V_R: volume of R



We can extend this simple histogram:

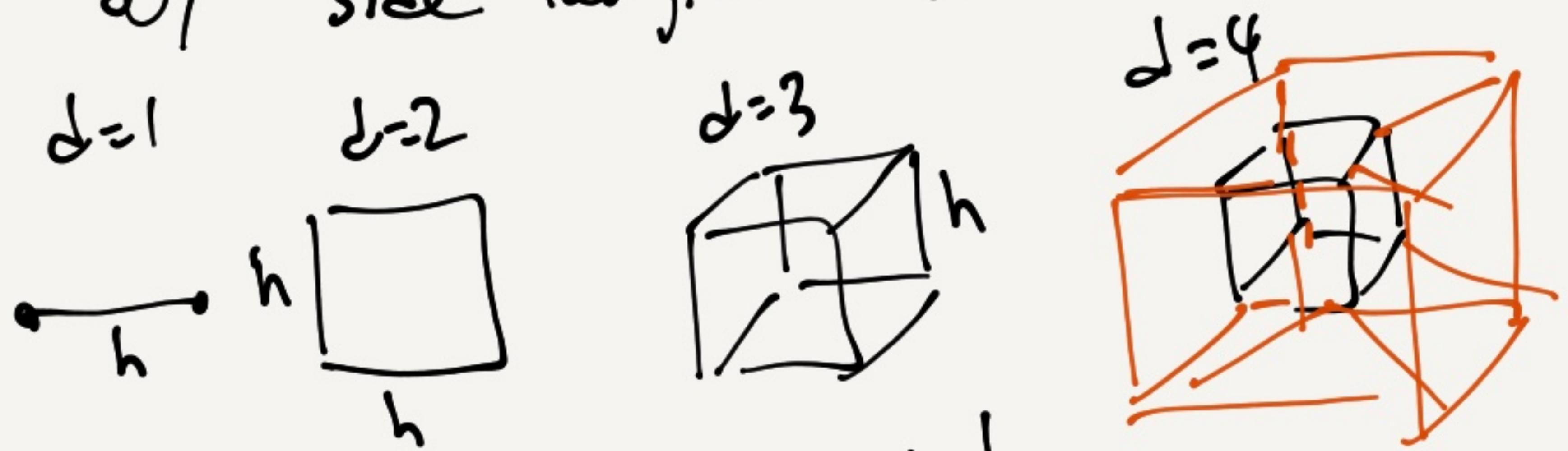
How to choose R?

1) Keep V_R fixed, & let K_R vary. ✓✓
   $\Rightarrow$ Kernel density estimator (KDE); Parzen windows
2) Keep K_R fixed, let V_R vary.
   $\Rightarrow$ k-NN estimator (very bad) PRML

CS5487 Lecture Notes (2020)
Dr. Antoni B. Chan
Dept of Computer Science
City University of Hong Kong

Kernel Density Estimation (KDE) – Parzen window – Parzen Estimators

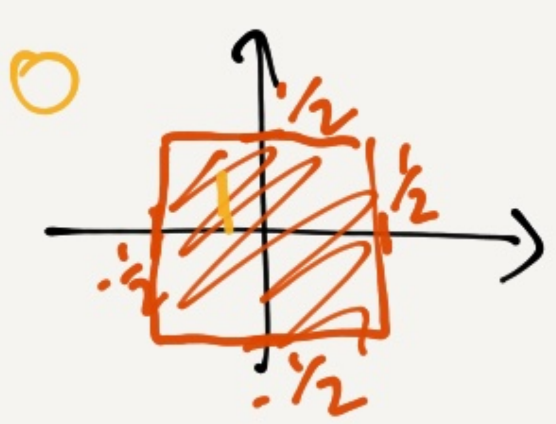
- let R be a d-dim hypercube w/ side length h.

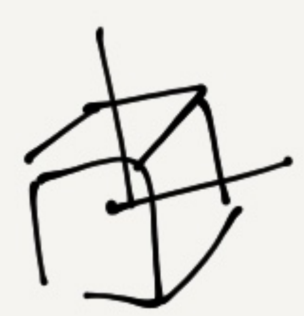


volume of hypercube $= h^d$

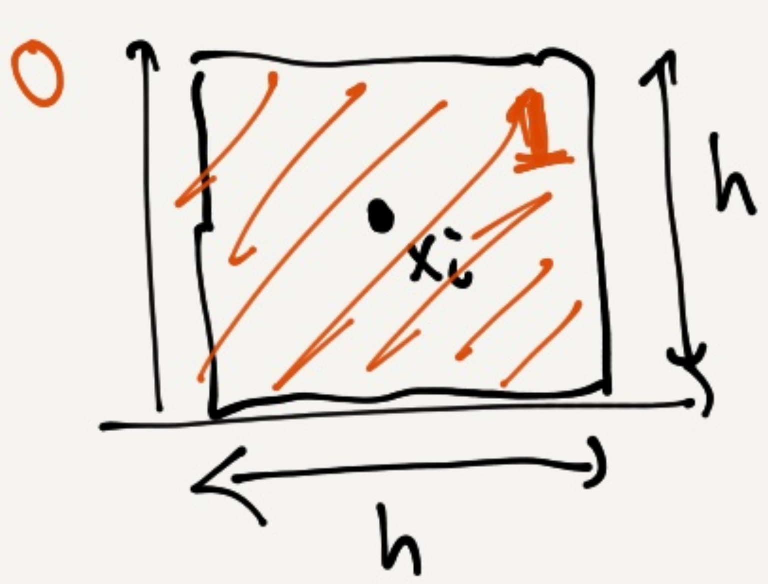
- introduce a window (kernel) (unit box)

$$K(x) = \begin{cases} 1, & |x_i| \le \frac{1}{2}, \quad \forall i \in \{1, \dots, d\} \\ 0, & \text{otherwise} \end{cases}$$

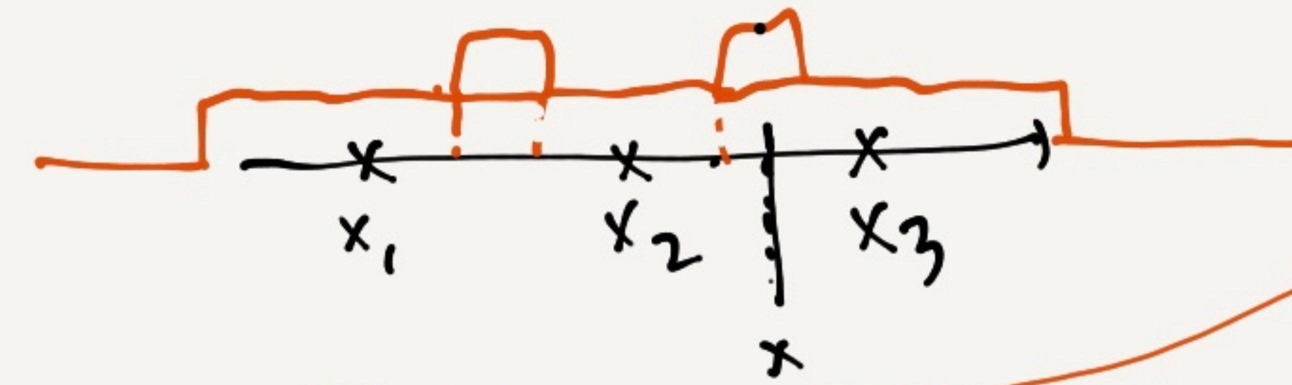




Note: $K\left(\frac{x - x_i}{h}\right) = \begin{cases} 1, & \text{if } x \text{ falls inside cube w/ side } h, \text{ centered at } x_i \\ 0, & \text{otherwise} \end{cases}$



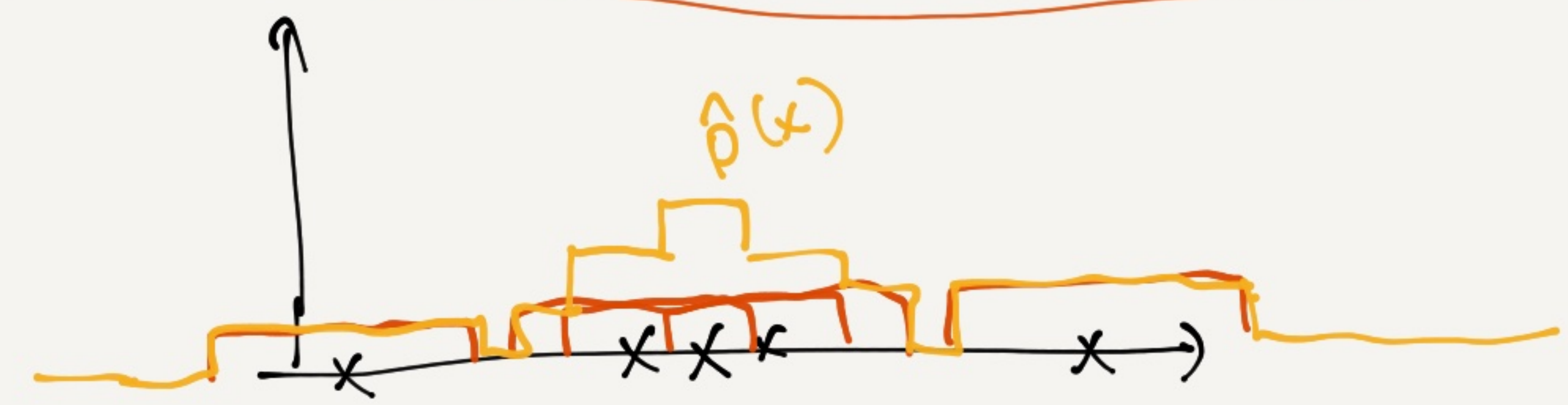
$\Rightarrow$ # of points near to x $= K = \sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right)$



Thus

$$\hat{p}(x) = \frac{1}{N}\frac{K_R}{V_R} = \frac{1}{\underbrace{Nh^d}_{V_R}} \overbrace{\sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right)}^{K_R}$$

estimate of $p(x)$ from $\{x_1, \dots, x_N\}$



estimation using interpolation between samples x_i.

$\to$ each x_i contributes to a local region.

Kernel Functions

Constraints: $K(x) \geq 0$, $\int K(x)\,dx = 1$ } it must be a valid pdf.

Example:

uniform box: $K(x) = \begin{cases} 1, & |x_i| \leq \frac{1}{2} \quad \forall i = \{1, \dots, d\} \\ 0, & \text{otherwise} \end{cases}$

unit sphere: $K(x) = \begin{cases} 1/c, & \|x\|^2 \leq 1 \\ 0, & \text{otherwise} \end{cases}$

c = volume of sphere.

Gaussian: $K(x) = \frac{1}{(2\pi)^{d/2}} e^{-\frac{1}{2}\|x\|^2}$

$$\hat{p}(x) = \frac{1}{Nh^d} \sum_i K\left(\frac{x - x_i}{h}\right)$$

$$= \frac{1}{Nh^d} \sum_i \frac{1}{(2\pi)^{d/2}} e^{-\frac{1}{2}\left\|\frac{x-x_i}{h}\right\|^2}$$

$|h^2 I|^{1/2}$

$$= \underbrace{\frac{1}{N}}_{\pi_i = \frac{1}{N}} \sum_i \underbrace{N(x \mid x_i, h^2 I)}_{\text{Gaussian component}}$$

Gaussian component
mean $= x_i$
cov $= h^2 I$

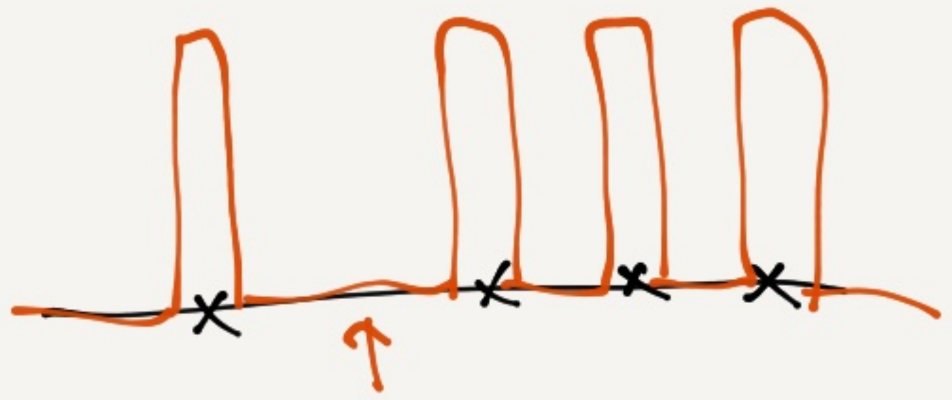
GMM w/ N components

Bandwidth Parameter

h controls the size of the region. (covariance of the Gaussian)

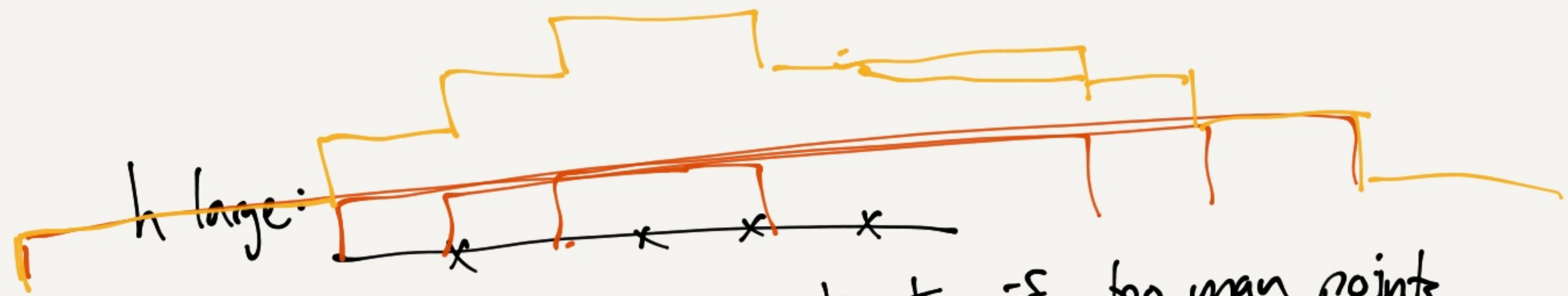
Intuitively,

h small:



might be noisy if not enough samples.

h large:



blurry estimate if too many points.

h is important and controls the quality of $\hat{p}(x)$.

Is there an optimal setting to recover the true $p(x)$?

Convergence Analysis

* Will $\hat{p}(x)$ converge to the true pdf $p(x)$?

$\hat{p}(x)$ depends on samples $\{x_i\}$, which are r.v. → bias / variance.

- We say $\hat{p}(x)$ converges to $p(x)$ if:

 1) $\lim_{N\to\infty} E[\hat{p}(x)] = p(x)$

 2) $\lim_{N\to\infty} var(\hat{p}(x)) = 0$

- Define: $\tilde{k}(x) = \frac{1}{h^d} k\left(\frac{x}{h}\right)$

scale amplitude ($\frac{1}{h^d}$), scale width ($\frac{x}{h}$).

Then: $\hat{p}(x) = \frac{1}{N}\sum_{i=1}^{N} \tilde{k}(x - x_i)$

Mean: $E[\hat{p}(x)] = \ldots\ldots$ (Tutorial S.1)

$$= \int p(\mu)\tilde{k}(x-\mu)\,d\mu \quad \leftarrow \text{defn of convolution}$$

$$= p(x) * \tilde{k}(x) \quad \leftarrow \text{conv. of } p(x) \text{ with the kernel } \tilde{k}(x).$$

e.g.

blurry $p(x)$, where the blur comes from the kernel

to have unbiased $\hat{p}(x)$, we want

$$E[\hat{p}(x)] = p(x) * \tilde{k}(x) = p(x)$$

$$\Rightarrow \tilde{k}(x) = \delta(x) = \lim_{h\to 0} \tilde{k}(x)$$

dirac delta

to be unbiased, we want $h = 0$. or $\tilde{k}(x) = \delta(x)$

Variance (Tut. Sol)

$var(\hat{p}(x)) : \ldots$

$$\boxed{var(\hat{p}(x)) \leq \frac{1}{Nh^d}\left[\max_x k(x)\right] E[\hat{p}(x)]}$$

For small variance, we need:

- h to be large.

OR

- N to be large.

---

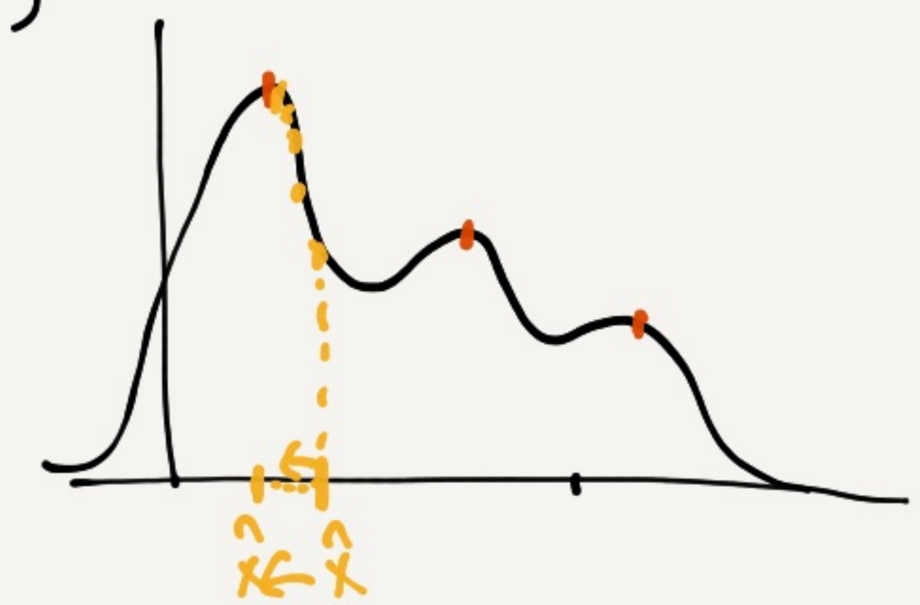
* h controls the tradeoff btwn bias & variance:

$$\begin{cases} h \to 0 \Rightarrow \text{bias} = 0, & var = \infty \\ h \to \infty \Rightarrow \text{bias} > 0, & var = 0 \end{cases}$$

No theoretical optimal choice ⇒ choose h based on problem.

Mean-Shift Algorithm (Comaniciu & Meer)

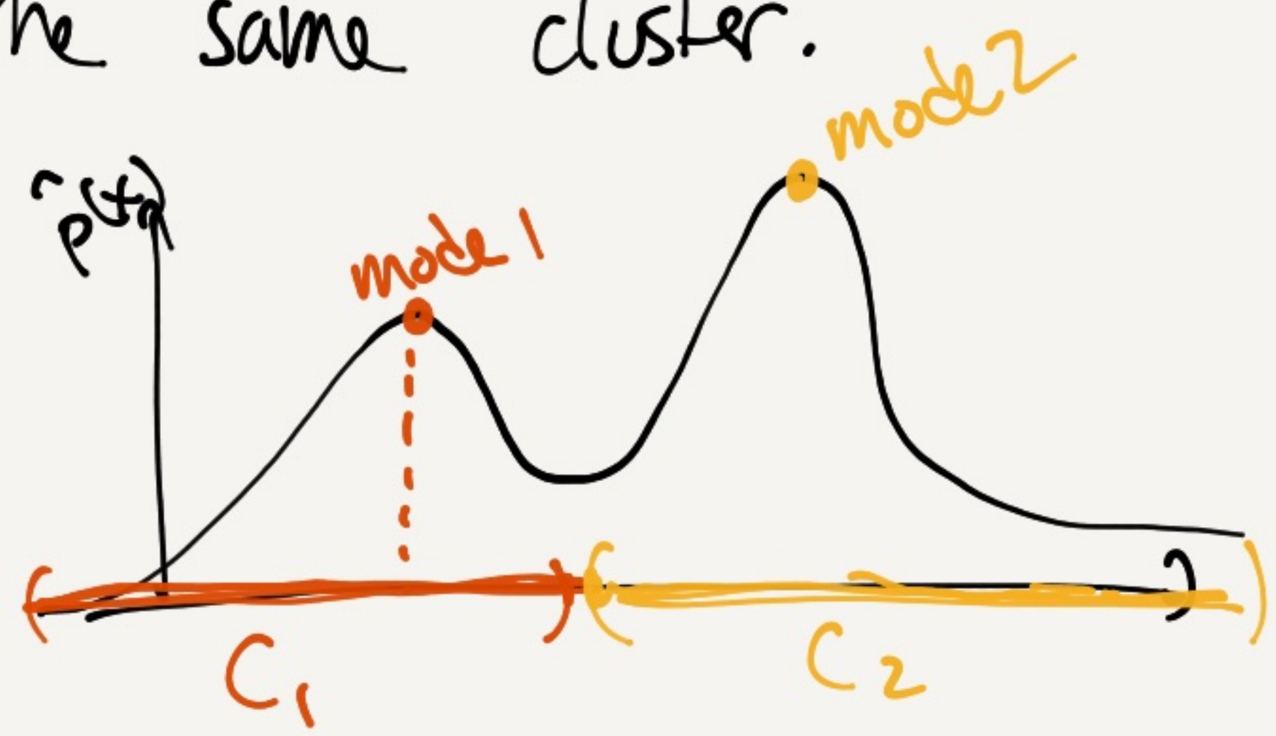
- Find the modes of $\hat{p}(x)$



- Idea:
  1) Start at a point $\hat{x}$.
  2) use gradient ascent to move uphill ($\hat{x} \leftarrow \hat{x} + \lambda \nabla \hat{p}(x)$)
  3) eventually $\hat{x}$ will converge to the mode.

Modes: Repeat w/ many initial $\hat{x}$'s.
Remove duplicate converged $\hat{x}$s $\rightarrow$ modes.

Clustering: given some x_i, all the x_i that yield the same mode are in the same cluster.



Consider only radially symmetric kernels

$$K(x) = \alpha \bar{k}(\|x\|^2)$$

α: constant; $\bar{k}$: kernel profile

e.g. Gaussian: $\bar{k}(r) = e^{-\frac{1}{2}r}$, $\alpha = (2\pi)^{-d/2}$, $r \geq 0$

Density Estimate

$$\hat{p}(x) = \frac{1}{Nh^d} \alpha \sum_{i=1}^{N} \bar{k}\left(\left\|\frac{x - x_i}{h}\right\|^2\right)$$

Gradient

Define: $\bar{g}(r) = -\bar{k}'(r)$, Gaussian: $\bar{g}(r) = \frac{1}{2} e^{-\frac{1}{2}r} = \frac{1}{2}\bar{k}(r)$

$$\nabla \hat{p}(x) = \underbrace{\frac{\alpha}{Nh^{d+2}}\left(\sum_{i=1}^{N} \bar{g}\left(\left\|\frac{x-x_i}{h}\right\|^2\right)\right)}_{\approx \text{ density estimate using } \bar{g}(r) \text{ instead of } \bar{k}(r) \; = \hat{g}(x)} \underbrace{\left[\underbrace{\frac{\sum_i x_i \bar{g}\left(\left\|\frac{x-x_i}{h}\right\|^2\right)}{\sum_i \bar{g}\left(\left\|\frac{x-x_i}{h}\right\|^2\right)}}_{\text{weighted mean of samples } x_i\text{. weights depend on distance to a point } x\text{. (closer samples have higher weights)}} - x\right]}_{\text{"mean-shift vector" - difference btwn weighted mean and window center } = m(x)}$$

Gradient Ascent

$$\hat{x}^{(k+1)} = \hat{x}^{(k)} + \lambda \nabla \hat{p}(\hat{x}^{(k)})$$

↑ stepsize (important for convergence)

Use an adaptive stepsize.

$$\lambda = \frac{1}{\hat{g}(x)}$$

- $\hat{g}(x)$ is small (low density region) → stepsize is large.
- $\hat{g}(x)$ is large (high density region) → stepsize is small

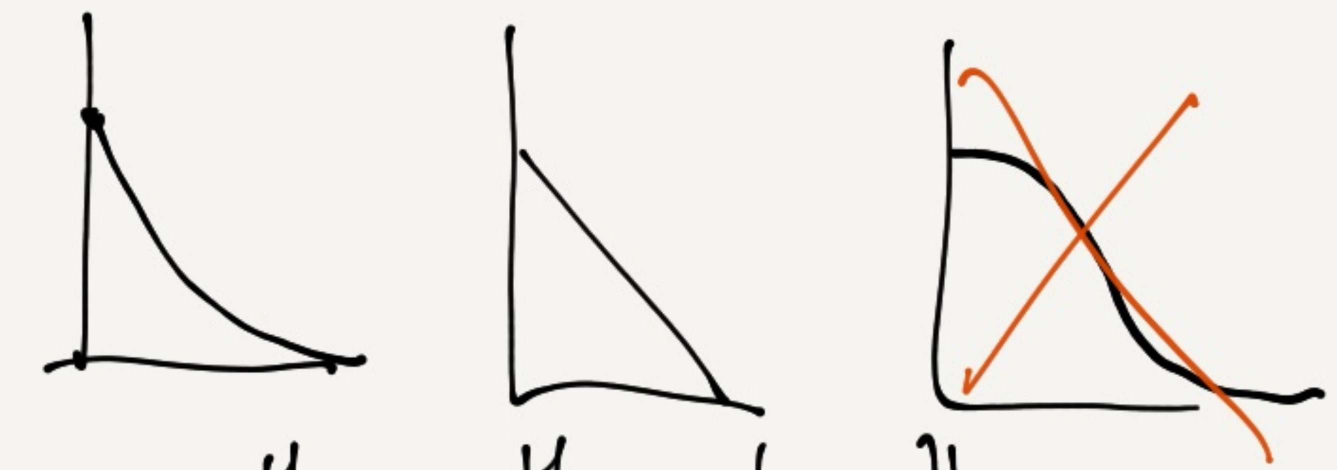
$$\Rightarrow \hat{x}^{(k+1)} = \hat{x}^{(k)} + \frac{1}{\cancel{\hat{g}(x^{(k)})}} \hat{g}\cancel{(x^{(k)})}\, \hat{m}(x^{(k)})$$

$$\Rightarrow \boxed{\hat{x}^{(k+1)} = \frac{\sum_i x_i\, \bar{g}\left(\left\|\frac{\hat{x}^{(k)} - x_i}{h}\right\|^2\right)}{\sum_i \bar{g}\left(\left\|\frac{\hat{x}^{(k)} - x_i}{h}\right\|^2\right)}}$$

mean-shift procedure

in each iteration, shift to the weighted mean of the nearby points.

- The profile should be monotonically decreasing & convex.



if so, then the algorithm is gauranteed to converge to a stationary point.