Kernel-based density estimation

Nuno Vasconcelos

ECE Department, UCSD

Announcement

- last week of classes we will have "Cheetah Day" (exact day TBA)
- what:
  - 4 teams of 6 people
  - each team will write a report on the 4 cheetah problems
  - each team will give a presentation on one of the problems
- why:
  - to make sure that we get the "big picture" out of all this work
  - presenting is always good practice

Announcement

- how much:
  - 10% of the final grade (5% report, 5% presentation)
- what to talk about:
  - report: comparative analysis of all solutions of the problem (8 page)
  - as if you were writing a conference paper
  - presentation: will be on one single problem
    - review what solution was
    - what did this problem taught us about learning?
    - what "tricks" did we learn solving it?
    - how well did this solution do compared to others?

Announcement

- details:
  - get together and form groups
  - let me know what they are by Wednesday (November 19) (email is fine)
  - I will randomly assign the problem on which each group has to be expert
  - prepare a talk for 20min (max 10 slides)
  - feel free to use my solutions, your results
  - feel free to go beyond what we have done (e.g. search over features, whatever…)

Plan for today

- we have talked a lot about the BDR and methods based on density estimation
- practical densities are not well approximated by simple probability models
- today: what can we do if have complicated densities?
  - use better probability density models!

Non-parametric density estimates

- Given iid training set $\mathcal{D} = \{\mathbf{x}_1, \ldots \mathbf{x}_n\}$, the goal is to estimate

$$P_{\mathbf{X}}(\mathbf{x})$$

- Consider a region $\mathcal{R}$, and define

$$P = P_{\mathbf{X}}[\mathbf{x} \in \mathcal{R}] = \int_{\mathcal{R}} P_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}.$$

  and define

$$K = \sharp\{\mathbf{x}_i \in \mathcal{D} | \mathbf{x}_i \in \mathcal{R}\}.$$

- This is a binomial distribution of paramter P

$$\begin{aligned} P_K(k) &= \mathcal{B}(n, P) \\ &= \begin{pmatrix} n \\ k \end{pmatrix} P^k (1 - P)^{n-k} \end{aligned}$$

Binomial random variable

- ML estimate of P

$$\hat{P} = \frac{k}{n}.$$

and statistiscs

$$\begin{aligned} E[\hat{P}] &= \frac{1}{n}E[k] = \frac{1}{n}nP = P \\ var[\hat{P}] &= \frac{1}{n^2}var[k] = \frac{P(1-P)}{n}. \end{aligned}$$

- Note that $var[\hat{P}] \leq 1/4n$ goes to zero very quickly, i.e.

$$\hat{P} \rightarrow P.$$

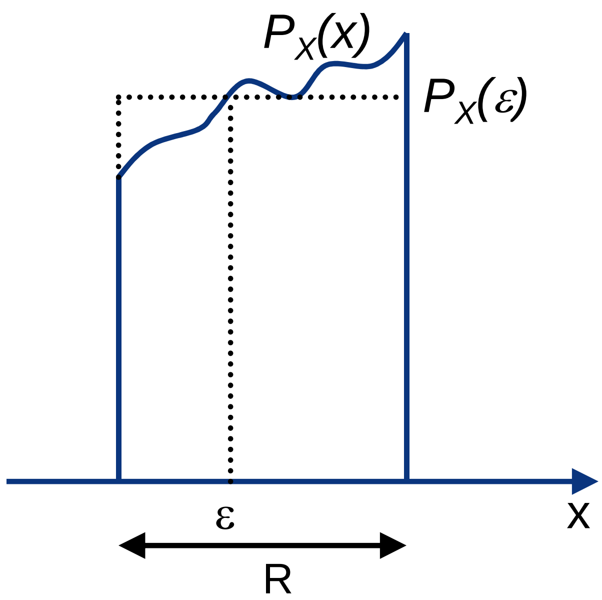
N	10	100	1,000	…
Var[P] <	0.025	0.0025	0.00025	

Histogram

- this means that k/n is a very good estimate of P
- on the other hand, from the mean value theorem, if $P_X(x)$ is continuous $\exists \epsilon \in \mathcal{R}$ such that

$$P = \int_{\mathcal{R}} P_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} = P_{\mathbf{X}}(\epsilon) \int_{\mathcal{R}} d\mathbf{x} = P_{\mathbf{X}}(\epsilon) V(\mathcal{R}).$$

- this is easiest to see in 1D
  - can always find a box such that the integral of the function is equal to that of the box
  - since $P_X(x)$ is continuous there must be a ε such that $P_X(\varepsilon)$ is the box height



Histogram

- hence

$$P_{\mathbf{X}}(\epsilon) = \frac{P}{V(\mathcal{R})} \approx \frac{\widehat{P}}{V(\mathcal{R})} = \frac{k}{nV(\mathcal{R})}$$

- using continuity of $P_X(x)$ again and assuming R is small

$$P_{\mathbf{X}}(\mathbf{x}) \approx \frac{k}{nV(\mathcal{R})}, \ \forall \mathbf{x} \in V(\mathcal{R})$$

- this is the histogram
- it is the simplest possible non-parametric estimator
- can be generalized into kernel-based density estimator

Kernel density estimates

- assume $\mathcal{R}$ is the d-dimensional cube of side h

$$V = h^d$$

and define *indicator* function of the unit hypercube

$$\phi(\mathbf{u}) = \begin{cases} 1, & \text{if } |u_i| < 1/2 \\ 0, & \text{otherwise.} \end{cases}$$

hence

$$\phi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) = 1$$

iif $\mathbf{x}_i \in$ hypercube of volume V centered at $\mathbf{x}$.

- the number of sample points in the hypercube is

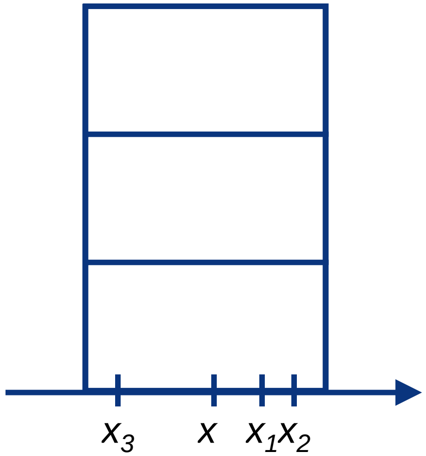
$$k_n = \sum_{i=1}^{n} \phi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right)$$

Kernel density estimates

- this means that the histogram can be written as

$$P_{\mathbf{X}}(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^{n} \phi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right)$$

- which is equivalent to:
  - "put a box around X for each X_i that lands on the hypercube"
  - can be seen as a very crude form of interpolation
  - better interpolation if contribution of X_i decreases with distance to X
- consider other windows $\phi(x)$



Windows

- what sort of functions are valid windows?
- note that $P_X(x)$ is a pdf if and only if

$$P_{\mathbf{X}}(\mathbf{x}) \geq 0, \forall \mathbf{x} \text{ and } \int P_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} = 1$$

- since

$$\begin{aligned} \int P_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} &= \frac{1}{nh^d} \sum_{i=1}^{n} \int \phi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) d\mathbf{x} \\ &= \frac{1}{nh^d} \sum_{i=1}^{n} \int \phi(\mathbf{y})\, h^d d\mathbf{y} \\ &= \frac{1}{n} \sum_{i=1}^{n} \int \phi(\mathbf{y})\, d\mathbf{y} \end{aligned}$$

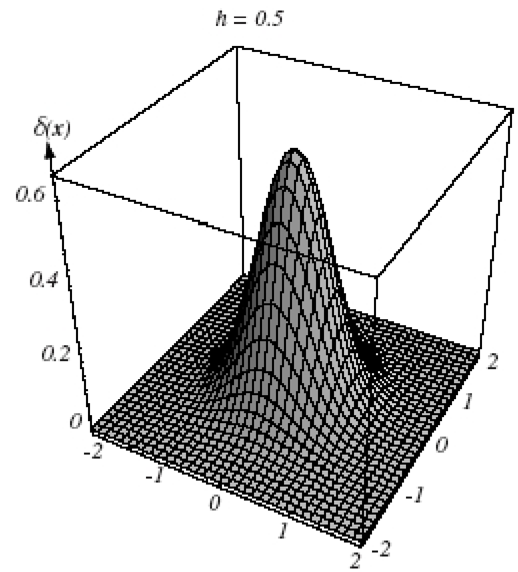
- these conditions hold if $\phi(x)$ is itself a pdf

$$\phi(\mathbf{x}) \geq 0, \forall \mathbf{x} \text{ and } \int \phi(\mathbf{x}) d\mathbf{x} = 1$$

Gaussian kernel

- probably the most popular in practice

$$\phi(\mathbf{x}) = \frac{1}{\sqrt{2\pi}^d} e^{-\frac{1}{2}\mathbf{x}^T\mathbf{x}}$$

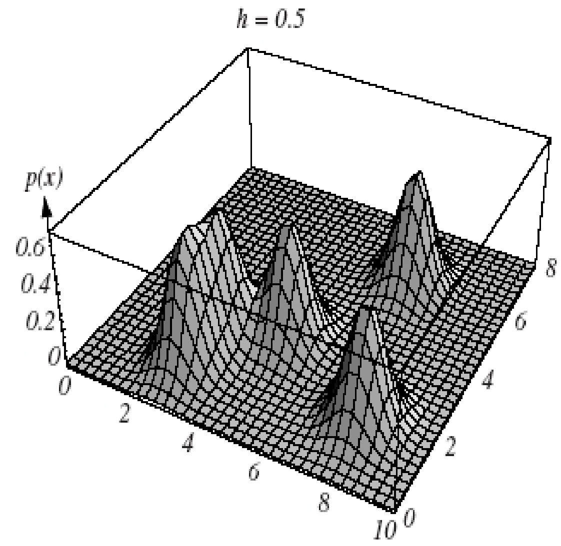


- note that $P_X(x)$ can also be seen as a sum of pdfs centered on the X_i when $\phi(x)$ is symmetric in X and X_i

$$P_{\mathbf{X}}(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^{n} \phi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right)$$

Gaussian kernel

- Gaussian case can be interpreted as
  - sum of *n* Gaussians centered at the X_i with covariance $h\mathbf{I}$
  - more generally, we can have a full covariance



$$P_{\mathbf{X}}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{x}_i)^T\Sigma^{-1}(\mathbf{x}-\mathbf{x}_i)}$$

- sum of *n* Gaussians centered at the X_i with covariance Σ
- Gaussian kernel density estimate: *"approximate the pdf of X with a sum of Gaussian bumps"*

Kernel bandwidth

- back to the generic model

$$P_{\mathbf{X}}(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^{n} \phi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right)$$

- what is the role of *h* (bandwidth parameter)?
- defining

$$\delta(\mathbf{x}) = \frac{1}{h^d} \phi\left(\frac{\mathbf{x}}{h}\right)$$

- we can write

$$P_{\mathbf{X}}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} \delta\left(\mathbf{x} - \mathbf{x}_i\right)$$

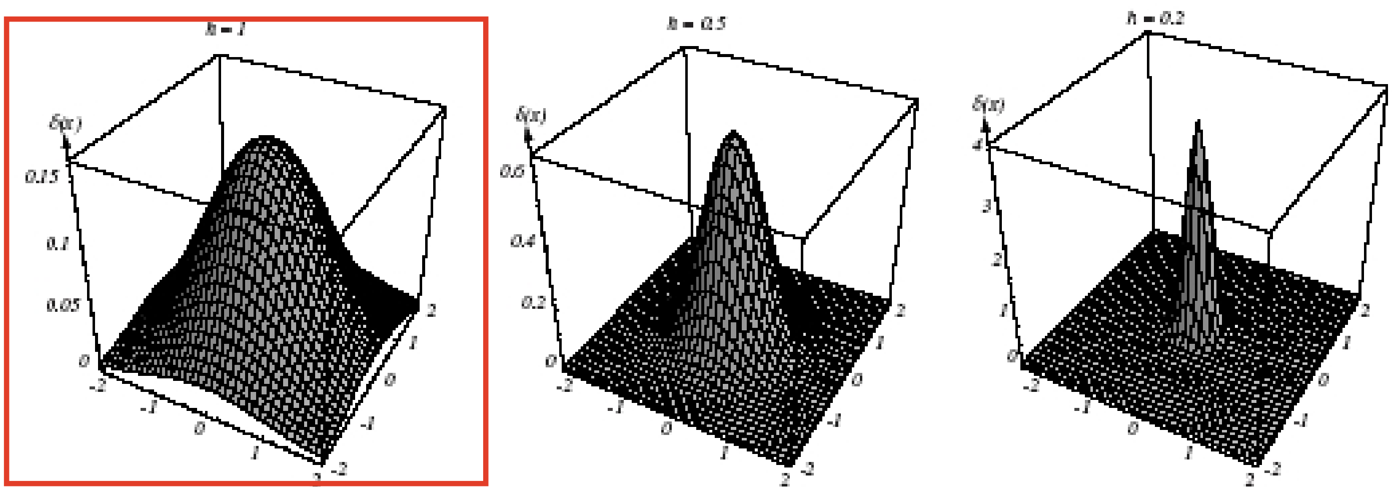
- i.e. a sum of translated replicas of $\delta(x)$

Kernel bandwidth

- *h* has two roles:
  1. rescale the *x*-axis
  2. rescale the amplitude of *δ(x)*

$$\delta(\mathbf{x}) = \frac{1}{h^d}\phi\left(\frac{\mathbf{x}}{h}\right)$$

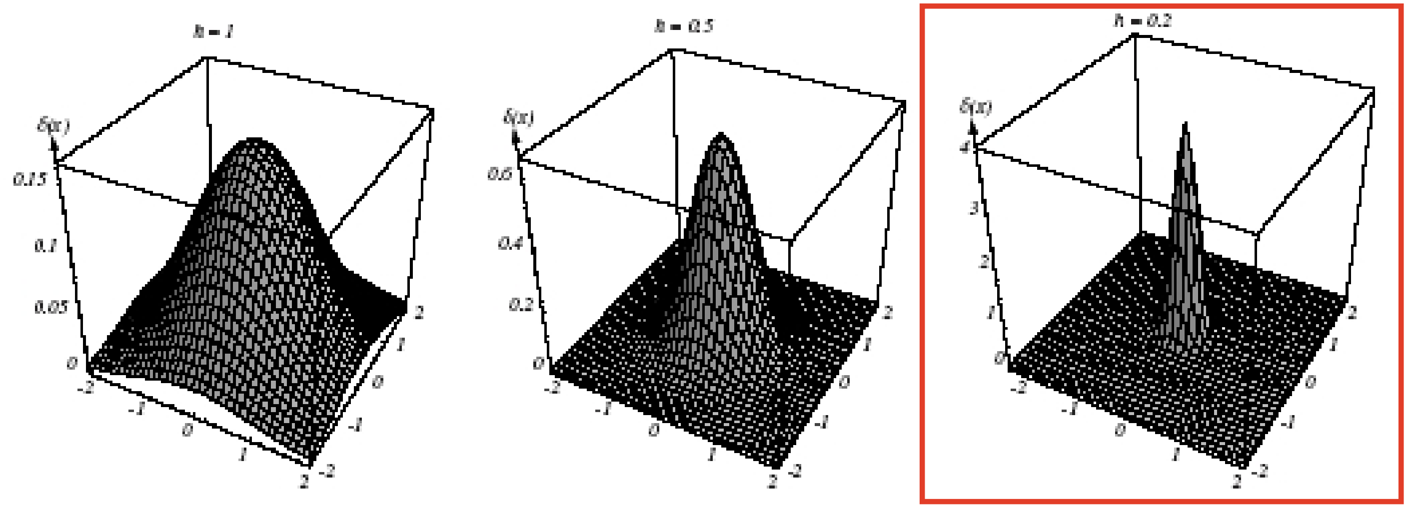
- this implies that for large *h*:
  1. *δ(x)* has low amplitude
  2. iso-contours of *h* are quite distant from zero
     (*x* large before *ϕ(x/h)* changes significantly from *ϕ(0)*)



Kernel bandwidth

$$\delta(\mathbf{x}) = \frac{1}{h^d}\phi\left(\frac{\mathbf{x}}{h}\right)$$

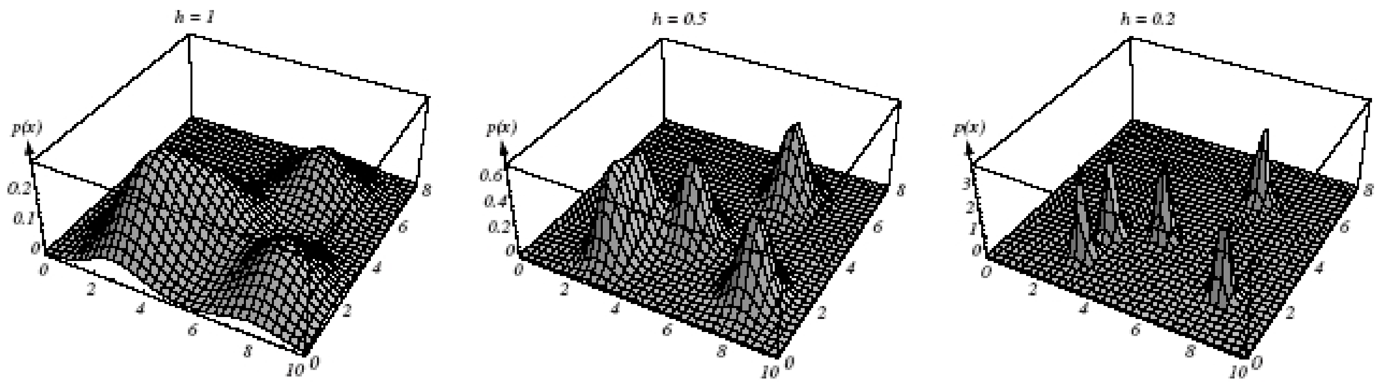
- for small *h*:
  1. $\delta(x)$ has large amplitude
  2. iso-contours of *h* are quite close to zero
     (*x* small before $\phi(x/h)$ changes significantly from $\phi(0)$)



- what is the impact of this on the quality of the density estimates?

Kernel bandwidth

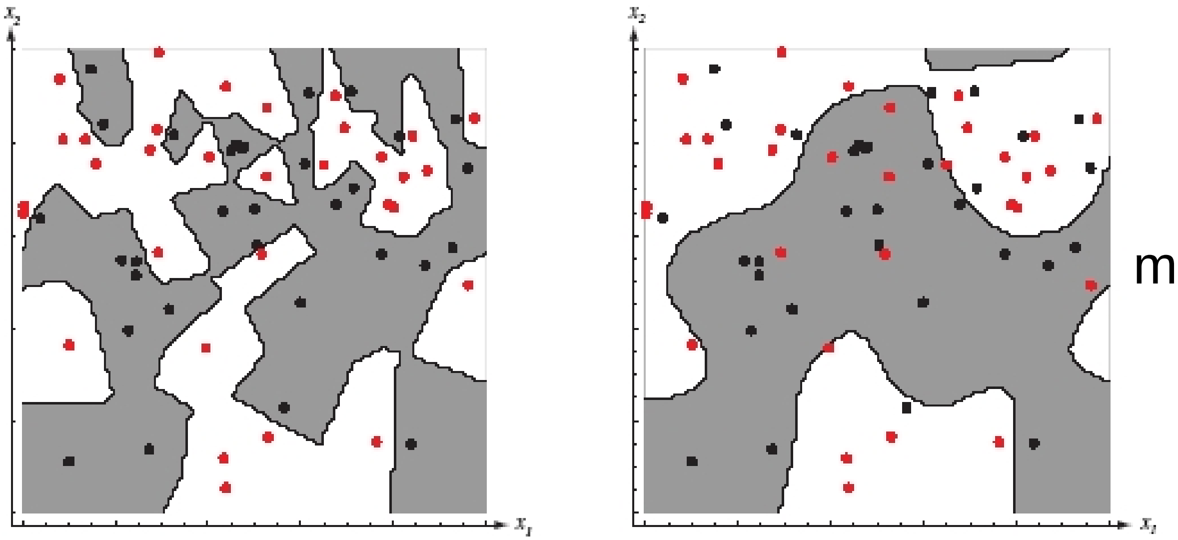
- it controls the smoothness of the estimate
    - as h goes to zero we have a sum of delta functions (very “spiky” approximation)
    - as h goes to infinity we have a sum of constant functions (approximation by a constant)
    - in between we get approximations that are gradually more smooth



Kernel bandwidth

- why does this matter?
- when the density estimates are plugged into the BDR
- smoothness of estimates determines the smoothness of the boundaries

less smooth



more smooth

- this affects the probability of error!

Convergence

- since $P_x(x)$ depends on the sample points X_i, it is a random variable
- as we add more points, the estimate should get “better”
- the question is then whether the estimate ever converges
- this is no different than parameter estimation
- as before, we talk about convergence in probability
- $\hat{P}_{\mathbf{X}}(\mathbf{x})$ converges to $P_{\mathbf{X}}(\mathbf{x})$ if

$$\begin{aligned}\lim_{n\to\infty} E_{\mathbf{X}_1,\ldots\mathbf{X}_n}[\hat{P}_{\mathbf{X}}(\mathbf{x})] &= \hat{P}_{\mathbf{X}}(\mathbf{x}) \\ \lim_{n\to\infty} var_{\mathbf{X}_1,\ldots\mathbf{X}_n}[\hat{P}_{\mathbf{X}}(\mathbf{x})] &= 0\end{aligned}$$

Convergence of the mean

- from the linearity of $P_X(x)$ on the kernels

$$
\begin{aligned}
& E_{\mathbf{X}_1,\ldots\mathbf{X}_n}[\widehat{P}_{\mathbf{X}}(\mathbf{x})] = \\
&= \frac{1}{nh^d}\sum_{i=1}^{n} E_{\mathbf{X}_i}\left[\phi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\int \frac{1}{h^d}\phi\left(\frac{\mathbf{x}-\mathbf{v}}{h}\right)P_{\mathbf{X}}(\mathbf{v})d\mathbf{v} \\
&= \int \frac{1}{h^d}\phi\left(\frac{\mathbf{x}-\mathbf{v}}{h}\right)P_{\mathbf{X}}(\mathbf{v})d\mathbf{v} \\
&= \int \delta(\mathbf{x}-\mathbf{v})P_{\mathbf{X}}(\mathbf{v})d\mathbf{v}
\end{aligned}
$$

Convergence of the mean

- hence

$$E_{\mathbf{X}_1,\ldots\mathbf{X}_n}[\hat{P}_{\mathbf{X}}(\mathbf{x})] = \int \delta(\mathbf{x}-\mathbf{v})P_{\mathbf{X}}(\mathbf{v})d\mathbf{v}$$

- this is the convolution of $P_X(x)$ with $\delta(x)$
- it is a blurred version ("low-pass filtered") unless $h = 0$
- in this case $\delta(x-v)$ converges to the Dirac delta and so

$$\lim_{h\to 0} E_{\mathbf{X}_1,\ldots\mathbf{X}_n}[\hat{P}_{\mathbf{X}}(\mathbf{x})] = P_{\mathbf{X}}(\mathbf{x})$$

Convergence of the variance

- since the X_i are iid

$$
\begin{aligned}
var_{\mathbf{X}_1,\ldots\mathbf{X}_n}[\widehat{P}_{\mathbf{X}}(\mathbf{x})] = \\
&= \sum_{i=1}^{n} var_{\mathbf{X}_i}\left[\frac{1}{nh^d}\phi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)\right] \\
&\leq nE_{\mathbf{X}}\left[\frac{1}{n^2h^{2d}}\phi^2\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)\right] \\
&= \frac{1}{nh^d}\int\frac{1}{h^d}\phi^2\left(\frac{\mathbf{x}-\mathbf{v}}{h}\right)P_{\mathbf{X}}(\mathbf{v})d\mathbf{v} \\
&\leq \frac{1}{nh^d}\sup\left[\phi\left(\frac{\mathbf{x}}{h}\right)\right]\int\frac{1}{h^d}\phi\left(\frac{\mathbf{x}-\mathbf{v}}{h}\right)P_{\mathbf{X}}(\mathbf{v})d\mathbf{v} \\
&= \frac{1}{nh^d}\sup\left[\phi\left(\frac{\mathbf{x}}{h}\right)\right]E_{\mathbf{X}_1,\ldots\mathbf{X}_n}[\widehat{P}_{\mathbf{X}}(\mathbf{x})]
\end{aligned}
$$

Convergence

- in summary

$$E_{\mathbf{X}_1,\ldots\mathbf{X}_n}[\widehat{P}_{\mathbf{X}}(\mathbf{x})] \;=\; \delta(\mathbf{x}) \odot P_{\mathbf{X}}(\mathbf{x})$$

$$var_{\mathbf{X}_1,\ldots\mathbf{X}_n}[\widehat{P}_{\mathbf{X}}(\mathbf{x})] = \\ \leq \; \frac{1}{nh^d}\,\mathrm{sup}\left[\phi\left(\frac{\mathbf{x}}{h}\right)\right] E_{\mathbf{X}_1,\ldots\mathbf{X}_n}[\widehat{P}_{\mathbf{X}}(\mathbf{x})]$$

- this means that:
  - to obtain small bias we need *h ~ 0*
  - to obtain small variance we need *h* infinite

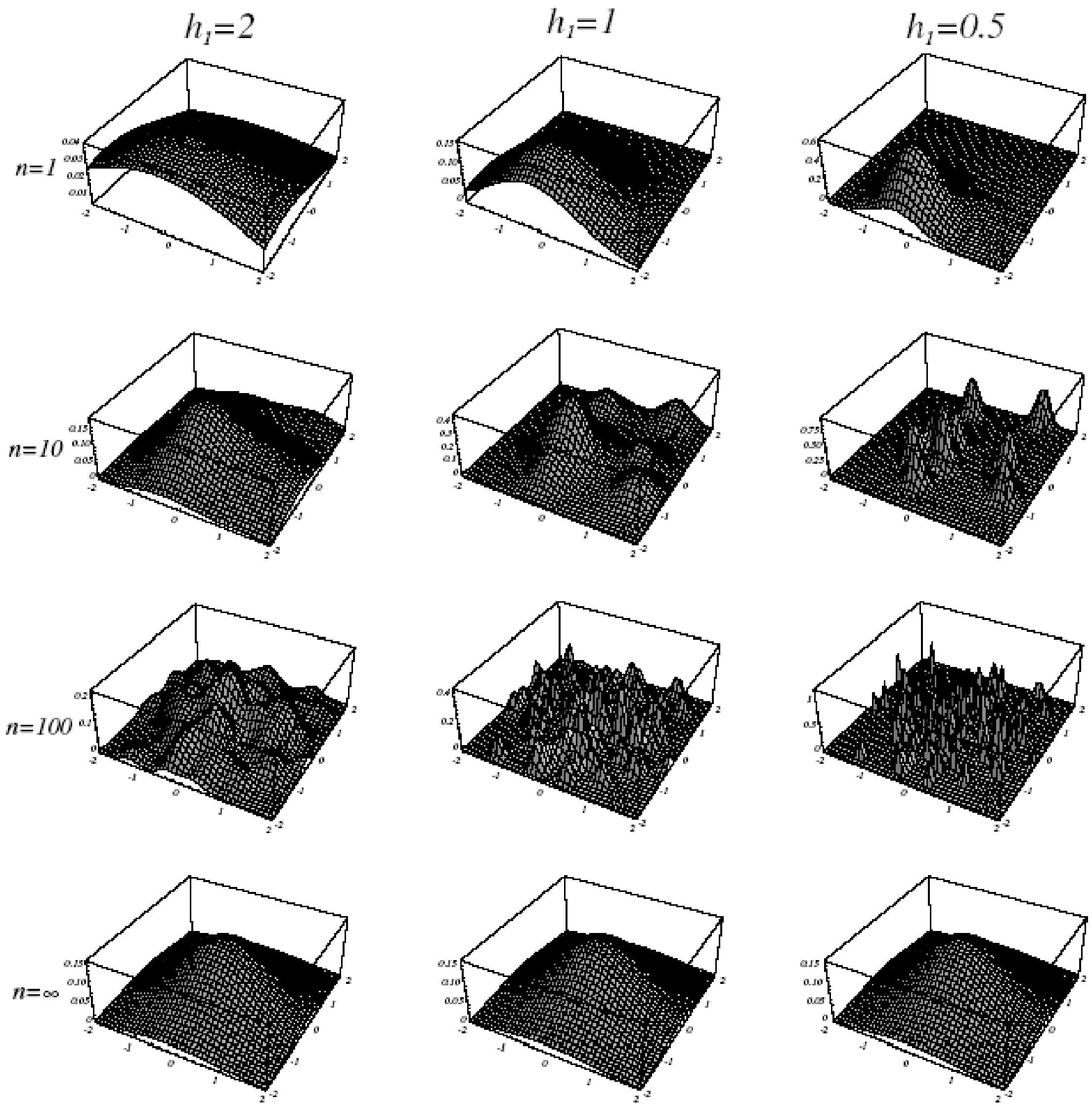
Convergence

- intuitively makes sense
  - $h \sim 0$ means a Dirac around each point
  - can approximate any function arbitrarily well
  - there is no bias
  - but if we get a different sample, the estimate is likely to be very different
  - there is large variance
  - as before, variance can be decreased by getting a larger sample
  - but, for fixed *n*, smaller h always means greater variability
- example: fit to N(0,I) using $h = h_1/n^{1/2}$

Example

- small h: spiky
- need a lot of points to converge (variance)

- large h: approximate N(0,I) with a sum of Gaussians of larger covariance
- will never have zero error (bias)



Optimal bandwidth

- we would like
  - *h ~ 0* to guarantee zero bias
  - zero variance as *n* goes to infinity
- solution:
  - make h a function of n that goes to zero
  - since variance is $O(1/nh^d)$ this is fine if nh^d goes to infinity
- hence, we need

$$\lim_{n \to \infty} h(n) = 0 \text{ and } \lim_{n \to \infty} nh(n) \infty$$

- optimal sequences exist, e.g.

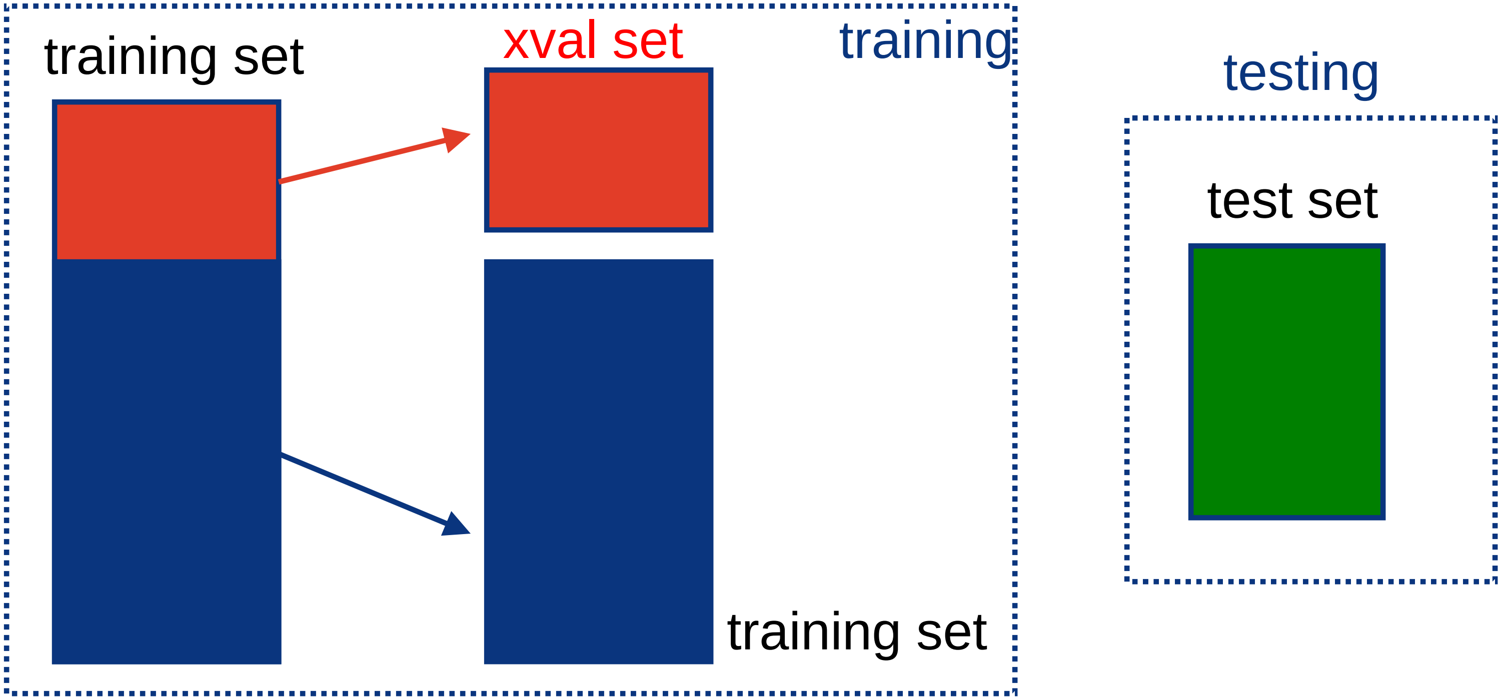
$$h(n) = \frac{k}{\sqrt{n}} \text{ or } h(n) = \frac{k}{\log n}$$

Optimal bandwidth

- in practice this has limitations
  - does not say anything about the finite data case (the one we care about)
  - still have to find the best k
- usually we end up using trial and error or techniques like cross-validation

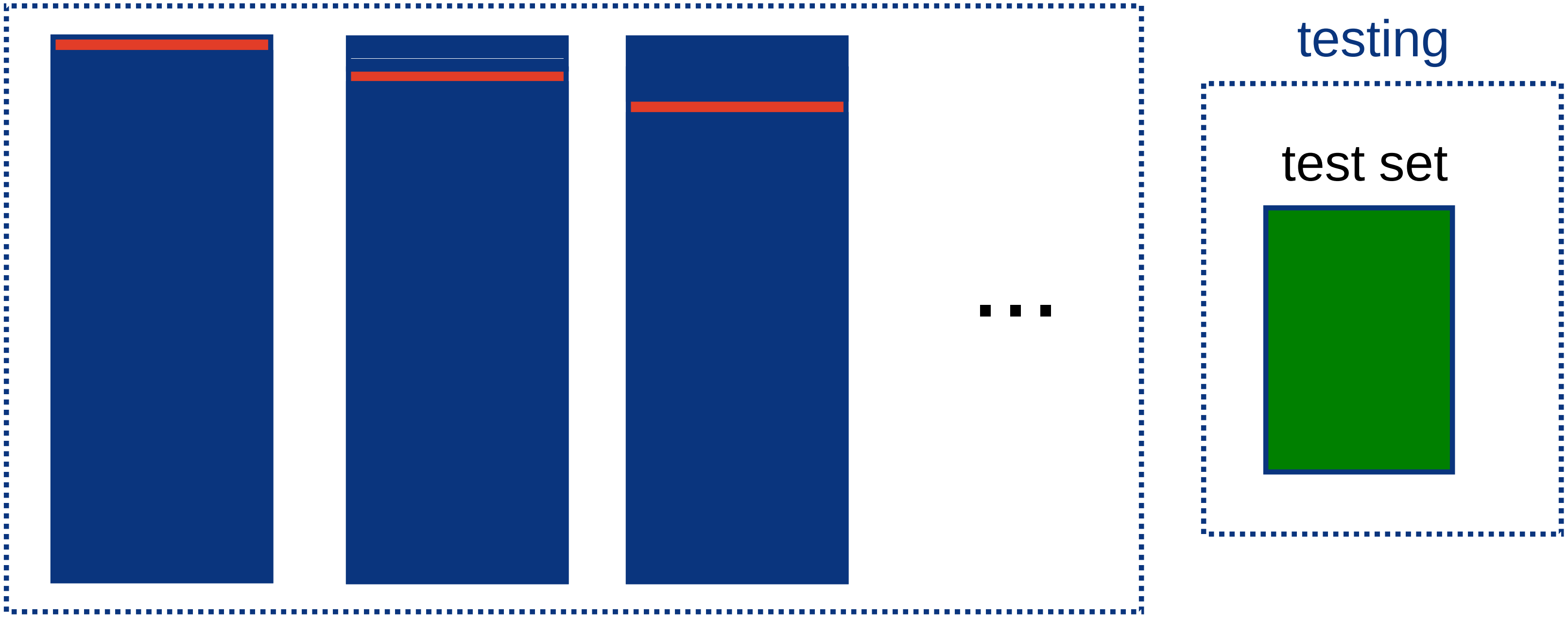
Cross-validation

- basic idea:
  - leave some data out of your training set (cross validation set)
  - train with different parameters
  - evaluate performance on cross validation set
  - pick best parameter configuration



Leave-one-out cross-validation

- many variations
- leave-one-out CV:
  - compute n estimators of $P_X(x)$ by leaving one X_i out at a time
  - for each $P_X(x)$ evaluate $P_X(X_i)$ on the point that was left out
  - pick $P_X(x)$ that maximizes this likelihood



Any Questions?