

CSE 6740 Lecture 5

How Do I Learn Any Density? (Model Selection and Nonparametric Estimation)

Alexander Gray

agray@cc.gatech.edu

Georgia Institute of Technology

Today

1. Model selection and generalization (*How do we minimize future error? Two Gaussians or three?*)
2. Nonparametric estimation (*What if I don't want to specify a simple parametric form?*)
3. Kernel density estimation (*How can I estimate a density nonparametrically?*)

Model Selection and Generalization

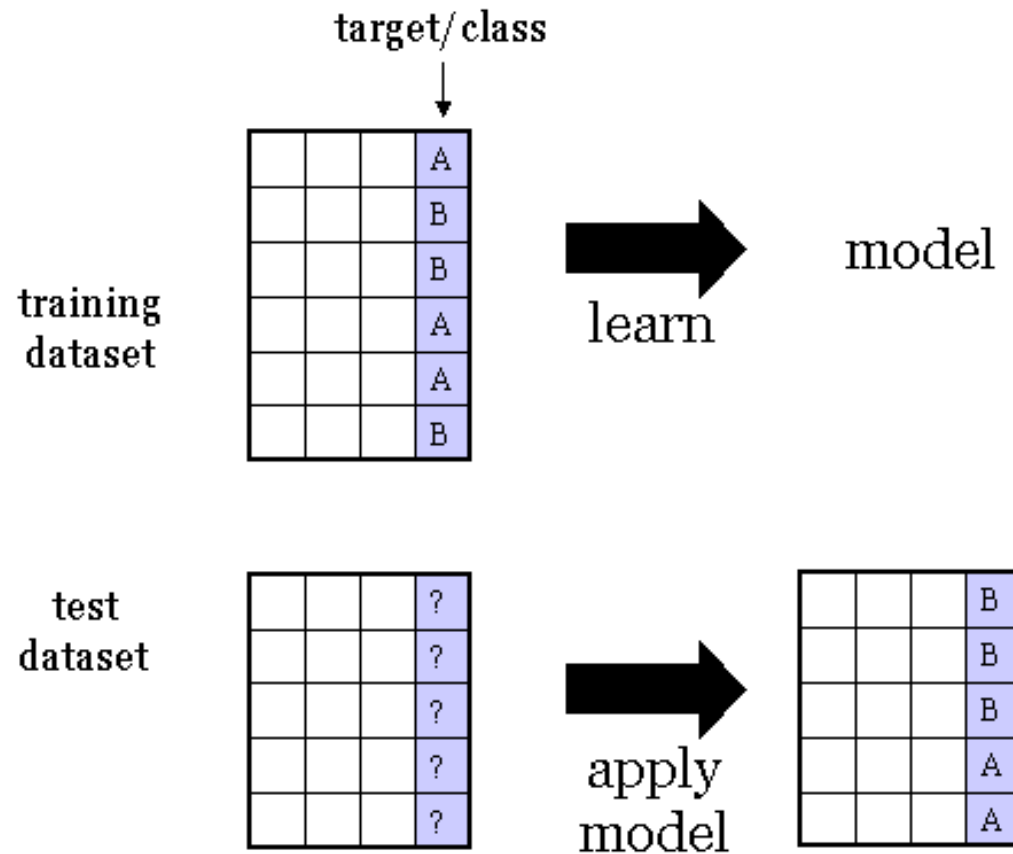
How do we minimize *future* error? Two Gaussians or three?

Model (Class) Selection

How many components K should we use? We could say that our model class is the set of all possible mixtures with a parameter K thrown in. However, this is no longer a finite number of parameters; we'll consider this scenario (*nonparametric estimation*) separately.

So we usually consider *parameter selection* (e.g. choosing $\Theta = (\mu, \Sigma)$) separately from *model selection* (e.g. choosing K or choosing between two different kinds of distributions). The extent to which these two processes should be different or the same is subtle and we will talk about this more.

Training and Testing



Recall the meaning of *training* and *testing*.

Training and Test Error

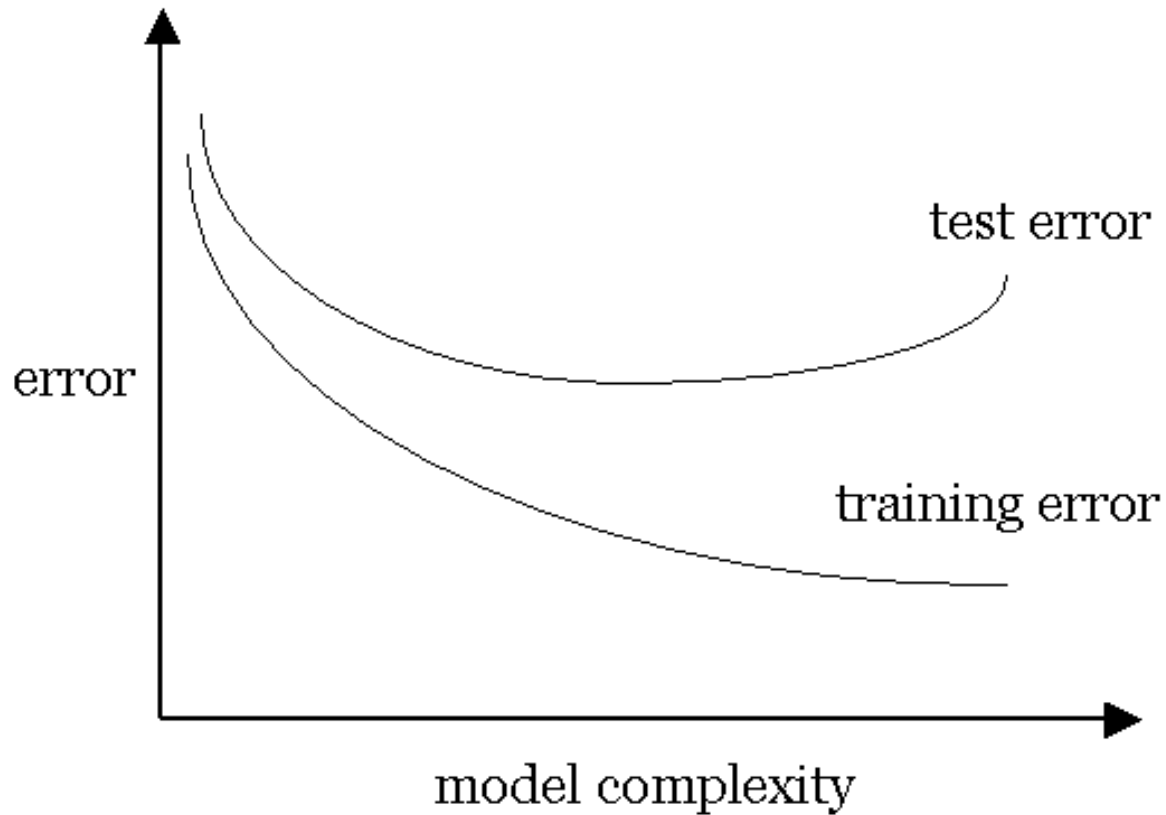
Test error, or generalization error, or prediction risk, is the expected error over an independent test sample drawn from the same distribution as that of the training data:

$$R(M) = \mathbb{E}L \left(\hat{Y}(M), Y \right). \quad (1)$$

Training error is the average loss over the training data:

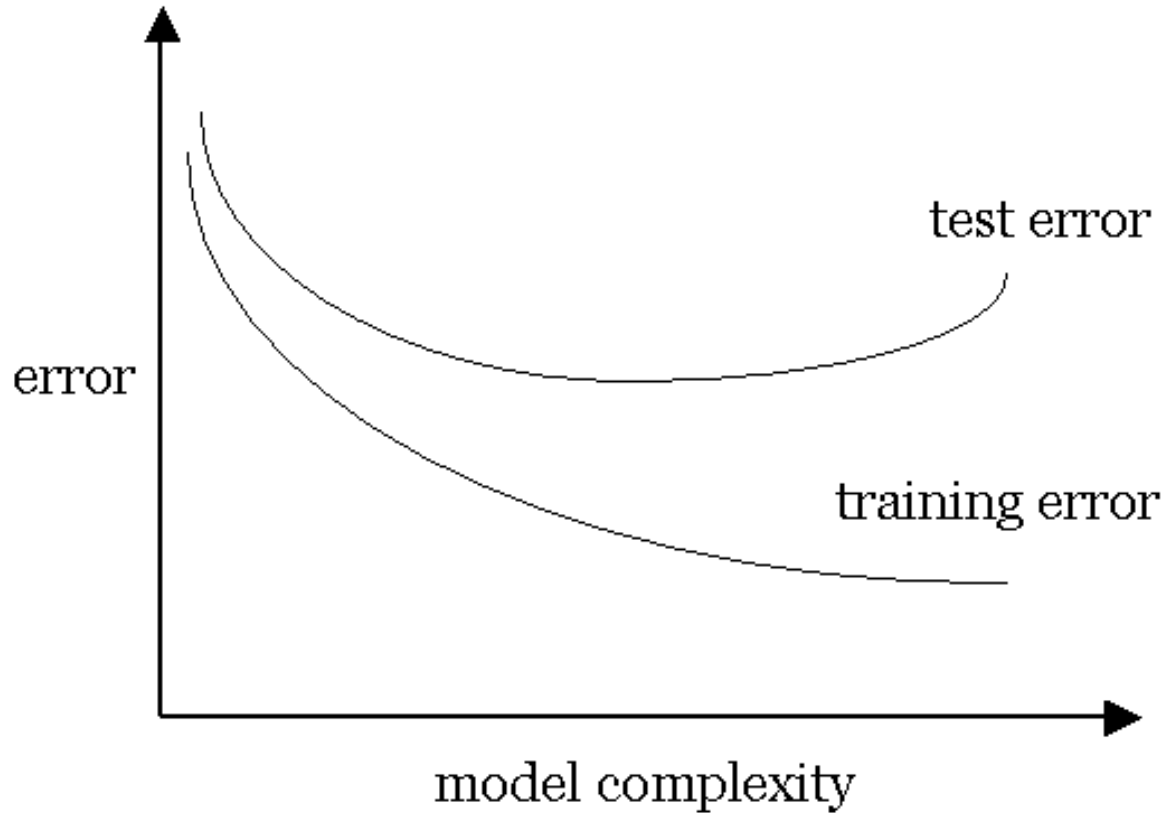
$$\hat{R}_{\text{tr}}(M) = \frac{1}{N} \sum_{i=1}^N L \left(\hat{Y}(M), Y_i \right). \quad (2)$$

Training and Test Error



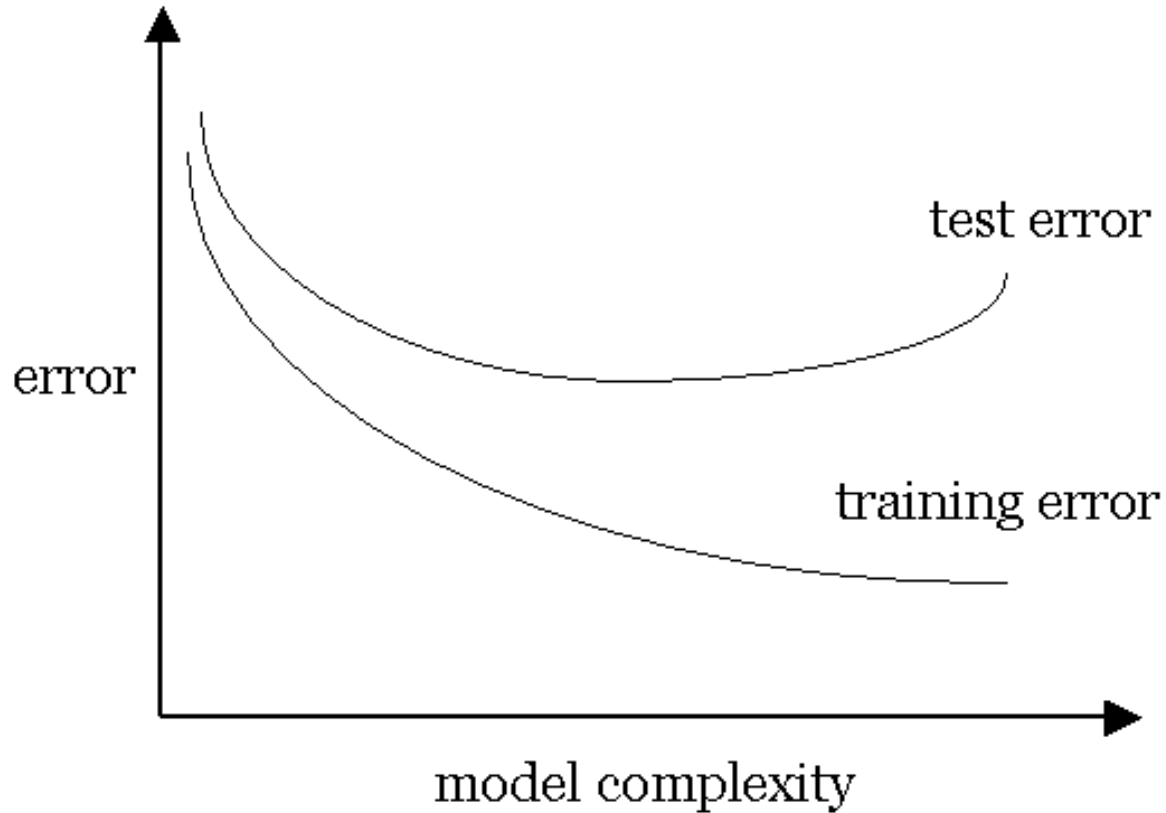
Our goal: find the model M which minimizes the test error: $\inf_M R(M)$. This is called *model selection*.

Training and Test Error



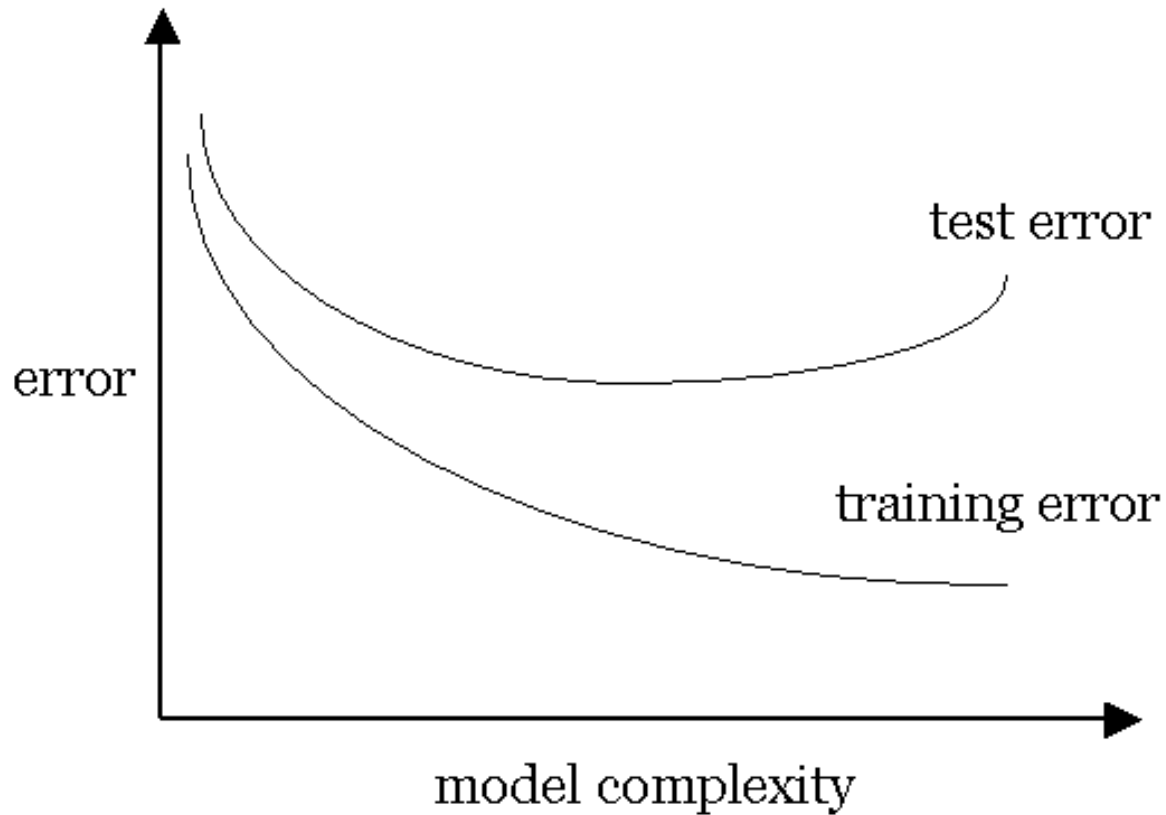
We call choosing a suboptimal model *overfitting* or *underfitting*.

Training and Test Error



In L_2 loss (coming up): too-simple models give too much *bias*, and too-complex models give too much *variance*.

Training and Test Error



The training error is a downward-biased estimate of the prediction risk: $\hat{R}_{\text{tr}}(M) < R(M)$.

Model Selection and Assessment

To perform model selection, we just need to know the **relative values** of the test error for different models.

Asymptotic approximations can sometimes be useful for comparing the test error for different models.

These are generally not good estimators of the **actual values** of the errors. Asymptotic arguments are generally not good enough to give us good finite-sample estimates, except possibly for *very simple* models, such as linear regression. For this we will turn to resampling methods.

Of course if we have a good way direct estimator of the test error, we can use it for model selection. We should when we can.

Optimism of the Training Error

The *optimism* of the training error is

$$\text{op}(M) = -\text{bias}(\hat{R}_{\text{tr}}(M)) \quad (3)$$

$$= -\mathbb{E} \left(\hat{R}_{\text{tr}}(M) - R(M) \right) \quad (4)$$

$$= \frac{2}{N} \sum_{i=1}^N \text{Cov}(\hat{Y}_i, Y_i) \quad (5)$$

$$= \frac{2}{N} \sum_{i=1}^N \mathbb{E} \left((\hat{Y}_i - \mathbb{E}\hat{Y}_i)(Y_i - \mathbb{E}Y_i) \right). \quad (6)$$

In other words, the amount by which $\hat{R}_{\text{tr}}(M)$ underestimates $R(M)$ depends on how strongly y_i affects its own prediction. The harder we fit the data, the greater the optimism will be.

Optimism of the Training Error

In general we have

$$R(M) = \mathbb{E}\hat{R}_{\text{tr}}(M) + \text{op}(M) \quad (7)$$

$$\approx \text{lack of fit} + \text{complexity penalty} \quad (8)$$

Thus to select the model we can:

1. obtain an estimate $\widehat{\text{op}}(M)$, or
2. directly estimate $R(M)$ some other way.

Asymptotic Approach

First, let's try to estimate the optimism asymptotically.

Under squared-error loss, with the error model $Y = f(X) + \epsilon$ where the error ϵ has zero mean and variance σ^2 ,

$$\text{op}(M) = \frac{2}{N} \sum_{i=1}^N \text{Cov}(\hat{Y}_i, Y_i) = 2 \frac{|M|}{N} \sigma^2 \quad (9)$$

where $|M|$ is the number of parameters of the model.

Mallows' C_p Statistic

This leads to an estimate of $R(M)$ called *Mallows' C_p statistic*:

$$C_p(M) = \hat{R}_{\text{tr}}(M) + 2\frac{|M|}{N}\hat{\sigma}^2 \quad (10)$$

where $\hat{\sigma}^2$ is obtained from the MSE of a low-bias (complex) model.

AIC Statistic

Suppose we are using the log-likelihood for our loss:

$$l(M) = \log L(M) = \log \left(\prod_{i=1}^N f(X_i; M) \right) = \sum_{i=1}^N \log f(X_i; M). \quad (11)$$

(Actually we use $-2l(M)$.)

If $M = \{\mathcal{F}, \hat{\theta}\}$, it turns out that as $N \rightarrow \infty$,

$$-2\mathbb{E} \log f(X; M) \approx -2\mathbb{E} \hat{l}_{\text{tr}}(M_{\hat{\theta}}) + 2 \frac{|M|}{N} \quad (12)$$

where $\hat{\theta}$ is the MLE for \mathcal{F} and $\hat{l}_{\text{tr}}(M_{\hat{\theta}})$ is the likelihood on the training data.

AIC Statistic

This leads to an estimate of $R(M)$ called *Akaike's Information Criterion* (AIC):

$$\text{AIC}(M) = \widehat{R}_{\text{tr}}(M_{\widehat{\theta}}) + 2 \frac{|M|}{N} \widehat{\sigma}^2 \quad (13)$$

where $\widehat{\sigma}^2$ is obtained from the MSE of a low-bias (complex) model.

This is the same as Mallows' C_p statistic except that it holds under broader assumptions (it is a generalization). Note however that this does not hold in general, for example, for 0-1 loss.

Note that AIC is not consistent, i.e. does not choose the right model asymptotically.

Finite-Sample Risk Estimation

We can also use try to *directly estimate* the test error using the actual data we have.

Suppose we had infinite data. We could use a chunk of it to train a model and a chunk of it (a *validation set*) to estimate the test error of the model.

However, it's never clear when we have enough data to be able to throw some away. So we simulate validation sets by *resampling*. Cross-validation and the bootstrap are examples of this approach. They directly estimate $R(M)$.

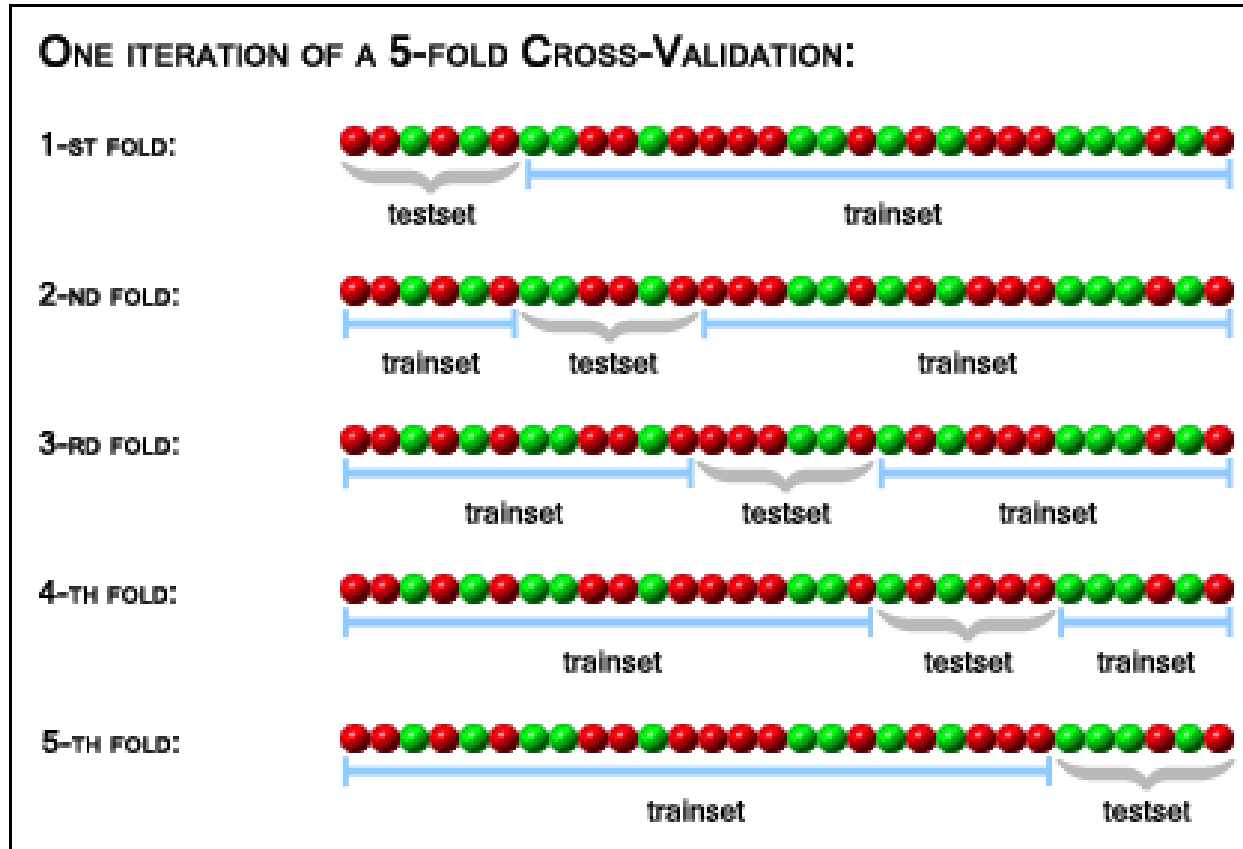
Cross-Validation

V-fold cross-validation splits the data into V roughly equal-sized chunks, using each chunk in turn as the validation set while taking the remainder as the training set.

Denote by $\hat{f}_M^{-v}(x)$ the estimate using the model M trained on the data with the v^{th} part removed. Then the cross-validation *estimate of the test error* is

$$\text{CV}(M) = \frac{1}{N} \sum_{v=1}^V \sum_{i=1}^{N_v} L \left(y_i, \hat{f}_M^{-v}(x_i) \right). \quad (14)$$

Cross-Validation



V -fold cross-validation requires running V training optimizations.

Cross-Validation: Choice of V ?

$V = N$ is called *leave-one-out cross-validation* (LOOCV). It is approximately unbiased, though some argue it can have high variance. LOOCV and AIC can be shown to be asymptotically approximately equivalent.

It is computationally intensive in general. For linear models of the form $\hat{y} = Ay$ for some matrix A under squared-error loss, there is a convenient approximation called *generalized cross-validation*.

Having too small a value for V will overestimate $R(M)$, because smaller training sets yield poorer estimators. Often $V = 10$ is chosen as a compromise.

Main Things You Should Know

- What a mixture of Gaussians is
- What the EM algorithm is for, and its iterative form
- What AIC is
- What cross-validation is

Nonparametric Estimation

What if I don't want to specify a simple parametric form?

Nonparametric Estimation

What exactly do we mean by “nonparametric”? Example of a nonparametric model class, called a *Sobolev space*:

$$\mathcal{F} = \left\{ f : \int (f''(x))^2 dx < \infty \right\} \quad (15)$$

“Nonparametric” doesn’t mean there are no parameters. There is typically a local “model”. It refers to model classes, like the one above, which aren’t parametric (having finite number of parameters). We sometimes say such a class is *distribution-free*.

A *nonparametric method* is one for which we can pretend the model class is actually such a class, as far as its asymptotic properties.

Examples of Nonparametric Methods

Some examples of popular nonparametric methods:

- Histogram, kernel density estimation (density estimation)
- Splines, wavelet regression (regression)
- Kernel discriminant analysis, nearest neighbor, support vector machines (classification)

Histogram

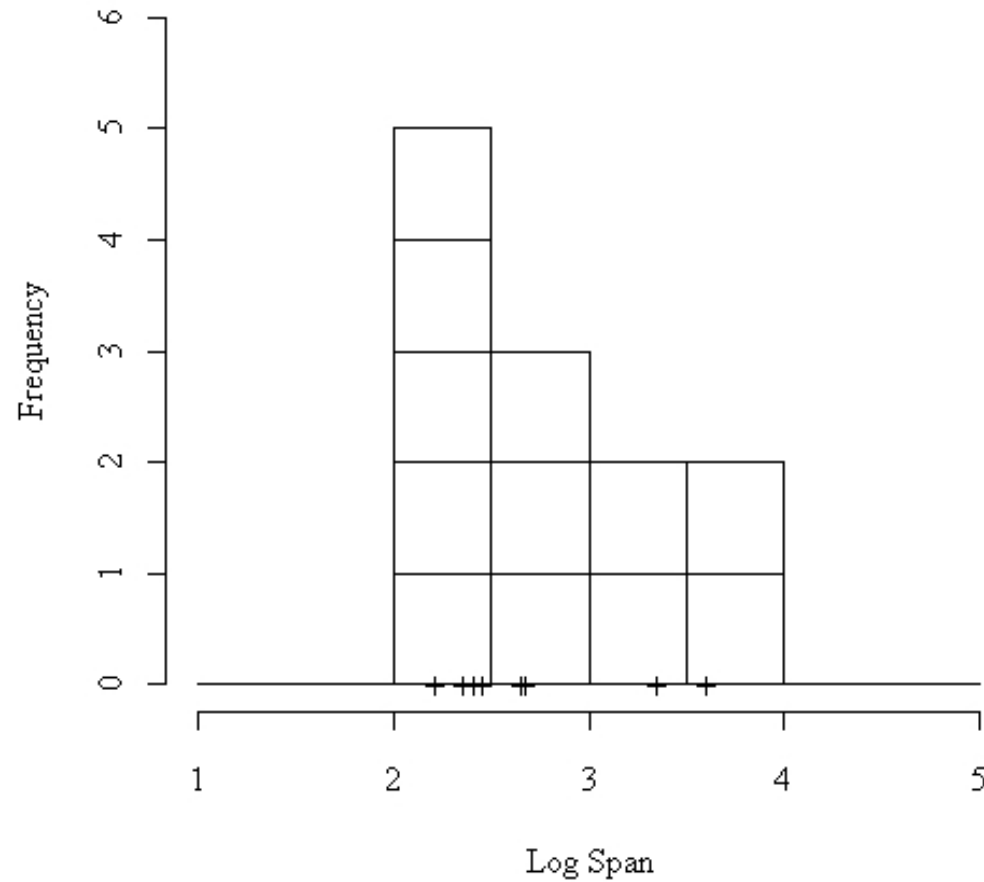
Perhaps the simplest nonparametric density estimator is the *histogram*:

$$\hat{f}_N(x) = \sum_{j=1}^m \frac{\hat{p}_j}{h} I(x \in B_j) \quad (16)$$

where $h = 1/m$ is the *binwidth*, Y_j is the number of observations in bins $B_1 = [0, \frac{1}{m})$, $B_2 = [\frac{1}{m}, \frac{2}{m})$, \dots , $\hat{p}_j = Y_j/N$, and $p_j = \int_{B_j} f(u)du$.

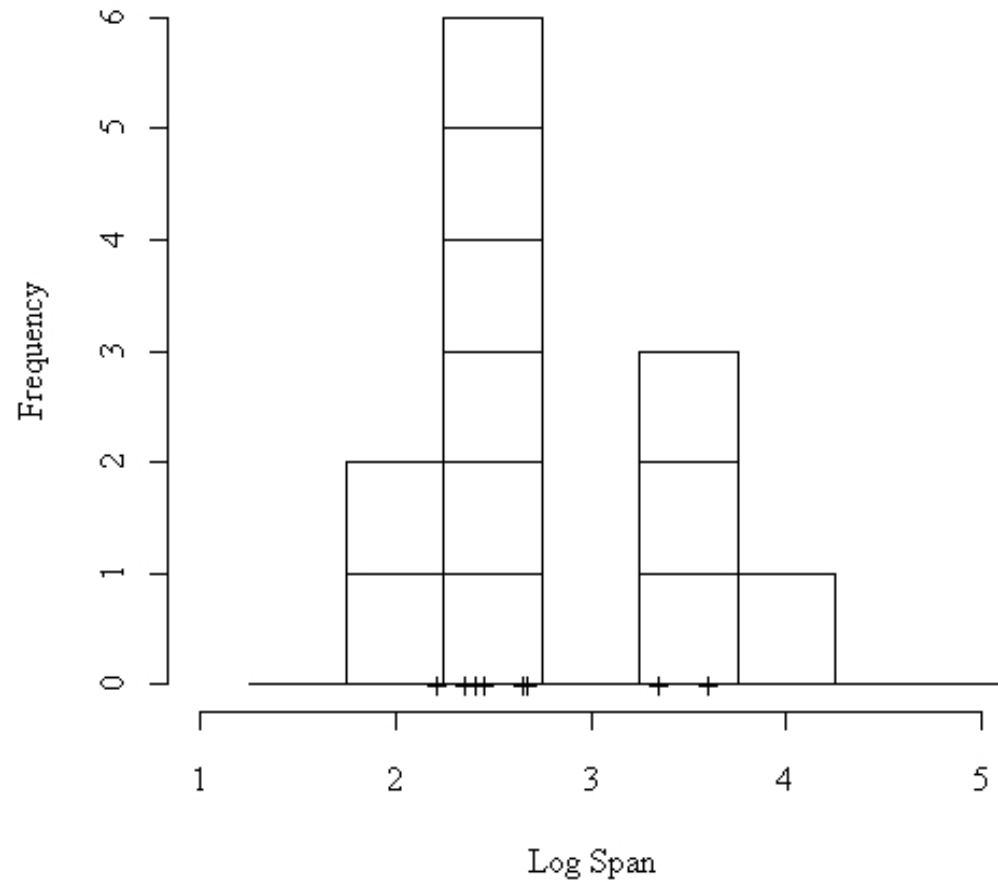
Histogram

**Histogram with breaks at n.0 and n.5
binwidth=0.5**



Histogram

**Histogram with breaks at n.25 and n.75
binwidth=0.5**



Histogram

Note a few things. First, the placement of the bins (*i.e.* shifting a bit to the left or right) can make a significant qualitative difference. Second, the density estimate is not smooth.

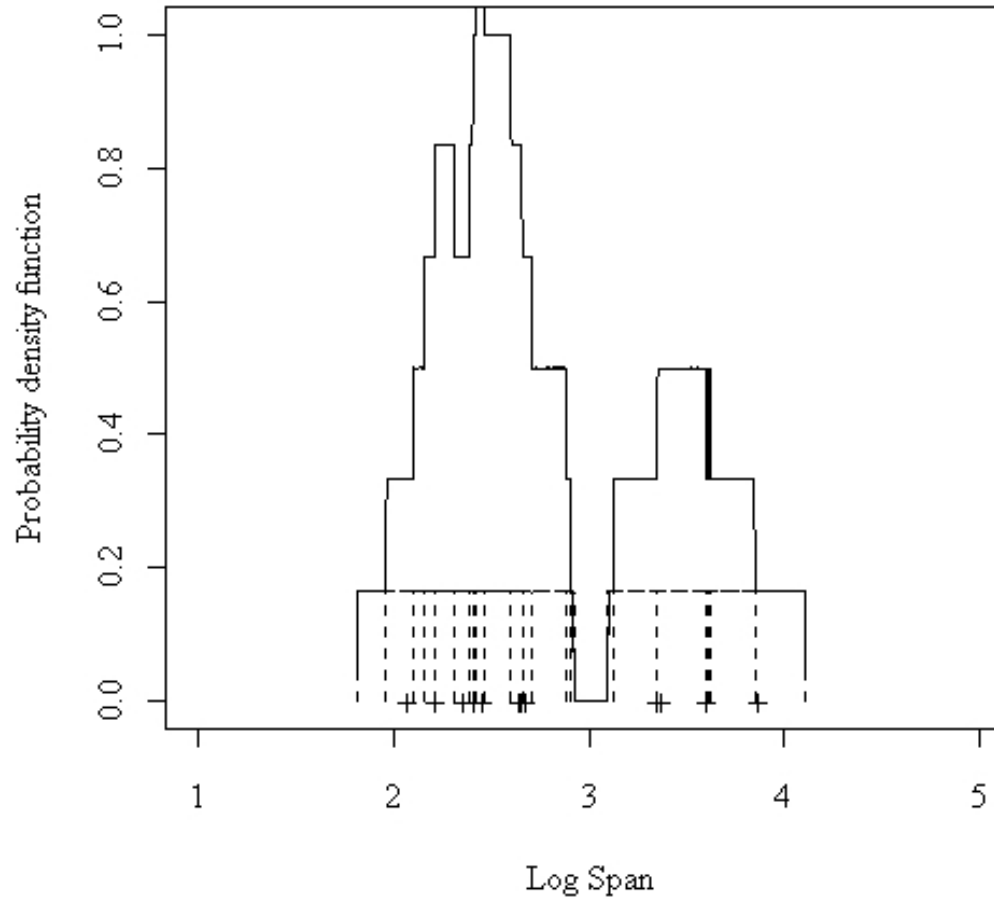
Nonetheless, we can show that $\mathbb{E}(\hat{f}_N(x)) \approx f(x)$, under certain conditions. Remarkable, but we can do better.

Kernel Density Estimation

How can I estimate a density nonparametrically?

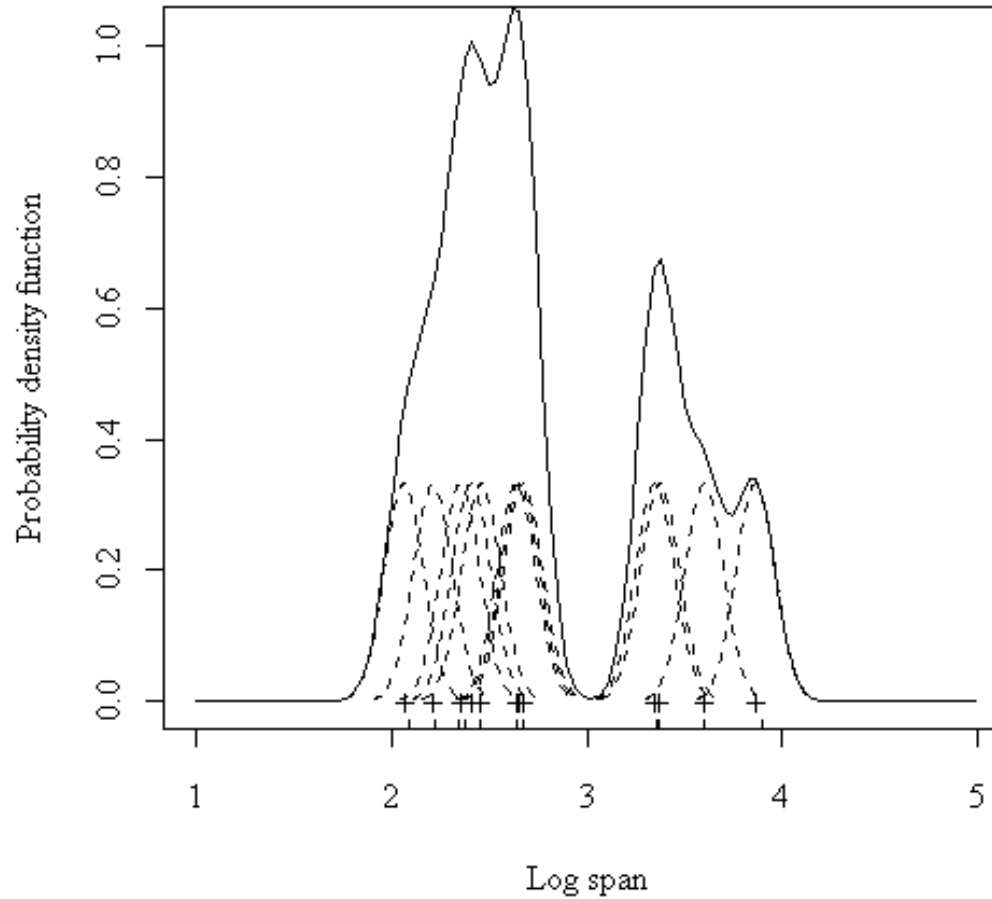
Kernel Density Estimator

'Histogram' with blocks centred over data points



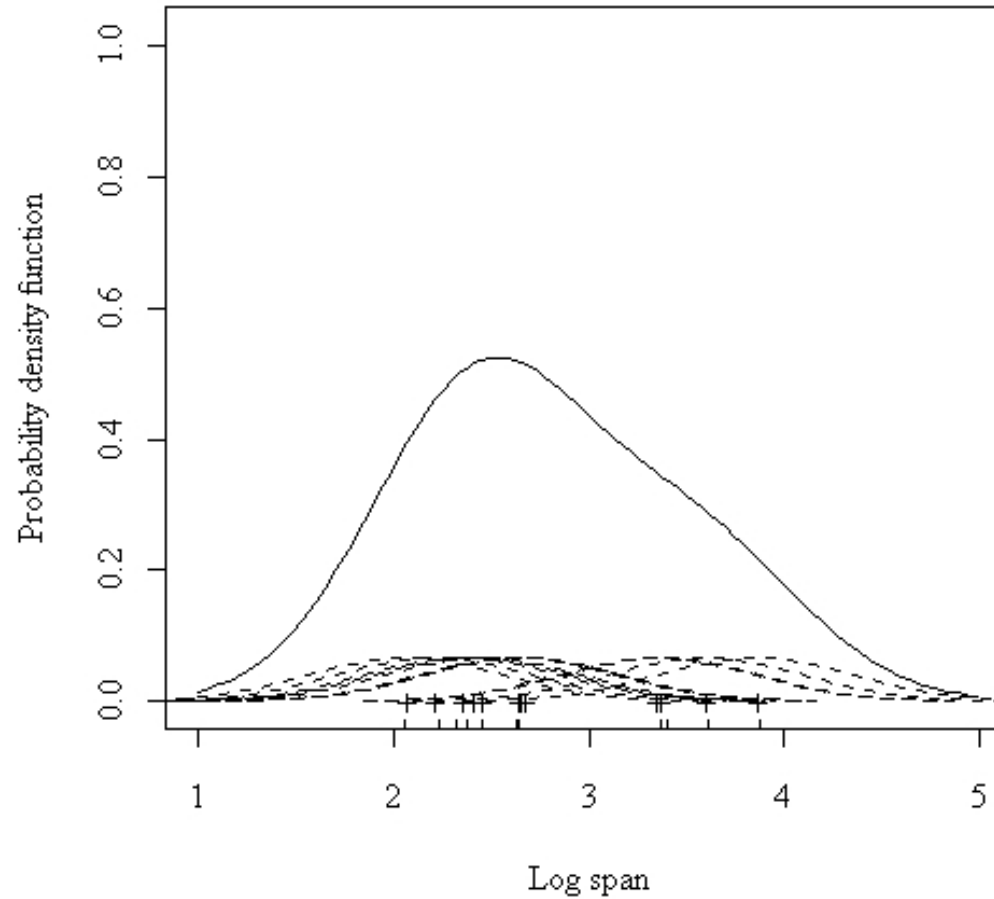
Kernel Density Estimator

Undersmoothed



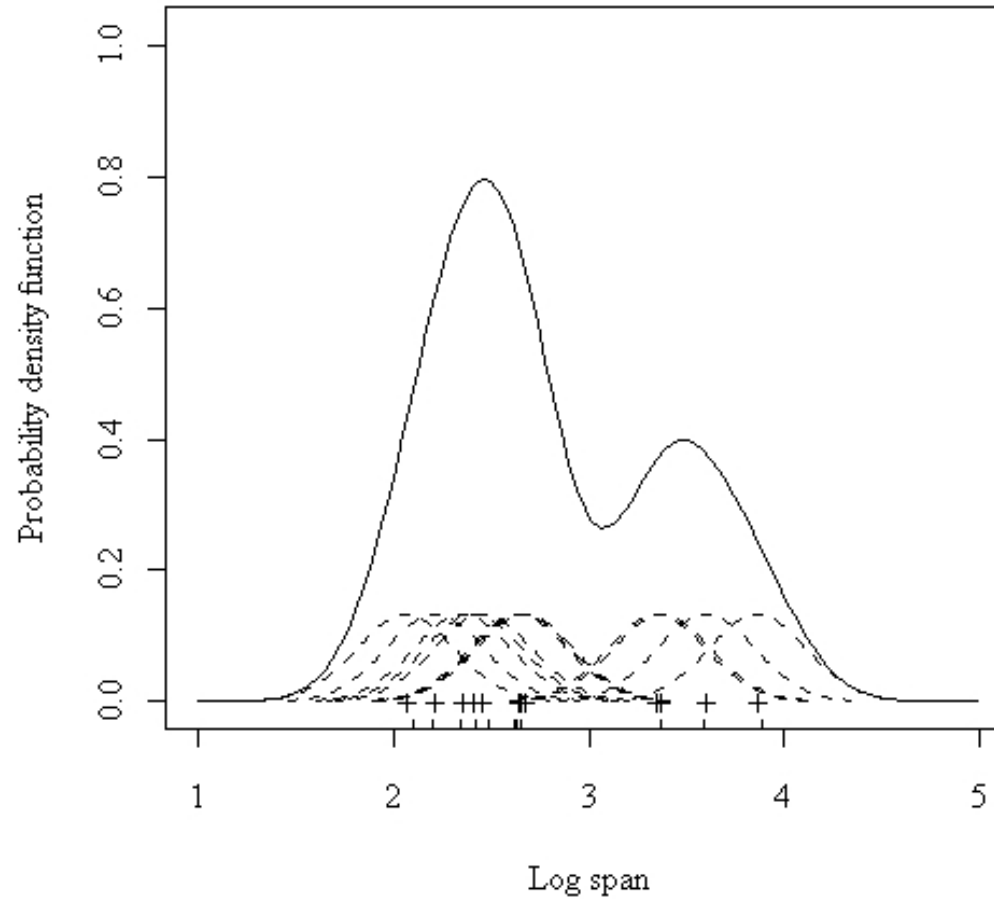
Kernel Density Estimator

Oversmoothed



Kernel Density Estimator

Optimally smoothed



Histogram Versus KDE

By centering the blocks on each data point, and generalizing to a smooth kernel function from the block (which we'll now call the *rectangular kernel*), we have a more satisfying density estimator, called the *kernel density estimator* (KDE). We also saw both histograms and KDE *have a parameter*, the kernel width, and that its proper choice is critical.

We will be able to show asymptotically that KDE is a better estimator of the density than the histogram, and be able to specify a procedure for choosing the optimal kernel width in KDE (as well as the optimal bin width for a histogram).

Kernel Density Estimator

The kernel density estimator is defined as

$$\hat{f}_N(x) = \frac{1}{N} \sum_{i=1}^N \frac{1}{h} K\left(\frac{x - X_i}{h}\right), \quad (17)$$

where the *kernel function* is any smooth function K such that $K(u) \geq 0$, $\int K(u)du = 1$, $\int uK(u)du = 0$, and $\sigma_K^2 = \int u^2 K(u)du > 0$, and its parameter h is called the *bandwidth*.

An example is the Gaussian kernel $K(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}$.

Kernel Density Estimator

The parts of KDE:

Task: density estimation

Model class: Sobolev space

Loss: L_2 error

Optimizer: exhaustive or gradient descent

Generalization mechanism: cross-validation

Evaluation algorithm: *generalized N -body algorithm*

Note that we have changed from the likelihood, which we used in our parametric example, mixtures of K Gaussians, to L_2 error. We will return to the reason for this. Also note the need for a fast evaluation algorithm, which we will discuss in a later lecture.

L_2 : MSE, MISE

Suppose $\hat{f}_N(x)$ is an estimate of a function $f(x)$. The *squared error* (or L_2) loss is

$$L(f(x), \hat{f}_N(x)) = (f(x) - \hat{f}_N(x))^2. \quad (18)$$

The average of any loss is called the *risk* or in this case the *mean squared error*

$$\text{MSE} = R(f(x), \hat{f}_N(x)) = \mathbb{E}L(f(x), \hat{f}_N(x)). \quad (19)$$

To summarize the risk over all values of x , we use the *integrated risk* or in this case the *mean integrated squared error* (MISE):

$$\text{MISE} = R(f, \hat{f}_N) = \int R(f(x), \hat{f}_N(x)) dx. \quad (20)$$

L_2 : Bias-Variance Tradeoff

For L_2 loss we have a convenient decomposition (dropping reference to x and N for the moment):

$$\mathbb{E} \left(f - \hat{f} \right)^2 \quad (2)$$

$$= \mathbb{E} \left(f - \mathbb{E}\hat{f} + \mathbb{E}\hat{f} - \hat{f} \right)^2 \quad (2)$$

$$= \mathbb{E} \left(\left(f - \mathbb{E}\hat{f} \right) + \left(\mathbb{E}\hat{f} - \hat{f} \right) \right)^2 \quad (2)$$

$$= \mathbb{E} \left(f - \mathbb{E}\hat{f} \right)^2 + \mathbb{E} \left(\mathbb{E}\hat{f} - \hat{f} \right)^2 + 2\mathbb{E} \left(\left(\mathbb{E}\hat{f} - \hat{f} \right) \left(f - \mathbb{E}\hat{f} \right) \right) \quad (2)$$

$$= \text{bias}^2(\hat{f}) + \mathbb{V}(\hat{f}) + 2 \left(\mathbb{E} \left(f\mathbb{E}\hat{f} \right) - \mathbb{E} \left(\mathbb{E}\hat{f}^2 \right) - \mathbb{E}f\hat{f} + \mathbb{E} \left(\hat{f}\mathbb{E}\hat{f} \right) \right) \quad (2)$$

$$= \text{bias}^2(\hat{f}) + \mathbb{V}(\hat{f}) + 2 \left(f\mathbb{E}\hat{f} - \mathbb{E}\hat{f}^2 - f\mathbb{E}\hat{f} + \mathbb{E}\hat{f}^2 \right) \quad (2)$$

$$= \text{bias}^2(\hat{f}) + \mathbb{V}(\hat{f}) \quad (2)$$

L_2 : Bias-Variance Tradeoff

...by just using a “completing the square” trick and the properties of expectation. Back to our original notation, we have that

$$R(f(x), \hat{f}_N(x)) = \text{bias}^2(\hat{f}_N(x)) + \mathbb{V}(\hat{f}_N(x)) \quad (28)$$

where

$$\text{bias}(\hat{f}_N(x)) = \mathbb{E}(\hat{f}_N(x)) - f(x). \quad (29)$$

KDE Consistency

Assume that f is continuous at x and that $h_N \rightarrow 0$ and $Nh_N \rightarrow \infty$ as $N \rightarrow \infty$. Then

$$\hat{f}_N(x) \xrightarrow{p} f(x) \quad \text{as } N \rightarrow \infty. \quad (30)$$

Note that the bandwidth must shrink as we get more data, but not go to zero as rapidly as $1/N$, *i.e.* the expected number of points falling in the interval $x \pm h_N$ must tend to infinity, however slowly, as N tends to infinity.

A stronger notion, *uniform consistency*, can also be shown, under some conditions that are only slightly stronger:

$$\sup_x \left| \hat{f}_N(x) - f(x) \right| \xrightarrow{p} 0 \quad \text{as } N \rightarrow \infty. \quad (31)$$

KDE Risk

Now, we're going to come up with a detailed expression for the risk of a kernel density estimator. This will let us conclude some important things.

Let $R(x) = \mathbb{E}(\hat{f}_N(x) - f(x))^2$ be the risk at a point x and $R = \int R(x)dx$ denote the integrated risk. Assume that f'' is absolutely continuous and that $\int (f''(x))^2 dx < \infty$.

Also recall our assumptions about K . We'll write $K_h(x, X) = \frac{1}{h} K((x - X)/h)$.

KDE Risk

For our estimator $\hat{f}_N(x) = \frac{1}{N}K_h(x, X)$ we have

$$\mathbb{E}(\hat{f}_N(x)) = \mathbb{E}(K_h(x, X)) \quad (32)$$

$$= \int \frac{1}{h} K\left(\frac{x-t}{h}\right) f(t) dt \quad (33)$$

$$= \int K(u) f(x-hu) du \quad (34)$$

$$= \int K(u) \left[f(x) - hu f'(x) + \frac{1}{2} h^2 u^2 f''(x) + \dots \right] du \quad (35)$$

$$= f(x) + \frac{1}{2} h^2 f''(x) \int u^2 K(u) du + \dots \quad (36)$$

since $\int K(x) dx = 1$ and $\int x K(x) dx = 0$.

KDE Risk

Then the bias is

$$\mathbb{E}(\hat{f}_N(x)) - f(x) = \frac{1}{2}\sigma_K^2 h_N^2 f''(x) + O(h_N^4). \quad (37)$$

By a similar calculation, the variance is

$$\mathbb{V}(\hat{f}_N(x)) = \frac{f(x) \int K^2(x) dx}{N h_N} + O\left(\frac{1}{N}\right). \quad (38)$$

Putting them together we have

$$R = \frac{1}{4}\sigma_K^4 h_N^4 (f''(x))^2 + \frac{f(x) \int K^2(x) dx}{N h_N} + O\left(\frac{1}{N}\right) + O(h_N^4). \quad (39)$$

KDE Optimal Bandwidth

If we differentiate the risk with respect to h and set it equal to 0, we see that the asymptotically optimal bandwidth is

$$h^* = \left[\left(\frac{\int K(x)^2 dx}{(\int x^2 K(x) dx)^2 \int (f''(x))^2 dx} \right) \frac{1}{N} \right]^{1/5}. \quad (40)$$

So the best bandwidth decreases at rate $N^{-1/5}$.

Effectively balances bias and variance.

KDE Convergence Rate

Now if we plug h^* into the risk, we see that if the optimal bandwidth is used then $R = O(N^{-4/5})$.

It can be shown that histograms converge at rate $O(N^{-2/3})$.

It turns out there does not exist a density estimator that converges faster than $O(N^{-4/5})$.

Main Things You Should Know

- What it means to be nonparametric
- What kernel density estimation (KDE) and its parts are
- What asymptotic properties we can show about KDE
- What the bias-variance tradeoff is