Density Estimation

James M. Flegal

Agenda

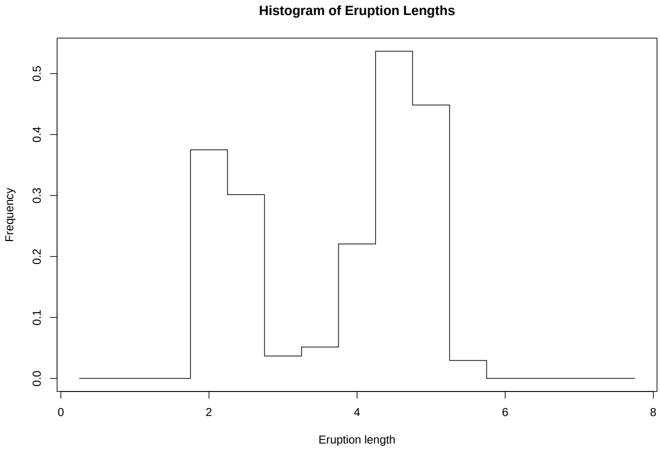
- Histograms
- Glivenko-Cantelli theorem
- Error for density estimates
- Kernel density estimates
- Bivariate density estimates

Histograms

- Histograms are one of the first things learned in "Introduction to Statistics"
- Simple way of estimating a distribution
  - Split the sample space up into bins
  - Count how many samples fall into each bin
- If we hold the bins fixed and take more and more data, then the relative frequency for each bin will converge on the bin's probability

Example: Old Faithful Geyser Data

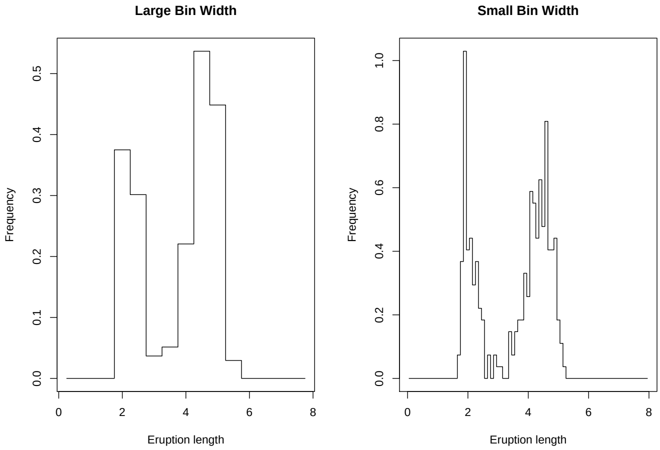
```
data(faithful); x0 <- 0; x1 <- 8; h <- .5
my.breaks<-seq(from=x0, to=x1, by=h)
myhist<-hist(faithful$eruptions, breaks=my.breaks, right=F, plot=F)
plot(myhist$mids, myhist$density, type="s", xlab="Eruption length",
     ylab="Frequency", main="Histogram of Eruption Lengths")
```



Histograms

- What about a density function?
- Could take our histogram estimate and say that the probability density is uniform within each bin
- Gives us a piecewise-constant estimate of the density
- ***Problem***: Will not converge on the true density – unless we shrink the bins as we get more and more data
- Bias-variance trade-off
  - A large number of very small bins, the minimum bias in our estimate of any density becomes small
  - But the variance grows for very small bins

Example: Old Faithful Geyser Data



Histograms

Bin width primarily controls the amount of smoothing, lots of guidance available

1. **Sturges' rule**: Optimal width of class intervals is given by
$$\frac{R}{1 + \log_2 n}$$
where R is the sample range – Designed for data sampled from symmetric, unimodal populations
2. **Scott's Normal reference rule**: Specifies a bin width
$$3.49\hat{\sigma} n^{-1/3}$$
where $\hat{\sigma}$ is an estimate of the population standard deviation σ
3. **Freedman-Diaconis rule**: Specifies the bin width to be
$$2(IQR)n^{-1/3}$$
where the IQR is the sample inter-quartile range

Histograms

- Is learning the whole distribution non-parametrically even feasible?
- How can we measure error to deal with the bias-variance trade-off?

Empirical CDF

- Learning the whole distribution is ***feasible***
- Something even dumber than shrinking histograms will work
- Suppose we have one-dimensional samples $x_1, \ldots, x_n$ with CDF F
- Define the empirical cumulative distribution function on n samples as

$$\hat{F}_n(a) = \frac{1}{n} \sum_{i=1}^{n} I(-\infty < x_i \leq a)$$

- Just the fraction of the samples less than or equal to a

Glivenko-Cantelli theorem

- Then the ***Glivenko-Cantelli theorem*** says

$$\max_a |\hat{F}_n(a) - F(a)| \to 0$$

- So the empirical CDF converges to the true CDF ***everywhere***, i.e. the maximum gap between the two of them goes to zero
- Pitman (1979) calls this the "fundamental theorem of statistics"
- Can learn distributions just by collecting enough data

Glivenko-Cantelli theorem

- Can we use the empirical CDF to estimate a density?
- Yes, but it's discrete and doesn't estimate a density well
- Usually we can expect to find some new samples between our old ones
- So we want a non-zero density between our observations
- Uniform distribution within each bin of a histogram doesn't have this issue
- Can we do better?

Error for density estimates

- Yes, but what do we mean by "better" density estimates?
- Three ideas:
  1. Squared deviation from the true density should be small

  $$\int \left(f(x) - \hat{f}(x)\right)^2 dx$$

  2. Absolute deviation from the true density should be small

  $$\int \left|f(x) - \hat{f}(x)\right| dx$$

  3. Average log-likelihood ratio should be kept low

  $$\int f(x) \log \frac{f(x)}{\hat{f}(x)} dx$$

Error for density estimates

- Squared deviation is similar to MSE criterion used in regression
  - Used most frequently since it's mathematically tractable
- Absolute deviation considers L_1 or total variation distance between the true and the estimated density
  - Nice property that $\frac{1}{2}\int \left| f(x) - \hat{f}(x) \right| dx$ is exactly the maximum error in our estimate of the probability of any set
  - But it's tricky to work with, so we'll skip it
- Minimizing the log-likelihood ratio is intimately connected both to maximizing the likelihood and to minimizing entropy
  - Called Kullback-Leibler divergence or relative entropy

Error for density estimates

- Notice that

$$\int \left(f(x) - \hat{f}(x)\right)^2 dx = \int f^2(x)dx - 2\int f(x)\hat{f}(x)dx + \int \hat{f}^2(x)dx$$

- First term doesn't depend on the estimate, so we can ignore it for purposes of optimization
- Third term only involves $\hat{f}(x)$, and is just an integral, which we can do numerically
- Second term involves both the true and the estimated density; we can approximate it using Monte Carlo by

$$\frac{2}{n}\sum_{i=1}^{n} \hat{f}(x_i)$$

Error for density estimates

- Then our error measure is

$$\frac{2}{n}\sum_{i=1}^{n}\hat{f}(x_i) + \int \hat{f}^2(x)dx$$

- In fact, this error measure does not depend on having one-dimension data
- For purposes of cross-validation, we can estimate $\hat{f}(x)$ on the training set and then restrict the sum to points in the testing set

Naive estimator

- If a random variable X has probability density f, then

$$f(x) = \lim_{h \to 0} \frac{1}{2h} P(x - h < X < x + h)$$

- Thus, a naive estimator would be

$$\widehat{f}(x) = \frac{1}{2nh} \left[\# \text{ of } x_i \text{ falling in } (x - h, x + h)\right]$$

Naive estimator

- Or, equivalently

$$\widehat{f}(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}w\left(\frac{x - x_i}{h}\right)$$

where w is a weight function defined as

$$w(x) = \begin{cases} 1/2 & |x| < 1 \\ 0 & \text{otherwise} \end{cases}$$

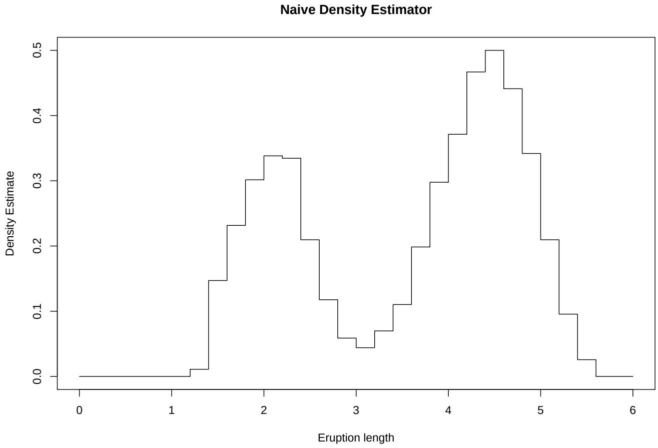
- In short, a naive estimate is constructed by placing a box of width $2h$ and height $\frac{1}{2nh}$ on each observation, then summing to obtain the estimate

Example: Old Faithful Geyser Data

```r
my.w<-function(x){
    if (abs(x) < 1)
        w <- 1/2
    else
        w <- 0
    w
    }
x <- seq(0,6,0.2)
m <- length(x)
n <- length(faithful$eruptions)
h <- .5
fhat <- rep(0,m)
for (i in 1:m){
    S <- 0
    for (j in 1:n){
        S <- S+(1/h)*my.w((faithful$eruptions[j]-x[i])/h)
        }
    fhat[i] <- (1/n)*S
    }
```

Example: Old Faithful Geyser Data

```
plot(x,fhat, type="s", xlab="Eruption length", ylab="Density Estimate",
     main="Naive Density Estimator")
```



Naive estimator

- Not wholly satisfactory, from the point of view of using density estimates for presentation
- Estimate $\widehat{f}$ is a step function
- In the formula for the naive estimate, we can replace the weight function w by another function K with more desirable properties
- Function K is called a **kernel**

Kernel density estimates

- Resulting estimate is a **kernel estimator**:

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h} K\left(\frac{x - x_i}{h}\right).$$

- h is the **window width**, **smoothing parameter**, or **bandwidth**
- Usually the kernel K is taken to be a probability density function itself (i.e., normal density)
- Resulting estimate inherit smoothness properties of K

Kernel density estimates

Most popular choices for the kernel K are

Family	Kernel
Gaussian	$K(t) = \frac{1}{\sqrt{2\pi}} e^{-t^2/2}$
Rectangular	$K(t) = 1/2$ for $\lvert t\rvert < 1$
Triangular	$K(t) = 1 - \lvert t\rvert$ for $\lvert t\rvert < 1$
Epanechnikov	$K(t) = \frac{3}{4}(1 - (1/5)t^2)$ for $\lvert t\rvert < \sqrt{5}$

Kernel density estimates

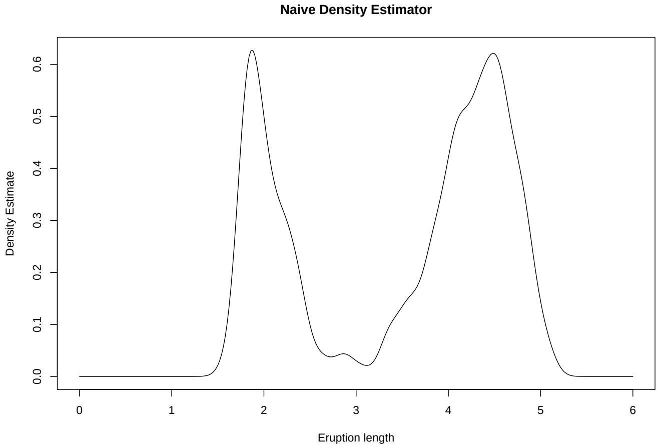
```
my.w<-function(x, type="gaussian"){
if(type=="gaussian"){
  w <- dnorm(x)
return(w)
}
if(type=="naive"){
  if (abs(x) < 1)
        w <- 1/2
    else
        w <- 0
return(w)
}
print("You have asked for an undefined kernel.")
return(NULL)
}
```

Example: Old Faithful Geyser Data

```
x <- seq(0,6,0.02)
m <- length(x)
n <- length(faithful$eruptions)
h <- .1
fhat <- rep(0,m)
for (i in 1:m){
    S <- 0
    for (j in 1:n){
        S <- S+(1/h)*my.w((faithful$eruptions[j]-x[i])/h)
        }
    fhat[i] <- (1/n)*S
    }
```

Example: Old Faithful Geyser Data

```
plot(x,fhat, type="l", xlab="Eruption length", ylab="Density Estimate",
     main="Naive Density Estimator")
```



Bandwidth selection

- Cross-validation, which could be time consuming
- Optimal bandwidth for a Gaussian kernel to estimate a Gaussian distribution is $1.06\sigma/n^{1/5}$
- Called the **Gaussian reference rule** or the **rule-of-thumb** bandwidth
- When you call `density` in R, this is basically what it does

Kernel density estimate samples

- There are times when one wants to draw a random sample from the estimated distribution
- Easy with kernel density estimates, because each kernel is itself a probability density
- Suppose the kernel is Gaussian, that we have scalar observations $x_1, \ldots, x_n$ and bandwidth h
    1. Pick an integer uniformly at random from 1 to n
    2. Use `rnorm(1,x[i],h)`, or `rnorm(q,sample(x,q,replace=TRUE),h)` for q draws
- Using a different kernel, we'd just need to use the random number generator function for the corresponding distribution

Other Approaches

- Histograms and kernels are not the only possible way of estimating densities
- Can try the local polynomial trick, series expansions, splines, penalized likelihood approaches, etc
- For some of these, avoid negative probability density estimates using the log density

Density estimation in R

- `density()` function is the most common

```
density(x, ...)
## Default S3 method:
density(x, bw = "nrd0", adjust = 1,
        kernel = c("gaussian", "epanechnikov", "rectangular",
                   "triangular", "biweight",
                   "cosine", "optcosine"),
        weights = NULL, window = kernel, width,
        give.Rkern = FALSE,
        n = 512, from, to, cut = 3, na.rm = FALSE, ...)
```

- `ASH` and `KernSmooth` are both fast, accurate, and well-maintained (Deng and Wickham, 2011)

Bivariate density estimation

- To construct a bivariate density histogram, it is necessary to define two-dimensional bins and count the number of observations in each bin
- Can use `bin2d` function in R will bin a bivariate data set

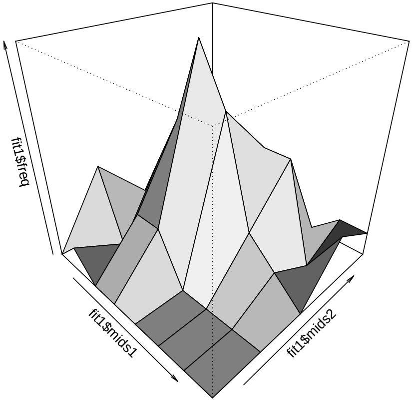
```
bin2d <- function(x, breaks1 = "Sturges", breaks2 = "Sturges"){
      histg1 <- hist(x[,1], breaks = breaks1, plot = FALSE)
      histg2 <- hist(x[,2], breaks = breaks2, plot = FALSE)
      brx <- histg1$breaks
      bry <- histg2$breaks
      freq <- table(cut(x[,1], brx),  cut(x[,2], bry))

      return(list(call = match.call(), freq = freq,
                breaks1 = brx, breaks2 = bry,
                mids1 = histg1$mids, mids2 = histg2$mids))
      }
```

Bivariate density estimation

- Following example computes the bivariate frequency table
- After binning the data, the `persp` function plots the density histogram

```
data(iris)
fit1=bin2d(iris[1:50, 1:2])
persp(x=fit1$mids1, y=fit1$mids2, z=fit1$freq, shade=T, theta=45, phi=30, ltheta=60)
```



Bivariate kernel methods

- Suppose the data is $X_1, \ldots, X_n$, where each $X_i \in \mathbb{R}^2$
- Kernel density estimates can be extended to a multivariate (bivariate) setting
- Let $K(\cdot)$ be a bivariate kernel (typically a bivariate density function), then bivariate kernel density estimate is

$$\hat{f}(X) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{X - X_i}{h}\right)$$

Example: Bivariate normal

- Estimate the bivariate density when the data is generated from a mixture model with three components with identical covariance $\Sigma = I_2$ and different means

$$\mu_1 = (0, 0) \quad \mu_2 = (1, 3), \quad \mu_3 = (4, -1).$$

- Mixture probabilities are $p = (0.2, 0.3, 0.5)$

Example: Bivariate normal

```r
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.0.2
```

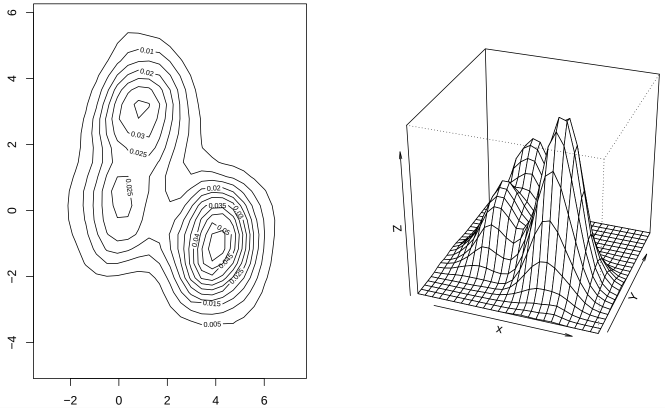
```r
n <- 2000
p <- c(.2, .3, .5)
mu <- matrix(c(0, 1, 4, 0, 3, -1), 3, 2)
Sigma <- diag(2)
i <- sample(1:3, replace = TRUE, prob = p, size = n)
k <- table(i)

x1 <- mvrnorm(k[1], mu = mu[1,], Sigma)
x2 <- mvrnorm(k[2], mu = mu[2,], Sigma)
x3 <- mvrnorm(k[3], mu = mu[3,], Sigma)
X <-  rbind(x1, x2, x3)   #the mixture data
x <- X[,1]
y <- X[,2]

fhat <- kde2d(x, y, h=c(1.87, 1.84))
```

Example: Bivariate normal

```r
par(mfrow = c(1, 2))
contour(fhat)
persp(fhat, phi = 30, theta = 20, d = 5, xlab = "x")
```




```r
par(mfrow = c(1, 1))
```

Summary

- Over 20 packages that perform density estimation (Deng and Wickham, 2011)
- Kernel density estimation is the most common approach
- Density estimation can be parametric, where the data is from a known family
- Bayesian approaches are also available