BMEG 802 – Advanced Biomedical Experimental Design and Analysis

Maximum Likelihood Estimation

Joshua G. A. Cashaback, PhD

Recap

- ANCOVA
  - covariates
  - can use for any combination of between and within designs.

Today

- Maximum Likelihood Estimation (MLE)
  - Probability Distribution Function
  - Likelihood function
  - 3 Ways to find the Maximum Likelihood Estimation
    - Analytical (Calculus)
    - Brute Force (Grid Search)
    - Optimization (Gradient Descent)

Maximum Likelihood Estimation

- Tool for parameter estimation
- good approach for cases when OLS (ordinary least squares) assumptions are violated
- e.g. for non-linear models with non-normal data
- in MLE, we estimate the parameters of a model that maximize the likelihood of your data

Probability Density Function

- assume an observed **data** vector

$$y = (y_1, y_2, ..., y_n)$$

- Goal of MLE: identify the population (the model) that is **most likely** to have generated the data

Probability Density Function

- Here we assume population (model) is associated with a corresponding probability distribution
- Each probability distribution is characterized by a unique value of the model's parameter(s)
- As model parameters change, different probability distributions are generated
- Model = the family of probability distributions indexed by the model's parameter(s)

Probability Density Function

- $f(y|w)$ is the probability density function (PDF) specifying the probability of observing data y, given model parameter(s) w
  - note: w may be a parameter vector, $w = (w_1, w_2, ..., w_n)$
    - e.g. for a normal PDF: $w = (\mu, \sigma)$

Probability Density Function

- If observations yi are i.i.d. (indepedent and identically distributed), then the PDF for the data as a whole, $y = (y_1, y_2, ..., y_n)$ given the parameter vector $\mathbf{w} = (w_1, w_2, ..., w_n)$, can be expressed as the multiplication of PDFs for individual observations:

$$f(y_1, y_2, ..., y_n|\mathbf{w}) = f_1(y_1|\mathbf{w})f_2(y_2|\mathbf{w}), ..., f_n(y_n|\mathbf{w})$$

Or, more concisely $f(\mathbf{y}|\mathbf{w}) = \prod_{i=1}^{n} f_n(y_n|\mathbf{w})$\$

PDF Example with a Normal Distribution

- Let's say our data vector Y is made up of 3 observations:
  $y_1 = 80, y_2 = 110, y_3 = 130$
- We want to compute the PDF for a Normal distribution:

$$f(y_i|\mu,\sigma)\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{y_i-\mu}{\sigma}\right)^2}$$

Let's assume $\mu = 100, \sigma = 15$

$$f(80|\mu = 100, \sigma = 15)\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{80-\mu}{\sigma}\right)^2} = 0.010934$$

$$f(110|\mu = 100, \sigma = 15)\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{100-\mu}{\sigma}\right)^2} = 0.010934$$

$$f(130|\mu = 100, \sigma = 15)\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{130-\mu}{\sigma}\right)^2} = 0.010934$$

$$f(y_1, y_2, y_3|\mu,\sigma) = f(y_1|\mu,\sigma)f(y_2|\mu,\sigma)f(y_3|\mu,\sigma) = (.010934)(.021297)(.003599) = .000000838$$

Binomial Distribution Example

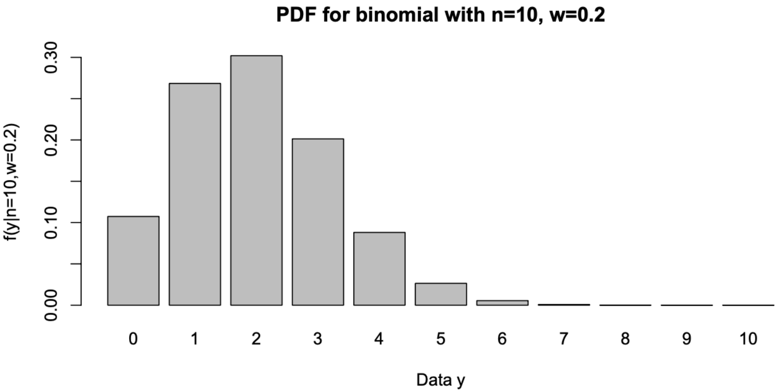
- y is the number of successes in a sequence of 10 Bernoulli trials (e.g. tossing a coin 10 times)
- a Bernoulli trial is an experiment whose outcome is random and can be either of two possible outcomes: success or failure.
- Binomial Distribution PDF:

$$f(y|n, w) = \frac{n!}{y!(n-y)!} w^y (1-w)^{n-y}$$

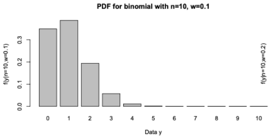
- assume probability of a success on any one trial is 0.2 (a biased coin)
- parameter vector w is n=10, w=0.2

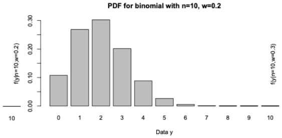
$$f(y|n=10, w=0.2) = \frac{10!}{y!(10-y)!} 0.2^y (1-0.2)^{10-y}; (y=0,1,...,10)$$

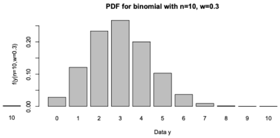
Binomial Distribution

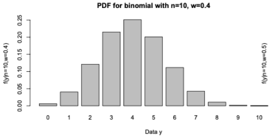


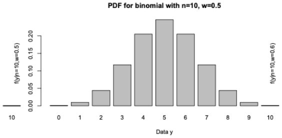
Binomial Distribution - Varying a Parameter

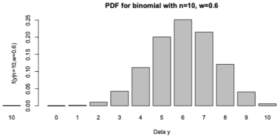


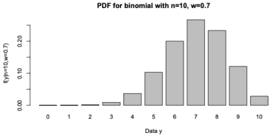








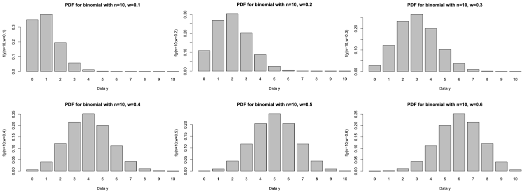


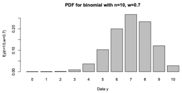


and so on ...

Binomial Distribution - A Model

The collection of all such PDFs generated by varying the parameter across its range defines a **model**





and so on ...

Likelihood Function

Likelihood Function

- Given a set of parameter values, the corresponding PDF will show that some data are more probable than other data
- In fact we have already observed the data

Likelihood Function

- We are faced with the inverse problem
- Given the observed data, and a model of the process by which the data was generated
  - find the one PDF, among all the probability densities that the model prescribes, that is **most likely to have produced the data**

Likelihood Function

- we define the likelihood function by reversing the roles of the data vector y and the parameter vector w in $f(y|w)$:

$$\mathcal{L}(w|y) = f(y|w)$$

$\mathcal{L}(w|y)$ represents the likelihood of the parameter w given the observed data y

- note: a likelihood function does not need to sum to 1.0
- For our one-dimensional binomial example the likelihood function for y=7 and n=10 is

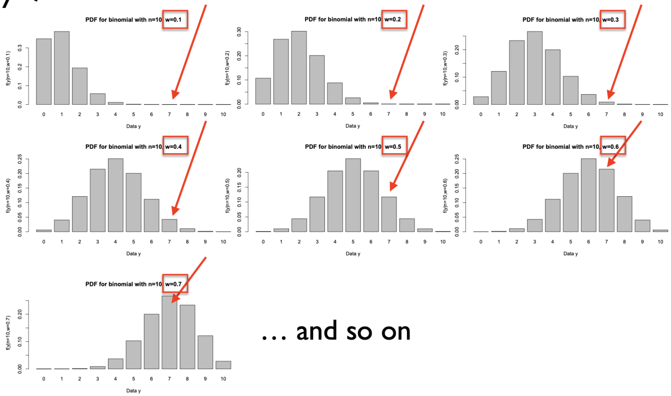
$$\mathcal{L}(w|n=10, y=7) = \frac{10!}{7!(10-7)!} w^7 (1-w)^{10-7}; (0 \leq w \leq 1)$$

But, what is the value of w???

Likelihood Function - Iterate Through Variable

Let's try all value of w between 0 and 1

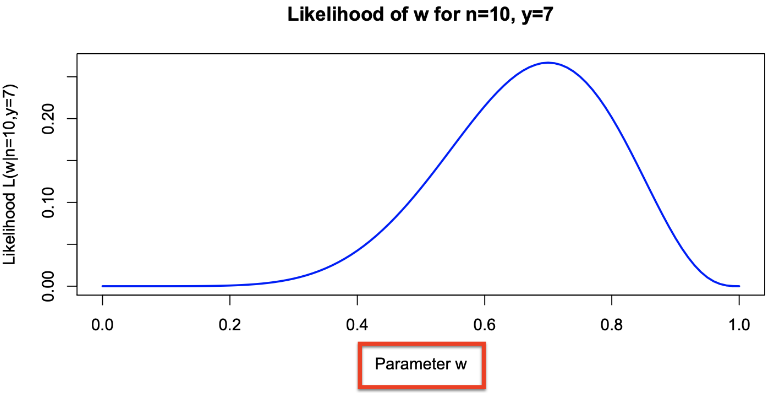
y=7



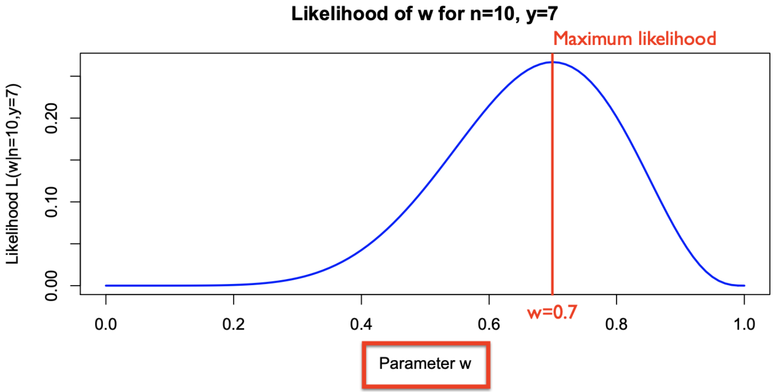
… and so on

Notice $\mathcal{L}(w|n = 10, y = 7)$ is highest when $w = 0.7$

Graphing the Likelihood Function



Graphing the Likelihood Function



$w = 0.7$ is the Maximum Likelihood Estimate!!!

Maximum Likelihood Estimate (MLE)

- find the probability distribution (the model) that makes the observed data most likely
- seek the value of the parameter vector w that maximizes the likelihood function

$\mathcal{L}(w|y)$ - the resulting parameter vector w is known as the MLE estimate

Maximum Likelihood Estimate (MLE)

Three ways of finding the MLE

1. Analytical: use calculus to solve for the parameter value(s) w that result in a peak
2. Brute Force: exhaustive search through parameter space in a grid
3. Optimization: use non-linear optimization (e.g. gradient descent) to iteratively find the peak

Numerical Considerations

- we saw before that the PDF for observed data, $y = (y_1, y_2, ..., y_n)$ given a parameter vector w, can be expressed as the **product (multiply) of PDFs for individual observations**

$$\mathcal{L}(w|y_1, y_2, ..., y_n) = \mathcal{L}_1(w|y_1)\mathcal{L}_2(w|y_2)...\mathcal{L}_n(w|y_n)$$

- multiplying together a lot of values that lie between 0 and 1, (as many as there are data points) will result in a very small number
- in fact the more data, the smaller the resulting product will be
- computers are not good at representing very small numbers

Numerical Considerations

- solution: take the logarithm
- this reformulates the series of products, as a series of sums
- the more data, the higher the resulting sum

$$ln[\mathcal{L}_1(w|y_1)\mathcal{L}_2(w|y_2)...\mathcal{L}_n(w|y_n)] = ln[\mathcal{L}_1(w|y_1) + \mathcal{L}_2(w|y_2)+, ..., \mathcal{L}_n(w|y_n)]$$

Numerical Considerations

- another problem: most optimization algorithms are formulated in terms of minimizing an objective function, not maximizing
- solution: rather than maximizing the log-likelihood, we will minimize the negative log-likelihood
- find w that minimizes:

$$argmin_w \left[-1.0 \left(ln \left[\mathcal{L}_1(w|y_1) + \mathcal{L}_2(w|y_2) +, ..., \mathcal{L}_n(w|y_n) \right] \right) \right]$$

An Example

An Example

Likelihood Function

Likelihood Function:

$$\mathcal{L}(w|n,y) = \frac{n!}{y!(n-y)!} w^y (1-w)^{n-y}$$

Log Likelihood Function:

$$ln[\mathcal{L}(w|n,y)] = ln\left(\frac{n!}{y!(n-y)!}\right) + y \cdot ln(w) + (n-y) \cdot ln(1-w)$$

MLE - ANALYTICAL

MLE - ANALYTICAL

We want:

$$\frac{d}{dw}\left(ln[\mathcal{L}(w|n,y)]\right) = 0$$

Log Likelihood Function:

$$ln[\mathcal{L}(w|n,y)] = ln\left(\frac{n!}{y!(n-y)!}\right) + y \cdot ln(w) + (n-y) \cdot ln(1-w)$$

Taking the partial derivative of the log likelihood function:

$$\frac{d}{dw}\left(ln[\mathcal{L}(w|n,y)]\right) = \frac{d}{dw}\left(ln\left(\frac{n!}{y!(n-y)!}\right) + y \cdot ln(w) + (n-y) \cdot ln(1-w)\right) = 0$$

$$\frac{d}{dw}\left(ln[\mathcal{L}(w|n,y)]\right) = 0 + \frac{n}{w} - \frac{n-y}{1-w} = 0$$

MLE - ANALYTICAL

$$\frac{n}{w} - \frac{n-y}{1-w} = 0$$

Finding the common denominator:

$$\frac{y(1-w)}{w(1-w)} - \frac{w(n-y)}{w(1-w)} = 0$$

$$\frac{y(1-w) - w(n-y)}{w(1-w)} = 0$$

$$\frac{y - y \cdot w - w \cdot n + y \cdot w}{w(1-w)} = 0$$

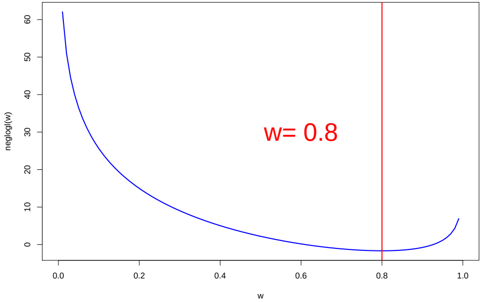
$$w = \frac{y}{n}$$

MLE $= 0.8 = \frac{16}{20}$

MLE - BRUTE FORCE

MLE - BRUTE FORCE

```
neglogl <- function(w) {
  loglik <- log(116280) + 16 * log(w) + 4 * log(1-w)
  return(-1 * loglik)
}
w <- seq(0,1,.01) # iterate through a range of w's
plot(w, neglogl(w), type="l", col="blue", lwd=2)
imin <- which(neglogl(w)==min(neglogl(w)))
abline(v=w[imin], col="red", lwd=2)
text(.6, 30, paste("w=",w[imin]),col="red", cex = 3)
```



MLE - OPTIMIZER

MLE - OPTIMIZER

```
neglogl <- function(w) {
  loglik <- log(116280) + 16 * log(w) + 4 * log(1-w)
  return(-1 * loglik)
}
opt <- nlm(f=neglogl, p=0.5)
```

```
## Warning in log(1 - w): NaNs produced

## Warning in nlm(f = neglogl, p = 0.5): NA/Inf replaced by maximum positive value

## Warning in log(1 - w): NaNs produced

## Warning in nlm(f = neglogl, p = 0.5): NA/Inf replaced by maximum positive value
```

```
opt$estimate
```

```
## [1] 0.7999995
```

Finds the Maximum Likelihood Estimate: 0.8

Beyond the Binomial

likelihood function for a normal distribution. likelihood estimators for linear regression. any model you want

Next Week

- Bayesian Statistics

Next Week

Homework = likelihood function for a normal distribution. Find the maximum likelihood estimators using (calculus, gradient descent, grid) Final = find the likelihood estimators for linear regression.