Generalized Linear Models (GLMs/GLIMs)

MATH 420/620

Monday, 25 Sept, 2017

Principlan of Maximum Likelihood

Consider a probability model of independent sample data

$$x = \{x_1, x_2, \ldots, x_n\}$$

can be formulated as a joint probability density (or mass) function

$$f_\theta(x).$$

Then the likelihood function is defined as

$$\mathcal{L}(\theta|x) = f_\theta(x)$$

where θ is a parameter vector (the argument to the Likelihood function) and the x values (the data) are fixed. That is, the likelihood function is just the joint density or mass function, but where we reverse the roles of parameter and data.

Q: What is the domain of the joing PDF/PMF? What is the domain of the Likelihood function?

Likelihood Examples

Q: What is a plausible likelihood function for the following data set where an unfair coin was tossed 14 times and the number of outcomes that were heads was counted:
$y = (4, 6, 6, 7, 7, 9, 10, 8, 2, 6)$?

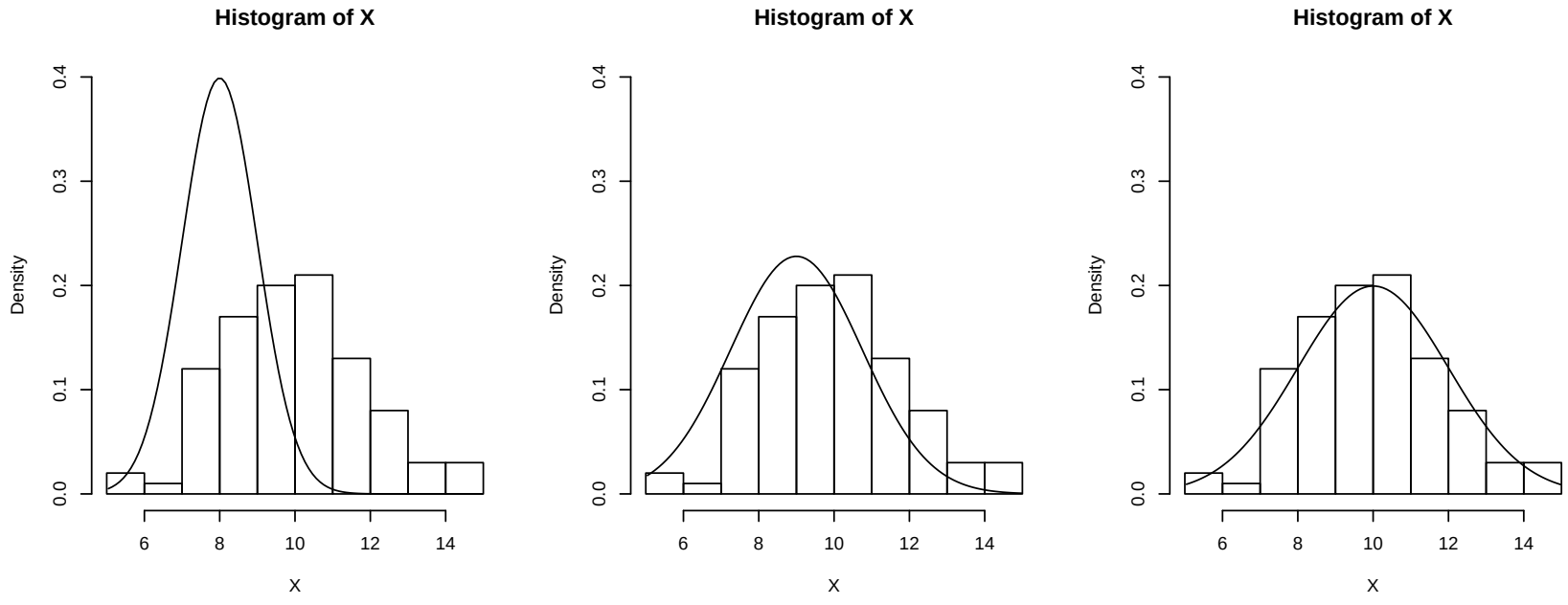
Q: What is the likelihood function for a data set of tree heights?

Q: How would you determine a reasonable likelihood function for the distances from UNR to all of the Starbucks stores in Nevada?

Maximum Likelihood Estimators

If $\hat{\theta}$ maximizes the likelihood, we call it a Maximum Likelihood Estimator (MLE). The Cramer-Rao Inequality tells us MLEs are *minimum variance estimators* but MLEs are often *biased*.

Intuition: Suppose you estimate the mean μ and variance σ^2 of a Normal distribution by plotting a histogram, choose an initial μ and σ and overlay the corresponding density curve, then iteratively adjust μ and σ until it looks like a good match.

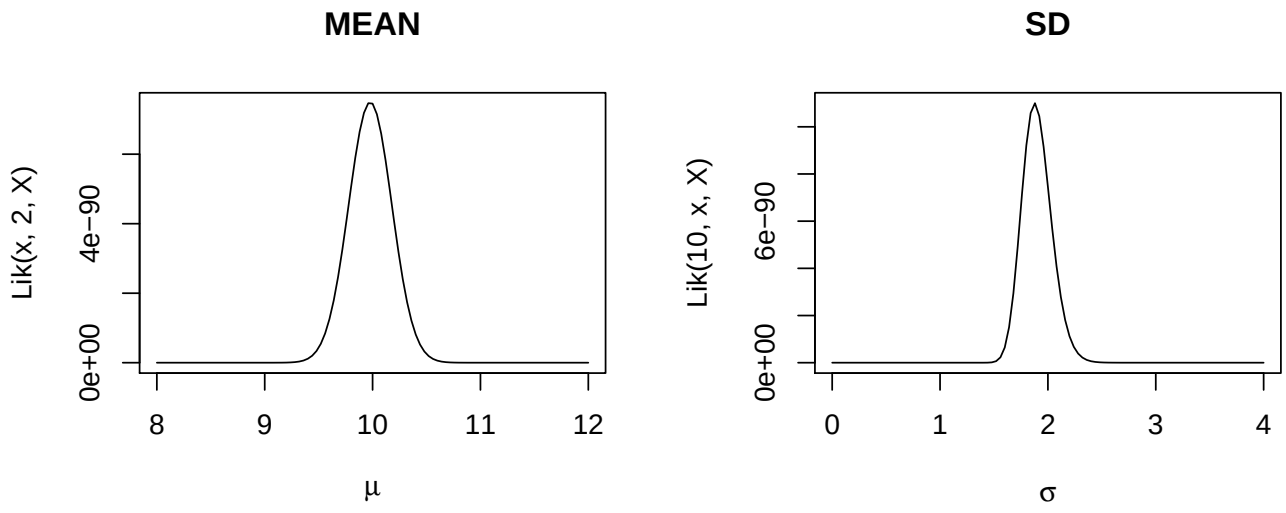


MLE Example

For iid Normal data ($f(x_i; \mu, \sigma)$) the joint density $f_X(\mathbf{x}; \mu, \sigma) = \prod_{i=1}^{n} f(x_i, \mu, \sigma)$ defines the likelihood function for parmeters μ and σ,

$$\mathcal{L}(\mu, \sigma; \mathbf{x}) = \prod_{i=1}^{n} f(x_i, \mu, \sigma).$$

Plotting likelihood values over a range of possible parameter values (here holding one parameter constant while varying the other) in **R** yields. . .



The MLEs for μ and σ are the pair of values that yield the maximum likelihood value. In this case, using the `optim()` function yields $\mu = 9.98$ and $\sigma = 1.88$.

GLM Model Framework

The GLM is described by three components:

1. The *random component* specifies the conditional distribution of $y_i|\vec{x}_i$ and is typically a member of the exponential family of distributions (Normal, binomial, Poisson, Negative-binomial, etc.) but other distributions are possible.
2. We call our linear sum of predictors the *linear predictor*, and denote it as $\eta_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}$
3. We call the transformation that links the expected response values $\mu_i = E(y_i|\vec{x})$ and the linear predictor η_i the *link function*: $g(\mu_i) = \eta_i$. This link function is assumed to be smooth (differentiable) and invertible. It's inverse g^{-1} is often called the *mean function* since $\mu_i = g^{-1}(\eta_i)$.

GLM vs MLR

Recall the MLR model with untransformed response values can be written as

$$y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \epsilon_i.$$

In that case, we model $E(g(y_i))$ as a linear sum of x_i values, and further assume Normal errors with constant variance.

Consider, for now, the simple untransformed case (i.e., g is the identity function).

GLM vs MLR

One could pose the MLR model as a GLM (not to be confused with a General Linear Model) as follows:

1. The *random component* is Normally distributed.
2. The *linear predictor* is $\eta_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}$ (nothing new here!)
3. The *link function* is the identity function: $g(E(y_i)) = E(y_i) = \eta_i$

GLM vs MLR

Note that transforming Y values under MLR is different than specifying a non-identity link function!

GLMs model $g(E(y_i))$, the transformed expecation of the response, using the linear predictor. This gives more flexibility to apply linearizing transformations without affecting the distribution about that trend. For example, compare the two models by comparing y_i values and inverse-transforms:

$$\text{MLR: } y_i = g^{-1}(\eta_i + \epsilon_i)$$

$$\text{GLM: } y_i = g^{-1}(\eta_i) + \epsilon_i$$

This distinction often makes GLMs preferrable over MLR.

Parameter Estimation, etc.

Parameter estimation is done via Maximum Liklihood, and most of the diagnostics for multiple linear regresion carry over to GLMs.

For more information, please see Ch. 15 of *Applied Regression Analysis & Generalized Linear Models* by John Fox.
`http://www.sagepub.com/sites/default/files/upm-binaries/21121_Chapter_15.pdf`

Example: Logistic Regression (Sheather, Ch. 8)

A common form of response data are counts of a particular type of outcome among m trials. For example, the number of individuals in a sample with a specific genotype. In such cases, the data are best modeled using a binomial distribution, not a Normal distribution, using *logistic regression*.

$$E(Y|x) \sim binom(m, p)$$

Example: Logistic Regression

Here the parameter of interest is p – the probability of a *success* on each of our m trials. Since m is known and not a parameter that needs to be estimated, the goal is to estimate p as a function of our *linear predictor*. In logistic regression, this is done by assuming a logit link function,

$$E(Y|X) = m\,p = m\frac{\exp(\eta_i)}{1+\exp(\eta_i)} = \frac{m}{1+\exp(-\eta_i)}$$

Thus, a little algebra gives that

$$\eta_i = \log\left(\frac{p(x)}{1-p(x)}\right)$$

We call the right side of that equation the *logit* function, and $p(x)/(1-p(x))$ the *odds*.

Example: Logistic Regression

To see how this can be cast as a GLM, note that:

1. The distribution is binomial.
2. The relationship between the mean (let's use $\mathrm{E}(y_i/m = \theta_i)$) and linear predictor η_i is given by the logit function

$$\eta_i = g(\theta) = \log\left(\frac{\theta(x)}{1-\theta(x)}\right)$$

Exercise: Form a group of 2-4 people to find some online examples of logistic regression analyses in R using `glm()` and work through them, or simulate binomial count data with known p using either a logic or probit (or other) link function, then use glm() to conduct a logistic regression on your synthetic data.