6 Likelihood and All That

This chapter presents the basic concepts and methods you need in order to estimate parameters, establish confidence limits, and choose among competing hypotheses and models. It defines likelihood and discusses frequentist, Bayesian, and information-theoretic inference based on likelihood.

6.1 Introduction

Previous chapters introduced all the ingredients you need to define a model—mathematical functions to describe the deterministic patterns and probability distributions to describe the stochastic patterns—and showed how to use these ingredients to simulate simple ecological systems. The final steps of the modeling process are estimating parameters from data and testing models against each other. You may be wondering by now how you would actually do this.

Estimating the parameters of a model means finding the parameters that make that model fit the data best. To compare among models we have to figure out which one fits the data best, and decide if one or more models fit sufficiently better than the rest that we can declare them the winners. Our goodness-of-fit metrics will be based on the *likelihood*, the probability of seeing the data we actually collected given a particular model. Depending on the context, "model" could mean either the general form of the model or a specific set of parameter values.

6.2 Parameter Estimation: Single Distributions

Parameter estimation is simplest when we have a a collection of independent data that are drawn from a distribution (e.g., Poisson, binomial, normal), with the same parameters for all observations.* As an example with discrete data, we will select one particular case out of Vonesh's tadpole predation data (p. 47)—small tadpoles at a density of 10—and estimate the per-trial probability parameter of a binomial distribution (i.e., each individual's probability of being eaten by a predator). As an

* In statistical jargon, such data are called *independent and identically distributed* (iid).

example with continuous data, we will introduce a new data set on myxomatosis virus concentration (titer) in experimentally infected rabbits (`Myxo` in the `emdbook` package; Fenner et al., 1956; Dwyer et al., 1990). Although the titer actually changes systematically over time, we will gloss over that problem for now and pretend that all the measurements are drawn from the same distribution so that we can estimate the parameters of a Gamma distribution that describes the variation in titer among different rabbits.

6.2.1 Maximum Likelihood

We want the *maximum likelihood estimates* of the parameters—those parameter values that make the observed data most likely to have happened. Since the observations are independent, the joint likelihood of the whole data set is the product of the likelihoods of each individual observation. Since the observations are identically distributed, we can write the likelihood as a product of similar terms. For mathematical convenience, we almost always maximize the logarithm of the likelihood (log-likelihood) instead of the likelihood itself. Since the logarithm is a monotonically increasing function, the maximum log-likelihood estimate is the same as the maximum likelihood estimate. Actually, it is conventional to *minimize* the negative log-likelihood rather than maximizing the log-likelihood. For continuous probability distributions, we compute the probability *density* of observing the data rather than the probability itself. Since we are interested in relative (log-)likelihoods, not the absolute probability of observing the data, we can ignore the distinction between the density ($P(x)$) and the probability (which includes a term for the measurement precision: $P(x)\,dx$).

6.2.1.1 TADPOLE PREDATION DATA: BINOMIAL LIKELIHOOD

For a single observation from the binomial distribution (e.g., the number of small tadpoles killed by predators in a single tank at a density of 10), the likelihood that k out of N individuals are eaten as a function of the per capita predation probability p is $\text{Prob}(k|p,N) = \binom{N}{k}p^k(1-p)^{N-k}$. If we have n observations, each with the same total number of tadpoles N, and the number of tadpoles killed in the ith observation is k_i, then the likelihood is

$$\mathcal{L} = \prod_{i=1}^{n} \binom{N}{k_i} p^{k_i}(1-p)^{N-k_i}.^{*} \qquad (6.2.1)$$

The log-likelihood is

$$L = \sum_{i=1}^{n} \left(\log \binom{N}{k_i} + k_i \log p + (N-k_i)\log(1-p) \right). \qquad (6.2.2)$$

In R, this would be `sum(dbinom(k,size=N,prob=p,log=TRUE))`.

* The symbol $\prod$ denotes a product, like $\sum$ but for multiplication.

Analytical Approach

In this simple case, we can actually solve the problem analytically, by differentiating with respect to p and setting the derivative to zero. Let $\hat{p}$ be the maximum likelihood estimate, the value of p that satisfies

$$\frac{dL}{dp} = \frac{d\sum_{i=1}^{n}\left(\log\binom{N}{k_i} + k_i \log p + (N-k_i)\log(1-p)\right)}{dp} = 0. \tag{6.2.3}$$

Since the derivative of a sum equals the sum of the derivatives,

$$\sum_{i=1}^{n}\frac{d\log\binom{N}{k_i}}{dp} + \sum_{i=1}^{n}\frac{d\,k_i\log p}{dp} + \sum_{i=1}^{n}\frac{d\,(N-k_i)\log(1-p)}{dp} = 0. \tag{6.2.4}$$

The term $\log\binom{N}{k_i}$ is a constant with respect to p, so its derivative is zero and the first term disappears. Since k_i and $(N-k_i)$ are constant factors, they come out of the derivatives and the equation becomes

$$\sum_{i=1}^{n} k_i \frac{d\log p}{dp} + \sum_{i=1}^{n}(N-k_i)\frac{d\log(1-p)}{dp} = 0. \tag{6.2.5}$$

The derivative of $\log p$ is $1/p$, so the chain rule says the derivative of $\log(1-p)$ is $d(\log(1-p))/d(1-p)\cdot d(1-p)/dp = -1/(1-p)$. Remembering that $\hat{p}$ is the value of p that satisfies this equation:

$$\frac{1}{\hat{p}}\sum_{i=1}^{n} k_i - \frac{1}{1-\hat{p}}\sum_{i=1}^{n}(N-k_i) = 0$$

$$\frac{1}{\hat{p}}\sum_{i=1}^{n} k_i = \frac{1}{1-\hat{p}}\sum_{i=1}^{n}(N-k_i)$$

$$(1-\hat{p})\sum_{i=1}^{n} k_i = \hat{p}\sum_{i=1}^{n}(N-k_i)$$

$$\sum_{i=1}^{n} k_i = \hat{p}\left(\sum_{i=1}^{n} k_i + \sum_{i=1}^{n}(N-k_i)\right) = \hat{p}\sum_{i=1}^{n} N$$

$$\sum_{i=1}^{n} k_i = \hat{p}nN$$

$$\hat{p} = \frac{\sum_{i=1}^{n} k_i}{nN}. \tag{6.2.6}$$

So the maximum likelihood estimate, $\hat{p}$, is just the overall fraction of tadpoles eaten, lumping all the observations together: a total of $\sum k_i$ tadpoles were eaten out of a total of nN tadpoles exposed in all of the observations.

We seem to have gone to a lot of effort to prove the obvious, that the best estimate of the per capita predation probability is the observed frequency of predation.

Other simple distributions like the Poisson behave similarly. If we differentiate the likelihood, or the log-likelihood, and solve for the maximum likelihood estimate, we get a sensible answer. For the Poisson, the estimate of the rate parameter $\hat{\lambda}$ is equal to the mean number of counts observed per sample. For the normal distribution, with two parameters μ and σ^2, we have to compute the *partial derivatives* (see the appendix) of the likelihood with respect to both parameters and solve the two equations simultaneously ($\partial L/\partial\mu = \partial L/\partial\sigma^2 = 0$). The answer is again obvious in hindsight: $\hat{\mu} = \bar{x}$ (the estimate of the mean is the observed mean) and $\hat{\sigma}^2 = \sum(x_i - \bar{x})^2/n$ (the estimate of the variance is the variance of the sample).*

Some simple distributions like the negative binomial, and all the complex problems we will be dealing with hereafter, have no easy analytical solution, so we will have to find the maximum likelihood estimates of the parameters numerically. The point of the algebra here is just to convince you that maximum likelihood estimation makes sense in simple cases.

Numerics

This chapter presents the basic process of computing and maximizing likelihoods (or minimizing negative log-likelihoods) in R; Chapter 7 will go into much more technical detail. First, you need to define a function that calculates the negative log-likelihood for a particular set of parameters. Here's the R code for a binomial negative log-likelihood function:

```
> binomNLL1 = function(p, k, N) {
+     -sum(dbinom(k, prob = p, size = N, log = TRUE))
+ }
```

The `dbinom` function calculates the binomial likelihood for a specified data set (vector of number of successes) `k`, probability `p`, and number of trials `N`; the `log=TRUE` option gives the log-probability instead of the probability (more accurately than taking the log of the product of the probabilities); `-sum` adds the log-likelihoods and changes the sign to compute an overall negative log-likelihood for the data set.

Load the data and extract the subset we plan to work with:

```
> data(ReedfrogPred)
> x = subset(ReedfrogPred, pred == "pred" & density == 10
+   & size == "small")
> k = x$surv
```

The total number of tadpoles exposed in this subset of the data is 40 (10 in each of 4 trials), 30 of which were eaten by predators, so the maximum likelihood estimate will be $\hat{p} = 0.75$.

We can use the `optim` function to numerically **optim**ize (by default, minimizing rather than maximizing) this function. You need to give `optim` the *objective function*—the function you want to minimize (`binomNLL1` in this case)—and a vector of starting parameters. You can also give it other information, such as a data set, to

* Maximum likelihood estimation gives a biased estimate of the variance, dividing the sum of squares $\sum(x_i - \bar{x})^2$ by n instead of $n-1$.

be passed on to the objective function. The starting parameters don't have to be very accurate (if we had accurate estimates already we wouldn't need optim), but they do have to be reasonable. That's why we spent so much time in Chapters 3 and 4 on eyeballing curves and the method of moments.

```
> opt1 = optim(fn = binomNLL1, par = c(p = 0.5), N = 10,
+      k = k, method = "BFGS")
```

fn is the argument that specifies the objective function and par specifies the vector of starting parameters. Using c(p=0.5) names the parameter p—probably not necessary here but very useful for keeping track when you start fitting models with more parameters. The rest of the command specifies other parameters and data and optimization details; Chapter 7 explains why you should use method="BFGS" for a single-parameter fit.

Check the estimated parameter value and the maximum likelihood—we need to change sign and exponentiate the minimum negative log-likelihood that optim returns to get the maximum log-likelihood:

```
> opt1$par

p
0.7499998
```

Because it was computed numerically the answer is almost, but not exactly, equal to the theoretical answer of 0.75.

```
> exp(-opt1$value)

[1] 0.0005150149
```

The mle2 function in the bbmle package provides a "wrapper" for optim that gives prettier output and makes standard tasks easier.* Unlike optim, which is designed for general-purpose optimization, mle2 assumes that the objective function is a negative log-likelihood function. The names of the arguments are easier to understand: minuslogl instead of fn for the negative log-likelihood function, start instead of par for the starting parameters, and data for additional parameters and data.

```
> library(bbmle)
> m1 = mle2(minuslogl = binomNLL1, start = list(p = 0.5),
+      data = list(N = 10, k = k))
> m1

Call:
mle2(minuslogl = binomNLL1, start = list(p = 0.5), data =
    list(N = 10, k = k))
```

* Why mle2? There is an mle function in the stats4 package that comes with R, but I added some features—and then renamed it to avoid confusion with the original R function.

```
Coefficients:
        p
0.7499998

Log-likelihood: -7.57
```

The mle2 package has a shortcut for simple likelihood functions. Instead of writing an R function to compute the negative log-likehood, you can specify a formula:

```
> mle2(k ~ dbinom(prob = p, size = 10),
+      start = list(p = 0.5))
```

gives exactly the same answer as the previous commands. R assumes that the variable on the left-hand side of the formula is the response variable (k in this case) and that you want to sum the negative log-likelihood of the expression on the right-hand side for all values of the response variable.

Another way to find maximum likelihood estimates for data drawn from most simple distributions—although not for the binomial distribution—is the fitdistr command in the MASS package, which will even guess reasonable starting values for you. However, it works only in the very simple case where none of the parameters of the distribution depend on other covariates.

The estimated value of the per capita predation probability, 0.7499..., is very close to the analytic solution of 0.75. The estimated value of the maximum likelihood (Figure 6.1) is quite small ($\mathcal{L} = 5.15 \times 10^{-4}$). That is, the probability of *this particular outcome*—5, 7, 9 and 9 out of 10 tadpoles eaten in four replicates—is low.* In general, however, we will be interested only in the relative likelihoods (or log-likelihoods) of different parameters and models rather than their absolute likelihoods.

Having fitted a model to the data (even a very simple one), it's worth plotting the predictions of the model. In this case the data set is so small (four points) that sampling variability dominates the plot (Figure 6.1b).

6.2.1.2 MYXOMATOSIS DATA: GAMMA LIKELIHOOD

As part of the effort to use myxomatosis as a biocontrol agent against introduced European rabbits in Australia, Fenner and co-workers (1956) studied the virus concentrations (*titer*) in the skin of rabbits that had been infected with different virus strains. We'll choose a Gamma distribution to model these continuously distributed, positive data.† For the sake of illustration, we'll use just the data for one viral strain (grade 1).

```
> data(MyxoTiter_sum)
> myxdat = subset(MyxoTiter_sum, grade == 1)
```

*I randomly simulated 1000 samples of four values drawn from the binomial distribution with $p = 0.75$, $N = 10$. The maximum likelihood was smaller than the observed value given in the text 22% of the time. Thus, although small, this likelihood is not significantly lower than would be expected by chance.

† We could also use a log-normal distribution or (since the minimum values are far from zero and the distributions are reasonably symmetric) a normal distribution.

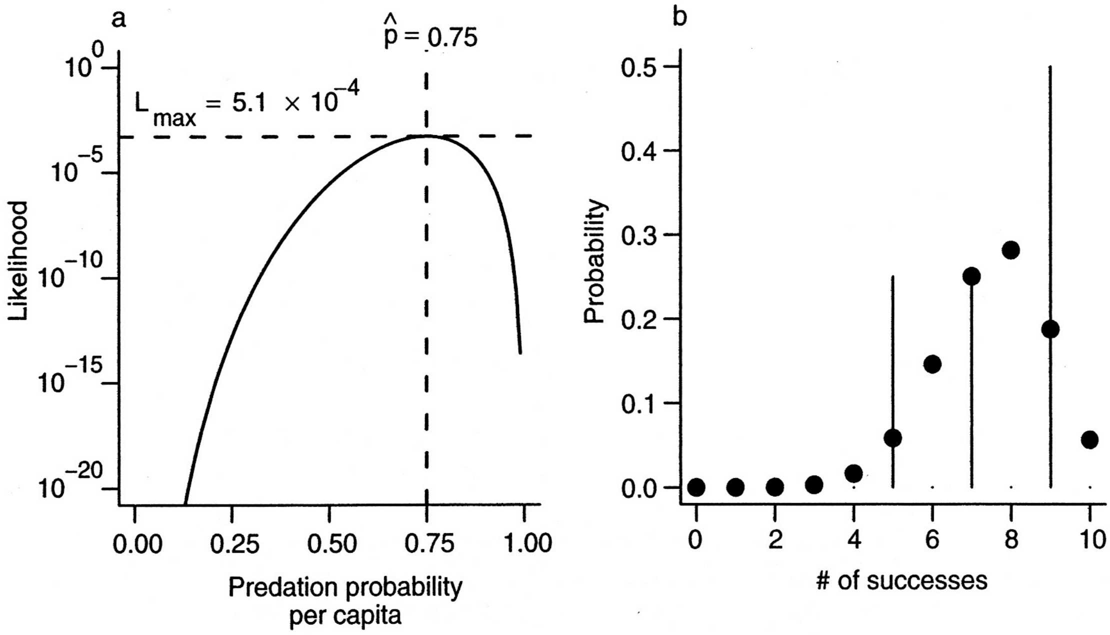

Figure 6.1 Binomial-distributed predation. (a) Likelihood curve, on a logarithmic *y* scale. (b) Best-fit model prediction compared with the data.

The likelihood equation for Gamma-distributed data is hard to maximize analytically, so we'll go straight to a numerical solution. The negative log-likelihood function looks just very much like the one for binomial data.*

```
> gammaNLL1 = function(shape, scale) {
+     -sum(dgamma(myxdat$titer, shape = shape, scale = scale,
+          log = TRUE))
+ }
```

It's harder to find starting parameters for the Gamma distribution. We can use the method of moments (Chapter 4) to determine reasonable starting values for the scale (= variance/mean) and shape (= mean^2/variance = 1/(coefficient of variation)^2) parameters.†

```
> gm = mean(myxdat$titer)
> gvm = var(myxdat$titer)/mean(myxdat$titer)
```

Now fit the data:

```
> m3 = mle2(gammaNLL1, start = list(shape = gm/gvm,
+       scale = gvm))

> m3
```

* optim insists that you specify all of the parameters packed into a single numeric vector in your negative log-likelihood function. mle prefers the parameters as a list. mle2 will accept either a list, or, if you use parnames to specify the parameter names, a numeric vector (p. 183).

† Because the estimates of the shape and scale are very strongly correlated in this case, I ended up having to tweak the starting conditions slightly away from the method of moments estimates, to {45.8,0.151}.

```
Call:
mle2(minuslogl = gammaNLL1, start = list(shape = 45.8,
     scale = 0.151))

Coefficients:
  shape         scale
49.3421124    0.1403326

Log-likelihood: -37.67
```

I could also use the formula interface,

```
> m3 = mle2(myxdat$titer ~ dgamma(shape, scale = scale),
+       start = list(shape = gm/gvm, scale = gvm))
```

Since the default parameterization of the Gamma distribution in R uses a rate parameter instead of a scale parameter, I have to make sure to specify the scale parameter explicitly. Or I could use `fitdistr` from the `MASS` package:

```
> f1 = fitdistr(myxdat$titer, "gamma")
```

`fitdistr` gives slightly different values for the parameters and the likelihood, but not different enough to worry about. A greater possibility for confusion is that `fitdistr` reports the rate (= 1/scale) instead of the scale parameter.

Figure 6.2 shows the negative log-likelihood (now a negative log-likelihood *surface* as a function of two parameters, the shape and scale) and the fit of the model to the data (virus titer for grade 1). Since the "true" distribution of the data is hard to visualize (all of the distinct values of virus titer are displayed as jittered values along the bottom axis), I've plotted the nonparametric (kernel) estimate of the probability density in gray for comparison. The Gamma fit is very similar, although it takes account of the lowest point (a virus titer of 4.2) by spreading out slightly rather than allowing the bump in the left-hand tail that the nonparametric density estimate shows. The large shape parameter of the best-fit Gamma distribution (shape = 49.34) indicates that the distribution is nearly symmetrical and approaching normality (Chapter 4). Ironically, in this case the plain old normal distribution actually fits slightly better than the Gamma distribution, despite the fact that we would have said the Gamma was a better model on biological grounds (it doesn't allow virus titer to be negative). However, according to criteria we will discuss later in the chapter, the models are not significantly different and you could choose either on the basis of convenience and appropriateness for the rest of the story you were telling. If we fitted a more skewed distribution, like the damselfish settlement distribution, the Gamma would certainly win over the normal.

6.2.2 Bayesian Analysis

Bayesian estimation also uses the likelihood, but it differs in two ways from maximum likelihood analysis. First, we combine the likelihood with a prior probability distribution in order to determine a posterior probability distribution. Second, we often report the mean of the posterior distribution rather than its mode (which would

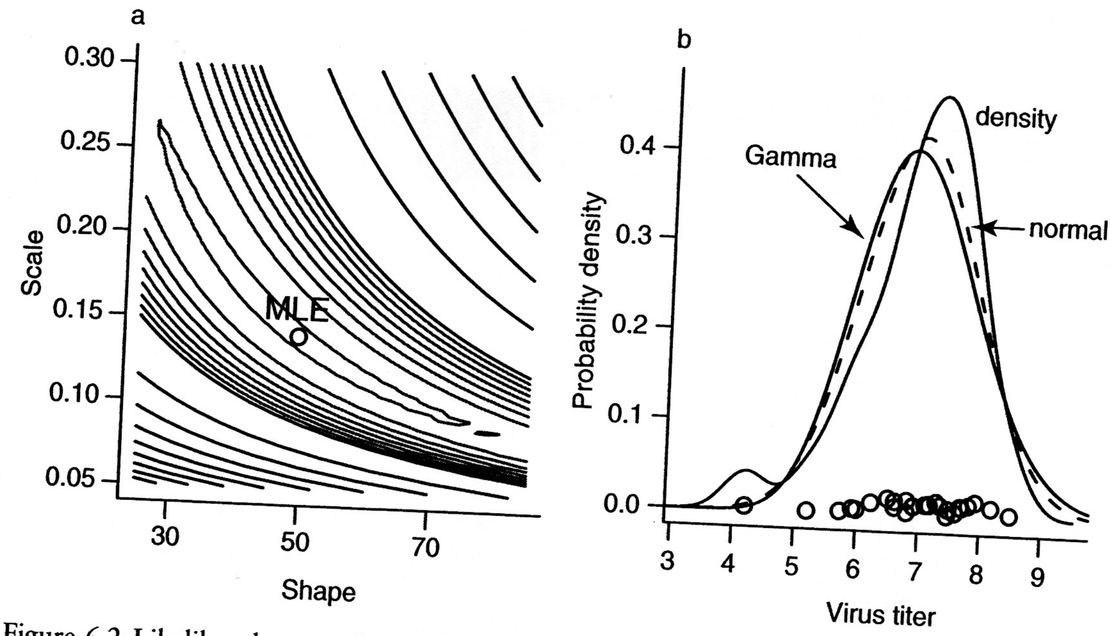

Figure 6.2 Likelihood curves for a simple distribution: Gamma-distributed virus titer. Black contours are spaced 200 log-likelihood units apart; gray contours are spaced 20 log-likelihood units apart. In the right-hand plot, the gray line is a kernel density estimate; solid line is the Gamma fit; and dashed line is the normal fit.

equal the MLE if we were using a completely uninformative, or "flat," prior). Unlike the mode, which reflects only local information about the peak of the distribution, the mean incorporates the entire pattern of the distribution, so it can be harder to compute.

6.2.2.1 BINOMIAL DISTRIBUTION: CONJUGATE PRIORS

In the particular case when we have so-called *conjugate priors* for the distribution of interest, Bayesian estimation is easy. As introduced in Chapter 4, a conjugate prior is a choice of the prior distribution that matches the likelihood model so that the posterior distribution has the same form as the prior distribution. Conjugate priors also allow us to interpret the strength of the prior in simple ways.

For example, the conjugate prior of the binomial likelihood that we used for the tadpole predation data is the Beta distribution. If we pick a Beta prior with shape parameters a and b, and if our data include a total of $\sum k$ "successes" (predation events) and $nN - \sum k$ "failures" (surviving tadpoles) out of a total of nN "trials" (exposed tadpoles), the posterior distribution is a Beta distribution with shape parameters $a + \sum k$ and $b + (nN - \sum k)$. If we interpret $a - 1$ as the total number of previously observed successes and $b - 1$ as the number of previously observed failures, then the new distribution just combines the total number of successes and failures in the complete (prior plus current) data set. When $a = b = 1$, the Beta distribution is flat, corresponding to no prior information ($a - 1 = b - 1 = 0$). As a and b increase, the prior distribution gains more information and becomes peaked. We can also see that, as far as a Bayesian is concerned, how we divide our experiments up doesn't matter. Many small experiments, aggregated with successive uses of Bayes'