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: $P(x)\,dx$).

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 i^{th} observation is k_i, then the likelihood is

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

The log-likelihood is

$$L = \sum_{i=1}^{n} \left(\log \binom{N}{k_i} + k_i \log p + (N-k_i) \log(1-p) \right). \tag{2}$$

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

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

$$\frac{dL}{dp} = \frac{d\sum_{i=1}^{n}\left(\log\binom{N}{k_i} + k_i \log p + (N-k_i)\log(1-p)\right)}{dp} = 0. \tag{3}$$

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

$$\sum_{i=1}^{n}\frac{d\log\binom{N}{k_i}}{dp} + \sum_{i=1}^{n}\frac{dk_i \log p}{dp} + \sum_{i=1}^{n}\frac{d(N-k_i)\log(1-p)}{dp} = 0 \tag{4}$$

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

$$\sum_{i=1}^{n} k_i \frac{d\log p}{dp} + \sum_{i=1}^{n}(N-k_i)\frac{d\log(1-p)}{dp} = 0. \tag{5}$$

The derivative of $\log p$ is $1/p$, so the chain rule says the derivative of $\log(1-p)$ is $d(\log(1-p))/d(1-p)\cdot d(1-p)/dp = -1/(1-p)$. We will denote the particular value of p we're looking for as $\hat{p}$. So

$$\begin{aligned}
\frac{1}{\hat{p}}\sum_{i=1}^{n} k_i - \frac{1}{1-\hat{p}}\sum_{i=1}^{n}(N-k_i) &= 0 \\
\frac{1}{\hat{p}}\sum_{i=1}^{n} k_i &= \frac{1}{1-\hat{p}}\sum_{i=1}^{n}(N-k_i) \\
(1-\hat{p})\sum_{i=1}^{n} k_i &= \hat{p}\sum_{i=1}^{n}(N-k_i) \\
\sum_{i=1}^{n} k_i &= \hat{p}\left(\sum_{i=1}^{n} k_i + \sum_{i=1}^{n}(N-k_i)\right) = \hat{p}\sum_{i=1}^{n} N \\
\sum_{i=1}^{n} k_i &= \hat{p}nN \\
\hat{p} &= \frac{\sum_{i=1}^{n} k_i}{nN}
\end{aligned} \tag{6}$$

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 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*.).

For some simple distributions like the negative binomial, and for all the complex problems we will be dealing with hereafter, there is no easy analytical solution and we 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 **??** will go into much more detail on the technical details. 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 get 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
```

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 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

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

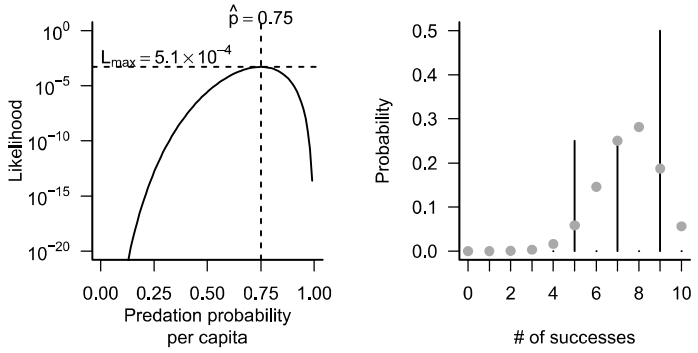

Figure 1: Likelihood curves for a simple distribution: binomial-distributed predation.

why we spent so much time in Chapters ?? and ?? on eyeballing curves and the method of moments.

```
> O1 = 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 ?? 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:

```
> O1$par

        p
0.7499998

> exp(-O1$value)

[1] 0.0005150149
```