An introduction to Maximum Likelihood in R

Stephen P. Ellner (spe2@cornell.edu)
Department of Ecology and Evolutionary Biology, Cornell University
Last compile: June 3, 2010

1 Introduction

Maximum likelihood as a general approach to estimation and inference was created by R. A. Fisher between 1912 and 1922, starting with a paper written as a third-year undergraduate. Then, and for many years, it was more of theoretical than practical interest. Now, the ability to do nonlinear optimization on the computer has made likelihood methods practical and very popular.

Let's start with the *probability density function* for one observation x from normal random variable with mean μ and variance σ^2,

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \tag{1}$$

For a set of n replicate independent observations $x_1, x_2, \cdots, x_n$ the joint density is

$$f(x_1, x_2, \cdots, x_n|\mu,\sigma) = \prod_{i=1}^{n} f(x_i|\mu,\sigma) \tag{2}$$

We interpret this as follows: given the values of μ and σ, equation (2) tells us the relative probability of different possible values of the observations. Maximum likelihood turns this around by defining the *likelihood function*

$$\mathcal{L}(\mu,\sigma|x_1, x_2, \cdots, x_n) = \prod_{i=1}^{n} f(x_i|\mu,\sigma) \tag{3}$$

The right-hand side is the same as (2) but the interpretation is different: given the observations, the function $\mathcal{L}$ tells us the relative likelihood of different possible values of the parameters for the process that generated the data. Note: the likelihood function *is not a probability*, and it does not specifying the relative probability of different parameter values.

It is advantageous to work with the negative log of the likelihood. Log transformation turns the product of f's in (3) into the sum of log f's. For the Normal likelihood (3) this is a one-liner in R :

```
set.seed(1066); x=rnorm(50,mean=1,sd=2); # generate data
NegLogLik1=function(mu,sigma) {
        -sum(dnorm(x,mean=mu,sd=sigma,log=TRUE))
}
```

Here `dnorm` computes the Normal density, `log=TRUE` says to return the log of the density, then we add up the values and change the sign.

The Maximum Likelihood estimates of μ and σ are the values that *minimize* `NegLogLik`. To do that, we turn things over to the function `mle` in the **stats4** package. The `NegLogLik` function above is written the way `mle` needs: the parameters being estimated are named arguments of the function, the data are a "global" variable (NOT a function argument) and the returned value is $-\log \mathcal{L}$. To do the fit:

```
require(stats4)
MLE.fit = mle(minuslogl=NegLogLik1, start=list(mu=0,sigma=1), method="Nelder-Mead")
```

To use `mle`, you need to tell it the name of the negative log likelihood function, give it a list of starting guesses for the parameters, and tell it what method to use for minimization (Nelder-Mead is inefficient but robust, so I tend to try it first). Here's what you get:

```
> summary(MLE.fit)

Maximum likelihood estimation
Call:
mle(minuslogl = NegLogLik1, start = list(mu = 0, sigma = 1), method = "Nelder-Mead")

Coefficients:
      Estimate Std. Error
mu    1.166517  0.2841623
sigma 2.009331  0.2009723

-2 log L: 211.6609
```

The standard errors from `summary`, and the implied confidence limits, are based on a quadratic approximation to the likelihood and a Gaussian approximation to the distribution of parameter estimates. The `vcov` function (e.g., `vcov(MLE.fit)`) returns the full variance-covariance matrix of estimated parameters, under the same approximations.

More accurate confidence limits can be obtained by *profiling*, which is explained below when we get to Likelihood Ratio tests. In R this is done by the `confint` function:

```
> confint(MLE.fit);
Profiling...
          2.5 %   97.5 %
mu    0.5988314 1.734453
sigma 1.6716469 2.477810
```

For μ this is the same as `summary` gave us, but for σ there is a small difference.

Equation (3) is essentially the same for any other situation where observations are mutually

independent[1] and their distribution depends on a parameter vector θ:

$$\mathcal{L}(\theta|x_1, x_2, \cdots, x_n) = \prod_{i=1}^{n} f(x_i|\theta). \tag{4}$$

R has many probability distributions built in, such as `dpois` (Poisson), `dbinom` (Binomial), `dgamma` (gamma), `dlnorm` (lognormal), and many others in the **SuppDists** package.

But the real power of Maximum Likelihood becomes evident when you need to fit a nonstandard model that isn't in your statistics textbook or package. For example: suppose you suspect that your data might come from a mixture of Normal distributions, mostly "good" data with a smaller variance σ_1^2 and a fraction p of "outliers" with a larger variance σ_2^2. But you're willing to assume (for now...) that the means are the same. The probability density function for this model is (in R)

```
(1-p)*dnorm(x,mean=mu,sd=sigma1)+ p*dnorm(x,mean=mu,sd=sigma2)
```

And here are some results (edited) on simulated data:

```
> set.seed(1066);
> x=c(rnorm(75,mean=0,sd=1),rnorm(25,mean=0,sd=2)); # generate data
> MLE.fit2 = mle(minuslogl=NegLogLik2,
     start= list(mu=0, sigma1=0.5*sd(x),sigma2=2*sd(x),p=0.5),
     method="Nelder-Mead")
> summary(MLE.fit2);
Coefficients:
          Estimate Std. Error
mu     -0.04153355  0.1171499
sigma1  0.94477441  0.1164661
sigma2  2.36694799  0.5923218
p       0.21361786  0.1270933
-2 log L: 336.0567

> confint(MLE.fit2);
             2.5 %    97.5 %
mu     -0.27080970 0.1936796
sigma1  0.68922076 1.1953209
sigma2  1.29469619 4.4543330
p       0.05506485 0.6020390
```

Note that profiling gives us a 95% confidence interval on p that is strictly positive, while the results from `summary` do not.

Exercise 1.1 Code the `NegLogLik2` function for the mixture of Normals model, and use it to replicate the results presented above. If you get stuck, *ask for help* so you can go on to the next exercise.

[1] The form of $\mathcal{L}$ is different when observations are not mutually independent, but that's beyond what we can go into here.

Exercise 1.2 . For the same Normal-mixture data, use `mle` to fit an alternative model in which the two components can have different means $\mu_1 < \mu_2$, but have equal variances. Compare the maximized $\log \mathcal{L}$ value for this model (printed out as part of the `summary`), with the maximized $\log \mathcal{L}$ value for correct model, fitted to the same data. Which model fits the data better?

Exercise 1.3 . Now for some real data. Reilly and Hajek (2008) studied how resistance of gypsy moth to its nucleopolyhedrovirus was affected by the crowding that individuals experienced as larvae[2]. The data tabulated below show how rearing density affects survival after exposure to a standardized dose of the virus.

Rearing Density	# Larvae	# Surviving
1	90	60
5	90	60
10	89	56
15	87	41
20	93	31

More compactly, so you can cut-and-paste the data into R :

```
D=c(1,5,10,15,20); N=c(90, 90, 89, 87, 93); S=c(60,60,56,41,31);
```

You will use `mle` to fit a non-standard model for survival as a function of rearing density.

(a) Plot the data. Because a survival probability p must lie in $[0, 1]$, it is often useful to use the logit transformation, so that the response variable is

$$\text{logit}(p) = \log(p/(1-p)).$$

Therefore, make a plot of `logit(S/N)` versus `D`.

(b) The plot suggests that logit(survival) is a nonlinear function of density. Some exploratory data analysis suggests a power function $a + bD^\delta$ with $a > 0, b < 0, \delta \approx 2$, where D is rearing density. The model for survival probability is then

$$p = \frac{e^u}{1+e^u} \quad , \qquad \text{where } u = a + bD^\delta.$$

Write a Negative Log Likelihood function for this model in R , and then use `mle` to estimate the parameters.

To get you started: the simplest probability model for survival is binomial. The probability of having S survivors, out of N total larvae, when the survival probability is p, can be computed in R as `dbinom(x=S,size=N,prob=p)`. Much as with `dnorm`, `dbinom` can take vectors of S, N, p values and return a vector of probabilities, and `log=TRUE` comes in handy. Keep on trying *and keep on asking for help* until you get:

```
> MLE.MothSurv=mle(minuslogl=MothSurvNLL,start=list(a=0.7,b=-0.001,delta=2),
      method="Nelder-Mead")
> summary(MLE.MothSurv);
```

[2] James R. Reilly and Ann E. Hajek, 2008. Density-dependent resistance of the gypsy moth *Lymantria dispar* to its nucleopolyhedrovirus, and the consequences for population dynamics. Oecologia 154: 691-701.

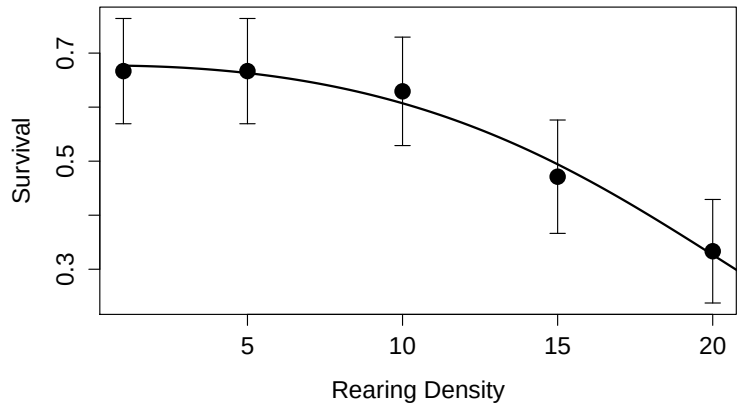

Figure 1: Plot of the Reilly-Hajek data. Points are the experimental survivorships with 95% confidence intervals, and the solid curve is the fitted model with parameters estimated by `mle`.

```
Maximum likelihood estimation
Coefficients:
          Estimate   Std. Error
a      0.740626932 0.1385460586
b     -0.001672166 0.0006476368
delta  2.261457740 0.1232030815
-2 log L: 24.78222
```

(c) (Optional) Check your results: write a script to replicate Figure 1. The `coef` function will extract the parameter estimates for you, e.g. `coef(MLE.MothSurv)`. I used `plotCI` in the **plotrix** package to draw the error bars.

2 Theory and inference

Now we'll go into some of the properties of Maximum Likelihood that justify its use for estimation and inference. All of these require technical assumptions that we will not go into here. The most important are that the likelihood function is a smooth and bounded function of its arguments, and that the set of possible values for the observations is the same for all possible parameter values.

Maximum Likelihood estimates are consistent and asymptotically Normal. "Consistent" means that they converge to the true values as the number of independent observations becomes infinite. The asymptotic Normality is the basis for the approximate standard errors returned by `summary`.

Maximum Likelihood estimates are asymptotically efficient. This is important but a bit delicate. As the number of independent observations n increases, the standard errors on each parameter decrease in proportion to $C/\sqrt{n}$ for some constant C. "Asymptotically efficient" means that there is no unbiased way of estimating parameters for which the standard

errors shrink at a strictly faster rate (e.g., a smaller value of C, or a higher power of n in the denominator). Because maximum likelihood estimates are asymptotically unbiased, this result says that maximum likelihood is a universal "Swiss Army Knife". When it can do the job, it's rarely the best tool for the job but it's rarely much worse than the best (at least for large samples).

Nested models can be compared using a Likelihood Ratio test. "Nested" means that one model is a special case of the other, in which a smaller number of parameters is fitted. These are often called the "reduced" (simpler) and "full" (more complex) models. Let r be the *difference* in the number of fitted parameters between the full and reduced models. Then for large sample sizes, twice the difference in their maximized log $\mathcal{L}$ values follows an approximately χ^2 distribution with r degrees of freedom. As an example, we can take the mixture-of-normals as our "full" model, and the single normal as the "reduced" model. Here $r = 2$ because the full model adds p and σ_2.

```
## likelihood ratio test for mixture vs. single distribution
set.seed(1066); x=c(rnorm(75,mean=0,sd=1),rnorm(25,mean=0,sd=2)); # generate data
MLE.fit1 = mle(minuslogl=NegLogLik1, start=list(mu=0,sigma=1), method="Nelder-Mead")
MLE.fit2 = mle(minuslogl=NegLogLik2, start=list(mu=0,sigma1=0.5,sigma2=2,p=0.5),
    method="Nelder-Mead")
W=2*(logLik(MLE.fit2) - logLik(MLE.fit1));
1-pchisq(W,2); # upper tail probability
[1] 0.003476646
```

Likelihood ratio tests are the basis for profile confidence intervals. The interval for a parameter θ is obtained by fitting the restricted model with an assumed (rather than fitted) value $\theta = \theta_0$, for a grid of θ_0 values. The 95% profile confidence interval is the range of values such that the reduced model is not rejected at level $\alpha = 0.05$. The same approach can be used to construct bivariate confidence regions for a pair of parameters, or multivariate confidence regions for any set of parameters.

Non-nested models can be compared using AIC or BIC. AIC takes the minimized $-\log \mathcal{L}$ values and adds a penalty for the number of fitted parameters. Models with smaller AIC values are preferred over models with larger AIC values.

```
> AIC(MLE.fit1,MLE.fit2)
         df      AIC
MLE.fit1  2 351.3801
MLE.fit2  4 344.0567
```

This agrees with the result of the Likelihood Ratio test. However, the meaning of the two comparisons is very different. The "target" of AIC is out-of-sample prediction accuracy: which model will do best at predicting future observations? A Likelihood Ratio test asks whether the data provide evidence that the full model is a better description of the process generating the data. A big difference in AIC between models leads to a conclusion about how to make the best forecasts. A significant Likelihood Ratio test leads to a conclusion about the true state of nature. Sometimes those coincide, sometimes they don't. In particular, AIC is not *order*

consistent: in many situations, if data come from a model with a finite number of parameters, then as the amount of data goes up, AIC is increasingly likely to select a model that is more complex than the true model. Reasonable people disagree as to whether this is a bug or a feature.

Warning: R has a very relaxed attitude about computing AIC values. In some cases (e.g., mixed models in `lme4`) `AIC`'s default is to compute the basic large-sample formula ($AIC = -2\log\mathcal{L} + 2k$, where k is the number of fitted parameters) without worrying about whether it's valid for the model in question (and for mixed models, it isn't).

Warning: Be careful with the parameter count. If you fit a linear regression model $y = ax + b$ by maximum likelihood, the number of fitted parameters is **3**, because you are also estimating the error variance.

BIC puts a higher penalty on model complexity: the basic formula is $BIC = -2\log\mathcal{L} + \log(n)k$ where n is the sample size. BIC is a large-sample approximation to Bayesian model selection starting from a uniform prior on the set of models under consideration. BIC is philosophically closer to a Likelihood Ratio test, in that the "target" of BIC is the model closest to the truth, not the one with the best prediction accuracy.

Final warning: AIC, BIC and LRT are all *large sample approximations*. Always use something better when it's available (e.g., an F test for Gaussian linear models).

Exercise 2.1 Use AIC to compare the two mixture-of-normals models that you have previously fitted, one with unequal means and the other with unequal variances. Use the data that you already generated, `x=c(rnorm(75,mean=0,sd=1),rnorm(25,mean=0,sd=2))`.

Exercise 2.2 Use Likelihood Ratio tests to compare the sequence of nested models for the simulated mixture-of-normals data:

$$\begin{aligned} M_1 &: \mu_1 = \mu_2, \sigma_1 = \sigma_2 \\ M_2 &: \mu_1 = \mu_2, \sigma_1 \neq \sigma_2 \\ M_3 &: \mu_1 \neq \mu_2, \sigma_1 \neq \sigma_2 \end{aligned} \tag{5}$$

Exercise 2.3 Use a Likelihood Ratio test on the Reilly-Hajek data to compare the full model that you previously fitted, to a reduced model in which $\delta = 2$ is assumed. Repeat the same comparison using AIC and BIC.

Exercise 2.4 Design and carry out a simulation study of the χ^2 approximation for the Likelihood Ratio statistic. That is: generate data from a "reduced" model (e.g., a single normal distribution); fit the "reduced" and a "full" model (e.g., mixture of normals) to the data; compute the Likelihood Ratio statistic. Repeat many times, and compare the test statistic values to the theoretical χ^2 distribution. In particular, how does the 95^{th} percentile of your test statistic values compare to the 95^{th} quantile of the theoretical χ^2 distribution?

3 Maximum Likelihood trajectory matching

The goal is easy to describe: use $-\log \mathcal{L}$ as the objective function for fitting a differential equation. The work is in specifying the likelihood function.

Trajectory matching assumes that the epidemic dynamics are deterministic. Residuals (deviations between model output and the data) are therefore assumed to be the result of *measurement noise*: sampling variability, inaccuracies in antibody assays, etc. So the likelihood we need is the probability distribution of measurement errors, conditional on the model trajectory that results from the fitted parameter values and initial conditions. I will discuss three methods for specifying that likelihood and using it for model fitting.

3.1 Variability of repeated measurements

The ideal is to have data on measurement errors. It's rare, but it happens. Jost and Ellner (2000) fitted predator-prey models to data of Veilleux (1976, 1979) with *Paramecium aurelia* as the prey and *Didinium nasutum* as predator. Each of Veilleux's population estimates were the mean across 8 replicate 1ml samples taken from the culture medium. The original data were no longer available, but Veilleux (1976) included a table of the mean and variance of replicates at a wide range of population densities.

What mean-variance relationship do we expect? A reasonable sampling model is the Binomial: the population count in each 1ml sample is $\sim B(N,p)$ where N is the total population and p the probability that an individual is included in the sample. The resulting population estimate from one count is

$$\hat{N} \sim \frac{1}{p} B(N,p) \tag{6}$$

with variance

$$Var(\hat{N}|N) = \frac{1}{p^2} Np(1-p) = \frac{N(1-p)}{p}. \tag{7}$$

Taking 8 of these and averaging them, reduces the variance by a factor of 8. We therefore expect a relationship $Var(\hat{N}|N) = cN$ for some constant c. Veilleux's tabulated data fitted this prediction very well ($r^2 > 0.95$ for both species), with $c = 0.42$ for prey, $c = 0.17$ for the predator. That's good and bad: good that the predicted form of the relationship held, bad because the coefficients should have been the same. Maybe prey density varies in space due to local predation? So we used the fitted variance models, but we assumed a Normal distribution of sampling errors because the data were averages of 8 samples.

The rest is straightforward. As an example, consider fitting the θ-logistic model

$$\frac{dN}{dt} = rN\left(1 - \left(\frac{N}{K}\right)^{\theta}\right)$$

to data on the prey alone. With the data in a vector `xvals`, and a function `ThetaLogist` to solve the ODE, the negative log likelihood is

```
TLNegLogLik=function(r,K,theta,x0) {
```

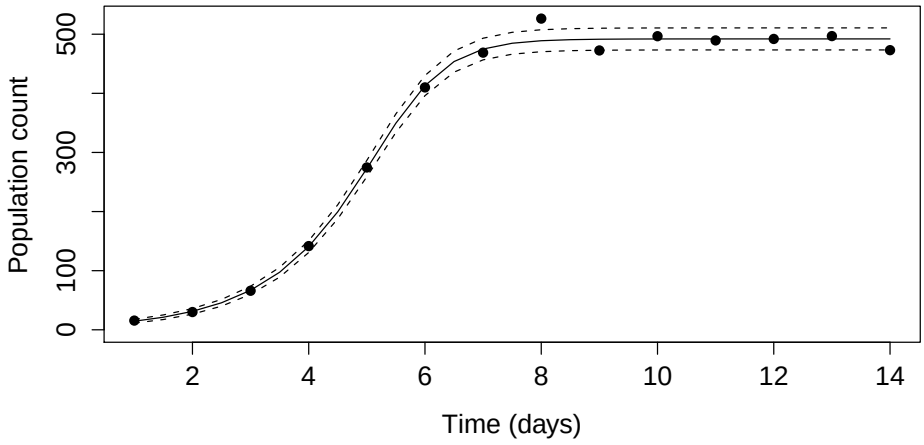

Figure 2: Maximum Likelihood fit of the θ-logistic model to data on population growth of *Paramecium*. Points are the experimental data; the curves are the model trajectory with estimated parameters and initial population, ± 2 times the assumed standard deviation of measurement errors conditional on population size.

```
    parms=c(r,K,theta); times=1:14;
    out=lsoda(x0,times,ThetaLogist,parms); Nhat=out[,2];
    -sum(dnorm(xvals-Nhat,mean=0,sd=0.42*sqrt(Nhat),log=TRUE));
}
fitTL.MLE=mle(TLNegLogLik,start=list(r=1,K=5,theta=1,x0=15),method="Nelder-Mead")
> confint(fitTL.MLE);
Profiling...
           2.5 %     97.5 %
r      0.7135616  0.8259327
K      4.8523098  4.9900337
theta  1.7934743  3.0492361
x0    12.6354302 16.6062693
```

The results are shown in Figure 2. Observe that day 8 is problematic.

3.2 Model and fit the sampling process

In the analysis of Veilleux's data, our model of the sampling process led us to the error model

$$\hat{N}(t) - N(t) \sim \text{Normal}(\mu = 0, \sigma^2 = c^2 N(t)) \tag{8}$$

His data gave us the values of c. Lacking those data, we could have assumed (8) with c as an additional parameter to be fitted. This is a very common approach: posit a sampling-error model with some unknown parameters, and estimate the sampling parameters along with the process-model parameters. Then after the model is fitted, we can check whether the residuals appear to satisfy the assumptions of the sampling model.

Ideally the sampling model should have a mechanistic justification, but this may be difficult without direct information on the sampling process. For example, for case reports on human infectious diseases, under-reporting is common.

1. The simplest model for this source of error is that each case is reported independently with probability p, where p is either constant or has some smooth deterministic trend (e.g., a linear increase or decrease. This leads to a Binomial likelihood with p as one of the fitted parameters, or (for example) a linear trend model $p(t) = p_0 + p_1 t$ where p_0, p_1 are fitted parameters. Under this model, the magnitude of sampling variation (relative to the actual number of cases) is a decreasing function of population size (error variance is proportional to number of cases, so the variance of (error/cases) is inversely proportional to the number of cases).

2. Another potential source of error is random variability over time in $p(t)$. This component of sampling error will have variance proportional to the number of cases squared, so its importance relative to the number of cases will be independent of the number of cases.

A fit using the first likelihood will "try harder" to match outbreak peaks than outbreak troughs, in terms of relative error, while the latter puts peaks and troughs on an equal footing. He et al. (2010) combined both sources of error through a discretized Normal likelihood for measles case reports. With C the number of cases, R the number of reported cases, and p the reporting rate, their model is

$$R = \text{nearest integer to Normal}(\mu = pC, \quad \sigma^2 = Cp(1-p) \;\; + \;\; \psi^2 p^2 C^2) \tag{9}$$

The first term in σ^2 is the binomial sampling variance, the second is the variance that would result from random fluctuations in p, and both p and ψ were fitted parameters.

3.3 Two-stage error modeling

Under the assumptions of trajectory matching, the residuals (deviations between the data and fitted model trajectories) are the result of sampling error. So a strategy for modeling the sampling errors is to do a pilot fit (say, by least squares), regard the residuals as estimates of the sampling errors, and use those "data" to develop a model of the sampling variability.

Miller et al. (2006) used this approach to fit models for chronic wasting disease in captive mule deer populations. The data were cumulative mortalities over the course of two outbreaks. They fitted a series of models with different assumptions about transmission by maximum likelihood, and used AIC to compare the support for different models. The model below assumes indirect transmission through an environmental reservoir E, and M is the cumulative number of deaths due to the disease:

$$\begin{aligned} dS/dt &= a - S(\gamma E + m) \\ dI/dt &= \gamma SE - I(m + \mu) \\ dE/dt &= \varepsilon I - \tau E \\ dM/dt &= \mu I \end{aligned} \tag{10}$$

They fitted the models by least squares, computed the standard deviation σ_{obs} of residuals between M (the model prediction) and the data, and from this defined the Gaussian likelihood

$$M_n \sim \text{normal}(\hat{M}_n, \sigma_{obs})$$

In effect this is probably not very different from assuming Gaussian errors and estimating the model parameters and error variance simultaneously.

A related approach is to seek a variance-stabilizing transformation. For example, if error variance is proportional to the square of the true value, then the error variance will be approximately constant if the data are log-transformed. The strategy is to fit (for example, by least squares) on a transformed scale, plot residuals versus fitted values, and look for any trends in the size of residuals. Once the variance has been stabilized, a histogram of residuals can guide the choice of a statistical model for the error distribution.

Exercise 3.1 Modify the `TLNegLogLik` function so that the constant c in the error variance model is fitted rather than assumed, and refit the model estimating c along with the other parameters. Are the results consistent with the value of c estimated directly from replicate samples? What effect does fitting c have on the `confint` confidence limits for the other parameters?

Exercise 3.2 The file `PlagueBombay.csv` gives weekly mortality for the plague outbreak in Mumbai, December 1905 to July 1906 (data from Kermack and McKendrick 1927). Use those data to fit a closed-population SIR model for plague in the rat population,

$$\begin{aligned} dS/dt &= -\beta SI \\ dI/dt &= \beta SI - \gamma I \\ dR/dt &= \gamma I \end{aligned} \tag{11}$$

using Maximum Likelihood trajectory matching of $\mu I(t)$ to the weekly number of human deaths. Here S, I, R refer to proportions in the rat population, and μ is a scaling factor beween I in rats and the human weekly death rate. You can assume that $S(0) = 1 - I(0), I(0) \ll 1, R(0) = 0$. The parameters to fit are $(\beta, \gamma, I(0), \mu)$. A reasonable *a priori* error model is a Poisson distribution: the recorded human deaths in week t has a Poisson distribution with mean $\mu I(t)$.

Once you get a fit,

(a) Draw a plot like Figure 3. The solid curve is $\mu I(t)$ from solving the model with the estimated parameters, and the dots are the weekly number of deaths.

(c) Compute the residuals on square-root scale, i.e. (human deaths)$^{0.5}$- $(\mu I(t))^{0.5}$, and plot them as a function of $\mu I(t)$. If the Poisson distribution is valid, the residuals on square-root scale should have constant variance. Does that seem to be true?

(c) Draw a contour plot of the likelihood function as a function of β and γ near the estimated values of those parameters, with the other parameters held fixed at their estimated values. What does the contour plot say about the identifiability of β, γ? What does it say about the identifiability of $R_0 = \beta/\gamma$?

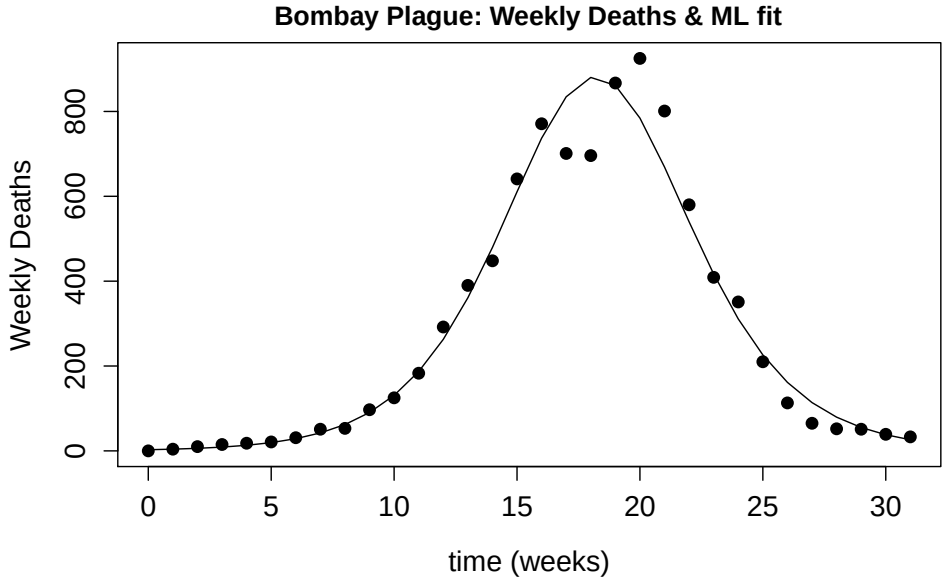

Figure 3: Maximum Likelihood fit of the rat population SIR model to weekly mortality for the plague outbreak in Mumbai, December 1905 to July 1906, using a Poisson model for sampling variability. The solid curve is $\mu I(t)$ from the fitted model, and the dots are the reported numbers of human deaths per week.

Advice: All parameters are necessarily positive, so it helps optimization if negative values are impossible. One way of doing that is to work with the logs of parameters. For example, define $b = \log(\beta)$, and write the model with $dS/dt = -e^b SI$, and so on.

Exercise 3.3 The file `CholeraDaccaExercise.R` contains an SIR^3 model for cholera deaths in Dacca, fitted by least squares to the first 20 years of historical data provided by Menno Bouma via Mercedes Pascual. The seasonal variation in contact rate is defined by three parameters: the mean, amplitude, and phase of $\beta(t)$ modeled by a sine wave with two peaks per year. Estimate these parameters by maximum likelihood trajectory matching to the first 20 years of data. Then go on to (a) using larger and larger subsets of the data OR (b) developing a better model for the seasonal variation in contact rate. Note: before you use these data for any purpose outside this workshop, you must obtain permission from Menno or Mercedes.

4 References

Bolker, B. 2008. Ecological Models and Data in R. Princeton University Press, Princeton NJ.

Cox, D.R. and D.V. Hinkley. 1974. Theoretical Statistics. Chapman and Hall, London.

He, D., E. L. Ionides and A. A. King. 2010. Plug-and-play inference for disease dynamics: measles in large and small populations as a case study. J. R. Soc. Interface 7: 271-283.

Jost, C. and S. P. Ellner. 2000. Testing for predator dependence in predator-prey dynamics:

a nonparametric approach. Proceedings of the Royal Society of London Series B: Biological Sciences 267: 1611-1620.

Konishi, S. and G. Kitagawa. 2008. Information Criteria and Statistical Modeling. Springer, New York.

Miller, M.W., N. T. Hobbs, S.J. Tavener. 2006. Dynamics of prion disease transmission in mule deer. Ecological Applications 16: 2208-2214.

Veilleux, B. G. 1976. The analysis of a predatory interaction between *Didinium* and *Paramecium*. Master's thesis, University of Alberta.

Veilleux, B. G. 1979. An analysis of the predatory interaction between *Paramecium* and *Didinium* Journal of Animal Ecology 4: 787-803.