Lecture 14. Nonlinear least squares and introduction to model selection.

A brief into to nonlinear models

The next major topic I want to cover is model selection. But before we do that I would like to introduce nonlinear models, because they provide a particularly clear example of the importance of model selection, and because it's important to know how to fit them anyways.

We've already been fitting nonlinear curves when we use GLMs, because the link functions are nonlinear. However, these are a special case of nonlinear models, because we are specifying a linear formula for the predictors and then connecting that to the response variable with the nonlinear link function. For the wider universe of nonlinear curves that you might want to use in a model, we need to use a more general approach.

The approach I will cover here is called nonlinear least squares (NLS). In terms of the model structure, we fit a nonlinear relationship between the response and the predictor(s), and assume the data is normally distributed around the expected value of the response:

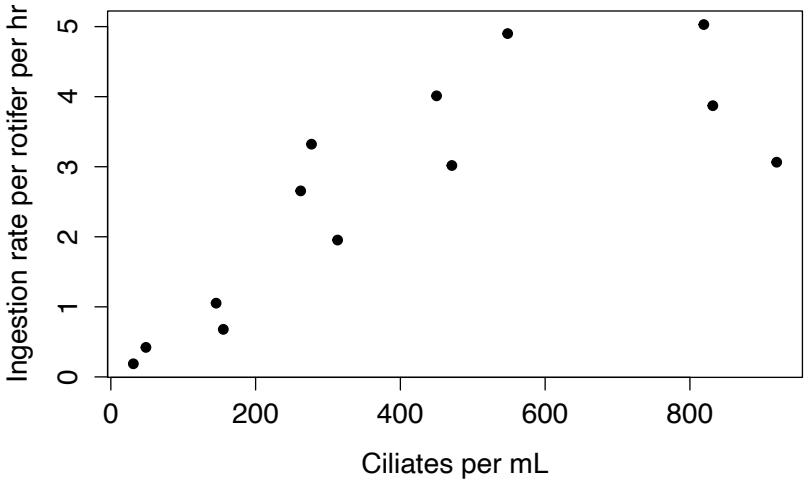
$$\mu_i = f(X_{1i}, X_{2i}, \ldots)$$
$$Y_i \sim \text{Normal}(\mu_i, \sigma)$$

Where Y_i is observation i, μ_i is the expected (mean) value for Y_i, f is some nonlinear function of the predictor(s) X_{1i}, X_{2i}, etc., and σ is the standard deviation of the 'noise' around μ_i. Examples of nonlinear functions you might encounter include the curves I covered in lecture 3, such as the Michaelis-Menten curve $v(S) = \frac{V_{\max} S}{K+S}$.

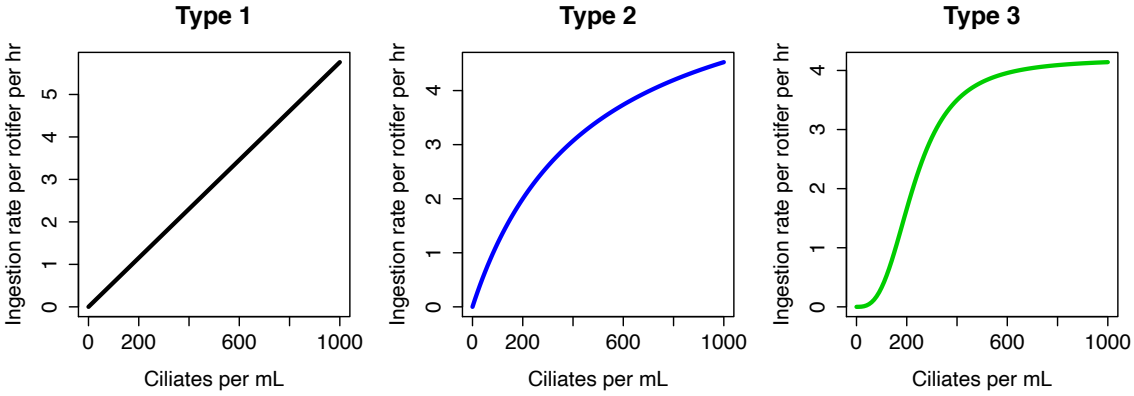
Models like this can be fit with maximum likelihood methods just like the GLMs we've already covered. When we assume the data are normally distributed, the model can be fit by *minimizing the residual sums of squares*, rather than maximizing the likelihood *per se*. Hence the algorithm is called nonlinear least squares. As we've already learned, parameter estimates from a least squares fit are equivalent to maximum likelihood parameter estimates, because for normally distributed data the likelihood is maximized when the residual variance is minimized. I'll focus on the NLS method here because it is relatively easy to do in R. A nonlinear model can be fit with any of the probability distributions we've talked about, not just the normal distribution, but fitting nonlinear non-normal models is more involved than what I want to get into here. If you are interested in doing this, the book by Bolker, "Ecological Models and Data in R" is a great reference.

An NLS example

A classic example of a nonlinear function we might want to fit in biology is a functional response curve. Here are some data from a study by Mohr & Adrian, where they used culture experiments to quantify the functional response of the rotifer *Brachionus calyciflorus* feeding on the ciliate *Tetrahymena pyriformis*. Ciliate cultures of different density were created, and the rate at which the rofiter ingested the ciliate was measured at the different prey densities.



The authors used this data to fit different potential functional response curves and compare how well they fit the data. To keep things simple I will modify their approach, and fit these three responses:



Type 1 is a linear functional response, $f(R) = aR$, where R is the prey density, a is the attack rate, and f is the per capita rate of ingestion by the predator. The Type 1 response assumes that the number of ciliates eaten by a rotifer is directly proportional to the number of ciliates it encounters, and there is no limit to how many ciliates a rotifer can eat per unit time. The lack of an upper limit seems impossible, but this is a useful scenario for comparison, and if the prey never get very dense in nature then the functional response may be approximately linear. The

Type 2 functional response is the saturating curve I introduced in lecture 3, $f(R) = \frac{aR}{1+ahR}$. This curve has an additional parameter *h*, handling time. There are different ways to derive a curve like this, and the parameters may be interpret as processes other than 'attack rate' and 'handling time'. But the important point is that as prey density increases, something (the time it takes to subdue, ingest, digest prey, etc) causes the predator's consumption rate to saturate. The third curve, Type 3, adds a further parameter *k*: $f(R) = \frac{aR^k}{1+ahR^k}$. The parameter *k* does not have as concrete interpretation as *a* and *h*, but the effect of adding this parameter is that the curve becomes sigmoid, and the degree of sigmoid-ness increases as *k* increases. At low prey density, the functional response is not approximately linear (as in Type 2); instead, at low density the consumption rate actually increases at an accelerating rate. Essentially, the predator becomes ineffective at low prey density. There are many possible causes for this: predators invest less energy in foraging at low prey density because the energy gained is not worth the energy cost; at low prey densities the predator switches to feeding on something else (that we haven't measured); prey density needs to be higher for the predator to form an effective search image (for visual predator).

We would like to use the experimental data to fit these three curves. Functional responses are a key component in modeling and predicting the outcome of predator-prey relationships, and the different functional response types have different consequences for things like the stability of population dynamics. For example, a Type 3 functional response tends to produce more stable population dynamics than a Type 2, which is more likely to lead to cyclic or chaotic population dynamics.

After this digression into theoretical ecology, how do we actually fit these curves? The nls() function is quite convenient, you just need to specify the formula for the curve you want to fit. For example, for the Type 2 curve:

```
type2 = nls(ingestion ~ a*ciliates/(1 + a*h*ciliates), data = rotifer,
start = list(a = 0.01, h = 1/4))
```

In the dataframe rotifer, the columns are named 'ingestion' and 'ciliates'. To fit the type 2 functional response I just write the correct formula, including the parameters 'a' and 'h'. I could have given these two parameters any name I want. The other part we haven't seen before is the argument 'start'. These are the starting values for the algorithm that searches for the least-squares (maximum likelihood) parameter estimates. Recall that for linear models (lm), there is a formula that solves for the least squares parameter estimates. For NLS models there is no such formula, so the model is fit using an algorithm that searches through parameter space until it can no longer reduce the residual sums of squares. When we specify an nls() model, we need to give the function some parameter values from which to start the search. It

is important to provide values in the right ballpark, because the algorithm can run into problems such as stopping on a local peak of the likelihood surface that does not yield the highest possible likelihood. For this reason it is also important to plot the fitted curve and see if it looks like the algorithm converged on the right answer. It's noteworthy that the GLM models we've been using also use an algorithm to find the maximum likelihood parameter estimates, but those models use a specialized algorithm for GLMs that generates its own start values, so we didn't have to worry about it.

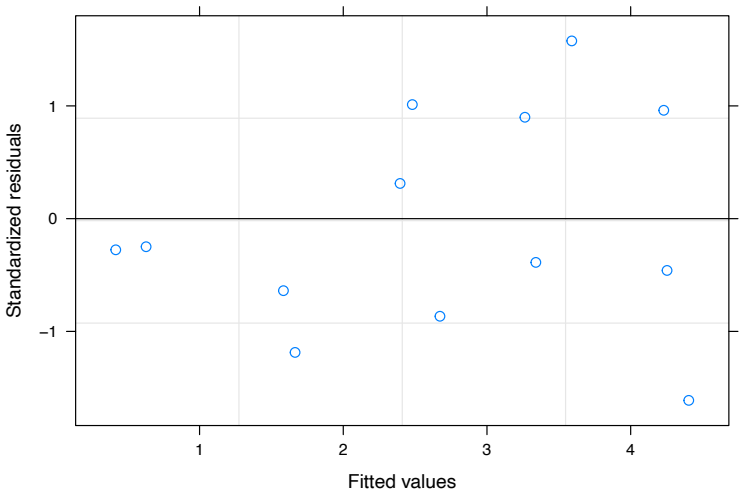
The start values can be specified as a list (what I did above), or as a named vector. I chose 0.01 for *a* because that is roughly what the initial slope of the curve looks like (roughly 2/200). I chose ¼ for *h* because 1/*h* is the asymptote of this curve, and the curve looks like it levels out around 4. Let's look at the fit:

```
summary(type2)

##
## Formula: ingestion ~ a * ciliates/(1 + a * h * ciliates)
##
## Parameters:
##   Estimate Std. Error t value Pr(>|t|)
## a  0.01431    0.00499    2.87   0.0153 *
## h  0.15111    0.04474    3.38   0.0062 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.831 on 11 degrees of freedom
##
## Number of iterations to convergence: 10
## Achieved convergence tolerance: 5.02e-06
```

Similar info to an lm() fit, but with some added info on the bottom about how the search algorithm worked. The estimate for *a* is 0.014 mL h^{-1} (yes these are the same units as a clearance rate, you can think of *a* as the maximum clearance rate, seen at the lowest prey density). The estimate for *h* is 0.15 h^{-1}, which might mean that it takes a rotifer 9 minutes to capture and ingest a ciliate (if that interpretation of the curve is accurate). In any case it means that the ingestion rate saturates at 1/*h* = 6.67 per hour.

Just like for other kinds of models, it's important to look at some residual diagnostics:

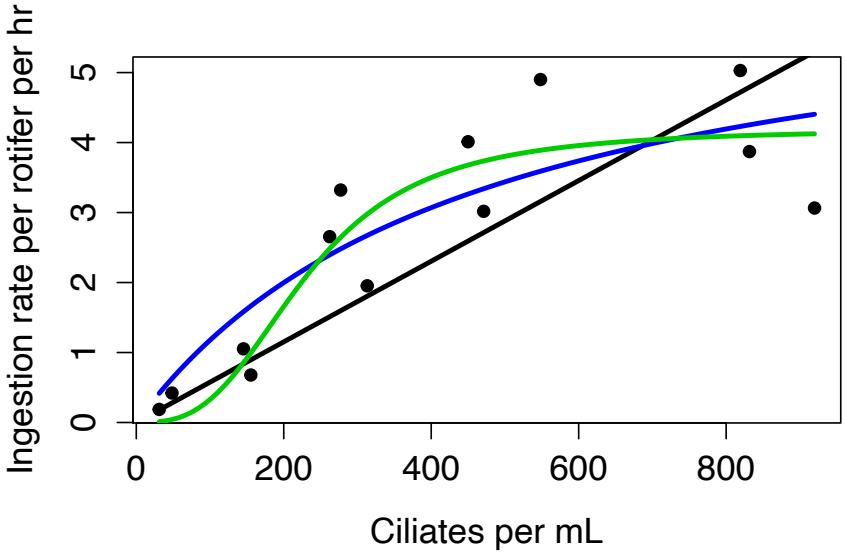


This looks pretty good. There is a hint that the variance in the residuals may increase as the fitted values increase. This would not be surprising, because the ingestion rate is bounded below by zero. I've treated the response as normally distributed because that is easier and looks reasonable, but if there was a strong deviation from homoscedascticity we would have to consider a more complex model.

We can also fit the Type 1 and Type 3 functional responses:

```
type1 = nls(ingestion ~ a*ciliates, data = rotifer, start = list(a = 0.01))
type3 = nls(ingestion ~ a*(ciliates^k)/(1 + a*h*(ciliates^k)), data = rotifer, start = list(a = 0.01, h = 1/4, k = 1))
```

And take look at all the fitted curves:



Each of these curves does a decent job of fitting the data. There definitely seems to be some nonlinearity that is not captured by the Type 1 curve. The relative merits of Type 2 vs Type 3 is harder to assess. How can we decide which curve is a better fit to the data? We could compare R^2, but we know that R^2 tends to increase with the number of parameters. In addition, R^2 is actually problematic for nonlinear relationships, for reasons I won't get into here (note that summary() does not give you an R^2, unlike for lm() fits). In this case it would be possible to do a likelihood ratio test, because Type 2 is a special case of Type 3 ($k = 1$). But for many other situations we might want to compare curves that are not nested in this way. In addition a LRT would tell us if the parameter k is different from zero, but it would be nice to say something more direct about how much evidence we have for the three different options, something more than a p-value from a null hypothesis test. To answer these questions we need to learn some techniques for model selection.

Introduction to model selection with information criteria

What we've covered so far is relatively straightforward and uncontroversial, once you're learned the methods. Fitting generalized linear models (or nonlinear least squares) is mostly a matter of choosing the right probability distribution, using the canned software to get maximum likelihood parameter estimates for the predictors in the model, and using likelihood profiles and/or likelihood ratio tests to quantify parameter uncertainty and null hypothesis tests for the predictors. Predictive performance can be assessed with (pseudo-)R^2 and cross-validation.

All of these procedures assume that you already know the model you want to fit/use. It is more challenging to answer questions like these:

1. I have multiple models that represent distinct hypotheses (e.g. different functional response curves). Which is best supported by the data?

2. I have a set of predictors that are all hypothesized to be important for the response. Which are supported by the data? What is their relative importance? Should all be included in a single model, or should a smaller model of 'significant' predictors be used? How should such a smaller model be chosen?

3. I have a large number of predictors that may or may not be important, and I want to do an exploratory analysis to see which are best supported by the data. How do I construct model(s) to choose among them and quantify their importance?

I tend to think of 1-3 as three distinct but related kinds of analyses:

1. Different models represent different hypotheses
2. Different predictors represent different hypotheses (but a particular model could include one or more predictors)
3. Exploratory analysis / data mining / data dredging (not comparing *a priori* hypotheses, sifting for relationships).

These are all problems of *model selection*, because they involve constructing and comparing multiple models fit to the same data. Model selection does not have the same kind of well-established and (relatively) uncontroversial foundation as maximum likelihood estimation. Different statisticians and scientists use different approaches, new methods are still being developed, and articles are commonly written that argue for the superiority/flaws of different approaches. My goal is to present some of the most common methodology and discuss its relative merits, but be aware that this is probably the most subjective part of the course.

The limitations of null hypothesis tests and p-values

Null hypothesis tests are ubiquitous in science and also frequently critiqued. We've been using them plenty in the course so far. A likelihood ratio test compares the likelihood of our chosen model to the likelihood of a model that represents a null hypothesis. Usually the null hypothesis is that some parameter, or multiple parameters, is equal to zero, which is equivalent to removing the parameter(s) from the model to make a 'restricted' model. We than use the approximate sampling distribution of the likelihood ratio (chi-square or F distribution, depending on the circumstances) to ask 'If the null hypothesis were true, how frequently would we observe a likelihood ratio $\geq$ the observed likelihood ratio?'

This is all fairly standard, so what is the critique? Some of the critiques are related to model selection, and some are more general. Now is an appropriate time to cover both; here are the more general criticisms:

1) The null hypothesis may be trivial, or is usually trivial. The idea here is that it usually unlikely that some parameter is *truly* zero. If we remove limpets from the rocky intertidal, it is likely the removal has *some* effect, even if very small, on the other species in the community. If we're asking what drives chlorophyll concentration in the ocean, then nearly anything we might measure probably has some relationship with chlorophyll, albeit small. Furthermore, if we have a large enough sample size, we will be able to detect small effects, while if we have a small sample size we will only be able to detect large effects. So p-values mostly reflect how much data we have, and using the data to look for evidence against the null hypothesis doesn't tell us very much, because we know *a priori* that the null is unlikely to be true.

This is an interesting critique, especially for biological systems where anything we measure is likely affected by many other processes, with some large effects and many small effects. I think it's important to keep this critique in mind when doing null hypothesis tests. Personally, I still find p-values to be useful. Data are often limited, and p-values are useful as a reality check on whether an effect I'm interested in is really important enough to be detectable in the data, or whether I'm just imagining that it is important. But it is essential to remember what p-values are (a measure of evidence against the null hypothesis), not enshrine them as some magical truth about reality, and to combine p-values with a measure of effect size, which is the next point.

2) P-values distract from looking at effect size. Instead of saying 'what is the magnitude of the effect of limpets on snails', I might just say 'the effect was signficant ($p < 0.001$)'. In this case I would be focusing on 'significance' to the exclusion of thinking about what the estimated relationships are. This is certainly a valid critique, but mostly it means that model results should be plotted or at least discussed in table format. And the results should be interpreted in terms of whether the magnitude of the effects are likely to result in a substantial or minor influence on the process we're studying. This is why I ask you to plot raw data and fitted relationships, and think about the magnitude of the effects, for every analysis in the homework assignment.

3) P-values encourage the use of arbitrary thresholds ($p < 0.05$, $p < 0.01$, etc). This is certainly true, and is further reinforced by journal editors/reviewers, depending on your field/journal. But mostly this means that full p-values should be reported, and effect sizes should be visualized and discussed.

The three criticisms just listed apply to null hypothesis tests in general. Personally I often use them anyways, but with some skepticism and as one tool among many. If you think p-values really do seem to be kind of useless, then you could abandon them entirely and use the alternative methods I'll cover shortly. But first I'll list two more critiques of null hypothesis tests which are more specific to the issue at hand, which is model selection.

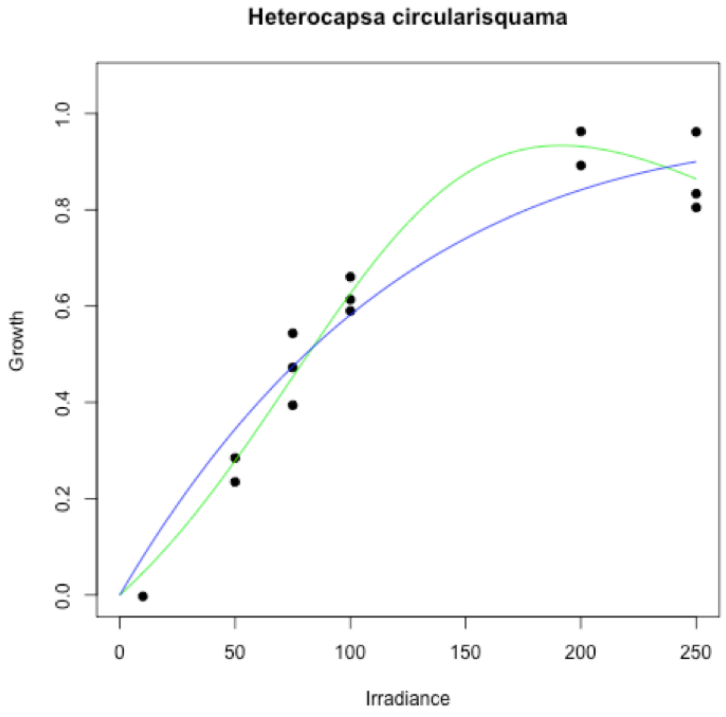
4) Null hypothesis tests can't compare non-nested models. Nested models refer to the case where one model is a special case of another model. For likelihood ratio tests, the restricted model is a special case of the full model. Sometimes we would like to compare models that are non-nested alternatives. For example, in a manuscript I'm working on I fit growth-irradiance relationships for phytoplankton using this curve:

$$\mu(I) = \frac{\mu_{\max} I}{\frac{\mu_{\max}}{\alpha I_{\text{opt}}^2} I^2 + \left(1 - 2\frac{\mu_{\max}}{\alpha I_{\text{opt}}}\right) I + \frac{\mu_{\max}}{\alpha}}$$

but a reviewer wanted to know how well this model fit the data relative to another common growth-irradiance model:

$$\mu(I) = \mu_{\max}\left(1 - e^{-\frac{\alpha I}{\mu_{\max}}}\right) e^{-\frac{\beta I}{\mu_{\max}}}$$

Here's an example of the fit of the two curves to an experiment, with the first curve in green:



These curves have similar but not identical shapes, and they are distinct nonlinear functions that do not have a nested relationship. We can't ask 'is the first curve significantly better than the second curve?' using the standard likelihood ratio test (there is actually a special kind of LRT that can test this question, but it is rarely used and I won't discuss it here). Even if we could get a p-value for comparing these models, we would also like some sense of the relative support for the two models from the data. Something like R^2 might be helpful, though when models have different numbers of parameters we would be concerned about overfitting, and R^2 has some weird properties for nonlinear relationships and so should be avoided in this case. Cross-validation would be a good option for comparing the models in terms of their predictive performance, and I think that would actually be the best option for comparing these models, out of the techniques we've discussed so far. You may not be interested in prediction *per se*, but as we'll shortly, predictive performance is closely related to other measures of model support.

5) Null hypothesis tests are a poor way to select among many possible predictors.
In a homework assignment you analyzed Sole presence/absence as a function of salinity. The survey data included a number of other environmental variables, such as depth, temperature, month, %mud substrate, %gravel substrate, sampling site, etc. Maybe we want to test for the relative importance of these different variables on Sole occurrence. We may have a set of variables that represent *a priori* hypotheses, or we may just want to dredge through all the predictors we have to look for important associations. And often analyses lie somewhere between these two extremes, because we have *a priori* hypotheses we want to test but we also want to see if there are any surprises or new insights to be gained from the data

How can we evaluate the support for each predictor from the data? This is a surprisingly tricky problem. Perhaps we should just fit a model with every predictor we're interested in, like this:

```
big.model = glm(Solea_solea ~ salinity + depth + temperature + month +
mud + Area + gravel, data = solea, family = binomial)
```

We could use this model to do marginal likelihood ratio tests on all the predictors, to see if their effect is significantly different from zero. This is certainly a valid thing to do, and in the past I have used this approach myself. But here are some downsides to fitting one model with all predictors of interest:

A) *Correlated predictors*: as we discussed earlier, problems can arise when predictors are partially correlated with each other. Correlated predictors can make it so that the estimated effect of one predictor depends on which other predictors are in the model, and this only becomes more acute when the number of predictors is large.

B) *Parameter uncertainty*: This is a related issue. When we have a large number of predictors relative to how much data we have (in this case, 7 predictors for 62 observations), adding more predictors can make the coefficient estimate for each predictor more uncertain (i.e., larger confidence intervals). The way I think about this is that there is only so much variation in the data to explain, and as you add predictors the variation in the data that can be uniquely attributed to any single predictor will tend to diminish. An extreme case occurs when we have more potential predictors than observations. This case is outside of the scope of this course, but it is a situation that arises in the prediction-focused data mining approaches usually called *machine learning*.

So it seems like it might be advantageous to use a reduced model, probably one that drops 'unimportant' predictors and allows the others to be estimated with more confidence. And it might also be good to compare a variety of models, due to issues with correlated predictors. Is there a good way to do these things?

In the past it was common to make a reduced model using *stepwise regression*. There are different ways to do this, but the principle is the same. You could start with a full model, do a marginal LRT for each term, and drop the term with the largest p-value (i.e. the least 'significant' term). Then iterate this algorithm until the only terms remaining are all 'significant' according to some predefined threshold. This kind of algorithm is called *backward elimination* or backward selection. Alternatively you could start with a model with no predictors, and sequentially add the most-significant predictor (lowest p-value), until no more significant predictors remain to be added. This is *forward selection*. Finally, there are algorithms that combine forward and backward steps.

Although stepwise regression is fairly common, it is much maligned by statisticians. There is no real theoretical framework that explains why this approach should give you the 'right' answer, or what the 'right' answer means in this context. Using backward or forward selection, or a mix, can give you different 'best' models. In addition, the model you get at the end will tend to be overfit, and the degree of overfitting will be more extreme with more predictors to sift through. Overfitting occurs because by sorting through predictors based on their p-values, you are greatly increasing the chance of finding whatever spurious patterns are in the data by chance.

The information theory approach

We've covered some of the criticisms / limitations of null hypothesis testing and p-values. For some situations (e.g. testing experimental manipulations, testing regression models with a small number of predictors) I tend to think p-values are fine as long as you are aware of the limitations and focus on the model-fitted effect

sizes and what these actually mean biologically. However, there are model selection situations (non-nested models, models with many predictors) that are hard to manage in the classic frequentist framework. Statisticians have come up with a variety of alternative model selection methods. For this class I will focus on the information theory approach based around AIC, as popularized by Ken Burnham and David Anderson. This has emerged as the most popular alternative to the more traditional methods in many areas of biology and elsewhere. At the end of the course we will get into Bayesian methods as well, which are the other main alternative.

The approach we're going to cover now essentially represents a different paradigm for doing inference with statistical models. So this is going to get a bit philosophical, before we get concrete about calculations and procedures.

The information theory approach starts by assuming there is some truth out in the world that creates the data we observe. Mathematically you can think of this as an imaginary model that could perfectly simulate reality. For example, if we're thinking about rotifer functional responses, there are some complex rules that determine exactly how quickly a rotifer can ingest ciliates, and these are determined by things like rotifer behavior (which is genetically coded and maybe learned as well), the fluid dynamics and biomechanics that underlie locomotion and prey capture, etc. If we knew how all of these components worked, we could write down a mathematical model that perfectly describes the rotifer functional response. Let's call this model *M*.

In reality we have some different proposed models that we can fit to experimental functional response data. Let's call those models, m_1, m_2, etc. *We can think of the model selection problem like this*: we want to know how close each model m_i is to the true model *M*. And in a certain sense, a model *m* that is closer to the true model *M* is a better model. How can we quantify the distance between a model and the truth? Important methods for answer this have come from information theory.

Information theory is a branch of mathematics invented in the mid-20th century to quantify things like data compression and signal processing. Later it was applied to statistical problems as well, because a statistical model is an approximation of reality in the same way that a low-resolution image is an approximation of a high-resolution image:

(note I did not make this image).

When a low-res format is used to capture an image, some information is lost. In the same way, when we use a statistical model to represent biological/physical/chemical processes, some information is lost. However, a model that better approximates reality (the true model *M*) will lose less information. So the mathematical question is how to quantify the information loss of different models.

Akaike Information Criterion

We're imagining that there is some true model *M* that perfectly represents how reality generates the data we collect. We would like to know how much information is lost when we approximate reality with some model m_i. But if we don't know this true model *M* (and we never will), how do we know how close m_i is to it? In 1974 Akaike showed that even though we don't know the true model, we can still quantify the *relative* information loss of different models. He humbly called this *an information criterion*, but it has come be known as the *Akaike information criterion* (AIC). The formula is suprisingly simple:

$$\mathrm{AIC} = -2 * \log\left(L\left(\hat{\theta}\middle|Y\right)\right) + 2K$$

Here $L\left(\hat{\theta}\middle|Y\right)$ is the likelihood of the fitted (MLE) parameters $\hat{\theta}$ given the data *Y*, and *K* is the number of parameters in the model. So calculating AIC just requires the log-likelihood of the fitted parameters and the number of parameters in the model. What is AIC telling us? This number estimates the relative information loss of different models; in other words, it quantifies how well different models approximate the unknown true model *M*. A smaller AIC indicates a 'better' model.

Why is a smaller AIC better? Let's think about the two parts of this formula. The first term is the log-likelihood, multiplied by -2. If a model does a better job of fitting the data, then that will increase the likelihood of the fitted parameters, because the

observed data is more probable under those parameter values. This will make the term -2*log-likelihood smaller. *So a better-fitting model will tend to have a lower AIC*. The second term is the number of parameters multiplied by 2. Therefore *a model with more parameters is effectively penalized for that complexity*.

Putting these two pieces together, we can think of AIC as accounting for the tradeoff between underfitting the data (the likelihood will be smaller if too few predictors or the wrong functional forms are included) and overfitting the data (adding more parameters will tend to increase the log-likelihood, even if those parameters are only fitting the noise, but adding more parameters will also increase 2*K).

AIC can only be used to compare models fit to the same data. It is based on likelihoods, and the likelihoods from different models are only comparable if they are calculated from the same data.

AIC is derived assuming a large sample size (technically, it is an asymptotic approximation as sample size n -> Infinity). When sample size is smaller, it is recommended to use the bias-corrected AICc:

$$\text{AICc} = -2 * \log\left(L\left(\hat{\theta}|Y\right)\right) + 2K\left(\frac{n}{n-K-1}\right)$$

You can see that as n gets large relative to K, the second term -> $2K$, as in AIC. When n is small relative to K, the second term is larger than $2K$, which effectively penalizes more complex models more strongly.