Influenza Models and data fitting

2017 MITII Summer School on Immune Response Modeling
Author: Andreas Handel, Department of Epidemiology and Biostatistics, University of Georgia
ahandel@uga.edu

Mechanistic modeling – why?

- **Understanding**
  - We can use mechanistic models to gain conceptual insights into a system's behavior. E.g. how does some level of drug treatment affect the infection outcome.

- **Prediction & What-if scenarios**
  - We can make specific testable predictions
  - We can perform virtual experiments that would be unfeasible to do (costly, lengthy, unethical)

- Those two approaches are "data-free".

- It is useful and valuable to build and explore models in a "data free" manner, but for maximum impact/usefulness, at some point they need to be brought in contact with data.

Mechanistic modeling – why?

- **Hypothesis testing & Parameter Estimation**
  - We can use mechanistic models to test different mechanisms/hypotheses
  - For hypothesis testing, one can compare models with data in a qualitative manner (e.g. action of IFN on HCV, features that the simple HIV model can't reproduce).
  - A step further, one can compare models with data in a rigorous, quantitative manner by fitting models to data (inference).
  - We can also use inference to estimate parameters that are not directly measurable.

Why fit mechanistic models to data

▸ We can use model fitting to rigorously test possible mechanisms, hypotheses and causal relationships.

$$\dot{U} = \lambda - dU - (1-f)bUV$$

$$\dot{I} = (1-f)bUV - \delta I$$

$$\dot{V} = (1-e)pI - cV$$

▸ We can obtain estimates for important biological parameters that are not easily accessibly by direct measurement.

$$\dot{U} = -bUV \qquad \text{(uninfected cells)}$$

$$\dot{I} = bUV - \delta I \quad \text{(infected cells)}$$

$$\dot{V} = pI - cV \qquad \text{(free virus)}$$

Possible outcomes – "Failure"

- If a model does not fit the data well, we suspect that the hypotheses/biological mechanisms) underlying the model (or our implementation of them) are not correct.

- If we built our model(s) based on biologically reasonable assumptions, we have learned something useful.

- **Model rejection can sometimes be very insightful – maybe even more so than having a model that fits well.**

Possible outcomes – "Success"

- If a model fits the data well, it lends support to our hypotheses about certain mechanisms/causal relations underlying the biological system.

- **But: It is possible to produce models that describe/fit the data very well but are wrong or useless.** For instance a polynomial model with enough parameters can pretty much always fit, but is useless for the purpose of gaining mechanistic insights.

- (Large) mechanistic models are often so flexible that they can fit all kinds of data, without allowing one to conclude that the mechanisms included in the model are actually the ones important for the real system.

"With four parameters I can fit an elephant, and with five I can make him wiggle his trunk." — John von Neumann

How to fit models to data

▸ Goal: minimize discrepancies between data and model results.
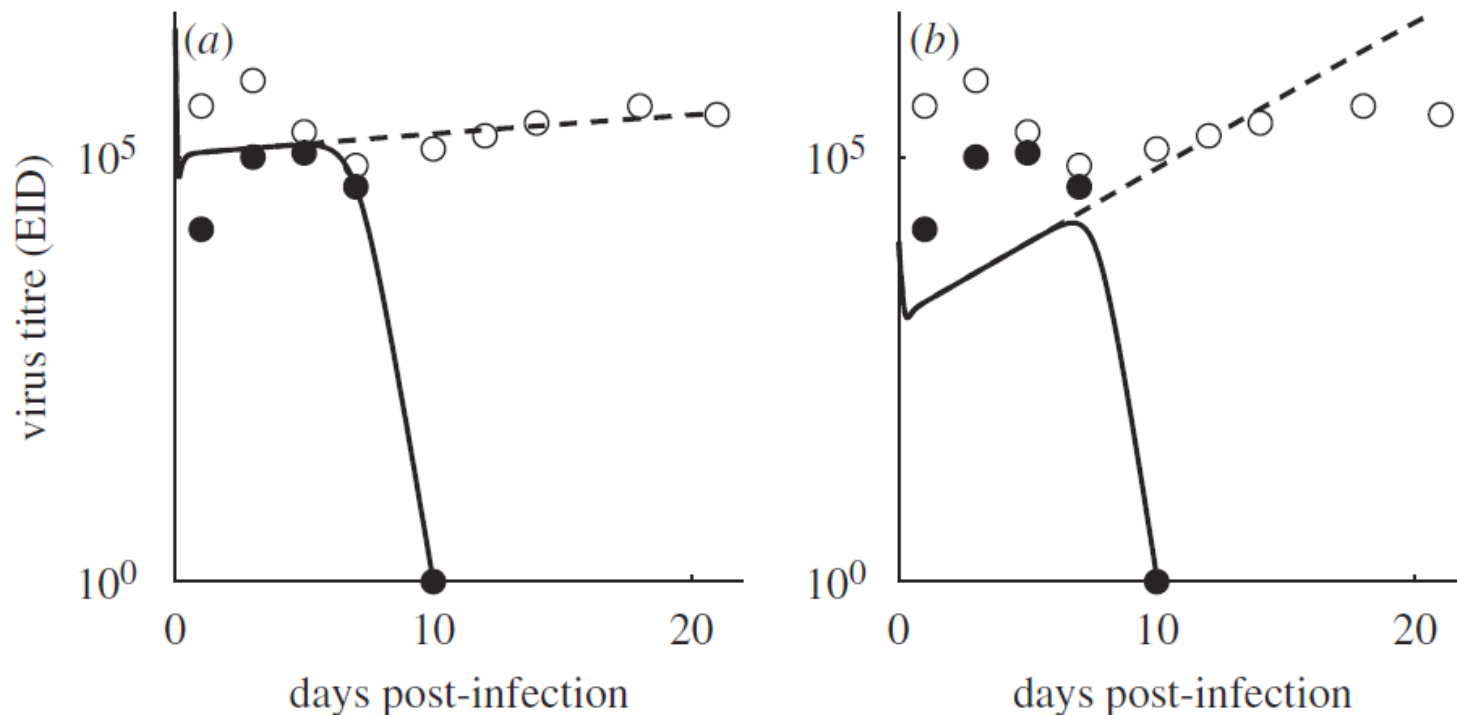
How to fit models to data

- Get yourself some data.

- Write down a model.

- Choose initial guesses for the parameters that you want to fit (e.g. U_0, b, p). Note: For fitting, unknown initial conditions of variables are treated like unknown parameters.

- Simulate your model (e.g. integrate the differential equations).

- Look at the difference between data and model (more on that next).

- Change the values for some/all parameters, integrate model again, re-evaluate differences between model and data.

- Keep doing this until changing parameters does not further reduce the differences between model and data.

- You (hopefully) found the best fit.

$$\dot{U} = -bUV$$

$$\dot{I} = bUV - \delta I$$

$$\dot{V} = pI - cV$$

How to fit models to data

- We need to mathematically specify the quality of our fit, i.e. we need to define what we mean by "a model that describes data well".

- The most general way to assess quality of model fit is by using the likelihood. This works with both in a frequentist/likelihood and Bayesian framework.

- We will only look at one special case of likelihood-based fitting, namely using **least squares** as a measure of model fit.

Objective function

▸ We have to quantify the difference between model and data to see if a given set of parameter values is better than another set.

▸ To do so, we specify an ***objective function*** (also known as loss/cost function) which quantifies the difference/agreement between the model and data.

▸ Usually, one sets up the problem such that larger discrepancies between model and data lead to a larger numerical value of the objective function. Then our goal is to minimize the objective function.

Least Squares

▸ One way of specifying an objective function (quantifying differences between data and model) is the method of **least squares**. Here, each discrepancy (d) between a data point and a model result is penalized as d^2, then all values summed.

▸ Example: Distance squared is $3^2+(-1)^2=10$ for scenario 1 and $2^2+(-2)^2=8$ for scenario 2.

data-model difference

3

-1

model

data

t_1

t_2

2

-2

t_1

t_2
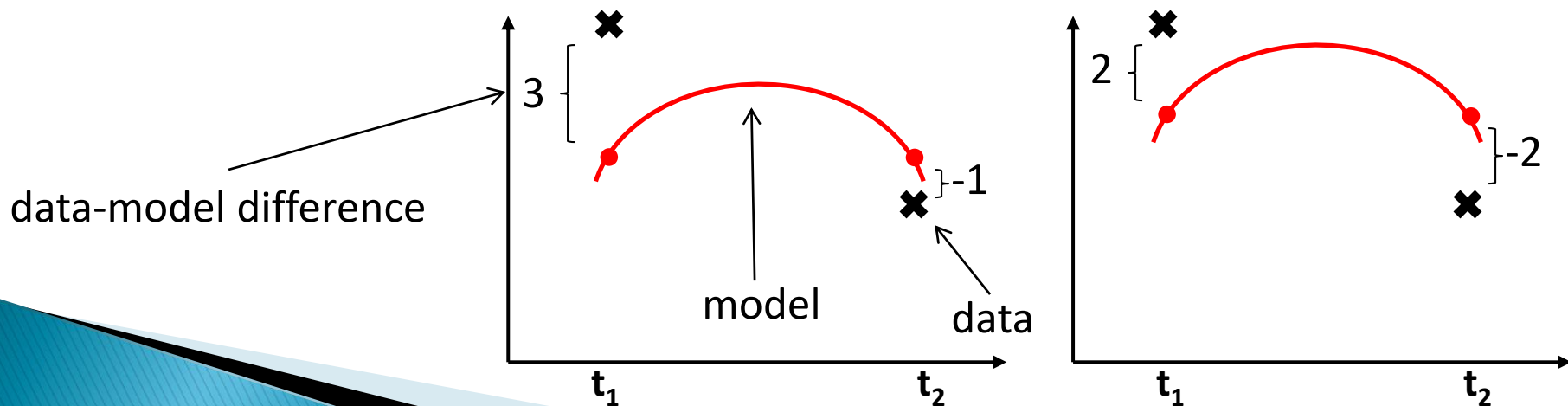
Least Squares

- It seems to make sense to increase penalty more than linearly as distance between data and model increases. Least squares does that.

$$SS = \sum_i (\text{data}_i - \text{model}_i)^2$$

- The objective function (least squares equation) is often called **SSR=Sum of Square Residuals**, RSS, SS, RSE, SE, …

- Least Squares is simple and frequently used.

- It is a special case of the likelihood approach, making the assumption that errors are normally distributed with the same SD.

- Variants of least squares, e.g. weighted least squares, robust least squares, etc. exist.

Dynamic models

- We want to fit dynamic, mechanistic models. These models could be stochastic. This makes model fitting (both theory and implementation) difficult.

- We therefore focus on models where the "true" dynamics of the system is described by a deterministic model, and variability in data only comes from errors in measurement.
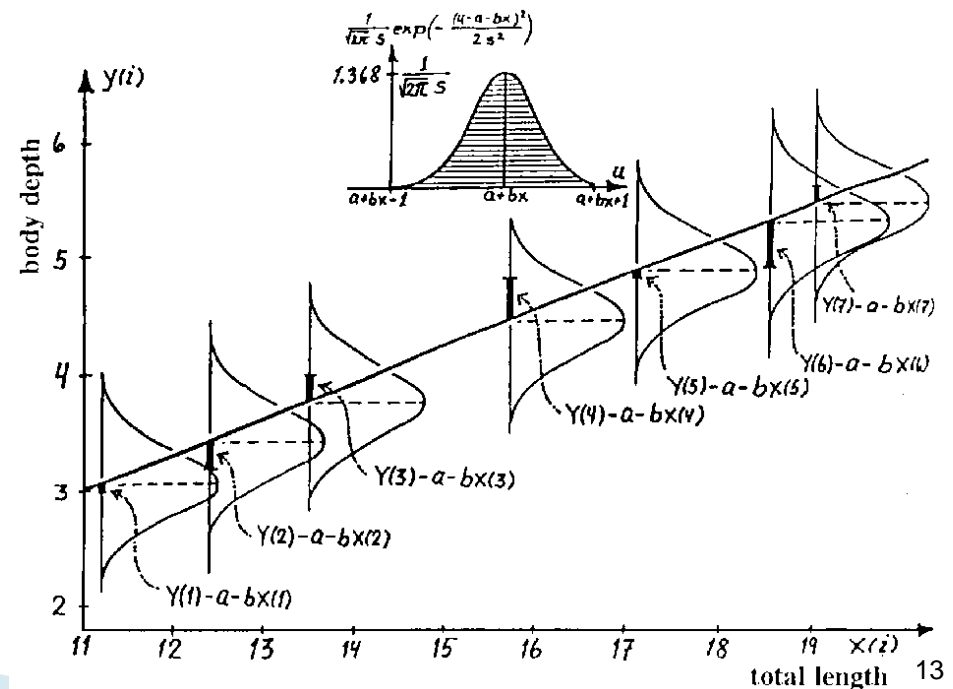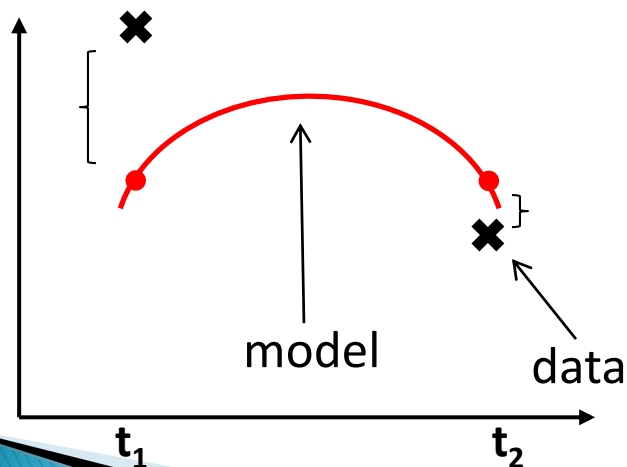
Measurement errors

▸ We will assume that errors

　◦ do not affect the dynamics of the system

　◦ are **independent** from each other

　◦ come from the same normal distribution, have the same magnitude/variance

▸ Because of these features, the objective function is the least squares equation:

$$SSR = \sum_{i} (\text{data}_i - \text{model}_i)^2$$
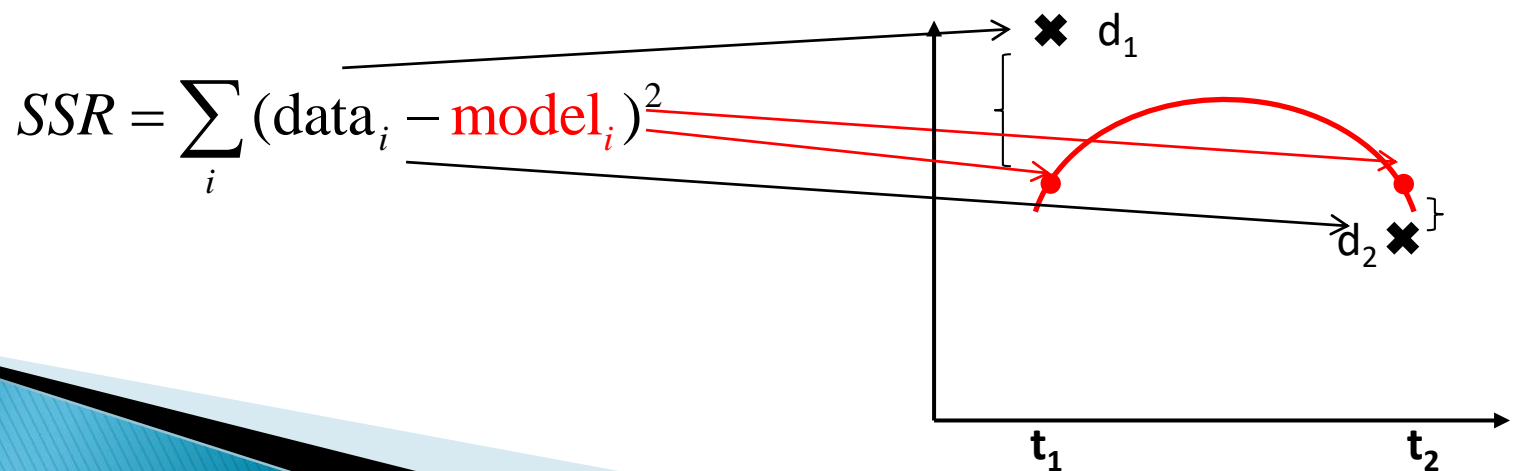
Fitting in practice

▸ The basic scheme of fitting is simple:

◦ Run model

◦ Compare model results with data using objective function

◦ Change model parameters, run again

◦ Keep doing this until you found the lowest possible value for the objective function.

Fitting in practice

▸ The – sometimes enormous – challenge is how to do the fitting properly & efficiently.

  ◦ There might be many parameters, too many to properly estimate.

  ◦ It might be difficult to estimate parameters in practice, even if in principle we can.

  ◦ Parameters have bounds.

  ◦ Data is censored.

 ◦ Data is on different scales.

  ◦ …

Fitting with the computer

- Most often in infectious disease modeling, the models that we fit are nonlinear, dynamical, and somewhat complicated.

- Analytical approaches don't work. We need to do it numerically.

- Packages such as R come with solvers.

- Often, using these solvers can be tricky. A lot of trial and error is involved.

Fitting in practice

- We need to vary parameters until we found the best fit.

- One approach is to **systematically** sample parameters and for each sample, run the simulation, compute **objective function** (=SSR/NLL/whatever you want to minimize), and see which parameter sample gives the best fit.

- Big disadvantage: You need to cover the space of parameter values rather densely, therefore even with smart sampling (e.g. Latin Hypercube Sampling), this approach is only feasible if you fit few parameters (<= approx. 4-5) and/or the ranges for the parameters are rather narrow.

Local algorithms

▸ One fitting approach is to start with one set of parameter values and explore the neighborhood of the objective function, then systematically move towards the parameter values that give the best fit.
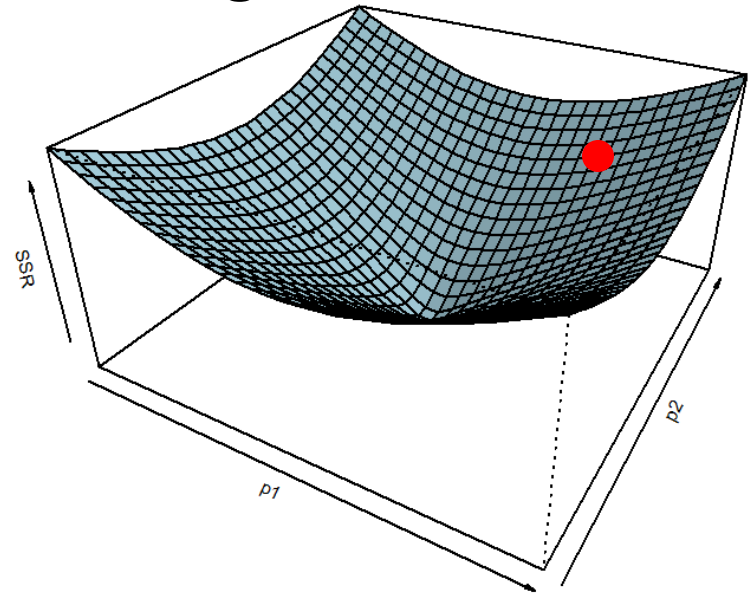
▸ Big disadvantage: If your objective function is "ugly" this approach doesn't work well.

Problems when fitting

▸ Imagine that the least-squares difference between your data and your 2-parameter model looks like this (Rastrigin's Function).

Global minimum at [0 0]

SSR

p_1

p_2
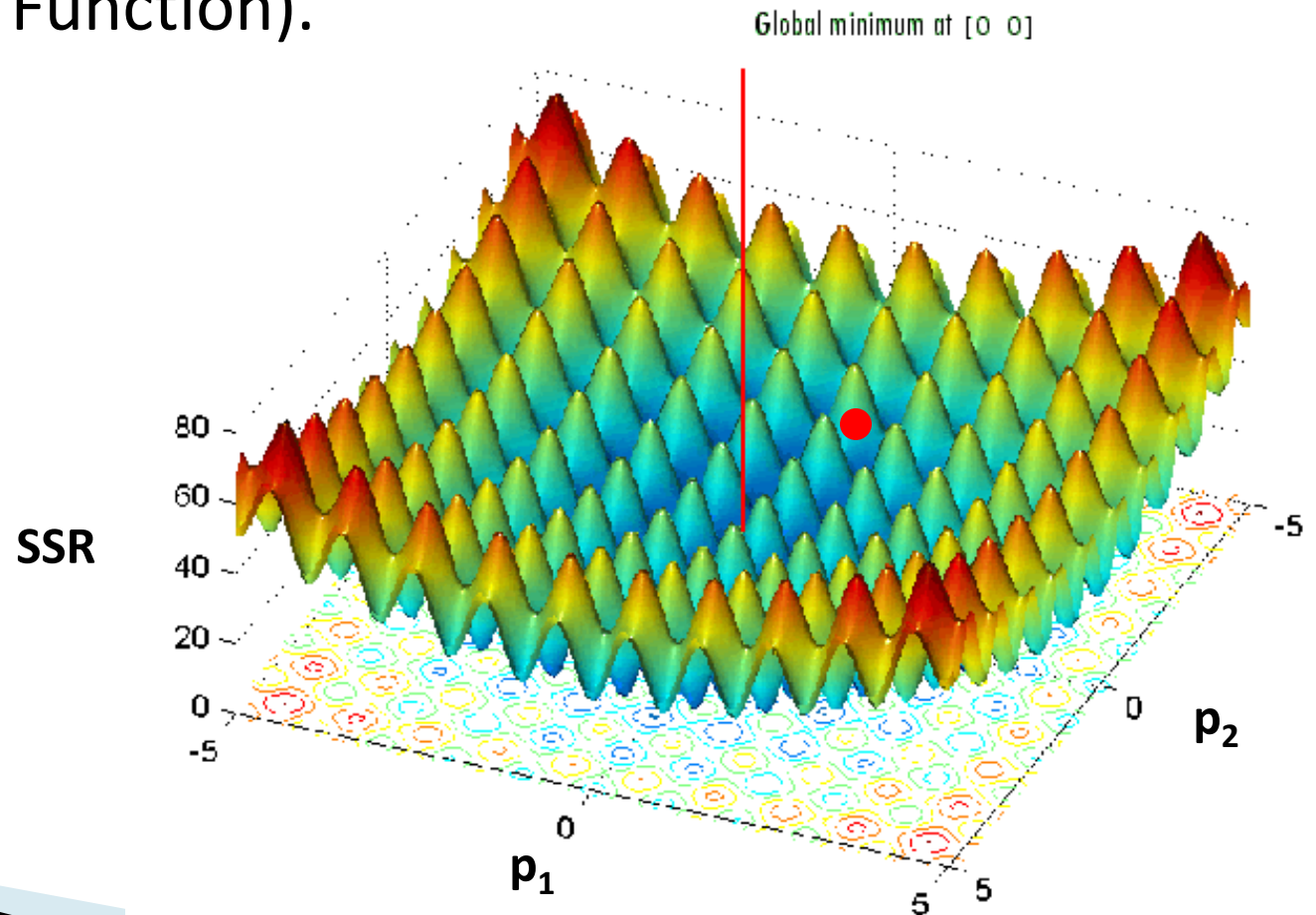
Dealing with local solutions

▸ Try many different initial conditions (choosing initial conditions using LHS might be a good idea). If for every condition the solver returns the same answer, you can be somewhat confident that it found the global best fit.

▸ Start with a simple model with only few free parameters. Then slowly increase model complexity.

▸ Try different numerical solvers/methods. R has different built-in algorithms. A good suite of solvers is in the nloptr package. Others exist in different R packages.

▸ Some solvers/optimizers are called "global", they are supposed to find the overall solution, even if you have a complicated optimization problem. But: even "global" solvers can get stuck in local minima or take too long.

Global minimum at [0 0]
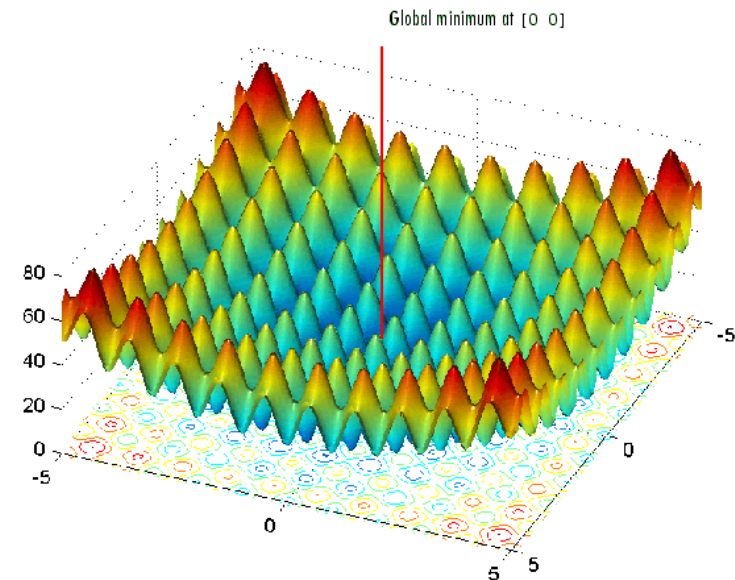
Different fitting algorithms

- Doing the fitting in a robust and efficient way is tricky.

- Many different numerical solvers exist. Packages like R, Matlab, Mathematica, etc. have built-in fitting routines.

- Different routines exist, they differ in how they go about finding the best fits. For a good introduction, see e.g. Bolker (2008) "Ecological Models and Data in R".

- My current favorite collection of R optimizers/fitting routines are the ones found in the nloptr package.

Parameter bounds

▸ Often, certain parameter values make no biological sense

- Almost always, our parameters need to be positive
- Sometimes parameters are between 0 and 1 (e.g. when we model the efficacy of a drug)
- We often know that certain parameters are in a certain range

Dealing with parameter bounds

▸ Bad: Ignore them. Fit the model and hope that the best fit is obtained for biologically sensible parameters (always check!). Problems:

  ◦ During the optimization, the optimizer tries all kind of parameter values, even nonsensical ones. That can make the ode solver "blow up".
  ◦ Without bounds, the range of possible values the solver needs to try is much larger, making the code run much longer.

▸ Meh: Impose bounds using "brute force". Explicitly add bounds inside your code. It's very flexible, but not too "clean".

▸ Good: Use optimizers that can deal with bounds (e.g. those in nloptr).

▸ Good: Transform parameters. Log-transform (log(p)) for positive parameters, Logit-transform (log(p/(1-p)) for parameters between 0 and 1, etc.
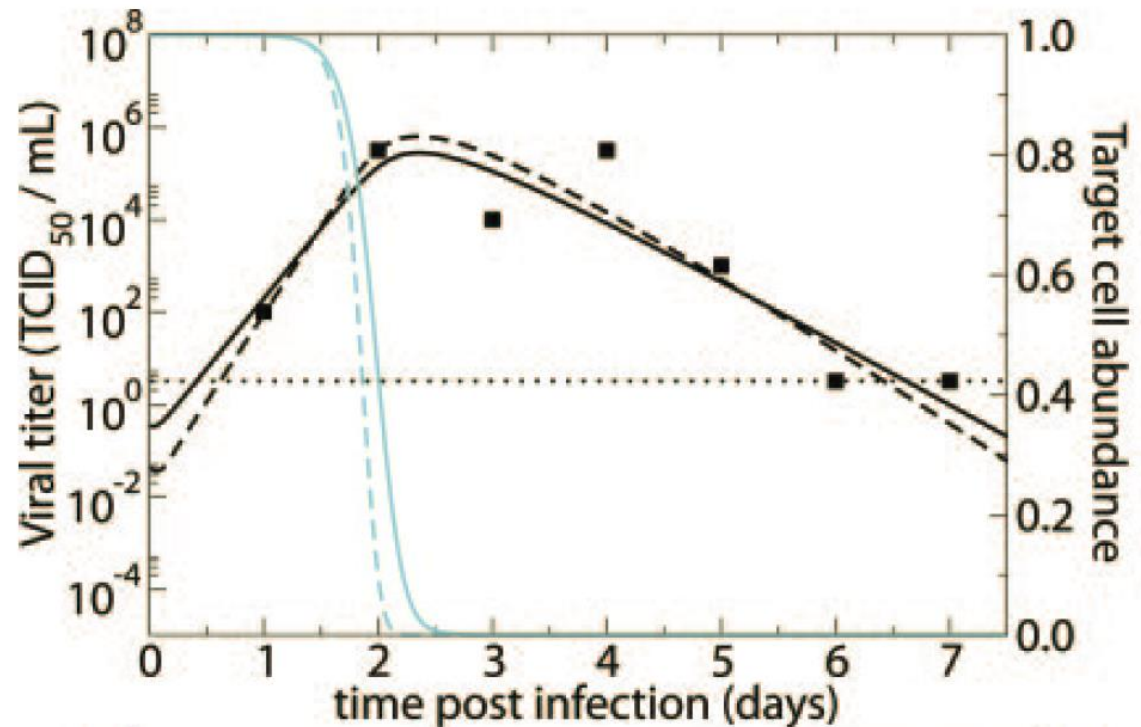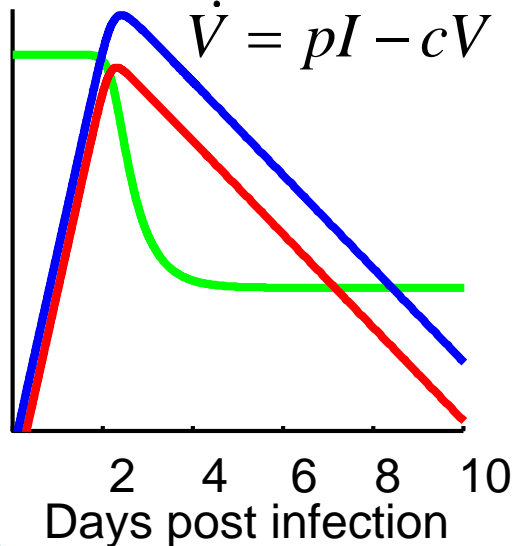
Example: Fitting the basic flu model

▸ Virus dynamics for an uncomplicated influenza infection can be reproduced by a simple model for an acute virus infections without an immune response.

$$\dot{U} = -bUV$$

$$\dot{I} = bUV - \delta I$$

$$\dot{V} = pI - cV$$

Days post infection

Baccam et al. (2006), Journal of Virology

Let's fit the basic virus model to data
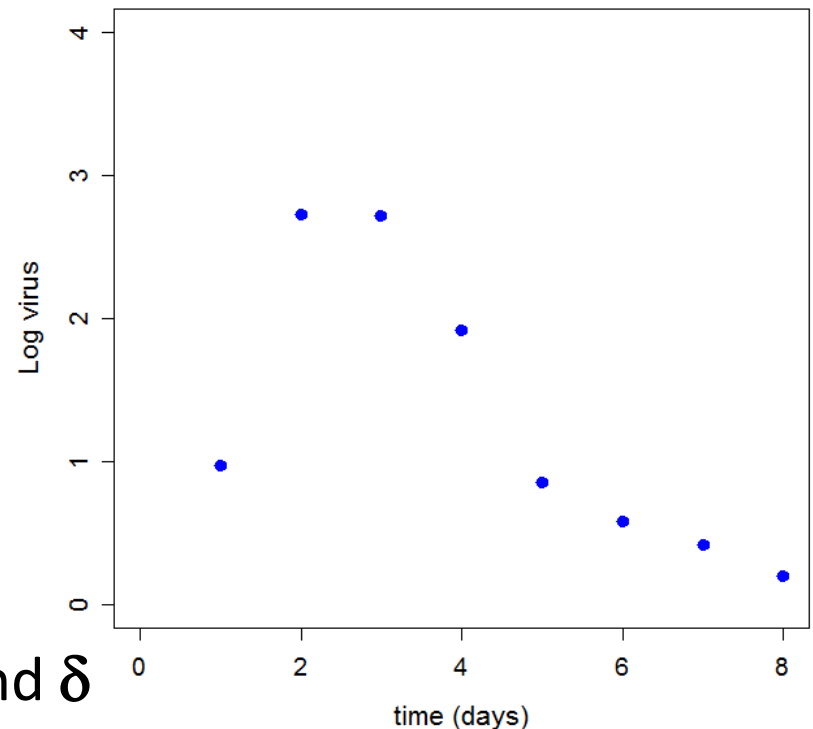
- The data is average virus load during an uncomplicated influenza infection of humans (Hayden et al. 1996, JAMA)

- We fit the simple model to the data

$$\dot{U} = -bUV$$

$$\dot{I} = bUV - \delta I$$

$$\dot{V} = pI - cV$$

- We fit the parameters **b, p, c**

- We assume we know U_0, I_0, V_0 and δ

- We fit in log space. Why?

Data from Hayden et al. (1996), JAMA

Fitting the basic flu model – R example

- Get & open SISMID-U8-basic.r. Install the package **nloptr** if you haven't done so yet.

- Read the code to try and understand what's going on. We fit the model to the data 4 times, with different rules for how to deal with the parameter bounds.

- Note that this program has 2 functions. The main program calls the fit routine/function "fitfunction", this function in turn calls odeequations to integrate the differential equations.

- Run the program, interpret the results.

- Try different initial guess values for the parameters (b0, p0, clear0).

Model non-identifiability

Non-identifiability (unidentifiability)

- It might be that we have more parameters that we can possibly "identify" with the data at hand.

- Strong non-identifiability (full degeneracy):

$$\dot{U} = -bUV$$

$$\dot{I} = bUV - \delta I$$

$$\dot{V} = fpI - cV$$

- Weak non-identifiability (partial degeneracy):

$$\dot{U} = -bUV$$

$$\dot{I} = bUV - \delta I$$

$$\dot{V} = pI - cV$$

Non-identifiability

- The problem of (non)-identifiability becomes worse for larger models with more parameters (over-parameterization, over-fitting).

- Big models need lots of data to allow meaningful fitting!

- To prevent over-parameterization, one can:
  - simplify models
  - re-scale to get rid of some parameters
  - fix some parameters based on prior biological knowledge
  - try to obtain more/better data

Testing for identifiability

▸ Optional: Do a mathematical analysis to test for structural/full/strong/theoretical identifiability (see e.g. Miao et al. (2011) SIAM Review "On Identifiability of Nonlinear ODE Models and Applications in Viral Dynamics")

▸ Recommended (Required): Do test-runs with artificial data.

　◦ For fixed parameter values, use your model to simulate data. E.g. run deterministic model, add normal noise on top.
　◦ Fit the model to the artificial data. You should get the best fit parameter values close to those that you actually used. If not, you have a problem.
　◦ Try this procedure for several parameter combinations.

▸ Doing test simulations can probe for both strong/structural and weak/practical identifiability.

▸ Only once you have tested/confirmed that you can in principle fit the model does it make sense to use it on real data!

Degeneracy/Identifiability – R example

- Load SISMID-U8-testing.r

- The model fits the parameters **b**, **p** and **c**.

- The model is fit to the artificial data (with or without noise).

- Change the amount of noise added to the data, also change initial conditions and optimizer settings.

- What do you conclude about overfitting, is it a problem here?

$$\dot{U} = -bUV$$

$$\dot{I} = bUV - \delta I$$

$$\dot{V} = fpI - cV$$

Measures of Uncertainty

Confidence Intervals (CI)

▸ The fitting routine only returns one value for each parameter as the best fit value, i.e. a point estimate.

▸ We don't know what level of uncertainty is attached to each parameter value. E.g. is the average lifespan of an infected cell 2 days $\pm$ 1h or 2 days $\pm$ 1 day?

$$\dot{U} = -bUV$$

$$\dot{I} = bUV - \delta I$$

$$\dot{V} = pI - cV$$

CI - Bootstrapping

▸ We can obtain a measure of uncertainty by getting confidence intervals.

▸ Remember how a (95%) confidence interval is defined: If we repeated an experiment many times, in 95% of the experiments the CI constructed for each of the experiments would include our estimated parameter.

▸ We rarely repeat an experiment 100 times. But we can "simulate" repeated experiments by using the data we have.

▸ Using this approach to get CI is called "bootstrapping".

▸ Basic approach: resample data **with** replacement, perform fit for each data sample.

▸ Each fit will give best-fit values for parameters. Use distribution of best-fit parameter values to estimate confidence intervals.

Bootstrapping - Example

Original data, N=4

2x

2x

2x

2x

4x
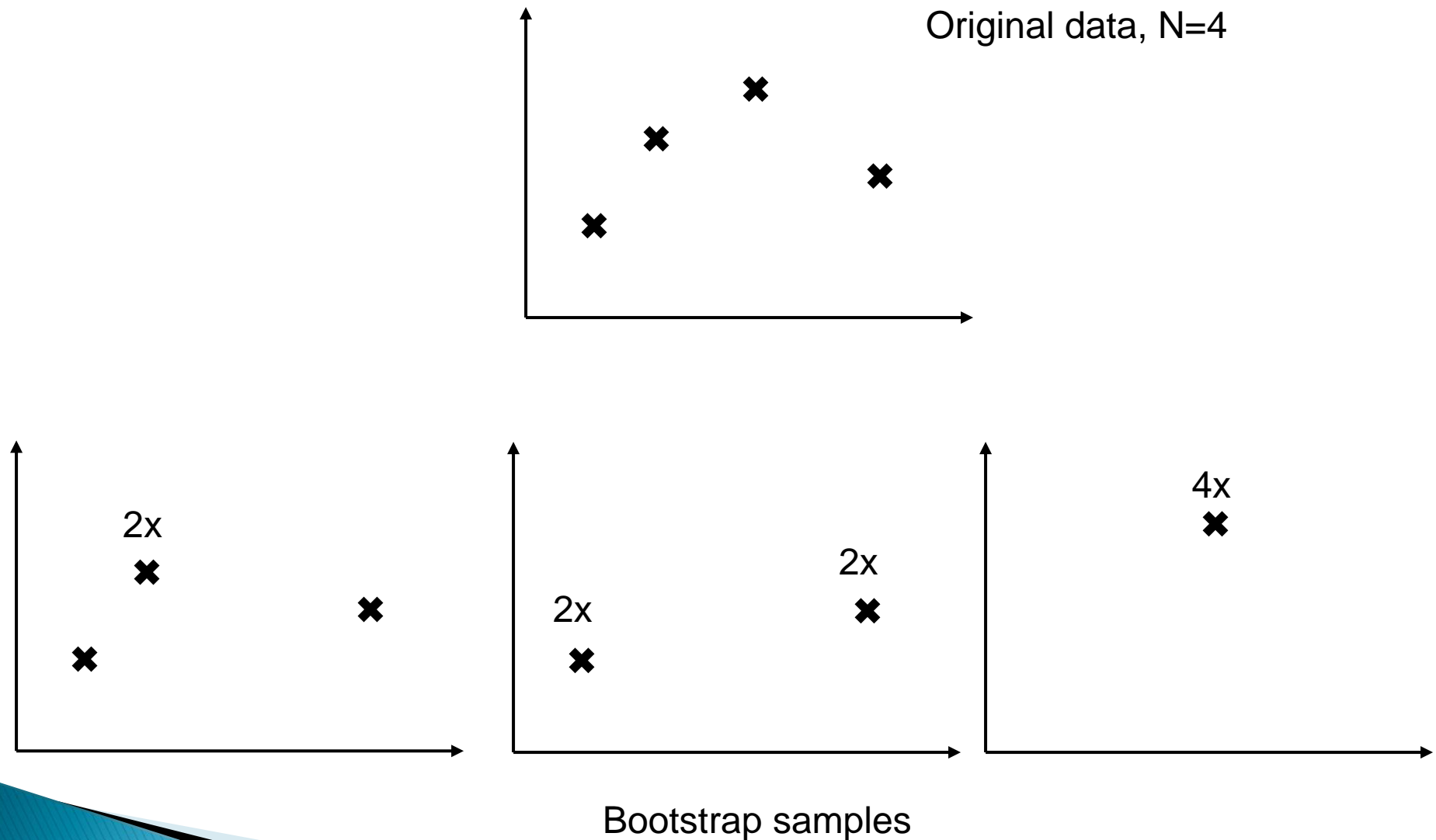
Bootstrap samples

CI - Bootstrapping

▸ For each data/bootstrap sample, you will get a best fit and associated best-fit parameter values.

▸ Use the distribution of best-fit parameter values to estimate confidence intervals.

▸ Example: Sample data 1000 times, sort the 1000 best-fit values for your cell-lifespan parameter. Entry 25 and 975 will form your 95% CI bounds.

▸ R has an easy-to-use routine/package called **boot** that does bootstrapping and can also compute CI in different ways.

Bootstrapping – R Example

▸ Let's try bootstrapping on the within-host flu data.

▸ Open SISMID-U8-boot.r, also install package **boot**.

▸ Try to understand as much as possible of the code (there is lots going on).

▸ While you are testing your model, set the number of bootstraps to a low value (e.g. Rmax=5).

▸ Once your model works, run it with a higher Rmax (e.g. 50). Be patient, the bootstrap takes a while.

▸ Make sure you understand the results.

▸ If your computer is fast enough: increase bootstrap samples (to 200 or more), rerun.

▸ Type ?boot.ci to learn about some of the methods to obtain CI.

Model Selection & Hypothesis Testing

Hypothesis testing

- If we have only one model, we can test if this specific model describes the data well. If it does, we can use it to estimate parameters, e.g. duration of infectiousness, fraction of asymptomatic, etc.

- Note: One could interpret each specific choice of parameter values as its own model, especially if a parameter is zero.

- Often, we want to test multiple specific mechanisms/hypotheses. To do so, we need to build multiple models and see which model best describes the data.

Comparing different models

▸ If we are certain we have a good model, we can use it to obtain parameter estimates.

▸ Often, we don't know which model is good/best.

▸ We can design different models and fit them to data, than compare how well different models fit

  ◦ Example: HCV & IFN model

$$\dot{U} = \lambda - dU - (1-f)bUV$$

$$\dot{I} = (1-f)bUV - \delta I$$

IFN might reduce susceptibility of cells to infection

$$\dot{V} = (1-e)pI - cV$$

IFN might reduce production of virions

Based on Neumann et al. (1998) Science

Comparing different models

- Check "by eye". If it looks wrong/bad, it probably is.

- Are there systematic deviations between model and data?

- Do the best fit parameter values make biological sense? **A model that is statistically the best but biologically unreasonable is pointless!**

- Statistical measures.

Statistical model selection tools

- "Classical" approaches, e.g. F-test, Likelihood ratio.

  - Commonly used.
  - Limited. Cannot deal with non-nested models, awkward for multiple model comparison.

- Information Theoretical Measures, for instance Akaike's "An Information Criterion" (AIC).

  - More general than classical approaches.
  - Easy to compute/use.
  - Theory/fundamentals are complicated.

- Cross-validation, train/test approach.

  - Closest to what we really care about (performance of model on independent data).
  - Very flexible.
  - Needs a decent amount of data to be able to withhold some for testing purposes.
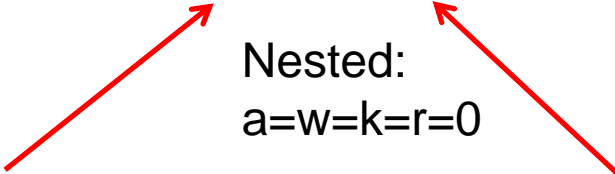
Nested and non-nested models

$$\dot{U} = bUV$$

$$\dot{I} = bUV - \delta I$$

$$\dot{V} = pI - cV$$

Nested:
a=w=k=r=0

$$\dot{U} = bUV$$

$$\dot{I} = bUV - \delta I - kXI$$

$$\dot{V} = pI - cV$$

$$\dot{X} = aV + rX$$

Non-nested

$$\dot{U} = bUV$$

$$\dot{I} = bUV - \delta I$$

$$\dot{V} = pI - cV - kXV$$

$$\dot{X} = aVX - wX$$

Akaike's Information Criterion (AIC)

▶ Theory behind it is rather mathematical, but computation/application is easy.

▶ For least-squares fitting, AIC is given by:

$$AIC = N \ln(\frac{SSR}{N}) + 2K$$

▶ N=number of data points, SSR=sum of square residuals, K number of parameters that are fit (+1 for the implicit fitting of the standard deviation in the least squares approach).

▶ Lower AIC is better.

▶ SSR usually decreases for more parameters, so 1st term becomes smaller. 2nd term increases: AIC is a trade-off between good fit (low SSR) and parsimony (simple model, low K).

▶ For more details:

   ◦ Anderson 2008, Springer "Model Based Inference in the Life Sciences"
   ◦ Burnham and Anderson (2003) "Model Selection and Multi-Model Inference"

AICc

- AIC doesn't work well for small sample size

- Corrected AIC (AICc) does better:

$$AICc = AIC + \frac{2K(K+1)}{N-K-1}$$

- K = number of fitted parameters (for likelihood approach), K = number of explicitly fitted parameters (+1 for implicit fitting of SD in least squares approach).

AIC & Quality of Model

- Absolute value of AIC is meaningless, only differences in AIC between models matter.

- For two models with AIC1>AIC2, define Δ=AIC1-AIC2. Then you can apply the following guidelines:
  - For Δ<2, models are essentially equally good.
  - For Δ>10, there is strong support for the model with smaller AIC.
  - In between is "gray area".
  - These are only "rules of thumb" there is – fortunately! – nothing like a 0.05 p-value.

- Since AIC is based on information theory, one should avoid using the terms "significant" and "reject", which have clearly defined meanings in classical statistics.

AIC & relatives

▸ AIC works for both nested and non-nested models.

▸ There is no arbitrary cutoff (e.g. p=0.05).

▸ It is easy to compare between more than 2 models.

▸ I recommend AIC over "classical" approaches (F-test/likelihood ratio). (Train/test even better if you got enough data).

▸ There are other quantities similar to AIC:

◦ Bayes Information Criterion (BIC)
◦ Deviance Information Criterion (DIC)
◦ Others (TIC, HQIC, QAIC)...

▸ AIC is widely used, but BIC and DIC are catching up.

▸ The field is very dynamic and under rapid development – lots of issues are not yet settled. (E.g. what to do if AIC tells you model 1 is better than model 2 but BIC tells you the models are similarly good?)

Model selection – R example

- Open SISMID-U8-aic.r. The code fits two different models to the virus load data:

$$\dot{U} = bUV$$

$$\dot{I} = bUV - \delta I - kXI$$

$$\dot{V} = pI - cV$$

$$\dot{X} = aV + rX$$

$$\dot{U} = bUV$$

$$\dot{I} = bUV - \delta I$$

$$\dot{V} = pI - cV - kXV$$

$$\dot{X} = aVX - wX$$

- For model 1, we fit **a** and **r**. For model 2, we fit **a**, **w** and **k**.

- Also try different initial conditions.

- Which model is statistically better supported by the data?

Model selection - summary

- AIC or similar can suggest which model is "best".

- Always be careful with interpretation:
  - If the model with lowest AIC still doesn't fit the data well, or produces biologically unrealistic parameter values, you should interpret this as "all the models you tested are poor"
  - If several models fit the data well, maybe the best one can say is that they are all reasonable and one would need more data to further discriminate.

- For your best model, repeat all the steps you – hopefully – did before you did the stats:
  - Check "by eye". If it looks wrong/bad, it probably is.
  - Are there systematic deviations between model and data?
  - Do the best fit parameter values make biological sense? **A model that is statistically the best but biologically unreasonable is pointless!**
  - Use the model to simulate data. Then try to fit the model to the simulated data. If the model can't fit that, something went wrong.

Fitting – Summary

- Fitting data to models is useful for testing hypotheses/mechanisms and obtaining parameter estimates.

- Fitting is can be tricky due to numerical issues, parameter bounds, censored data, correlations, multiple variable measurements…

- Especially for fitting, it is better to start with simple models and slowly include more complexity. The built-up of models should be driven by availability of data and the questions asked. The quality of the data provides the main limitation. The better/richer the data, the more complex the model may be without risk of overfitting.

Some further comments

- Always perform reality checks!
  - Do the variables and units match (e.g. RNA copies versus PFU)
  - Are there biological reasons to fix or bound some parameters?
  - What does it mean when the best fit occurs for a biologically unreasonable parameter but an almost equally good fit can be found for biologically reasonable parameter values?
- It's not unlikely that you come away "empty-handed", i.e. you have to conclude that several alternative models fit the data equally well.

Some further comments

- Data fitting is a huge area, one could spend a whole course just on that topic.

- We focused on least squares fitting. Other fitting approaches are possible, for instance maximum likelihood. ML is equivalent to least square fitting if errors are normally distributed.

- Bayesian approaches are increasing in popularity. They are usually technically more difficult and take potentially very long to run. R and/or BUGS are commonly used.

Fitting – more topics

▸ Some further topics and references:

◦ Fitting of different datasets over varying conditions (e.g. with and without treatment); heterogeneous data (e.g. fitting virus and infected cells at the same time). Handel et al (2010) JRSI.

◦ Models with process noise (i.e. stochastic models). He et al. (2010) JRSI.

◦ Multilevel models that combine data from "similar but different" sources (e.g. different patients). Canini et al (2011) JVI.

◦ Bayesian approaches to fitting. O'Neill 2010 Stat Mech

Further reading - fitting

- A nice introduction that is not too technical and freely available online: "Fitting Models to Biological Data using Linear and Nonlinear Regression" at http://www.graphpad.com/manuals/prism4/RegressionBook.pdf

  - This is the manual to the fitting software Prism but most of the book is a general and very readable introduction to fitting. The graphpad.com webpage contains additional resources.
  - The explanation of AIC weights as probabilities is wrong.

- Clark (2007) "Models for Ecological Data": Comprehensive, discusses static and dynamic models, a mix of classic and Bayesian methods

- Bolker (2008) "Ecological Models and Data in R": This book focuses mostly on static models, but it is **very** good in describing concepts, pitfalls, etc.

- *Hilborn and Mangel (1997) "The Ecological Detective: Confronting models with data": L*ess "hands-on" compard to Clark and Bolker, but has nice explanations of data fitting, model comparison, etc.

- A good (but technical) book for information theoretic approaches to model comparison is *Burnham and Anderson (2003) "Model Selection and Multi-Model Inference"*

Further reading – fitting flu data

- *Baccam et al. (2006) J Virology*

- *Handel et al. (2007) PLoS Computational Biology*

- *Beauchemin et al. (2008) J Theoretical Biology*

- *Saenz et al. (2010) J Virology*

- *Handel et al. (2010) J Royal Society Interface*

- *Beauchemin and Handel (2011) BMC Public Health*