Project 7: Bacterial growth and parameter fitting

MCSB Bootcamp — Mathematical and Computational Track

Jun Allard

Model simulation

Given the parameters of a model, we can use ODE solvers to predict the behavior of a model, $x(t)$.

Given data, can we find which model parameters are the best fit?

a. Write code to simulate the following model

$$\frac{dN}{dt} = \lambda N(1 - (N/\theta)^{\alpha}) \tag{1}$$

where the user (you) specifies the parameter values $\lambda = 1$, $\theta = 10^3$, $\alpha = 2$, and the initial condition $N(0) = 200$. Plot a time series of $N_{\text{sim}}(t)$.

```
# Your code here.
```

Data analysis

b. Write code that loads bacterial growth experimental data into two arrays: one for the amount of bacteria $N_{\text{exp}}(t_i)$ and the other for the measurement time points t_i. Plot the time series $N_{\text{exp}}(t_i)$.

```
# Your code here.
```

c. Plot the solution $N(t)$ and the experimental data $N_{\text{exp}}(t)$ on the same axes. Set $N(0)$ to match the experimental data exactly. Manually explore different parameter values of λ, θ, α to see which values best fit the data.

```
# Your code here.
```

Parameter learning

d. Write code that computes the sum of squared error (SSE), which is

$$\text{SSE} = \sum_i \left(N_{\text{sim}}(t_i) - N_{\text{exp}}(t_i)\right)^2$$

Hint: `scipy.integrate.solve_ivp` takes a `t_eval` argument, an array of times. Given one, it returns the solution at those times and no others. This is useful if you only need the solution at the measurement times t_i.

```
# Your code here.
```

e. Using the code from the previous part, create a *function* that takes parameters and returns the SSE. Hint: Define it with `def` in a cell of its own, so that the cells after it can call it.

```
# Your code here.
```

f. Use an automated minimization algorithm like `scipy.optimize.minimize` to find the parameters that minimize the sum of squared error. These parameters are the "best fit" or "maximum likelihood" parameters $\hat{\lambda}, \hat{\theta}, \hat{\alpha}$. Plot, on the same axes, the experimental and best-fit model time series, $N_{\text{exp}}(t)$ and $N_{\text{sim}}(t)$.

```
# Your code here.
```

Model learning

g. Now repeat the above, but use the model with $\alpha = 1$. Did the fit improve, or worsen?

```
# Your code here.
```

h. Now repeat the above, but with

$$\frac{dN}{dt} = \lambda N.$$

Note this is numerically the same as Equation 1 with $\theta \gg N$. Did the fit improve, or worsen?

```
# Your code here.
```