MATHEMATICAL AND COMPUTER MODELING OF NONLINEAR BIOSYSTEMS I

COMPUTER LABORATORY V: Interactions between populations (competition/mutualism), May model

Ph. D. Programme 2013/2014

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Project co-financed by European Union within the framework of European Social Fund

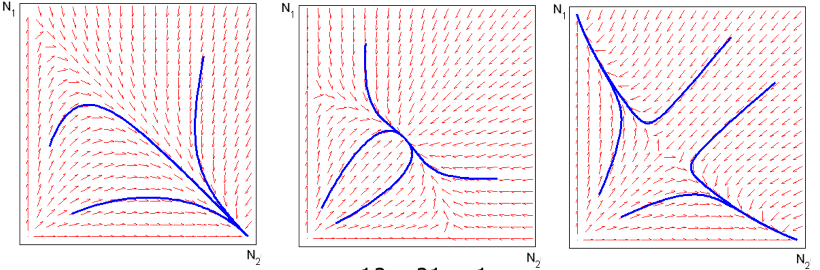
Comptetition between two species

The classic two species competition model reads

$$\begin{aligned}\dot{N}_1 &= r_1 N_1 \left(1 - \frac{N_1}{K_1} - a_{12}\frac{N_2}{K_2}\right),\\ \dot{N}_2 &= r_2 N_2 \left(1 - \frac{N_2}{K_2} - a_{21}\frac{N_1}{K_1}\right),\end{aligned}$$

where all parameters are positive.

Three scenarios are possible depending on the parameters values:



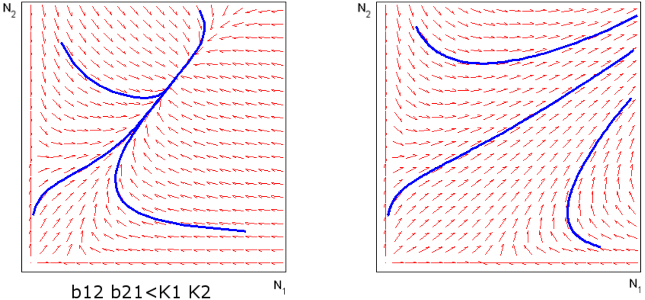
a12, a21 < 1

The classic two species mutualism model reads

$$
\begin{aligned}
\dot{N}_1 &= r_1 N_1 \left(1 - \frac{N_1}{K_1} + b_{12} N_2\right), \\
\dot{N}_2 &= r_2 N_2 \left(1 - \frac{N_2}{K_2} + b_{21} N_1\right),
\end{aligned}
$$

where all parameters are positive.

Two scenarios are possible depending on the parameters values:



b12 b21<K1 K2

Case study

Suppose that we performed an ecological experiment and introduced 10 individuals from species A and 40 individuals from species B into a fixed and previously empty environment.

We counted carefully number of individuals after 0, 7, 14, 28, 42, 56, 70, 84, 98 days.

On the basis of observations we want to answer the following question:
Question: Do we observe competition or cooperation between those two populations?

Possible way of answering: fit both considered models to data.
That is exactly what we will do today.

Step 1 - read data to MATLAB

1. Download a csv file containing experimental measurements from:

   `http://www.mimuw.edu.pl/~poleszczuk/Jan_Poleszczuk/Courses/Entries/ 2014/3/4_Mathematical_and_computer_modeling_of_nonlinear_biosystems_I_files/data.csv`

   Part of file content:

Day	Species 1	Species 2
0	10	40
7	57	147

2. Implement MATLAB function which reads data from downloaded file
   - input argument: file name;
   - output variable: structure with fields "t" and "measurements" (moments of measurements and measurements values, respectively).

Hint: use csvread function

Step 1 - solution

```
function data = readData( filename )

    rawData = csvread(filename,2);
    data.t = rawData(:,1)';
    data.measurements = rawData(:,2:3)';

end
```

Step 2 - solution to the competition model

1. Implement MATLAB function which for given parameters and initial conditions returns solution to the competition model:
2. function should take three arguments:
   - vector of initial conditions,
   - mesh of points at which the solution has to be calculated,
   - structure with parameters values;
3. function should return solution as a matrix.

Step 2 - solution

```
function sol = solveCompetition( init, T, par )

    sol = ode45(@model,[0 T(end)],init);
    sol = deval(sol, T);

    function y=model(~,x)
       y=zeros(2,1);
       y(1)=par.r1*x(1)*(1-x(1)/par.K1-par.a12*x(2)/par.K2);
       y(2)=par.r2*x(2)*(1-x(2)/par.K2-par.a21*x(1)/par.K1);
    end
end
```

Step 3 - solution to the cooperation model

1. Implement MATLAB function which for a given parameters and initial conditions returns solution to the cooperation model
2. function should take three arguments:
    - vector of initial conditions,
    - mesh of points at which the solution has to be calculated,
    - structure with parameters values;
3. function should return solution as a matrix.

Step 3 - solution

```
function sol = solveMutualism( init, T, par )

    sol = ode45(@model,[0 T(end)],init);
    sol = deval(sol, T);

    function y=model(~,x)
       y=zeros(2,1);
       y(1)=par.r1*x(1)*(1-x(1)/par.K1+par.b12*x(2));
       y(2)=par.r2*x(2)*(1-x(2)/par.K2+par.b21*x(1));
    end
end
```

Step 4 - auxiliary functions

1. Implement a MATLAB function which for a given structure returns a vector containing values of all its fields (we assume that each field has a scalar value).

1. Implement a MATLAB function which for given vector and structure returns a structure with the same fields as the input structure, but with values taken from the input vector (we assume that the vector has the same length as the number of field in the structure).

Step 4 - solution

```
function x = initialize(par)
    f = fields(par);
    x = zeros(size(f));
    for i=1:length(f)
        x(i) = initParam.(f{i});
    end
end

function par = update(x, par)
    f = fields(par);
    for i=1:length(f)
        par.(f{i}) = x(i);
    end
end
```

We could also use struct2cell and cell2struct functions.

Step 5 - fitting function

1. Implement a MATLAB function which for given model, initial parameters and experimental data finds the best fitting model curve.
2. In order to find the best fitting model curve we minimize the sum of squared differences between data and model solution.
3. Function should take three arguments
   - model as a function handle,
   - initial parameters values as a structure,
   - structure with data (taken from step 1).
4. Function should return two variables
   - parameters for which the model solution has the lowest approximation error,
   - approximation error,

Hint: Use fminsearch function.

Step 5 - solution

```
function [par, err] = fitModel( model, initParam, data)

    x0 = initialize(initParam);

    [x, err]=fminsearch(@F,x0);
    par = update(x, initParam);

    function y = F(x)
        parN = update(x,initParam);
        y=(feval(model,data.measurements(:,1),data.t,parN)-...
          data.measurements)./data.measurements;
        y=sum(sum(y.^2));
    end
end
```

Step 6 - initial parameters values

1. Implement a MATLAB function which returns the initial parameters values (initial guesses) for both considered models.
2. Parameter should be estimated from experimental data if possible.
3. Function should take the structure with experimental data as an input (same as in Step 1).
4. Function should return two data structures which are consistent with models implementation (Step 2 and 3).

Step 6 - solution

```
function [initParC, initParM] = initialParams(data)
initPar.K1 = max(data.measurements(1,:));
initPar.K2 = max(data.measurements(2,:));

initPar.r1 = (data.measurements(1,2)-data.measurements(1,1))...
/data.t(2)/data.measurements(1);
initPar.r2 = (data.measurements(2,2)-data.measurements(2,1))...
/data.t(2)/data.measurements(2);

initParC=initPar;
initParC.a12 = 0.5; %stability
initParC.a21 = 0.5; %stability

initParM=initPar;
initParM.b12 = 0.05/(initPar.K1);
initParM.b21 = 0.05/(initPar.K2);
end
```

Final step

1. Write a MATLAB script in which both models are fitted to the experimental data.
2. Use functions from previous steps.
3. Compare minimal approximation errors obtained with both models.
4. Make two plots (separate for each model) showing experimental data and best approximating model curves. Plotting should be done using separate function.

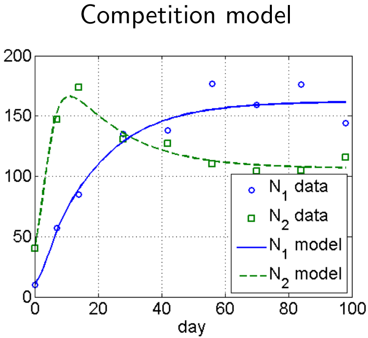
Final step - solution (part 1)

```
clear all;

data = readData('data.csv');

[initParC, initParM] = initialParams(data);

[parC, errC] = fitModel(@solveCompetition, initParC, data);
[parM, errM] = fitModel(@solveMutualism, initParM, data);

[errC, errM]

T = linspace(data.t(1),data.t(end),100);
solC=solveCompetition( data.measurements(:,1), T, parC );
solM=solveMutualism( data.measurements(:,1), T, parM );

plotFit(T,solC,data,1)
plotFit(T,solM,data,2)
```

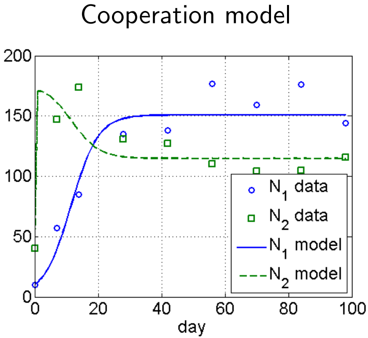
Final step - solution (part 2)

```
function plotFit(T,sol,data,f)
set(0,'DefaultAxesFontSize',18)
figure(f)
clf
hold on
pl1=plot(data.t, data.measurements,'LineStyle','none',...
'LineWidth',2);
set(pl1(1),'Marker','o')
set(pl1(2),'Marker','s')
pl2=plot(T,sol,'LineWidth',2);
set(pl2(2),'LineStyle','--')
hold off
legend({'N_1 data','N_2 data','N_1 model','N_2 model'},...
'Location','SouthEast')
xlabel('day')
grid on
box on
end
```

Final outcome



Error: 0.0678



Error: 0.2775

It seems that the populations compete with each other.
In addition b_{21} after fitting became negative...

May model

We could also consider the following predator-prey model

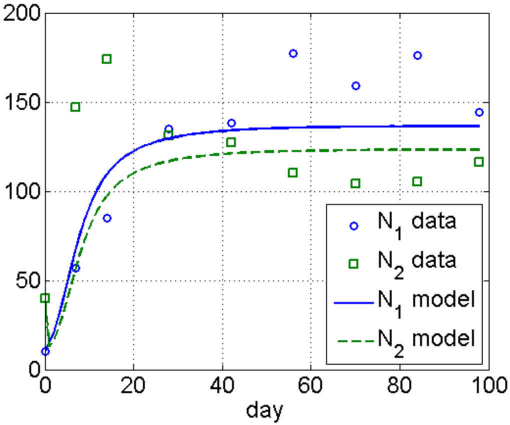
$$\begin{aligned}\dot{N}_1 &= r_1 N_1 \left(1 - \frac{N_1}{K_1}\right) - a\frac{N_1 N_2}{1+aN_1},\\ \dot{N}_2 &= r_2 N_2 \left(1 - \frac{N_2}{K_2 N_1}\right),\end{aligned}$$

where all parameters are positive.

It could be that the populations that we investigated experimentally are not competing/cooperating, but are in the predator-prey relationship described by the above model.

Exercise: Compare the May model with the experimental data considered in the previous exercise.

Comparizon of the May model with data



Approximation error: 0.9122.

May model gives the biggest approximation error.

IMPORTANT

Fminsearch is a local fitting procedure, and thus the above conclusions might be dependent on the initial parameter guesses.