Experimental data analysis
Lecture 2: Nonlinear regression

Dodo Das

Review of lecture 1

- Likelihood of a model.
- Likelihood maximization + Normal errors = Least squares regression
- Linear regression. Normal equations.

Demo 1: Simple linear regression in MATLAB

```matlab
%% Simple linear regression. Generate some linear data, add normally distrib

%% Specify model parameters
a = 1;
b = 2;

%% Generate some data
x = (1:10)';
yTrue = a + b*x;

%% Add normal error to simulate experimental noise
NoiseStd = 0.5; % The standard deviation of the noise
err = NoiseStd*randn(length(x), 1);
yExpt = yTrue + err;

%% Create design matrix
A = [ones(size(x)) x];

%% Solve normal equation using the MATLAB backslash operator
betaHat = A \ yExpt

%% Plot data and fit
plot(x, yExpt, 'ro', 'MarkerSize', 8)
hold on
xFit = (1:0.1:10)';
yFit = [ones(size(xFit)) xFit]*betaHat;
plot(xFit, yFit, 'b-', 'LineWidth', 2)
xlim([0, 11])
ylim([0, 23])
xlabel('x')
ylabel('y')
grid on
legend('Data', 'Fit', 'Location', 'NW')
```

Demo 1: Simple linear regression in MATLAB

Command Window

New to MATLAB? Watch this Video, see Demos, or read Getting Started.

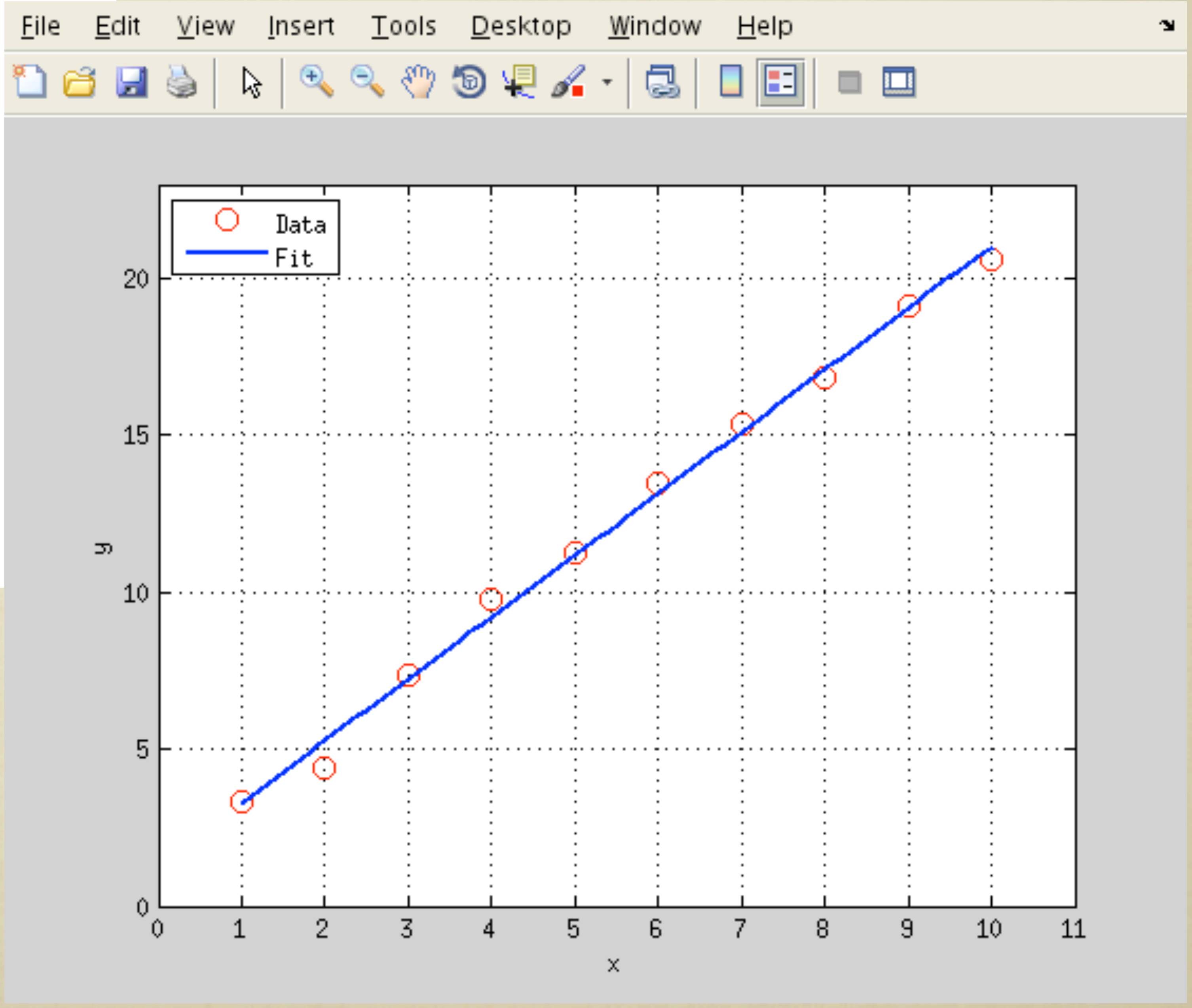
```
>> a = 1;
b = 2;
>> x = (1:10)';
yTrue = a + b*x;
>> NoiseStd = 0.5; % The standard deviation of the noise
err = NoiseStd*randn(length(x), 1);
yExpt = yTrue + err;
>> A = [ones(size(x)) x];
>> betaHat = A \ yExpt

betaHat =

    1.3280
    1.9700

>> plot(x, yExpt, 'ro', 'MarkerSize', 8)
hold on
xFit = (1:0.1:10)';
yFit = [ones(size(xFit)) xFit]*betaHat;
plot(xFit, yFit, 'b-', 'LineWidth', 2)
xlim([0, 11])
ylim([0, 23])
xlabel('x')
ylabel('y')
grid on
legend('Data', 'Fit', 'Location', 'NW')
fx >>
```

Demo 1: Simple linear regression in MATLAB

Command Window

New to MATLAB? Watch this Video, see Demos, or read Getting Started.

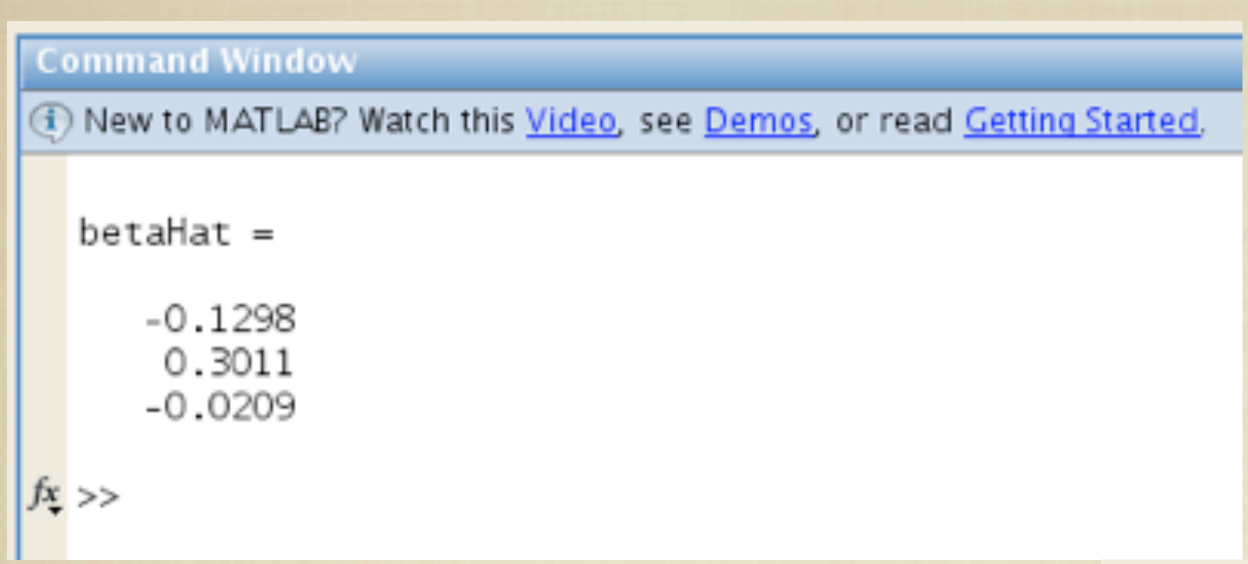
```
>> a = 1;
b = 2;
>> x = (1:10)';
yTrue = a + b*x;
>> NoiseStd = 0.5; % The standard deviation of the noise
err = NoiseStd*randn(length(x), 1);
yExpt = yTrue + err;
>> A = [ones(size(x)) x];
>> betaHat = A \ yExpt

betaHat =

    1.3280
    1.9700

>> plot(x, yExpt, 'ro', 'MarkerSize', 8)
hold on
xFit = (1:0.1:10)';
yFit = [ones(size(xFit)) xFit]*betaHat;
plot(xFit, yFit, 'b-', 'LineWidth', 2)
xlim([0, 11])
ylim([0, 23])
xlabel('x')
ylabel('y')
grid on
legend('Data', 'Fit', 'Location', 'NW')
>>
```

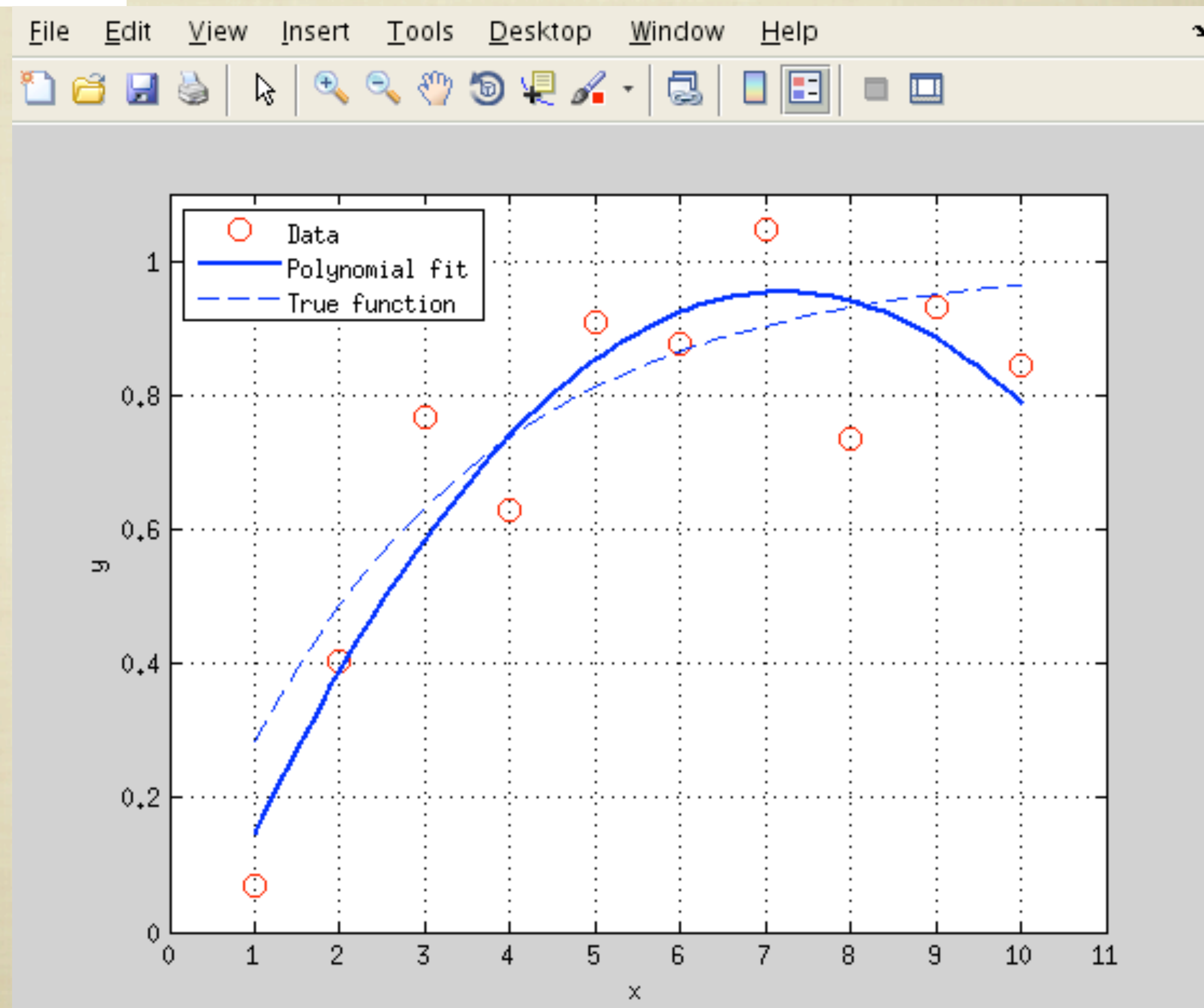


Demo II: Polynomial regression in MATLAB

```matlab
%% Polynomial regression. Generate a some data for a saturable curve model.

%% Specify model parameters
ymax = 1;
tau = 3;

%% Generate some data
t = (1:10)';
yTrue = ymax*(1-exp(-t/tau));

%% Add normal error to simulate experimental noise
NoiseStd = 0.1; % The standard deviation of the noise
err = NoiseStd*randn(length(t), 1);
yExpt = yTrue + err;

%% Fit with a 2nd order polynomial. Create design matrix
A = [ones(size(t)) t t.^2];

%% Solve normal equation using the MATLAB backslash operator
betaHat = A \ yExpt

%% Plot data and fit
plot(t, yExpt, 'ro', 'MarkerSize', 8)
hold on
tFit = (1:0.1:10)';
yFit = [ones(size(tFit)) tFit tFit.^2]*betaHat;
plot(tFit, yFit, 'b-', 'LineWidth', 2)
plot(tFit, ymax*(1-exp(-tFit/tau)), 'b--', 'LineWidth', 1)
xlim([0, 11])
ylim([0, 1.1])
xlabel('x')
ylabel('y')
grid on
legend('Data', 'Polynomial fit', 'True function', 'Location', 'NW')
```

Demo II: Polynomial regression in MATLAB





Nonlinear regression

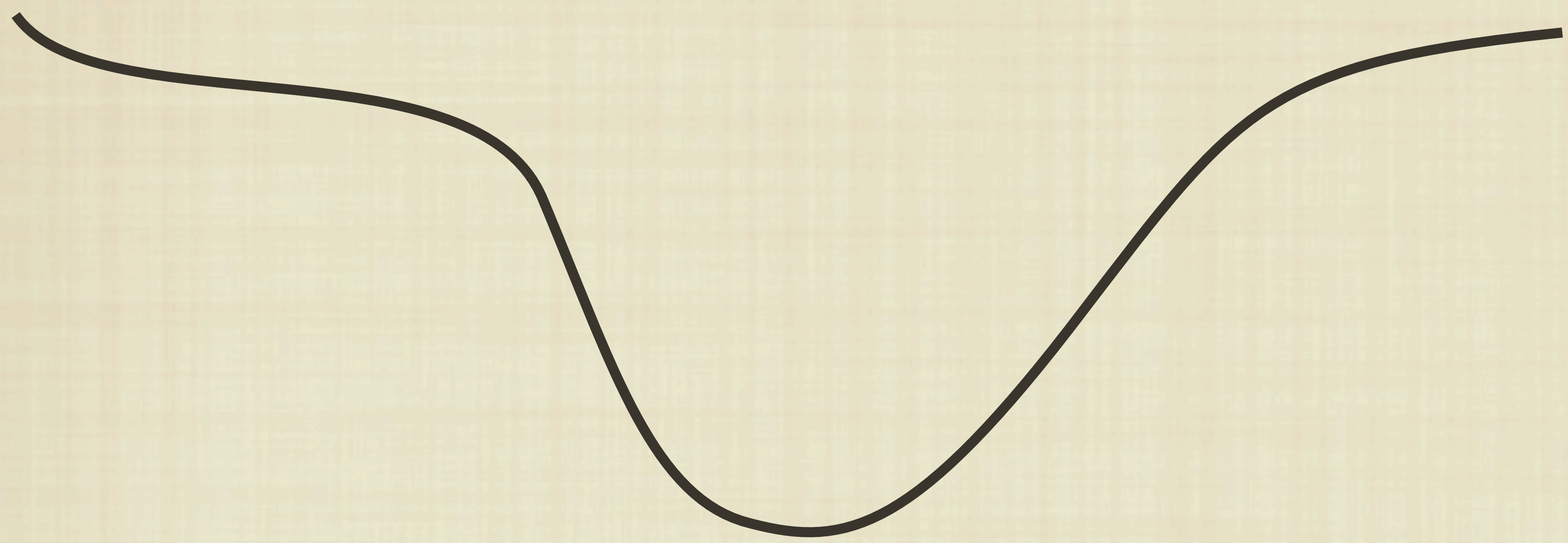
- The model is a nonlinear function of the parameters.
- We can still write down the likelihood as before.
- But the maximum likelihood equations cannot be solved analytically.

Iterative least-squared minimization

- Choose an initial guess for the parameters.
- Evaluate SSR.
- Propose a move in parameter space.
- If move reduces SSR, then update parameter values.
- Otherwise, propose a different move.

How to choose the move in parameter space?

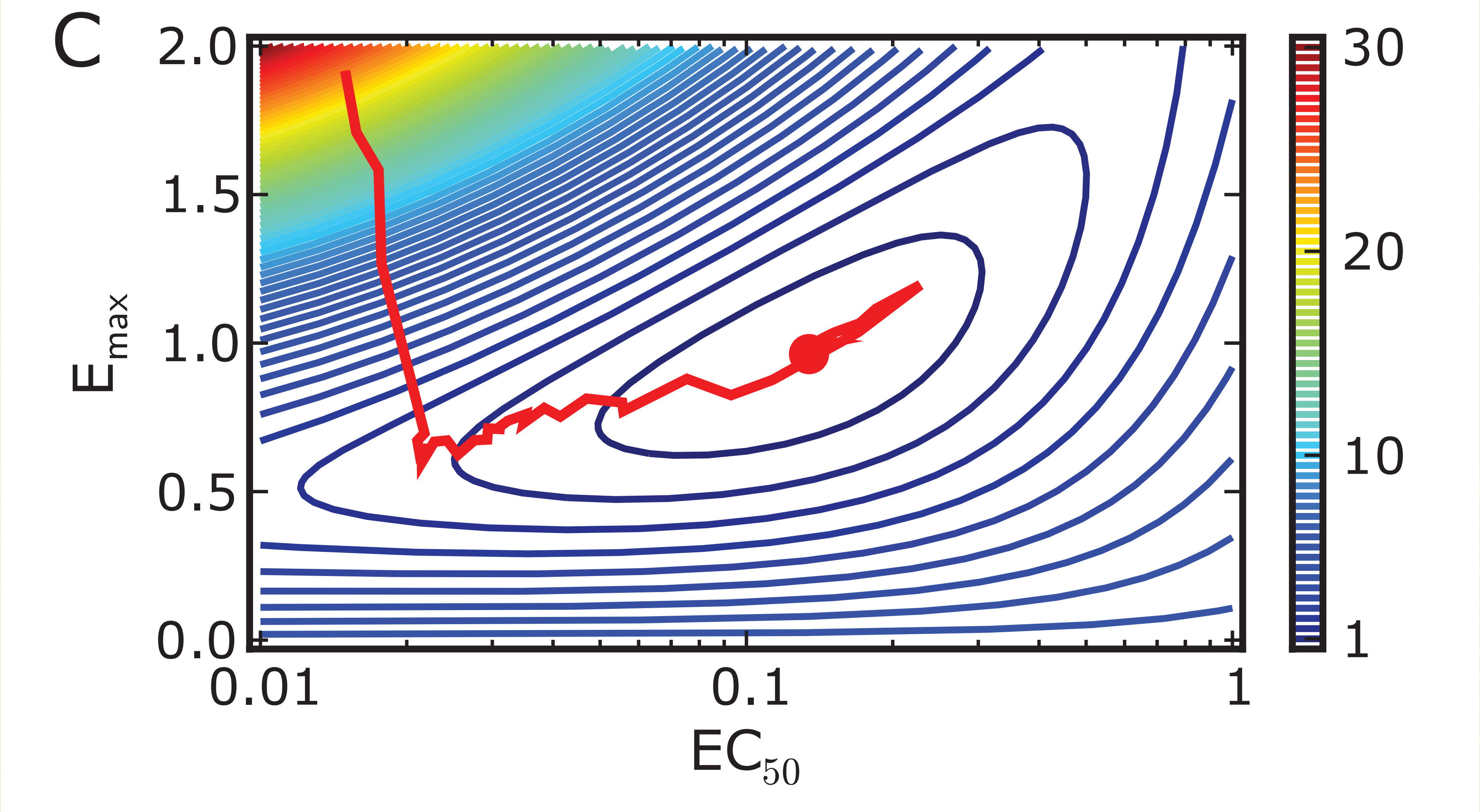
Gradient descent: Far from a minima, it is best to find the gradient (i.e. direction of steepest descent), and move down the gradient of the SSR function.



Gauss-Newton: Near a minima, construct a Taylor-series approximation of the function (to 2nd order) and determine the location of the minima.

A compromise - Levenberg-Marquardt

Switches between Gradient descent when far from minima, and to Gauss-Newton when close to minima.



Practical considerations

- Need to specify initial guess.
- Can be trapped in local minima if initial guess is not good.
- Try a number of random initial guesses, and pick the final result that has the lowest SSR.
- If computationally feasible, good to plot the SSR landscape over some reasonable parameter range.

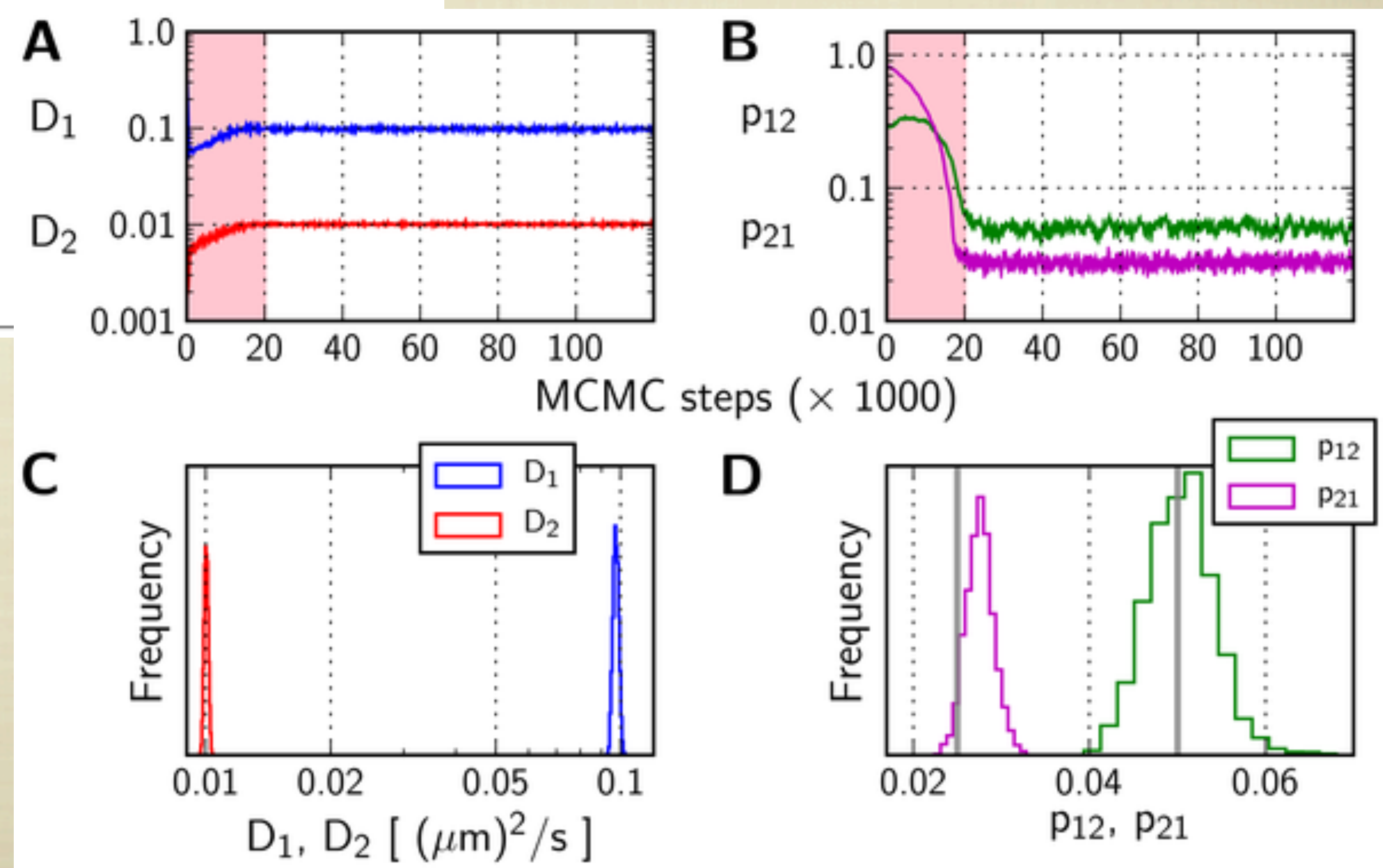
Other likelihood maximization schemes

- Based on stochastic simulations:
  - Markov chain Monte Carlo (MCMC)
  - Simulated annealing
- Also, many other optimization techniques [Major branch of applied math].

An example of MCMC

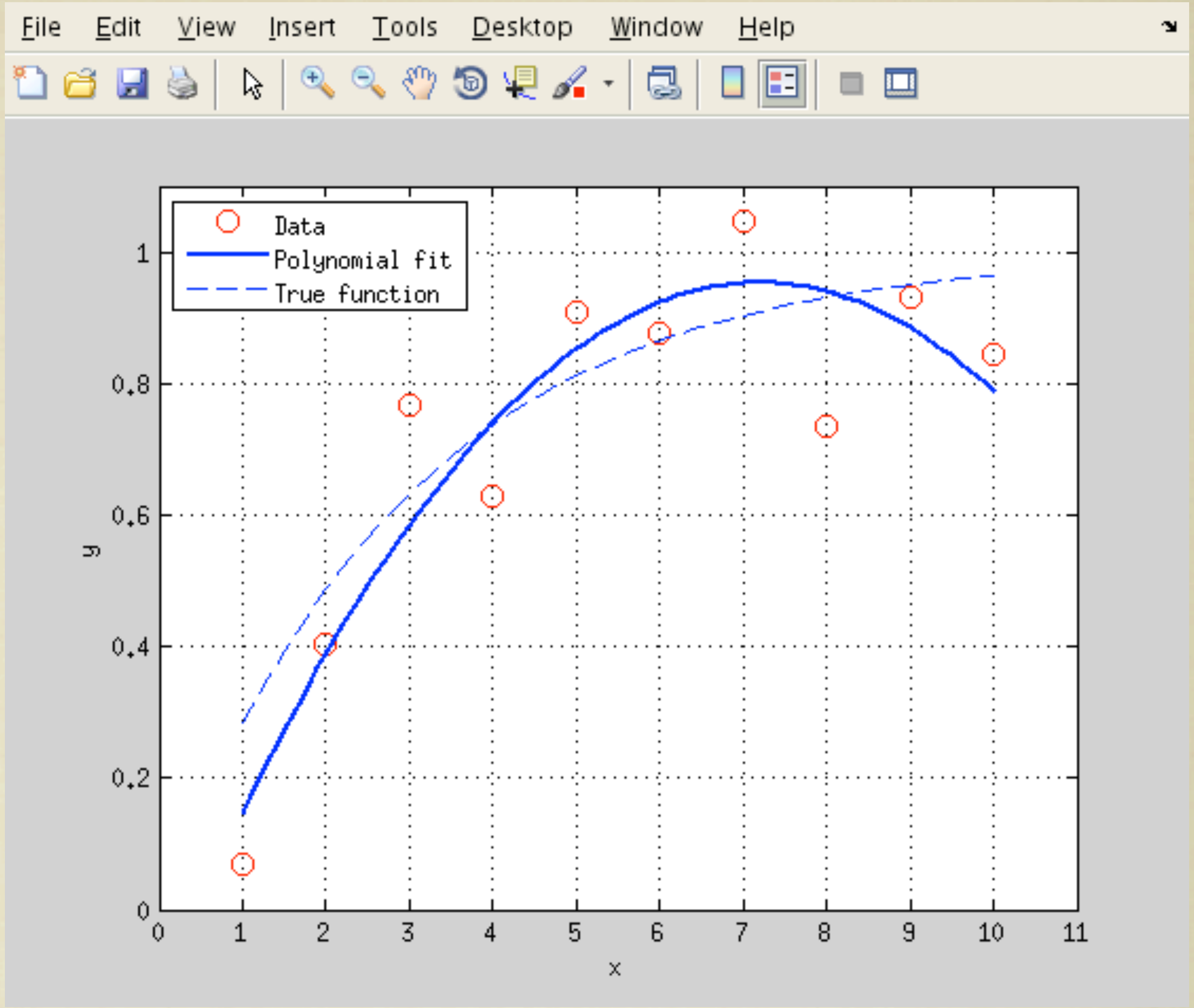
Algorithm 3

```
n ⟵ Number of MCMC steps.
s = {s_1, s_2, s_3, s_4} ⟵ Scale for displacement along each parameter axis.
Choose a random initial position in parameter space θ^(0) = {θ_1^(0), θ_2^(0), θ_3^(0), θ_4^(0)}
Calculate the log likelihood of the guessed parameter set, L(θ^(0)|O) using algorithm 2.
for i = 1 to i = n/4 do
  for k = 1 to k = 4 do
    l = 4(i − 1) + k − 1
    Propose a displacement δθ_k along the kth parameter axis, drawn from a normal
    distribution with mean 0 and variance s_k: θ^(proposed) = θ^(l) + δθ_k.
    Calculate L(θ^(proposed)|O).
    if L(θ^(proposed)|O) ≥ L(θ^(l)|O) then
      θ^(l+1) = θ^(proposed)
    else
      Generate a uniformly distributed random number u ∈ U(0, 1)
      if log u ≤ L(θ^(proposed)|O) − L(θ^(l)|O) then
        θ^(l+1) = θ^(proposed)
      else
        θ^(l+1) = θ^(l)
      end if
    end if
  end for
end for
```



Diagnostics: Asessing quality of fits

- Visual assessment: Does the fit look reasonable?
- Are the parameters estimates physically possible?
- Quantify: R^2

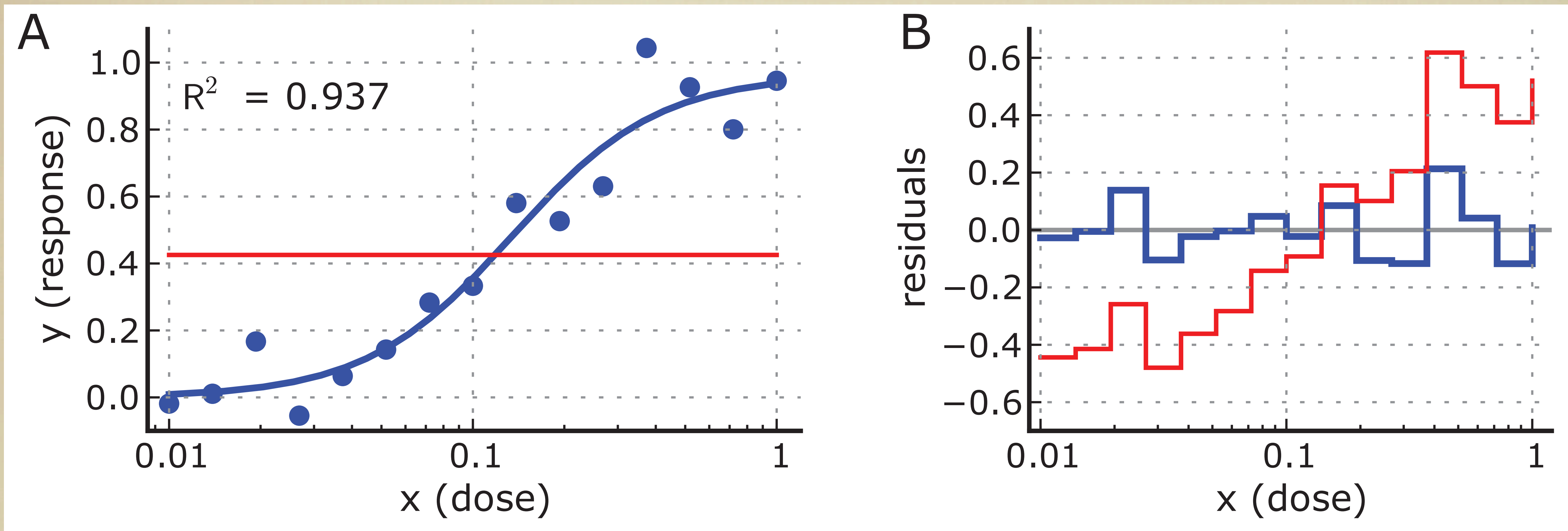


$$R^2 = 1 - \mathrm{SSR}/\mathrm{SST},$$

where SST is the total sum of squares (SST),

$$\mathrm{SST} = \sum_{i=1}^{N} \left[(y_i)_{\text{observed}} - \bar{y}_{\text{observed}} \right]^2,$$

Diagnostics: Asessing quality of fits



- Are the residuals randomly distributed?

Tomorrow

- Parameter confidence intervals.
- Bootstrap.
- Comparing parameters from two different fits - Hypothesis testing.