Scientific Computing – Statistics

Fabian Sinz
Dept. Neuroethology, University Tübingen
Bernstein Center Tübingen

10/23/2014

Day 4-5 – curve fitting and maximum likelihood

Overview

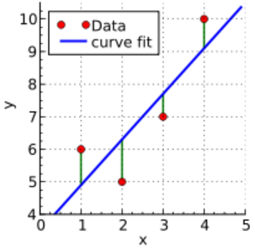
- minimizing/maximizing a function numerically (optimization) is ubiquitous in science (curve fitting, maximum likelihood, ...)
- today we will look at the basic elements of optimization and apply it to curve fitting
- tomorrow, we will apply it to maximum likelihood

plotting surfaces

```
range = linspace(-1,1,20);
[X,Y] = meshgrid(range, range);

surf(X,Y, (X.^2 + Y.^2));
colormap('winter');
```

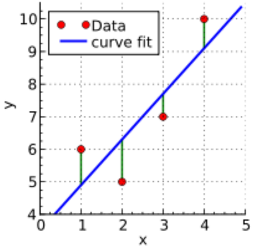
linear least squares



http://en.wikipedia.org/wiki/Linear_least_squares_%28mathematics%29

- The most common curve fitting problem is linear least squares.
- Its goal is to predict a set of output values $y_1, \ldots, y_n$ from their corresponding input values $x_1, \ldots, x_n$ with a line $f_{a,b}(x) = ax + b$.
- How is the line chosen?

linear least squares



http://en.wikipedia.org/wiki/Linear_least_squares_%28mathematics%29

- The most common curve fitting problem is linear least squares.
- Its goal is to predict a set of output values $y_1, \ldots, y_n$ from their corresponding input values $x_1, \ldots, x_n$ with a line $f_{a,b}(x) = ax + b$.
- How is the line chosen?

  By minimization of the mean squared error

$$g(a, b) = \sum_{i=1}^{n} (y_i - f_{a,b}(x_i))^2$$

error surface

plotting the error surface

- Write a function `lserr` that takes 2-dimensional parameter vector (slope a and offset b), an array of inputs `x`, an array of corresponding outputs `y`, and compute the least squares error

$$g(a,b) = \sum_{i=1}^{n} (y_i - f_{a,b}(x_i))^2$$

  with

$$f_{a,b}(x_i) = ax_i + b.$$

- Generate an example dataset with `x=linspace(-5,5,20)` and `y = .5*x + 1 + randn(length(x),1)`.
- Write a script that plots the error surface as a function of a and b.

optima and derivatives

- How did you find maxima/minima of functions at school?

optima and derivatives

- How did you find maxima/minima of functions at school?
  Compute the derivative, set it to zero, and solve for x.

optima and derivatives

- How did you find maxima/minima of functions at school?
  Compute the derivative, set it to zero, and solve for x.
- Can anybody remember how a derivative is defined (hint: differential quotient)?

optima and derivatives

- How did you find maxima/minima of functions at school?
  Compute the derivative, set it to zero, and solve for x.
- Can anybody remember how a derivative is defined (hint: differential quotient)?

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

- Could you write down this expression for the partial derivative $\frac{\partial g(a,b)}{\partial a}$ of $g(a, b)$ w.r.t. a?

optima and derivatives

- How did you find maxima/minima of functions at school?
  Compute the derivative, set it to zero, and solve for x.
- Can anybody remember how a derivative is defined (hint: differential quotient)?

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

- Could you write down this expression for the partial derivative $\frac{\partial g(a,b)}{\partial a}$ of $g(a, b)$ w.r.t. a?

$$\frac{\partial g(a, b)}{\partial a} = \lim_{h \to 0} \frac{g(a+h, b) - g(a, b)}{h}$$

optima and derivatives

- How did you find maxima/minima of functions at school?

  Compute the derivative, set it to zero, and solve for x.
- Can anybody remember how a derivative is defined (hint: differential quotient)?

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

- Could you write down this expression for the partial derivative $\frac{\partial g(a,b)}{\partial a}$ of $g(a, b)$ w.r.t. a?

$$\frac{\partial g(a, b)}{\partial a} = \lim_{h \to 0} \frac{g(a+h, b) - g(a, b)}{h}$$

- What about $\frac{\partial g(a,b)}{\partial b}$?

gradient and numerical derivative

gradient

The gradient

$$\nabla g(a,b) = \left(\frac{\partial g(a,b)}{\partial a}, \frac{\partial g(a,b)}{\partial b} \right)$$

is the vector with partial derivatives of g w.r.t. a and b.

We can numerically approximate it, by using the definition of the derivative

$$\frac{\partial g(a,b)}{\partial a} = \lim_{h \to 0} \frac{g(a+h,b) - g(a,b)}{h} \approx \frac{g(a+h,b) - g(a,b)}{h},$$

for very small h (e.g. `h=1e-6`).

error surface

plotting the gradient field

- Write a function `lserr_gradient` that takes the same arguments as `lserr`, but numerically computes the gradient

$$\nabla g(a, b) = \left(\frac{\partial g(a, b)}{\partial a}, \frac{\partial g(a, b)}{\partial b} \right)$$

- Add the gradient field as a vector field to your plot (use `quiver`).
- Add a contour plot of the error surface as well (use `contour`).
- What can you observer about the directions of the gradient with respect to the contour lines?

gradient descent

- The gradient $\nabla g(a, b)$ always points in the direction of steepest ascent.

gradient descent

- The gradient $\nabla g(a, b)$ always points in the direction of steepest ascent.
- How do we get the direction of steepest descent?

gradient descent

- The gradient $\nabla g(a, b)$ always points in the direction of steepest ascent.
- How do we get the direction of steepest descent?

  We take minus the gradient $-\nabla g(a, b)$.

gradient descent

- The gradient $\nabla g(a, b)$ always points in the direction of steepest ascent.
- How do we get the direction of steepest descent?

  We take minus the gradient $-\nabla g(a, b)$.

gradient descent algorithm

1. Start at some starting point $\mathbf{p}_0 = (a_0, b_0)$.
2. Repeat while gradient is large enough
   - Compute the gradient at the current position $\mathbf{p}_t = (a_t, b_t)$.
   - Walk a small step into the gradient direction via

   $$\mathbf{p}_{t+1} = \mathbf{p}_t - \varepsilon \nabla g(a_t, b_t)$$

   where ε is a small number.

gradient descent

gradient descent

- Implement a gradient descent for our linear regression problem.
- At each step in the algorithm, plot the error surface and the current parameter point (hint use `plot3` to plot a point in 3D).
- At each step also plot the linear regression line along with the data points in a separate plot.
- It is a good idea to use `pause(.1)` after each plot, so matlab has time updating the plots and you have time watching the gradient descent at work.

optimization with matlab

A little adaptation for the objective function

```
function [err, grad] = lserr(param, x, y)
    err = mean( (param(1)*x + param(2) - y).^2 );

    if nargout == 2
        grad = lserr_gradient(param, x,y);
    end
```

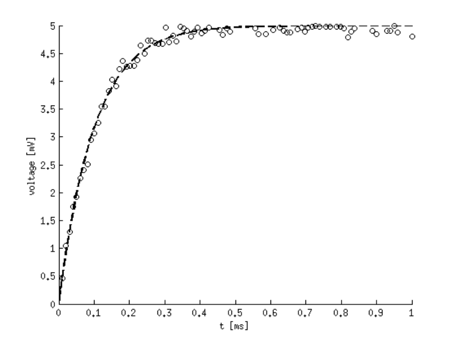
The actual optimization

```
function param = estimate_regression(x,y, param0)
    myfunc = @(p)(lserr(p,x,y));
    param = fminunc(myfunc,param0, options);
```

nonlinear regression

fit a charging curve

The following problem arises when estimating the time constant of a membrane from data.



- Download the data `membraneVoltage.mat`. It contains the points plotted on the right hand side.
- Write a nonlinear least squares fit to fit the function

$$f_{A,\tau}(t) = A \cdot \left(1 - e^{-\frac{t}{\tau}}\right)$$

  to the data.
- This looks scary, but it is not: If you programmed everything correctly beforehand you only need to adapt the function `lserr` and use the optimization from the slide before.
- Plot the final result along with the data points.

That's it.