Nonlinear Regression

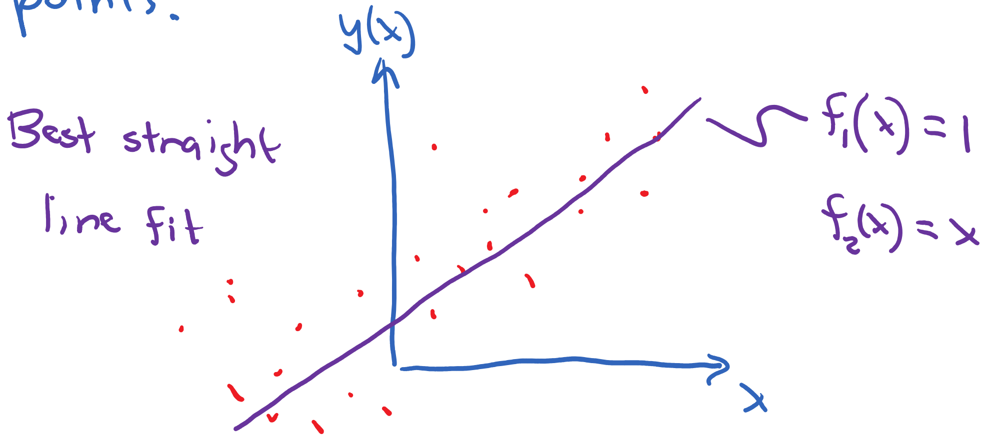
Linear regression

regression function

Fit $\hat{y}(x) = a_1 f_1(x) + a_2 f_2(x) + \ldots + a_p f_p(x)$

scalar

to n data points.



Functions $f_i(x)$ could be nonlinear, but this involves a linear combination of the regression functions

Example: $f(x) = \{1, x, x^2\}$ $\quad p = 3$

$$\hat{y}(x) = \sum_{i=1}^{3} a_i f_i(x) = a_1(1) + a_2(x) + a_3(x^2)$$

$$= a_1 + a_2 x + a_3 x^2$$

Determine a_1, a_2 & a_3 from the data

Example: $f(x) = \{\sin(x), \cos(x), \sin(2x), \cos(2x)\}$ $p=4$

$$\hat{y}(x) = a_1 \sin(x) + a_2 \cos(x) + a_3 \sin(2x) + a_4 \cos(2x)$$

Ideally, $\hat{y}(x_i) = y_i$ for data pair (x_i, y_i)

$$\hat{y}(x_1) = a_1 f_1(x_1) + a_2 f_2(x_1) + \dots + a_p f_p(x_1) = y_1$$

$\vdots$ known regressors $\vdots$

unknown coefficients

$$\hat{y}(x_n) = a_1 f_1(x_n) + a_2 f_2(x_n) + \dots + a_p f_p(x_n) = y_n$$

Construct linear system matrices

$$\underbrace{\begin{bmatrix} f_1(x_1) & \dots & f_p(x_1) \\ f_1(x_2) & \dots & f_p(x_2) \\ \vdots & & \vdots \\ f_1(x_n) & \dots & f_p(x_n) \end{bmatrix}}_{n \times p} \underbrace{\begin{bmatrix} a_1 \\ \vdots \\ a_p \end{bmatrix}}_{p \times 1} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}}_{n \times 1}$$

usually

$n \gg p$

$$\underline{F} \quad \underline{a} = \underline{y}$$

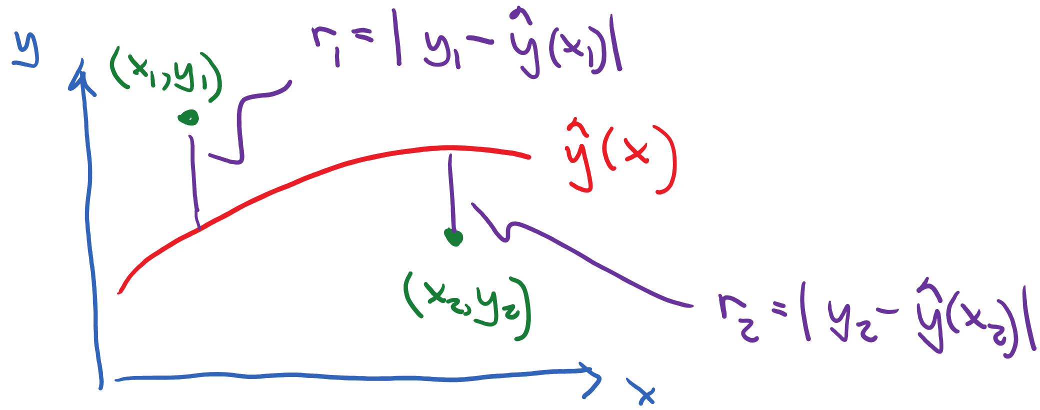
If $n = p$ & $x_i \neq x_j$ if $i \neq j$, then $\underline{F}^{-1}$ exists & $\underline{a}$ can be found

In general, $n \neq p$ & $x_i = x_j$ if $i \neq j$ is possible, then $\underline{F}^{-1}$ does not exist

No solution to $\underline{F}\,\underline{a} = \underline{y}$ in this case

Instead, find best approximate solution.

Find the $\hat{\underline{a}}$ that minimizes $\|\underline{r}\|_2 = \|\underline{y} - \underline{F}\,\hat{\underline{a}}\|_2$



The solution to the Normal Equations

$$\underline{F}^T \underline{F}\, \hat{\underline{a}} = \underline{F}^T \underline{y}$$

gives the least-squares solution to $\underline{F}\,\hat{\underline{a}} = \underline{y}$

However, never form the normal equations

Why?

Recall, matrix condition number

$$\kappa(\underline{A}^T) = \kappa(\underline{A})$$

$$\kappa(\underline{A}\,\underline{B}) = \kappa(\underline{A})\,\kappa(\underline{B})$$

$$\Rightarrow \kappa(\underline{F}^T\underline{F}) = \kappa(\underline{F}^T)\,\kappa(\underline{F}) = (\kappa(\underline{F}))^2$$

If $\kappa(\underline{F})$ is large, then

$\kappa(\underline{F}^T\underline{F})$ is huge!

Later in course, we will show indirect methods to solve the normal equations (e.g. QR, SVD)

$$\underline{F}\,\underline{a} = \underline{y} \qquad\qquad \underline{F}^T\underline{F}\,\hat{\underline{a}} = \underline{F}^T\underline{y}$$

$$\underline{F}\,\underline{a} = \underline{y}$$
$$(n \times p)(p \times 1) = (n \times 1)$$

$$\underline{F}^T \underline{F}\,\underline{\hat{a}} = \underline{F}^T \underline{y}$$
$$(p \times n)(n \times p)(p \times 1) = (p \times n)(n \times 1)$$
$$(p \times p)\,(p \times 1) = (p \times 1)$$

$\underline{F}^T \underline{F}$ is symmetric & real

$(\underline{F}^T \underline{F})^{-1}$ will always exist

In Matlab, the backslash operator will give the least-squares solution:

$$a = F \backslash y \qquad \text{for } \underline{a} = \underline{F}^{-1} \underline{y}$$

Nonlinear regression

Now, consider a nonlinear function, such as

$$y(x) = \frac{a_1 x}{a_2 + x} \rightarrow \underline{a} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$$

No longer possible to make a linear system

For solution

Use the Gauss-Newton algorithm

Let a set of n data points (x_i, y_i) be given and the objective is to fit a function $y(x, \underline{a})$, where $\underline{a}$ is the vector of p unknown coefficients (typically $n \gg p$)

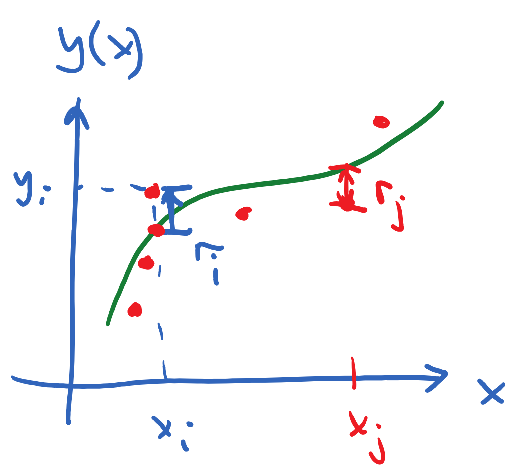
Define the residual at the n points as

$$r_i = y_i - y(x_i, \underline{a}) \quad \text{for } i = 1, 2, \dots, n$$

Want to minimize the objective function

$$S(\underline{a}) = \sum_{i=1}^{n} r_i^2$$

$$\underline{a} \in \mathbb{R}^p$$



Using a Newton method

$$\underline{a}_{m+1} = \underline{a}_m - \underline{H}_m^{-1} \underline{g}_m \quad \text{for iteration } m$$

with

$\underline{g}_m$: gradient of S wrt $\underline{a}$ $\quad \left(\underline{g}_m = \nabla S(\underline{a}_m) \right)$

$$g_j = \left.\frac{\partial S}{\partial a_j}\right|_{\underline{a}_m}$$

$\underline{H}_m$: Hessian of S @ $\underline{a}_m$

$$H_{jk} = \frac{\partial^2 S}{\partial a_j \partial a_k} = \left.\frac{\partial g_j}{\partial a_k}\right|_{\underline{a}_m}$$

Since $S = \sum_{i=1}^{n} r_i^2$

$$g_j = 2 \sum_{i=1}^{n} r_i \frac{\partial r_i}{\partial a_j}$$

$$H_{jk} = 2 \sum_{i=1}^{n} \left(\frac{\partial r_i}{\partial a_j} \frac{\partial r_i}{\partial a_k} + \underbrace{\cancel{r_i \frac{\partial^2 r_i}{\partial a_j \partial a_k}}} \right)$$

Gauss-Newton ignores this term
(Second derivatives are noisy)

$$\therefore \; H_{jk} \approx 2 \sum_{i=1}^{n} \frac{\partial r_i}{\partial a_j} \frac{\partial r_i}{\partial a_k} = 2 \sum_{i=1}^{n} J_{ij} J_{ik}$$

$$J_{ij} = \frac{\partial r_i}{\partial a_j} \Rightarrow \text{Jacobian of } \underline{r} \text{ wrt } \underline{a}$$

$$ {}_{ij} \quad \partial a_j $$

Notice that

$$\sum_{i=1}^{n} \frac{\partial r_i}{\partial a_j} r_i \leftrightarrow \underline{J}^T \underline{r}$$

$$\sum_{i=1}^{n} J_{ij} J_{jk} \leftrightarrow \underline{J}^T \underline{J}$$

$$\Rightarrow \underline{g} = 2 \underline{J}^T \underline{r} \quad + \quad \underline{H} \approx 2 \underline{J}^T \underline{J}$$

$$\underline{a}_{m+1} = \underline{a}_m - \underline{H}_m^{-1} \underline{g}_m$$

$$= \underline{a}_m - \frac{1}{2} \left(\underline{J}_m^T \underline{J}_m \right)^{-1} \left(2 \underline{J}_m^T \underline{r}_m \right)$$

iteration m

$$\underline{a}_{m+1} = \underline{a}_m - \left(\underline{J}_m^T \underline{J}_m \right)^{-1} \underline{J}_m^T \underline{r}_m$$

Typically this is combined with a line search, for example,

$$\underline{\Delta}_m = - \left(\underline{J}_m^T \underline{J}_m \right)^{-1} \underline{J}_m^T \underline{r}_m$$

and then

$$\underline{a}_{m+1} = \underline{a}_m + \alpha_m \underline{\Delta}_m$$

$$\underline{a}_{m+1} = \underline{a}_m + \alpha_m \underline{\Delta}_m$$

for an α_m such that $S(\underline{a}_{m+1}) < S(\underline{a}_m)$

Advantage of this method: No need for 2nd derivatives of $\underline{r}$

Disadvantage: Might not converge

For convergence, one needs the approximate $\underline{H}$ to be close to the true $\underline{H}$, that is, when

$$\left| r_i \frac{\partial^2 r_i}{\partial a_j \partial a_k} \right| \ll \left| \frac{\partial r_i}{\partial a_j} \frac{\partial r_i}{\partial a_k} \right|$$

Typically, this happens if r_i is already small or if $y(x, \underline{a})$ is $\underbrace{\text{only mildly nonlinear}}$

Derivatives higher than first are small