COS 424: Interaction with Data

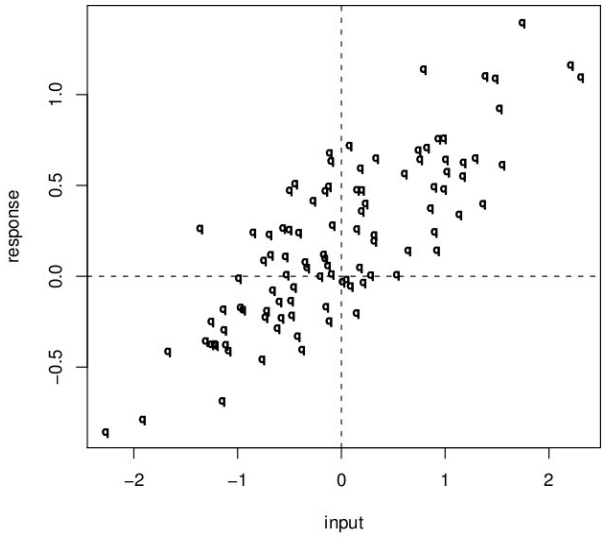
Lecturer: David Blei
Scribe: Xiaobai Chen, Jialu Huang

Lecture # 15
April. 1, 2008

Linear Regression (1)

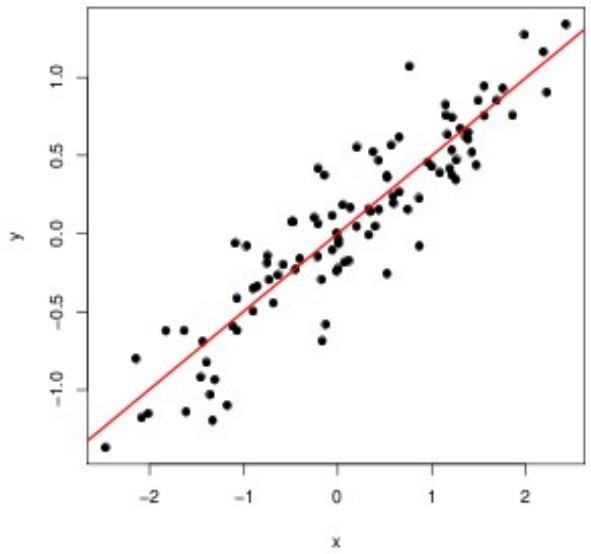
What is regression ?
Regression is the problem to predict a real-valued variable from input data

An example below



Data are a set of inputs and outputs $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N}$

So now the question is how to find a linear function so that whenever given a x, we can predict the value of y:



Other examples that linear regression may be applied to include:

1. How tall with respect to how much the weight
2. to guess the score between two basketball teams, given any statistics about the teams
3. How much people earn after graduation Vs. his/her grade at school
4. How much do people spend on their cars according to their salary

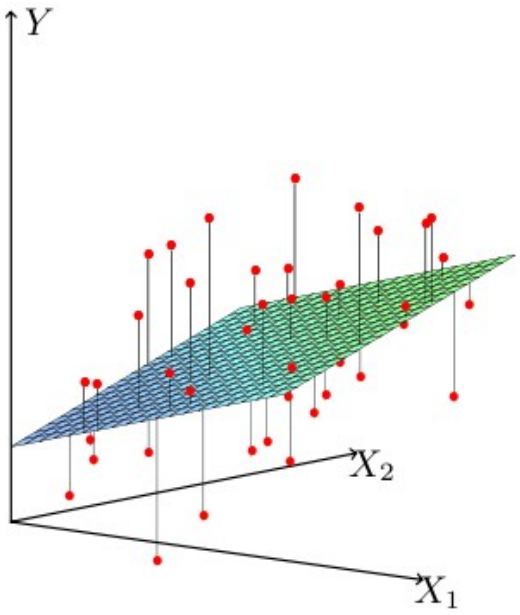
We talk about multi-inputs ("x" is a vector now, and each element represents a different feature)

$$x = \langle x_1, x_2, \ldots, x_p \rangle$$

The response is assumed to be a linear function of the input:

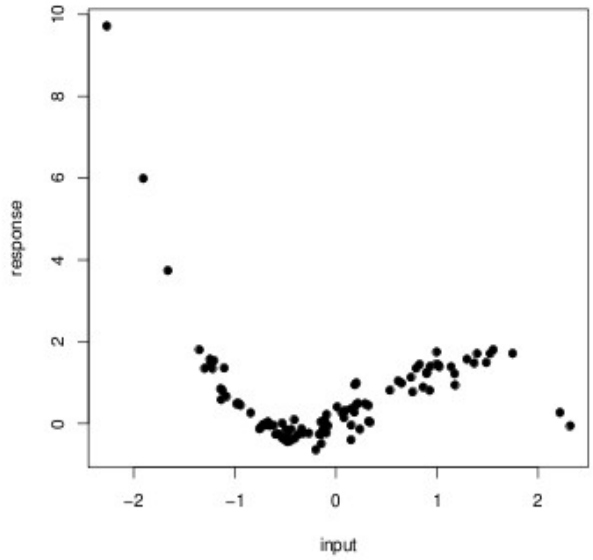
$$f(x) = \beta_0 + \sum_{i=1}^{p} x_i \beta_i$$

$\beta^T x = 0$ is a hyperplane as shown in the following graph:

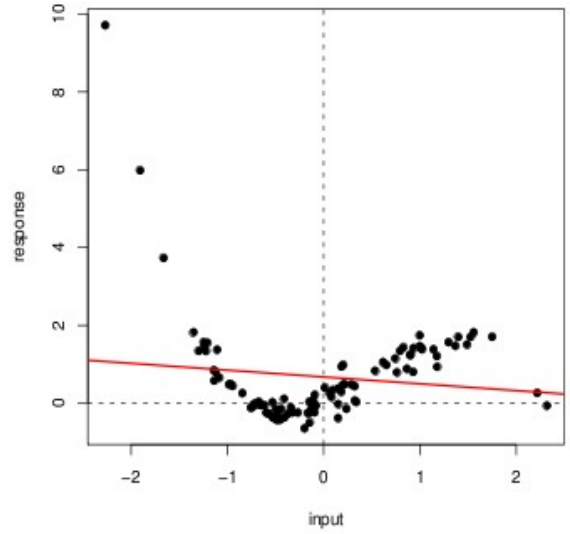


One thing we need to point out here is that linear regression has much more flexibility than you might imagine: We may use any function of x as input, like $x^2, x^3 \ldots$, etc. In a word, its simplicity and flexibility make linear regression one of the most important and widely used statistical prediction techniques.

Of course, we also have problems which cannot be solved by linear regression. Here we give an example for polynomial regression:

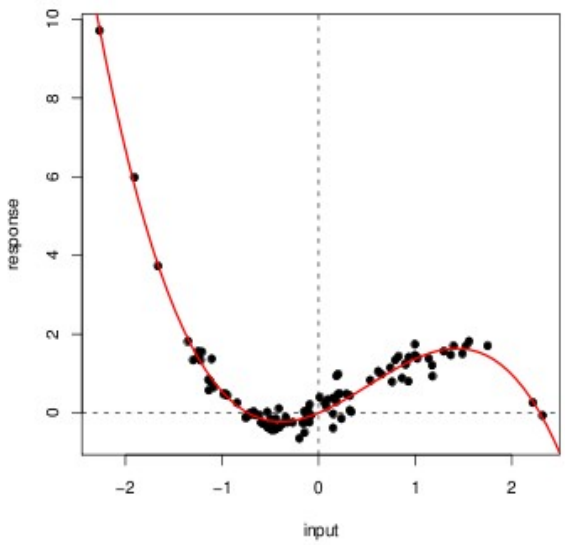


If we use linear regression to plot a function, we might get result like:



$$f(x) = \beta_0 + \beta x$$

which obviously does not fit to the data points at all. But if we use polynomial regression, we can get result like:



$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$$

which fits all the data points much better.

So now we start considering how to fit a regression:

Give data $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N}$,

now we want to find the coefficient β that can predict y_{new} from x_{new}.

We start from a simple case: a single input feature, and β_0 = 0 in this stage:
A reasonable approach is to minimize sum of squared Euclidean distance between each prediction and truth, so we get our objective function for optimization :

$$\mathrm{RSS}(\beta) = \frac{1}{2}\sum_{n=1}^{N}(y_n - \beta x_n)^2$$

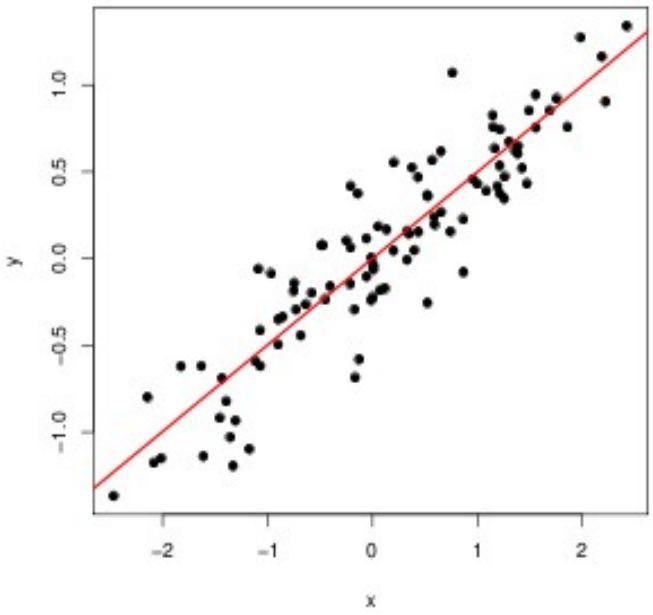
The derivation of β :

$$\frac{d}{d\beta}\mathrm{RSS}(\beta) = -\sum_{n=1}^{N}(y_n - \beta x_n)x_n$$

The optimal value is:

$$\hat{\beta} = \frac{\sum_{n=1}^{N} y_n x_n}{\sum_n x_n^2}$$

According this optimal value, we can plot a line like the following:



Based on this line, when we need to predict a new output from a new input, we just use the point on the line at that input:

Now we want to move on to multiple inputs:

In general:

$$y = \beta_0 + \sum_{i=1}^{p} \beta_i x_i$$

To simplify, let β be a p+1 vector and set $x_{p+1} = 1$. Now the RSS is:

$$\mathrm{RSS}(\beta) = \frac{1}{2} \sum_{n=1}^{N} (y_n - \beta^\top x_n)^2$$

The derivative with respect to β_i is:

$$\frac{d}{d\beta_i} = -\sum_{n=1}^{N} (y_n - \beta_i x_{n,i}) x_{n,i}$$

As a vector, the gradient is:

$$\frac{d}{d\beta_i} = -\sum_{n=1}^{N} (y_n - \beta_i x_{n,i}) x_{n,i}$$

First, we define: the design matrix is an N x (p+1) matrix X, the response vector is an N-vector y, and the parameter vector is a (p+1)-vector β, then the gradient of

the RSS is:

$$\nabla_\beta \mathrm{RSS} = -\sum_{n=1}^{N}(y_n - \beta^\top x_n)x_n$$

Setting to the 0-vector and solving for β , we get:

$$\hat{\beta} = (X^\top X)^{-1}X^\top y$$

This works as long as $X^\top X$ is invertible, i.e., X is full rank.

Question: What is the probabilistic interpretation for linear regression?

Linear regression assumes that the output are drawn from a Normal distribution whose mean is a linear function of the coefficients and the input:

$$Y_n \,|\, x_n, \beta \sim \mathcal{N}(\beta \cdot x_n, \sigma^2)$$

This is like putting a Gaussian "bump" around the mean, which is a linear function of the input.

We find the parameter vector β that maximizes the conditional likelihood. The conditional log likelihood of data $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^N$ is:

$$\begin{aligned}
\mathcal{L}(\beta) &= \log \prod_{n=1}^{N} p(y_n \,|\, x_n, \beta) \\
&= \log \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{\frac{-(y_n - \beta^\top x_n)^2}{2\sigma^2}\right\} \\
&= \sum_{n=1}^{N} -\frac{1}{2}\log 2\pi\sigma^2 - \frac{1}{2}(y_n - \beta^\top x_n)^2/\sigma^2
\end{aligned}$$

Then we maximize the conditional log likelihood with respect to β :

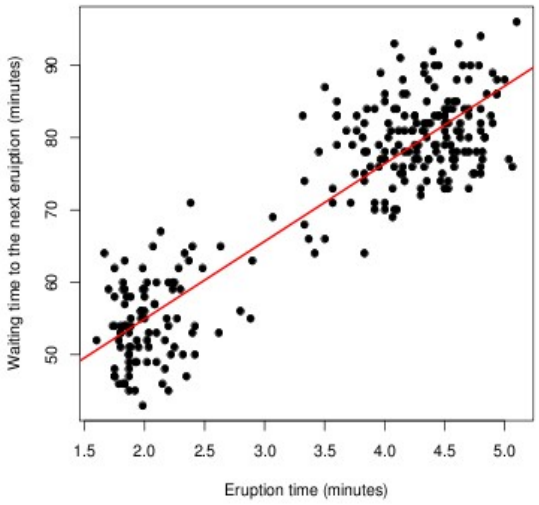
$$\mathcal{L}(\beta) = \sum_{n=1}^{N} -\frac{1}{2}\log 2\pi\sigma^2 - \frac{1}{2}(y_n - \beta^\top x_n)^2/\sigma^2$$

which is the same as minimizing the residual sum of squares

$$\mathrm{RSS}(\beta) = \frac{1}{2}(y_n - \beta^\top x_n)^2$$

So the maximum likelihood estimates are identical to the estimates we obtained earlier. We are actually estimating the conditional expectation of y given x

A real-world example is given the eruption time, we try to predict the waiting time to the next eruption:



Important asides:

Bias-Variance trade off
Consider a random data set that is drawn from a linear regression model. We can contemplate the maximum likelihood estimate $\hat{\beta}$ as a random variable whose distribution is governed by the distribution of the data set. Suppose we observe a new data input x, we can consider the mean squared error of our estimate of $\mathrm{E}[y \mid x] = \hat{\beta}x$.

$$\mathrm{MSE}(\hat{\beta}x) = \mathrm{E}_{\mathcal{D}}[(\hat{\beta}x - \beta x)^2]$$

Here β is not random and $\hat{\beta}$ is random, so:

$$\begin{aligned}
\mathrm{MSE} &= \mathrm{E}[(\hat{\beta}x)^2] - 2\mathrm{E}[\hat{\beta}x]\beta x + (\beta x)^2 \\
&= \mathrm{E}[(\hat{\beta}x)^2] - 2\mathrm{E}[(\hat{\beta}x)](\beta x) + (\beta x)^2 + \mathrm{E}[(\hat{\beta}x)]^2 - \mathrm{E}[(\hat{\beta}x)]^2 \\
&= \left(\mathrm{E}[(\hat{\beta}x)^2 - \mathrm{E}[\hat{\beta}x]^2\right) + \left(\mathrm{E}[\hat{\beta}x] - \beta x\right)^2
\end{aligned}$$

As a result, we can get MSE:

$$\mathrm{MSE} = \left(\mathrm{E}[(\hat{\beta}x)^2] - \mathrm{E}[\hat{\beta}x]^2\right) + \left(\mathrm{E}[\hat{\beta}x] - \beta x\right)^2$$

The second term is the squared bias:

$$\mathrm{bias} = \mathrm{E}[\hat{\beta}x] - \beta x$$

An estimate for which this term is zero is an unbiased estimate.
The first term is the variance:

$$\mathrm{variance} = \mathrm{E}[(\hat{\beta}x)^2] - \mathrm{E}[\hat{\beta}x]^2$$

This reflect s how sensitive the estimate is to the randomness inherent in the data.