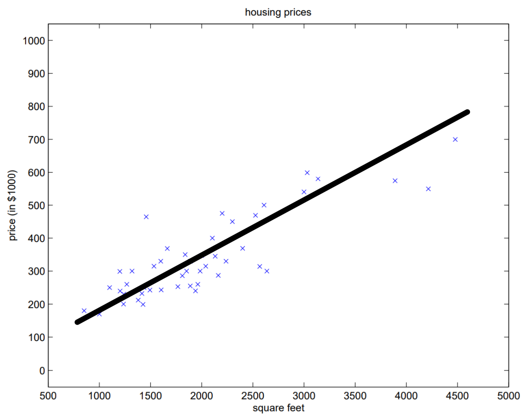
Lecture 1. Linear Model.



$x^{(i)}$: features.

$y^{(i)}$: output

$(x^{(i)}, y^{(i)})$: training example.

$h : X \to Y$: hypothesis.

*: predict – continous: regression
discrete: classification.

- Linear Regression:

(1) $h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$.

$h_\theta(x) = \theta^T x$. (vector-form)

(2) $h_\theta(x) \to$ close to y.

cost function:

$J(\theta) = \frac{1}{2} \cdot \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

$\min_\theta J(\theta)$

- LMS Alogrithm:

(1) gradient descent:

$\theta_j = \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j}$

$\frac{\partial}{\partial \theta_j} J(\theta) = (h_\theta(x) - y) x_j$ (single training data)

$\theta_j = \theta_j - \alpha (h_\theta(x) - y) x_j$

$\Rightarrow \theta_j := \theta_j - \alpha \sum_{i=1}^{m} (h_\theta(x) - y) x_j$

(2) entire training set $\Rightarrow$ batch gradient descent.

(3) stochastic GD:

for $i = 1 \sim m$:

$\theta_j := \theta_j - \alpha (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$

*: make progress right away.

- Normal Equations:

(1) Matrix derivatives:

$\nabla_A f(A) = \begin{bmatrix} \frac{\partial f}{\partial A_{11}} & \cdots & \\ \vdots & & \\ \frac{\partial f}{\partial A_{m1}} & \cdots & \frac{\partial f}{\partial A_{mn}} \end{bmatrix}$ $\rightarrow$ a matrix

$tr(AB) = tr(BA)$

$\nabla_A tr(AB) = B^T$

$\nabla_{A^T} f(A) = [\nabla_A f(A)]^T$

$\nabla_A |A| = |A| (A^{-1})^T$ (adjoint)

(2) Least Square:

$X = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_m^T \end{bmatrix}$ (design matrix)

$y = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$ $\quad \theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}$

$h_\theta(x_i) = x_i^T \theta$

We can know:

$$X\theta - y = \begin{bmatrix} x_1^T\theta - y_1 \\ \vdots \\ x_m^T\theta - y_m \end{bmatrix}$$

$J(\theta) = \frac{1}{2}(X\theta - y)^T(X\theta - y)$ (using derivative)

$$= \frac{1}{2}(\theta^T X^T - y^T)(X\theta - y)$$

$$\nabla_\theta J(\theta) = \nabla_\theta \frac{1}{2}(\theta^T X^T X\theta - 2y^T X\theta)$$

$$= X^T X\theta - X^T y = 0$$

$\Leftrightarrow$ $\theta = (X^T X)^{-1} X y$.

- Probabilistic interpretation:

(1) $y_i = \theta^T x_i + \varepsilon_i$ → noise/error term.

assuming: $\varepsilon_i \sim$ i.i.d Gaussian.

$$p(\varepsilon_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\varepsilon_i^2}{2\sigma^2}\right)$$

$$\Leftrightarrow p(y_i | \theta, x_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - \theta^T x_i)^2}{2\sigma^2}\right)$$

(2) Likelihood function:

$$L(\theta) = p(\vec{y} | X; \theta)$$

$$= \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - \theta^T x_i)^2}{2\sigma^2}\right)$$

*: maximum likelihood. → log trick.

$l(\theta) = \log L(\theta)$ (minimum)

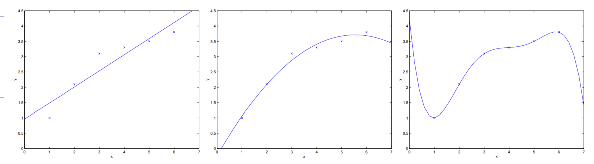
$$= m\log\frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{2\sigma^2}\sum_{i=1}^{m}(y_i - \theta^T x_i)^2$$

(3) different assumption → loss function.

- Locally weighted linear regression:

  (1) not a line → extra figure.

  underfit / overfit.



  (2) local weighted:

  minimize: $\sum w_i (y_i - \theta x_i)^2$ $(w_i > 0)$

  non-parametric

  (don't know the distribution)

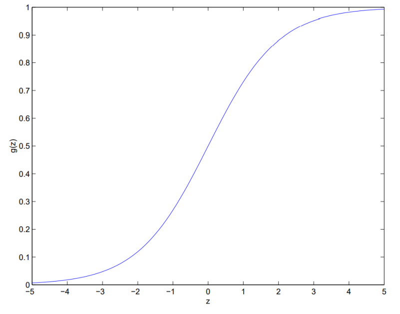
- Classification

  (1) predicting value y:

  discrete few values.

  (2) binary classification.

  $\left.\begin{array}{l} 1 \sim \text{positive} \\ 0 \sim \text{negative.} \end{array}\right] \rightarrow$ label.



- Logistic regression:

  (1) hypothesis:

  $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$ (sigmod function)

  bounded between [0,1].

$$g(z) = \frac{1}{1+e^{-z}}$$

$$g'(z) = \frac{e^{-z}}{(1+e^{-z})^2} = g(z)(1-g(z))$$

(2) probabilistic assumption:

$P(y=1 \mid x, \theta) = h_\theta(x)$ output: $P \in (0,1)$

$P(y=0 \mid x, \theta) = 1 - h_\theta(x)$.

$P(y \mid x, \theta) = [h_\theta(x)]^y [1-h_\theta(x)]^{1-y}$

$l(\theta) = \log L(\theta)$ (log likelihood)

$= \sum_{i=1}^{m} y_i \log[h_\theta(x_i)] + (1-y_i)\log[1-h_\theta(x_i)]$.

$\frac{\partial l}{\partial \theta_j} = (y - h_\theta(x))x_j$

$\theta := \theta + \alpha(y_i - h_\theta(x_i))x_i$

(3) same form, different function.

*: GLM. models.

- Perceptron learning alogrithm:

(1) output value: either 1 / 0.

$$g(z) = \begin{cases} 1 & z \geq c \\ 0 & z < c. \end{cases}$$

(2) starting point for learning theory.

- Another optimization method.

(1) Newton method.

$\theta \rightarrow (\theta, f(\theta))$

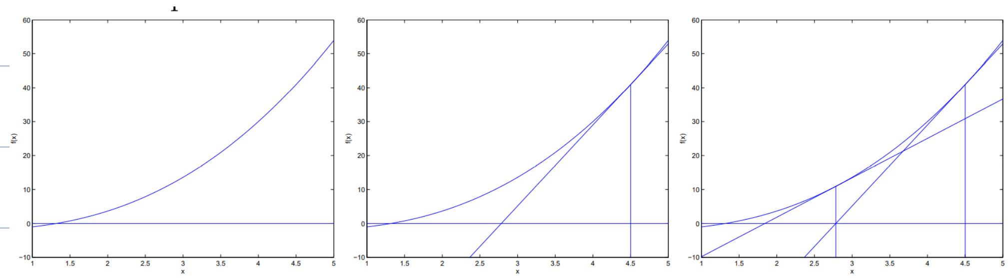
targent: $y - f(\theta) = f'(\theta)(x-\theta)$

Let $y=0$: $x = \theta - \frac{f(\theta)}{f'(\theta)}$

(2) maximize $l(\theta)$. $\Rightarrow l'(\theta) = 0$ (convex)

Let $f(x) = l'(\theta)$. $\Rightarrow$ find root.

$\theta := \theta - \frac{l'(\theta)}{l''(\theta)}$



(3) Vector-valued: (multi-dimension)

$\theta := \theta - H^{-1} \nabla_\theta l(\theta)$

H: Hessian Matrix.

$H_{ij} = \frac{\partial^2 l(\theta)}{\partial \theta_i \partial \theta_j}$

*: faster convergence.

- Generalized linear model:

(1) distribution:

regression: $y | x, \theta \sim N(\mu, \sigma^2)$

classification: $y | x, \theta \sim \text{Bernoulli}(\Phi)$

$\rightarrow$ GLM.

(2) exponential family:

$P(y, \eta) = b(y) \exp(\eta^T T(y) - a(\eta))$

η: natural parameter.

$T(y)$: sufficient statistic.

fixed T, a, b, parameter η $\Rightarrow$ family.

(5) Example:

$p(y;\Phi) = \Phi^{y}(1-\Phi)^{1-y}$

$= \exp(y\log\Phi + (1-y)\log(1-\Phi))$.

$= \exp\left[y\log\frac{\Phi}{1-\Phi} + \log(1-\Phi)\right]$.

- Constructing GLM:

(1) knowing distribution

$\Rightarrow$ constructing models.

(2) assumptions:

① $y|x;\theta \sim \text{Exponential}(\eta)$

② predict y.

$h(x) = E[y|x]$.

③ $\eta = \theta^T x$. (?)

(3) Examples:

$h_\theta(x) = E(y|x,\theta)$

$= \Phi$

$= 1/1+e^{-\eta}$

(Logistic Regression)

(4) response function.

$g(\eta) = E(T(y);\eta]$.

- Softmax Regression:

(1) k-classification

$y \in \{1, 2, \cdots k\}$.

$*$: multi-nomial distribution

(2) k-1 parameters. $\Phi_i \ (i = 1, 2, \cdots k-1)$

$\Phi_k = 1 - \sum_{i=1}^{k-1} \Phi_i$

(3) indicator function:

$T(y)_i = 1\{y = i\} \Rightarrow E(T(y)_i) = \Phi_i$

(4) multi-nomial distribution $\Rightarrow$ exponential.

$$P(y, \Phi) = \Phi_1^{T(y)_1} \Phi_2^{T(y)_2} \cdots \Phi_k^{1 - \sum_{i=1}^{k-1} T(y)_i}$$

$$= \exp\left[\sum_{i=1}^{k-1} T(y)_i \log \Phi_i + \left(1 - \sum_{i=1}^{k-1} T(y)_i\right) \log \Phi_k \right.$$

$$= \exp\left[\sum_{i=1}^{k-1} T(y)_i \log \frac{\Phi_i}{\Phi_k} + \log \Phi_k \right].$$

$$= b(y) \exp\left(\eta^T T(y) - a(\eta) \right)$$

where: $b(y) = 1$. $\quad a(\eta) = -\log \Phi_k \qquad \eta = \begin{bmatrix} \log \Phi_1 / \Phi_k \\ \vdots \\ \log \Phi_{k-1} / \Phi_k \end{bmatrix}$

(5) response function:

$\Phi_i = e^{\eta_i} / \sum_{j=1}^{k} e^{\eta_j}$ (softmax function)

output the estimated probability:

$P(y = i \mid x, \theta)$ for $i = 1, 2, \cdots k$.