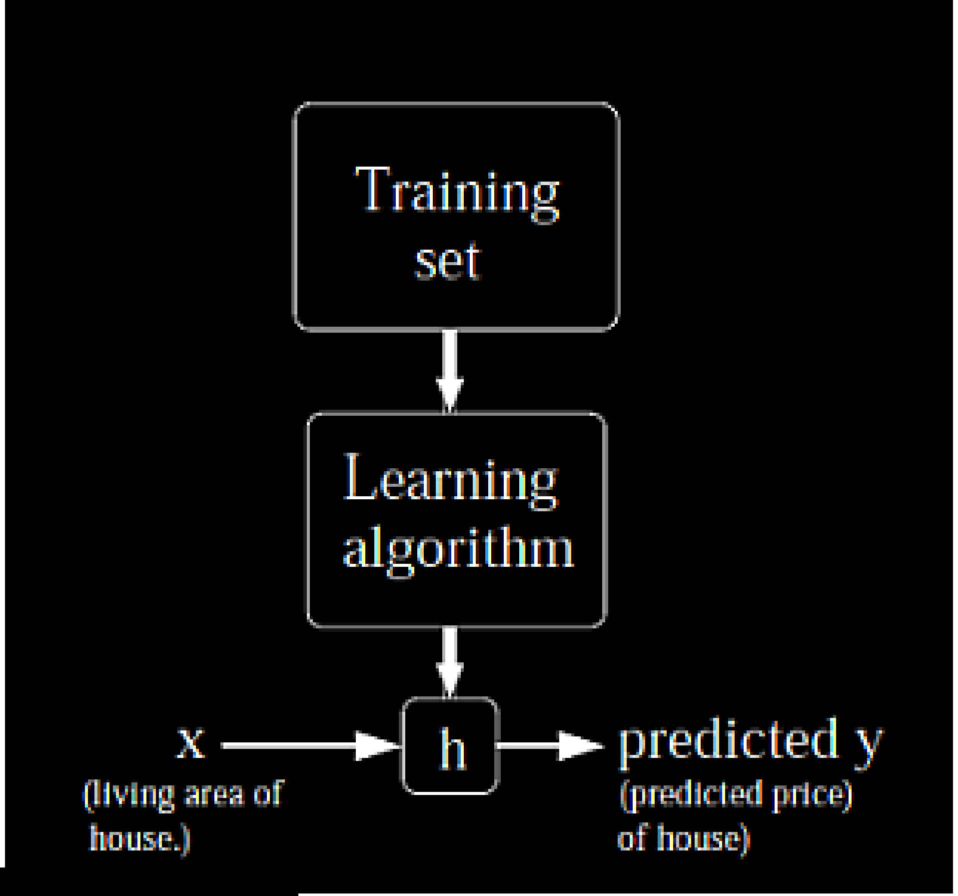
LOGISTIC REGRESSION



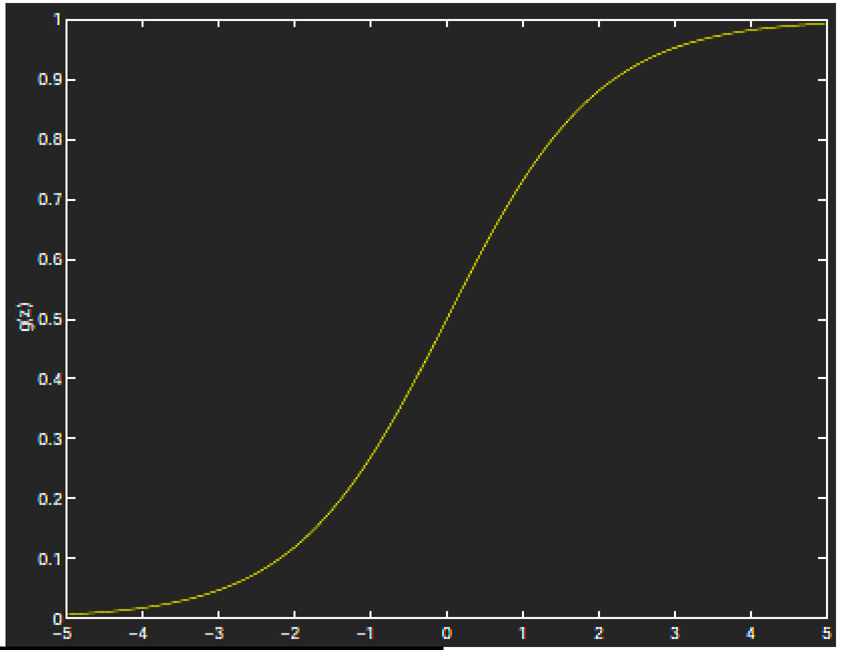
$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}},$$

where

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

is called the logistic function or the sigmoid function.

$$\theta^T x = \theta_0 + \sum_{j=1}^{n} \theta_j x_j$$



We define the cost function:

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

We want to choose θ so as to minimize $J(\theta)$. To do so, let's use a search algorithm that starts with some "initial guess" for θ, and that repeatedly changes θ to make $J(\theta)$ smaller, until hopefully we converge to a value of θ that minimizes $J(\theta)$. Specifically, let's consider the **gradient descent** algorithm, which starts with some initial θ, and repeatedly performs the update:

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

(This update is simultaneously performed for all values of $j = 0, \ldots, n$.) Here, α is called the **learning rate**. This is a very natural algorithm that repeatedly takes a step in the direction of steepest decrease of J.

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

where

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

is called the logistic function or the sigmoid function.

$$\begin{aligned}
g'(z) &= \frac{d}{dz}\frac{1}{1 + e^{-z}} \\
&= \frac{1}{(1 + e^{-z})^2}\left(e^{-z}\right) \\
&= \frac{1}{(1 + e^{-z})} \cdot \left(1 - \frac{1}{(1 + e^{-z})}\right) \\
&= g(z)(1 - g(z)).
\end{aligned}$$

Logistic Regression

$$p(X) = \Pr(Y = 1|X)$$

Logistic regression uses the form

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}.$$

($e \approx 2.71828$ is a mathematical constant [Euler's number.])
It is easy to see that no matter what values β_0, β_1 or X take, $p(X)$ will have values between 0 and 1.

A bit of rearrangement gives ***(Solve it now)***

$$\log\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 X.$$

This monotone transformation is called the *log odds* or *logit* transformation of $p(X)$.

Maximum Likelihood

We use maximum likelihood to estimate the parameters.

$$\ell(\beta_0, \beta) = \prod_{i:y_i=1} p(x_i) \prod_{i:y_i=0} (1 - p(x_i)).$$

This *likelihood* gives the probability of the observed zeros and ones in the data. We pick β_0 and β_1 to maximize the likelihood of the observed data.

Logistic regression with several variables

$$\log\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

$$p(X) = \frac{e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}}$$

So, given the logistic regression model, how do we fit θ for it?

Let us assume that

$$\begin{aligned} P(y=1 \mid x;\theta) &= h_\theta(x) \\ P(y=0 \mid x;\theta) &= 1-h_\theta(x) \end{aligned}$$

Prob. model + MLE process

Note that this can be written more compactly as

$$p(y \mid x;\theta) = (h_\theta(x))^y \, (1-h_\theta(x))^{1-y}$$

Assuming that the m training examples were generated independently, we can then write down the likelihood of the parameters as

$$\begin{aligned} L(\theta) &= p(\vec{y} \mid X;\theta) \\ &= \prod_{i=1}^{m} p(y^{(i)} \mid x^{(i)};\theta) \\ &= \prod_{i=1}^{m} \left(h_\theta(x^{(i)})\right)^{y^{(i)}} \left(1-h_\theta(x^{(i)})\right)^{1-y^{(i)}} \end{aligned}$$

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

it will be easier to maximize

where

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

is called the logistic function or the sigmoid function.

$$\begin{aligned}\ell(\theta) &= \log L(\theta)\\ &= \sum_{i=1}^{m} y^{(i)} \log h(x^{(i)}) + (1-y^{(i)}) \log(1-h(x^{(i)}))\end{aligned}$$

one training example (x, y), and take derivatives to derive the stochastic gradient ascent rule:

$$\begin{aligned}\frac{\partial}{\partial\theta_j}\ell(\theta) &= \left(y\frac{1}{g(\theta^T x)} - (1-y)\frac{1}{1-g(\theta^T x)}\right)\frac{\partial}{\partial\theta_j} g(\theta^T x)\\ &= \left(y\frac{1}{g(\theta^T x)} - (1-y)\frac{1}{1-g(\theta^T x)}\right) g(\theta^T x)(1-g(\theta^T x))\frac{\partial}{\partial\theta_j}\theta^T x\\ &= \left(y(1-g(\theta^T x)) - (1-y)g(\theta^T x)\right) x_j\\ &= (y - h_\theta(x))\, x_j\end{aligned}$$

the stochastic gradient ascent rule

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

Same as LMS learning rule - except the non-linear sigmoid in "h".

Newton Raphson's method for maximizing $l(\theta)$

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

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

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

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

Logistic regression with more than two classes

So far we have discussed logistic regression with two classes. It is easily generalized to more than two classes. One version (used in the R package `glmnet`) has the symmetric form

$$\Pr(Y = k|X) = \frac{e^{\beta_{0k}+\beta_{1k}X_1+\ldots+\beta_{pk}X_p}}{\sum_{\ell=1}^{K} e^{\beta_{0\ell}+\beta_{1\ell}X_1+\ldots+\beta_{p\ell}X_p}}$$

Here there is a linear function for *each* class.

Multiclass logistic regression is also referred to as *multinomial regression.*

Consider a general classification problem, in which the response variable y can take on any one of k values, so y ∈ {1, 2, . . . , k}.

To parameterize a multinomial over k possible outcomes, one could use k parameters $\phi_1, \ldots, \phi_k$ specifying the probability of each of the outcomes.

(for i = 1, . . . , k)

$$\eta_i = \log \frac{\phi_i}{\phi_k}$$

let $\phi_k = 1 - \sum_{i=1}^{k-1} \phi_i$

response function

$$\phi_i = \frac{e^{\eta_i}}{\sum_{j=1}^{k} e^{\eta_j}}$$

response function

$$\phi_i = \frac{e^{\eta_i}}{\sum_{j=1}^{k} e^{\eta_j}}$$

This function mapping from the η's to the ϕ's is called the softmax function

$$\eta_i = \theta_i^T x \text{ (for } i = 1, \ldots, k-1)$$

where $\theta_1, \ldots, \theta_{k-1} \in \mathbb{R}^{n+1}$ are the parameters of our model

The conditional distribution of y given x is :

$$
\begin{aligned}
p(y=i|x;\theta) &= \phi_i \\
&= \frac{e^{\eta_i}}{\sum_{j=1}^{k} e^{\eta_j}} \\
&= \frac{e^{\theta_i^T x}}{\sum_{j=1}^{k} e^{\theta_j^T x}}
\end{aligned}
$$

This model, which applies to classification problems where y ∈ {1, . . . , k}, is called **softmax regression**.
It is a generalization of logistic regression.

If we have a training set of m examples {(x(i), y(i)); i = 1, . . . ,m} and would like to learn the parameters θ_i of this model, write down the log-likelihood, as:

$$
\begin{aligned}
\ell(\theta) &= \sum_{i=1}^{m} \log p(y^{(i)}|x^{(i)};\theta) \\
&= \sum_{i=1}^{m} \log \prod_{l=1}^{k} \left(\frac{e^{\theta_l^T x^{(i)}}}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \right)^{1\{y^{(i)}=l\}}
\end{aligned}
$$

----- XXXX -------