2019-03-20

Nankai-Baidu
Joint Laboratory

Parallel and Distributed
Software Technology Lab

Logistic Regression

参考资料：[Kevin P. Murphy] Machine Learning A Probabilistic Perspective
[Andrew Ng] CS229 Lecture notes 1

Nankai-Baidu
Joint Laboratory

Parallel and Distributed
Software Technology Lab

Supervised Learning

- training example: $(x^{(i)}, y^{(i)})$
- training set: $\{(x^{(i)}, y^{(i)}); i = 1, \dots, m\}$
- Our goal is, given a training set, to learn a function $h(x)$
  so that
  $h(x)$ is a “good” predictor for the corresponding value of y.
- a regression problem: continuous
- a classification problem: When y can take on only a small number of discrete values

Nankai-Baidu
Joint Laboratory

Parallel and Distributed
Software Technology Lab

Linear Regression

- a regression problem

- To perform supervised learning, we must decide how we're going to represent functions $h(x)$ in a computer. Let's say we decide to approximate y as a linear function of x:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots \theta_d x_d, \qquad x = [x_1, x_2, \ldots, x_d]$$

- When there is no risk of confusion, we will write $h_\theta(x)$ more simply as $h(x)$.

- Given a training set, how do we pick, or learn, the parameters θ?

  One reasonable method seems to be to make h(x) close to y,

  at least for the training examples we have.

Nankai-Baidu
Joint Laboratory

Parallel and Distributed
Software Technology Lab

Linear Regression

- Given a training set, how do we pick, or learn, the parameters θ?
  One reasonable method seems to be to make $h(x)$ close to y,
  at least for the training examples we have.

- To formalize this, we will define a function that measures, for each value of the θ's,
  how close the $h\left(x^{(i)}\right)$'s are to the corresponding $y^{(i)}$'s. The cost function:

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

least-squares cost function 最小二乘损失函数

ordinary least squares 普通最小二乘法

Linear Regression

- To formalize this, we will define a function that measures, for each value of the θ's, how close the $h(x^{(i)})$'s are to the corresponding $y^{(i)}$'s. The cost function:

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

- To choose θ so as to minimize $J(\theta)$, 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 $J(\theta)$.

- Gradient Descent: a very natural algorithm that repeatedly takes a step in the direction of steepest decrease of J.
- The normal equations: a second way of doing so.

Linear Regression

- To formalize this, we will define a function that measures, for each value of the θ's, how close the $h\left(x^{(i)}\right)$'s are to the corresponding $y^{(i)}$'s. The cost function:

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

- Gradient Descent: starts with some initial θ, and repeatedly performs the update:

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

The case of only one training example (x, y):

$$\frac{\partial}{\partial\theta_j}J(\theta) = \frac{\partial}{\partial\theta_j}\frac{1}{2}(h_\theta(x) - y)^2 = (h_\theta(x) - y)x_j$$

Nankai-Baidu Joint Laboratory

Parallel and Distributed Software Technology Lab

Linear Regression

- Gradient Descent: starts with some initial θ, and repeatedly performs the update:

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

The case of only one training example (x, y):

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{\partial}{\partial \theta_j} \frac{1}{2} (h_\theta(x) - y)^2 = (h_\theta(x) - y) x_j$$

- For a single training example, this gives the update rule:

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

the LMS update rule (LMS stands for "least mean squares" 最小均方误差)

- Batch GD: This method looks at every example in the entire training set on every step.
- Stochastic GD: We repeatedly run through the training set, and each time we encounter a training example, we update the parameters according to the gradient of the error with respect to that single training example only.

Linear Regression

- To formalize this, we will define a function that measures, for each value of the θ's, how close the $h(x^{(i)})$'s are to the corresponding $y^{(i)}$'s. The cost function:

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

- The normal equations: we will minimize J by explicitly taking its derivatives with respect to the θ_j's, and setting them to zero.

  the value of θ that minimizes J(θ) is given in closed form by the equation

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

  X : $m \times d$ matrix (actually $m \times (d+1)$, if we include the intercept term)

  $\vec{y}$: m – dimensional column vector

Linear Regression

- Why might linear regression, and specifically why might the least-squares cost function J, be a reasonable choice ?

  we will give a set of probabilistic assumptions, under which least-squares regression is derived as a very natural algorithm. **3 Probabilistic interpretation**

- Let us assume that the target variables and the inputs are related via the equation:

$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)},$$

  where $\epsilon^{(i)}$ is an error term that (a) captures either unmodeled effects (隐含变量); and (b) random noise.

- Let us further assume that the $\epsilon^{(i)}$ are distributed independently and identically distributed according to a Gaussian distribution with mean zero and some variance σ^2.

$$\epsilon^{(i)} \sim \mathcal{N}(0, \sigma^2)$$

Linear Regression

- Let us assume that the target variables and the inputs are related via the equation:

$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)},$$

- Let us further assume that the $\epsilon^{(i)}$ are distributed independently and identically distributed according to a Gaussian distribution with mean zero and some variance σ^2.

$$\epsilon^{(i)} \sim \mathcal{N}(0, \sigma^2)$$

- The density of $\epsilon^{(i)}$ is given by

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

This implies that

$$p\left(y^{(i)} | x^{(i)}; \theta\right) = \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2}\right).$$

We can also write the distribution of $y^{(i)}$ as $y^{(i)} | x^{(i)}; \theta \sim \mathcal{N}\left(\theta^T x^{(i)}, \sigma^2\right)$

Linear Regression

- Let us assume that the target variables and the inputs are related via the equation:

$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)},$$

- We can also write the distribution of $y^{(i)}$ as $y^{(i)} | x^{(i)};\ \theta \sim \mathcal{N}\left(\theta^T x^{(i)}, \sigma^2\right)$
- Given X and θ, what is the distribution of the $y^{(i)}$'s ?

  This quantity is typically viewed a function of $\vec{y}$ (and perhaps X), for a fixed value of θ.

  when we wish to explicitly view this as a function of θ, instead call it likelihood function.

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

  Note that by the independence assumption on the $\epsilon^{(i)}$'s (and hence also the $y^{(i)}$'s given the $x^{(i)}$'s), this can also be written

$$L(\theta) = \prod_{i=1}^{m} p\left(y^{(i)} | x^{(i)};\ \theta\right) = \prod_{i=1}^{m} p\left(y^{(i)} | x^{(i)};\ \theta\right) = \frac{1}{\sqrt{2\pi\sigma}} exp\left(-\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2}\right)$$

Nankai-Baidu Joint Laboratory

Parallel and Distributed Software Technology Lab

Linear Regression

- Let us assume that the target variables and the inputs are related via the equation:

$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)},$$

- We can also write the distribution of $y^{(i)}$ as $y^{(i)}|x^{(i)};\ \theta \sim \mathcal{N}\left(\theta^T x^{(i)}, \sigma^2\right)$
- Given X and θ, what is the distribution of the $y^{(i)}$'s ?

what is a reasonable way of choosing our best guess of the parameters θ ?

The principal of maximum likelihood says that we should choose θ so as to make the data as high probability as possible. I.e., we should choose θ to maximize $L(\theta)$.

Note that by the independence assumption on the $\epsilon^{(i)}$'s (and hence also the $y^{(i)}$'s given the $x^{(i)}$'s), this can also be written

$$L(\theta) = \prod_{i=1}^{m} p\left(y^{(i)}|x^{(i)};\ \theta\right) = \prod_{i=1}^{m} p\left(y^{(i)}|x^{(i)};\ \theta\right) = \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2}\right)$$

Linear Regression

- Let us assume that the target variables and the inputs are related via the equation:

$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)},$$

- We can also write the distribution of $y^{(i)}$ as $y^{(i)}|x^{(i)};\ \theta \sim \mathcal{N}\left(\theta^T x^{(i)}, \sigma^2\right)$
- Given X and θ, what is the distribution of the $y^{(i)}$'s ?

在統計學中，最大似然估計(maximum likelihood estimation) 是用來估計一個機率模型的參數的一種方法。這裡的似然函數是指$x^{(i)}$不變時，關於θ的一個函數。最大似然估計不一定存在，也不一定唯一。Instead of maximizing $L(\theta)$, we can also maximize any strictly increasing function of $L(\theta)$. In particular, the derivations will be a bit simpler if we instead maximize the log likelihood $\ell(\theta)$:

$$\ell(\theta) = \log L(\theta) = m \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{\sigma^2} \cdot \frac{1}{2} \sum_{i=1}^{m} \left(y^{(i)} - \theta^T x^{(i)}\right)^2$$

Hence, maximizing $\ell(\theta)$ gives the same answer as minimizing

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

Nankai-Baidu Joint Laboratory

Parallel and Distributed Software Technology Lab

Linear Regression

- Let us assume that the target variables and the inputs are related via the equation:

$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)},$$

- We can also write the distribution of $y^{(i)}$ as $y^{(i)} | x^{(i)};\ \theta \sim \mathcal{N}\left(\theta^T x^{(i)}, \sigma^2\right)$
- Given X and θ, what is the distribution of the $y^{(i)}$'s ?

To summarize: Under the previous probabilistic assumptions on the data, least-squares regression corresponds to finding the maximum likelihood estimate of θ. This is thus one set of assumptions under which least-squares regression can be justified as a very natural method that's just doing maximum likelihood estimation. There are other natural assumptions that can also be used to justify it.

Note also that, in our previous discussion, our final choice of θ did not depend on what was σ^2, and indeed we'd have arrived at the same result even if σ^2 were unknown.

Hence, maximizing $\ell(\theta)$ gives the same answer as minimizing

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

Nankai-Baidu Joint Laboratory

Parallel and Distributed Software Technology Lab

Linear Regression

- 线性回归模型可以推广成分类模型，如二分类模型（逻辑回归模型），不过需要对其做出两个改变：

  第一，要把假设y服从的高斯分布换成适合于二分类模型的伯努利分布，从而可以把模型的输出看作成概率值, 因此适合于使用最大似然估计求参数;

  第二，在计算输入数据的线性组合后，还要将该计算结果代入另一个函数中，该函数需要确保其输出值介于0和1之间。如果该函数是sigmoid函数，此时即为逻辑回归模型。

Bernoulli distribution, 又名0-1分布, 是一個離散型概率分布。若伯努利試驗成功, 則伯努利隨机變量取值為1。若伯努利試驗失敗, 則伯努利隨机變量取值為0。記其成功概率為 p, 失敗概率為 q=1-p。

- 这种通过对假定的参数模型进行建模求参数而从构建概率性分类器的方法就是判别式法。

另一种生成法是通过建模联合分布得到条件分布从而构建出概率性分类器。

Nankai-Baidu Joint Laboratory

Parallel and Distributed Software Technology Lab

Logistic Regression

- a classification problem

- Linear Regression: We could approach the classification problem ignoring the fact that y is discrete-valued, and use linear regression algorithm to try to predict y given x.

- When we know that y$\in \{0, 1\}$, intuitively, it doesn't make sense for $h_\theta(x)$ to take values larger than 1 or smaller than 0.
- To fix this, let's change the form for our hypotheses $h_\theta(x)$. We will choose

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

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

Nankai-Baidu Joint Laboratory

Parallel and Distributed Software Technology Lab

Logistic Regression

- For now, let's take the choice of g as given. Other functions that smoothly increase from 0 to 1 can also be used,
  but for a couple of reasons, the choice of the logistic function is a fairly natural one.
- $g'(z) = g(z)\big(1 - g(z)\big)$

- Let us assume that

$$P(y = +\mathbf{1}|x;\ \theta) = h_\theta(x)\ ;\ P(y = -\mathbf{1}|x;\ \theta) = 1 - h_\theta(x)$$

Note that this can be written more compactly as

$$p(y\,|x;\ \theta) = h_\theta(x)$$

Nankai-Baidu Joint Laboratory

Parallel and Distributed Software Technology Lab

Logistic Regression

- For now, let's take the choice of g as given. Other functions that smoothly increase from 0 to 1 can also be used,

  but for a couple of reasons, the choice of the logistic function is a fairly natural one.
- $g'(z) = g(z)\big(1 - g(z)\big)$

- Let us assume that

$$P(y = \mathbf{1} | x;\ \theta) = h_\theta(x)\ ;\ P(y = \mathbf{0} | x;\ \theta) = 1 - h_\theta(x)$$

  Note that this can be written more compactly as

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

Nankai-Baidu
Joint Laboratory

Parallel and Distributed
Software Technology Lab

Logistic Regression

- So, given the logistic regression model, how do we fit θ for it? Following how we saw least squares regression could be derived as the maximum likelihood estimator under a set of assumptions, let's endow our classification model with a set of probabilistic assumptions, and then fit the parameters via maximum likelihood.
- $g'(z) = g(z)\big(1 - g(z)\big)$
- Let us assume that

$$P(y = \mathbf{1} | x;\ \theta) = h_\theta(x)\ ;\ P(y = \mathbf{0} | x;\ \theta) = 1 - h_\theta(x)$$

Note that this can be written more compactly as

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

Logistic Regression

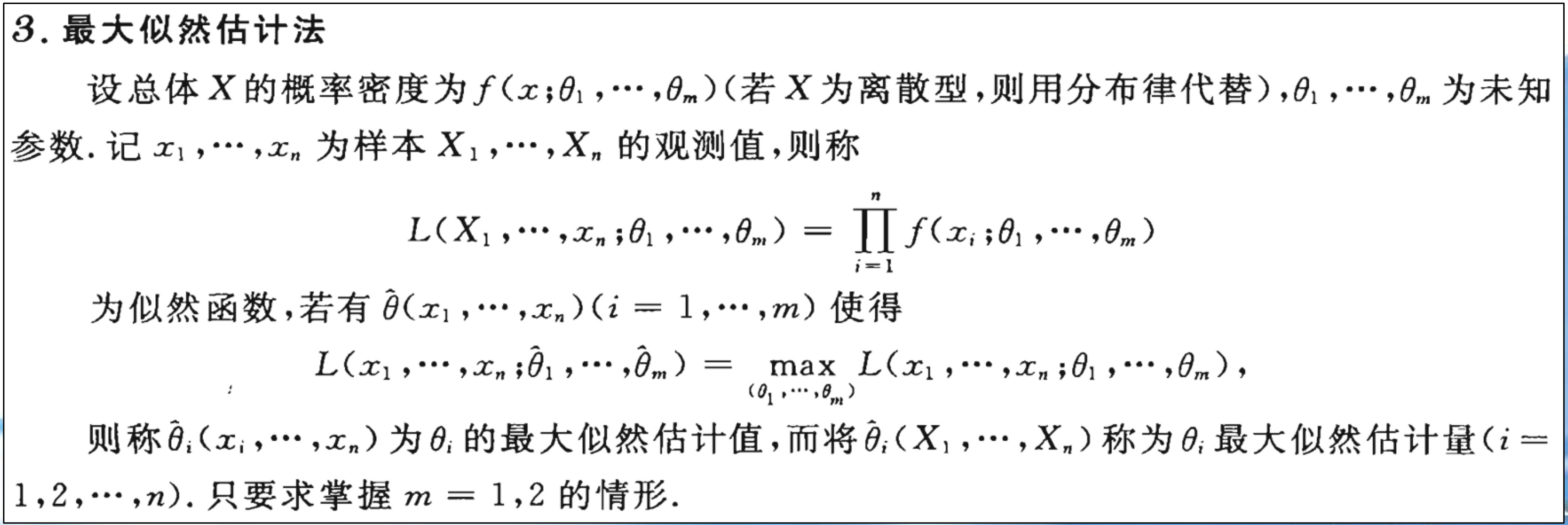
- 最大似然估计：The table below shows how incorrect predictions by our hypothesis function (i.e. h(x)=.25, y=1) are penalized by generation low values.

h(x)	Y=0	Y=1
.25	.75	.25
.5	.5	.5
.75	.25	.75
.9	.1	.9

https://thelaziestprogrammer.com/sharrington/math-of-machine-learning/solving-logreg-newtons-method

Logistic Regression

- 最大似然估计：The table below shows how incorrect predictions by our hypothesis function (i.e. h(x)=.25, y=1) are penalized by generation low values.



3. 最大似然估计法

设总体 X 的概率密度为 $f(x;\theta_1,\cdots,\theta_m)$（若 X 为离散型，则用分布律代替），$\theta_1,\cdots,\theta_m$ 为未知参数. 记 $x_1,\cdots,x_n$ 为样本 $X_1,\cdots,X_n$ 的观测值，则称

$$L(X_1,\cdots,x_n;\theta_1,\cdots,\theta_m)=\prod_{i=1}^{n} f(x_i;\theta_1,\cdots,\theta_m)$$

为似然函数，若有 $\hat{\theta}(x_1,\cdots,x_n)(i=1,\cdots,m)$ 使得

$$L(x_1,\cdots,x_n;\hat{\theta}_1,\cdots,\hat{\theta}_m)=\max_{(\theta_1,\cdots,\theta_m)} L(x_1,\cdots,x_n;\theta_1,\cdots,\theta_m),$$

则称 $\hat{\theta}_i(x_i,\cdots,x_n)$ 为 θ_i 的最大似然估计值，而将 $\hat{\theta}_i(X_1,\cdots,X_n)$ 称为 θ_i 最大似然估计量（$i=1,2,\cdots,n$）. 只要求掌握 $m=1,2$ 的情形.

陈文灯考研数学复习指南(理工类) p625

Logistic Regression

- Let us assume that

$$P(y=1|x;\ \theta) = h_\theta(x)\ ;\ P(y=0|x;\ \theta) = 1 - h_\theta(x)$$

  Note that this can be written more compactly as

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

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

$$L(\theta) = p(\vec{y}|X;\ \theta) = \prod_{i=1}^{m} p\left(y^{(i)}|x^{(i)};\ \theta\right)$$

  It will be easier to maximize the log likelihood:

$$\ell(\theta) = \log L(\theta) = \sum_{i=1}^{m} y^{(i)} \log h\left(x^{(i)}\right) + \left(1-y^{(i)}\right)\log\left(1-h\left(x^{(i)}\right)\right)$$

  How do we maximize the likelihood?

  Similar to our derivation in the case of linear regression, we can use GD.

Nankai-Baidu Joint Laboratory

Parallel and Distributed Software Technology Lab

Logistic Regression

- Let us assume that

$$P(y=1|x;\ \theta) = h_\theta(x)\ ;\ P(y=0|x;\ \theta) = 1 - h_\theta(x)$$

Note that this can be written more compactly as

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

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

$$L(\theta) = p(\vec{y}|X;\ \theta) = \prod_{i=1}^{m} p\left(y^{(i)}|x^{(i)};\ \theta\right)$$

It will be easier to maximize the log likelihood:

$$\ell(\theta) = \log L(\theta) = \sum_{i=1}^{m} y^{(i)} \log h\left(x^{(i)}\right) + \left(1 - y^{(i)}\right)\log\left(1 - h\left(x^{(i)}\right)\right)$$

How do we maximize the likelihood?

求最值的问题：目标函数$\ell(\theta)$ 其海森矩阵负 全局最大值 一阶导数为0.

Logistic Regression

- Let us assume that

$$P(y=1|x;\ \theta) = h_\theta(x)\ ;\ P(y=0|x;\ \theta) = 1 - h_\theta(x)$$

Note that this can be written more compactly as

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

- $\ell(\theta) = \log L(\theta) = \sum_{i=1}^{m} y^{(i)} \log h\left(x^{(i)}\right) + \left(1-y^{(i)}\right)\log\left(1-h\left(x^{(i)}\right)\right)$

  How do we maximize the likelihood? GD

  $\theta := \theta + \alpha \nabla_\theta \ell(\theta)$

  Let's start by working with just one training example (x, y)

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

This gives us the stochastic gradient ascent rule

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

Logistic Regression

- 求最值的问题：目标函数$\ell(\theta)$ 其海森矩阵负 全局最大值 一阶导数为0

梯度下降法(… …)

牛顿法

拟牛顿法

Coordinate ascent

Conjugate gradient ascent

Fixed Hessian Newton method

Iterative Scaling

…