3*

Date
Page

OM

Weighted least Square, Logestics Regression Newton method

★ Probabilistic interpretation

⇒ Let us assume that the target varicbles and the inpits are related via the equction:

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

$\epsilon^{(i)}$ → {Error term that captures Unmodeled effects or random noise}

{$y^{(i)}, x^{(i)}, \epsilon^{(i)}$ are random varicbles}

⇒ Let us further assume that the $\epsilon^{(i)}$ are distributed IID.

IID → (Independet and Identicdly Distributed)

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

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

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

Date
Page

or $Y^{(i)} | X^{(i)}; \theta \sim N(\theta^T x^{(i)}, \sigma^2)$

Let

$$X = \begin{bmatrix} x^{(1)T} \\ x^{(2)T} \\ \vdots \\ x^{(m)T} \end{bmatrix} \qquad Y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}$$

⇒ The probability of the data is given by $P(Y|X;\theta)$

⇒ This quantity is typically viewed as a function of Y for a fixed value of θ.

⇒ When we wish to explicitly view this as a function of θ, we will instead call it the likelihood function:

$$L(\theta) = L(\theta; X, Y) = P(Y|X;\theta)$$

$$\Rightarrow L(\theta) = \prod_{i=1}^{m} P(y^{(i)}|x^{(i)};\theta) \quad \{\varepsilon^{(i)} \text{ are independent}\}$$

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

⇒ We should choose θ to maximize $L(\theta)$.

⇒ Instead of maximizing $L(\theta)$, we can also maximize any strictly increasing function of $L(\theta)$

⇒ Let $\ell(\theta) = \log L(\theta)$ {log likelihood}

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

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

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

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

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

Date
Page

★ Locally weighted linear regression

⇒ It assumes that there are sufficient training data, and makes the choice of features less critical.

⇒ An locally weighted linear regression algorithm does the following:

① Fit θ to minimize $\sum_i w^{(i)} (y^{(i)} - \theta^T x^{(i)})^2$

② Output $\theta^T x$

A fairly standard choice for the weights is

$$w^{(i)} = \exp\left(-\frac{(x^{(i)} - x)^2}{2\tau^2}\right)$$

{Controls how quickly the weights of a training example falls off with distance of its $x^{(i)}$ from the query point x.}

{τ is called the bandwidth parameter}

Date
Page

★ Classification and Logistic regression

⇒ Given $x^{(i)}$, the corresponding $y^{(i)}$ is called the label for the training example.

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

↳ However, it is easy to construct example where this method performs very poorly.

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

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

⇒ Given the logistic regression model how do we fit θ for it?

↳ Lets endow our classification model with a set of probabilistic assumptions, and then fit the parameters via maximum likelihood.

⇒ Let us assume that:

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

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

⇒ Assuming the m training examples were generated independently, we can then write down the likelyhood of the parameter as:

$$L(\theta) = P(Y|X;\theta)$$

$$= \prod_{i=1}^{m} P(y^{(i)}|x^{(i)};\theta)$$

$$= \prod_{i=1}^{m} \left(h_\theta(x^{(i)})\right)^{y^{(i)}} \left(1-h_\theta(x^{(i)})\right)^{1-y^{(i)}}$$

⇒ As before, it will be easier to maximize the log likelyhood:

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

$$= \sum_{i=1}^{m} y^{(i)} \log h(x^{(i)}) + (1-y^{(i)}) \log(1-h(x^{(i)}))$$

⇒ To maximize the log likelyhood we can use gradient ascent.

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

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

{This is a property of much bigger class of problem called generalized linear model}

★ Another algorithm for maximizing $\ell(\theta)$

{Newton's method}

⇒ Let $f: \mathbb{R} \to \mathbb{R}$, and we wish to find a value of θ so that $f(\theta) = 0$. $\forall \theta \in \mathbb{R}$

⇒ Newton's method performs the following update:

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

⇒ This method has a natural interpretation in which we can think of it as approximating the function f via a linear function

- → That is tangent to f at the current guess θ
- → Solving for where that linear function equals to zero
- → And letting the next guess for θ be where that linear function is zero.

⇒ Suppose we wat to maximize some function ℓ.

⇒ Maxima of ℓ corresponds to points where its first derivative $\ell'(\theta)$ is zero.

⇒ So by letting $f(\theta) = l'(\theta)$, we can use the same algorithm to maximize l, and we obtain update rule:

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

⇒ Lastly, in our logistic regression setting, θ is vector-valued, so we need to generalize Newton's method to this setting.

⇒ The generalization of Newtons method to this multidimensional setting is given by

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

⇒ When Newton's method is applied to maximize the logistic regression log likelihood function $l(\theta)$, the resulting method is also called Fisher Scoring.

Let $\nabla_\theta l(\theta) = f(\theta) = 0$ {for maximizing $l(\theta)$}

$$f(\theta + \Delta\theta) \approx f(\theta) + \frac{\delta f}{\delta \theta} \Delta\theta = 0$$

$$= \nabla_\theta l(\theta) + H(\theta)\Delta\theta = 0$$

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

So $\boxed{\theta := \theta - H^{-1}(\theta) \nabla_\theta l(\theta)}$ Hessian matrix

Date
Page

★ The perceptron learning algorithm

⇒ Consider modifying the logistic regression method to "force" it to output values that are either 0 or 1.

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

⇒ If we then let $h_\theta(x) = g(\theta^T x)$ and if we use the update rule:

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

⇒ We have perceptron learning algorithm.

⇒ In the 1960s, this perceptron was argued to be a rough model for how individual neurons in the brain work.

⇒ It is difficult to endow the perceptron's predictions with meaningful probabilistic interpretations, or derived the perceptron as a maximum likelihood estimation algorithm.