# Lecture03

## Outline

1. Linear Regression (recap)
2. Locally weighted regression
3. Probabilistic interpretation
4. Logistic regression
5. Newton’s method

## 1.Recap

$(x^i, y^i) - i^{th} example$

$x^i \in R^{n+1}, y^i \in R$

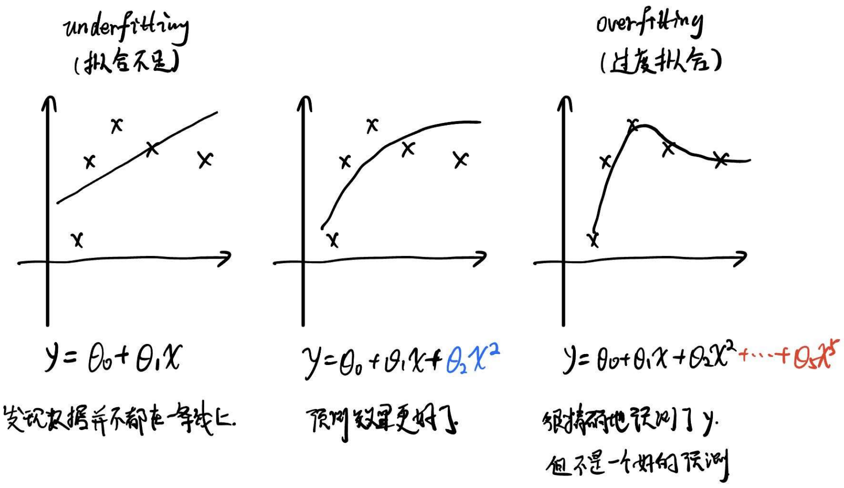
m = #example ， n = #features

$$h_\theta(x) = \sum_{j=0}^{m} \theta_j x_j = \theta^T x$$

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

## 2.Locally weighted linear regression

引出：




结论：

features的选择对于学习算法的表现很重要！

引入LWR（局部加权线性回归），它有充足的训练数据，使得features的选择不用那么纠结。

### Non-parametric algorithm & parametric algorithm

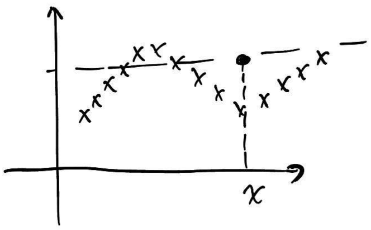
#### Parametric algorithm

有一个固定的参数 θ，一旦确定 θ 并保存起来之后，就不用保留训练集，将来预估新样本的值时只需要用该训练参数 θ 进行预测；

### Non-parametric algorithm

要把整个训练集都存放在内存中，训练一次得到一个参数值，当有新样本时要重新装载训练集进行训练，模型中的参数量和数据集大小成线性关系。（比如LWR）

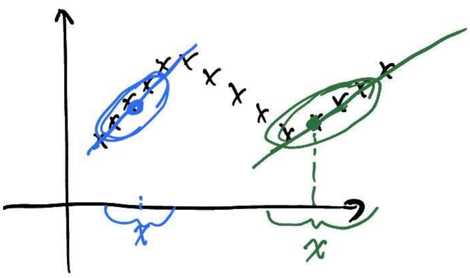
对于指定的点预测：
Linear Regression：




1.拟合θ，最小化： $\sum(y^{(i)}-\theta^T x^{(i)})^2$

2.输出 $\theta^T x$

### LWR




1.拟合θ，最小化： $\sum\omega^{(i)}(y^{(i)}-\theta^T x^{(i)})^2$

2.输出 $\theta^T x$

$\omega^{(i)}$ ：权重

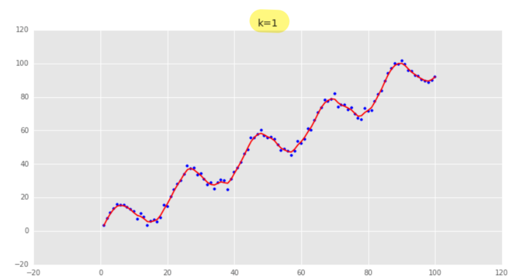
$$\omega^{(i)}=exp(-\frac{(x^{(i)}-x)^2}{2\tau^2})$$

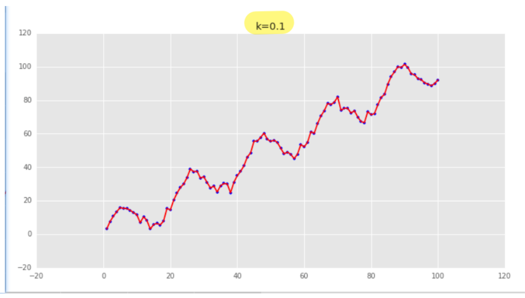
x：the location where you want to make a prediction

$x^{(i)}$ ：the input of x for your $i_{th}$ training example

$\tau$ (bandwidth parameter)：选定x点旁观察数据宽度。

- 当 $\tau$ 大时，会出现拟合不足，预估曲线是平滑的；
- 当 $\tau$ 小时，可能会出现过度拟合，预估曲线会出现锯齿状。




if $\left|x^{(i)}-x\right|$ is small，则 $\omega^{(i)}\approx 1$

if $\left|x^{(i)}-x\right|$ is large，则 $\omega^{(i)}\approx 0$

理解：

当 $(y^{(i)}-\theta^T x^{(i)})$ 很接近的时候，即样本点 $x^{(i)}$ 与x很接近，权重为1，加上这一项。而当很远的时候，权重为0，则不加这一项，所以J(θ)就是对于所有接近的示例的平方误差。

# 3.Probabilistic interpretation

问题引出（Why least squares?）：

对于 $J(\theta)=\frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})^2$ 中，为什么是平方，而不是绝对值?

Assume:

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

$\varepsilon^{(i)}$ ：”error term”，捕获 unmodeled effects and random noise（例如卖家心情好坏对房价有一定影响）

$\varepsilon^{(i)}\sim N(0,\sigma^2)$ （服从正态分布），且独立同分布（I.I.D）

概率密度： $p(\varepsilon^{(i)})=\frac{1}{\sqrt{2\pi}\sigma}exp(-\frac{(\varepsilon^{(i)})^2}{2\sigma^2})$

由 $y^{(i)}=\theta^T x^{(i)}+\varepsilon^{(i)}$ 可知： $\varepsilon^{(i)}=y^{(i)}-\theta^T x^{(i)}$

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

即 $y^{(i)}|x^{(i)};\theta\sim N(\theta^T x^{(i)},\sigma^2)$ ，均值（期望）为 $\theta^T x^{(i)}$ ，方差为 $\sigma^2$

### Likelihood function

（似然函数）

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

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

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

取对数，log likelihood，MLE（Maximum Likelihood Estimation），选取θ可以最大化数据的概率

$$l(\theta)=logL(\theta)$$

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

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

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

则，最大化 $l(\theta)$ 就相当于最小化减去的那一项，即 J(θ)

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

# 4.Logistic regression

## Classification

### Binary Classification

y can take on only two values, 0 and 1.

示例：垃圾邮件过滤器

$x^{(i)}$ ：一封邮件的一些特征

y = 1（positive class），这个邮件是一个垃圾邮件；

y = 0（negative class），这个邮件是其他邮件。

## Logistic regression

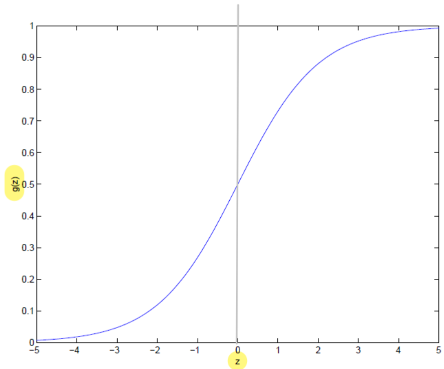
Want： $h_\theta(x) \in [0,1]$ ，（想要 h(x) 输出的值在 0 和 1之间）

对于 线性回归 $h_\theta(x) = \theta^T x$ 输出值可能大于1也可能小于0，为了让它输出值在0和1之间

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

### ”sigmoid” or ”logistic” function

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



对于肿瘤的示例，输入病人features，则告诉我这种肿瘤是恶性(y=1)的几率

Assume:

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

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

$\Rightarrow$

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

MLE：

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

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

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

log likelihood

$$l(\theta) = logL(\theta)$$

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

Choose θ to maximum l(θ)：用 Batch Gradient Descent

$\theta_j := \theta_j + \alpha \frac{\partial}{\partial \theta_j} l(\theta)$ ，这里使用的是 “+”是因为求最大化θ，用梯度上升

假定这里只有一个测试用例（x, y），则

$l(\theta) = ylog(h_\theta(x)) + (1-y)log(1-h_\theta(x))$ ，又 $h_\theta(x) = g(\theta^T x)$

$$l(\theta) = ylog(g(\theta^T x)) + (1-y)log(1-g(\theta^T x))$$

$$\frac{\partial}{\partial \theta_j} l(\theta) = (y\frac{1}{g(\theta^T x)} - (1-y)\frac{1}{1-g(\theta^T x)})\frac{\partial}{\partial \theta_j} g(\theta^T x)$$

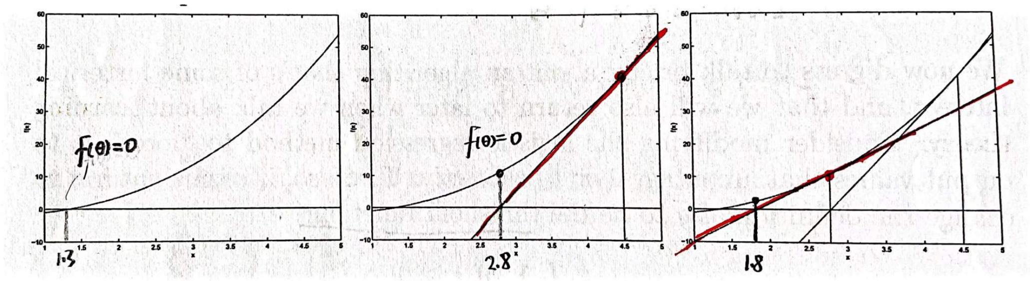
由 sigmoid function $g(z) = \frac{1}{1+e^{-z}}$ ， $g'(z) = g(z)(1-g(z))$

$$\frac{\partial}{\partial \theta_j} l(\theta) = (y\frac{1}{g(\theta^T x)} - (1-y)\frac{1}{1-g(\theta^T x)})g(\theta^T x)(1-g(\theta^T x))\frac{\partial}{\partial \theta_j}\theta^T x$$
$$= (y(1-g(\theta^T x)) - (1-y)g(\theta^T x))x_j$$
$$= (y - h_\theta(x))x_j$$

# 5.Newton’s method

Have f
Want to find θ s.t. f(θ)=0
Want to maximum l(θ)  I.e. want to l’(θ)=0

### 牛顿迭代法图解




在图中找到 θ，使得 f(θ)=0，图1中显示为 θ=1.3

图2，随机初始化一个 θ=4.5，即 $\theta_0 = 4.5$

然后在 f(4.5)这个点作切线，这条切线交 f(θ)=0 于 θ=2.8的位置，即 $\theta_1 = 2.8$ ，

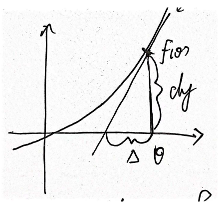
再沿着 f(2.8)这个点作切线，这条切线交 f(θ)=0 于 θ=1.8的位置，即 $\theta_2 = 1.8$

若干次迭代之后，可以快速接近 θ=1.3

### θ 更新规则

$$\theta := \theta - \frac{f(\theta)}{f'(\theta)} \qquad ( \theta^{(t+1)} := \theta^{(t)} - \frac{f(\theta^{(t)})}{f'(\theta^{(t)})} )$$

推导：




由斜率定义： $f'(\theta) = \frac{dy}{dx} = \frac{f(\theta)}{\Delta}$ 有 $\Delta = \frac{f(\theta)}{f'(\theta)}$

### maximum l(θ)

l(θ)最大的点就是它一阶导数为0的点 $l'(\theta) = 0$ ，所以让 $f(\theta) = l'(\theta)$ ，得到：

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

### Quadratic convergence

牛顿迭代法具有“二次收敛”的特性
解释：
在第一次迭代后，距离真实f(θ)=0的距离为 0.01，
则第二次迭代后，距离真实f(θ)=0的距离为 0.0001，
则第三次迭代后，距离真实f(θ)=0的距离为 0.00000001。
这就是为什么牛顿迭代法需要相对较少次数迭代的原因。

### when θ is a vector

update rule: $\theta^{(t+1)} := \theta^{(t)} + H^{-1}\nabla_{\theta} l(\theta)$

$\nabla_{\theta} l(\theta)$

*H,* （Hessian）： $H_{ij} = \dfrac{\partial^2 l(\theta)}{\partial\theta_i \partial\theta_j}$

### 优缺点

优点：比批量梯度下降方法收敛的更快，且只需要更少的迭代次数就可接近min；
缺点：当处理高维问题时，1次迭代的成本很高，因为要转置高维矩阵。