Week 3 笔记

by 骆剑 2017/6/1

1.Logistic Regression with two classification（二类逻辑回归）

Basic Concept（基本概念）：

1.Hypothesis

$$h_\theta(x) = g(\theta^T x)$$

$$z = \theta^T x$$

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

注意：

上面的 x 是小写，是某一个样本数据，这里当作列向量

下面的 X 是大写，是整个样本

2.Cost Function

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$
$$\mathrm{Cost}(h_\theta(x), y) = -\log(h_\theta(x)) \qquad \text{if y = 1}$$
$$\mathrm{Cost}(h_\theta(x), y) = -\log(1 - h_\theta(x)) \qquad \text{if y = 0}$$

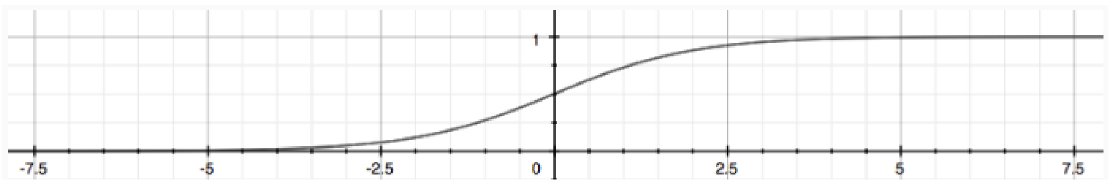
简化后的 Cost Function

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

h 和 J 的向量表示

$$h = g(X\theta)$$
$$J(\theta) = \frac{1}{m}\cdot\left(-y^T\log(h) - (1-y)^T\log(1-h)\right)$$

$$g(z) = \frac{1}{1+e^{-z}}$$ s 型函数，如下所示



h 函数输出为预测为该类的概率，区间为 [0,1]

当 X*theata >= 0 时，即为类别 1

X*theta<0 时，为类别 0

Gradient descent algorithm（梯度下降算法）

$$
\begin{aligned}
&Repeat\ \{\\
&\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)\\
&\}
\end{aligned}
$$

$$
\begin{aligned}
&Repeat\ \{\\
&\theta_j := \theta_j - \frac{\alpha}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}\\
&\}
\end{aligned}
$$

偏导数在格式上与线性回归一致，不同的是 h 函数

梯度下降的向量表示

$$\theta := \theta - \frac{\alpha}{m} X^T (g(X\theta) - \vec{y})$$

2.Logistic Regression with muti-classification

（　多类逻辑回归

➔one-vs-rest 思想

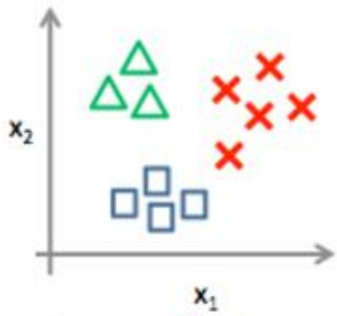
➔多个二类逻辑回归）

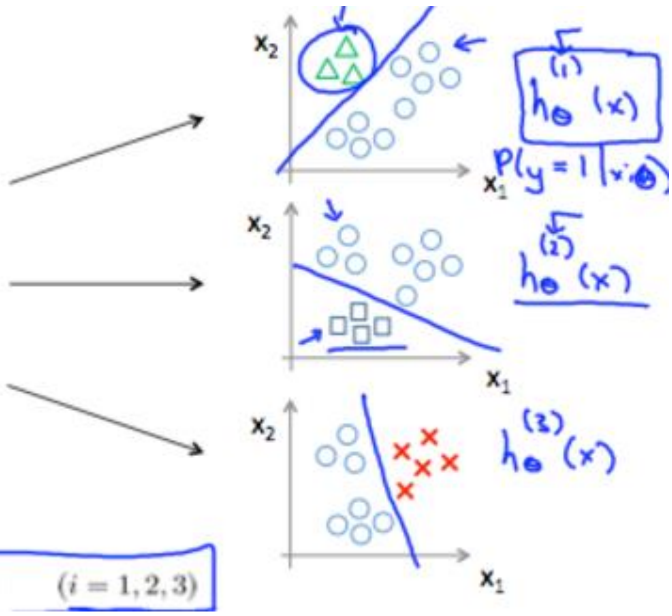
$$
\begin{aligned}
& y \in \{0, 1 \ldots n\} \\
& h_\theta^{(0)}(x) = P(y = 0|x;\theta) \\
& h_\theta^{(1)}(x) = P(y = 1|x;\theta) \\
& \cdots \\
& h_\theta^{(n)}(x) = P(y = n|x;\theta) \\
& \text{prediction} = \max_i(h_\theta^{(i)}(x))
\end{aligned}
$$

若类别是 0 到 n，则就需要 n+1 个 h 函数，预测的时候选择概率值最大的类作为预测输出

思想是 one vs all（one vs rest）

One-vs-all (one-vs-rest):





Class 1: △

Class 2: □

Class 3: ✕

$$\boxed{h_\theta^{(i)}(x) = P(y=i|x;\theta) \qquad (i=1,2,3)}$$

3.使用 Regularization 解决 Overfitting

线性回归：

1.Cost Function

$$\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})^2+\frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

[注]：按照惯例，j 从 1 开始，theta0 不做正则化处理

2.Gradient Descent

$$\begin{aligned}&\text{Repeat } \{\\&\quad \theta_0 := \theta_0 - \alpha\ \frac{1}{m}\ \sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})x_0^{(i)}\\&\quad \theta_j := \theta_j - \alpha\ \left[\left(\frac{1}{m}\ \sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})x_j^{(i)}\right)+\frac{\lambda}{m}\theta_j\right] \qquad\qquad j\in\{1,2...n\}\\&\}\end{aligned}$$

第二项可以改写为

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

3.Normal Equation

$$\theta = \left(X^TX+\lambda\cdot L\right)^{-1}X^Ty$$

$$\text{where }\ L = \begin{bmatrix}0 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1\end{bmatrix}$$

逻辑回归：

1.Cost Function

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\ \log(h_\theta(x^{(i)})) + (1-y^{(i)})\ \log(1-h_\theta(x^{(i)}))] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

[注]：按照惯例，j 从 1 开始，theta0 不做正则化处理

2.Gradient Descent

$$\begin{aligned}
&\text{Repeat } \{\\
&\quad \theta_0 := \theta_0 - \alpha\ \frac{1}{m}\ \sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_0^{(i)}\\
&\quad \theta_j := \theta_j - \alpha\ \left[\left(\frac{1}{m}\ \sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}\right) + \frac{\lambda}{m}\theta_j\right] \qquad\qquad j \in \{1,2...n\}\\
&\}
\end{aligned}$$

第二项可以改写为

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

其中 h 是 s 型函数

4.逻辑回归大作业

(1) cost function 和 grad 的计算代码

非正则化：

```
function [J, grad] = costFunction(theta, X, y)
h = sigmoid(X*theta);
J = 1/m * (-y'*log(h)-(1-y)'*log(1-h));
grad = 1/m * X'*(h-y);
end
```

正则化：

```
function [J, grad] = costFunctionReg(theta, X, y, lambda)
h = sigmoid(X*theta);
theta_remain = theta(2:end);
J = 1/m * (-y'*log(h)-(1-y)'*log(1-h))
  + lambda/(2*m) * (theta_remain'*theta_remain);

theta(1) = 0;
grad = 1/m * X'*(h-y) + lambda/m * theta;
end
```

（2）训练 theta 的代码，fminunc 函数的使用

```
%先定义J和grad
function [J, grad] = costFunction(theta, X, y)
……
end
```

%使用fminunc函数

```
options = optimset('GradObj', 'on', 'MaxIter', 400);
[theta, cost] = ...
    fminunc(@(t)(costFunction(t, X, y)), initial_theta, options);
```