第三讲
Logistic/Softmax 回归

夏 睿
计算机科学与工程学院
南京理工大学
rxia@njust.edu.cn

监督学习

- 回归

Share Price
"$ 24.50"

Continuous Labels
Regression

- 分类

Feature Space $\mathcal{X}$

Label Space $\mathcal{Y}$

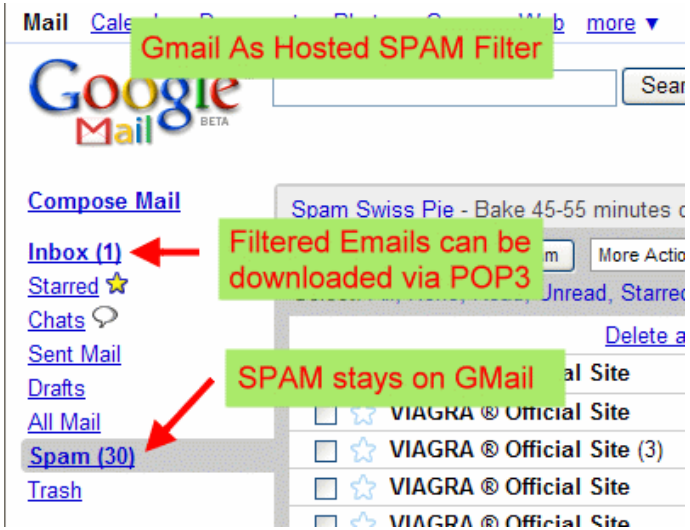
"Sports"
"News"
"Science"
...

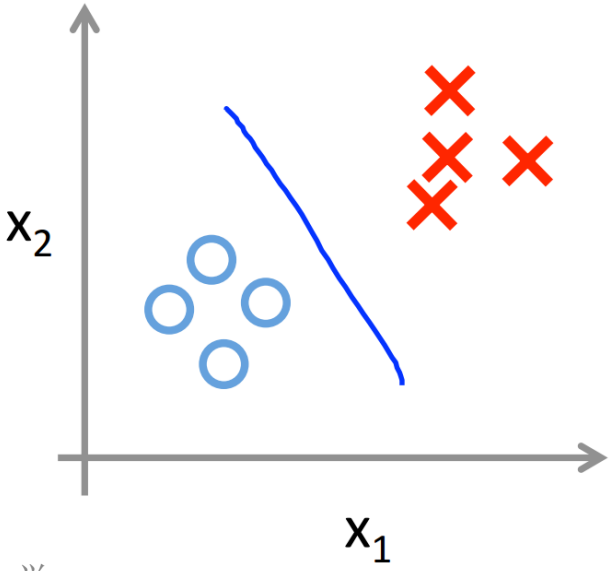
Discrete Labels
Classification

Logistic回归

介绍

- 逻辑回归是一种分类模型，尽管它被称作“回归”；
- 逻辑回归是一种二分类模型;
- 逻辑回归是一种线性分类模型.它有一个线性决策边界（超平面），但用一个非线性激活函数（Sigmoid函数）来模拟后验概率。



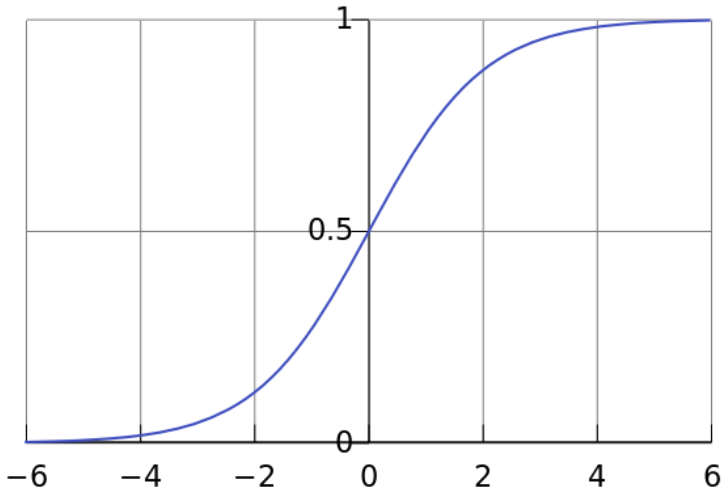


模型假设

- Sigmoid 函数

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

$$\frac{d\delta(z)}{dz} = \delta(z)\,(1-\delta(z))$$



- 假设

$$p(y=1|x;\theta) = h_\theta(x) = \delta(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$$

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

- 假设 (简洁形式)

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

学习算法

- (条件) 似然函数

$$
\begin{aligned}
L(\theta) &= \prod_{i=1}^{N} p\left(y^{(i)} \middle| x^{(i)}; \theta\right) \\
&= \prod_{i=1}^{N} \left(h_\theta\left(x^{(i)}\right)\right)^{y^{(i)}} \left(1 - h_\theta\left(x^{(i)}\right)\right)^{(1-y^{(i)})} \\
&= \prod_{i=1}^{N} \left(\frac{1}{1+e^{-\theta^T x^{(i)}}}\right)^{y^{(i)}} \left(1 - \frac{1}{1+e^{-\theta^T x^{(i)}}}\right)^{(1-y^{(i)})}
\end{aligned}
$$

- 最大似然估计

$$
\max_\theta L(\theta) \Leftrightarrow \max_\theta \sum_{i=1}^{n} y^{(i)} \log h_\theta\left(x^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - h_\theta\left(x^{(i)}\right)\right)
$$

对数似然函数也被称为互熵代价函数

无约束优化

- 无约束最优化问题

$$\max_{\theta} \sum_{i=1}^{n} y^{(i)} \log h_\theta\left(x^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - h_\theta\left(x^{(i)}\right)\right)$$

- 优化方法
  - 梯度下降
  - 随机梯度下降
  - 牛顿法
  - 拟牛顿法
  - 共轭梯度法
  - …

梯度下降/上升

- 梯度计算

$$\frac{dl(\theta)}{d\theta} = \sum_{i=1}^{N}\left(y^{(i)}\frac{1}{h_\theta(x^{(i)})} - (1-y^{(i)})\frac{1}{1-h_\theta(x^{(i)})}\right)\frac{\partial}{\partial\theta}h_\theta(x^{(i)})$$

$$= \sum_{i=1}^{N}\left(y^{(i)}\frac{1}{h_\theta(x^{(i)})} - (1-y^{(i)})\frac{1}{1-h_\theta(x^{(i)})}\right)h_\theta(x^{(i)})\left(1-h_\theta(x^{(i)})\right)\frac{\partial}{\partial\theta}\theta^T x^{(i)}$$

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

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

误差 × 特征

- 梯度上升优化

$$\theta := \theta + \alpha\sum_{i=1}^{N}\left(y^{(i)} - h_\theta(x^{(i)})\right)x^{(i)}$$

随机梯度下降

- 随机选择一个训练样本

$$(x, y)$$

- 计算梯度

$$(y - h_\theta(x))x$$

- 更新权重

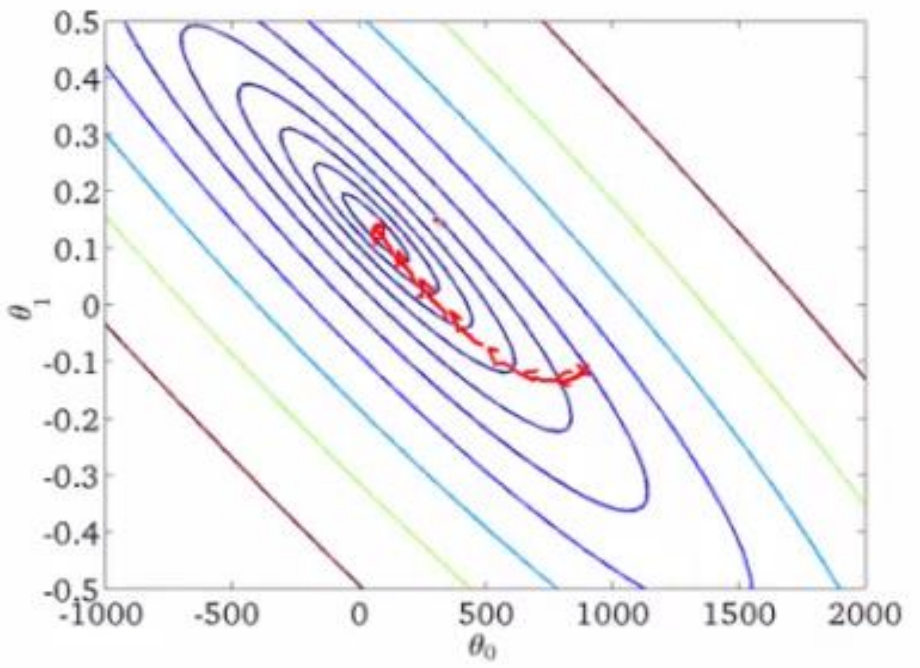
$$\theta := \theta + \alpha(y - h_\theta(x))x$$

- 重复...

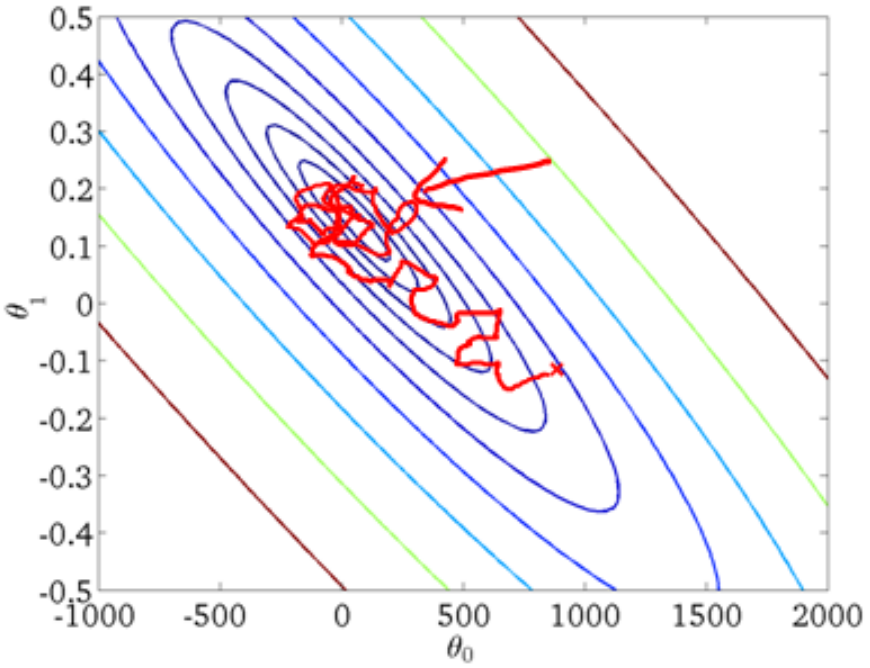
梯度下降– 批次更新

随机梯度下降– 在线更新

GD vs. SGD



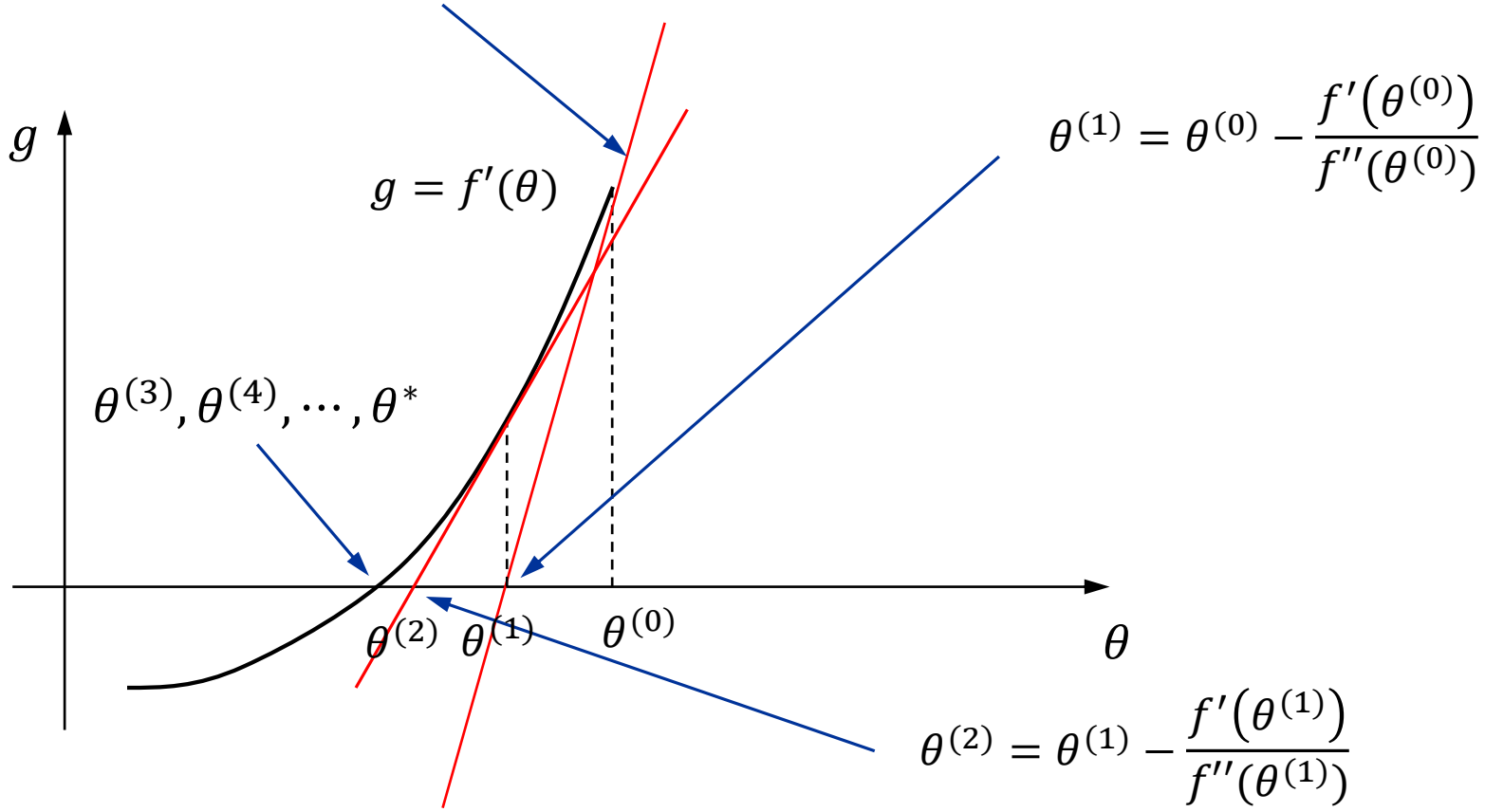
梯度下降(GD)



随机梯度下降(SGD)

牛顿法图解

切线: $g = f'(\theta_0) + f''(\theta_0)(\theta - \theta_0)$



牛顿法

- 问题

$$\arg\min f(\theta) \Leftrightarrow solve: \nabla f(\theta) = 0$$

- 二阶泰勒展开式

$$\phi(\theta) = f\left(\theta^{(k)}\right) + \nabla f\left(\theta^{(k)}\right)\left(\theta - \theta^{(k)}\right) + \frac{1}{2}\nabla^2 f(\theta^{(k)})\left(\theta - \theta^{(k)}\right)^2 \approx f(\theta)$$

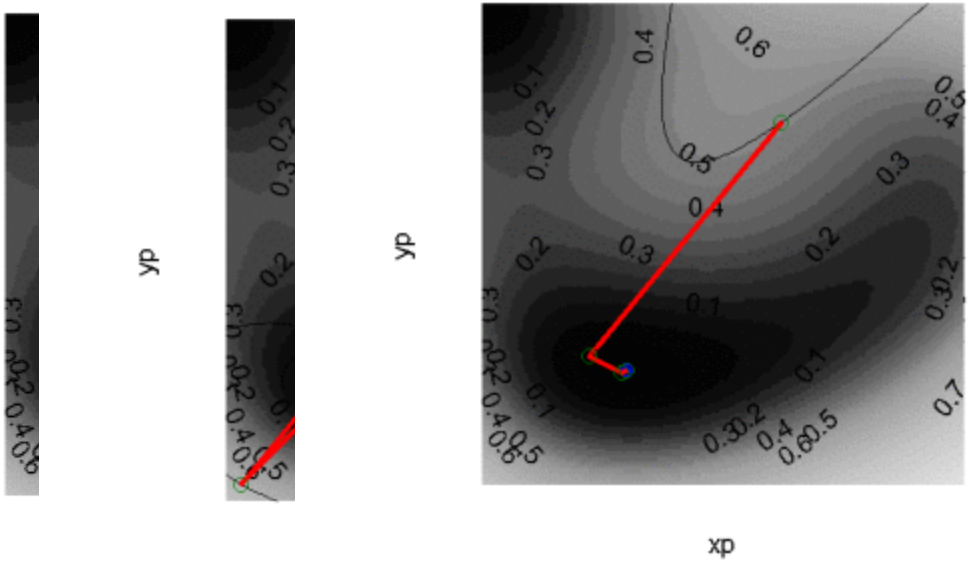
$$\nabla\phi(\theta) = 0 \Rightarrow \theta = \theta^{(k)} - \nabla^2 f(\theta^{(k)})^{-1}\nabla f(\theta^{(k)})$$

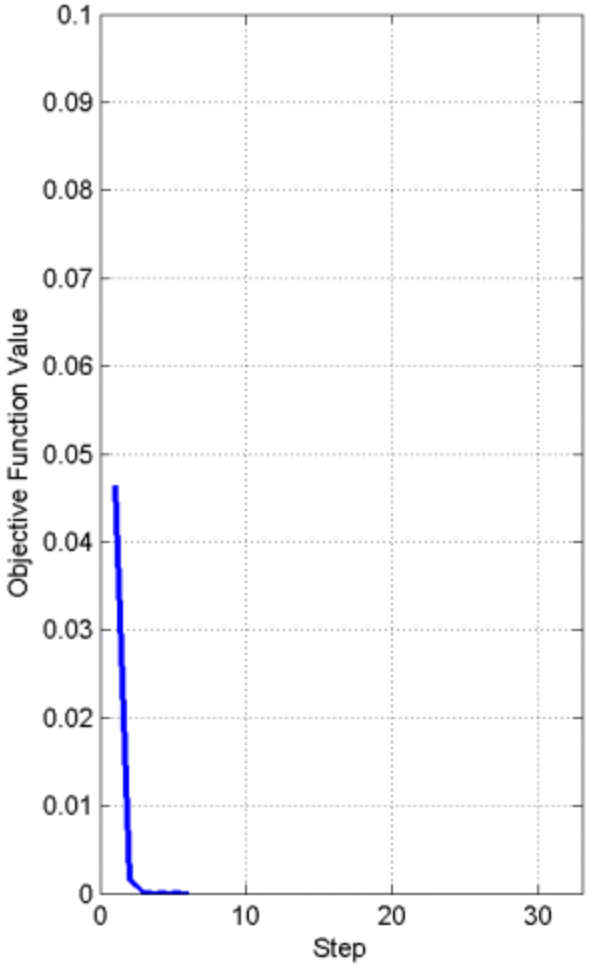
- 牛顿法（又称牛顿-拉夫逊法）

$$\theta^{(k+1)} = \theta^{(k)} - \boxed{\nabla^2 f(\theta^{(k)})^{-1}}\nabla f(\theta^{(k)})$$

Hessian Matrix

梯度法 vs. 牛顿法


Damped Newton
yp
xp


Objective Function Value
Step
0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0
10
20
30

牛顿的逻辑回归方法

- 最优化问题

$$\arg\min \frac{1}{N}\sum_{i=1}^{N} -y^{(i)}\ \log h_\theta\left(x^{(i)}\right) - \left(1-y^{(i)}\right)\log\left(1-h_\theta\left(x^{(i)}\right)\right)$$

- 最优化问题

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

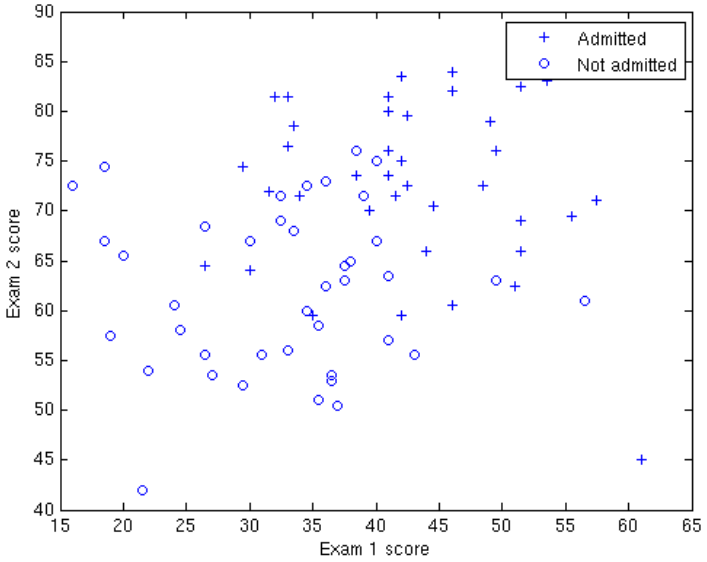
$$H = \frac{1}{N}\sum_{i=1}^{N} h_\theta\left(x^{(i)}\right)^{\mathrm{T}}\left(1-h_\theta\left(x^{(i)}\right)\right)x^{(i)}(x^{(i)})^{\mathrm{T}}$$

- 使用牛顿法更新权重

$$\theta^{(t+1)} = \theta^{(t)} - H^{-1}\nabla J(\theta^{(t)})$$

练习: 逻辑回归

- 根据下面的训练数据:



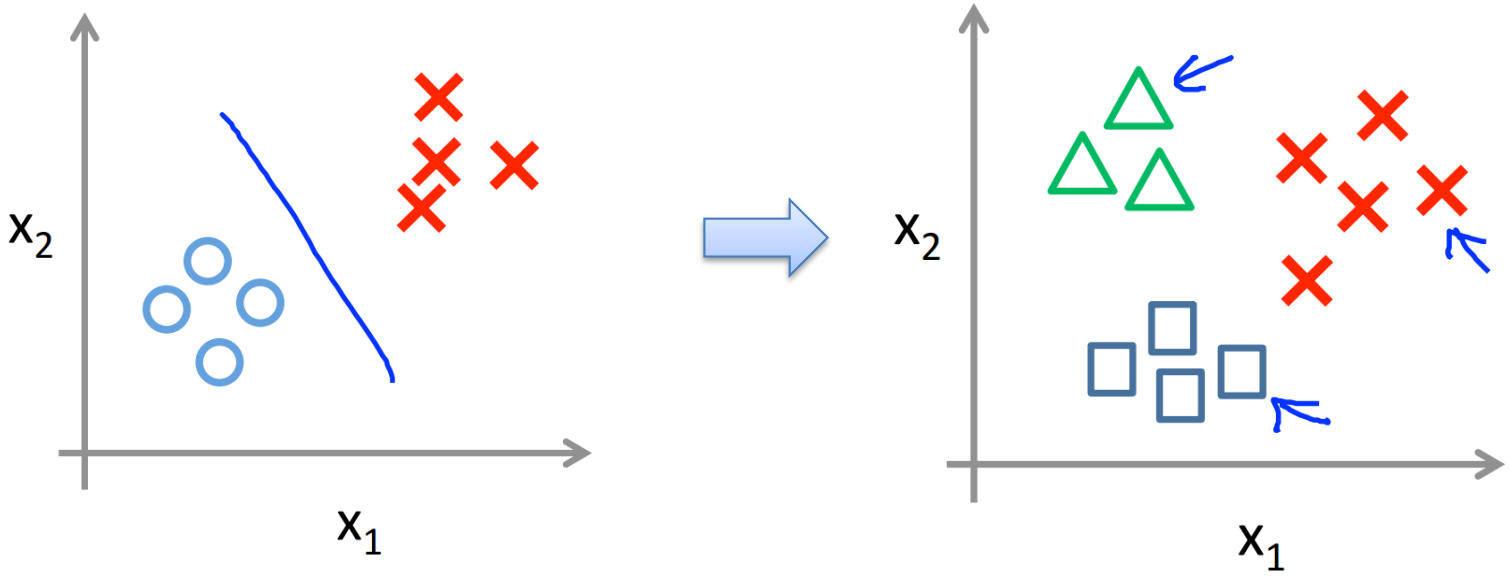
http://openclassroom.stanford.edu/MainFolder/DocumentPage.php?course=DeepLearning&doc=exercises/ex4/ex4.html

- 实现 1) 梯度下降; 2) 随机梯度下降; 3) 逻辑回归的牛顿法, 设置初始参数 $\theta = 0$.
- 决定迭代的次数,计算每次迭代并绘制结果.

Softmax 回归

Softmax 回归

- Softmax 回归是一种多分类模型, 也叫做多分类逻辑回归;
- 在 NLP中, 也被称做最大熵模型;
- 它是一种经常使用的分类算法.



模型描述

- 模型假设

$$p(y=j|x;\theta)=h_j(x)=\frac{e^{{\theta_j}^{\mathrm{T}}x}}{1+\sum_{j'=1}^{C-1}e^{{\theta_{j'}}^{\mathrm{T}}x}},j=1,\dots,C-1$$

$$p(y=C|x;\theta)=h_C(x)=\frac{1}{1+\sum_{j'=1}^{C-1}\exp\{{\theta_{j'}}^{\mathrm{T}}x\}}$$

- 模型假设(简洁形式)

$$p(y=j|x;\theta)=h_j(x)=\frac{e^{{\theta_j}^{\mathrm{T}}x}}{\sum_{j'=1}^{C}e^{{\theta_{j'}}^{\mathrm{T}}x}},j=1,2,\dots,C,\text{where }\theta_C=\vec{0}$$

- 参数

$$\theta_{C\times M}$$

最大似然估计

- (有条件的) 对数似然函数

Softmax Regression

$$l(\theta) = \sum_{i=1}^{N} \log p(y^{(i)}|x^{(i)};\theta)$$

$$= \sum_{i=1}^{N} \log \prod_{j=1}^{C} \left(\frac{e^{\theta_j{}^{\mathrm{T}} x}}{\sum_{j'=1}^{C} e^{\theta_{j'}{}^{\mathrm{T}} x}} \right)^{1\{y^{(i)}=j\}}$$

$$= \sum_{i=1}^{N} \sum_{j=1}^{C} 1\{y^{(i)} = j\} \log \left(\frac{e^{\theta_j{}^{\mathrm{T}} x}}{\sum_{j'=1}^{C} e^{\theta_{j'}{}^{\mathrm{T}} x}} \right)$$

$$= \sum_{i=1}^{N} \sum_{j=1}^{C} 1\{y^{(i)} = j\} \log h_j(x^{(i)})$$

Logistic Regression

$$l(\theta) = \sum_{i=1}^{N} y^{(i)} \log h_\theta\left(x^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - h_\theta\left(x^{(i)}\right)\right)$$

梯度下降优化

- 梯度

$$\frac{\partial \log\, h_j(x)}{\partial \theta_k} = \begin{cases} \big(1 - h_k(x)\big)x, & j = k \\ -h_k(x)x, & j \neq k \end{cases}$$

$$\frac{\partial \sum_{j=1}^{C} 1\{y = j\} \log h_j(x)}{\partial \theta_k} = \begin{cases} \big(1 - h_k(x)\big)x, & y = k \\ -h_k(x)x, & y \neq k \end{cases}$$

$$= \big(1\{y = k\} - h_k(x)\big)x$$

$$\frac{\partial l(\theta)}{\partial \theta_k} = \sum_{i=1}^{N} \boxed{\big(1\{y^{(i)} = k\} - h_k(x^{(i)})\big)\, x^{(i)}}$$

误差 × 特征

梯度下降优化

- 梯度下降

$$\theta_k := \theta_k + \alpha \sum_{i=1}^{N} \left(1\{y^{(i)} = k\} - h_k(x^{(i)})\right) x^{(i)}$$

$$\text{where } h_k(x) = \frac{e^{\theta_k{}^{\mathrm{T}} x}}{\sum_{k'=1}^{C} e^{\theta_{k'}{}^{\mathrm{T}} x}}, k = 1,2,\ldots,C$$

- 随机梯度下降

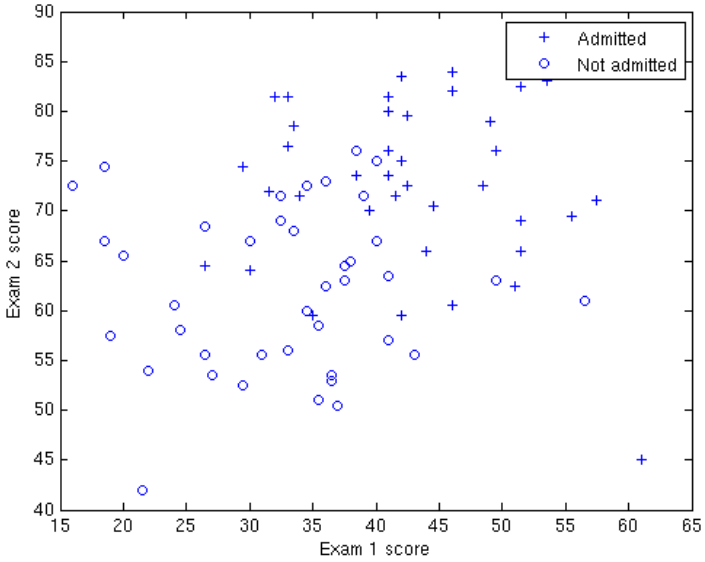
$$\theta_k := \theta_k + \alpha\left(1\{y = k\} - h_k(x)\right) x$$

其他优化方法

- 牛顿法
- 拟牛顿法(BFGS)
- 有限记忆拟牛顿法(L-BFGS)
- 共轭梯度法
- GIS
- IIS
- …

练习: Softmax回归

- 给出下列训练数据:



http://openclassroom.stanford.edu/MainFolder/DocumentPage.php?course=DeepLearning&doc=exercises/ex4/ex4.html

- 实现逻辑回归：1) GD; 2) SGD.
- 实现softmax回归：1) GD; 2) SGD.
- 比较逻辑回归和softmax回归.

欢迎提问！