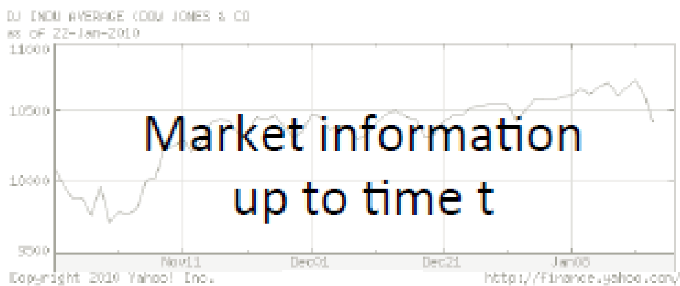
NUSTM

Lecture 3
Logistic Regression & Softmax Regression

Rui Xia
Text **M**ining Group
Nanjing **U**niversity of **S**cience & **T**echnology
rxia@njust.edu.cn

Supervised Learning

- Regression



Market information up to time t

Share Price
"$ 24.50"

Continuous Labels
Regression

- Classification

Feature Space $\mathcal{X}$

Words in a document

Label Space $\mathcal{Y}$

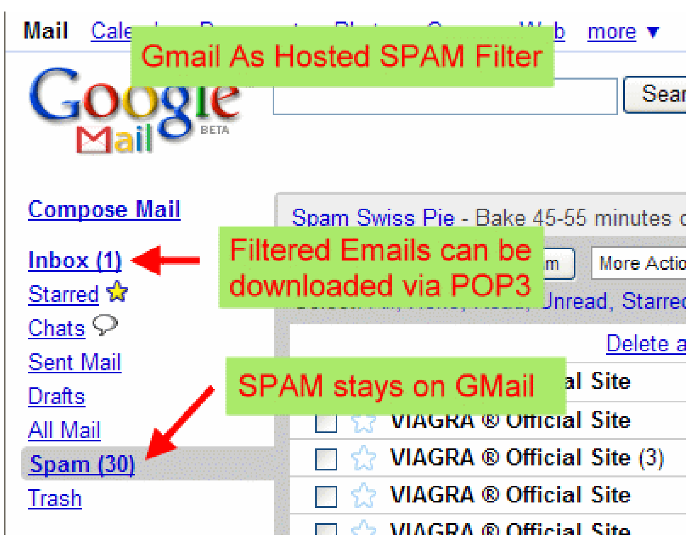
"Sports"
"News"
"Science"
...

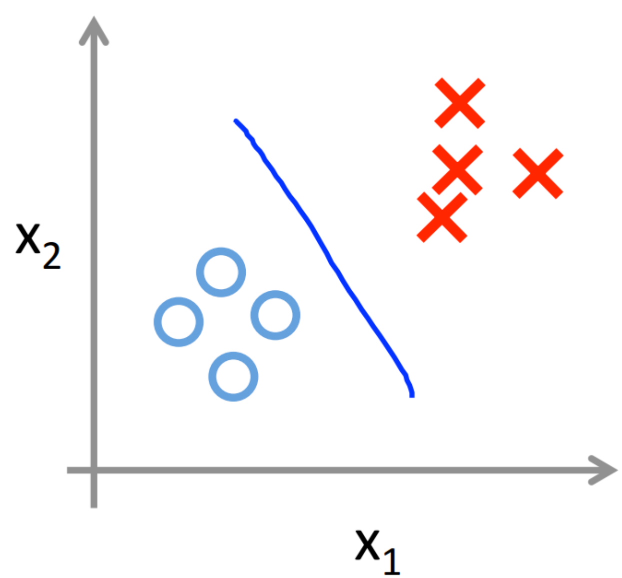
Discrete Labels
Classification

Logistic Regression

Introduction

- Logistic Regression is a classification model, although it is called "regression";
- Logistic regression is a binary classification model;
- Logistic regression is a linear classification model. It has a linear decision boundary (hyperplane), but with a nonlinear activation function (Sigmoid function) to model the posterior probability.



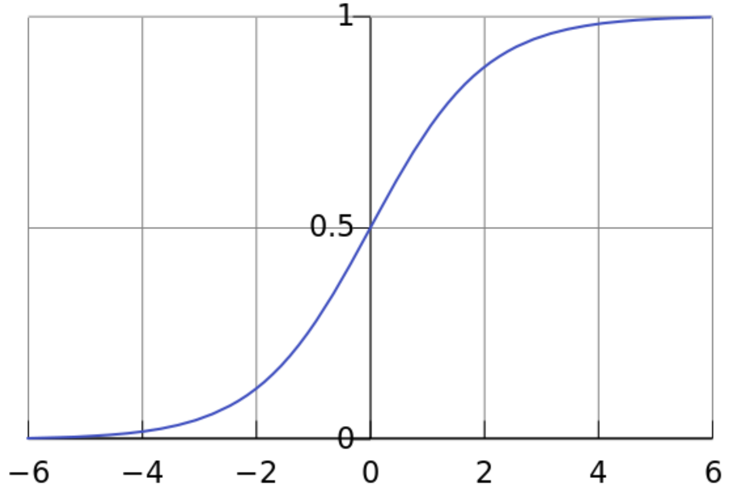


Model Hypothesis

- Sigmoid Function

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

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



- Hypothesis

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

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

- Hypothesis (Compact Form)

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

Learning Algorithm

- (Conditional) Likelihood Function

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

- Maximum Likelihood Estimation

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

The neg log-likelihood function is also known as the **Cross-Entropy** cost function

NUSTM

Unconstraint Optimization

- Unconstraint Optimization Problem

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

- Optimization Methods
  - Gradient Descent
  - Stochastic Gradient Descent
  - Newton Method
  - Quasi-Newton Method
  - Conjugate Gradient
  - …

Gradient Descent/Ascent

- Gradient Computation

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

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

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

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

Error × Feature

- Gradient Ascent Optimization

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

NUSTM

Stochastic Gradient Descent

- Randomly choose a training sample

$$(x, y)$$

- Compute gradient

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

- Updating weights

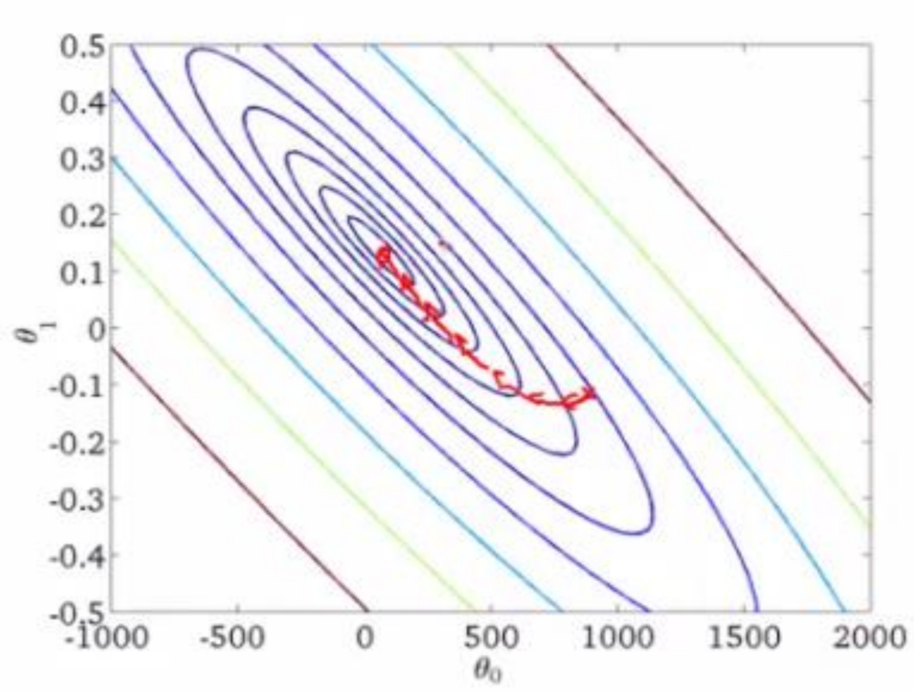
$$\theta := \theta + \alpha(y - h_\theta(x))x$$

- Repeat…

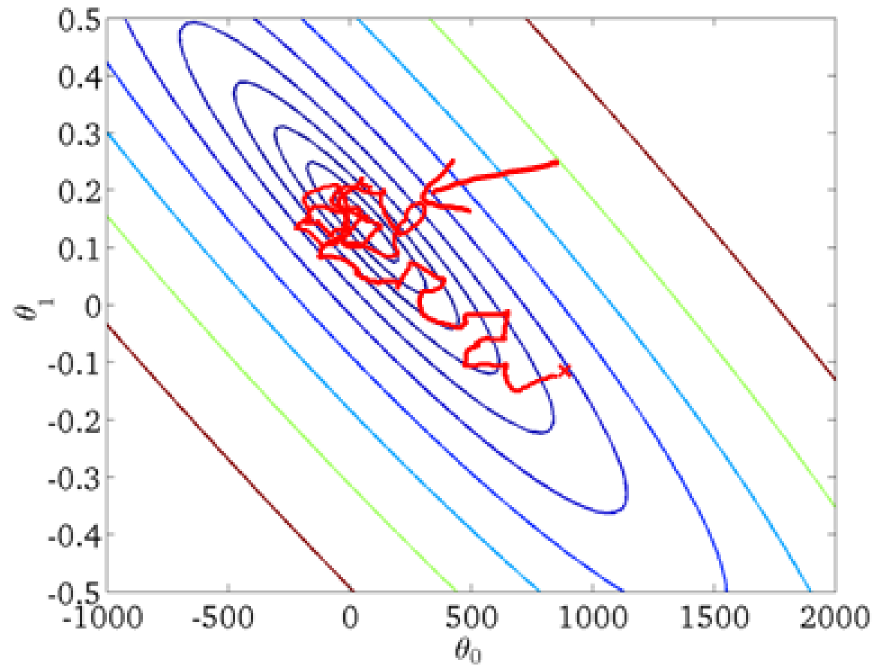
Gradient descent -- batch updating

Stochastic gradient descent -- online updating

GD vs. SGD



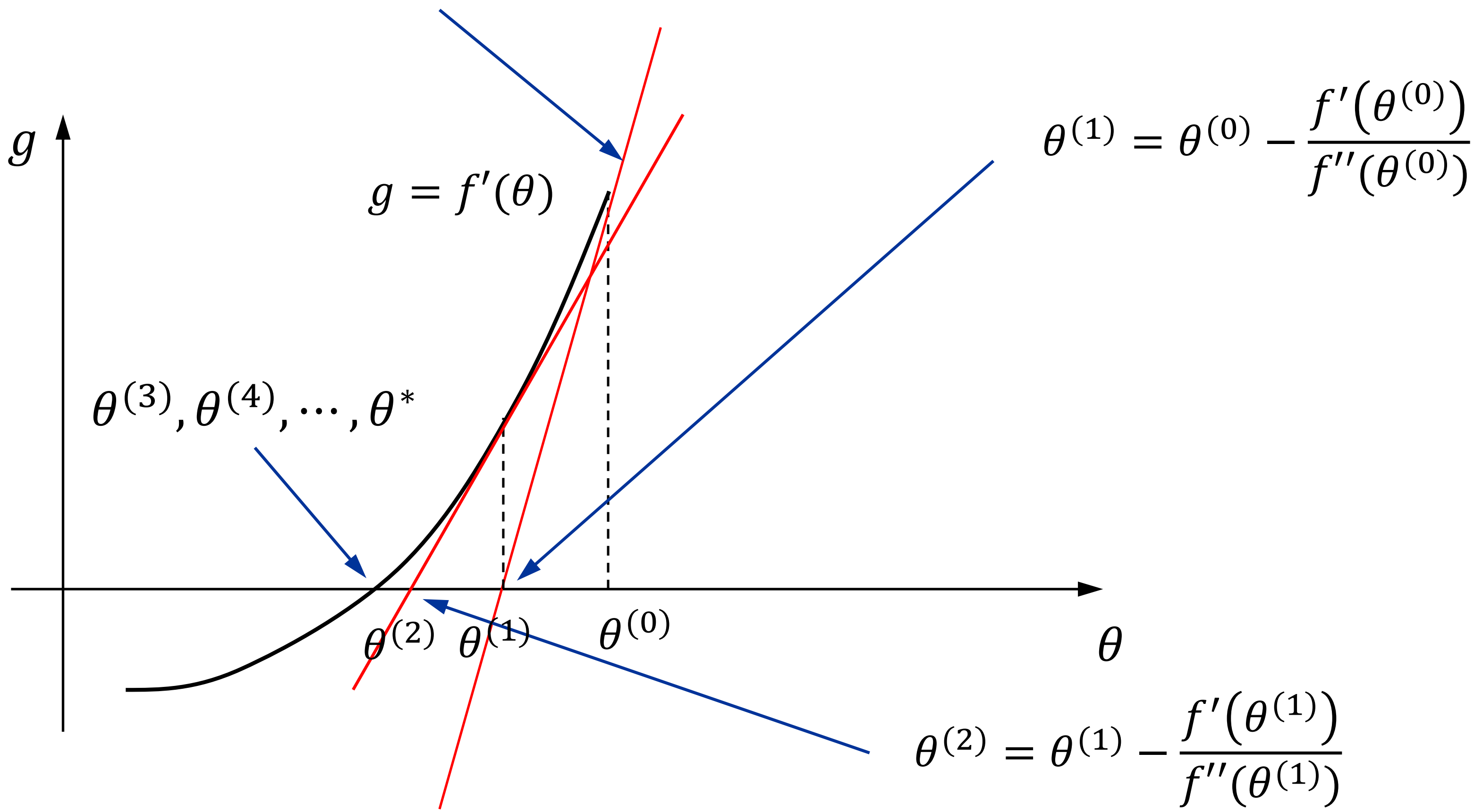
Gradient Descent (GD)



Stochastic Gradient Descent (SGD)

Illustration of Newton's Method

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



Newton's Method

- Problem

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

- Second-order Taylor expansion

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

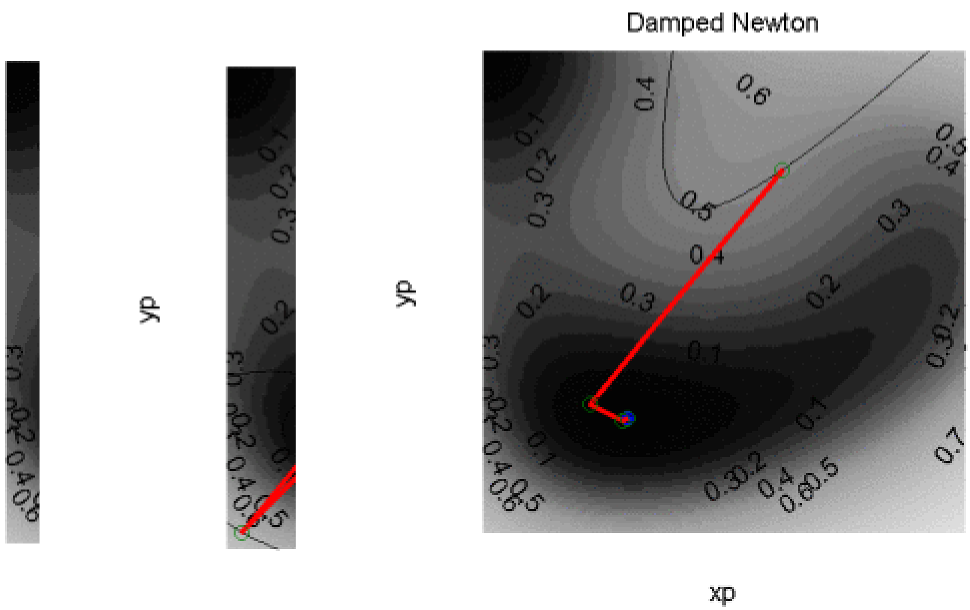
$$\nabla\phi(\theta) = 0 \Rightarrow \theta = \theta^{(k)} - \nabla^2 f(\theta^{(k)})^{-1}\nabla f(\theta^{(k)})$$

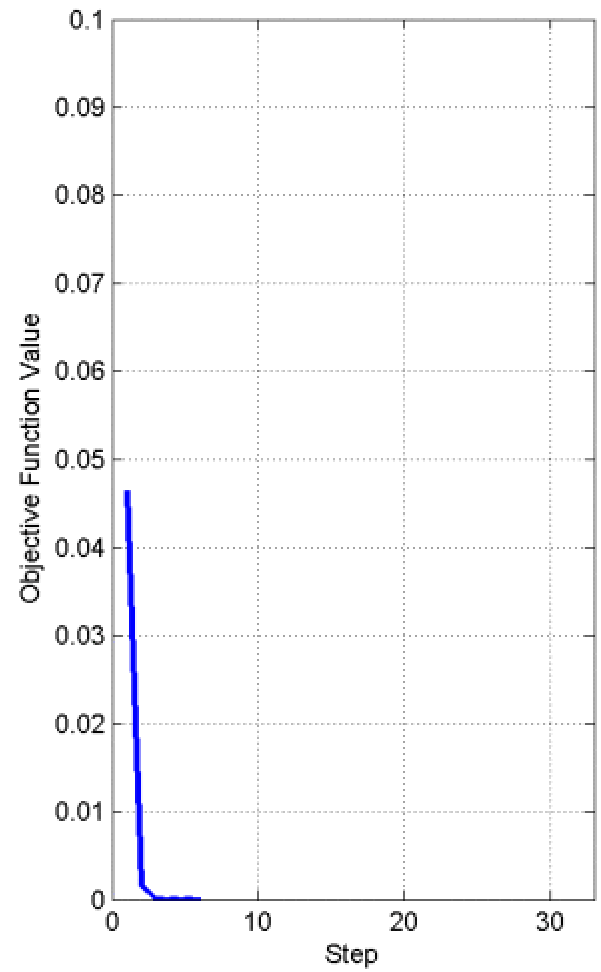
- Newton's method (also called Newton-Raphson method)

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

Hessian Matrix

Gradient' vs. Newton's Method





Newton's Method for Logistic Regression

- Optimization Problem

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

- Gradient and Hessian Matrix

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

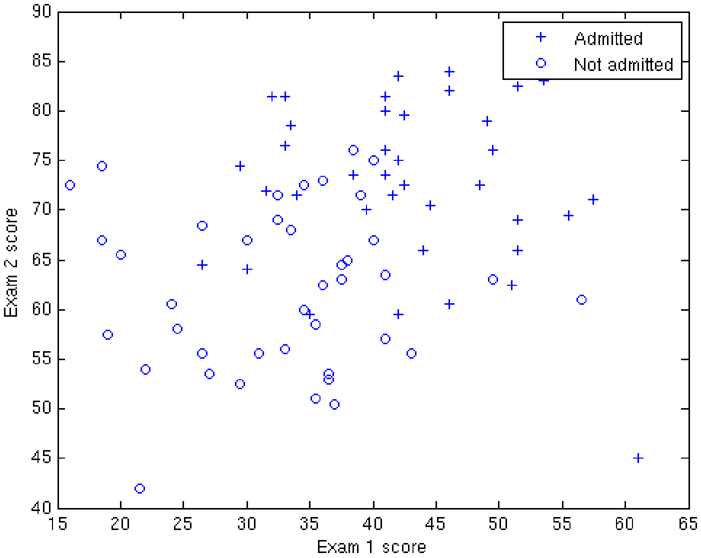
$$H = \frac{1}{N}\sum_{i=1}^{N} h_\theta\left(x^{(i)}\right)^{\mathrm{T}}\left(1-h_\theta\left(x^{(i)}\right)\right)x^{(i)}(x^{(i)})^{\mathrm{T}}$$

- Weight updating using Newton's method

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

Practice: Logistic Regression

- Given the following training data:



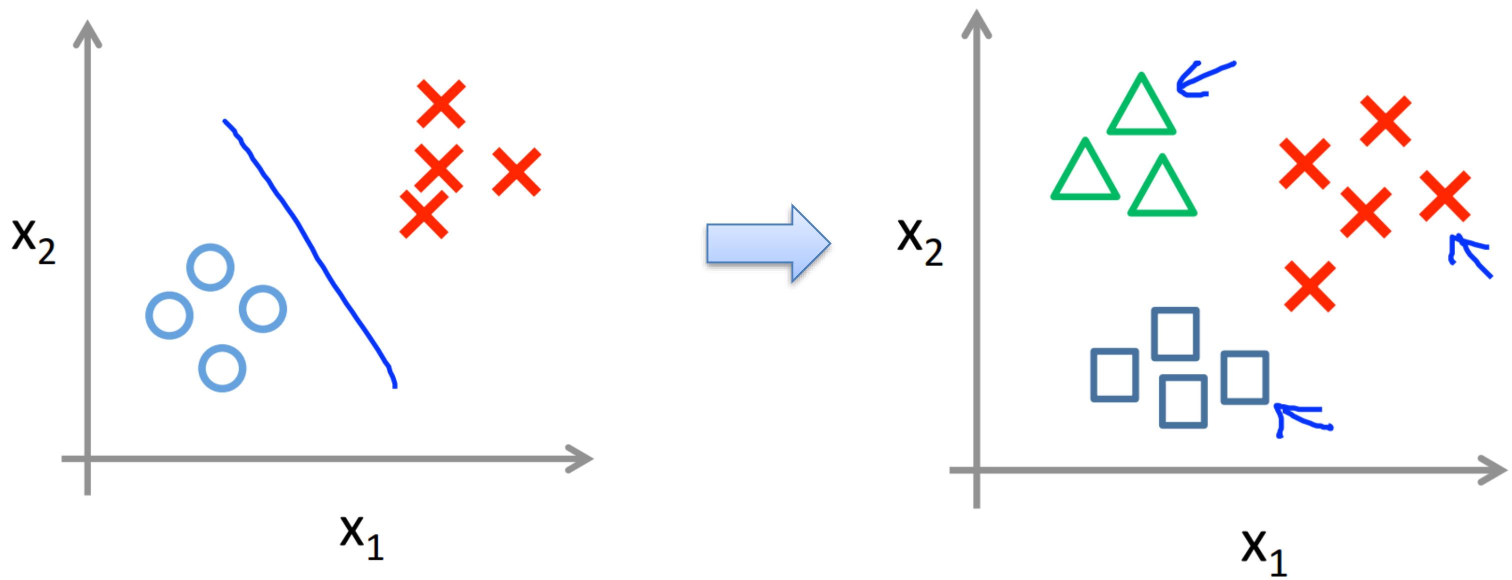
http://openclassroom.stanford.edu/MainFolder/DocumentPage.php?course=DeepLearning&doc=exercises/ex4/ex4.html

- Implement 1) GD; 2) SGD; 3) Newton's Method for logistic regression, starting with the initial parameter \theta=0.
- Determine how many iterations to use, and calculate for each iteration and plot your results.

Softmax Regression

Softmax Regression

- Softmax Regression is a multi-class classification model, also called Multi-class Logistic Regression;
- It is also known as the Maximum Entropy Model (in NLP);
- It is one of the most used classification algorithms.



Model Description

- Model Hypothesis

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

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

- Model Hypothesis (Compact Form)

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

- Parameters

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

Maximum Likelihood Estimation

- (Conditional) Log-likelihood

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

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

NUSTM

Gradient Descent Optimization

- Gradient

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

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

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

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

Error × Feature

NUSTM

Gradient Descent Optimization

- Gradient Descent

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

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

- Stochastic Gradient Descent

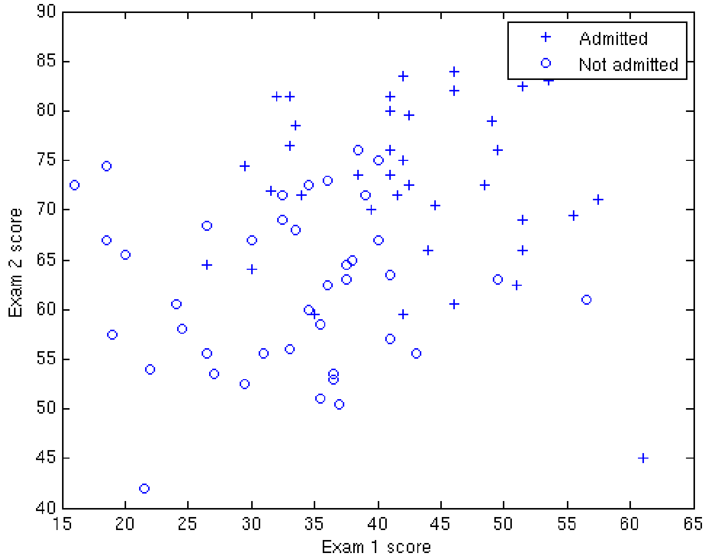
$$\theta_k := \theta_k + \alpha \left(1\{y = k\} - h_k(x)\right) x$$

The other optimization methods

- Newton Method
- Quasi-Newton Method (BFGS)
- Limited Memory BFGS (L-BFGS)
- Conjugate Gradient
- GIS
- IIS
- …

Practice: Softmax Regression

- Given the following training data:



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

- Implement logistic regression with 1) GD; 2) SGD.
- Implement softmax regression with 1) GD; 2) SGD.
- Compare logisitic regression and softmax regression.

Questions?