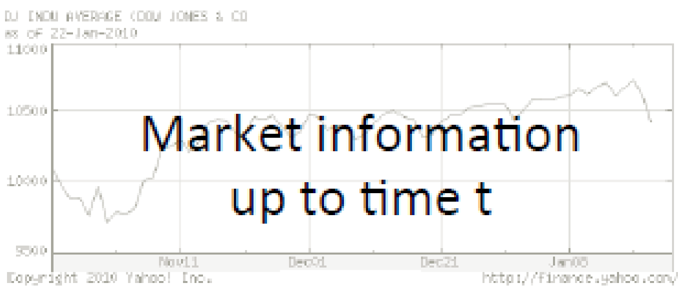
Lecture 3
Logistic Regression & Softmax Regression

Rui Xia
School of Computer Science & Engineering
Nanjing University of Science & Technology
http://www.nustm.cn/~rxia

Supervised Learning

- Regression

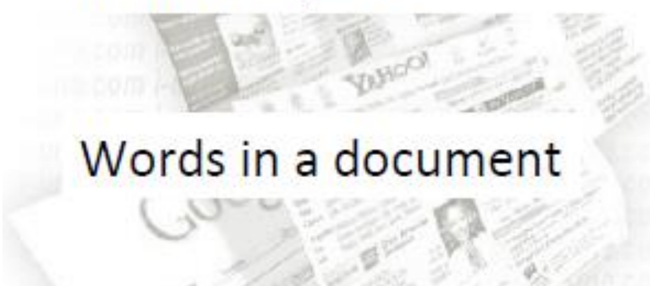


Share Price
"$ 24.50"

Continuous Labels
Regression

- Classification

Feature Space $\mathcal{X}$



Label Space $\mathcal{Y}$

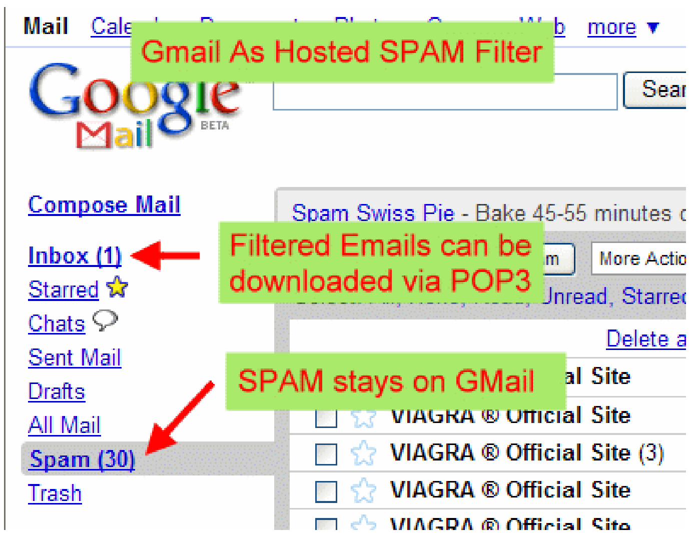
"Sports"
"News"
"Science"
…

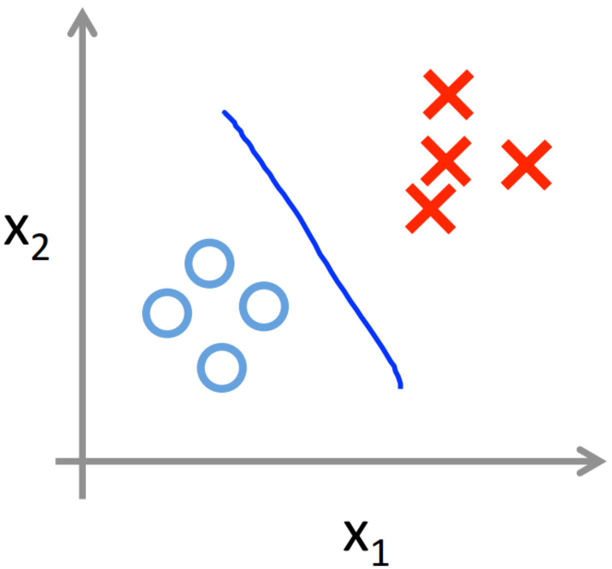
Discrete Labels
Classification

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.



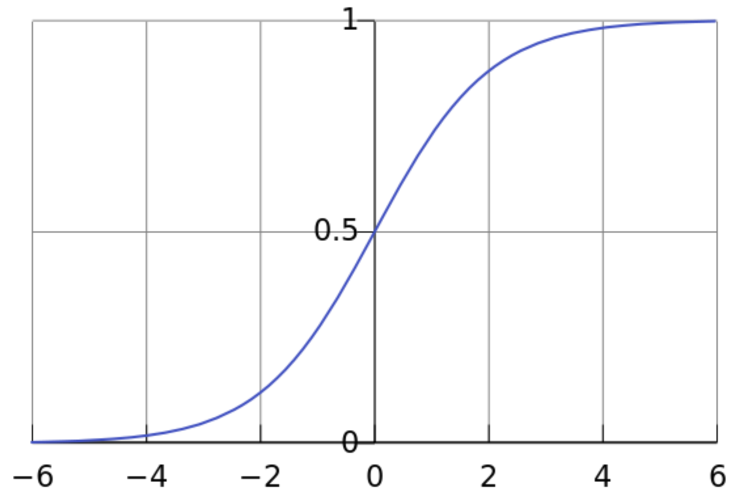


Model Hypothesis

- Sigmoid Function

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

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



- Hypothesis

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

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

- Hypothesis (Compact Form)

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

Learning Algorithm

- (Conditional) Likelihood Function

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

- Maximum Likelihood Estimation

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

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

Unconstraint Optimization

- Unconstraint Optimization Problem

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

- Optimization Methods
  – Gradient Descent
  – Stochastic Gradient Descent
  – Newton Method
  – Quasi-Newton Method
  – Conjugate Gradient
  – ...

Gradient Descent/Ascent

- Gradient Computation

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

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

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

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

Error × Feature

- Gradient Ascent Optimization

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

Stochastic Gradient Descent

- Randomly choose a training sample

$$(\boldsymbol{x}, y)$$

- Compute gradient

$$\big(y - h_{\boldsymbol{\theta}}(\boldsymbol{x})\big)\boldsymbol{x}$$

- Updating weights

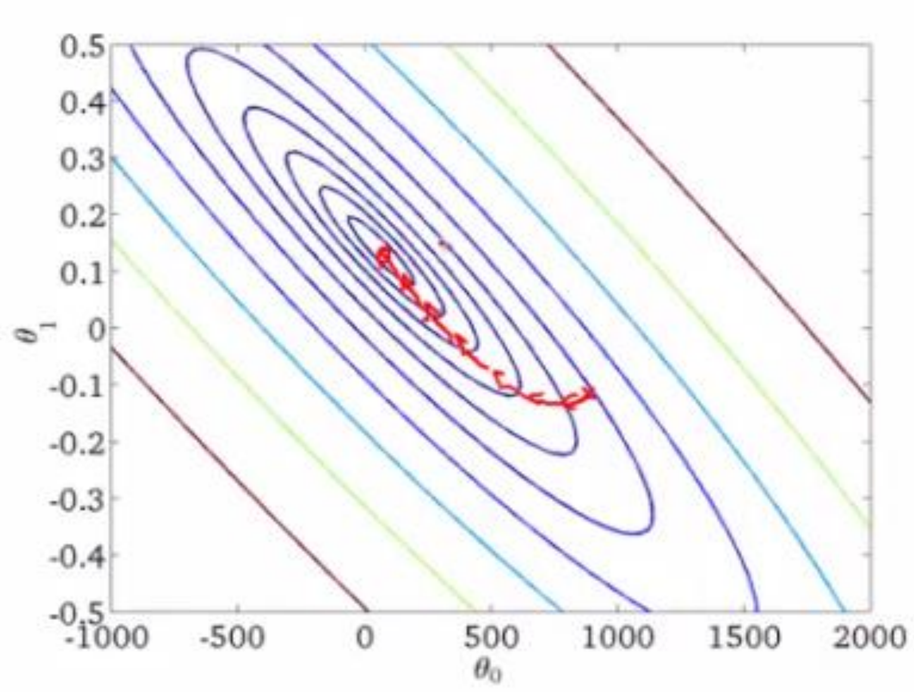
$$\boldsymbol{\theta} \coloneqq \boldsymbol{\theta} + \alpha\big(y - h_{\boldsymbol{\theta}}(\boldsymbol{x})\big)\boldsymbol{x}$$

- Repeat…

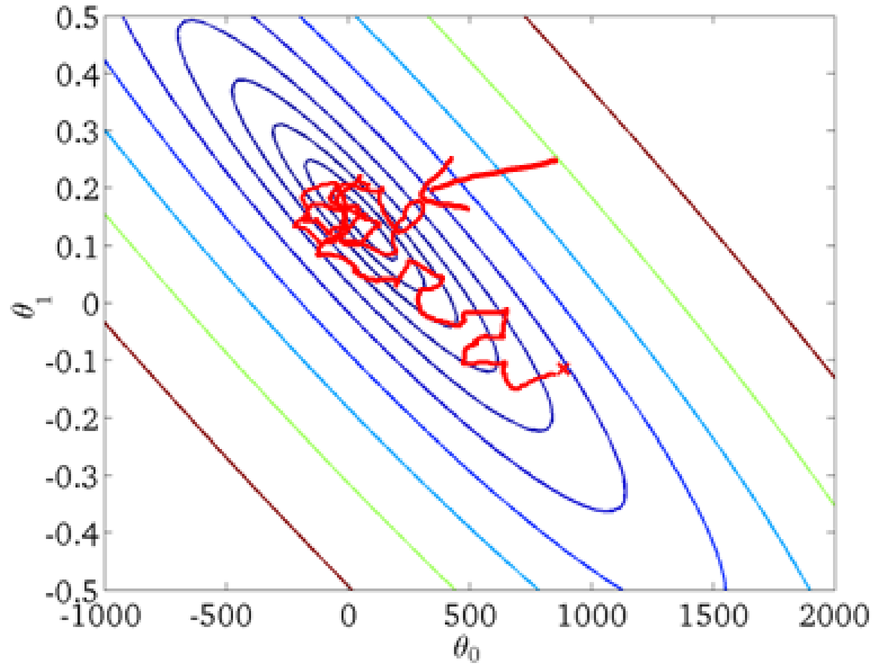
Gradient descent -- batch updating

Stochastic gradient descent -- online updating

GD vs. SGD



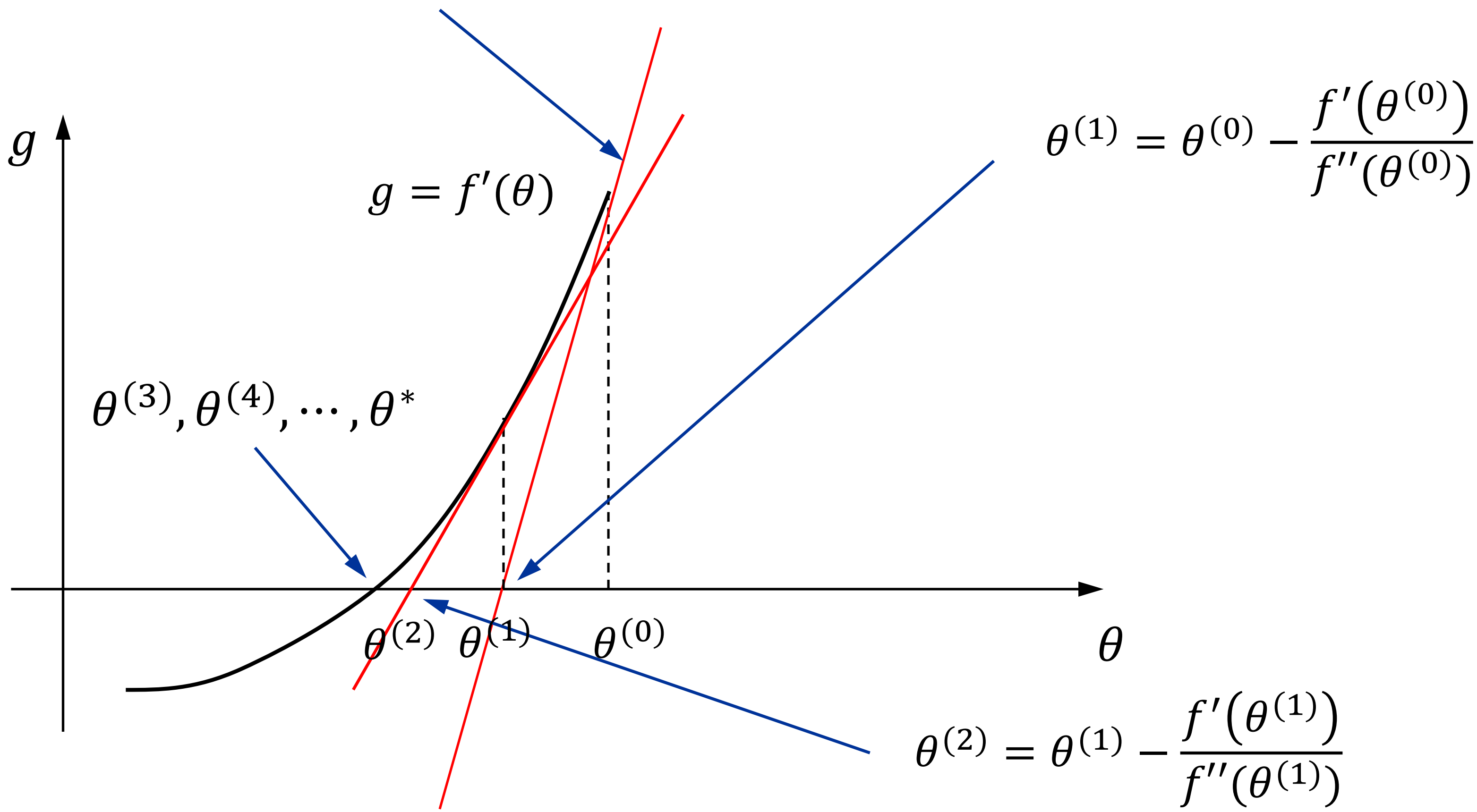
Gradient Descent (GD)



Stochastic Gradient Descent (SGD)

Illustration of Newton's Method

tangent line: $g = f'\left(\theta^{(0)}\right) + f''(\theta^{(0)})(\theta - \theta^{(0)})$



Newton's Method

- Problem

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

- Second-order Taylor expansion

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

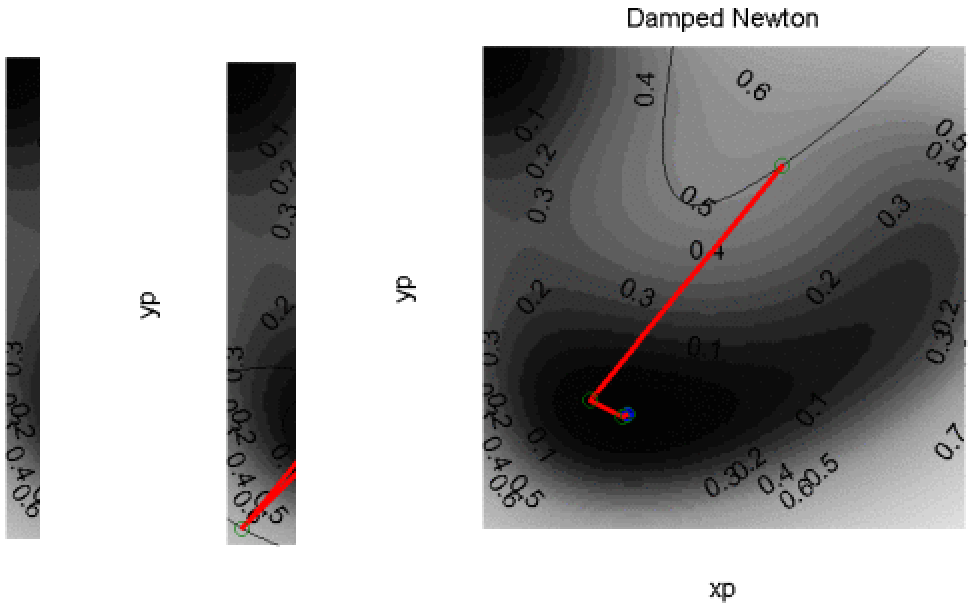
$$\nabla\phi(\boldsymbol{\theta}) = 0 \Rightarrow \boldsymbol{\theta} = \boldsymbol{\theta}^{(k)} - \nabla^2 f(\boldsymbol{\theta}^{(k)})^{-1}\nabla f(\boldsymbol{\theta}^{(k)})$$

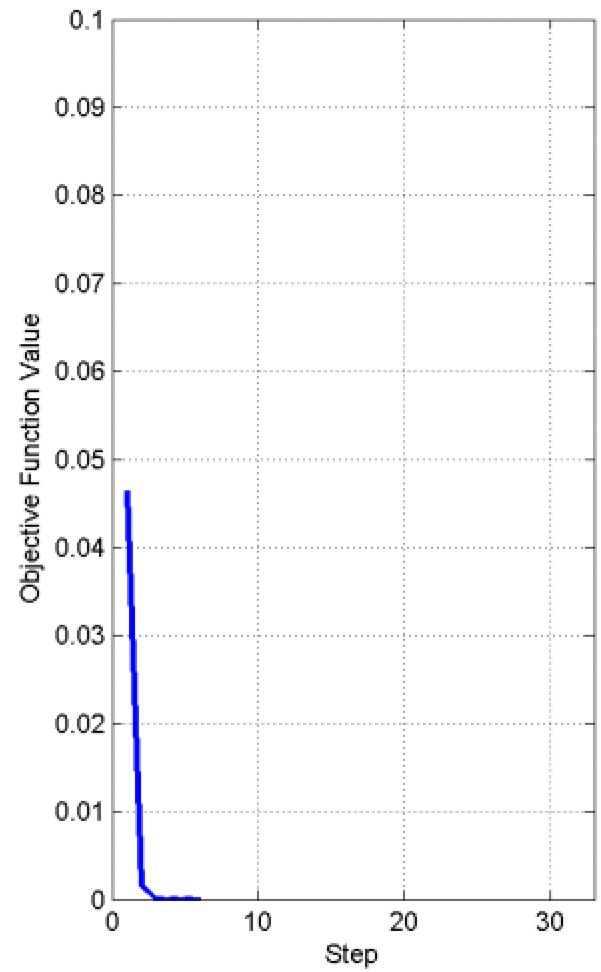
- Newton's method (also called Newton-Raphson method)

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

Hessian Matrix

Gradient' vs. Newton's Method





Newton's Method for Logistic Regression

- Optimization Problem

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

- Gradient and Hessian Matrix

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

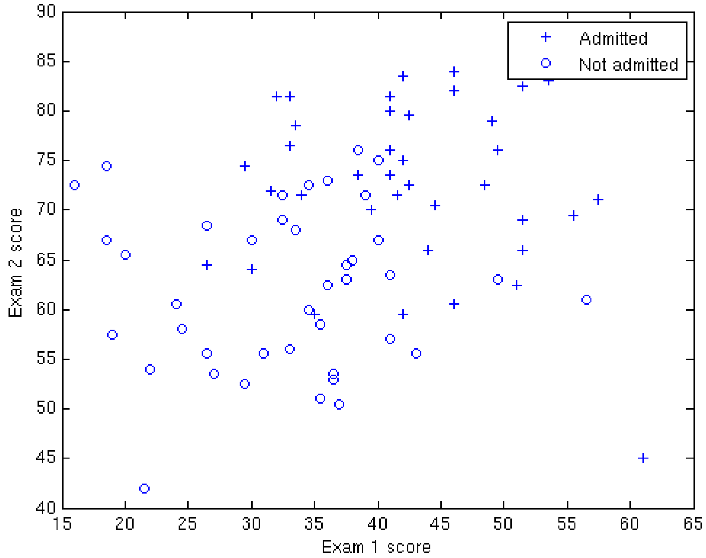
$$\boldsymbol{H} = \frac{1}{N}\sum_{i=1}^{N} h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right)^{\mathrm{T}}\left(1 - h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right)\right)\boldsymbol{x}^{(i)}\left(\boldsymbol{x}^{(i)}\right)^{\mathrm{T}}$$

- Weight updating using Newton's method

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

Practice 2: Logistic Regression

- Given the following training data:



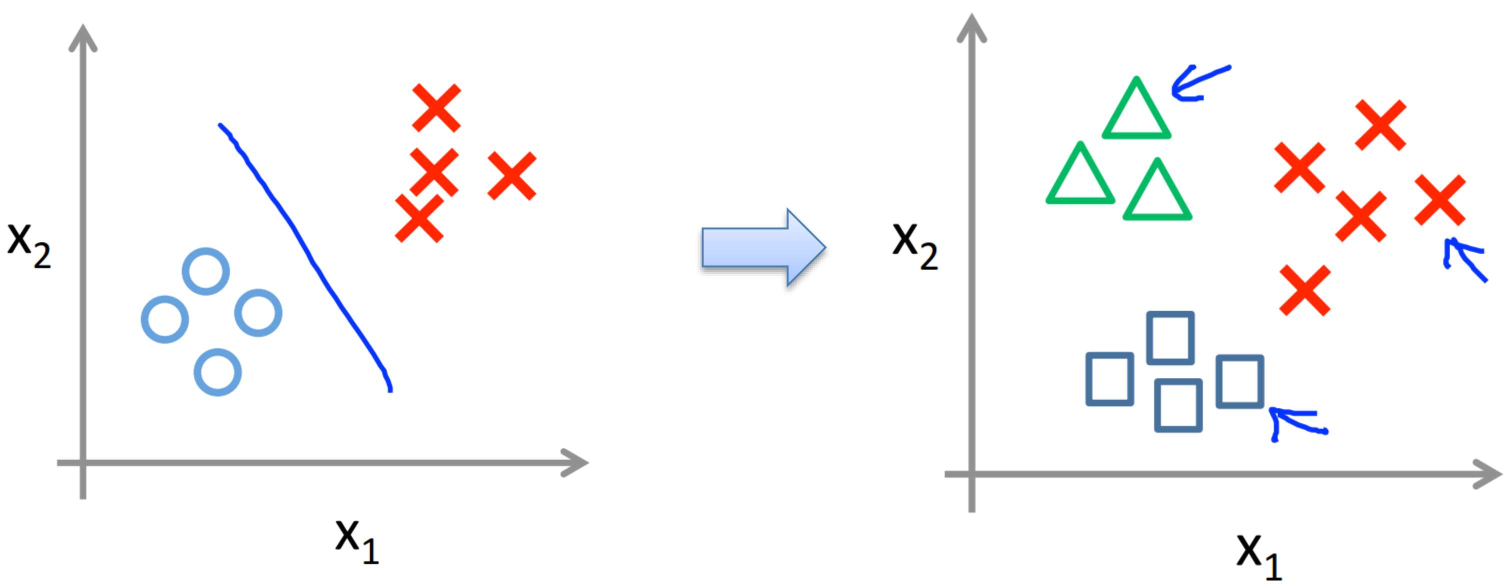
http://openclassroom.stanford.edu/MainFolder/DocumentPage.php?course=DeepLearning&doc=exercises/ex4/ex4.html

- Implement 1) GD; 2) SGD; 3) Newton's Method for logistic regression, starting with the initial parameter $\boldsymbol{\theta} = 0$.
- 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 Hypothesis

- Hypothesis

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

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

- Hypothesis (Compact Form)

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

- Parameters

$$\boldsymbol{\theta}_{C\times M}$$

Maximum Likelihood Estimation

- (Conditional) Log-likelihood

Softmax Regression

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

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

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

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

Logistic Regression

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

Gradient Descent Optimization

- Gradient

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

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

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

Error × Feature

Gradient Descent Optimization

- Gradient Descent

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

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

- Stochastic Gradient Descent

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

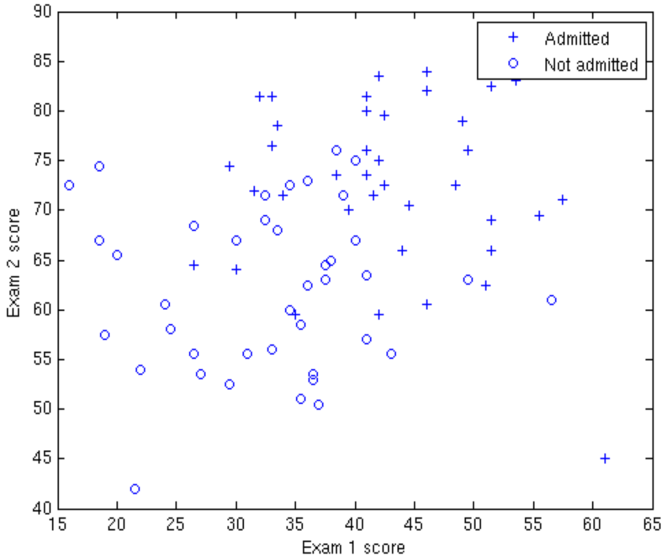
The other optimization methods

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

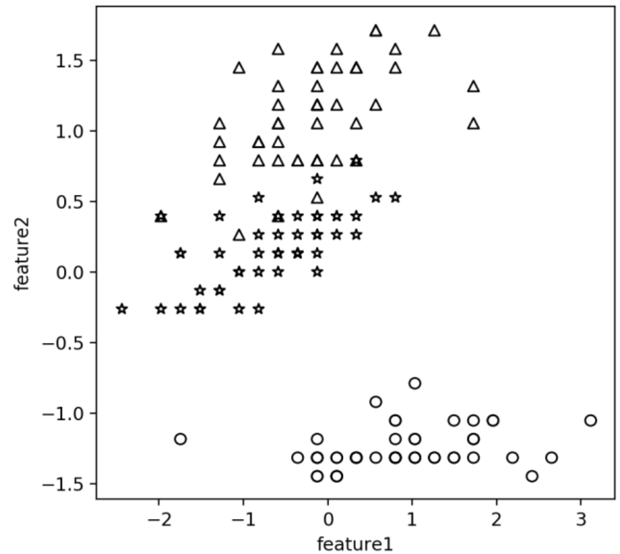
Practice 3: Softmax Regression

- Given the following training data sets:

(1)



(2)



(1) http://openclassroom.stanford.edu/MainFolder/DocumentPage.php?course=DeepLearning&doc=exercises/ex4/ex4.html
(2) https://pan.baidu.com/s/1gU81bKsIj8cRokOYEk1Jzw password: w2a8

- For data set (1), implement logistic regression and softmax regression with 1) GD; 2) SGD.
- For data set (2), implement softmax regression with 1) GD; 2) SGD.
- Compare logistic regression and softmax regression.

Any Questions?