UVA CS 6316: Machine Learning

Lecture 14 Extra: More about Logistic Regression

Dr. Yanjun Qi

University of Virginia

Department of Computer Science

Today: Extra

- ✓ Bayes Classifier and MAP Rule?
  - Bayes Classifier
  - Expected Prediction Error
  - 0-1 Loss function for Bayes Classifier

- ✓ Logistic regression
  - Parameter Estimation for LR

$$p(y|x) = \frac{e^{\beta x}}{1+e^{\beta x}}$$

$$\Downarrow$$

$$\vec{\beta}$$

Bayes classifiers

- Treat each feature attribute and the class label as random variables.

- Given a sample **x** with attributes (x_1, x_2, … , x_p):
  - Goal is to predict its class *C.*
  - Specifically, we want to find the value of C_i that maximizes $p(C_i \mid x_1, x_2, \ldots, x_p)$.

- Can we estimate $p(C_i \mid \mathbf{x}) = p(C_i \mid x_1, x_2, \ldots, x_p)$ directly from data?

Bayes classifiers

- Treat each feature attribute and the class label as random variables.

- Given a sample **x** with attributes ($x_1, x_2, \ldots, x_p$):
  - Goal is to predict its class *C.*
  - Specifically, we want to find the value of C_i that maximizes $p(C_i \mid x_1, x_2, \ldots, x_p)$.

- Can we estimate $p(C_i \mid \mathbf{x}) = p(C_i \mid x_1, x_2, \ldots, x_p)$ directly from data?

$\rightarrow \{C_1, C_2, \ldots, C_L\}$

Bayes classifiers
➔ MAP classification rule

- Establishing a probabilistic model for classification

➔ **MAP** classification rule

- **MAP**: **M**aximum **A** **P**osterior
- Assign x to c^* if

$$P(C = c^* \mid \mathbf{X} = \mathbf{x}) > P(C = c \mid \mathbf{X} = \mathbf{x})$$

$$\text{for } c \neq c^*, \ c = c_1, \cdots, c_L$$

Adapt from Prof. Ke Chen NB slides

Bayes classifiers
➔ MAP classification rule

- Establishing a probabilistic model for classification

➔ **MAP** classification rule

  – **MAP**: **M**aximum **A** **P**osterior

  – Assign x to c^* if

$$P(C=c^* \mid \mathbf{X}=\mathbf{x}) > P(C=c \mid \mathbf{X}=\mathbf{x}) \quad c \neq c^*,\ c = c_1, \cdots, c_L$$

$$\left\{ \begin{array}{l} P(C=C_1 \mid x) \\ P(C=C_2 \mid x) \\ P(C=C_3 \mid x) \end{array} \right\} \max \Rightarrow C_i$$

Adapt from Prof. Ke Chen NB slides

Bayes Classifiers – MAP Rule

Task: Classify a new instance X based on a tuple of attribute values $X = \langle X_1, X_2, \ldots, X_p \rangle$ into one of the classes

$$c_{MAP} = \underset{c_j \in C}{\operatorname{argmax}}\, P(c_j \mid x_1, x_2, \ldots, x_p)$$

WHY ?

MAP = Maximum Aposteriori Probability

Adapt From Carols' prob tutorial

0-1 LOSS for Classification

if $k = \ell$, $L(k,\ell) = 0$

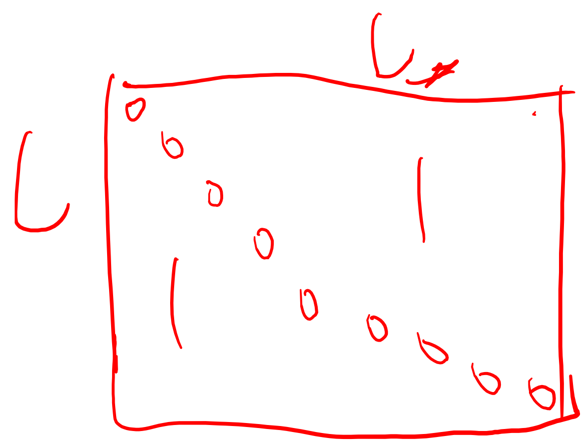
- Procedure for categorical output variable C

if $k \neq \ell$, $L(k,\ell) = 1$

- Frequently, 0-1 loss function used: $\boldsymbol{L}(k, \ell)$

- $\boldsymbol{L}(k, \ell)$ is the price paid for misclassifying an element from class C_k as belonging to class C_ℓ

  ➔ *L*L matrix*



$C_1, C_2, \cdots, C_L$.

Expected prediction error (EPE)

- Expected prediction error (EPE), with expectation taken w.r.t. the joint distribution Pr(C,X)
  - $\Pr(C,X)=\Pr(C\mid X)\Pr(X)$

e.g. 0-1 loss

$E_X(X)$

$E_X(g(X))$

$$\mathrm{EPE}(f)=E_{X,C}(L(C,f(X))$$

$$=\mathrm{E}_X\sum_{k=1}^{L}L[C_k,f(X)]\Pr(C_k\mid X)$$

Consider sample population distribution

$$EPE(f) = E_{X,C}\left(L(C, f(X))\right)$$

$$= E_X E_{C|X}\left[L(C, f(X)) \mid X\right]$$

$$E_C(C) = \sum_{i=1}^{L} C_i\, p(C_i)$$

Discrete RV's Expectation

$$= E_X \sum_{k=1}^{L} L[C_k, f(X)]\, Pr(C_k \mid X)$$

$$\arg\min_f EPE(f(X))$$

$\Rightarrow$ Pointwise minization when $X = x$

$$\Rightarrow \hat{f}(X=x) = \arg\min_{f(x) \in C} \sum_{k=1}^{L} \underbrace{L(C_k, f(x))}\, Pr(C_k \mid X=x)$$

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

$$\Rightarrow \hat{f}(x) = \arg\max_{C_k \in \{C_1, C_2, C_3, \ldots, C_L\}} Pr(C_k \mid X=x)$$

$$\left\{ \begin{array}{l} p(C_1 \mid x) \\ p(C_2 \mid x) \\ \vdots \\ p(C_L \mid x) \end{array} \right.$$

Expected prediction error (EPE)

$$\mathrm{EPE}(f) = E_{X,C}(L(C, f(X)) = \mathrm{E}_X \sum_{k=1}^{K} L[C_k, f(X)] \Pr(C_k \mid X)$$

Consider sample population distribution

- Pointwise minimization suffices

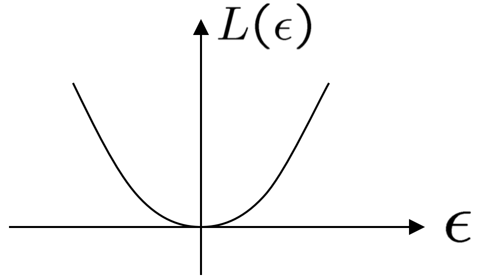
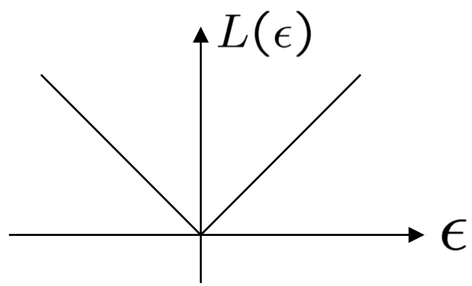
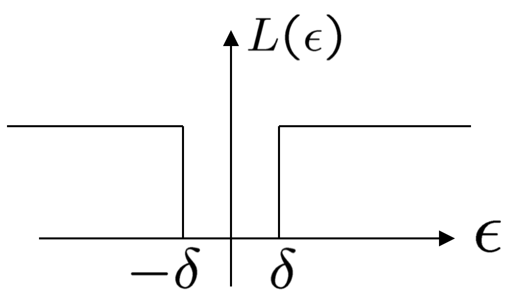
- ➔ simply $$\hat{f}(X) = \mathrm{argmin}_{g \in C} \sum_{k=1}^{K} L(C_k, g) \Pr(C_k \mid X = x)$$

$$\hat{f}(X) = C_k \text{ if}$$

$$\Pr(C_k \mid X = x) = \max_{g \in C} \Pr(g \mid X = x)$$

Bayes Classifier

SUMMARY: WHEN Expected prediction error (EPE) USES DIFFERENT LOSS

Loss Function		Estimator $\hat{f}(x)$
L_2		$EPE = E_{X,Y}(Y - f(X))^2$ $\hat{f}(x) = E[Y\|X = x]$
L_1		$\hat{f}(x) = \text{median}(Y\|X = x)$
0-1		$\hat{f}(x) = \arg\max_Y P(Y\|X = x)$ (Bayes classifier / MAP)

Today: Extra

- ✓ Why Bayes Classification – MAP Rule?
  - Expected Prediction Error
  - 0-1 Loss function for Bayes Classifier

- ✓ Logistic regression
  - Parameter Estimation for LR

$$p(y|x) = \frac{e^{\beta x}}{1+e^{\beta x}}$$

$$\Downarrow$$

$$\vec{\beta}$$

Newton's method for optimization

- The most basic second-order optimization algorithm
- Updating parameter with

GD: $\theta_{k+1} = \theta_k - \alpha g_k$

Newton: $$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \underbrace{\underset{p \times p}{\mathbf{H}_K^{-1}} \underset{p \times 1}{\mathbf{g}_k}}_{p \times 1}$$

Review: Hessian Matrix / n==2 case

Singlevariate → multivariate

$$f(x,y)$$

- 1st derivative to gradient,

$$g = \nabla f = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix}$$

- 2nd derivative to Hessian

$$H = \begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{pmatrix}$$

Review: Hessian Matrix

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is a function that takes a vector in $\mathbb{R}^n$ and returns a real number. Then the ***Hessian*** matrix with respect to x, written $\nabla_x^2 f(x)$ or simply as H is the $n \times n$ matrix of partial derivatives,

$$\nabla_x^2 f(x) \in \mathbb{R}^{n\times n} = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix}.$$

Newton's method for optimization

- Making a quadratic/second-order Taylor series approximation

$$\hat{f}_{quad}(\boldsymbol{\theta}) = f(\boldsymbol{\theta}_k) + \mathbf{g}_k^T(\boldsymbol{\theta} - \boldsymbol{\theta}_k) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_k)^T \mathbf{H}_k(\boldsymbol{\theta} - \boldsymbol{\theta}_k)$$

Finding the minimum solution of the above right quadratic approximation (quadratic function minimization is easy !)

$$\hat{f}(\theta) = f(\theta_k) + g_k^T(\theta - \theta_k) + \underbrace{\frac{1}{2}(\theta - \theta_k)^T H_k (\theta - \theta_k)}$$

$$\Downarrow$$

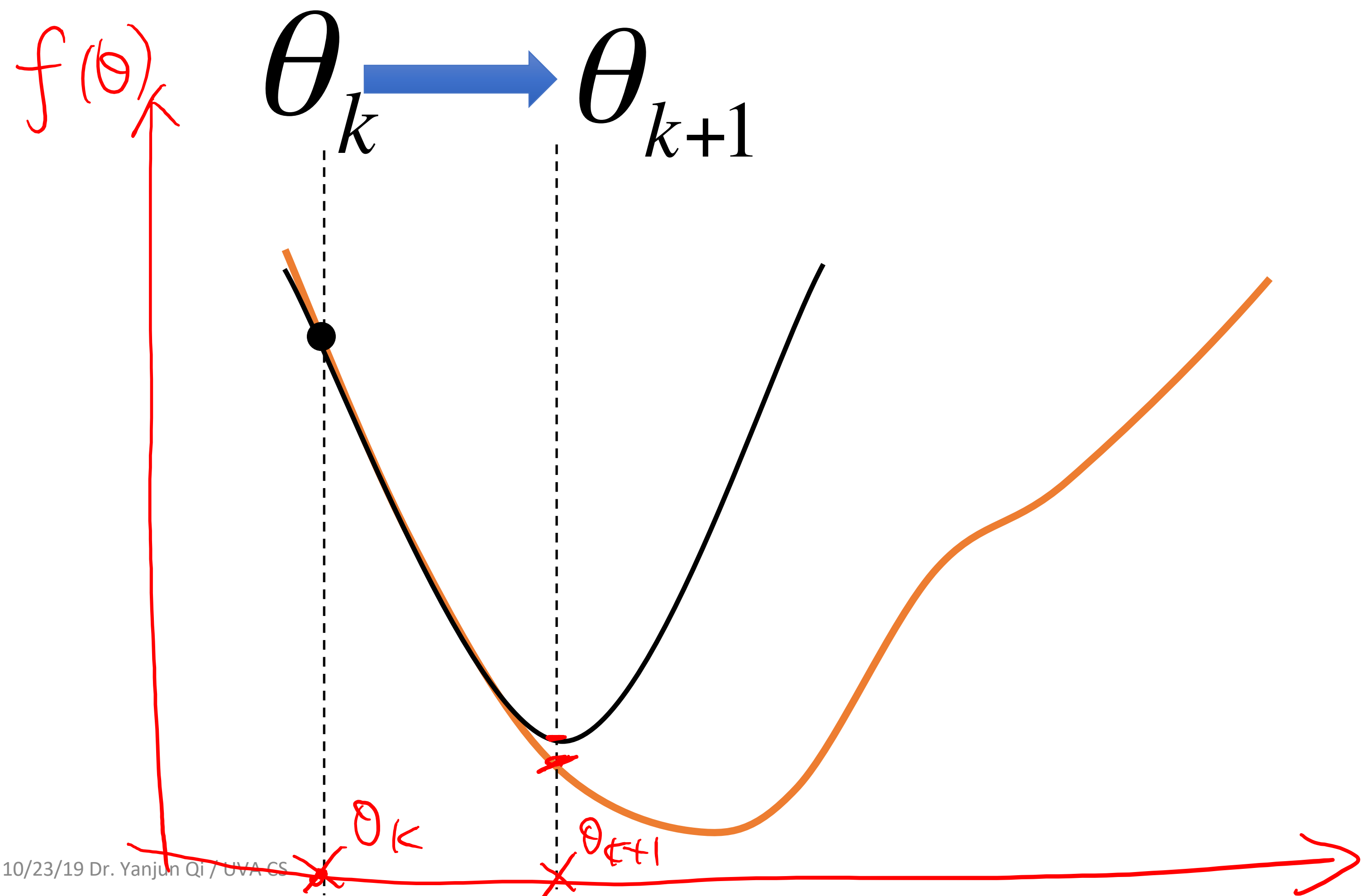
$$\frac{1}{2}\left(\theta^T H_k \theta - 2\theta^T H_k \theta_k + \theta_k^T H_k \theta_k\right)$$

$$\frac{\partial \hat{f}(\theta)}{\partial \theta} = 0 + g_k + \underbrace{\frac{2}{2} H_k \theta - \frac{2}{2} H_k \theta_k}_{\text{see p24 handout}} := 0$$

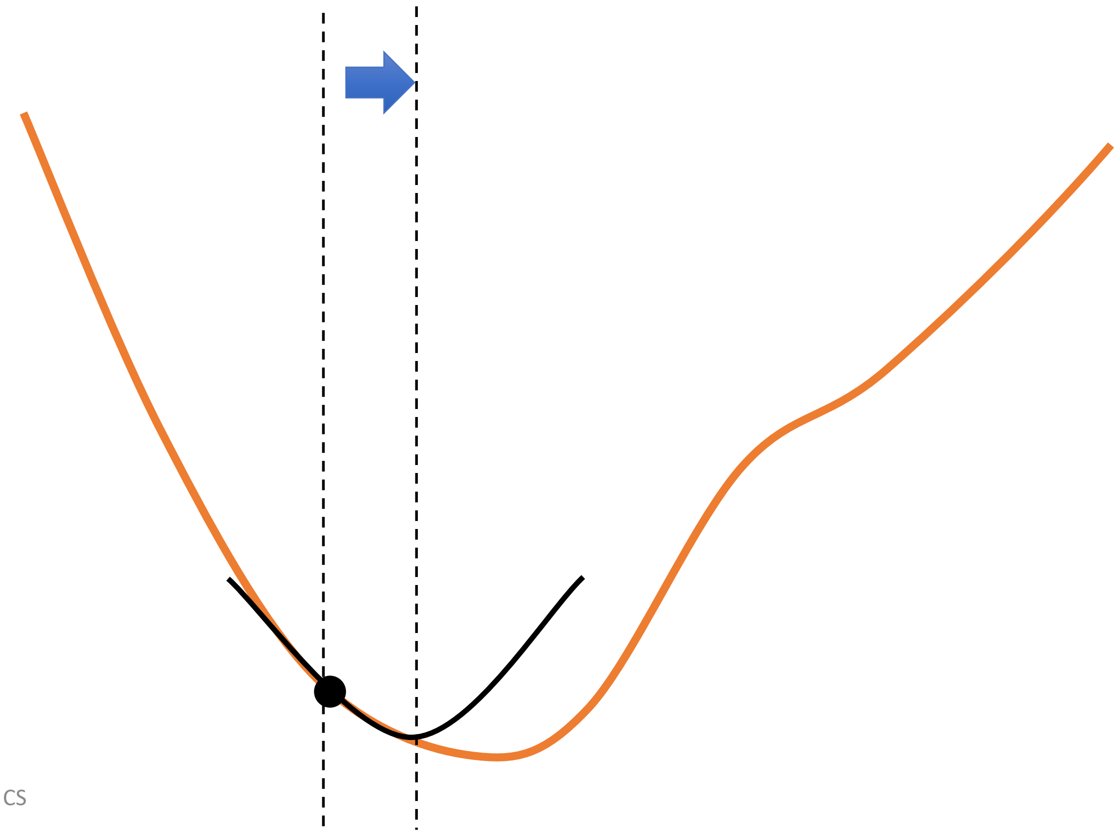
$$g_k + H_k(\theta - \theta_k) = 0$$

$$\Rightarrow \quad \theta = \theta_k - H_k^{-1} g_k \qquad \text{where } \begin{cases} H_k \in R^{p \times p} \\ g_k \in R^{p} \end{cases}$$

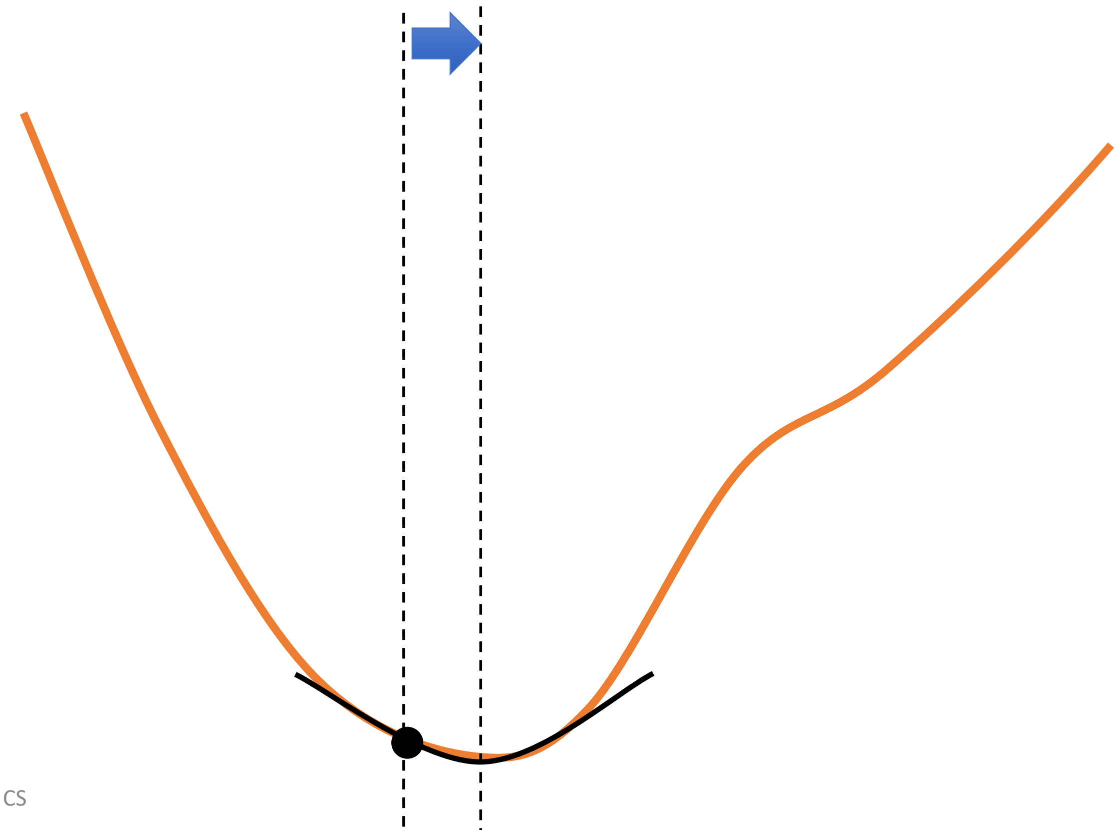
Newton's Method / second-order Taylor series approximation



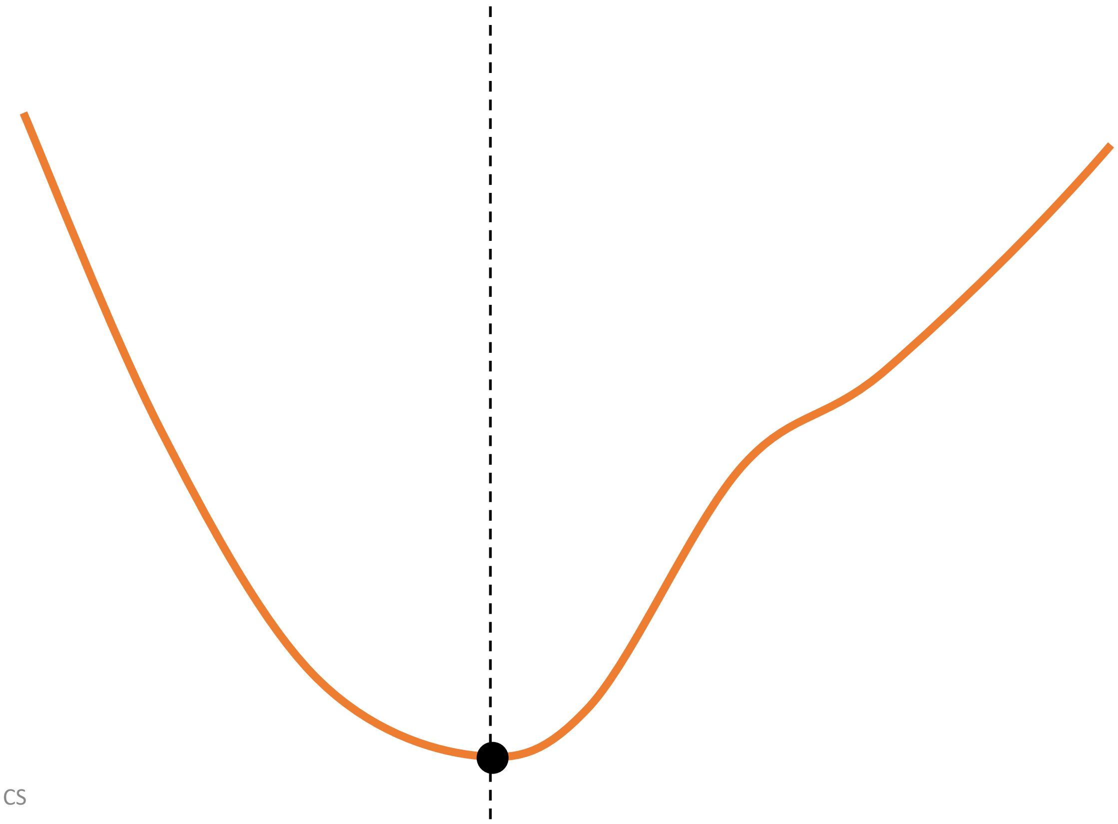
Newton's Method / second-order Taylor series approximation



Newton's Method / second-order Taylor series approximation



Newton's Method / second-order Taylor series approximation



Newton's Method

- At each step:

$$\theta_{k+1} = \theta_k - \frac{f'(\theta_k)}{f''(\theta_k)}$$

$$\theta_{k+1} = \theta_k - H^{-1}(\theta_k)\nabla f(\theta_k)$$

- Requires 1st and 2nd derivatives
- Quadratic convergence
- ➔ However, finding the inverse of the Hessian matrix is often expensive

Newton vs. GD for optimization

- Newton: a quadratic/second-order Taylor series approximation

$\theta_{k+1} = \theta_k - \frac{1}{H(\theta_k)} g(\theta_k)$

$$\hat{f}_{quad}(\boldsymbol{\theta}) = f(\boldsymbol{\theta}_k) + \mathbf{g}_k^T(\boldsymbol{\theta} - \boldsymbol{\theta}_k) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_k)^T \mathbf{H}_k (\boldsymbol{\theta} - \boldsymbol{\theta}_k)$$

Finding the minimum solution of the above right quadratic approximation (quadratic function minimization is easy !)

$$\hat{f}_{quad}(\boldsymbol{\theta}) = f(\boldsymbol{\theta}_k) + \mathbf{g}_k^T(\boldsymbol{\theta} - \boldsymbol{\theta}_k) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_k)^T \frac{1}{\alpha} (\boldsymbol{\theta} - \boldsymbol{\theta}_k)$$

$\Downarrow$

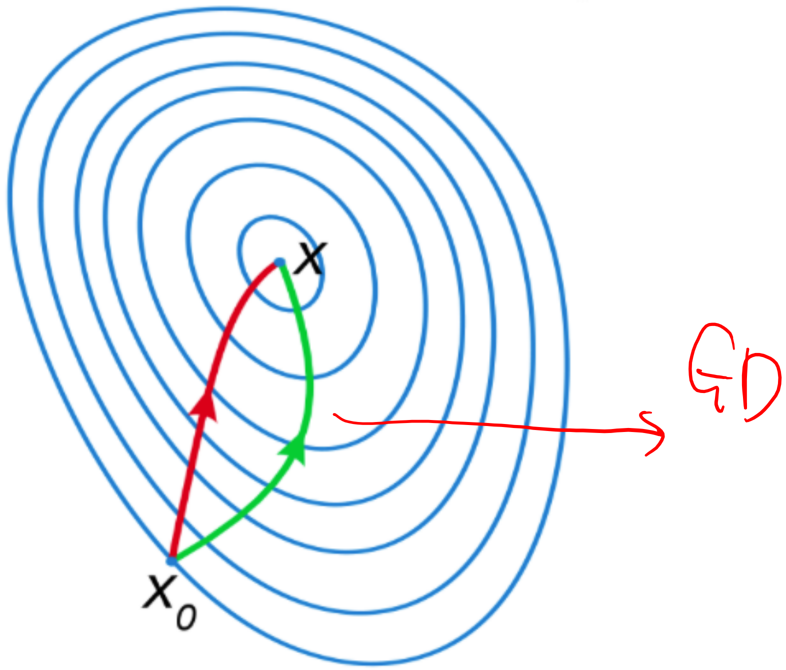
$\theta_{k+1} = \theta_k - \alpha\, g(\theta_k)$

Comparison

- Newton's method vs. Gradient descent

A comparison of gradient descent (green) and Newton's method (red) for minimizing a function (with small step sizes).

Newton's method uses curvature information to get a more direct route ...



MLE for Logistic Regression Training

Let's fit the logistic regression model for K=2, i.e., number of classes is 2

For Bernoulli distribution

$$p(y \mid x)^y (1-p)^{1-y}$$

Training set: (x_i, y_i), i=1,…,N

(conditional)
Log-likelihood:

How?

$$l(\beta) = \sum_{i=1}^{N} \{\log \Pr(Y = y_i \mid X = x_i)\}$$

$$= \sum_{i=1}^{N} y_i \log(\Pr(Y=1 \mid X = x_i)) + (1-y_i)\log(\Pr(Y=0 \mid X = x_i))$$

$$= \sum_{i=1}^{N} (y_i \log \frac{\exp(\beta^T x_i)}{1+\exp(\beta^T x_i)}) + (1-y_i)\log \frac{1}{1+\exp(\beta^T x_i)})$$

$$= \sum_{i=1}^{N} (y_i \beta^T x_i - \log(1+\exp(\beta^T x_i)))$$

x_i are (p+1)-dimensional input vector with leading entry 1
\beta is a (p+1)-dimensional vector

We want to maximize the log-likelihood in order to estimate *\beta*

$$l(\beta)=\sum_{i=1}^{N}\{\log\Pr(Y=y_i \mid X=x_i)\}$$

y_i

$$\log\{\Pr(Y=y_i \mid X=x_i) = p(y_i \mid x_i)\} \Rightarrow$$

$y_i = 1$ $\quad p(y_i=1 \mid x)$

$y_i = 0$ $\quad 1-p(y_i=1 \mid x)$

$$= \log\{p(y_i=1 \mid x)^{y_i}(1-p(y_i=1 \mid x_i))^{1-y_i}\}$$

$$= y_i \log p(y_i=1 \mid x) + (1-y_i)\log(1-p(y_i=1 \mid x))$$

Newton-Raphson for LR (optional)

$$\frac{\partial l(\beta)}{\partial \beta} = \sum_{i=1}^{N} (y_i - \frac{\exp(\beta^T x)}{1+\exp(\beta^T x)}) x_i = 0$$

Vector

(*p*+1) Non-linear equations to solve for (*p*+1) unknowns

β

Solve by Newton-Raphson method:

$$\beta^{new} \leftarrow \beta^{old} - [(\frac{\partial^2 l(\beta)}{\partial \beta \partial \beta^T})]^{-1} \frac{\partial l(\beta)}{\partial \beta},$$

$$\text{where, } (\frac{\partial^2 l(\beta)}{\partial \beta \partial \beta^T}) = -\sum_{i=1}^{N} x_i x_i^T (\frac{\exp(\beta^T x_i)}{1+\exp(\beta^T x_i)})(\frac{1}{1+\exp(\beta^T x_i)})$$

$p(x_i ; \beta)$

$1 - p(x_i ; \beta)$

minimizes a quadratic approximation
to the function we are really interested in.

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \mathbf{H}_K^{-1} \mathbf{g}_k$$

Newton-Raphson for LR...

$$\frac{\partial l(\beta)}{\partial \beta} = \sum_{i=1}^{N}(y_i - \frac{\exp(\beta^T x)}{1+\exp(\beta^T x)})x_i = X^T(y-p)$$

$p(y=1|x) = \frac{e^{\beta^T x}}{1+e^{\beta x}}$

$$(\frac{\partial^2 l(\beta)}{\partial \beta \partial \beta^T}) = -X^T W X$$

So, NR rule becomes: $\beta^{new} \leftarrow \beta^{old} + (X^T W X)^{-1} X^T (y-p),$

$$X = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_N^T \end{bmatrix}_{N-by-(p+1)}, \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}_{N-by-1}, \quad p = \begin{bmatrix} \exp(\beta^T x_1)/(1+\exp(\beta^T x_1)) \\ \exp(\beta^T x_2)/(1+\exp(\beta^T x_2)) \\ \vdots \\ \exp(\beta^T x_N)/(1+\exp(\beta^T x_N)) \end{bmatrix}_{N-by-1},$$

$X : N \times (p+1)$ matrix of x_i

$y : N \times 1$ matrix of y_i

$p : N \times 1$ matrix of $p(x_i; \beta^{old})$

$W : N \times N$ diagonal matrix of $p(x_i; \beta^{old})(1-p(x_i; \beta^{old}))$

$$(\frac{\exp(\beta^T x_i)}{(1+\exp(\beta^T x_i))})(1-\frac{1}{(1+\exp(\beta^T x_i))})$$

Newton-Raphson for LR…

1

- Newton-Raphson
  - $\beta^{new} = \beta^{old} + (X^T WX)^{-1} X^T (y-p)$

    $= (X^T WX)^{-1} X^T W (X\beta^{old} + W^{-1}(y-p))$

    $= (X^T WX)^{-1} X^T Wz$

    Re expressing Newton step as weighted least square step

    $(X^T X)^{-1} X^T Y$

    W $\quad$ WZ

  - Adjusted response

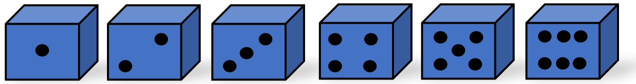
    $z = X\beta^{old} + W^{-1}(y-p)$

  - Iteratively reweighted least squares (IRLS)

    $\beta^{new} \leftarrow \arg\min_{\beta}(z - X\beta^T)^T W(z - X\beta^T)$

    $\leftarrow \arg\min_{\beta}(y-p)^T W^{-1}(y-p)$

Binary ➔ Multinoulli Logistic Regression Model



$p(G|x)$

Directly models the posterior probabilities as the output of regression

$$\Pr(G=k|X=x)=\frac{\exp(\beta_{k0}+\beta_k^T x)}{1+\sum_{l=1}^{K-1}\exp(\beta_{l0}+\beta_l^T x)},\ k=1,\ldots,K-1$$

$$\Pr(G=K|X=x)=\frac{1}{1+\sum_{l=1}^{K-1}\exp(\beta_{l0}+\beta_l^T x)}$$

x is p-dimensional input vector

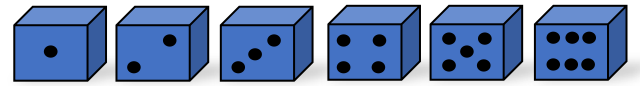
β_k^T is a p-dimensional vector for each class k

Total number of parameters is $(K-1)(p+1)$ $\beta_{k0}, \vec{\beta}_k, \quad k=1,2,\cdots,K-1$

Note that the class boundaries are linear

Binary → Multinoulli Logistic Regression Model

(e.g. $K=6$)



$$p(y=1|x) \Rightarrow \frac{e^{\beta x}}{1+e^{\beta x}}$$

$$p(y=0|x) \Rightarrow \frac{1}{1+e^{\beta x}}$$

$$\frac{e^{\beta_K^T x}}{1+e^{\beta_1^T x}+e^{\beta_2^T x}\ldots}$$

e.g.

$$\ln \frac{P(G=k|x)}{P(G=K|x)} = \underset{\beta_{k0}+\beta_k^T x}{\underset{\parallel}{0}} \Rightarrow \text{linear}$$

Note that the class boundaries are linear

References

- Prof. Tan, Steinbach, Kumar's "Introduction to Data Mining" slide
- Prof. Andrew Moore's slides
- Prof. Eric Xing's slides
- Prof. Ke Chen NB slides
- Hastie, Trevor, et al. *The elements of statistical learning*. Vol. 2. No. 1. New York: Springer, 2009.