

Lecture 3: Logistic Regression

Feng Li

Shandong University

fli@sdu.edu.cn

December 28, 2021

Lecture 3: Logistic Regression

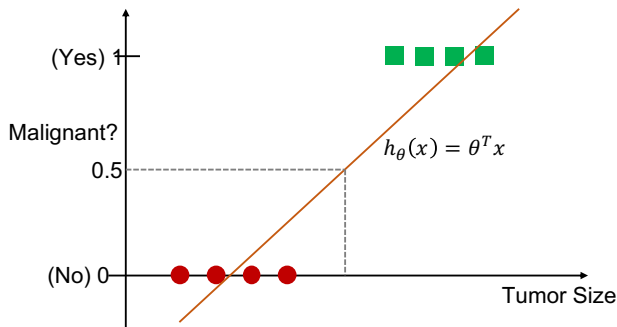
- 1 Classification
- 2 Logistic Regression
- 3 Newton's Method
- 4 Multiclass Classification

- Classification problems
 - Email: Spam / Not Spam?
 - Online Transactions: Fraudulent (Yes/No)?
 - Tumor: Malignant/Benign?
- The classification result can be represented by a binary variable $y \in \{0, 1\}$

$$y = \begin{cases} 0 : \text{"Negative Class"} \text{ (e.g., benign tumor)} \\ 1 : \text{"Positive Class"} \text{ (e.g., malignant tumor)} \end{cases}$$

Warm-Up

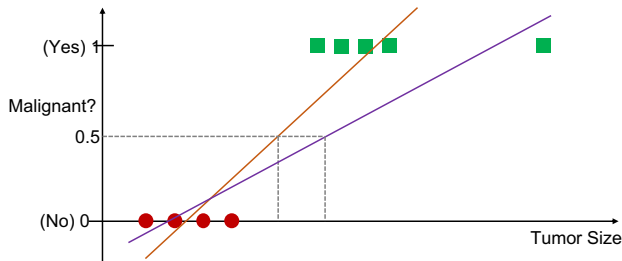
- What if applying linear regress to classification?



- The threshold classifier output $h_{\theta}(x)$ at 0.5
 - If $h_{\theta}(x) \geq 0.5$, predict $y = 1$
 - If $h_{\theta}(x) < 0.5$, predict $y = 0$

Warm-Up (Contd.)

- When a new training example comes



- An interesting observation
 - $y \in \{0, 1\}$ in classification problem, but the linear regression model $h_{\theta}(x) = \theta^T x$ can be > 1 or < 0 to fit the given training example
 - Logistic regression: $0 \leq h_{\theta}(x) \leq 1$

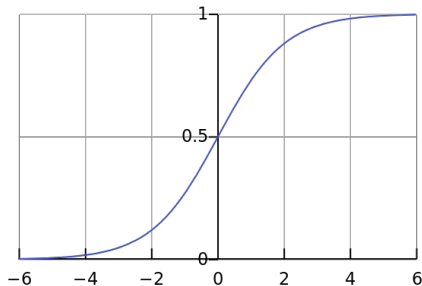
- Classification problem

- Similar to regression problem, but we would like to predict only a small number of discrete values (instead of continuous values)
- Binary classification problem: $y \in \{0, 1\}$ where 0 represents negative class, while 1 denotes positive class
- $y^{(i)} \in \{0, 1\}$ is also called the **label** for the training example

Logistic Regression (Contd.)

- Logistic function

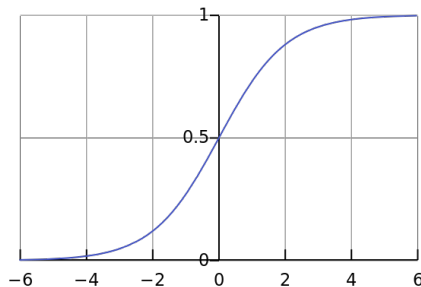
$$g(z) = \frac{1}{1 + e^{-z}}$$



Logistic Regression (Contd.)

- Properties of the logistic function

- Bound: $g(z) \in (0, 1)$
- Symmetric: $1 - g(z) = g(-z)$
- Gradient: $g'(z) = g(z)(1 - g(z))$



Logistic Regression (Contd.)

- Logistic regression defines $h_{\theta}(x)$ using the logistic function

$$h_{\theta}(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

- Interpretation of the hypothesis output
 - $h_{\theta}(x)$: Estimated probability that $y = 1$ on input x
 - Example: if $x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \text{Tumor Size} \end{bmatrix}$, $h_{\theta}(x) = 0.7$, which tells patient that 70% chance of tumor being malignant

Logistic Regression (Contd.)

- Data samples are drawn randomly
 - X : random variable representing feature vector
 - Y : random variable representing label
- Given an input feature vector x , we have
 - The conditional probability of $Y = 1$ given $X = x$

$$\Pr(Y = 1 \mid X = x; \theta) = h_{\theta}(x) = \frac{1}{(1 + \exp(-\theta^T x))}$$

- The conditional probability of $Y = 0$ given $X = x$

$$\Pr(Y = 0 \mid X = x; \theta) = 1 - h_{\theta}(x) = \frac{1}{(1 + \exp(\theta^T x))}$$

Logistic Regression: A Closer Look ...

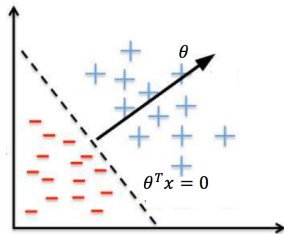
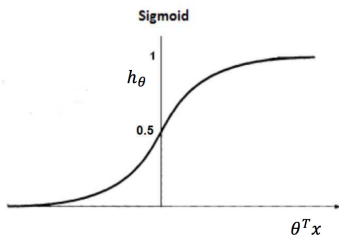
- What's the underlying decision rule in logistic regression?
- At the decision boundary, both classes are equiprobable; thus, we have

$$\begin{aligned}\Pr(Y = 1 \mid X = x; \theta) &= \Pr(Y = 0 \mid X = x; \theta) \\ \Rightarrow \frac{1}{1 + \exp(-\theta^T x)} &= \frac{1}{1 + \exp(\theta^T x)} \\ \Rightarrow \exp(\theta^T x) &= 1 \\ \Rightarrow \theta^T x &= 0\end{aligned}$$

- Therefore, the decision boundary of logistic regression is nothing but a linear hyperplane

Logistic Regression: A Closer Look ... (Contd.)

- Recall that $\Pr(Y = 1 \mid X = x; \theta) = 1/(1 + \exp(-\theta^T x))$
- The “score” $\theta^T x$ is also a measure of distance of x from the hyper-plane (the score is positive for pos. examples, and negative for neg. examples)
 - High positive score: High probability of label 1
 - High negative score: Low probability of label 1 (high prob. of label 0)



Logistic Regression Formulation

- Logistic regression model

$$h_{\theta}(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

- Assume

$$\Pr(Y = 1 \mid X = x; \theta) = h_{\theta}(x)$$

and

$$\Pr(Y = 0 \mid X = x; \theta) = 1 - h_{\theta}(x),$$

then we have the following *probability mass function*

$$p(y \mid x; \theta) = \Pr(Y = y \mid X = x; \theta) = (h_{\theta}(x))^y (1 - h_{\theta}(x))^{1-y}$$

where $y \in \{0, 1\}$

Logistic Regression Formulation (Contd.)

- $Y \mid X = x \sim \text{Bernoulli}(h_{\theta}(x))$
- If we assume $y \in \{-1, 1\}$ instead of $y \in \{0, 1\}$, then

$$p(y \mid x; \theta) = \frac{1}{1 + \exp(-y\theta^T x)}$$

- Assuming the training examples were generated independently, we define the likelihood of the parameters as

$$\begin{aligned} L(\theta) &= \prod_{i=1}^m p(y^{(i)} \mid x^{(i)}; \theta) \\ &= \prod_{i=1}^m (h_{\theta}(x^{(i)}))^{y^{(i)}} (1 - h_{\theta}(x^{(i)}))^{1-y^{(i)}} \end{aligned}$$

Logistic Regression Formulation (Contd.)

- Maximize the log likelihood

$$\ell(\theta) = \log L(\theta) = \sum_{i=1}^m \left(y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log(1 - h(x^{(i)})) \right)$$

- Gradient ascent algorithm
 - $\theta_j \leftarrow \theta_j + \alpha \nabla_{\theta_j} \ell(\theta)$ for $\forall j$, where

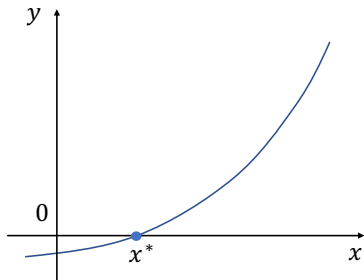
$$\begin{aligned} \frac{\partial}{\partial \theta_j} \ell(\theta) &= \sum_{i=1}^m \frac{y^{(i)} - h_{\theta}(x^{(i)})}{h_{\theta}(x^{(i)}) (1 - h_{\theta}(x^{(i)}))} \cdot \frac{\partial h_{\theta}(x^{(i)})}{\partial \theta_j} \\ &= \sum_{i=1}^m \left(y^{(i)} - h_{\theta}(x^{(i)}) \right) x_j^{(i)} \end{aligned}$$

Logistic Regression Formulation (Contd.)

$$\begin{aligned}& \frac{\partial}{\partial \theta_j} \ell(\theta) \\&= \sum_{i=1}^m \left(\frac{y^{(i)}}{h_{\theta}(x)} \frac{\partial h_{\theta}(x)}{\partial \theta_j} - \frac{1 - y^{(i)}}{1 - h_{\theta}(x)} \frac{\partial h_{\theta}(x)}{\partial \theta_j} \right) \\&= \sum_{i=1}^m \frac{y^{(i)} - h_{\theta}(x^{(i)})}{h_{\theta}(x^{(i)}) (1 - h_{\theta}(x^{(i)}))} \cdot \frac{\partial h_{\theta}(x^{(i)})}{\partial \theta_j} \\&= \sum_{i=1}^m \left(y^{(i)} - h_{\theta}(x^{(i)}) \right) \cdot \frac{(1 + \exp(-\theta^T x^{(i)}))^2}{\exp(-\theta^T x^{(i)})} \cdot \frac{\exp(-\theta^T x^{(i)}) \cdot x_j^{(i)}}{(1 + \exp(-\theta^T x^{(i)}))^2} \\&= \sum_{i=1}^m \left(y^{(i)} - h_{\theta}(x^{(i)}) \right) x_j^{(i)}\end{aligned}$$

Newton's Method

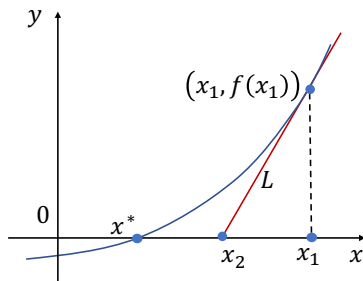
- Given a differentiable real-valued $f : \mathbb{R} \rightarrow \mathbb{R}$, how can we find x such that $f(x) = 0$?



Newton's Method (Contd.)

- A tangent line L to the curve $y = f(x)$ at point $(x_1, f(x_1))$
- The x -intercept of L

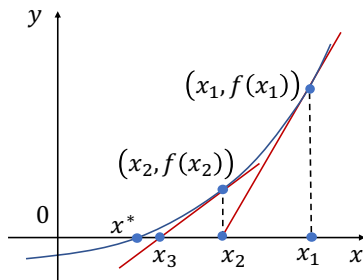
$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$



Newton's Method (Contd.)

- Repeat the process and get a sequence of approximations

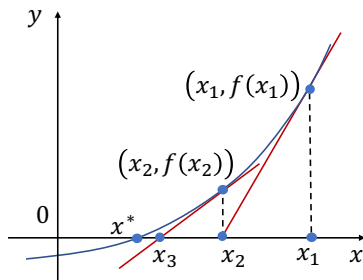
x_1, x_2, x_3, \dots



Newton's Method (Contd.)

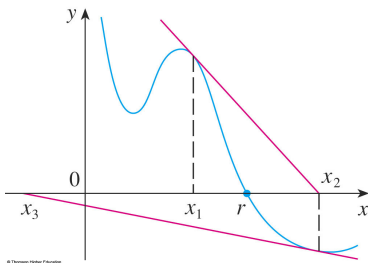
- In general, if convergence criteria is not satisfied,

$$x \leftarrow x - \frac{f(x)}{f'(x)}$$



Newton's Method (Contd.)

- Some properties
 - Highly dependent on initial guess
 - Quadratic convergence once it is sufficiently close to x^*
 - If $f' = 0$, only has linear convergence
 - Is not guaranteed to converge at all, depending on function or initial guess



Newton's Method (Contd.)

- To maximize $f(x)$, we have to find the stationary point of $f(x)$ such that $f'(x) = 0$.
- According to Newton's method, we have the following update

$$x \leftarrow x - \frac{f'(x)}{f''(x)}$$

- Newton-Raphson method:

For $\ell : \mathbb{R}^n \rightarrow \mathbb{R}$, we generalize Newton's method to the multi-dimensional setting

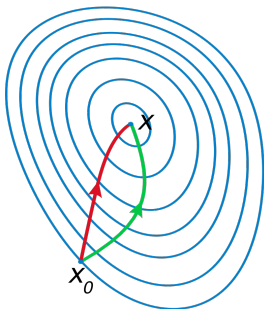
$$\theta \leftarrow \theta - H^{-1} \nabla_{\theta} \ell(\theta)$$

where H is the Hessian matrix

$$H_{i,j} = \frac{\partial^2 \ell(\theta)}{\partial \theta_i \partial \theta_j}$$

Newton's Method (Contd.)

- Higher convergence speed than (batch) gradient descent
- Fewer iterations to approach the minimum
- However, each iteration is more expensive than the one of gradient descent
 - Finding and inverting an $n \times n$ Hessian



More details about Newton's method can be found at https://en.wikipedia.org/wiki/Newton%27s_method

Multiclass Classification

- Multiclass (or multinomial) classification is the problem of classifying instances into one of the more than two classes
- The existing multiclass classification techniques can be categorized into
 - Transformation to binary
 - Extension from binary
 - Hierarchical classification

Transformation to Binary

- One-vs.-rest (one-vs.-all, OvA or OvR, one-against-all, OAA) strategy is to train a single classifier per class, with the samples of that class as positive samples and all other samples as negative ones
 - Inputs: A learning algorithm L , training data $\{(x^{(i)}, y^{(i)})\}_{i=1, \dots, m}$ where $y^{(i)} \in \{1, \dots, K\}$ is the label for the sample $x^{(i)}$
 - Output: A list of classifier f_k for $k \in \{1, \dots, K\}$
 - Procedure: For $\forall k \in \{1, \dots, K\}$, construct a new label $z^{(i)}$ for $x^{(i)}$ such that $z^{(i)} = 1$ if $y^{(i)} = k$ and $z^{(i)} = 0$ otherwise, and then apply L to $\{(x^{(i)}, z^{(i)})\}_{i=1, \dots, m}$ to obtain f_k . Higher $f_k(x)$ implies high probability that x is in class k
 - Making decision: $y^* = \arg \max_k f_k(x)$
 - Example: Using SVM to train each binary classifier

Transformation to Binary

- One-vs.-One (OvO) reduction is to train $K(K-1)/2$ binary classifiers
 - For the (s, t) -th classifier:
 - Positive samples: all the points in class s
 - Negative samples: all the points in class t
 - $f_{s,t}(x)$ is the decision value for this classifier such that larger $f_{s,t}(x)$ implies that label s has higher probability than label t
 - Prediction:

$$f(x) = \arg \max_s \left(\sum_t f_{s,t}(x) \right)$$

- Example: using SVM to train each binary classifier

Softmax Regression

- Training data $\{(x^{(i)}, y^{(i)})\}_{i=1,2,\dots,m}$
- K different labels $\{1, 2, \dots, K\}$
- $y^{(i)} \in \{1, 2, \dots, K\}$ for $\forall i$
- Hypothesis function

$$h_{\theta}(x) = \begin{bmatrix} p(y = 1 \mid x, \theta) \\ p(y = 2 \mid x, \theta) \\ \vdots \\ p(y = K \mid x, \theta) \end{bmatrix} = \frac{1}{\sum_{k=1}^K \exp(\theta^{(k)T} x)} \begin{bmatrix} \exp(\theta^{(1)T} x) \\ \exp(\theta^{(2)T} x) \\ \vdots \\ \exp(\theta^{(K)T} x) \end{bmatrix}$$

where $\theta^{(1)}, \theta^{(2)}, \dots, \theta^{(K)} \in \mathbb{R}^n$ are the parameters of the softmax regression model

- Log-likelihood function

$$\begin{aligned}\ell(\theta) &= \sum_{i=1}^m \log p(y^{(i)} | x^{(i)}; \theta) \\ &= \sum_{i=1}^m \log \prod_{k=1}^K \left(\frac{\exp(\theta^{(k)T} x^{(i)})}{\sum_{k'=1}^K \exp(\theta^{(k')T} x^{(i)})} \right)^{\mathbb{I}(y^{(i)}=k)}\end{aligned}$$

where $\mathbb{I} : \{True, False\} \rightarrow \{0, 1\}$ is an indicator function

- Maximizing $\ell(\theta)$ through gradient ascent or Newton's method

Thanks!

Q & A