DS 4400

Machine Learning and Data Mining I
Spring 2024

David Liu
Khoury College of Computer Science
Northeastern University

February 6 2024

Outline

- Review of Cross Validation
- Logistic regression
  - Objective for logistic regression
  - Gradient descent training
  - Regularization
- Logistic regression lab
- Evaluation of classifiers
  - Accuracy, error, precision, recall
  - ROC curves and the AUC metric
  - Why multiple metrics

Announcements

Based on class feedback there will be frequent code demos / examples to complement lecture slides.

Many of these will be the textbook’s labs. However, in coming weeks I will likely create my own code demos as well.

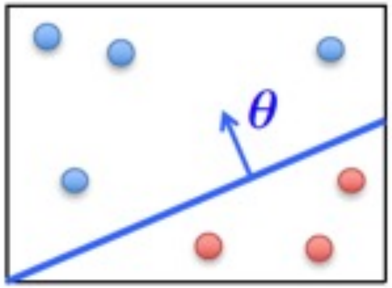
Lab Demo for Cross Validation and Regularization

LOGISTIC REGRESSION

Linear Classifiers

- **Linear classifiers:** represent decision boundary by hyperplane

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_d \end{bmatrix} \quad \boldsymbol{x}^\intercal = \begin{bmatrix} 1 & x_1 & \ldots & x_d \end{bmatrix}$$



$h_\theta(x) = f(\theta^T x)$ linear classifier

- If $\theta^T x > 0$ classify "Class 1"
- If $\theta^T x < 0$ classify "Class 0"

All the points x on the hyperplane satisfy: $\theta^T x = 0$

Logistic Regression

- Setup
  - Training data: $\{x_i, y_i\}$, for $i = 1, \dots, N$
  - Labels: $y_i \in \{0,1\}$
- Goals
  - Learn $h_\theta(x) = P(Y = 1|X = x)$
- Highlights
  - Probabilistic output
  - At the basis of more complex models (e.g., neural networks)
  - Supports regularization (Ridge, Lasso)
  - Can be trained with Gradient Descent

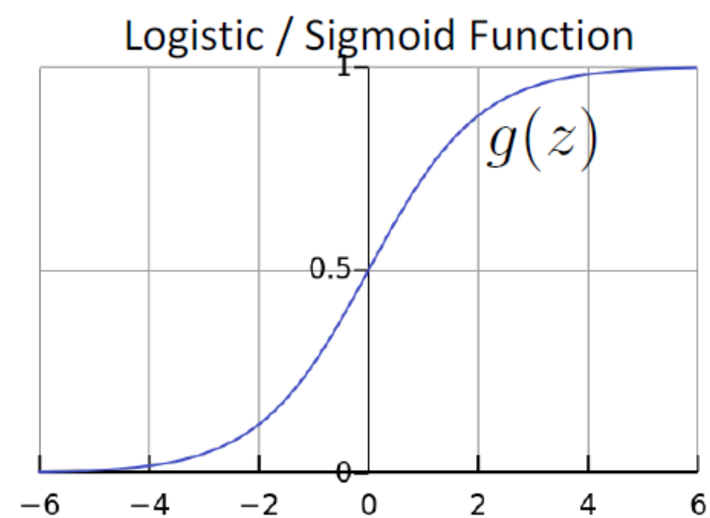
Logistic Regression

- Takes a probabilistic approach to learning discriminative functions (i.e., a classifier)
- $h_{\boldsymbol{\theta}}(\boldsymbol{x})$ should give $P(Y = 1 | X; \theta)$
  - Want $0 \leq h_{\boldsymbol{\theta}}(\boldsymbol{x}) \leq 1$
- Logistic regression model:

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = g\left(\boldsymbol{\theta}^{\mathsf{T}} \boldsymbol{x}\right)$$

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

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



Maximum Likelihood Estimation (MLE)

Training data $X = \{x_1, \ldots, x_N\}$, labels $Y = \{y_1, \ldots, y_N\}$

What is the likelihood of training data for parameter θ?

Define likelihood function $Max_\theta \, L(\theta) = P[Y|X;\theta]$

Assumption: *training labels are conditionally independent*

$$L(\theta) = \prod_{i=1}^{N} P[Y = y_i | X = x_i; \theta]$$

Log likelihood has the same maximum

$$\log L(\theta) = \sum_{i=1}^{N} \log P[Y = y_i | X = x_i; \theta]$$

MLE for Logistic Regression

$$\mathrm{P}(Y = y_i | X = x_i; \theta) = h_\theta(x_i)^{y_i} \left(1 - h_\theta(x_i)\right)^{1-y_i}$$

$$\theta_{MLE} = \mathrm{argmax}_\theta \sum_{i=1}^{N} \log P[Y = y_i | X = x_i; \theta]$$

$$= \mathrm{argmax}_\theta \sum_{i=1}^{N} y_\mathrm{i} \log h_\theta(x_i) + (1 - y_i) \log \left(1 - h_\theta(x_i)\right)$$

Logistic Regression Cross-Entropy Loss Objective

$$\min_\theta J(\theta)$$

$$J(\theta) = -\sum_{i=1}^{N} [y_\mathrm{i} \log h_\theta(x_i) + (1 - y_i) \log \left(1 - h_\theta(x_i)\right)]$$

Cross-Entropy Objective

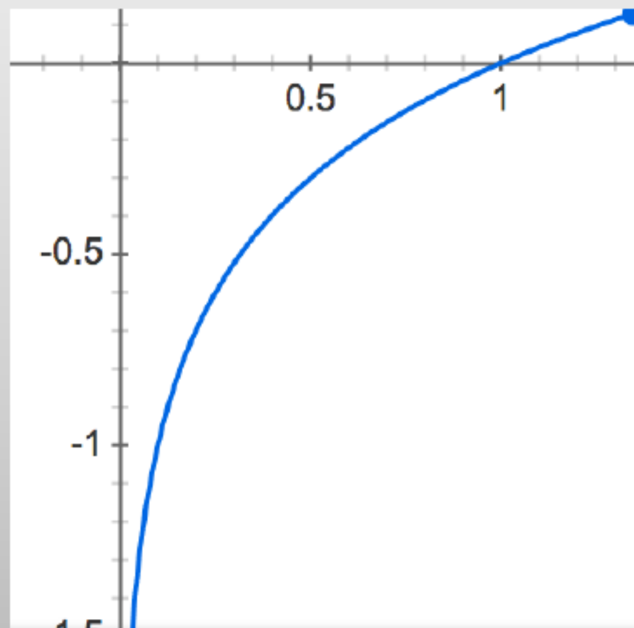
$$J(\theta) = -\sum_{i=1}^{N}[y_{\mathrm{i}}\log h_\theta(x_i) + (1 - y_i)\log\left(1 - h_\theta(x_i)\right)]$$

- Loss of a single instance:

Intuition

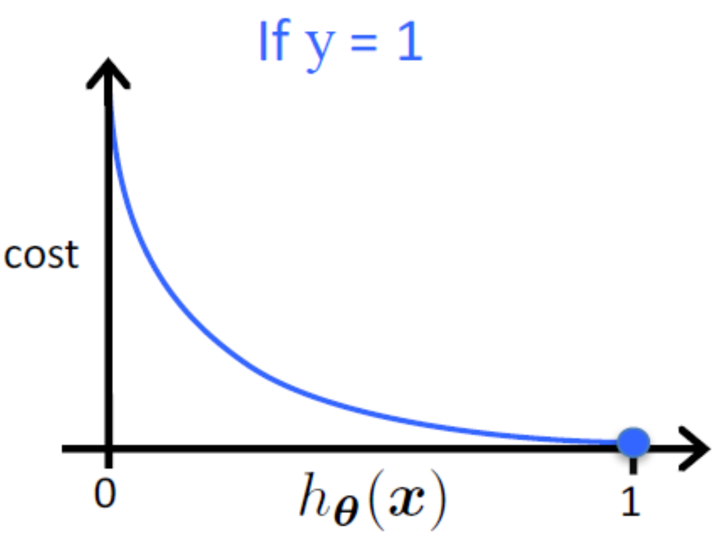
$$\text{loss } (h_{\boldsymbol{\theta}}(\boldsymbol{x}), y) = \begin{cases} -\log(h_{\boldsymbol{\theta}}(\boldsymbol{x})) & \text{if } y = 1 \\ -\log(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x})) & \text{if } y = 0 \end{cases}$$

Aside: Recall the plot of log(z)



Intuition

$$\text{loss}\ (h_{\boldsymbol{\theta}}(\boldsymbol{x}), y) = \begin{cases} \boxed{-\log(h_{\boldsymbol{\theta}}(\boldsymbol{x})) \quad \text{if } y = 1} \\ -\log(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x})) \quad \text{if } y = 0 \end{cases}$$

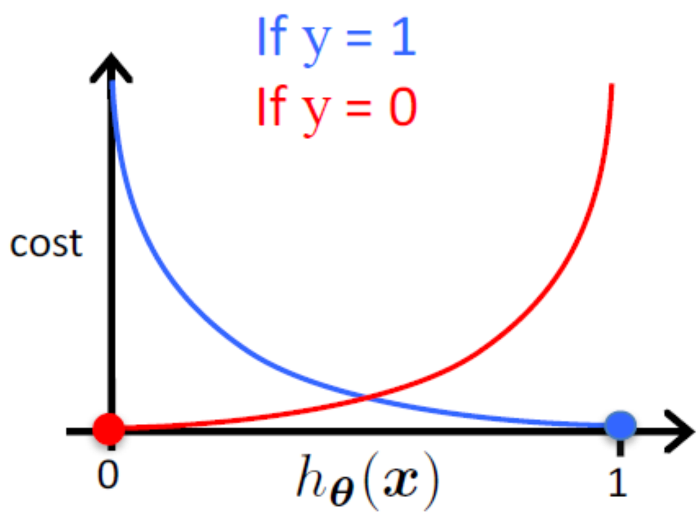


If y = 1

- loss = 0 if prediction is correct
- As $h_{\boldsymbol{\theta}}(\boldsymbol{x}) \to 0$, loss $\to \infty$
- Captures intuition that larger mistakes should get larger penalties
  - e.g., predict $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = 0$, but y = 1

Intuition

$$\text{loss}\ (h_{\boldsymbol{\theta}}(\boldsymbol{x}), y) = \begin{cases} \boxed{-\log(h_{\boldsymbol{\theta}}(\boldsymbol{x})) \quad \text{if } y = 1} \\ \boxed{-\log(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x})) \quad \text{if } y = 0} \end{cases}$$



If y = 0

- loss = 0 if prediction is correct
- As $(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x})) \to 0$, loss $\to \infty$
- Captures intuition that larger mistakes should get larger penalties

Gradient Descent for Logistic Regression

$$J(\theta) = -\sum_{i=1}^{N}[y_{\mathrm{i}}\log h_\theta(x_i) + (1-y_i)\log\left(1-h_\theta(x_i)\right)]$$

Want $\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$

- Initialize $\boldsymbol{\theta}$
- Repeat until convergence

$$\theta_j \leftarrow \theta_j - \alpha\frac{\partial}{\partial\theta_j}J(\boldsymbol{\theta})$$

simultaneous update
for j = 0 ... d

Gradient Computation

$$J(\theta) = -\sum_{i=1}^{N}[y_{\mathrm{i}}\log h_\theta(x_i) + (1-y_i)\log\left(1-h_\theta(x_i)\right)]$$

Derivative Facts to Know:

$$\frac{d}{dx}\log x = \frac{1}{x}$$

If $f(\theta^T x) = \frac{1}{1+e^\wedge-\theta^T x}$, then

$$\frac{df}{dx} = f(\theta^T X)\left(1-f(\theta^T X)\right)X$$

Computing Gradients

- Derivative of sigmoid
  - $g(z) = \frac{1}{1+e^{-z}}$; $g'(z) = \frac{e^{-z}}{(1+e^{-z})^2} = g(z)(1 - g(z))$
- Derivative of hypothesis
  - $h_\theta(x) = g(\theta^T x) = g(\theta_j x_j + \sum_{k \neq j} \theta_k x_k)$
  - $\frac{\partial h_\theta(x)}{\partial \theta_j} = \frac{\partial g(\theta^T x)}{\partial \theta_j} x_j = g(\theta^T x)\big(1 - g(\theta^T x)\big)x_j$
- Derivation of C_i
  - $\frac{\partial C_i}{\partial \theta_j} = y_i \frac{1}{h_\theta(x_i)} \mathrm{g}(\theta^T x_i)\big(1 - \mathrm{g}(\theta^T x_i)\big)x_{ij}$ -

    $(1 - y_i)\frac{1}{1-h_\theta(x_i)} \mathrm{g}(\theta^T x_i)\big(1 - \mathrm{g}(\theta^T x_i)\big)x_{ij}$

    $= \big(y_i - h_\theta(x_i)\big)x_{ij}$

Gradient Descent for Logistic Regression

$$J(\theta) = -\sum_{i=1}^{N} [y_{\mathrm{i}} \log h_\theta(x_i) + (1 - y_i) \log\left(1 - h_\theta(x_i)\right)]$$

Want $\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$

- Initialize $\boldsymbol{\theta}$
- Repeat until convergence (simultaneous update for j = 0 ... d)

$$\theta_0 \leftarrow \theta_0 - \alpha \sum_{i=1}^{N} (h_\theta(x_i) - y_i)$$

$$\theta_j \leftarrow \theta_j - \alpha \sum_{i=1}^{N} (h_\theta(x_i) - y_i) x_{ij}$$

Gradient Descent for Logistic Regression

Want $\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$

- Initialize $\boldsymbol{\theta}$
- Repeat until convergence (simultaneous update for j = 0 ... d)

$$\theta_0 \leftarrow \theta_0 - \alpha \sum_{i=1}^{N} (h_\theta(x_i) - y_i)$$

$$\theta_j \leftarrow \theta_j - \alpha \sum_{i=1}^{N} (h_\theta(x_i) - y_i) x_{ij}$$

This looks IDENTICAL to Linear Regression!

- However, the form of the model is very different:

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

Regularized Logistic Regression

$$J(\theta) = -\sum_{i=1}^{N}[y_\mathrm{i}\log h_\theta(x_i) + (1 - y_i)\log\left(1 - h_\theta(x_i)\right)]$$

- We can regularize logistic regression exactly as before:

$$\begin{aligned} J_{\text{regularized}}(\boldsymbol{\theta}) &= J(\boldsymbol{\theta}) + \lambda \sum_{j=1}^{d} \theta_j^2 \\ &= J(\boldsymbol{\theta}) + \lambda \|\boldsymbol{\theta}_{[1:d]}\|_2^2 \end{aligned}$$

L2 regularization