

DS 4400

Machine Learning and Data Mining I Spring 2022

Alina Oprea
Associate Professor
Khoury College of Computer Science
Northeastern University

February 16 2022

Project Proposal

- Project Title
- Project Team
- Problem Description
 - What is the prediction problem you are trying to solve?
- Dataset
 - Link to data, brief description, number of records, feature dimensionality (at least 20K records)
- Approach and methodology
 - Normalization
 - Feature selection
 - Machine learning models you will try (recommended ≥ 4)
 - Language and packages you plan to use
- Metrics (how you will evaluate your models)
- References
 - How did you find out about the dataset, did anyone else used the data for a similar prediction task
- **Project proposal due on March 7**

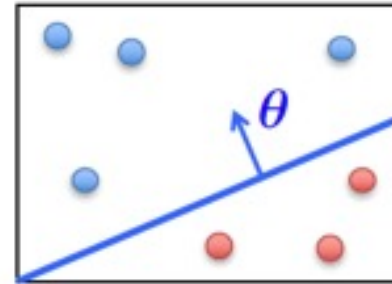
Outline

- 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

Linear Classifiers

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

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_d \end{bmatrix} \quad \mathbf{x}^\top = [1 \quad x_1 \quad \dots \quad x_d]$$



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

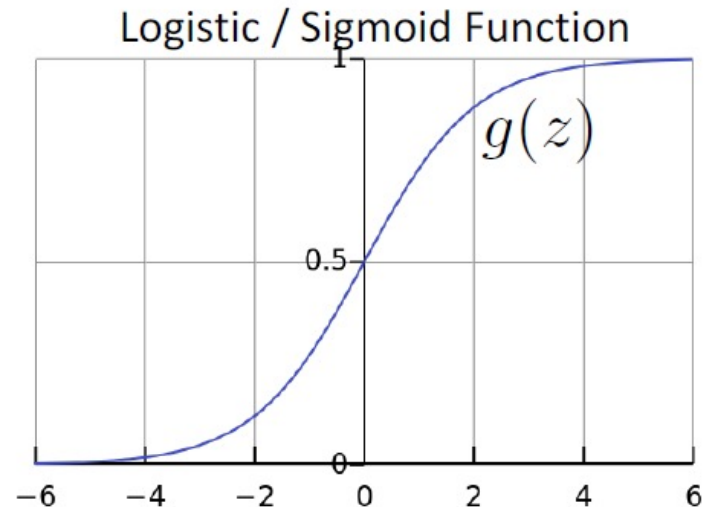
Logistic Regression

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

$$h_{\theta}(\mathbf{x}) = g(\boldsymbol{\theta}^T \mathbf{x})$$

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

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



Maximum Likelihood Estimation (MLE)

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

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

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

$$\begin{aligned}\theta_{MLE} &= \operatorname{argmax}_\theta \sum_{i=1}^N \log P[Y = y_i | X = x_i; \theta] \\ &= \operatorname{argmax}_\theta \sum_{i=1}^N y_i \log h_\theta(x_i) + (1 - y_i) \log (1 - h_\theta(x_i))\end{aligned}$$

Logistic Regression Cross-Entropy Loss Objective

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

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

Cross-Entropy Objective

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

- Loss of a single instance:

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

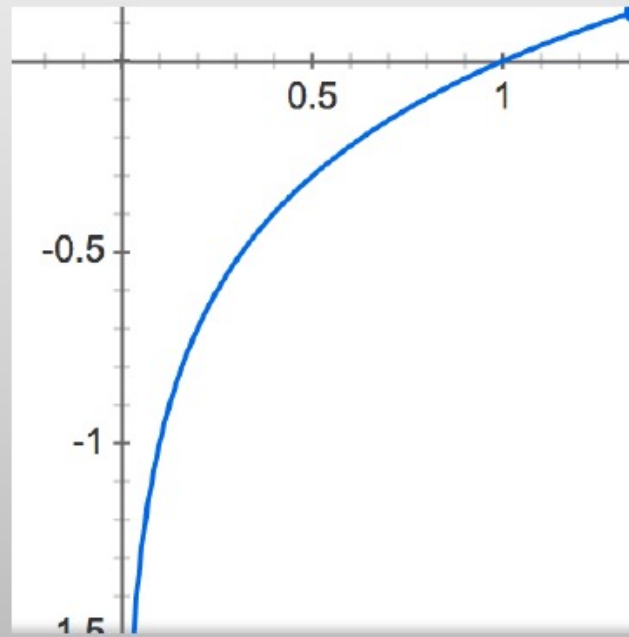
- Can re-write objective function as

$$J(\theta) = \sum_{i=1}^n \underbrace{\text{loss}(h_{\theta}(x_i), y_i)}_{\text{Cross-entropy loss}}$$

Intuition

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

Aside: Recall the plot of $\log(z)$



Intuition

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

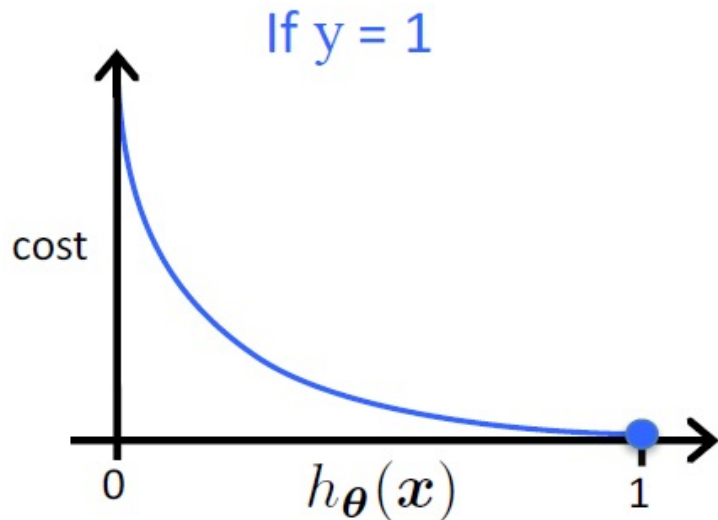
If $y = 1$

- $\text{loss} = 0$ if prediction is correct

- As $h_{\theta}(\mathbf{x}) \rightarrow 0$, $\text{loss} \rightarrow \infty$

- Captures intuition that larger mistakes should get larger penalties

- e.g., predict $h_{\theta}(\mathbf{x}) = 0$, but $y = 1$

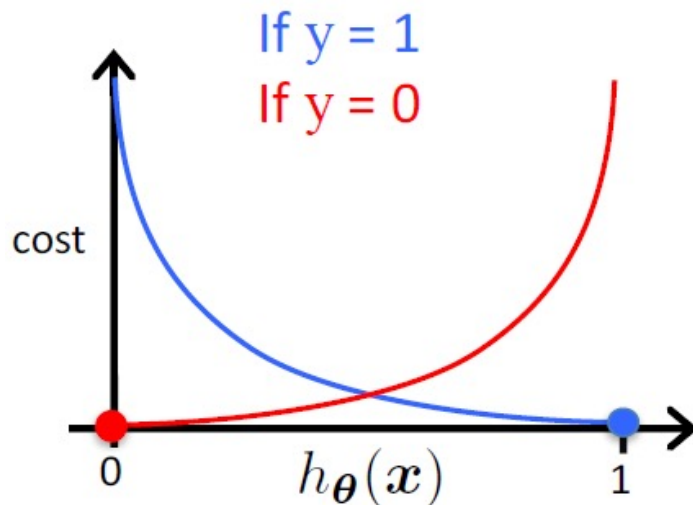


Intuition

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

If $y = 0$

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



Gradient Descent for Logistic Regression

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

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

- Initialize θ
- Repeat until convergence

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

simultaneous update
for $j = 0 \dots d$

Gradient Computation

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

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

$$- h_{\theta}(x) = g(\theta^T x) = g(\theta_j x_j + \sum_{k \neq j} \theta_k x_k)$$

$$- \frac{\partial h_{\theta}(x)}{\partial \theta_j} = \frac{\partial g(\theta^T x)}{\partial \theta_j} x_j = g(\theta^T x)(1 - g(\theta^T x))x_j$$

- Derivation of C_i

$$\begin{aligned} - \frac{\partial C_i}{\partial \theta_j} &= y_i \frac{1}{h_{\theta}(x_i)} g(\theta^T x_i)(1 - g(\theta^T x_i))x_{ij} - \\ &\quad (1 - y_i) \frac{1}{1-h_{\theta}(x_i)} g(\theta^T x_i)(1 - g(\theta^T x_i))x_{ij} \\ &= (y_i - h_{\theta}(x_i))x_{ij} \end{aligned}$$

Gradient Descent for Logistic Regression

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

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

- Initialize θ
- Repeat until convergence (simultaneous update for $j = 0 \dots d$)

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

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

Gradient Descent for Logistic Regression

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

- Initialize θ
- Repeat until convergence (simultaneous update for $j = 0 \dots d$)

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

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

This looks IDENTICAL to Linear Regression!

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

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

Regularized Logistic Regression

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

- We can regularize logistic regression exactly as before:

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

L2 regularization

Logistic Regression

Lab Example

Review

- Maximum Likelihood Estimation (MLE) is a general statistical method for parameter estimation
- Logistic regression is a linear classifier that predicts class probability
 - Cross-entropy objective derived with MLE
- Can be trained with Gradient Descent

Acknowledgements

- Slides made using resources from:
 - Andrew Ng
 - Eric Eaton
 - David Sontag
- Thanks!