ECE595 / STAT598: Machine Learning I
Lecture 14 Logistic Regression

Spring 2020

Stanley Chan

School of Electrical and Computer Engineering
Purdue University
In linear discriminant analysis (LDA), there are generally two types of approaches:

- **Generative approach**: Estimate model, then define the classifier.
- **Discriminative approach**: Directly define the classifier.
Outline

Discriminative Approaches
- Lecture 14 Logistic Regression 1
- Lecture 15 Logistic Regression 2

This lecture: Logistic Regression 1
- From Linear to Logistic
  - Motivation
  - Loss Function
  - Why not L2 Loss?
- Interpreting Logistic
  - Maximum Likelihood
  - Log-odd
- Convexity
  - Is logistic loss convex?
  - Computation
Geometry of Linear Regression

- The discriminant function $g(x)$ is linear
- The hypothesis function $h(x) = \text{sign}(g(x))$ is a unit step
Can we replace $g(x)$ by $\text{sign}(g(x))$?

How about a soft-version of $\text{sign}(g(x))$?

This gives a logistic regression.

$$h(x) = \frac{1}{1 + e^{-(w^T x + w_0)}}$$

$C_2 = \{x \mid h(x) < 1/2\}$

$C_1 = \{x \mid h(x) > 1/2\}$
Sigmoid Function

- The function
  \[ h(x) = \frac{1}{1 + e^{-g(x)}} = \frac{1}{1 + e^{-(w^T x + w_0)}} \]
  is called a sigmoid function.
- Its 1D form is
  \[ h(x) = \frac{1}{1 + e^{-a(x-x_0)}}, \quad \text{for some } a \text{ and } x_0, \]
- \( a \) controls the transient speed
- \( x_0 \) controls the cutoff location

![Graphs showing the effect of large and small a on the sigmoid function](image_url)
Sigmoid Function

- Note that
  
  \[ h(x) \to 1, \quad \text{as} \quad x \to \infty, \]
  
  \[ h(x) \to 0, \quad \text{as} \quad x \to -\infty, \]

- So \( h(x) \) can be regarded as a “probability”.

\[
\begin{array}{c}
  -\infty \quad \text{x} \quad x_0 \quad x \to \infty \\
\end{array}
\]
Sigmoid Function

- Derivative is

\[
\frac{d}{dx} \left( \frac{1}{1 + e^{-a(x-x_0)}} \right) = - \left( 1 + e^{-a(x-x_0)} \right)^{-2} \left( e^{-a(x-x_0)} \right) (-a) \\
= a \left( \frac{e^{-a(x-x_0)}}{1 + e^{-a(x-x_0)}} \right) \left( \frac{1}{1 + e^{-a(x-x_0)}} \right) \\
= a \left( 1 - \frac{1}{1 + e^{-a(x-x_0)}} \right) \left( \frac{1}{1 + e^{-a(x-x_0)}} \right) \\
= a[1 - h(x)][h(x)].
\]

- Since \(0 < h(x) < 0\), we have \(0 < 1 - h(x) < 1\).
- Therefore, the derivative is always positive.
- So \(h\) is an increasing function.
- Hence \(h\) can be considered as a “CDF”. 
Sigmoid Function

Input training data

Predictions for $\theta = [0.26105, 3.0097, -2.1347]$ - accuracy: 98.4%

Decision boundary defined by $\theta$

Contours of equal probability defined by $\theta$

Probability map defined by $\theta$

Probability map defined by $\theta$
Can we replace $g(x)$ by $\text{sign}(g(x))$?
How about a soft-version of $\text{sign}(g(x))$?
This gives a logistic regression.
All discriminant algorithms have a **Training Loss Function**

\[
J(\theta) = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}(g(x_n), y_n).
\]

In linear regression,

\[
J(\theta) = \frac{1}{N} \sum_{n=1}^{N} (g(x_n) - y_n)^2
\]

\[
= \frac{1}{N} \sum_{n=1}^{N} (w^T x_n + w_0 - y_n)^2
\]

\[
= \frac{1}{N} \left\| \begin{bmatrix} x^T_1 & 1 \\ \vdots & \vdots \\ x^T_N & 1 \end{bmatrix} \begin{bmatrix} w \\ w_0 \end{bmatrix} - \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} \right\|^2 = \frac{1}{N} \left\| A\theta - y \right\|^2.
\]
Training Loss for Logistic Regression

\[ J(\theta) = \sum_{n=1}^{N} \mathcal{L}(h_\theta(x_n), y_n) \]

\[ = \sum_{n=1}^{N} - \left\{ y_n \log h_\theta(x_n) + (1 - y_n) \log(1 - h_\theta(x_n)) \right\} \]

- This loss is also called the **cross-entropy loss**.
- Why do we want to choose this cost function?
- Consider two cases

\[ y_n \log h_\theta(x_n) = \begin{cases} 0, & \text{if } y_n = 1, \text{ and } h_\theta(x_n) = 1, \\ -\infty, & \text{if } y_n = 1, \text{ and } h_\theta(x_n) = 0, \end{cases} \]

\[ (1 - y_n)(1 - \log h_\theta(x_n)) = \begin{cases} 0, & \text{if } y_n = 0, \text{ and } h_\theta(x_n) = 0, \\ -\infty, & \text{if } y_n = 0, \text{ and } h_\theta(x_n) = 1. \end{cases} \]

- No solution if mismatch
Why Not L2 Loss?

Why not use L2 loss?

\[ J(\theta) = \sum_{n=1}^{N} (h_\theta(x_n) - y_n)^2 \]

Let’s look at the 1D case:

\[ J(\theta) = \left( \frac{1}{1 + e^{-\theta x}} - y \right)^2. \]

This is NOT convex!

How about the logistic loss?

\[ J(\theta) = y \log \left( \frac{1}{1 + e^{-\theta x}} \right) + (1 - y) \log \left( 1 - \frac{1}{1 + e^{-\theta x}} \right) \]

This is convex!
Why Not L2 Loss?

- **Experiment**: Set $x = 1$ and $y = 1$.
- **Plot** $J(\theta)$ as a function of $\theta$.

So the L2 loss is not convex, but the logistic loss is concave (negative is convex).

If you do gradient descent on L2, you will be trapped at local minima.
Outline

Discriminative Approaches

- Lecture 14 Logistic Regression 1
- Lecture 15 Logistic Regression 2

This lecture: Logistic Regression 1

- From Linear to Logistic
  - Motivation
  - Loss Function
  - Why not L2 Loss?
- Interpreting Logistic
  - Maximum Likelihood
  - Log-odd
- Convexity
  - Is logistic loss convex?
  - Computation
The Maximum-Likelihood Perspective

- We can show that

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

$$= \arg\min_{\theta} \sum_{n=1}^{N} \left\{ y_n \log h_\theta(x_n) + (1 - y_n) \log(1 - h_\theta(x_n)) \right\}$$

$$= \arg\min_{\theta} - \log \left( \prod_{n=1}^{N} h_\theta(x_n)^{y_n} (1 - h_\theta(x_n))^{1-y_n} \right)$$

$$= \arg\max_{\theta} \prod_{n=1}^{N} \left\{ h_\theta(x_n)^{y_n} (1 - h_\theta(x_n))^{1-y_n} \right\}.$$ 

- This is maximum-likelihood for a Bernoulli random variable $y_n$
- The underlying probability is $h_\theta(x_n)$
Interpreting $h(x_n)$

- Maximum-likelihood Bernoulli:

$$\theta^* = \arg\max_{\theta} \prod_{n=1}^{N} \left\{ h_\theta(x_n)^{y_n}(1 - h_\theta(x_n))^{1-y_n} \right\}.$$

- We can interpret $h_\theta(x_n)$ as a probability $p$. So:

$$h_\theta(x_n) = p, \quad \text{and} \quad 1 - h_\theta(x_n) = 1 - p.$$

- But $p$ is a function of $x_n$. So how about

$$h_\theta(x_n) = p(x_n), \quad \text{and} \quad 1 - h_\theta(x_n) = 1 - p(x_n).$$

- And this probability is “after” you see $x_n$. So how about

$$h_\theta(x_n) = p(1 \mid x_n), \quad \text{and} \quad 1 - h_\theta(x_n) = 1 - p(1 \mid x_n) = p(0 \mid x_n).$$

- So $h_\theta(x_n)$ is the **posterior** of observing $x_n$. 


Log-Odds

Let us rewrite $J$ as

$$J(\theta) = \sum_{n=1}^{N} - \left\{ y_n \log h_\theta(x_n) + (1 - y_n) \log(1 - h_\theta(x_n)) \right\}$$

$$= \sum_{n=1}^{N} - \left\{ y_n \log \left( \frac{h_\theta(x_n)}{1 - h_\theta(x_n)} \right) + \log(1 - h_\theta(x_n)) \right\}$$

In statistics, the term $\log \left( \frac{h_\theta(x_n)}{1 - h_\theta(x_n)} \right)$ is called the log-odd.

If we put $h_\theta(x_n) = \frac{1}{1 + e^{-\theta^T x}}$, we can show that

$$\log \left( \frac{h_\theta(x)}{1 - h_\theta(x)} \right) = \log \left( \frac{1}{1 + e^{-\theta^T x}} \right) = \log \left( \frac{e^{\theta^T x}}{1 + e^{\theta^T x}} \right) = \theta^T x.$$  

Logistic regression is linear in the log-odd.
Outline

Discriminative Approaches

- Lecture 14 Logistic Regression 1
- Lecture 15 Logistic Regression 2

This lecture: Logistic Regression 1

- From Linear to Logistic
  - Motivation
  - Loss Function
  - Why not L2 Loss?
- Interpreting Logistic
  - Maximum Likelihood
  - Log-odd
- Convexity
  - Is logistic loss convex?
  - Computation
Recall that

\[ J(\theta) = \sum_{n=1}^{n} - \left\{ y_n \log \left( \frac{h_\theta(x_n)}{1 - h_\theta(x_n)} \right) + \log(1 - h_\theta(x_n)) \right\} \]

- The first term is linear, so it is convex.
- The second term: Gradient:

\[
\nabla_\theta [-\log(1 - h_\theta(x))] = -\nabla_\theta \left[ \log \left( 1 - \frac{1}{1 + e^{-\theta^T x}} \right) \right] \\
= -\nabla_\theta \left[ \log \frac{e^{-\theta^T x}}{1 + e^{-\theta^T x}} \right] = -\nabla_\theta \left[ \log e^{-\theta^T x} - \log(1 + e^{-\theta^T x}) \right] \\
= -\nabla_\theta \left[ -\theta^T x - \log(1 + e^{-\theta^T x}) \right] = x + \nabla_\theta \left[ \log \left( 1 + e^{-\theta^T x} \right) \right] \\
= x + \left( \frac{-e^{-\theta^T x}}{1 + e^{-\theta^T x}} \right)x = h_\theta(x)x.
\]
Convexity of Logistic Training Loss

- Gradient of second term is

\[ \nabla_\theta [- \log(1 - h_\theta(x))] = h_\theta(x)x. \]

- Hessian is:

\[ \nabla^2_\theta [- \log(1 - h_\theta(x))] = \nabla_\theta [h_\theta(x)x] \]

\[ = \nabla_\theta \left[ \left( \frac{1}{1 + e^{-\theta^T x}} \right) x \right] \]

\[ = \left( \frac{1}{(1 + e^{-\theta^T x})^2} \right) \left( -e^{-\theta^T x} \right) xx^T \]

\[ = \left( \frac{1}{1 + e^{-\theta^T x}} \right) \left( 1 - \frac{1}{1 + e^{-\theta^T x}} \right) xx^T \]

\[ = h_\theta(x)[1 - h_\theta(x)]xx^T. \]
Convexity of Logistic Training Loss

- For any \( \mathbf{v} \in \mathbb{R}^d \), we have that

\[
\mathbf{v}^T \nabla^2_{\theta} [\log(1 - h_\theta(x))] \mathbf{v} = \mathbf{v}^T \left[ h_\theta(x)[1 - h_\theta(x)] \mathbf{x} \mathbf{x}^T \right] \mathbf{v} \\
= (h_\theta(x)[1 - h_\theta(x)]) \| \mathbf{v}^T \mathbf{x} \|^2 \geq 0.
\]

- Therefore the Hessian is positive semi-definite.
- So \( -\log(1 - h_\theta(x)) \) is convex in \( \theta \).
- Conclusion: The training loss function

\[
J(\theta) = \sum_{n=1}^{n} \left\{ y_n \log \left( \frac{h_\theta(x_n)}{1 - h_\theta(x_n)} \right) + \log(1 - h_\theta(x_n)) \right\}
\]

is convex in \( \theta \).
- So we can use convex optimization algorithms to find \( \theta \).
Convex Optimization for Logistic Regression

- We can use CVX to solve the logistic regression problem.
- But it requires some re-organization of the equations:

\[
J(\theta) = \sum_{n=1}^{N} - \left\{ y_n \theta^T x_n + \log(1 - h_\theta(x_n)) \right\}
\]

\[
= \sum_{n=1}^{N} - \left\{ y_n \theta^T x_n + \log \left(1 - \frac{e^{\theta^T x_n}}{1 + e^{\theta^T x_n}}\right) \right\}
\]

\[
= \sum_{n=1}^{N} - \left\{ y_n \theta^T x_n - \log \left(1 + e^{\theta^T x_n}\right) \right\}
\]

\[
= - \left\{ \left( \sum_{n=1}^{N} y_n x_n \right)^T \theta - \sum_{n=1}^{N} \log \left(1 + e^{\theta^T x_n}\right) \right\}.
\]

- The last term is a sum of log-sum-exp: \(\log(e^0 + e^{\theta^T x})\).
Convex Optimization for Logistic Regression
Reading List

**Logistic Regression** (Machine Learning Perspective)
- Chris Bishop’s *Pattern Recognition*, Chapter 4.3
- Hastie-Tibshirani-Friedman’s *Elements of Statistical Learning*, Chapter 4.4
- Stanford CS 229 Discriminant Algorithms
  [http://cs229.stanford.edu/]
- CMU Lecture [https://www.stat.cmu.edu/~cshalizi/uADA/12/lectures/ch12.pdf]
- Stanford Language Processing
  [https://web.stanford.edu/~jurafsky/slp3/](https://web.stanford.edu/~jurafsky/slp3/) (Lecture 5)

**Logistic Regression** (Statistics Perspective)
- Duke Lecture [https://www2.stat.duke.edu/courses/Spring13/sta102.001/Lec/Lec20.pdf]
- Princeton Lecture
  [https://data.princeton.edu/wws509/notes/c3.pdf]