10-301/601: Introduction to Machine Learning Lecture 9 – Logistic Regression

Matt Gormley & Henry Chai

9/23/24

Front Matter

- Announcements:
  - HW3 released 9/16, due 9/23 (today!) at 11:59 PM
    - **Only two grace days allowed on HW3**
  - Exam 1 on 9/30 (next Monday) from 6:30 PM - 8:30 PM
    - If you have a conflict, you must complete the Exam conflict form by 9/23 (today!) at 1 PM
    - Exam 1 practice problems released on the course website, under Coursework

Probabilistic Learning

- Previously:
  - (Unknown) Target function, $c^*: \mathcal{X} \rightarrow \mathcal{Y}$
  - Classifier, $h : \mathcal{X} \rightarrow \mathcal{Y}$
  - Goal: find a classifier, h, that best approximates c^*
- Now:
  - (Unknown) Target *distribution*, $y \sim p^*(Y|\boldsymbol{x})$
  - Distribution, $p(Y|\boldsymbol{x})$
  - Goal: find a distribution, p, that best approximates p^*

Likelihood

- Given N independent, identically distribution (iid) samples $\mathcal{D} = \{x^{(1)}, \ldots, x^{(N)}\}$ of a random variable X
  - If X is discrete with probability mass function (pmf) $p(X|\theta)$, then the *likelihood* of $\mathcal{D}$ is

$$L(\theta) = \prod_{n=1}^{N} p(x^{(n)}|\theta)$$

  - If X is continuous with probability density function (pdf) $f(X|\theta)$, then the *likelihood* of $\mathcal{D}$ is

$$L(\theta) = \prod_{n=1}^{N} f(x^{(n)}|\theta)$$

Log-Likelihood

- Given N independent, identically distribution (iid) samples $\mathcal{D} = \{x^{(1)}, \ldots, x^{(N)}\}$ of a random variable X
  - If X is discrete with probability mass function (pmf) $p(X|\theta)$, then the *log-likelihood* of $\mathcal{D}$ is

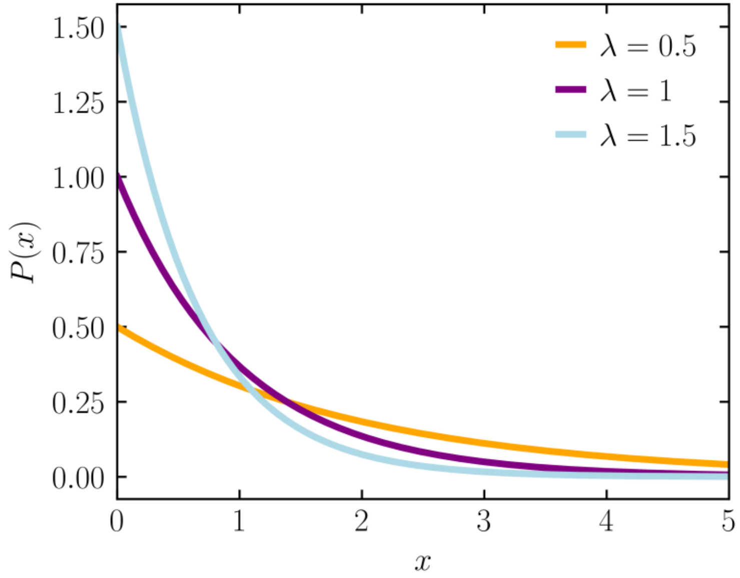
$$\ell(\theta) = \log \prod_{n=1}^{N} p(x^{(n)}|\theta) = \sum_{n=1}^{N} \log p(x^{(n)}|\theta)$$

  - If X is continuous with probability density function (pdf) $f(X|\theta)$, then the *log-likelihood* of $\mathcal{D}$ is

$$\ell(\theta) = \log \prod_{n=1}^{N} f(x^{(n)}|\theta) = \sum_{n=1}^{N} \log f(x^{(n)}|\theta)$$

Maximum Likelihood Estimation (MLE)

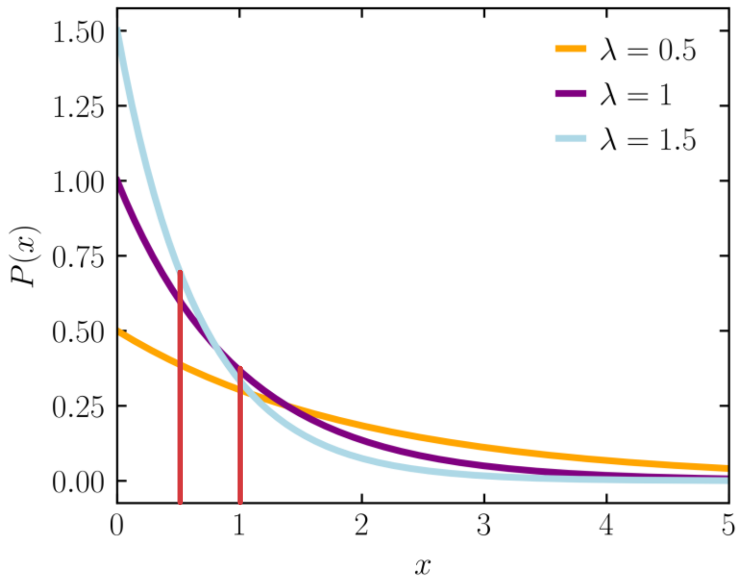
- Insight: every valid probability distribution has a finite amount of probability mass as it must sum/integrate to 1
- Idea: set the parameter(s) so that the likelihood of the samples is maximized
- Intuition: assign as much of the (finite) probability mass to the observed data *at the expense of unobserved data*
- Example: the exponential distribution



Source: https://en.wikipedia.org/wiki/Exponential_distribution#/media/File:Exponential_probability_density.svg

Maximum Likelihood Estimation (MLE)

- Insight: every valid probability distribution has a finite amount of probability mass as it must sum/integrate to 1
- Idea: set the parameter(s) so that the likelihood of the samples is maximized
- Intuition: assign as much of the (finite) probability mass to the observed data *at the expense of unobserved data*
- Example: the exponential distribution

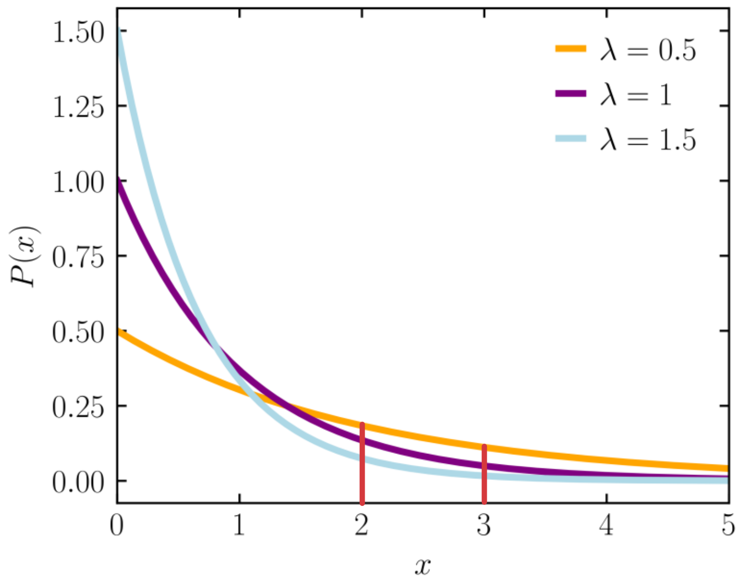


$\{x^{(1)} = 0.5, x^{(2)} = 1\}$

Source: https://en.wikipedia.org/wiki/Exponential_distribution#/media/File:Exponential_probability_density.svg

Maximum Likelihood Estimation (MLE)

- Insight: every valid probability distribution has a finite amount of probability mass as it must sum/integrate to 1
- Idea: set the parameter(s) so that the likelihood of the samples is maximized
- Intuition: assign as much of the (finite) probability mass to the observed data *at the expense of unobserved data*
- Example: the exponential distribution



$\{x^{(1)} = 2, x^{(2)} = 3\}$

Source: https://en.wikipedia.org/wiki/Exponential_distribution#/media/File:Exponential_probability_density.svg

Exponential Distribution MLE

- The pdf of the exponential distribution is

$$f(x|\lambda) = \lambda e^{-\lambda x}$$

- Given N iid samples $\{x^{(1)}, \dots, x^{(N)}\}$, the likelihood is

$$L(\lambda) = \prod_{i=1}^{N} f(x^{(i)}|\lambda) = \prod_{i=1}^{N} \lambda e^{-\lambda x^{(i)}}$$

$$\log(ab) = \log a + \log b$$

Exponential Distribution MLE

- The pdf of the exponential distribution is

$$f(x|\lambda) = \lambda e^{-\lambda x}$$

- Given N iid samples $\{x^{(1)}, \ldots, x^{(N)}\}$, the log-likelihood is

$$\ell(\lambda) = \sum_{i=1}^{N} \log f(x^{(i)}|\lambda) = \sum_{i=1}^{N} \log \lambda e^{-\lambda x^{(i)}}$$

$$= \sum_{i=1}^{N} \left(\log \lambda + (-\lambda x^{(i)}) \right)$$

$$= N \log \lambda - \lambda \sum_{i=1}^{N} x^{(i)}$$

$$\frac{\partial \ell}{\partial \lambda} = \frac{N}{\lambda} - \sum_{i=1}^{N} x^{(i)} \Rightarrow \frac{\partial^2 \ell}{\partial \lambda^2} = -\frac{N}{\lambda^2} < 0$$

$$\Rightarrow \frac{N}{\hat{\lambda}} - \sum_{i=1}^{N} x^{(i)} = 0 \Rightarrow \hat{\lambda} = \frac{N}{\sum_{i=1}^{N} x^{(i)}}$$

Building a Probabilistic Classifier

- Define a decision rule
  - Given a test data point $\boldsymbol{x}'$, predict its label $\hat{y}$ using the posterior distribution $P(Y = y|\boldsymbol{x}')$
  - Common choice: $\hat{y} = \underset{y}{\operatorname{argmax}} P(Y = y|\boldsymbol{x}')$
- Idea: model $P(Y|\boldsymbol{x})$ as some parametric function of $\boldsymbol{x}$

Modelling the Posterior

- Suppose we have binary labels $y \in \{0,1\}$ and D-dimensional inputs $\boldsymbol{x} = [1, x_1, \dots, x_D]^T \in \mathbb{R}^{D+1}$

  (1 prepended to $\boldsymbol{x}$)

- **Assume**

$$P(Y = 1 | \boldsymbol{x}, \boldsymbol{\theta}) = \sigma(\boldsymbol{\theta}^T \boldsymbol{x}) = \frac{1}{1 + \exp(-\boldsymbol{\theta}^T \boldsymbol{x})} = \frac{\exp(\boldsymbol{\theta}^T \boldsymbol{x})}{\exp(\boldsymbol{\theta}^T \boldsymbol{x}) + 1}$$

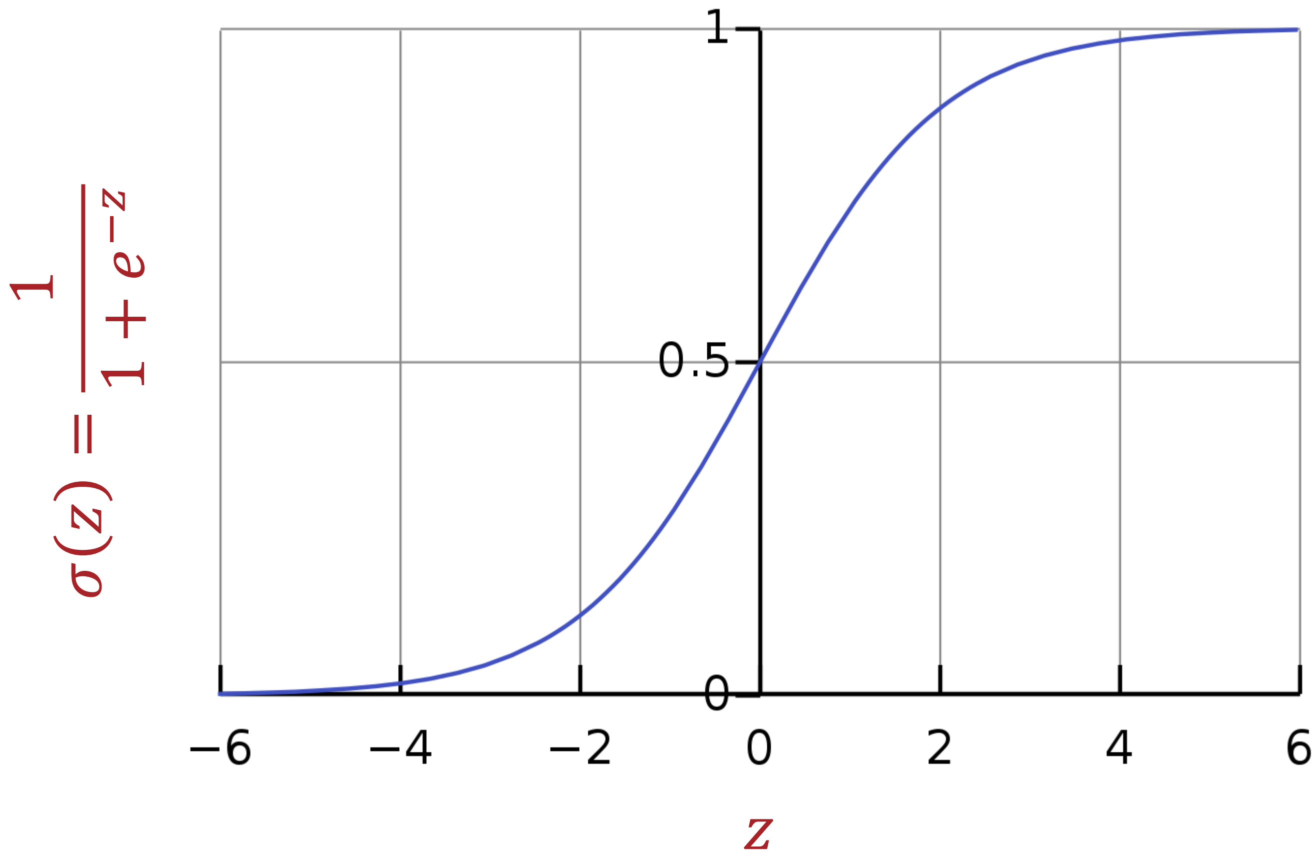
- This implies two useful facts:

1. $P(Y=0|x,\theta) = 1 - P(Y=1|x,\theta) = \frac{\exp(\theta^T x)+1}{\exp(\theta^T x)+1} - \frac{\exp(\theta^T x)}{\exp(\theta^T x)+1} = \frac{1}{\exp(\theta^T x)+1}$

2. $\frac{P(Y=1|x,\theta)}{P(Y=0|x,\theta)} = \exp(\theta^T x) \Rightarrow$ log odds is linear in x!

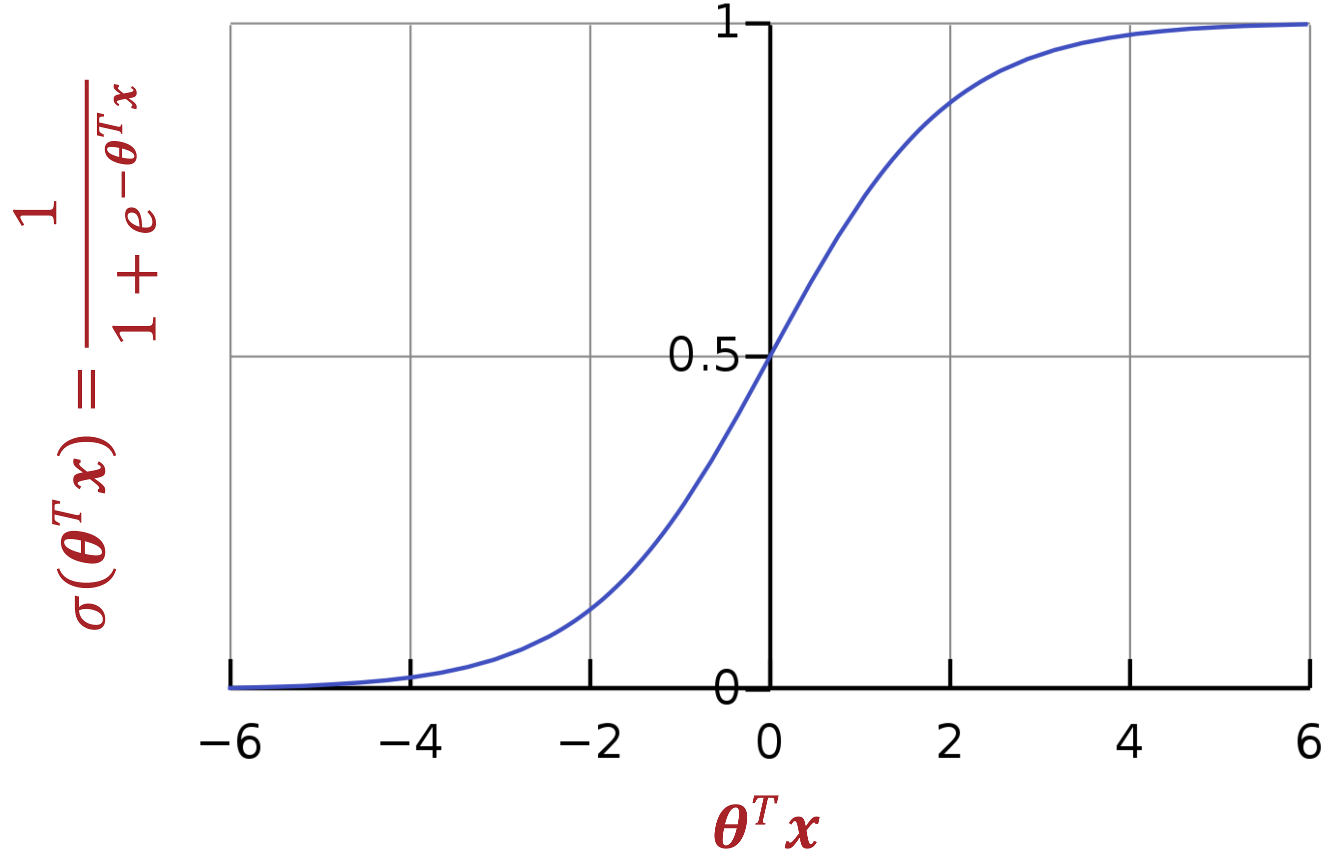
   $\log(\exp(\theta^T x)) = \theta^T x$

Logistic Function



Source: https://en.wikipedia.org/wiki/Logistic_function#/media/File:Logistic-curve.svg

Why use the Logistic Function?



1. $\sigma : \mathbb{R} \mapsto (0, 1)$
2. differentiable everywhere
3. the decision boundary is linear in x!

Source: https://en.wikipedia.org/wiki/Logistic_function#/media/File:Logistic-curve.svg

Logistic Regression Decision Boundary

$$\hat{y} = \begin{cases} 1 \text{ if } P(Y = 1|\boldsymbol{x}, \boldsymbol{\theta}) \geq \frac{1}{2} \\ 0 \text{ otherwise} \end{cases}$$

$$P(Y=1|x,\theta) = \frac{1}{2} = \frac{1}{1+\exp(-\theta^T x)}$$

$$\Rightarrow 1+\exp(-\theta^T x) = 2$$

$$\Rightarrow -\theta^T x = \log(2-1)$$

$$\Rightarrow \theta^T x = 0$$

Logistic Regression Decision Boundary

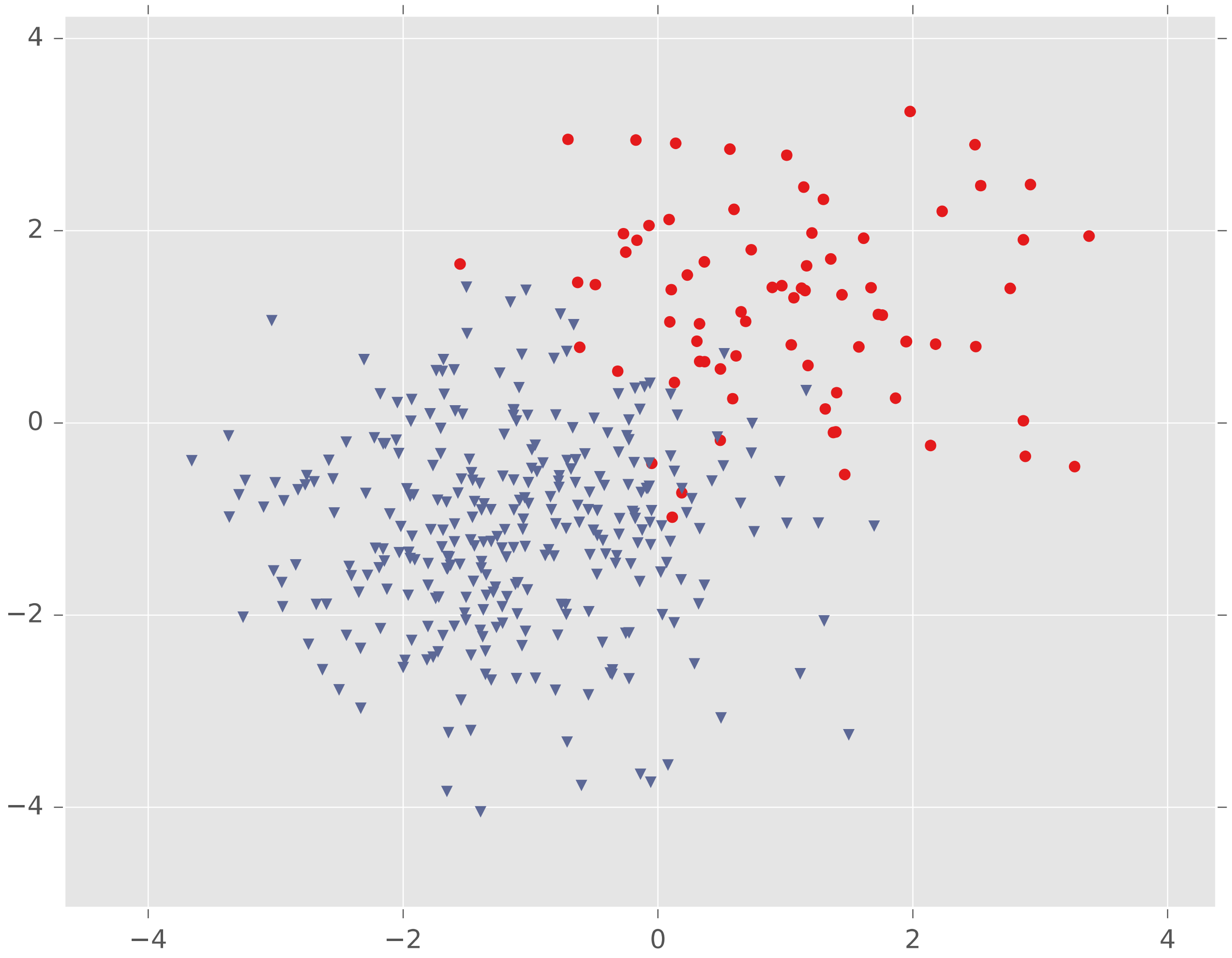


Figure courtesy of Matt Gormley

Logistic Regression Decision Boundary

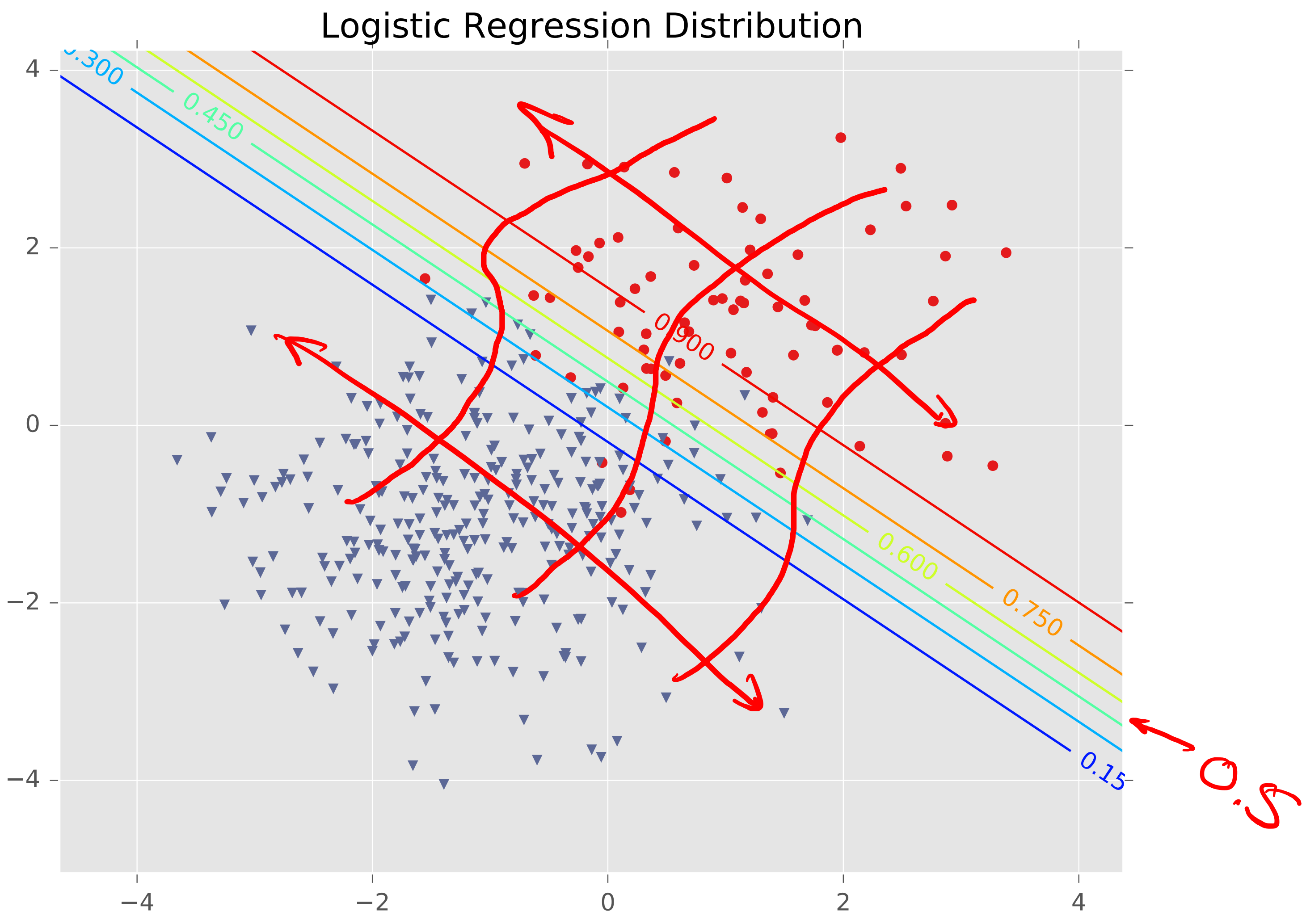


Figure courtesy of Matt Gormley

Logistic Regression Decision Boundary

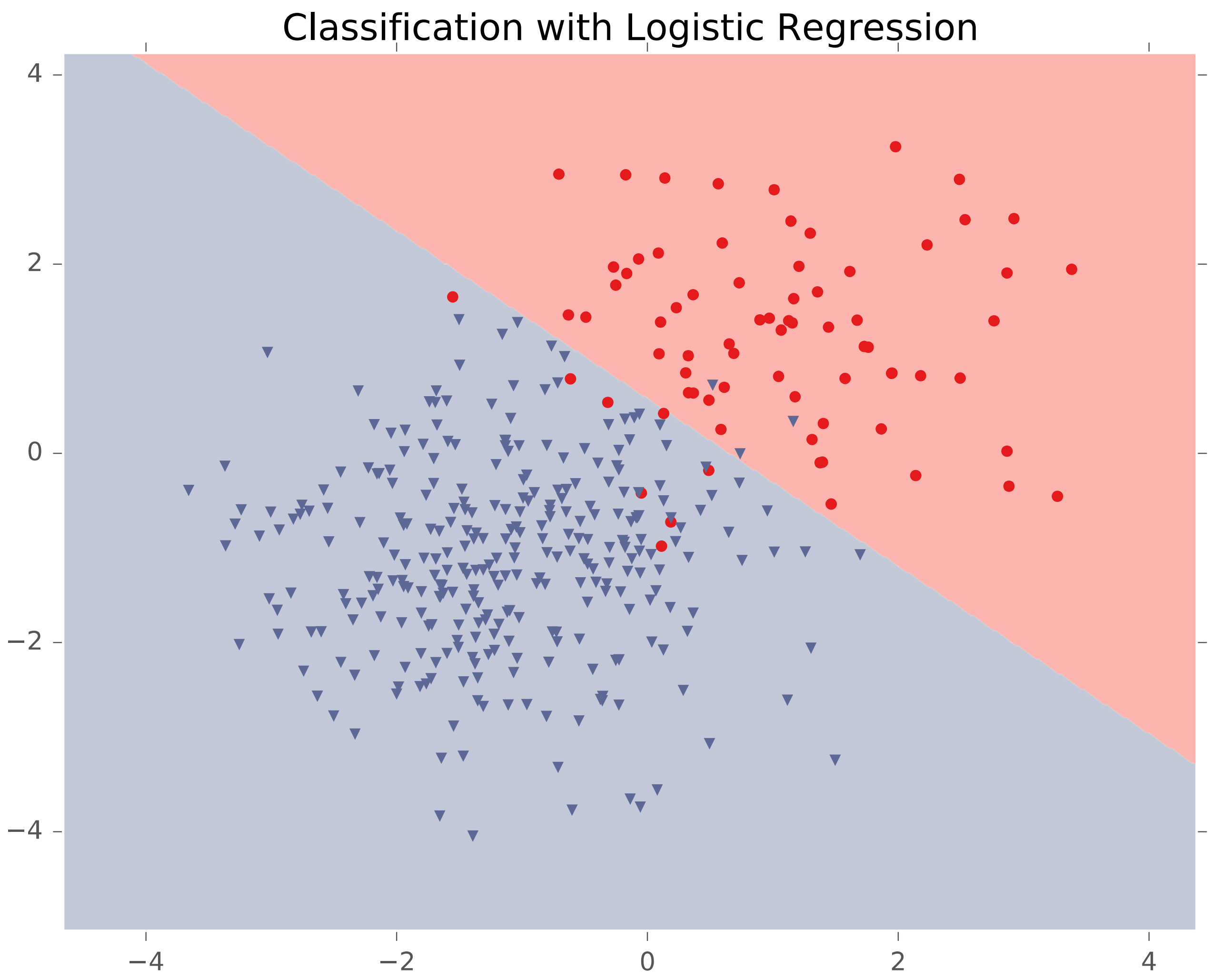


Figure courtesy of Matt Gormley

$$\log(a^b c^d) = b \log a + d \log c$$

Setting the Parameters via Minimum Negative Conditional (log-)Likelihood Estimation (MCLE)

$$P(y^{(1)}, x^{(1)}, y^{(2)}, x^{(2)}, \ldots)$$

$$\ell(\theta) = -\log P(y^{(1)}, y^{(2)}, \ldots, y^{(N)} \mid x^{(1)}, x^{(2)}, \ldots, x^{(N)}, \theta)$$

$$= -\log \prod_{i=1}^{N} P(y^{(i)} \mid x^{(i)}, \theta) = -\sum_{i=1}^{N} \log P(y^{(i)} \mid x^{(i)}, \theta)$$

$$= -\sum_{i=1}^{N} \log\left(P(Y=1 \mid x^{(i)}, \theta)^{y^{(i)}} \, P(Y=0 \mid x^{(i)}, \theta)^{1-y^{(i)}} \right)$$

$$= -\sum_{i=1}^{N} y^{(i)} \log P(Y=1 \mid x^{(i)}, \theta) + (1-y^{(i)}) \log P(Y=0 \mid x^{(i)}, \theta)$$

$$= -\sum_{i=1}^{N} y^{(i)} \left(\underbrace{\log \frac{P(Y=1 \mid x^{(i)}, \theta)}{P(Y=0 \mid x^{(i)}, \theta)}}_{\theta^T x^{(i)}} \right) + \log P(Y=0 \mid x^{(i)}, \theta)$$

$$= -\sum_{i=1}^{N} y^{(i)} \theta^T x^{(i)} + \log\left((1 + \exp(\theta^T x^{(i)}))^{-1} \right)$$

$$J(\theta) = \frac{1}{N} \ell(\theta) = -\frac{1}{N} \sum_{i=1}^{N} y^{(i)} \theta^T x^{(i)} - \log(1 + \exp(\theta^T x^{(i)}))$$

Key Takeaway: This objective function is convex but we cannot solve for the optimal parameters in closed form

Minimizing the Negative Conditional (log-)Likelihood

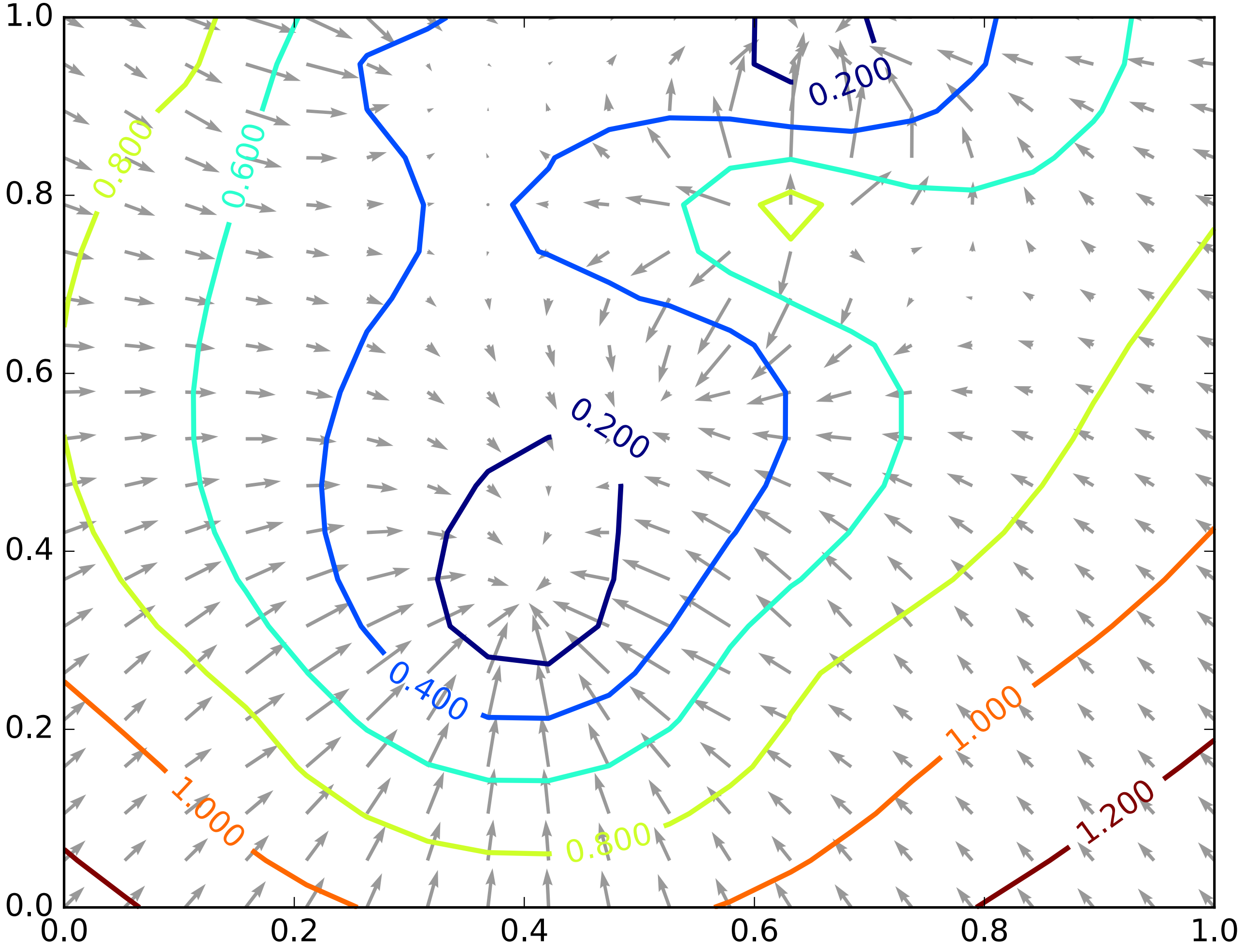
$$J(\boldsymbol{\theta}) = -\frac{1}{N}\sum_{n=1}^{N} y^{(n)}\boldsymbol{\theta}^T\boldsymbol{x}^{(n)} - \log\left(1 + \exp(\boldsymbol{\theta}^T\boldsymbol{x}^{(n)})\right)$$

$$\nabla_\theta J(\theta) = -\frac{1}{N}\sum_{n=1}^{N} y^{(n)}x^{(n)} - \frac{\exp(\theta^T x^{(n)})}{1+\exp(\theta^T x^{(n)})}x^{(n)}$$

$$= \frac{1}{N}\sum_{n=1}^{N}\left(\underbrace{\frac{\exp(\theta^T x^{(n)})}{1+\exp(\theta^T x^{(n)})}}_{P(Y=1 \mid x^{(n)}, \theta)} - y^{(n)}\right)x^{(n)}$$

$$\nabla_\theta J(\hat{\theta}) = 0$$

Recall: Gradient Descent



Gradient Descent

- Input: training dataset $\mathcal{D} = \left\{\left(\boldsymbol{x}^{(i)}, y^{(i)}\right)\right\}_{i=1}^{N}$ and step size γ

1. Initialize $\boldsymbol{\theta}^{(0)}$ to all zeros and set $t = 0$
2. While TERMINATION CRITERION is not satisfied
   a. Compute the gradient:

$$\nabla_{\boldsymbol{\theta}} J\left(\boldsymbol{\theta}^{(t)}\right) = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}^{(i)} \left(P\left(Y = 1 \middle| \boldsymbol{x}^{(i)}, \boldsymbol{\theta}^{(t)}\right) - y^{(i)}\right)$$

   b. Update $\boldsymbol{\theta}$: $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \gamma \nabla_{\boldsymbol{\theta}} J\left(\boldsymbol{\theta}^{(t)}\right)$
   c. Increment t: $t \leftarrow t + 1$

- Output: $\boldsymbol{\theta}^{(t)}$

A. $O(1)$ **(TOXIC)** B. $O(N)$ C. $O(D)$ D. $O(ND)$

Poll Question 1:

What is the computational cost of one iteration of gradient descent for logistic regression?

- Input: training dataset $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$ and step size γ

1. Initialize $\boldsymbol{\theta}^{(0)}$ to all zeros and set $t = 0$
2. While TERMINATION CRITERION is not satisfied
   a. Compute the gradient:

   $$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^{(t)}) = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}^{(i)} \left(P\left(Y = 1 \middle| \boldsymbol{x}^{(i)}, \boldsymbol{\theta}^{(t)}\right) - y^{(i)} \right)$$

   b. Update $\boldsymbol{\theta}$: $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \gamma \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^{(t)})$
   c. Increment t: $t \leftarrow t + 1$

- Output: $\boldsymbol{\theta}^{(t)}$

Gradient Descent

- Input: training dataset $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$ and step size γ

1. Initialize $\boldsymbol{\theta}^{(0)}$ to all zeros and set $t = 0$
2. While TERMINATION CRITERION is not satisfied
   a. Compute the gradient:

$$O(ND)\left\{ \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^{(t)}) = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}^{(i)}\left(P\left(Y = 1 \middle| \boldsymbol{x}^{(i)}, \boldsymbol{\theta}^{(t)}\right) - y^{(i)}\right)\right.$$

   b. Update $\boldsymbol{\theta}$: $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \gamma \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^{(t)})$
   c. Increment t: $t \leftarrow t + 1$

- Output: $\boldsymbol{\theta}^{(t)}$

Stochastic Gradient Descent (SGD)

- Input: training dataset $\mathcal{D} = \left\{\left(\boldsymbol{x}^{(i)}, y^{(i)}\right)\right\}_{i=1}^{N}$ and step size γ

1. Initialize $\boldsymbol{\theta}^{(0)}$ to all zeros and set $t = 0$
2. While TERMINATION CRITERION is not satisfied
   a. Randomly sample a data point from $\mathcal{D}$, $\left(\boldsymbol{x}^{(i)}, y^{(i)}\right)$
   b. Compute the pointwise gradient:
      $$\nabla_{\boldsymbol{\theta}} J^{(i)}\left(\boldsymbol{\theta}^{(t)}\right) = \boldsymbol{x}^{(i)}\left(P\left(Y = 1 \middle| \boldsymbol{x}^{(i)}, \boldsymbol{\theta}^{(t)}\right) - y^{(i)}\right)$$
   c. Update $\boldsymbol{\theta}$: $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \gamma \nabla_{\boldsymbol{\theta}} J^{(i)}\left(\boldsymbol{\theta}^{(t)}\right)$
   d. Increment t: $t \leftarrow t + 1$

- Output: $\boldsymbol{\theta}^{(t)}$

Logistic Regression Learning Objectives

You should be able to…

- Apply the principle of maximum likelihood estimation (MLE) to learn the parameters of a probabilistic model
- Given a discriminative probabilistic model, derive the conditional log-likelihood, its gradient, and the corresponding Bayes Classifier
- Explain the practical reasons why we work with the log of the likelihood
- Implement logistic regression for binary (and multiclass) classification
- Prove that the decision boundary of binary logistic regression is linear