DS 4400

Machine Learning and Data Mining I
Spring 2021

Alina Oprea
Associate Professor
Khoury College of Computer Science
Northeastern University

February 16 2021

Outline

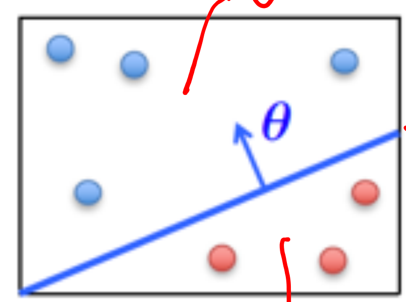
- Logistic regression
  - Classification based on probability
- Maximum Likelihood Estimation
  - Application to logistic regression
  - Cross-entropy objective
- Gradient descent for logistic regression
- Logistic regression lab
- Evaluation metrics for classifiers

Linear Classifiers

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

$\theta^T x > 0$

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



$\theta^T x = 0$

$\theta^T x < 0$

$h_\theta(x) = f(\theta^T x)$ linear function
- If $\theta^T x > 0$ classify "Class 1"
- If $\theta^T x < 0$ classify "Class 0"

$\theta_0 + \theta_1 x_1 + \dots + \theta_d x_d = 0$

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

Logistic Regression

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

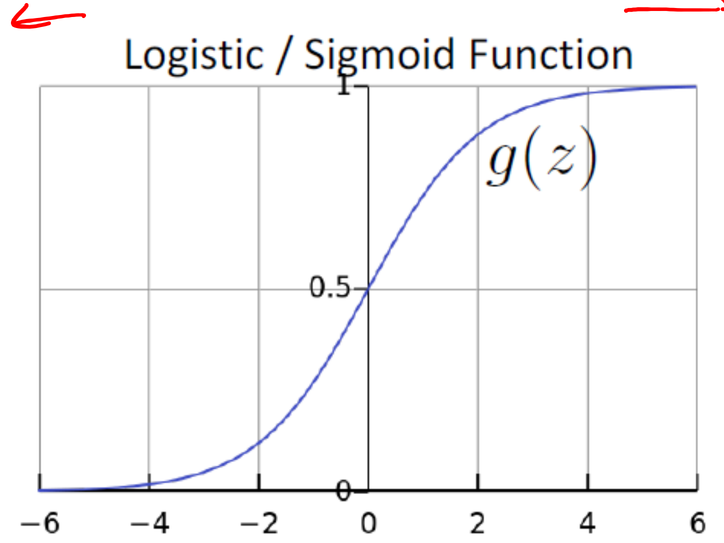
SIGMOID

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

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

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

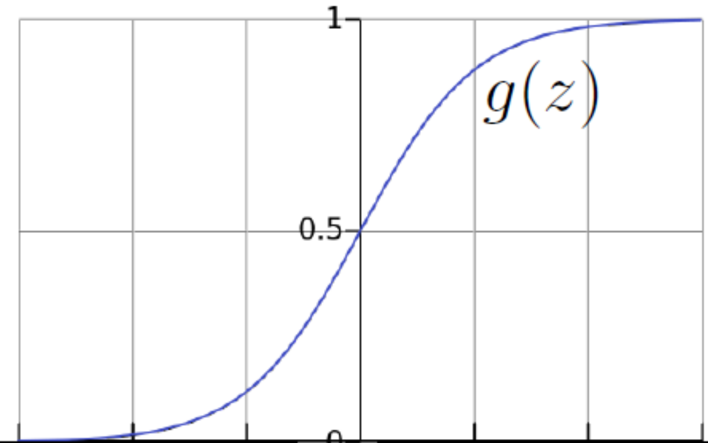
$-\infty$ ∞



Logistic Regression

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

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

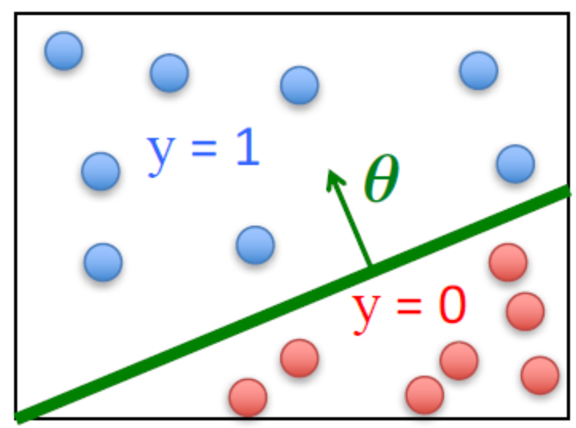


$\boldsymbol{\theta}^{\intercal}\boldsymbol{x}$ should be large negative values for negative instances

$\boldsymbol{\theta}^{\intercal}\boldsymbol{x}$ should be large positive values for positive instances

EQUIVALENT TO $\theta^T x > 0$

- Assume a threshold and...
  - Predict $Y = 1$ if $\boxed{h_{\boldsymbol{\theta}}(\boldsymbol{x}) \geq 0.5}$
  - Predict $Y = 0$ if $h_{\boldsymbol{\theta}}(\boldsymbol{x}) < 0.5$



Logistic Regression is a linear classifier!

Maximum Likelihood Estimation (MLE)

Given training data $X = \{\underbrace{x_1, \dots, x_N}\}$ with labels $Y = \{\underbrace{y_1, \dots, y_N}\}$, $y_i \in \{0,1\}$

What is the likelihood of training data for parameter θ?

Define likelihood function

$$Max_\theta \boxed{L(\theta) = P[Y|X;\theta]}$$ DEF

Assumption: training labels are conditionally independent

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

General probabilistic method for classifier training

Log Likelihood

- Max likelihood is equivalent to maximizing log of likelihood

log(XY) = log X + log y

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

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

- They both have the same maximum θ_{MLE}

MLE for Logistic Regression

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

, $y_i \in \{0, 1\}$

1) $y_i = 1$, $P[Y=1 \mid X = x_i; \theta] = h_\theta(x_i)$ FROM DEF OF h_θ FOR LOGISTIC REGRESSION

2) $y_i = 0$, $P[Y=0 \mid X = x_i; \theta] = 1 - h_\theta(x_i)$

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

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

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

Find θ to max $\log L(\theta)$ GIVEN $\{x_i, y_i\}$; $i = \overline{1, N}$

How to Train Logistic Regression

1) DEFINE OBJ. | LOSS: $J(\theta) = -\log L(\theta)$ CROSS-ENTROPY LOSS

USE GRADIENT DESCENT ON $J(\theta)$

2) GRADIENT ASCENT

$\max f(\theta)$

1) Init θ
2) Repeat until convergence
3) $\theta \leftarrow \theta + \alpha \cdot \frac{\partial f(\theta)}{\partial \theta}$

Cross-Entropy Objective

$$J(\theta) = -\sum_{i=1}^{N} \underbrace{[y_{\mathrm{i}} \log h_\theta(x_i) + (1 - y_i) \log \left(1 - h_\theta(x_i)\right)]}_{\text{COST | LOSS FOR TRAINING EXAMPLE } i}$$

Cross-Entropy Objective

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

- Cost of a single instance:

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

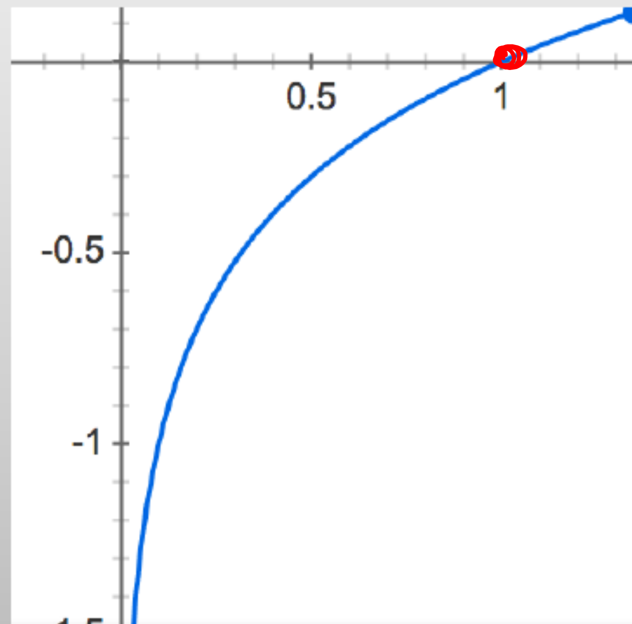
- Can re-write objective function as

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

Intuition

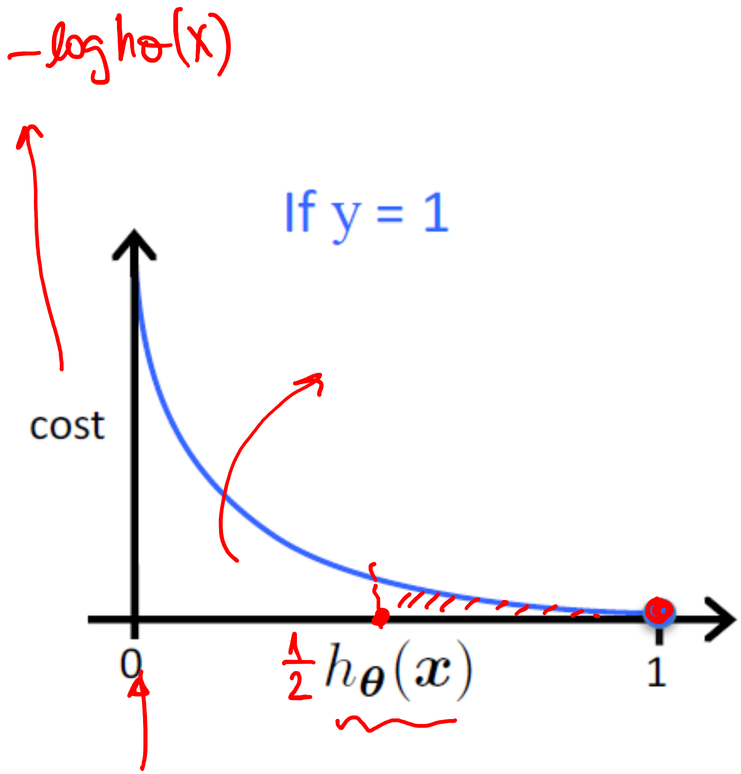
$$\text{cost}\left(h_{\boldsymbol{\theta}}(\boldsymbol{x}), y\right) = \begin{cases} -\log(h_{\boldsymbol{\theta}}(\boldsymbol{x})) & \text{if } y = 1 \\ -\log(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x})) & \text{if } y = 0 \end{cases}$$

Aside: Recall the plot of log(z)



Intuition

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



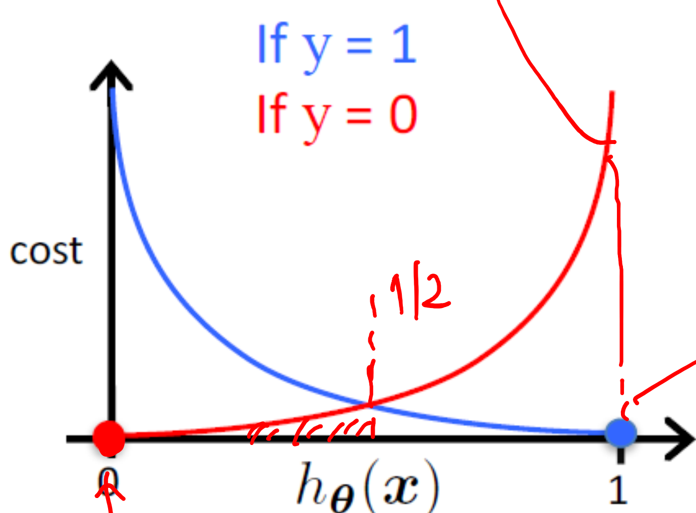
If y = 1

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

Intuition

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

$-\log(1 - h_\theta(x))$



HIGH LOSS

MODEL PREDICTS CORRECTLY ⇒ LOSS = 0

If y = 0

- Cost = 0 if prediction is correct
- As $\underbrace{(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x})) \rightarrow 0}_{h_\theta(x) \rightarrow 1}$, $\text{cost} \rightarrow \infty$
- Captures intuition that larger mistakes should get larger penalties

Gradient Descent for Logistic Regression

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

$$J(\theta) = -\sum_{i=1}^{N} c_i(\theta)$$

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

FOR ALL $j = 0, \dots, d$

$$\frac{\partial J(\theta)}{\partial \theta_j} = -\sum_{i=1}^{N} \frac{\partial c_i(\theta)}{\partial \theta_j}$$

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

$$g'(z) = +\frac{1}{(1+e^{-z})^2} \cdot e^{-z}$$

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

$$\frac{\partial h_\theta(x_i)}{\partial \theta_j} = \frac{\partial g(\theta^T x_i)}{\partial \theta_j} =$$

$$= g(\theta^T x_i)\left(1 - g(\theta^T x_i)\right) \cdot \boxed{\frac{\partial(\theta^T x_i)}{\partial \theta_j}}$$

$$\frac{\partial(\theta^T x_i)}{\partial \theta_j} \rightarrow x_{ij}$$

CHAIN RULE

CALCULUS:

$$\left(\frac{1}{x}\right)' = -\frac{1}{x^2} \qquad (e^x)' = e^x$$

$$(\log x)' = \frac{1}{x} \qquad (e^{-x})' = -e^{-x}$$

Gradient Computation

$$c_i(\theta) = y_i \log h_\theta(x_i) + (1-y_i)\log(1-h_\theta(x_i))$$

$$\frac{\partial h_\theta(x_i)}{\partial \theta_j} = g(\theta^T x_i)\,(1-g(\theta^T x_i))\,x_{ij}$$

$$\frac{\partial c_i(\theta)}{\partial \theta_j} = y_i \cdot \frac{1}{h_\theta(x_i)} \cdot \frac{\partial h_\theta(x_i)}{\partial \theta_j} - (1-y_i)\frac{1}{1-h_\theta(x_i)} \cdot \frac{\partial h_\theta(x_i)}{\partial \theta_j}$$

$$= y_i \cdot \frac{1}{\boxed{h_\theta(x_i)}} \cdot \boxed{g(\theta^T x_i)}\,(1-g(\theta^T x_i))\,x_{ij} - (1-y_i)\frac{1}{\boxed{1-h_\theta(x_i)}}\,g(\theta^T x_i)\,\boxed{(1-g(\theta^T x_i))}\,x_{ij}$$

$$= [y_i(1-g(\theta^T x_i)) - (1-y_i)\,g(\theta^T x_i)]\,x_{ij}$$

$$= [y_i - y_i\,g(\theta^T x_i) - g(\theta^T x_i) + y_i\,g(\theta^T x_i)]\,x_{ij} =$$

$$= [\,y_i - h_\theta(x_i)\,]\,x_{ij}$$

$$J(\theta) = -\sum_{i=1}^{N} C_i(\theta)$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = -\sum_{i=1}^{N} \frac{\partial C_i(\theta)}{\partial \theta_j} = \sum_{i=1}^{N} [h_\theta(x_i) - y_i]\, x_{ij}$$

Gradient Descent for Logistic Regression

CROSS-ENTROPY LOSS

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

1) LINEAR REG, $h_\theta(x_i) = \theta^T x_i$

2) LOGISTIC REG

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

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

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

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

$x_{i0} = 1$

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

, $j = 1, \ldots, d$

Gradient Descent for Logistic Regression

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

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

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

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

This looks IDENTICAL to Linear Regression!

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

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

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:

$$J_{\text{regularized}}(\boldsymbol{\theta}) = J(\boldsymbol{\theta}) + \boxed{\lambda \sum_{j=1}^{d} \theta_j^2}$$

REGULARIZATION TERM

$$= J(\boldsymbol{\theta}) + \lambda \|\boldsymbol{\theta}_{[1:d]}\|_2^2$$

L2 regularization

LASSO REG: $J_{LASSO}(\theta) = J(\theta) + \lambda \underbrace{\sum_{j=1}^{d} |\theta_j|}_{|\theta|_1}$

Logistic Regression
Lab Example

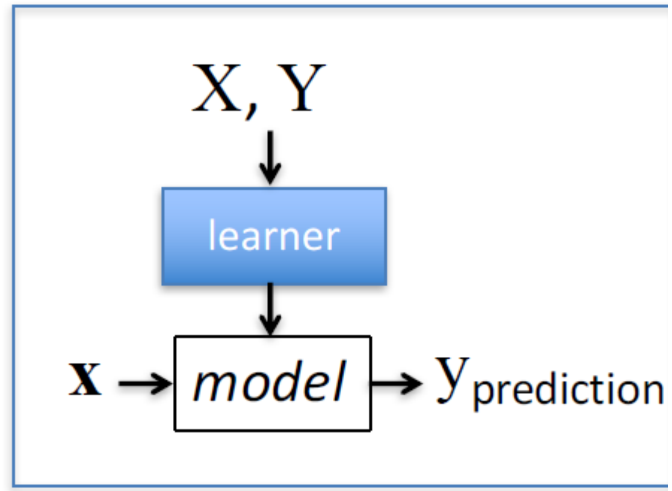
Classifier Evaluation

- Classification is a supervised learning problem
  - Prediction is binary or multi-class
- Classification techniques
  - Linear classifiers
    - ~~Perceptron (online or batch mode)~~
    - Logistic regression (probabilistic interpretation)
  - Instance learners
    - kNN: need to store entire training data
- Cross-validation should be used for parameter selection and estimation of model error

Evaluation of classifiers

Given: labeled training data $X, Y = \{\langle \boldsymbol{x}_i, y_i \rangle\}_{i=1}^{n}$

- Assumes each $\boldsymbol{x}_i \sim \mathcal{D}(\mathcal{X})$



Train the model:

model ← *classifier*.train(X, Y)

Apply the model to new data:

VALIDATION
TESTING

- Given: new unlabeled instance $\boldsymbol{x} \sim \mathcal{D}(\mathcal{X})$

$y_{\text{prediction}}$ ← *model*.predict($\mathbf{x}$)

Classification Metrics

$$\text{accuracy} = \frac{\#\text{ correct predictions}}{\#\text{ test instances}}$$

$$\text{error} = 1 - \text{accuracy} = \frac{\#\text{ incorrect predictions}}{\#\text{ test instances}}$$

- Training set accuracy and error
- Testing set accuracy and error

Confusion Matrix

Given a dataset of P positive instances and N negative instances:

		Predicted Class	
		Yes	No
Actual Class	Yes	TP	FN
	No	FP	TN

$P = TP + FN$

$N = TN + FP$

$All = P + N$

$$\text{ACCURACY} = \frac{TP + TN}{P + N}$$

$$\text{ERROR} = \frac{FN + FP}{P + N}$$

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
- Logistic regression can be trained with Gradient Descent

Acknowledgements

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