slido

Join at slido.com
#9774775

ⓘ Start presenting to display the joining instructions on this slide.

LECTURE 23

Logistic Regression I

Moving from regression to classification.

Data 100/Data 200, Spring 2023 @ UC Berkeley

Narges Norouzi and Lisa Yan

A Different Modeling Problem

9774775

(today)

Logistic Regression I:
The Model
Cross-Entropy Loss
The Probabilistic View

Logistic Regression II:
Classification Thresholds
Accuracy, Precision, Recall
Linear Separability

9774775

Today's Roadmap

Lecture 23, Data 100 Spring 2023

Regression vs. Classification

Deriving the Logistic Regression Model

- The Sigmoid (Logistic) Function

The Logistic Regression Model

- Comparison to Linear Regression

Parameter Estimation

- Pitfalls of Squared Loss
- Cross-Entropy Loss
- Maximum Likelihood Estimation

9774775

Regression vs. Classification

Lecture 23, Data 100 Spring 2023

Regression vs. Classification

Deriving the Logistic Regression Model

- The Sigmoid (Logistic) Function

The Logistic Regression Model

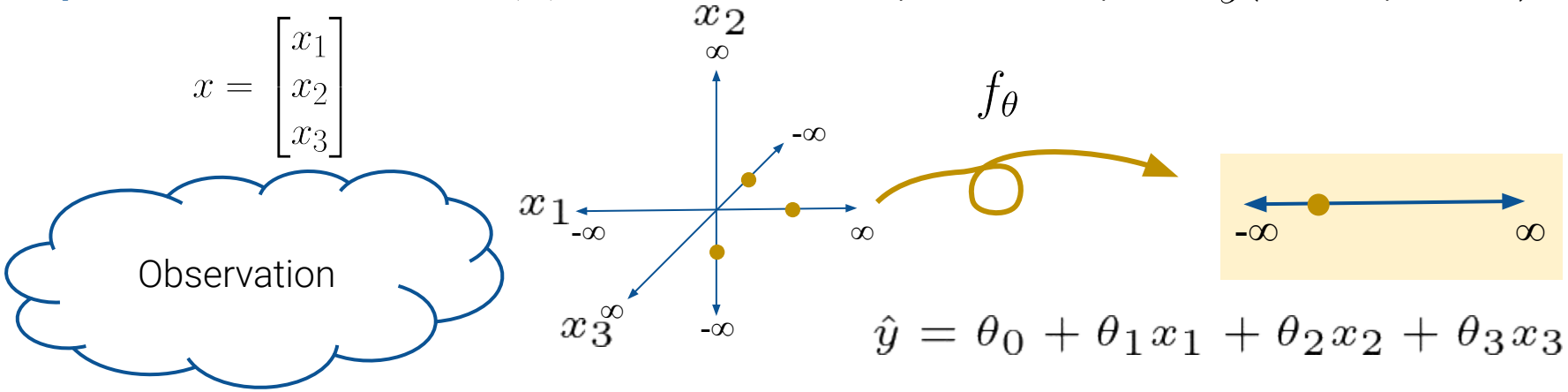
- Comparison to Linear Regression

Parameter Estimation

- Pitfalls of Squared Loss
- Cross-Entropy Loss
- Maximum Likelihood Estimation

So far: Regression

The **parametric model** $\hat{y} = f_\theta(x)$, uses a feature x to predict a response $\hat{y}$ (true response y)



Use training data to estimate optimal $\hat{\theta}$:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \underbrace{(y_i - f_\theta(x_i))^2}_{\text{Squared loss}} + \underbrace{\lambda \text{Reg}(\theta)}_{\text{Regularization}}$$

Regression:
Predict a **real** number y.

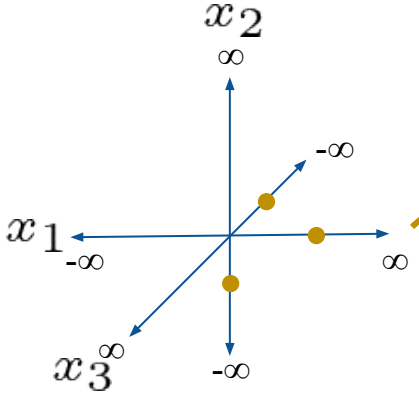
Today: Classification

The **parametric model** $\hat{y} = f_\theta(x)$, uses a feature x to predict a response $\hat{y}$ (true response y)

9774775

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Observation



f_θ

IsWin?
{0, 1}

Classification:
Predict a **categorical** variable y.

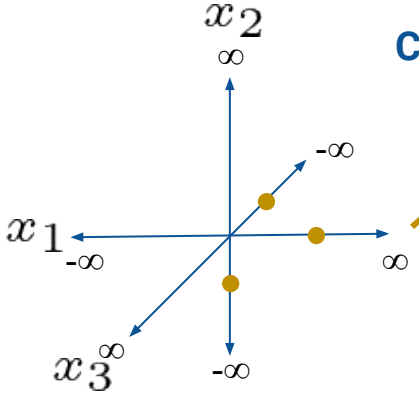
Decision Rule

The **parametric model** $\hat{y} = f_\theta(x)$, uses a feature x to predict a response $\hat{y}$(true response y)

9774775

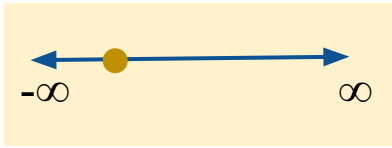
$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Observation



Classification Activation

$$z = f_\theta(x)$$



Decision Rule

IsWin?
{0, 1}

An example of a simple decision rule:

- **Sign function** $sign(z) = \begin{cases} 1, & \text{if } z \geq 0 \\ 0, & \text{if } z < 0 \end{cases}$

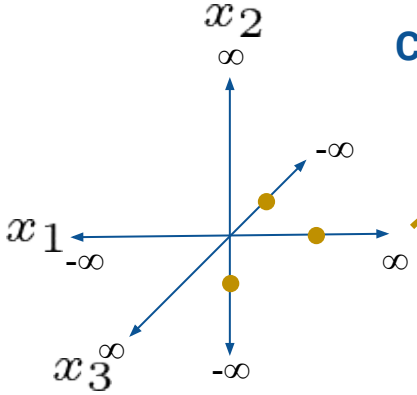
Binary Classification with Sign Function Decision Rule

The **parametric model** $\hat{y} = f_\theta(x)$, uses a feature x to predict a response $\hat{y}$(true response y)

9774775

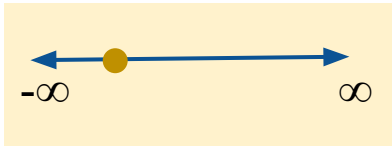
$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Observation



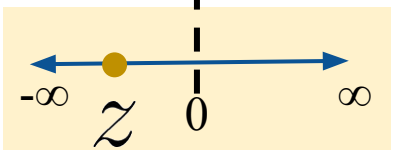
Classification Activation

$$z = f_\theta(x)$$



Decision Rule
= Sign Function

$$sign(z) = \begin{cases} 1, & \text{if } z \geq 0 \\ 0, & \text{if } z < 0 \end{cases}$$

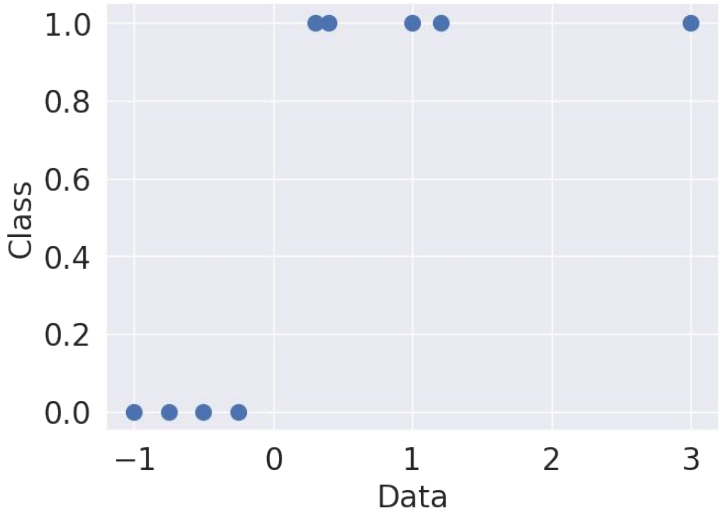


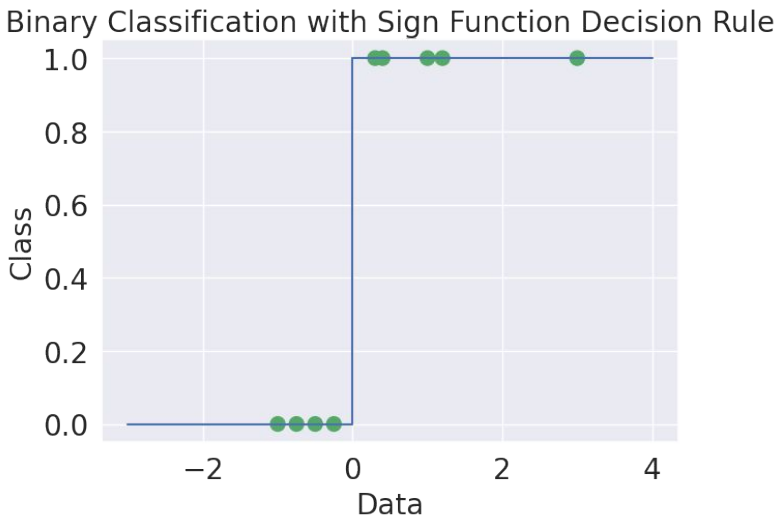
IsWin?
{0, 1}

Example of Binary Classification with Sign Function Decision Rule

9774775

	x	y
0	-1.00	0
1	-0.75	0
2	-0.50	0
3	-0.25	0
4	0.30	1
5	0.40	1
6	1.00	1
7	1.20	1
8	3.00	1



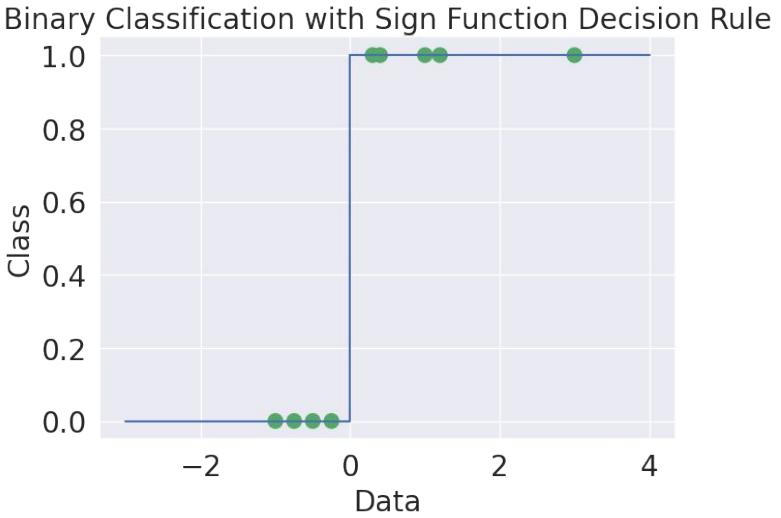


Summary of A Classification Task

1. Obtaining training data and selecting features for the classification task,
2. Learning the classification activation value z based on a weighted combination of input features $z = f_\theta(x)$ where $f_\theta(x) = \vec{\theta}.\vec{x}$
3. The decision rule (ex. Sign function) will be applied to calculate the final class association.

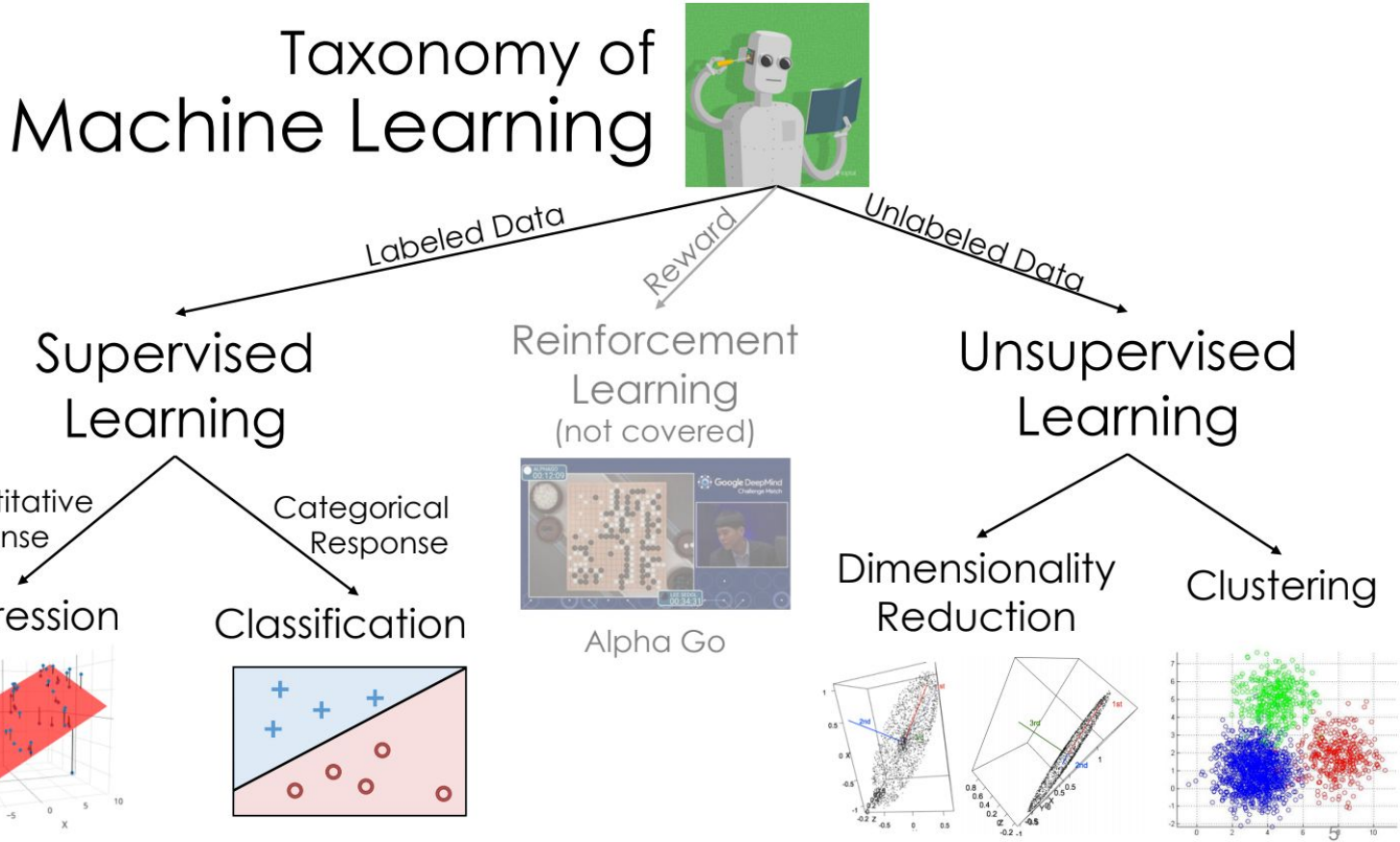
If decision rule is the step function:

$$\text{classify}(x) = \begin{cases} 1, & \text{if } f_\theta(x) \geq 0 \\ 0, & \text{if } f_\theta(x) < 0 \end{cases}$$



Taxonomy of Machine Learning

Regression and classification are both forms of **supervised learning**.



Kinds of Classification

9774775

We are interested in predicting some **categorical variable**, or **response**, y.

Binary classification [today]

- Two classes
- **Responses** y are either 0 or 1

win or lose

disease or no disease

spam or ham

Multiclass classification

- Many classes
- Examples:
  - Image labeling (tuxedo cat, black/white yarn, and moon crescent),
  - Next word in a sentence.

Structured prediction tasks

- Multiple related classification predictions.
- Examples: Translation, voice recognition, etc.

The Modeling Process

9774775

	Regression ($y \in \mathbb{R}$)	Classification ($y \in \{0, 1\}$)
1. Choose a model	Linear Regression $\hat{y} = f_\theta(x) = x^T\theta$	??
2. Choose a loss function	Squared Loss or Absolute Loss	??
3. Fit the model	Regularization Sklearn/Gradient descent	Regularization Sklearn/Gradient descent
4. Evaluate model performance	R^2, Residuals, etc.	?? (next time)

9774775

Deriving the Logistic Regression Model

Lecture 23, Data 100 Spring 2023

Regression vs. Classification

Deriving the Logistic Regression Model

- The Sigmoid (Logistic) Function

The Logistic Regression Model

- Comparison to Linear Regression

Parameter Estimation

- Pitfalls of Squared Loss
- Cross-Entropy Loss
- Maximum Likelihood Estimation

Probabilistic Model

9774775

- What if instead of a 0-1 prediction response, the model returns the probability of data belonging to binary classes.
  - If the model's predicted probability is **greater** than **0.5**, classify as **class 1**.
  - If the model's predicted probability is **less** than **0.5**, classify as **class 0**.

- The model still uses the concept of classification activation as a way to aggregate multiple features using a weighted combination of features.

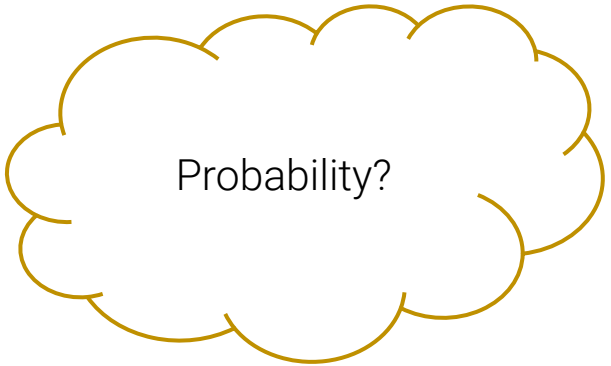
- The probabilistic classifier learns to generate the following probabilities and with the decision threshold of **0.5**, classifies datapoints into two categories:

$$\text{classify}(x) = \begin{cases} 1, & \text{if } P_\theta(Y=1|x) \geq 0.5 \\ 0, & \text{if } P_\theta(Y=1|x) < 0.5 \end{cases}$$

$$P_\theta(Y=1|x) = P(Y=1|z)$$

The Modeling Process

9774775

	Regression ($y \in \mathbb{R}$)	Classification ($y \in \{0, 1\}$)
1. Choose a model	Linear Regression $\hat{y} = f_\theta(x) = x^T\theta$	
2. Choose a loss function	Squared Loss or Absolute Loss	
3. Fit the model	Regularization Sklearn/Gradient descent	Regularization Sklearn/Gradient descent
4. Evaluate model performance	R^2, Residuals, etc.	?? (next time)

Probability Review - Odds Ratio $[0, 1) \rightarrow [0, +\infty)$

9774775

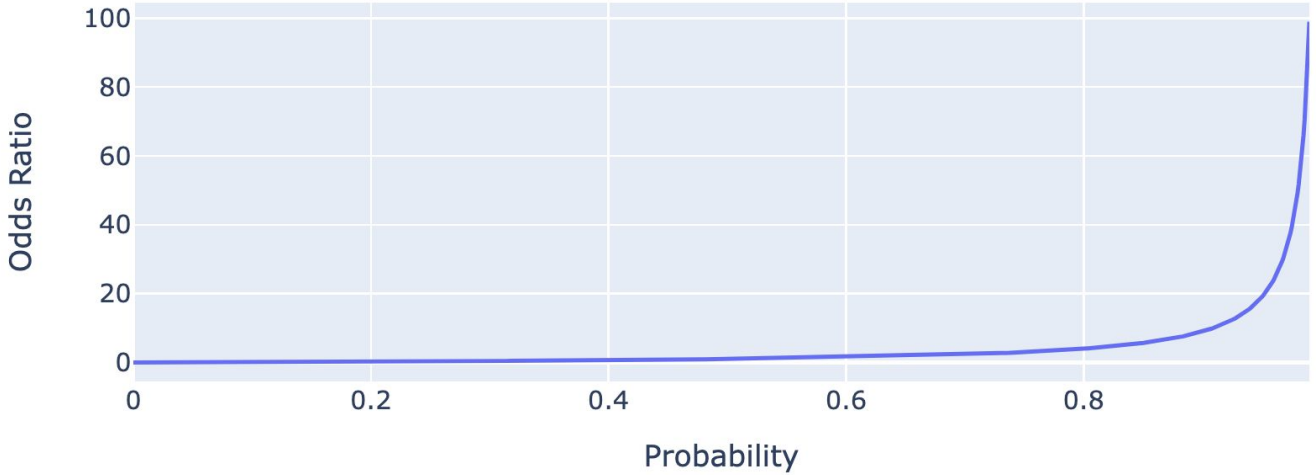
Odds is defined as the ratio of the probabilities of happening vs. not happening.

- "9 to 1 against", meaning a probability of 0.1.
- "Even odds", meaning p = 0.5.
- "3 to 1 on" meaning p = 0.75.

If p is the probability of response 1, then $\text{odds}(p) = \dfrac{p}{1-p}$

Domain: $[0, 1)$

Range: $[0, +\infty)$



One More Transformation

9774775

- Our end goal is to use a classification activation to output a probability value. The odds ratio takes a probability value and outputs a positive real number.
- **What transformation of odds generates an unbounded real number?**

$$[0, +\infty) \rightarrow (-\infty, +\infty)$$

slido

What function maps a positive real number to an unbounded real number?

ⓘ Start presenting to display the poll results on this slide.

Logit Function $(0, 1) \rightarrow (-\infty, +\infty)$

9774775

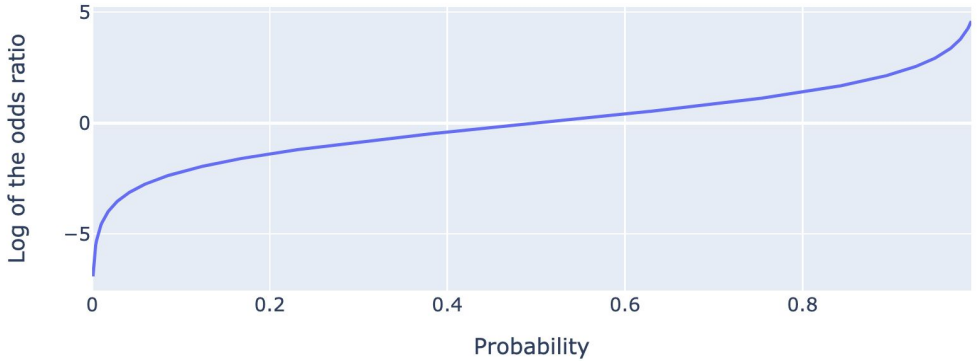
- Our end goal is to use a classification activation to output a probability value. The odds ratio takes a probability value and outputs a positive real number.
- What transformation of odds generates an unbounded real number? $[0, +\infty) \rightarrow (-\infty, +\infty)$
- Taking the $\log$ of a positive real number, creates an unbounded real number:

$$\log(x) \in \begin{cases} (-\infty, 0], & \text{if } x \in (0, 1] \\ (0, +\infty), & \text{if } x \in (1, +\infty) \end{cases}$$

- This function is called **Logit Function:** $\log(\frac{p}{1-p})$

Domain: $(0, 1)$

Range: $(-\infty, +\infty)$



Logistic Function: Inverse of Logit Function: $(-\infty, +\infty) \rightarrow (0, 1)$

9774775

Let's calculate the inverse of the logit function:

$$\log(\frac{p}{1-p}) = z \in \mathbb{R}$$

$$\frac{p}{1-p} = e^z$$

$$p = (1-p)e^z$$

$$p = e^z - pe^z$$

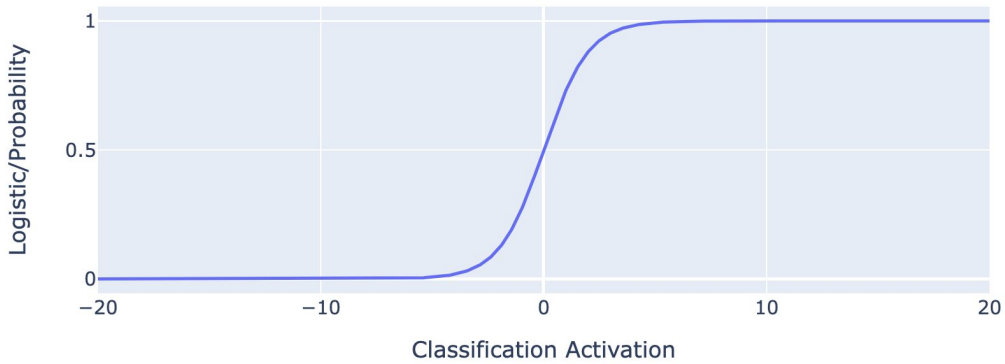
$$p + pe^z = e^z$$

$$p(1+e^z) = e^z$$

$$p = \frac{e^z}{1+e^z}$$

$$p = \frac{1}{1+e^{-z}}$$

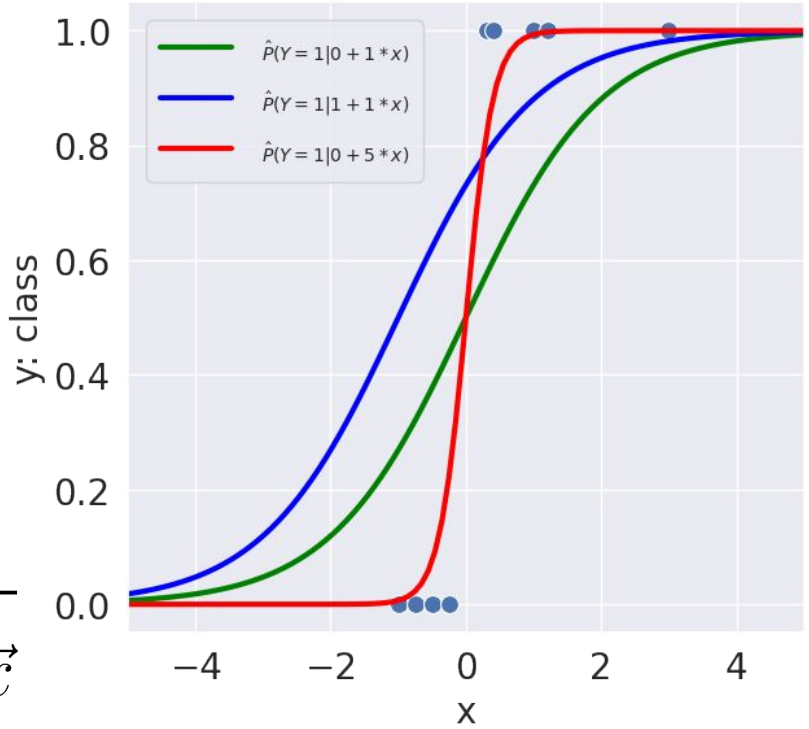
This function is called **Logistic Function** or **Sigmoid** and is shown by $\sigma(z)$



Logistic Regression Model Summary

- Logistic regression model summary:
  - Optimize the model to find parameters $\hat{\theta}$.
  - Use parameters to calculate classification activation $z = \hat{\vec{\theta}}.\vec{x}$
  - Apply Sigmoid function to create probability of a given datapoint belonging to class 1.

$$\hat{P} = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-\hat{\vec{\theta}}.\vec{x}}}$$



How to obtain the best fit $\hat{\theta}$? We will talk about this next.

9774775

The Sigmoid (Logistic) Function

Lecture 23, Data 100 Spring 2023

Regression vs. Classification

Deriving the Logistic Regression Model

- **The Sigmoid (Logistic) Function**

The Logistic Regression Model

- Comparison to Linear Regression

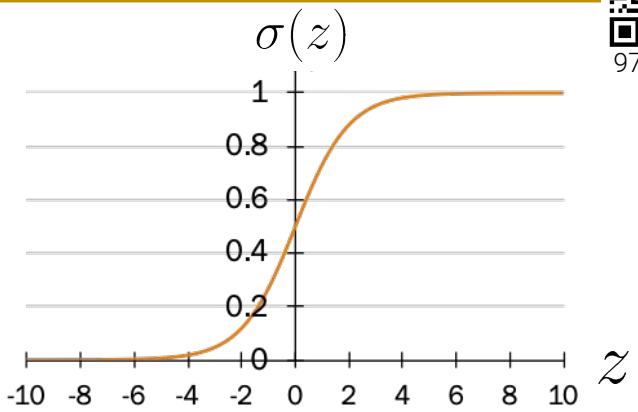
Parameter Estimation

- Pitfalls of Squared Loss
- Cross-Entropy Loss
- Maximum Likelihood Estimation

The Logistic Function

The **logistic function**:

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



This is a type of **sigmoid**, a class of functions that share certain properties.

Domain

$$z \in (-\infty, +\infty)$$

Range

$$\sigma(z) \in (0, 1)$$

The logistic function smoothly squashes a real number to between 0 and 1.

Reflection/ Symmetry

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

Derivative

$$\frac{d}{dz}\sigma(z) = \sigma(z)(1-\sigma(z)) = \sigma(z)\sigma(-z)$$

9774775

The Logistic Regression Model

Lecture 23, Data 100 Spring 2023

Regression vs. Classification

Deriving the Logistic Regression Model

- The Sigmoid (Logistic) Function

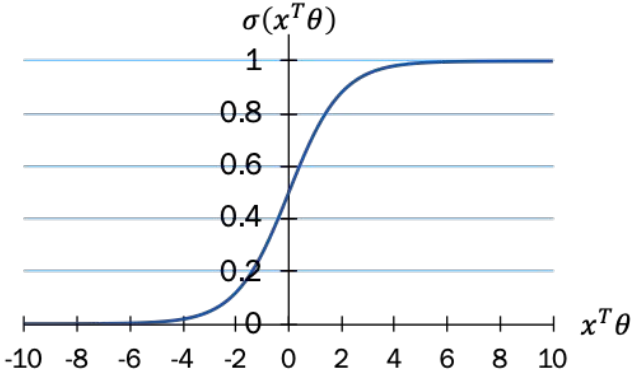
The Logistic Regression Model

- Comparison to Linear Regression

Parameter Estimation

- Pitfalls of Squared Loss
- Cross-Entropy Loss
- Maximum Likelihood Estimation

Logistic Regression Process



feature vector $\boldsymbol{x}$ → $\boldsymbol{z = x^T\theta}$ → scalar $z \in \mathbb{R}$ → $\boldsymbol{\sigma(\ )}$ → scalar $\sigma(x^T\theta) \in (0, 1)$

linear

parameter θ

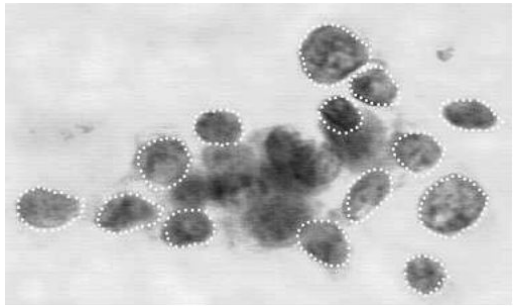
non-linear

The Breast Cancer Wisconsin Dataset

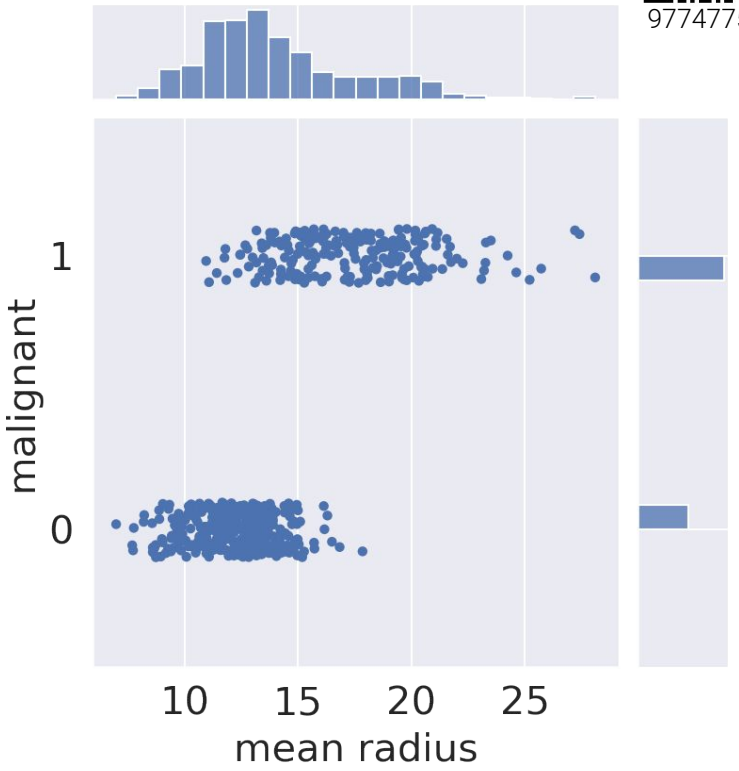
9774775

Input x: mean radius of breast tumor cells
Response y: 1 if malignant, 0 if benign

mean radius	malignant
17.99	1
20.57	1
19.69	1
11.42	1
20.29	1
...	...
21.56	1
20.13	1
16.60	1
20.60	1
7.76	0



512 training observations, 57 test



Classification labels are jittered to avoid overplotting.

Given the (single) **input feature**,
how can we predict the **label class**?

Logistic Regression on Breast Cancer Dataset

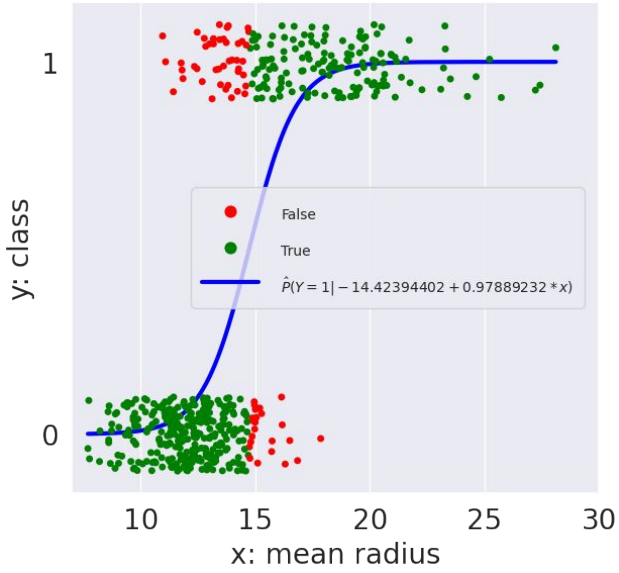
9774775

```
from sklearn.linear_model import LogisticRegression

model = LogisticRegression(fit_intercept=True)
model.fit(X, Y); # X, Y are training data

model.intercept_, model.coef_
```

```
(array([-14.42394402]), array([[0.97889232]]))
```



Demo

Example Calculation

9774775

Suppose I want to predict the probability that a tumor is malignant, given **mean radius** (first feature) and **mean smoothness** (second feature).

Suppose I fit a logistic regression model (with no intercept) using my training data, and somehow estimate the optimal parameters:

$$\hat{\theta}^T = [0.1 \qquad -0.5]$$

Now, you encounter a new breast tumor image:

$$x^T = [15 \qquad 1]$$

---

1. What is the probability that the tumor is malignant (Y = 1)?
2. What would you predict as response? 1 or 0?

$$\hat{P}_\theta(Y = 1|x) = \sigma(x^T\theta)$$

Example Calculation

9774775

Suppose I want to predict the probability that a tumor is malignant, given **mean radius** (first feature) and **mean smoothness** (second feature).

Suppose I fit a logistic regression model (with no intercept) using my training data, and somehow estimate the optimal parameters:

$$\hat{\theta}^T = [0.1 \qquad -0.5]$$

Now, you encounter a new breast tumor image:

$$x^T = [15 \qquad 1]$$

$$\begin{aligned}\hat{P}_{\hat{\theta}}(Y = 1|x) &= \sigma(x^T\hat{\theta}) \\ &= \sigma(0.1 \cdot 15 + (-0.5) \cdot 1) \\ &= \sigma(1) \\ &= \frac{1}{1 + e^{-1}} \\ &\approx 0.7311\end{aligned}$$

Because the response is more likely to be 1 than 0, a reasonable prediction is

$$\hat{y} = 1$$

9774775

Comparison to Linear Regression

Lecture 23, Data 100 Spring 2023

Regression vs. Classification

Deriving the Logistic Regression Model

- The Sigmoid (Logistic) Function

The Logistic Regression Model

- **Comparison to Linear Regression**

Parameter Estimation

- Pitfalls of Squared Loss
- Cross-Entropy Loss
- Maximum Likelihood Estimation

Linear Regression vs. Logistic Regression

9774775

Linear Regression model, parameter $\theta = (\theta_0, \theta_1, \ldots, \theta_p)$:

x → $f_\theta(x) = x^T\theta$ → $\hat{y}$ $\approx$ $y \in \mathbb{R}$

observation vector

prediction

quantitative response

Logistic Regression model, parameter $\theta = (\theta_0, \theta_1, \ldots, \theta_p)$:

x → $\hat{P}_\theta(Y = 1|x) = \sigma(x^T\theta)$ → Probability of response = 1

observation vector

classify → $\hat{y}$ $\approx$ $y \in \{0, 1\}$

prediction

categorical response

Logistic "Regression" Is Misleading

9774775

Linear Regression model, parameter $\theta = (\theta_0, \theta_1, \ldots, \theta_p)$:

x → $f_\theta(x) = x^T\theta$ → $\hat{y}$ $\approx$ $y \in \mathbb{R}$

observation vector

prediction

quantitative response

Logistic Regression model, parameter $\theta = (\theta_0, \theta_1, \ldots, \theta_p)$:

x → $\hat{P}_\theta(Y = 1|x) = \sigma(x^T\theta)$ → Probability of response = 1

observation vector

Probability of response = 1 → classify → $\hat{y}$ $\approx$ $y \in \{0, 1\}$

prediction

categorical response

Logistic Regression is used for **classification**. It has a confusing name!

Parameter Interpretation

9774775

Logistic Regression Model assumptions:

- The curve models probability: P(Y = 1 | z).
- Assume log-odds is a linear combination of x and θ.

$$\hat{P}_\theta(Y = 1|x) = \sigma(x^T\theta)$$

$$\log\left(\frac{p}{1-p}\right) = x^T\theta$$

Because we are dealing with binary classification,

$$P(Y = 1|x) + P(Y = 0|x) = 1$$

$$\frac{P(Y = 1|x)}{P(Y = 0|x)} = e^{x^T\theta}$$

Parameter Interpretation

9774775

Let's suppose our linear component has just a single feature, along with an intercept term.

$$\frac{P(Y=1|x)}{P(Y=0|x)} = e^{\theta_0+\theta_1 x}$$

What happens if you increase x by one unit?

- Odds is multiplied by e^{θ_1} .
- If $\theta_1 > 0$, the odds increase.
- If $\theta_1 < 0$, the odds decrease.

The odds ratio can be interpreted as the "number of successes for each failure."

What happens if $x^T\theta = \theta_0 + \theta_1 x = 0$?

- This means class 1 and class 0 are equally likely.
- $$e^0 = 1 \implies \frac{P(Y=1|x)}{P(Y=0|x)} = 1 \implies P(Y=1|x) = P(Y=0|x)$$

The Modeling Process

9774775

	Regression ($y \in \mathbb{R}$)	Classification ($y \in \{0, 1\}$)
1. Choose a model ✅	Linear Regression $\hat{y} = f_\theta(x) = x^T\theta$	Logistic Regression $\hat{P}_\theta(Y = 1\|x) = \sigma(x^T\theta)$
2. Choose a loss function	Squared Loss or Absolute Loss	??
3. Fit the model	Regularization Sklearn/Gradient descent	Regularization Sklearn/Gradient descent
4. Evaluate model performance	R^2, Residuals, etc.	?? (next time)

The Modeling Process

9774775

	Regression ($y \in \mathbb{R}$)	Classification ($y \in \{0, 1\}$)
1. Choose a model ✅	Linear Regression $\hat{y} = f_\theta(x) = x^T\theta$	Logistic Regression $\hat{P}_\theta(Y = 1\|x) = \sigma(x^T\theta)$
2. Choose a loss function	Squared Loss or Absolute Loss	Can squared loss still work?
3. Fit the model	Regularization Sklearn/Gradient descent	Regularization Sklearn/Gradient descent
4. Evaluate model performance	R^2, Residuals, etc.	?? (next time)

9774775

Pitfalls of Squared Loss

Lecture 23, Data 100 Spring 2023

Regression vs. Classification

Deriving the Logistic Regression Model

- The Sigmoid (Logistic) Function

The Logistic Regression Model

- Comparison to Linear Regression

Parameter Estimation

- **Pitfalls of Squared Loss**
- Cross-Entropy Loss
- Maximum Likelihood Estimation

Toy Dataset: L2 Loss

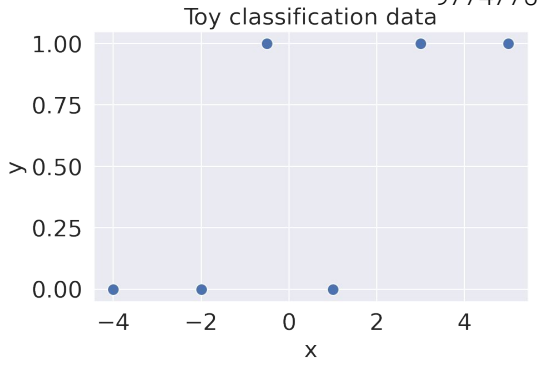
9774775

Logistic Regression model:

$$\hat{P}_\theta(Y = 1|x) = \sigma(x^T\theta)$$

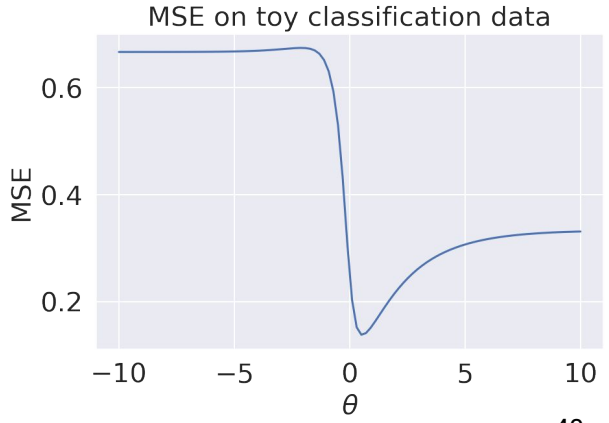
Assume no intercept.
So x, θ both scalars.

	x	y
0	-4.0	0
1	-2.0	0
2	-0.5	1
3	1.0	0
4	3.0	1
5	5.0	1



Mean Squared Error:

$$R(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \sigma(x_i^T\theta)\right)^2$$



The MSE loss surface for logistic regression has many issues!

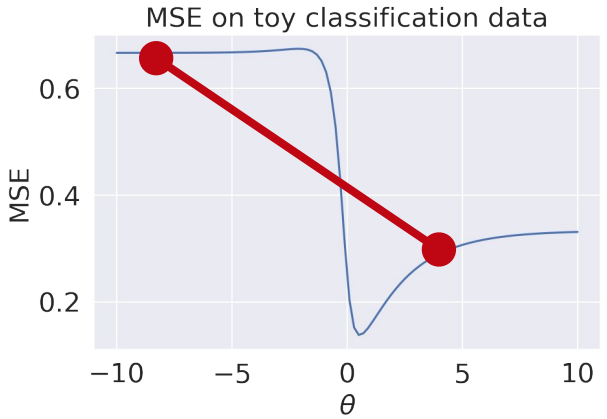
Demo

3 Pitfalls of Squared Loss

9774775

1. Non-convex. Gets stuck in local minima.

Secant line crosses function, so $R''(\theta)$ is not greater than 0 for all θ.



Demo

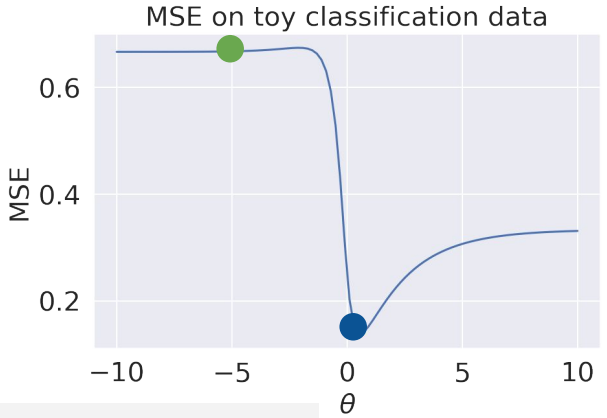
3 Pitfalls of Squared Loss

9774775

1. Non-convex. Gets stuck in local minima.

Secant line crosses function, so $R''(\theta)$ is not greater than 0 for all θ.

Gradient Descent: Different initial guesses will yield different optimal estimates.




```
from scipy.optimize import minimize

minimize(mse_loss_toy_nobias, x0 = 0)["x"][0]
```

```
0.5446601825581691
```

```
minimize(mse_loss_toy_nobias, x0 = -5)["x"][0]
```

```
-10.343653061026611
```

Demo

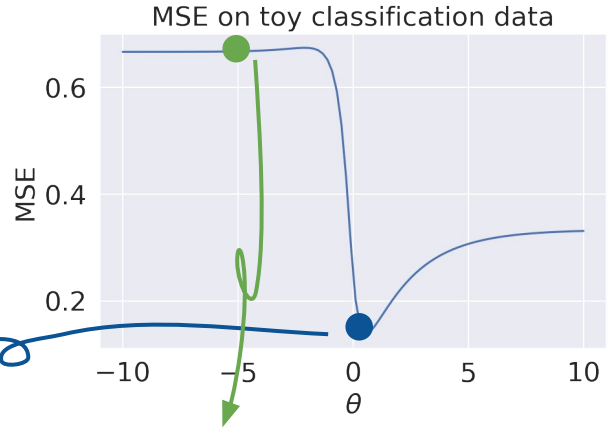
3 Pitfalls of Squared Loss

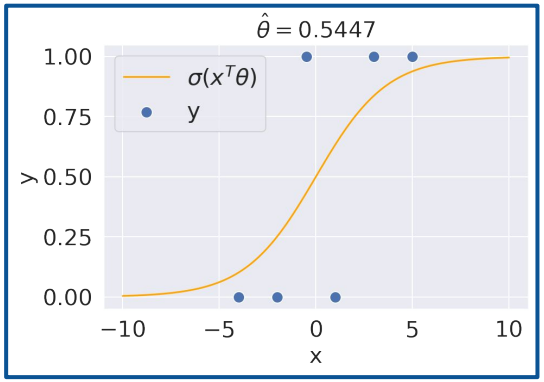
9774775

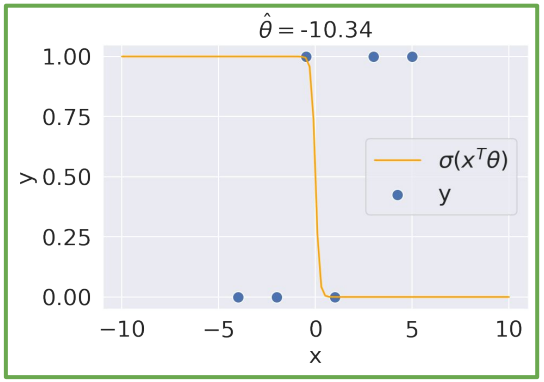
1. Non-convex. Gets stuck in local minima.

Secant line crosses function, so $R''(\theta)$ is not greater than 0 for all θ.

Gradient Descent: Different initial guesses will yield different optimal estimates.







Demo

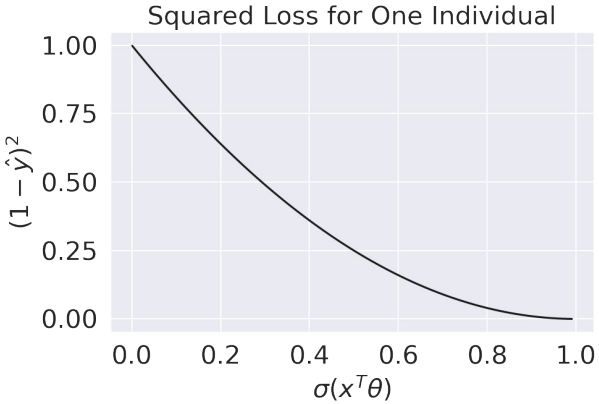
3 Pitfalls of Squared Loss

9774775

1. **Non-convex**. Gets stuck in local minima.

2. **Bounded**. Not a good measure of model error.

- We'd like loss functions to penalize "off" predictions.
- MSE never gets very large, because both response and predicted probability are bounded by 1.



Demo

3 Big Pitfalls of Squared Loss

9774775

1. **Non-convex**. Gets stuck in local minima.

2. **Bounded**. Not a good measure of model error.

3. **Conceptually questionable**.
   Tries to match probability to 0/1 class labels.

shutterstock.com · 406338748

MSE + classification is *occasionally* used in some neural network applications.
But overall, avoid.

Demo

9774775

Cross-Entropy Loss

Lecture 23, Data 100 Spring 2023

Regression vs. Classification

Deriving the Logistic Regression Model

- The Sigmoid (Logistic) Function

The Logistic Regression Model

- Comparison to Linear Regression

Parameter Estimation

- Pitfalls of Squared Loss
- **Cross-Entropy Loss**
- Maximum Likelihood Estimation

Choosing a Different Loss Function

9774775

	Regression ($y \in \mathbb{R}$)	Classification ($y \in \{0, 1\}$)
1. Choose a model ✅	Linear Regression $\hat{y} = f_\theta(x) = x^T\theta$	Logistic Regression $\hat{P}_\theta(Y = 1\|x) = \sigma(x^T\theta)$
2. Choose a loss function	Squared Loss or Absolute Loss	**Cross-Entropy Loss**
3. Fit the model	Regularization Sklearn/Gradient descent	Regularization Sklearn/Gradient descent
4. Evaluate model performance	R^2, Residuals, etc.	?? (next time)

Cross-Entropy Loss

9774775

Let y be a binary label {0, 1}, and p be the probability of the label being 1.

The **cross-entropy loss** is defined as

$$\text{CE loss} = \begin{cases} -\log(p), & \text{if } y = 1 \\ -\log(1-p), & \text{if } y = 0 \end{cases} = -\left(y \log(p) + (1-y)\log(1-p)\right)$$

Matches the probabilistic modeling of logistic regression if $p = \hat{P}_\theta(Y = 1|x) = \sigma(x^T\theta)$

Cross-Entropy loss addresses the 3 pitfalls of squared loss.

1. Convex. No local minima for logistic regression.
2. A good measure of model error. Strongly penalizes “off” predictions.
3. Conceptually sound.

For now, suspend your belief about #3 (what cross-entropy loss actually means).

We’ll focus on its mathematical properties (#1 and #2) first.

Cross-Entropy Loss Is Like, Two Loss Functions In One

9774775

Let y be a binary label {0, 1}, and p be the probability of the label being 1.

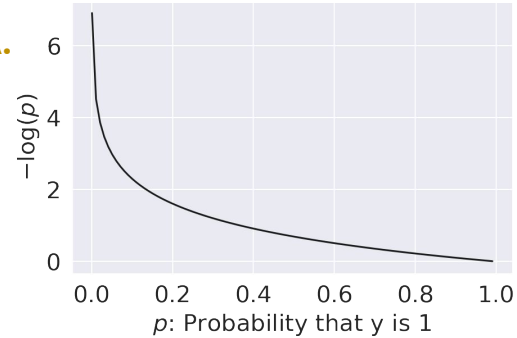
The **cross-entropy loss** is defined as

$$-\left(y\log(p)+(1-y)\log(1-p)\right)$$

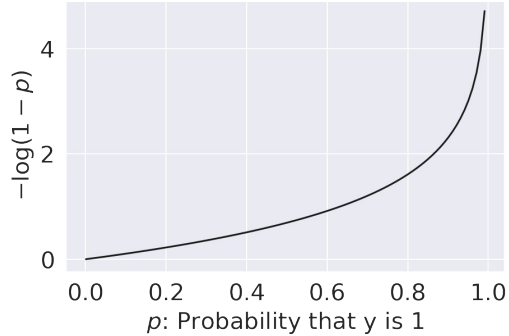
Suppose we can choose *p* to be any value between 0 and 1. Which plot best represents cross-entropy loss if:

1) The true response y = 1?
2) The true response y = 0?

A.



B.



C. Something else

Does cross-entropy loss strongly penalize “off” predictions?

slido

Which plot best represents cross-entropy loss if true response is y = 1?

ⓘ Start presenting to display the poll results on this slide.

Cross-Entropy Loss Is Like, Two Loss Functions In One

9774775

Let y be a binary label {0, 1}, and p be the probability of the label being 1.

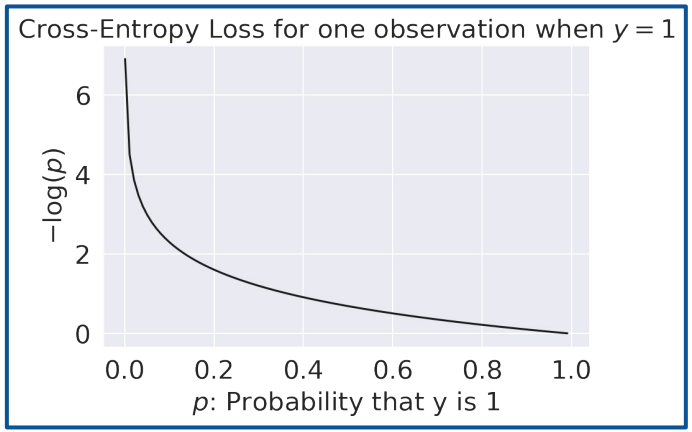
The **cross-entropy loss** is defined as

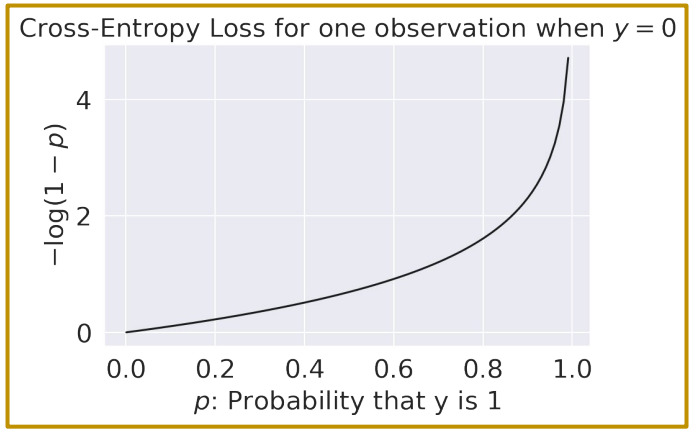
$$-(y\log(p) + (1-y)\log(1-p))$$

makes loss positive

for y = **1**, only this term stays

for y = **0**, only this term stays



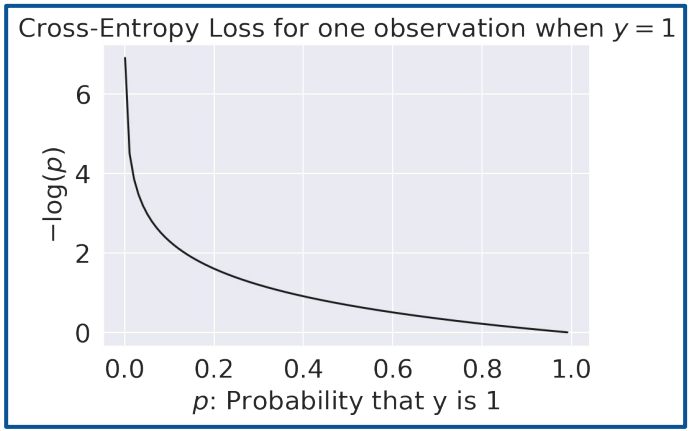


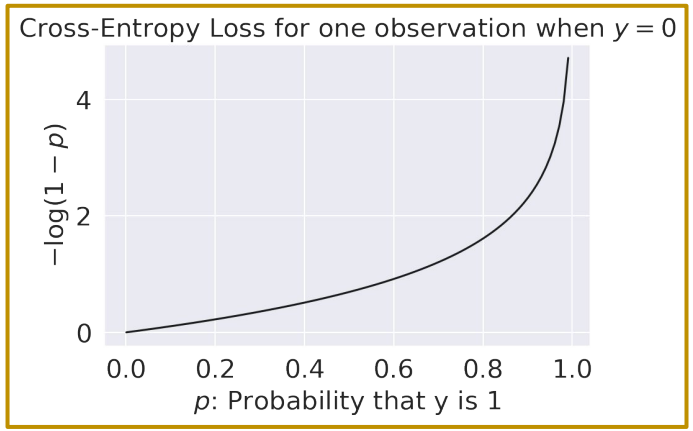
Cross-Entropy Loss Is a Good Measure of Model Error

Let y be a binary label {0, 1}, and p be the probability of the label being 1.

The **cross-entropy loss** is defined as

$$-\left(y\log(p) + (1-y)\log(1-p)\right)$$





For y = 1,

- p → 0: infinite loss
- p → 1: zero loss

For y = 0,

- p → 0: zero loss
- p → 1: infinite loss

Empirical Risk: Average Cross-Entropy Loss

For a single datapoint, the cross-entropy curve is convex. It has a global minimum.

$$-\left(y\log(p)+(1-y)\log(1-p)\right)$$

What about average cross-entropy loss, i.e., empirical risk?

For logistic regression, the empirical risk over a sample of size n is:

$$R(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\left(y_i\log(p_i)+(1-y_i)\log(1-p_i)\right)$$

(y_i is i-th response, and p_i is prob. that i-th response is 1)

$$= -\frac{1}{n}\sum_{i=1}^{n}\left(y_i\log(\sigma(X_i^T\theta)+(1-y_i)\log(1-\sigma(X_i^T\theta))\right)$$

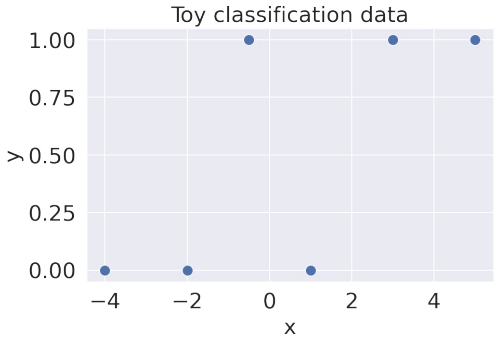
($p_i = \sigma(X_i^T\theta)$ and X_i is i-th feature vector)

The optimization problem is therefore to find the estimate $\hat{\theta}$ that minimizes R(θ):

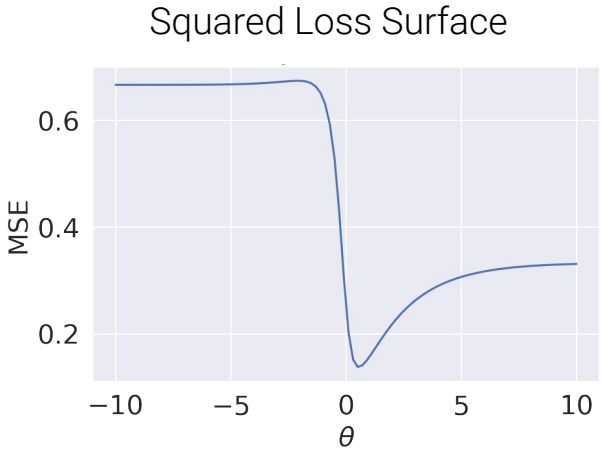
$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} -\frac{1}{n}\sum_{i=1}^{n}\left(y_i\log(\sigma(X_i^T\theta)+(1-y_i)\log(1-\sigma(X_i^T\theta))\right)$$

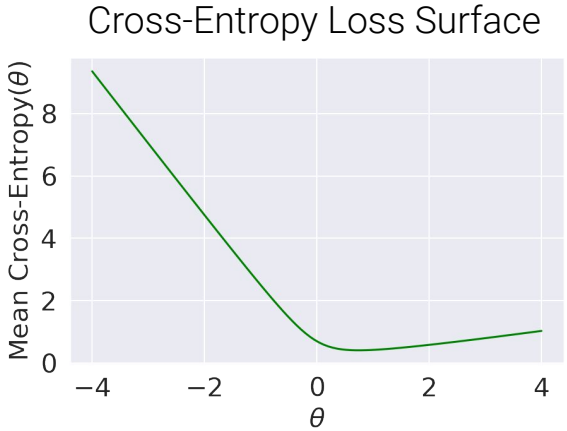
Convexity Proof By Picture

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} -\frac{1}{n}\sum_{i=1}^{n}\left(y_i \log(\sigma(X_i^T\theta)) + (1-y_i)\log(1-\sigma(X_i^T\theta))\right)$$



	x	y
0	-4.0	0
1	-2.0	0
2	-0.5	1
3	1.0	0
4	3.0	1
5	5.0	1





9774775

Maximum Likelihood Estimation

Lecture 23, Data 100 Spring 2023

Regression vs. Classification

Deriving the Logistic Regression Model

- The Sigmoid (Logistic) Function

The Logistic Regression Model

- Comparison to Linear Regression

Parameter Estimation

- Pitfalls of Squared Loss
- Cross-Entropy Loss
- **Maximum Likelihood Estimation**

Review of this Section

9774775

Let y be a binary label {0, 1}, and p be the probability of the label being 1.

The **cross-entropy loss** is defined as

$$-\left(y\log(p)+(1-y)\log(1-p)\right)$$

Matches the probabilistic modeling of logistic regression if $p=\hat{P}_\theta(Y=1|x)=\sigma(x^T\theta)$

Cross-Entropy loss addresses the 3 pitfalls of squared loss.

✅ Convex. No local minima for logistic regression.

✅ A good measure of model error. Strongly penalizes “off” predictions.

3. Conceptually sound.

Now let’s tackle #3.

The No-Input Binary Classifier

9774775

Suppose you observed some outcomes of a coin (1 = Heads, 0 = Tails):

$$\{0, 0, 1, 1, 1, 1, 0, 0, 0, 0\}$$

Training data has only responses $\mathbb{Y}$ (no features $\mathbb{X}$)

For the next flip, do you predict heads or tails?

The No-Input Binary Classifier

9774775

Suppose you observed some outcomes of a coin (1 = Heads, 0 = Tails):

$$\{0, 0, 1, 1, 1, 1, 0, 0, 0, 0\}$$

Training data has only responses $\mathbb{Y}$ (no features $\mathbb{X}$)

For the next flip, do you predict heads or tails?

A reasonable model is to **assume all flips are IID** (i.e., same coin; same prob. of heads θ).

Parameter θ: Probability that flip == 1 (Heads)

Prediction: 1 or 0

1. Of the below, which is the best theta θ? Why?

A. 0.8 B. 0.5 C. 0.4
D. 0.2 E. Something else

2. For the next flip, would you predict 1 or 0?

slido

Given a sample outcome of 10 flips of a coin with 4 heads and 6 tails, if we build a model that predicts the probability of heads with a constant model, what is the best theta value for y_hat = theta?

ⓘ Start presenting to display the poll results on this slide.

slido

Given a sample outcome of 10 flips of a coin with 4 heads and 6 tails, would you predict heads or tails for the next flip?

ⓘ Start presenting to display the poll results on this slide.

The No-Input Binary Classifier

9774775

Suppose you observed some outcomes of a coin (1 = Heads, 0 = Tails):

$$\{0, 0, 1, 1, 1, 1, 0, 0, 0, 0\}$$

Training data has only responses $\mathbb{Y}$ (no features $\mathbb{X}$)

For the next flip, do you predict heads or tails?

A reasonable model is to **assume all flips are IID** (i.e., same coin; same prob. of heads θ).

Parameter θ:
Probability that flip == 1 (Heads)

Prediction:
1 or 0

1. Of the below, which is the best theta θ? Why?

A. 0.8 **B.** 0.5 **C.** 0.4
D. 0.2 **E.** Something else

The No-Input Binary Classifier

9774775

Suppose you observed some outcomes of a coin (1 = Heads, 0 = Tails):

$$\{0, 0, 1, 1, 1, 1, 0, 0, 0, 0\}$$

Training data has only responses $\mathbb{Y}$ (no features $\mathbb{X}$)

For the next flip, do you predict heads or tails?

A reasonable model is to **assume all flips are IID** (i.e., same coin; same prob. of heads θ).

Parameter θ:
Probability that flip == 1 (Heads)

Prediction:
1 or 0

1. Of the below, which is the best theta θ? Why?

A. 0.8 **B.** 0.5 **C. 0.4**
D. 0.2 **E.** Something else

0.4 is the most "intuitive" for two reasons:
1. Frequency of heads in our data.
2. Maximizes the **likelihood** of our data.

The No-Input Binary Classifier

9774775

Suppose you observed some outcomes of a coin (1 = Heads, 0 = Tails):

$$\{0, 0, 1, 1, 1, 1, 0, 0, 0, 0\}$$

Training data has only responses $\mathbb{Y}$ (no features $\mathbb{X}$)

For the next flip, do you predict heads or tails?

A reasonable model is to **assume all flips are IID** (i.e., same coin; same prob. of heads θ).

Parameter θ:
Probability that flip == 1 (Heads)

Prediction:
1 or 0

1. Of the below, which is the best theta θ? Why?

A. 0.8 **B.** 0.5 **C. 0.4**
D. 0.2 **E.** Something else

2. For the next flip, would you predict 1 or 0?

The most frequent outcome in the sample which is tails.

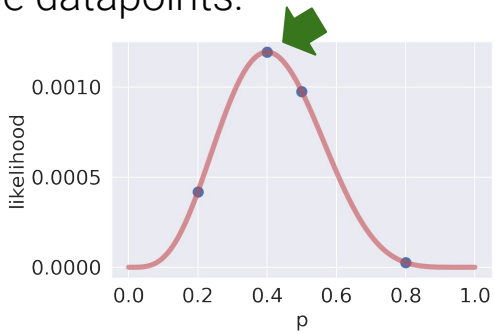
Likelihood of Data; Definition of Probability

A Bernoulli random variable Y with parameter p has distribution: $P(Y = y) = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{if } y = 0 \end{cases}$

Given that all flips are IID from the same coin (probability of heads = p), the **likelihood** of our data is **proportional to** the probability of observing the datapoints.

Training data: [0, 0, 1, 1, 1, 1, 0, 0, 0, 0]

Data likelihood: $p^4(1-p)^6$.



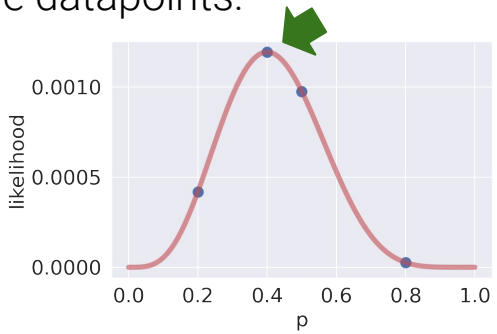
Likelihood of Data

A Bernoulli random variable Y with parameter p has distribution: $P(Y = y) = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{if } y = 0 \end{cases}$

Given that all flips are IID from the same coin (probability of heads = p), the **likelihood** of our data is **proportional to** the probability of observing the datapoints.

Training data: [0, 0, 1, 1, 1, 1, 0, 0, 0, 0]

Data likelihood: $p^4(1-p)^6$.



An example of a bad estimate is parameter $p = 0.1$ since the likelihood of observing the training data is going to be :

$$(0.1)^4(0.9)^6 = 0.000053$$

An example of a bad estimate is parameter $p = 0.4$ since the likelihood of observing the training data is going to be :

$$(0.4)^4(0.6)^6 = 0.001194$$

Generalization of the Coin Demo

For training data: {0, 0, 1, 1, 1, 1, 0, 0, 0, 0}

0.4 is the most "intuitive" θ for two reasons:

1. Frequency of heads in our data
2. Maximizes the **likelihood** of our data:

$$\hat{\theta} = \underset{\theta}{\text{argmax}} \left(\theta^4 (1-\theta)^6\right)$$

Parameter θ:
Probability that
IID flip == 1 (Heads)

Prediction:
1 or 0

How can we generalize this notion of likelihood to **any** random binary sample?

$$\{y_1, y_2, \ldots, y_n\} \Rightarrow \hat{\theta} = \underset{\theta}{\text{argmax}} \left(???\right)$$

data (1's and 0's)

likelihood

A Compact Representation of the Bernoulli Probability Distribution

9774775

How can we generalize this notion of likelihood to **any** random binary sample?

$$\{y_1, y_2, \ldots, y_n\} \Rightarrow \hat{\theta} = \underset{\theta}{\operatorname{argmax}} \, (???)$$

data (1's and 0's) — likelihood

Let Y be Bernoulli(p). The probability distribution can be written compactly:

$$P(Y = y) = p^y (1 - p)^{1-y}$$

For P(Y = **1**), only this term stays

For P(Y = **0**), only this term stays

(long, non-compact form):

$$P(Y = y) = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{if } y = 0 \end{cases}$$

Generalized Likelihood of Binary Data

How can we generalize this notion of likelihood to **any** random binary sample?

$$\{y_1, y_2, \ldots, y_n\} \Rightarrow \hat{\theta} = \underset{\theta}{\operatorname{argmax}} \, (???)$$

data (1's and 0's) — likelihood

Let Y be Bernoulli(p). The probability distribution can be written compactly:

$$P(Y = y) = p^y (1-p)^{1-y}$$

For P(Y = **1**), only this term stays

For P(Y = **0**), only this term stays

If binary data are **IID with same** probability p, then the likelihood of the data is:

$$\prod_{i=1}^{n} p^{y_i} (1-p)^{(1-y_i)}$$

Ex: {0, 0, 1, 1, 1, 1, 0, 0, 0, 0} → $p^4(1-p)^6$

Generalized Likelihood of Binary Data

9774775

How can we generalize this notion of likelihood to any random binary sample?

$$\{y_1, y_2, \ldots, y_n\} \Rightarrow \hat{\theta} = \underset{\theta}{\operatorname{argmax}}\,(???)$$

data (1's and 0's) — likelihood

Let Y be Bernoulli(p). The probability distribution can be written compactly:

$$P(Y = y) = p^y (1-p)^{1-y}$$

For P(Y = **1**), only this term stays

For P(Y = **0**), only this term stays

If binary data are **IID with same** probability p, then the likelihood of the data is:

$$\prod_{i=1}^{n} p^{y_i} (1-p)^{(1-y_i)}$$

If data are independent with **different** probability p_i, then the likelihood of the data is:

$$\prod_{i=1}^{n} p_i^{y_i} (1-p_i)^{(1-y_i)}$$

(spoiler: for logistic regression, $p_i = \sigma(X_i^T \theta)$)

Maximum Likelihood Estimation (MLE)

9774775

Our **maximum likelihood estimation** problem:

- For i = 1, 2, …, n, let Y_i be independent Bernoulli(p_i). Observe data $\{y_1, y_2, \ldots, y_n\}$.
- We'd like to estimate $p_1, p_2, \ldots, p_n$.

Find $\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n$ that **maximize** $\prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{(1-y_i)}$

Maximum Likelihood Estimation (MLE)

9774775

Our **maximum likelihood estimation** problem:

- For i = 1, 2, ..., n, let Y_i be independent Bernoulli(p_i). Observe data $\{y_1, y_2, \ldots, y_n\}$.
- We'd like to estimate $p_1, p_2, \ldots, p_n$.

Find $\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n$ that **maximize** $\prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{(1-y_i)}$

Equivalent, simplifying optimization problems (since we need to take the first derivative):

$$\underset{p_1, p_2, \ldots, p_n}{\textbf{maximize}} \quad \log\left(\prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{(1-y_i)}\right)$$

(log is an increasing function. If a > b, then log(a) > log(b).)

$$= \sum_{i=1}^{n} \log\left(p_i^{y_i}(1-p_i)^{(1-y_i)}\right) = \sum_{i=1}^{n} \left(y_i \log(p_i) + (1-y_i)\log(1-p_i)\right)$$

Maximum Likelihood Estimation (MLE)

9774775

Our **maximum likelihood estimation** problem:

- For i = 1, 2, ..., n, let Y_i be independent Bernoulli(p_i). Observe data $\{y_1, y_2, \ldots, y_n\}$.
- We'd like to estimate $p_1, p_2, \ldots, p_n$.

Find $\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n$ that **maximize** $\prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{(1-y_i)}$

Equivalent, simplifying optimization problems (since we need to take the first derivative):

maximize $p_1, p_2, \ldots, p_n$ $\log\left(\prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{(1-y_i)}\right)$ (log is an increasing function. If a > b, then log(a) > log(b).)

$$= \sum_{i=1}^{n} \log\left(p_i^{y_i}(1-p_i)^{(1-y_i)}\right) = \sum_{i=1}^{n} \left(y_i \log(p_i) + (1-y_i)\log(1-p_i)\right)$$

minimize $p_1, p_2, \ldots, p_n$ $-\frac{1}{n}\sum_{i=1}^{n} \left(y_i \log(p_i) + (1-y_i)\log(1-p_i)\right)$

Maximizing Likelihood == Minimizing Average Cross-Entropy

maximize $p_1, p_2, \ldots, p_n$ $\prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{(1-y_i)}$

Log is increasing; max/min properties

minimize $p_1, p_2, \ldots, p_n$ $-\frac{1}{n}\sum_{i=1}^{n}\left(y_i \log(p_i) + (1-y_i)\log(1-p_i)\right)$

For logistic regression, let $p_i = \sigma(X_i^T\theta)$

minimize θ $-\frac{1}{n}\sum_{i=1}^{n}\left(y_i \log(\sigma(X_i^T\theta)) + (1-y_i)\log(1-\sigma(X_i^T\theta))\right)$

Cross-Entropy Loss!!

Minimizing cross-entropy loss is equivalent to **maximizing the likelihood of the training data**.

- We are choosing the model parameters that are "most likely", given this data.

Assumption: all data drawn **independently** from the same logistic regression model with parameter θ

- It turns out that many of the model + loss combinations we've seen can be motivated using MLE (OLS, Ridge Regression, etc.)
- You will study MLE further in probability and ML classes. But now you know it exists.

We Did it!

9774775

	Regression ($y \in \mathbb{R}$)	Classification ($y \in \{0, 1\}$)
1. Choose a model ✅	Linear Regression $\hat{y} = f_\theta(x) = x^T\theta$	Logistic Regression $\hat{P}_\theta(Y = 1\|x) = \sigma(x^T\theta)$
2. Choose a loss function ✅	Squared Loss or Absolute Loss	Average Cross-Entropy Loss $-\frac{1}{n}\sum_{i=1}^{n}\left(y_i \log(\sigma(X_i^T\theta)) + (1 - y_i)\log(1 - \sigma(X_i^T\theta))\right)$
3. Fit the model	Regularization Sklearn/Gradient descent	Regularization Sklearn/Gradient descent
4. Evaluate model performance	R^2, Residuals, etc.	?? (next time)

LECTURE 23

Logistic Regression I

Content credit: Acknowledgments