

# Overview

## Classification basics + Thresholding

### Logistic Regression

#### Sigmoid function

What Are Odds?

Why Use Log Odds?

Modeling Log Odds with a Line

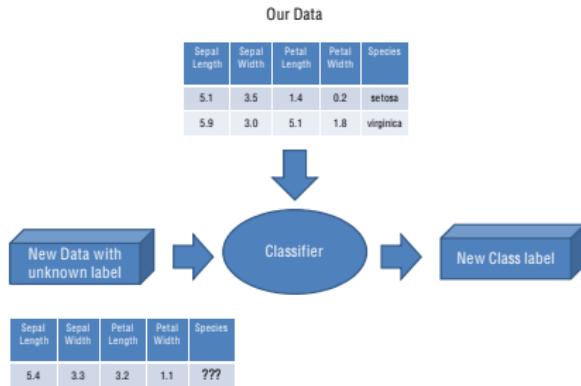
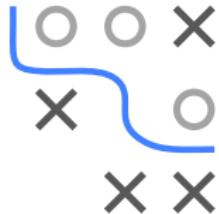
Deriving the Sigmoid Function

Conclusion

# Classification basics

# CLASSIFICATION

Learn functions that assign class labels to observation / feature vectors.  
Each observation belongs to exactly one class. The main difference to regression is that the target is categorical.



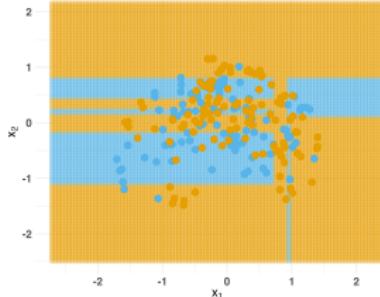
# BINARY AND MULTICLASS TASKS

Tasks have a finite number of (unordered) classes.

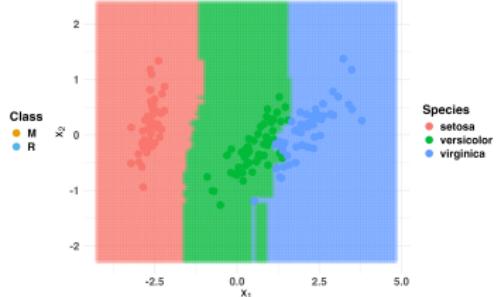
They can be **binary** or **multiclass**.



Sonar: binary classification

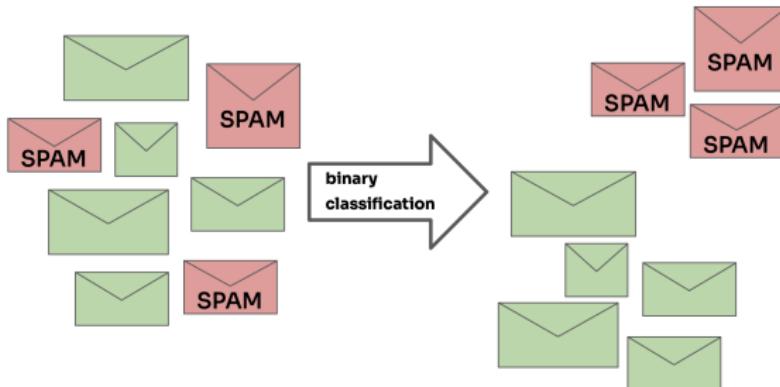


Iris: multiclass classification



## BINARY CLASSIFICATION TASK - EXAMPLES

- Credit risk prediction, based on personal data and transactions
- Spam detection, based on textual features
- Churn prediction, based on customer behavior
- Predisposition for specific illness, based on genetic data



Supervised Classification

113 / 598

## MULTICLASS TASK - IRIS

The iris dataset was introduced by the statistician Ronald Fisher and is one of the most frequent used data sets. Originally, it was designed for linear discriminant analysis.



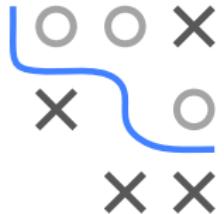
Setosa



Versicolor



Virginica

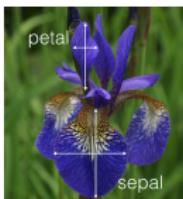


Source:

[https://en.wikipedia.org/wiki/Iris\\_flower\\_data\\_set](https://en.wikipedia.org/wiki/Iris_flower_data_set)

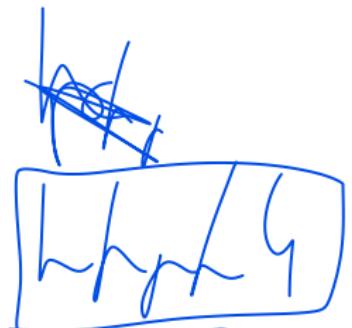
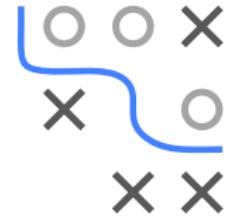
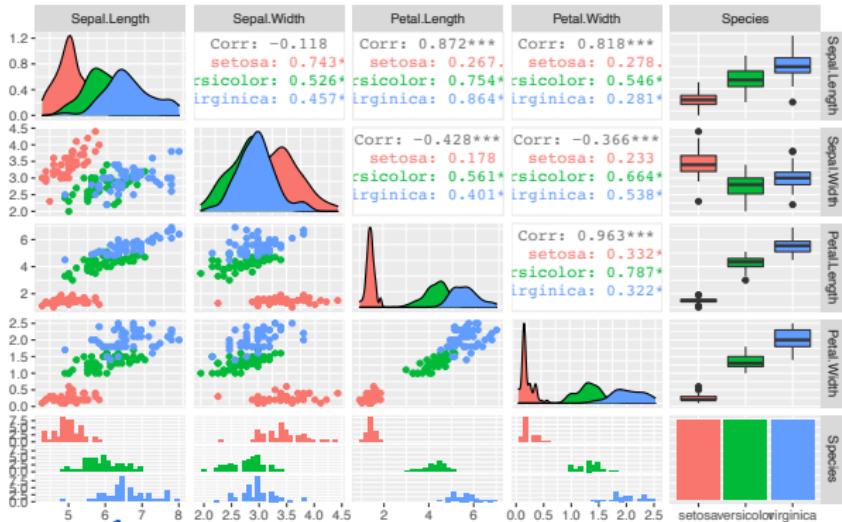
# MULTICLASS TASK - IRIS

- 150 iris flowers
- Predict subspecies
- Based on sepal and petal length / width in [cm]



```
##      Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1:          5.1       3.5        1.4       0.2   setosa
## 2:          4.9       3.0        1.4       0.2   setosa
## 3:          4.7       3.2        1.3       0.2   setosa
## 4:          4.6       3.1        1.5       0.2   setosa
## 5:          5.0       3.6        1.4       0.2   setosa
## ---
## 146:         6.7       3.0        5.2       2.3 virginica
## 147:         6.3       2.5        5.0       1.9 virginica
## 148:         6.5       3.0        5.2       2.0 virginica
## 149:         6.2       3.4        5.4       2.3 virginica
## 150:         5.9       3.0        5.1       1.8 virginica
```

# MULTICLASS TASK - IRIS



(uhfb) handw. hmp

Supervised Classification

117 / 598

## PROBABILISTIC CLASSIFIERS

- Construct  $g$  probability functions

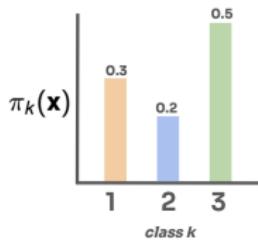
$$\pi_1, \dots, \pi_g : \mathcal{X} \rightarrow [0, 1], \sum_{k=1}^g \pi_k(\mathbf{x}) = 1$$

- Predicted class is usually the one with max probability

0.5 →

$$h(\mathbf{x}) = \arg \max_{k \in \{1, \dots, g\}} \pi_k(\mathbf{x})$$

- For  $g = 2$ , single  $\pi(\mathbf{x})$  is constructed, which models the predicted probability for the positive class (natural to encode  $\mathcal{Y} = \{0, 1\}$ )



predict class  
with highest prob.

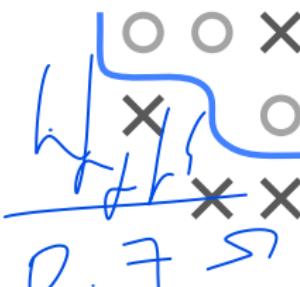
$$h(\mathbf{x}) = \arg \max_{k \in \{1, \dots, g\}} \pi_k(\mathbf{x}) = 3$$

Supervised Classification



1 - 0.4

0.7



0.5

= [0.3, 0.2, 0.5]

122 / 598

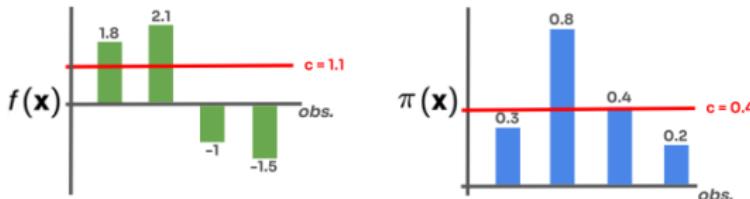
$f(x) \rightarrow 0.6 \rightarrow 1$

## THRESHOLDING

- For imbalanced cases or class with costs, we might want to deviate from the standard conversion of scores to classes
- Introduce basic concept (for binary case) and add details later
- Convert scores or probabilities to class outputs by thresholding:  
 $h(\mathbf{x}) := [\pi(\mathbf{x}) \geq c]$  or  $h(\mathbf{x}) := [f(\mathbf{x}) \geq c]$  for some threshold  $c$
- Standard thresholds:  $c = 0.5$  for probabilities,  $c = 0$  for scores

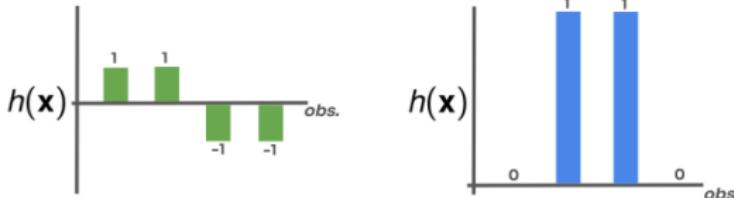


0. 4 8



# THRESHOLDING

- For imbalanced cases or class with costs, we might want to deviate from the standard conversion of scores to classes
- Introduce basic concept (for binary case) and add details later
- Convert scores or probabilities to class outputs by thresholding:  
 $h(\mathbf{x}) := [\pi(\mathbf{x}) \geq c]$  or  $h(\mathbf{x}) := [f(\mathbf{x}) \geq c]$  for some threshold  $c$
- Standard thresholds:  $c = 0.5$  for probabilities,  $c = 0$  for scores



## Logistic Regression

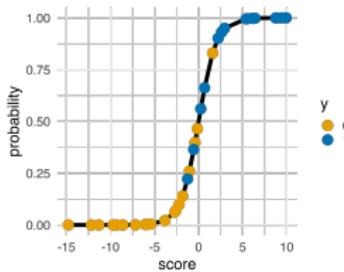
Reg  $\rightarrow$  C R  
cls  $\rightarrow$  { . . . }

Logistic Regression

# Introduction to Machine Learning

## Classification

### Logistic Regression



#### Learning goals

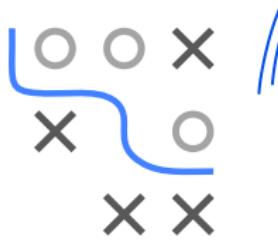
- Hypothesis space of LR
- Log-Loss derivation
- Intuition for loss
- LR as linear classifier

# MOTIVATION

- Let's build a **discriminant** approach, for binary classification, as a probabilistic classifier  $\pi(\mathbf{x} | \theta)$
- We encode  $y \in \{0, 1\}$  and use ERM:

$$\arg \min_{\theta \in \Theta} \mathcal{R}_{\text{emp}}(\theta) = \arg \min_{\theta \in \Theta} \sum_{i=1}^n L\left(y^{(i)}, \pi\left(\mathbf{x}^{(i)} | \theta\right)\right)$$

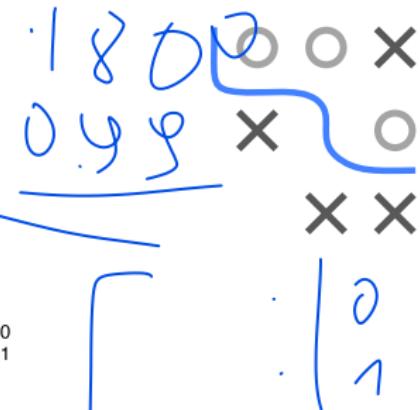
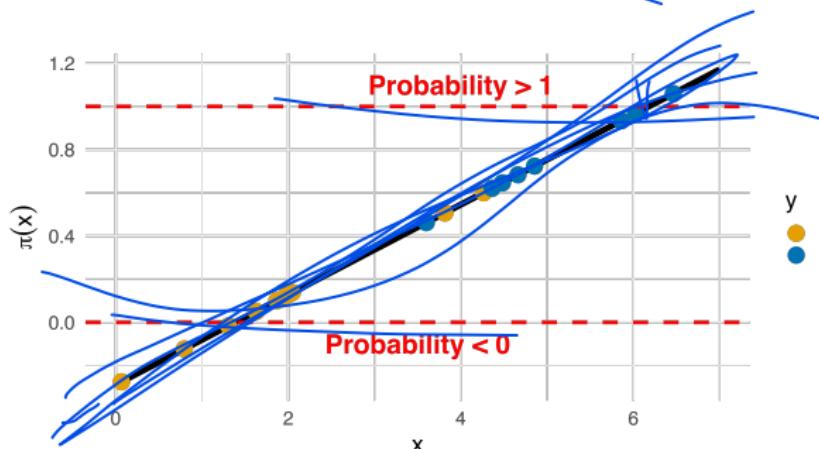
- We want to "copy" over ideas from linear regression
- In the above, our model structure should be "mainly" linear and we need a loss function



## DIRECT LINEAR MODEL FOR PROBABILITIES

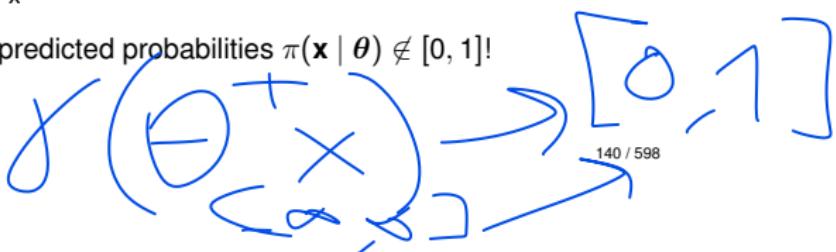
We could directly use an LM to model  $\pi(\mathbf{x} | \theta) = \underline{\theta^\top \mathbf{x}}$ .

And use L2 loss in ERM.



But: This obviously will result in predicted probabilities  $\pi(\mathbf{x} | \theta) \notin [0, 1]!$

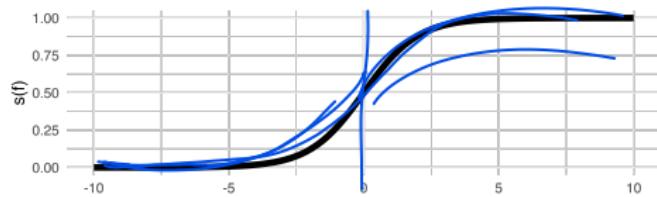
Supervised Classification



## HYPOTHESIS SPACE OF LR

To avoid this, logistic regression “squashes” the estimated linear scores  $\theta^\top \mathbf{x}$  to  $[0, 1]$  through the logistic function  $s$ :

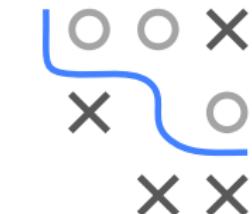
$$\pi(\mathbf{x} | \theta) = \frac{\exp(\theta^\top \mathbf{x})}{1 + \exp(\theta^\top \mathbf{x})} = \frac{1}{1 + \exp(-\theta^\top \mathbf{x})} = s(\theta^\top \mathbf{x}) = s(f(\mathbf{x}))$$



⇒ Hypothesis space of LR:

$$\mathcal{H} = \left\{ \pi : \mathcal{X} \rightarrow [0, 1] \mid \pi(\mathbf{x} | \theta) = s(\theta^\top \mathbf{x}) \mid \theta \in \mathbb{R}^{p+1} \right\}$$

Supervised Classification

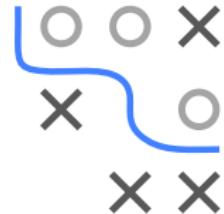
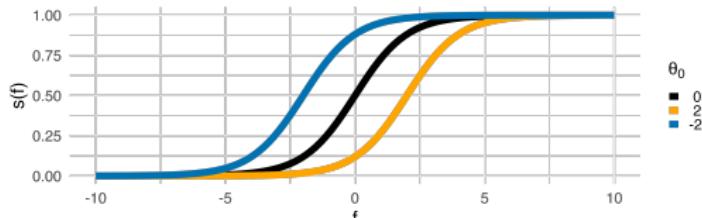


Handwritten notes on the right side of the slide:

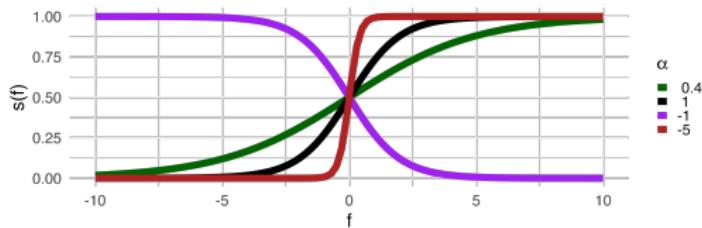
- A graph of the logistic function  $s(z) = \frac{e^z}{1 + e^z}$  is shown.
- The formula  $\frac{1}{1 + e^{-z}}$  is written inside a large oval.
- The value "141 / 598" is written near the bottom right of the oval.

## LOGISTIC FUNCTION

Intercept  $\theta_0$  shifts  $\pi = s(\theta_0 + f) = \frac{\exp(\theta_0 + f)}{1 + \exp(\theta_0 + f)}$  horizontally

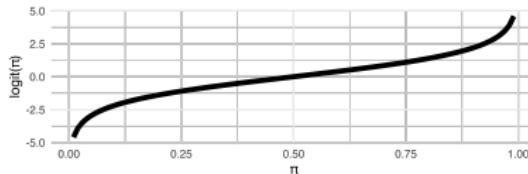


Scaling  $f$  like  $s(\alpha f) = \frac{\exp(\alpha f)}{1 + \exp(\alpha f)}$  controls slope and direction



# THE LOGIT

The inverse  $s^{-1}(\pi) = \log\left(\frac{\pi}{1-\pi}\right)$  where  $\pi$  is a probability is called **logit** (also called **log odds** since it is equal to the logarithm of the odds  $\frac{\pi}{1-\pi}$ )



- Positive logits indicate probabilities  $> 0.5$  and vice versa
- E.g.: if  $p = 0.75$ , odds are  $3 : 1$  and logit is  $\log(3) \approx 1.1$
- Features  $\mathbf{x}$  act linearly on logits, controlled by coefficients  $\theta$ :

$$s^{-1}(\pi(\mathbf{x})) = \log\left(\frac{\pi(\mathbf{x})}{1 - \pi(\mathbf{x})}\right) = \boldsymbol{\theta}^T \mathbf{x}$$

## DERIVING LOG-LOSS

We need to find a suitable loss function for **ERM**. We look at likelihood which multiplies up  $\pi(\mathbf{x}^{(i)} | \theta)$  for positive examples, and  $1 - \pi(\mathbf{x}^{(i)} | \theta)$  for negative.

$$\mathcal{L}(\theta) = \prod_{i \text{ with } y^{(i)}=1} \pi(\mathbf{x}^{(i)} | \theta) \cdot \prod_{i \text{ with } y^{(i)}=0} (1 - \pi(\mathbf{x}^{(i)} | \theta))$$

We can now cleverly combine the 2 cases by using exponents (note that only one of the 2 factors is not 1 and "active"):

$$\mathcal{L}(\theta) = \prod_{i=1}^n \pi(\mathbf{x}^{(i)} | \theta)^{y^{(i)}} \cdot (1 - \pi(\mathbf{x}^{(i)} | \theta))^{1-y^{(i)}}$$

$$0.8 \cdot 0.4 \quad 0.3 \\ 1 \quad 1 \quad 0$$

$$\log(\cdot) + (1-\cdot)$$

## DERIVING LOG-LOSS / 2

Taking the log to convert products into sums:

$$\ell(\theta) = \log \mathcal{L}(\theta) = \sum_{i=1}^n \log \left( \pi \left( \mathbf{x}^{(i)} | \theta \right)^{y^{(i)}} \left( 1 - \pi \left( \mathbf{x}^{(i)} | \theta \right) \right)^{1-y^{(i)}} \right)$$

$$= \sum_{i=1}^n y^{(i)} \log \left( \pi \left( \mathbf{x}^{(i)} | \theta \right) \right) + (1-y^{(i)}) \log \left( 1 - \pi \left( \mathbf{x}^{(i)} | \theta \right) \right)$$

Since we want to minimize the risk, we work with the negative  $\ell(\theta)$ :

$$-\ell(\theta) = \sum_{i=1}^n -y^{(i)} \log \left( \pi \left( \mathbf{x}^{(i)} | \theta \right) \right) - (1-y^{(i)}) \log \left( 1 - \pi \left( \mathbf{x}^{(i)} | \theta \right) \right)$$

$$\approx y_i \cdot -\log(0.8) - (1-y_i) \log(1-0.8)$$

$$[0.8 | 0.1] - 1 \cdot \log(0.8) - (1-0.8) \log(1-0.8)$$

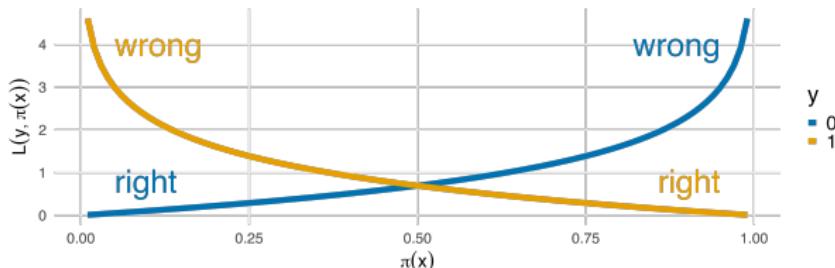
145 / 598

## BERNOULLI / LOG LOSS

The resulting loss

$$L(y, \pi) = -y \log(\pi) - (1 - y) \log(1 - \pi)$$

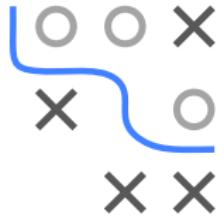
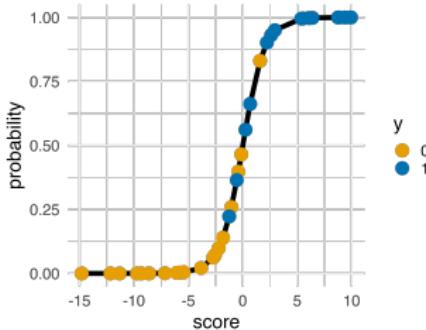
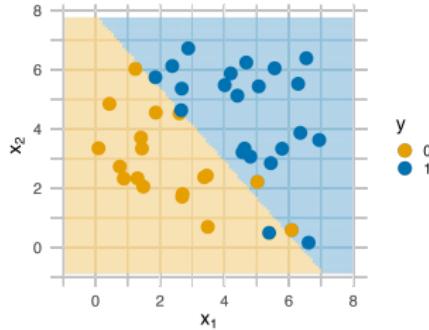
is called **Bernoulli, binomial, log or cross-entropy** loss



- Penalizes confidently wrong predictions heavily
- Is used for many other classifiers, e.g., in NNs or boosting

## LOGISTIC REGRESSION IN 2D

LR is a linear classifier, as  $\pi(\mathbf{x} | \theta) = s(\theta^\top \mathbf{x})$  and  $s$  is isotonic.



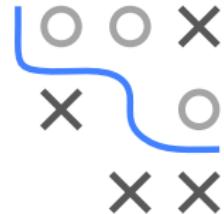
# OPTIMIZATION

- Log-Loss is convex, under regularity conditions LR has a unique solution (because of its linear structure), but not an analytical one
- To fit LR we use numerical optimization, e.g., Newton-Raphson
- If data is linearly separable, the optimization problem is unbounded and we would not find a solution; way out is regularization
- Why not use least squares on  $\pi(\mathbf{x}) = s(f(\mathbf{x}))$ ?  
Answer: ERM problem is not convex anymore :(
- We can also write the ERM as

$$\arg \min_{\boldsymbol{\theta} \in \Theta} \mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) = \arg \min_{\boldsymbol{\theta} \in \Theta} \sum_{i=1}^n L\left(y^{(i)}, f\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)\right)$$

With  $f(\mathbf{x} \mid \boldsymbol{\theta}) = \boldsymbol{\theta}^T \mathbf{x}$  and  $L(y, f) = -yf + \log(1 + \exp(f))$

This combines the sigmoid with the loss and shows a convex loss directly on a linear function



Sigmoid function

Sigmoid function



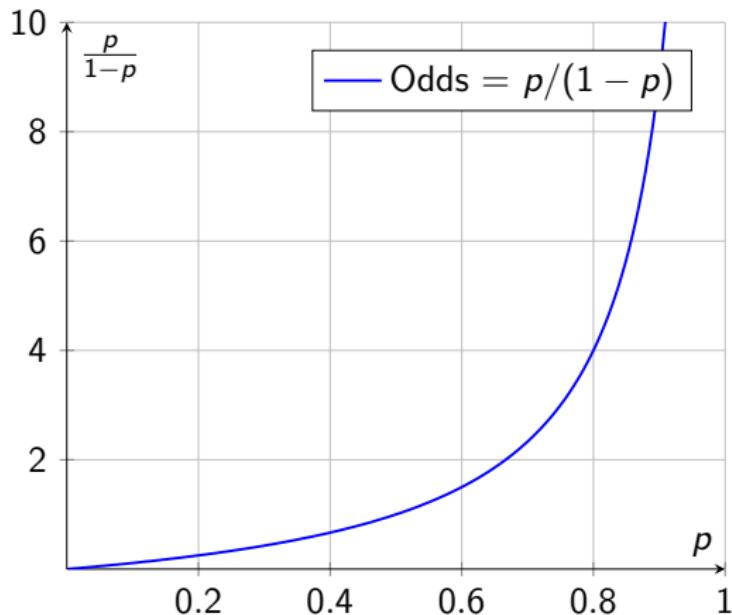
## Odds: Concept and Definition

- ▶ Consider a binary outcome: success (probability  $p$ ) vs. failure (probability  $1 - p$ ).
- ▶ **Odds** of success are defined as:

$$\text{odds} = \frac{p}{1 - p}.$$

- ▶ If  $p = 0.5$ , odds = 1 (success and failure equally likely).
- ▶ If  $p > 0.5$ , odds > 1 (success more likely).
- ▶ If  $p < 0.5$ , odds < 1 (failure more likely).

# Plot: Odds vs. Probability



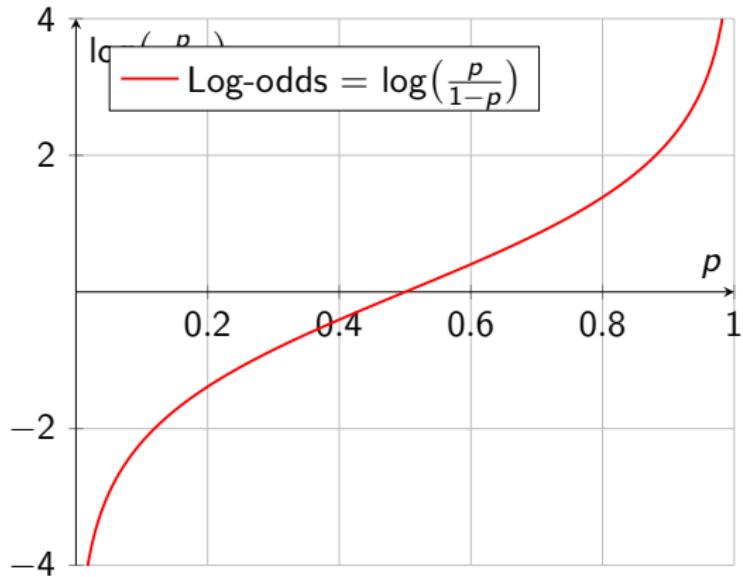
# Log-Odds (Logit)

**Log-odds** or **logit** is:

$$\log\left(\frac{p}{1-p}\right).$$

- ▶ Maps odds  $(0, \infty)$  to the entire real line  $(-\infty, +\infty)$ .
- ▶ Mathematical convenience: Addition in log-space corresponds to multiplication in probability space. - An event being 5 times bigger than another (odds ratio = 5) translates to  $\log(5)$ , and an event being 5 times smaller (odds ratio =  $1/5$ ) translates to  $\log(1/5) = -\log(5)$ . Hence, these two situations have the **\*\*same absolute effect\*\*** in log-scale, but opposite signs.
- ▶ Central to logistic regression: we assume a *linear* relationship in the log-odds.

# Plot: Log-Odds vs. Probability



# Linear Model for Log-Odds

- ▶ In **logistic regression**, we assume:

$$\log \left( \frac{p}{1-p} \right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n.$$

- ▶ Let  $z = \beta_0 + \sum_{i=1}^n \beta_i x_i$ .
- ▶ Then:  $\log \left( \frac{p}{1-p} \right) = z$ .

# From Log-Odds to Probability

Starting with:

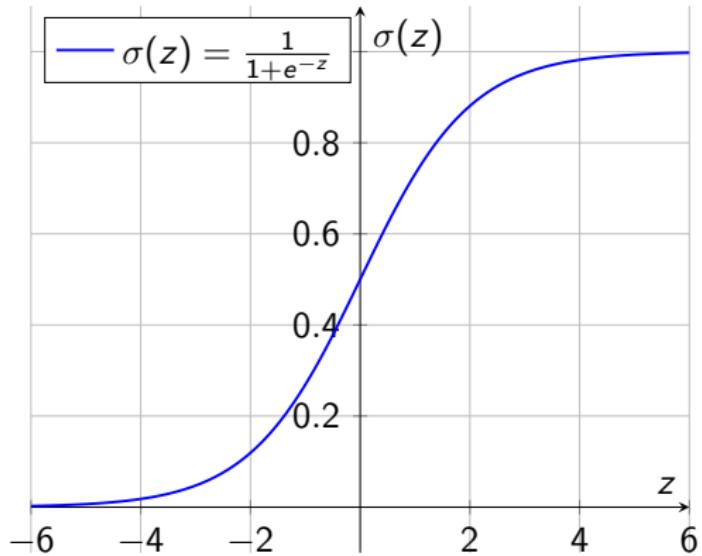
$$\log \left( \frac{p}{1-p} \right) = z \implies \frac{p}{1-p} = e^z.$$

Then solving for  $p$ :

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

- ▶ This function  $\sigma(z) = \frac{1}{1+e^{-z}}$  is known as the **sigmoid** (or logistic) function.
- ▶ Maps  $z \in (-\infty, +\infty)$  to  $p \in (0, 1)$ .

# Plot: Sigmoid Function



## Key Takeaways

1. **Odds:**  $\frac{p}{1 - p}$ .
2. **Log-odds (logit):**  $\log\left(\frac{p}{1 - p}\right)$  transforms  $(0, \infty)$  to  $(-\infty, +\infty)$ .
3. **Linear modeling:** logistic regression assumes log-odds is a linear combination of features.
4. **Sigmoid function:** arises by solving for  $p$  from the log-odds. Maps real numbers to probabilities (0 to 1).