zhaw Life Sciences and Facility Management

Institute of Computational Life Sciences

Logistic regression

Machine Learning

Norman Juchler

What you will learn today

- Logistic **regression** is a method used to solve **classification** problems! 😵 💫

- Logistic regression is a very basic method, but often represents a surprisingly strong baseline.

Recap: Types of classification problems

- Binary classification
  - The target variable only has values 0 or 1

Spam
No spam

- Multi-class or multinomial classification
  - The target variable can have K different values

Dog
Cat
Horse
Fish
Bird

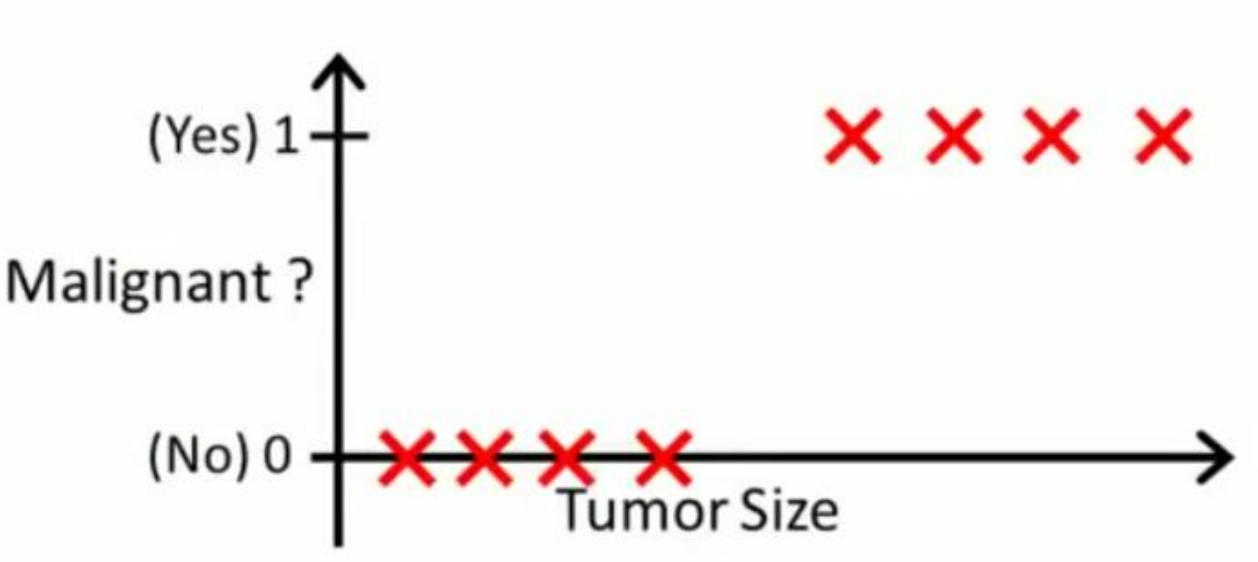
- Multi-label classification
  - Several different labels can be attached to a given sample
  - Use a separate classifier for each label

Genres:
Action
Crime
Thriller

Possible genres: Action, crime, thriller, comedy, drama, documentation, ...

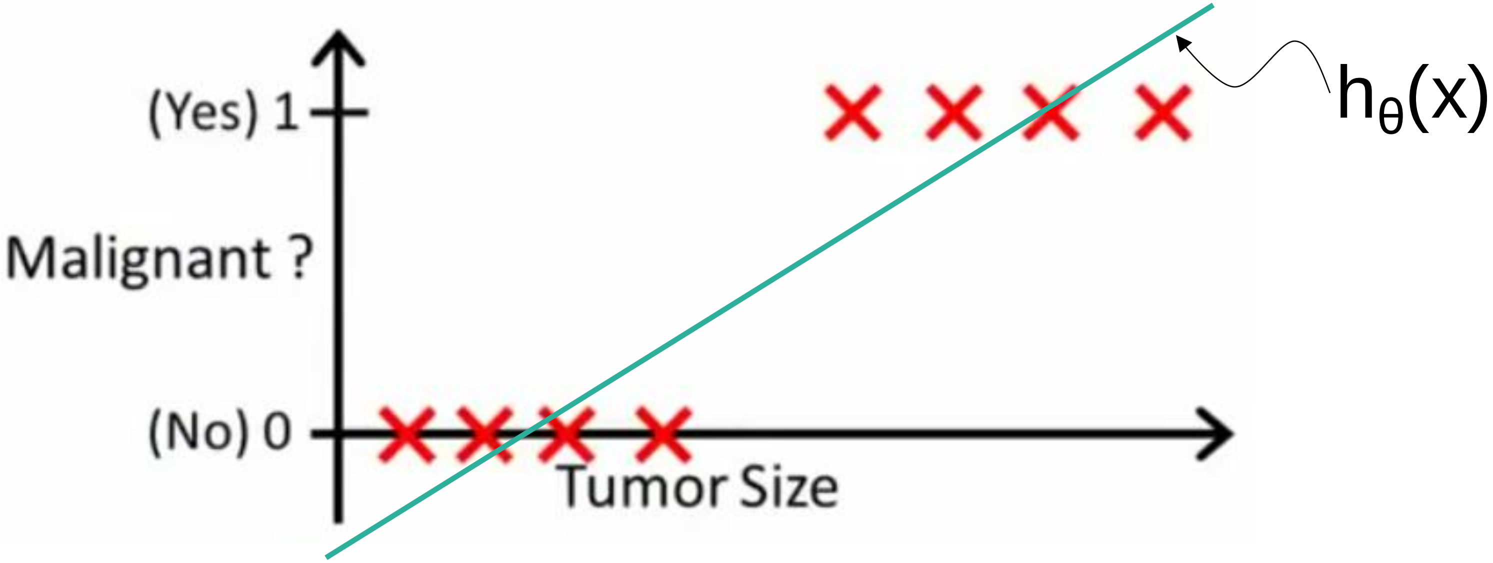
Motivational example

- How to solve this classification problem?



Motivational example

- Can we use linear regression?



- **Idea**:

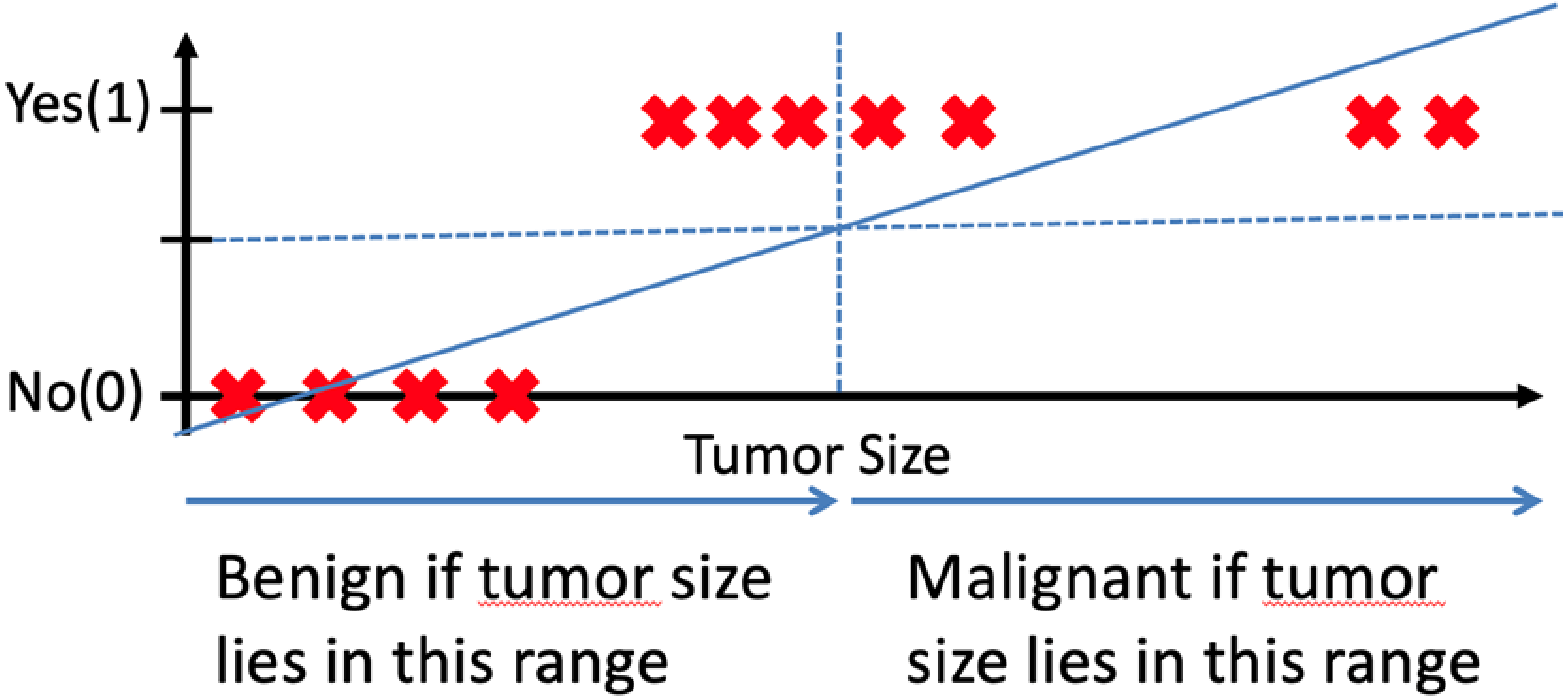
Threshold classifier output $h_\theta(x)$ at 0.5:

If $h_\theta(x) \geq 0.5$, predict "y = 1"

If $h_\theta(x) < 0.5$, predict "y = 0"

Motivational example: Problems

- Linear regression not robust with respect to outliers!
- Values of $h_\theta(x)$ can be <0 and >1



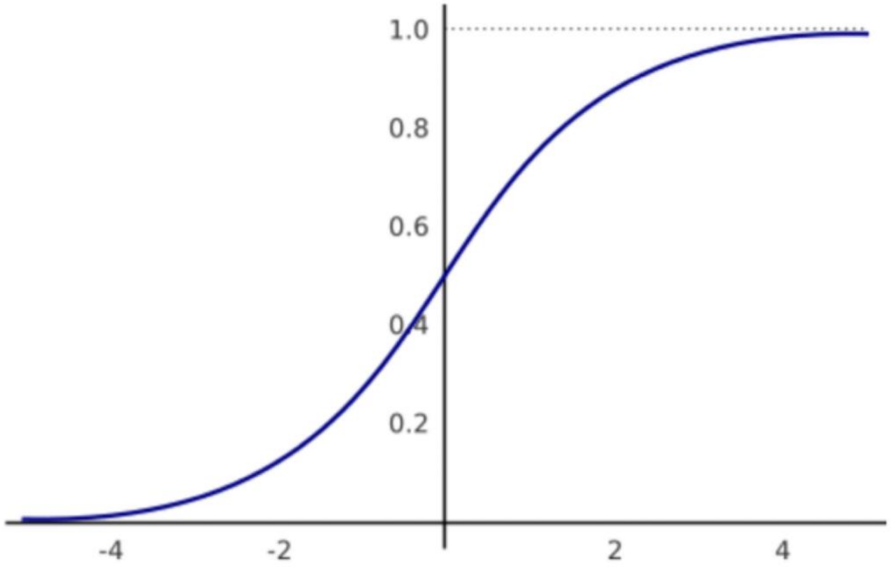
Defintions

The logistic function (aka sigmoid)

- The **logistic** function is a differential S-shaped function

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

- It describes a transition between 0 and 1
- It converts real values into a number between 0 and 1
- The output could potentially be used as a probability



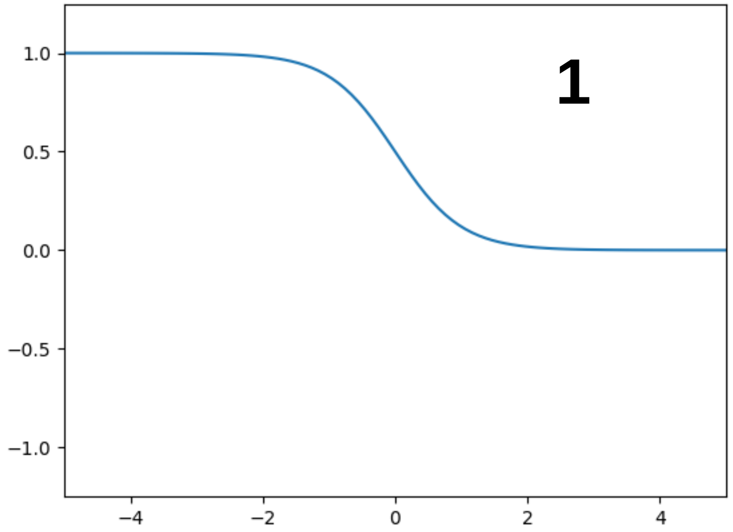
Exercise: Match the Parameters to the Graph

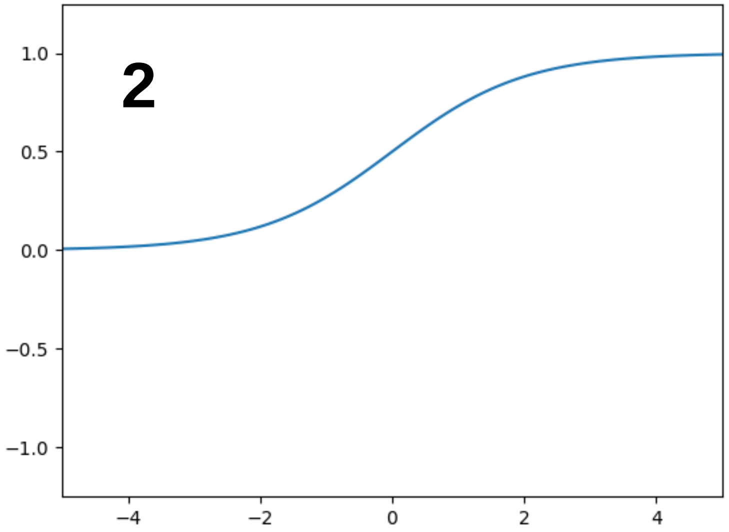
A. $\theta_0 = 0$ and $\theta_1 = -2$

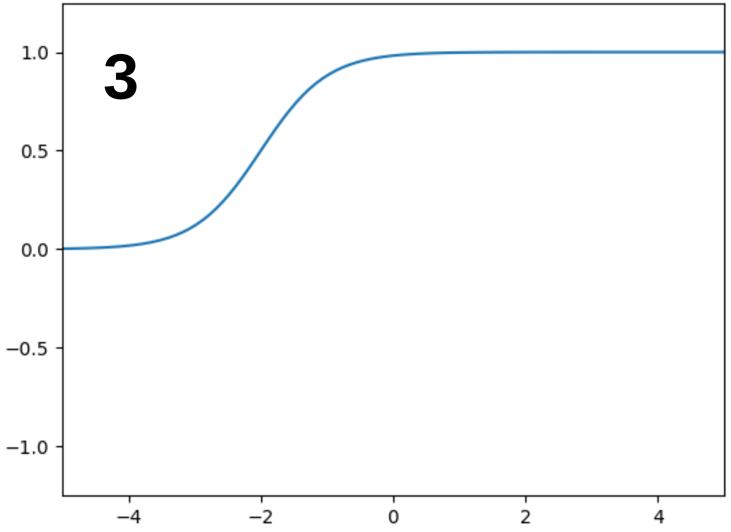
B. $\theta_0 = +4$ and $\theta_1 = 2$

C. $\theta_0 = 0$ and $\theta_1 = 1$

$$h_\theta = \sigma(\theta_0 + \theta_1 x) \qquad \text{where } \sigma(z) = \frac{1}{1+e^{-z}}$$





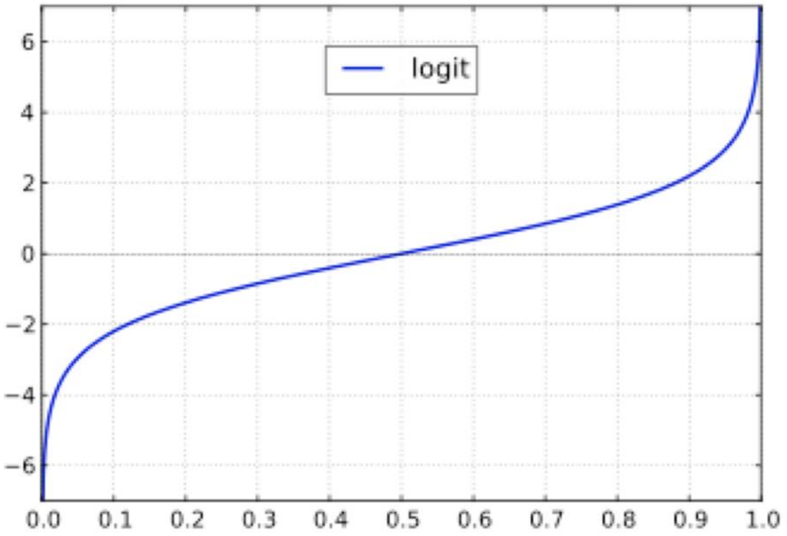


The logit function

- The logit function is the inverse of the sigmoid function:

$$\mathrm{logit}(p) = \sigma^{-1}(p) = \ln\left(\frac{p}{1-p}\right) \quad \text{for} \quad p \in (0,1)$$

- It equals the log of the odds ratio
- The inputs are probabilities
- The outputs are real numbers



The softmax function

- The softmax function generalizes the sigmoid to multiple dimensions

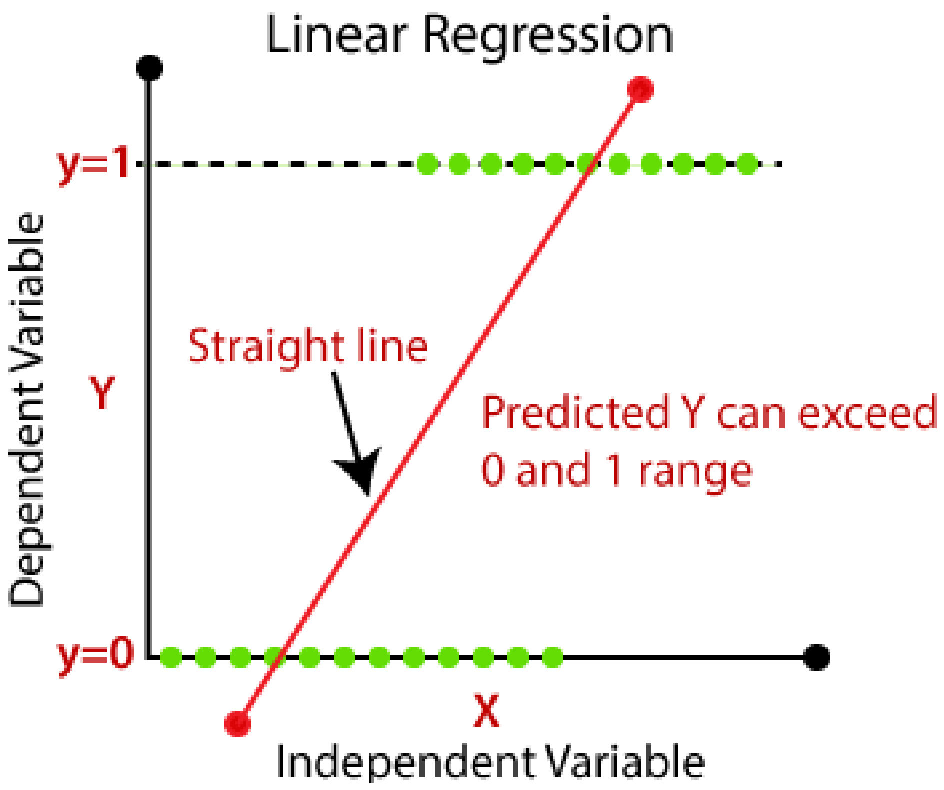
$$\sigma(\boldsymbol{x})_k = \frac{e^{x_k}}{\sum_{j=0}^{K} e^{x_j}}$$

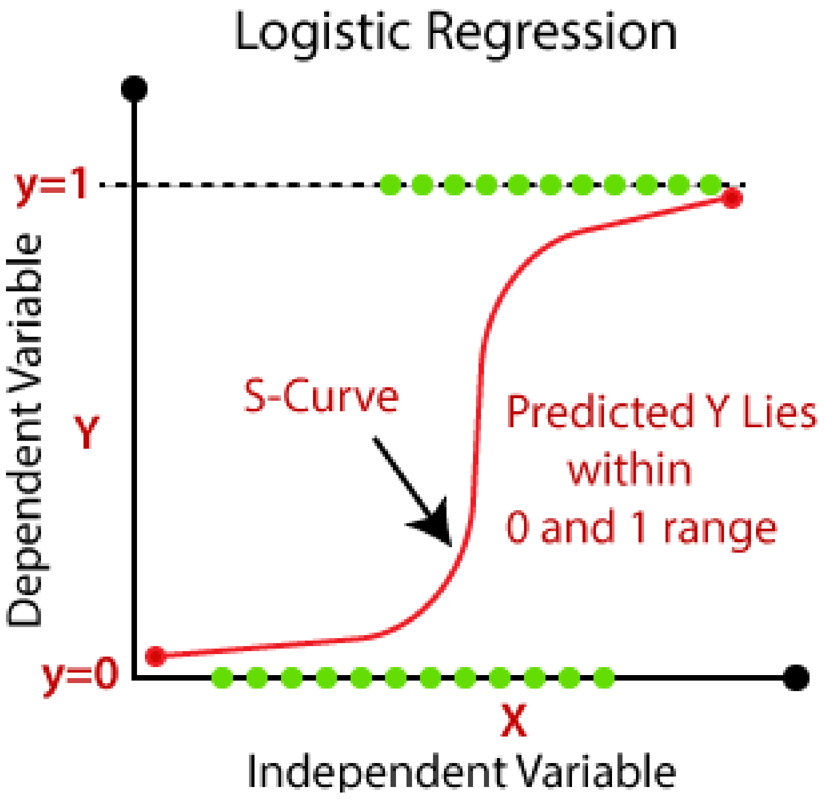
- Its input is a K-dimensional vector x
- The output is a vector of K probability values, with these properties:
  - They are proportional to the exponentials of the input numbers
  - Their sum is equal to 1

Basic ideas

Linear regression for classification problems

- Can we still use linear regression to solve a classification problem?

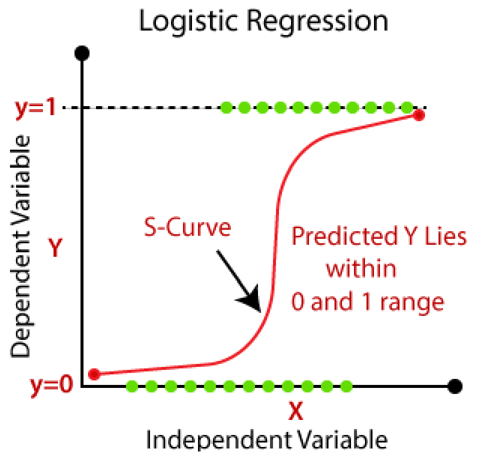




Linear regression for classification problems

- Can we still use linear regression to solve a classification problem?

- Idea:
  - Apply the logistic function to the regression output
  - Use an adapted loss function for classification
  - Solve a minimization problem using gradient decent (this is called logistic regression)



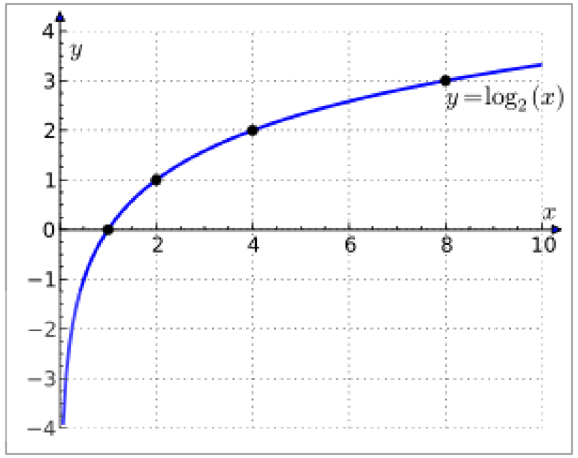
- We can interpret the resulting predictions of logistic regression as the probability of a sample belonging to the class y=1.
- We can convert this into a class assignment by selecting the class with the higher probability

Loss function for classification

- Cross-entropy loss for K classes:

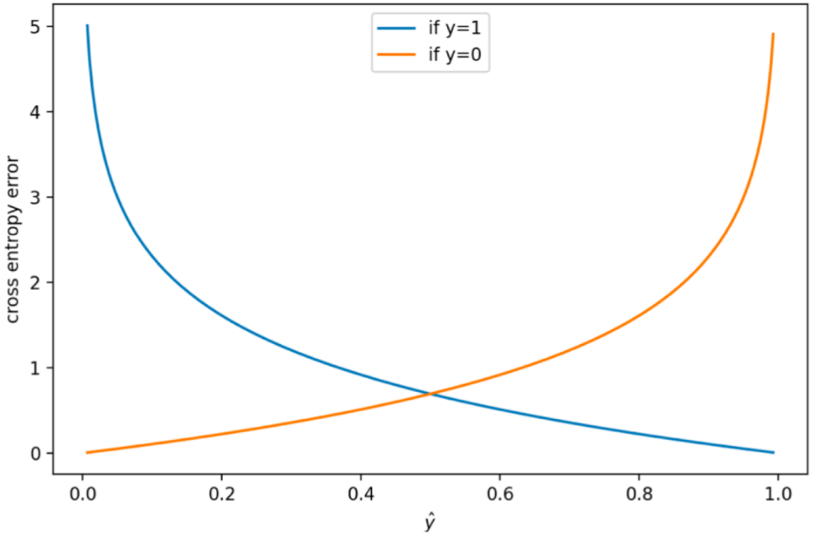
$$\mathcal{L}_K = -\sum_{c=1}^{K}\sum_{i=1}^{n} y_{i,c}\log(\hat{y}_{i,c})$$

Recap: Function $\log_2(x)$



- For binary classification (K=2)

$$\mathcal{L}_2 = \sum_{i=1}^{n} -y_i\log(\hat{y}_i) - (1 - y_i)\log(1 - \hat{y}_i)$$



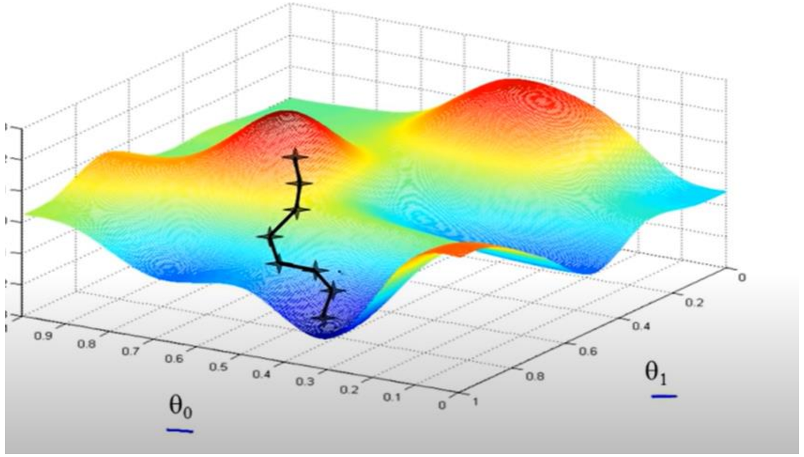
Gradient descent for logistic regression

$$\mathrm{L}(\theta) = -\frac{1}{M}\sum_{m=1}^{M}\left[y^{(m)}\log h_\theta\left(x^{(m)}\right) + (1 - y^{(m)})\log(1 - h_\theta\left(x^{(m)}\right))\right]$$

- This leads to the following update rule
  - Repeat until convergence

$$\theta_j = \theta_j - \alpha\frac{\partial}{\partial\theta_j}J(\theta)$$

  - This is similar to the least mean squares update rule in linear regression, except that here we use a non-linear function h_θ.



Performance metric for (binary) classification

For regression problems, we care about **how "close" the prediction was** to the true answer. Over an entire dataset, we calculate the mean squared error, residual sum of squares, etc.:

$$RSS = \sum_{m=1}^{M} (h_\theta\left(x^{(m)}\right) - y^{(m)})^2$$

For classification problems, we care about **whether the model predicted the correct class**. Over an entire dataset, we thus calculate the proportion of examples classified correctly, sometimes called the **accuracy**:

$$Acc = \sum_{m=1}^{M} 1[h_\theta\left(x^{(m)}\right) = y^{(m)}]$$

where 1[…] is an indicator function that equals 1 if the condition is true, 0 if it is false.

Comparison

Linear Regression

Hypothesis: $h_\theta(x) = \theta^T x$

Cost Function: Mean Squared Error

$$J(\theta) = \frac{1}{2M}\sum_{m=1}^{M}\left(h_\theta\left(x^{(m)}\right) - y^{(m)}\right)^2$$
$$= \frac{1}{2M}\sum_{m=1}^{M}\left(\theta^T x^{(m)} - y^{(m)}\right)^2$$

Logistic Regression

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

Loss Function "Log-Likelihood":

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

Cost Function

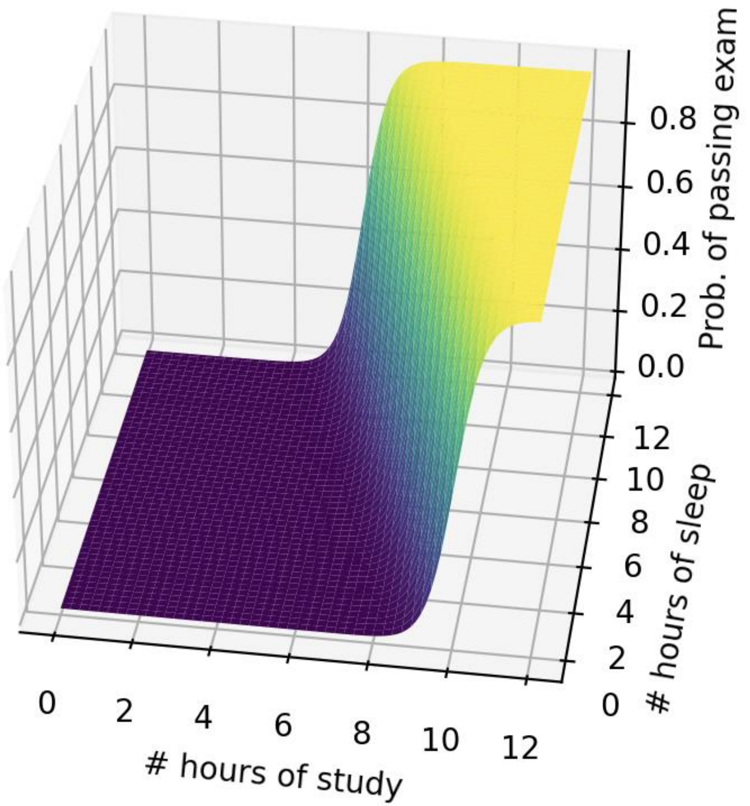
$$J(\theta) = \frac{1}{M}\sum_{m=1}^{M} Loss\left(h_\theta\left(x^{(m)}\right), y^{(m)}\right)$$

What if we have multiple predictors?

- Like linear regression, logistic regression can also process multidimensional inputs.

- Instead of an s-shaped prediction curve, we have a **prediction surface**.

- Example:
  - x_1 = # hours of **study** before an exam
  - x_2 = # hours of **sleep** before an exam
  - y = whether the student **passes** (1) the exam or not (0)



What if there are more than two classes?

- We can generalize this approach to multiple classes, by replacing:
  - linear regression → multivariate regression
  - sigmoid → softmax

Dog
Cat
Horse
Fish
Bird

- (In scikit-learn, the LogisticRegression object takes care of this on its by itself)