Maximum Likelihood Estimation & Logistic Regression

Slides adapted from David Sontag and Andrew Ng.

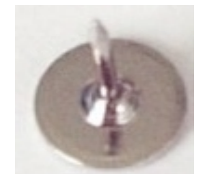
Maximum Likelihood Estimation (MLE)

Framework:

- Observed data D (observations)
- Hypothesize data has a specific probability distribution parameterized by unknown parameter values $\boldsymbol{\theta}$: i.e., distribution $P_{\boldsymbol{\theta}}(D)$ is known
- Goal: estimate (learn) the parameter values $\boldsymbol{\theta}$.
- MLE: Choose parameter values $\boldsymbol{\theta}$ that maximize $P_{\boldsymbol{\theta}}(D)$

Thumbtack example

- P_θ(Heads) = θ, P_θ(Tails) = 1-θ. We want to estimate θ from observational data.

    … 

- Make observations:

  $D = \{y_i \mid i = 1,\ldots,m\}$ $\quad$ y_i = H or T
- Need a model $P_\theta(D)$
- Let α_H be number of heads in D; α_H be number of tails in D.
- Natural model (Why?)

$$P_\theta(D) = \theta^{\alpha_H}(1-\theta)^{\alpha_T}$$

Called the "likelihood" of the data under the model.

Maximum Likelihood Estimation

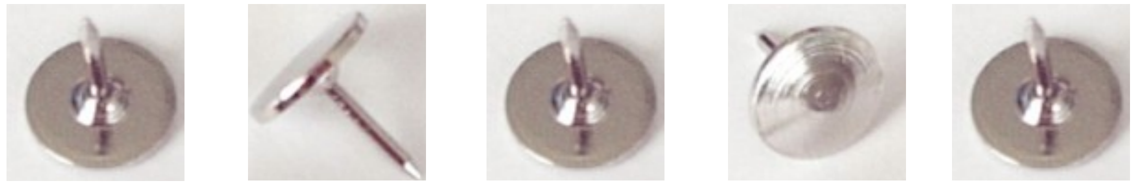
- **Data:** Observed set D: sequence of heads and tails, with α_H Heads and α_T Tails.
- **Model:** $P_{\boldsymbol{\theta}}(D) = \theta^{\alpha_H}(1-\theta)^{\alpha_T}$
- **Learning:** find θ that maximizes the probability of the observation *D*, i.e., find:

$$\widehat{\theta} = \arg\max_{\theta} P_{\boldsymbol{\theta}}(D)$$

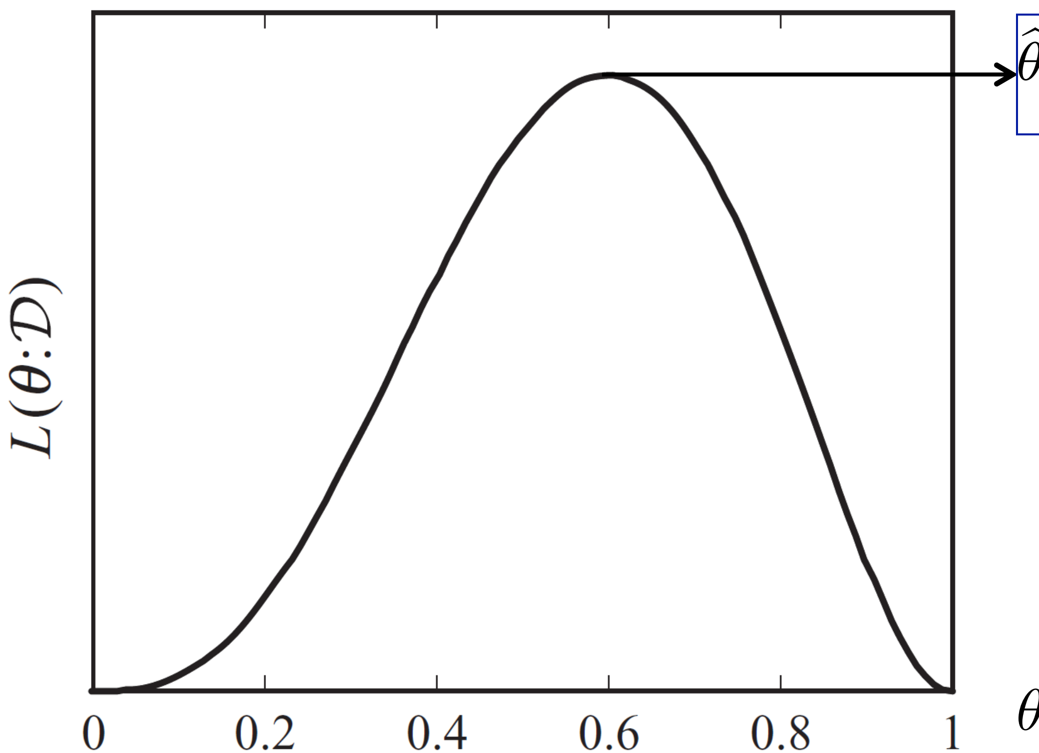
- **Taking derivative and setting to zero, get:**

$$\boxed{\widehat{\theta}_{MLE} = \frac{\alpha_H}{\alpha_H + \alpha_T}}$$

Data



$$L(\boldsymbol{\theta};D) = P_{\boldsymbol{\theta}}(D)$$



$$\widehat{\theta}_{MLE} = \frac{\alpha_H}{\alpha_H + \alpha_T}$$

Logistic Regression

- Popular type of supervised machine learning for classification
- Classification, not regression!
- Gives probabilities for classification, e.g., email is spam with probability 0.86
- Can be viewed as a MLE estimator
- Often used in neural networks

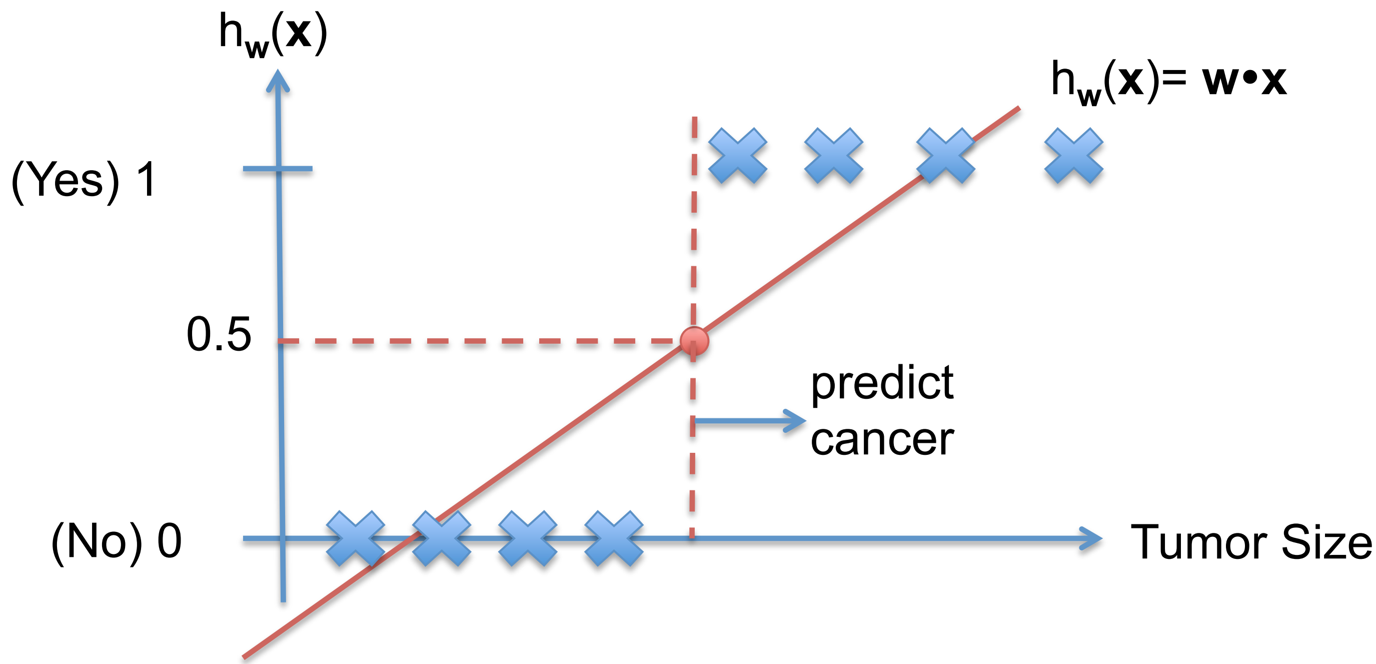
Example:

- Tumor: Malignant/ Benign?

y in {0,1}

0: “Negative Class” (e.g., benign tumor)
1: “Positive Class” (e.g., malignant tumor)

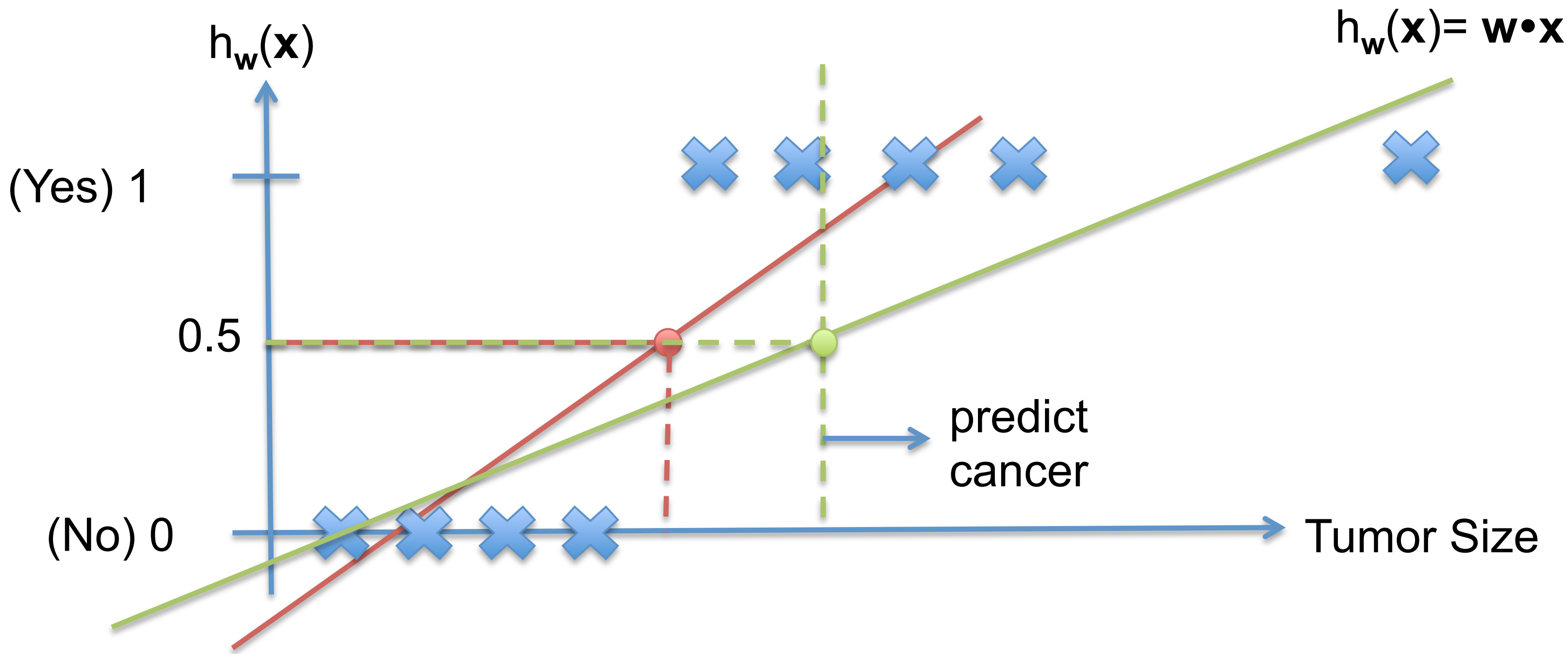
Let’s try to predict with ordinary regression



Natural threshold classifier:

- If $h_w(\mathbf{x}) \geq 0.5$, predict “y=1”
- If $h_w(\mathbf{x}) < 0.5$, predict “y=0”

Additional data point



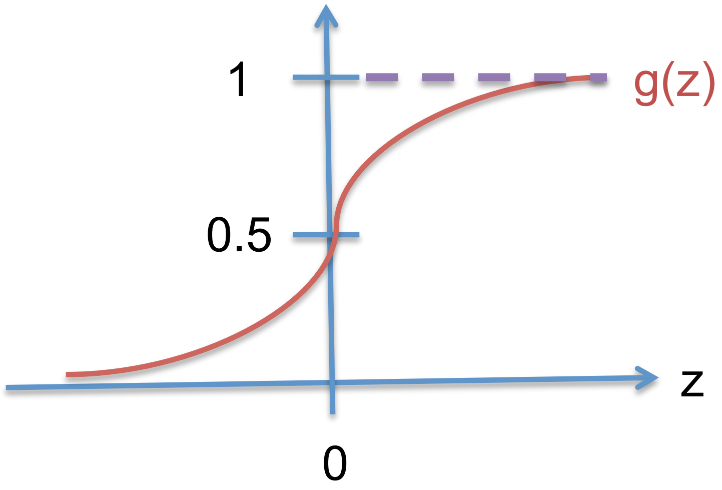
Linear regression with natural 0.5 threshold does not look good here.

Graphically, what kind of function would be a good fit?

Sigmoid function

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

Sigmoid function = Logistic function

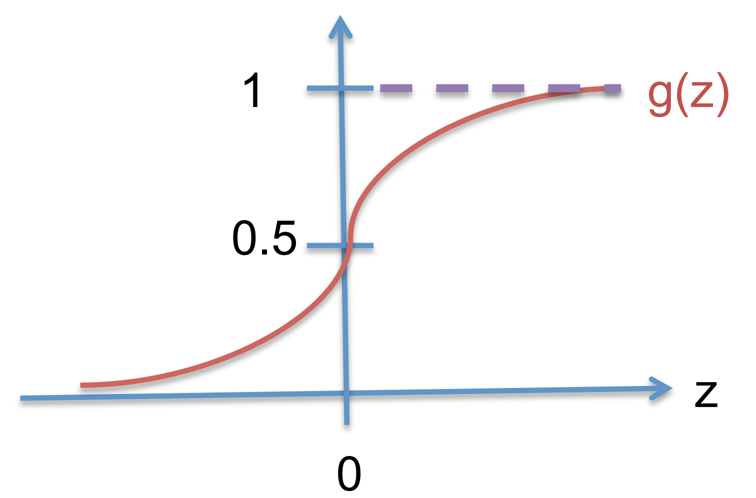


Logistic Regression Model

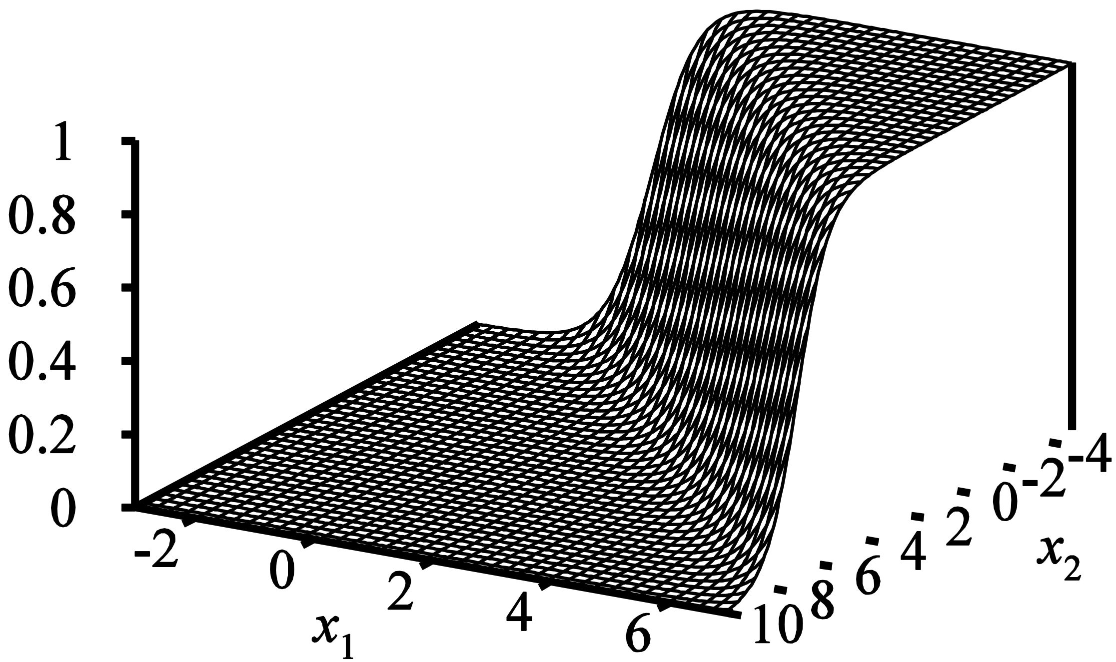
$$g(z) = \frac{1}{1 + e^{-z}}$$

$$h_{\boldsymbol{w}}(\boldsymbol{x}) = g(\boldsymbol{w} \cdot \boldsymbol{x})$$

- Note that $0 \le h_{\mathbf{w}}(\mathbf{x}) \le 1$
- Predict y =1 if $h_{\mathbf{w}}(\mathbf{x}) > ½$; otherwise predict y=0.
- $h_{\mathbf{w}}(\mathbf{x})$ can be interpreted as a probability, eg., probability of cancer
- Can choose **w** to optimize the fit to data (later).



Logistic Function in n Dimensions



Tumor example

- Suppose we have learned **w**. Observe **x** for new patient and want to predict if patient has cancer
- $h_{\mathbf{w}}(\mathbf{x}) = g(\mathbf{w} \bullet \mathbf{x})$ estimated probability that patient has cancer
- Example: $x = (x_0, x_1) = (1, \text{tumorSize})$
- Suppose $h_{\mathbf{w}}(\mathbf{x}) = 0.7$.
- Can tell patient 70% chance tumor is malignant
- That is, probability model $P_{\mathbf{w}}(y=1|\mathbf{x}) = h_{\mathbf{w}}(\mathbf{x})$

Summary

- Use labeled data to learn **w**
- Observe new **x**
- Given **x**, we say y=1 with estimated probability $h_{\mathbf{w}}(\mathbf{x}) = g(\mathbf{w} \bullet \mathbf{x})$, where

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

- Alternatively way of saying it: $P_{\mathbf{w}}(y=1|\mathbf{x}) = h_{\mathbf{w}}(\mathbf{x})$.
- But how do we learn **w** ?

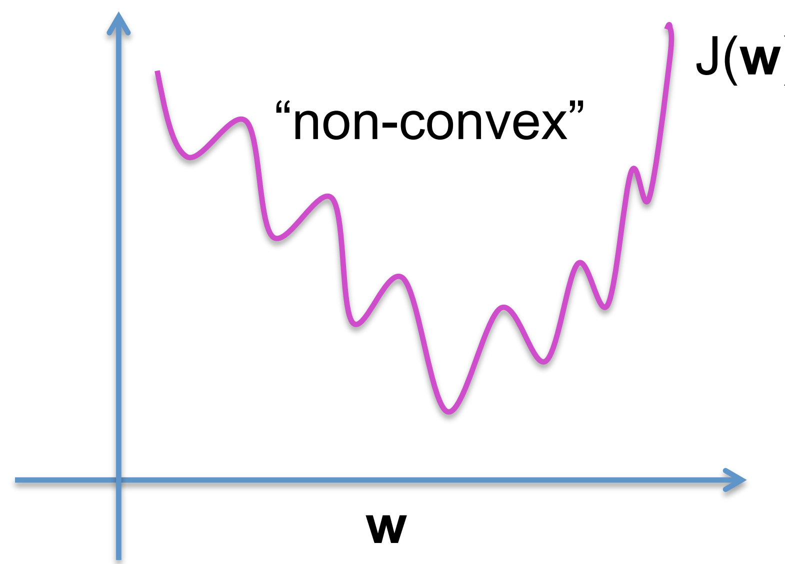
Learning the Parameters **w**

- Training set: { ($\mathbf{x}^{(1)}$, $y^{(1)}$), ($\mathbf{x}^{(2)}$, $y^{(2)}$),…, ($\mathbf{x}^{(m)}$,$y^{(m)}$)}
- How about choosing w to minimize MSE (as usual)?

$$J(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_{\boldsymbol{w}}(\boldsymbol{x}^{(i)}), y^{(i)})$$

$$Cost(h_{\boldsymbol{w}}(\boldsymbol{x}), y) = \frac{1}{2}(h_{\boldsymbol{w}}(\boldsymbol{x}) - \mathrm{y})^2$$

$$h_{\boldsymbol{w}}(x) = \frac{1}{1 + e^{-\boldsymbol{w}\cdot\boldsymbol{x}}}$$



Learning the Parameters **w**

- Instead try using MLE. Find **w** that maximizes the probability of the observation. Maximize:

  $$P_{\mathbf{w}}(y^{(1)},y^{(2)},\ldots,y^{(m)} \mid \mathbf{x}^{(1)}, \mathbf{x}^{(2)},\ldots, \mathbf{x}^{(m)})$$

- Need model for $P_{\mathbf{w}}(y^{(1)},y^{(2)},\ldots,y^{(m)} \mid \mathbf{x}^{(1)}, \mathbf{x}^{(2)},\ldots, \mathbf{x}^{(m)})$:
  - Assume each observed data point is conditionally independent:

    $P_{\mathbf{w}}(y^{(1)},y^{(2)},\ldots,y^{(m)} \mid \mathbf{x}^{(1)}, \mathbf{x}^{(2)},\ldots, \mathbf{x}^{(m)}) =$
    $P_{\mathbf{w}}(y^{(1)} \mid \mathbf{x}^{(1)}) \times P_{\mathbf{w}}(y^{(2)} \mid \mathbf{x}^{(2)}) \times \ldots \times P_{\mathbf{w}}(y^{(m)} \mid \mathbf{x}^{(m)})$

  - Assume logistic function probabilities:

    $P_{\mathbf{w}}(y^{(i)}=1 \mid \mathbf{x}^{(i)}) = h_{\mathbf{w}}(\mathbf{x}^{(i)})$
    $P_{\mathbf{w}}(y^{(i)}=0 \mid \mathbf{x}^{(i)}) = 1- h_{\mathbf{w}}(\mathbf{x}^{(i)})$

    $$h_{\boldsymbol{w}}(\boldsymbol{x}) = \frac{1}{1+e^{-\boldsymbol{w}\cdot\boldsymbol{x}}}$$

Choose **w** to maximize:

- $P_{\mathbf{w}}(y^{(1)},y^{(2)},\ldots,y^{(m)} \mid \mathbf{x}^{(1)}, \mathbf{x}^{(2)},\ldots, \mathbf{x}^{(m)})$
- Because log is an increasing function, we can instead maximize
  $\log (P_{\mathbf{w}}(y^{(1)},y^{(2)},\ldots,y^{(m)} \mid \mathbf{x}^{(1)}, \mathbf{x}^{(2)},\ldots, \mathbf{x}^{(m)})) =$
  $\log [P_{\mathbf{w}}(y^{(1)} \mid \mathbf{x}^{(1)})\times P_{\mathbf{w}}(y^{(2)} \mid \mathbf{x}^{(2)})\times\ldots\times P_{\mathbf{w}}(y^{(m)} \mid \mathbf{x}^{(m)})] =$
  $\log P_{\mathbf{w}}(y^{(1)} \mid \mathbf{x}^{(1)}) + \log P_{\mathbf{w}}(y^{(2)} \mid \mathbf{x}^{(2)})+\ldots+ \log P_{\mathbf{w}}(y^{(m)} \mid \mathbf{x}^{(m)})$
- $\log P_{\mathbf{w}}(y^{(i)} = 1 \mid \mathbf{x}^{(i)}) = \log h_{\mathbf{w}}(\mathbf{x}^{(i)})$
  $\log P_{\mathbf{w}}(y^{(i)}= 0 \mid \mathbf{x}^{(i)}) = \log (1- h_{\mathbf{w}}(\mathbf{x}^{(i)}))$
- So $P_{\mathbf{w}}(y^{(i)} \mid \mathbf{x}^{(i)}) = y^{(i)} \log h_{\mathbf{w}}(\mathbf{x}^{(i)}) + (1- y^{(i)}) \log (1- h_{\mathbf{w}}(\mathbf{x}^{(i)}))$

$$J(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_w(\boldsymbol{x}^{(i)}), y^{(i)})$$
$$= -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_w(\boldsymbol{x}^{(i)}) + (1-y^{(i)}) \log\left(1-h_w(\boldsymbol{x}^{(i)})\right)\right]$$

Convex function!

Summary

- Want to find a **w** so that the logistic regression fits the data.
- Usual MSE error cost function leads to non-convex optimization problem.
- Instead consider finding **w** that maximizes likelihood (MLE). This is equivalent to finding **w** that minimizes:

$$J(\boldsymbol{w}) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_{\boldsymbol{w}}\left(\boldsymbol{x}^{(i)}\right) + (1 - y^{(i)}) \log\left(1 - h_{\boldsymbol{w}}\left(\boldsymbol{x}^{(i)}\right)\right)\right]$$

- Convex optimization problem

Gradient Descent

$$J(\boldsymbol{w}) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_{\boldsymbol{w}}\left(\boldsymbol{x}^{(i)}\right) + (1 - y^{(i)}) \log\left(1 - h_{\boldsymbol{w}}\left(\boldsymbol{x}^{(i)}\right)\right)\right]$$

Want $\min_{\mathbf{w}} J(\mathbf{w})$:

Repeat {

$$w_j := w_j - \alpha \frac{\partial}{\partial w_j} J(\boldsymbol{w})$$

} (simultaneously update all $\mathbf{w}_j$)

$$\frac{\partial}{\partial w_j} J(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m}(h_{\boldsymbol{w}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)})x_j^{(i)}$$

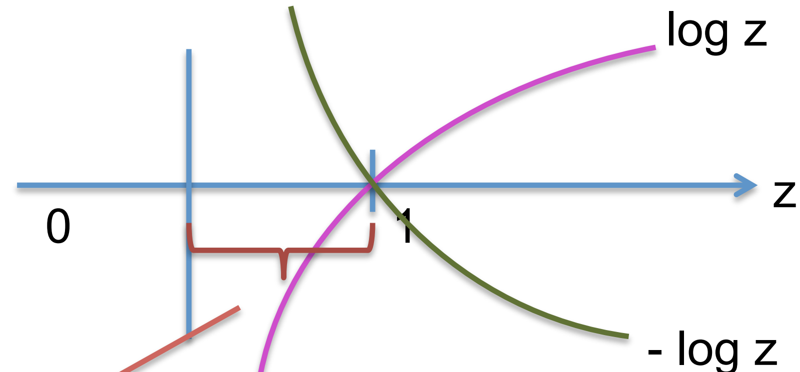
$$w_j := w_j - \alpha \sum_{i=1}^{m}(h_{\boldsymbol{w}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)})x_j^{(i)}$$

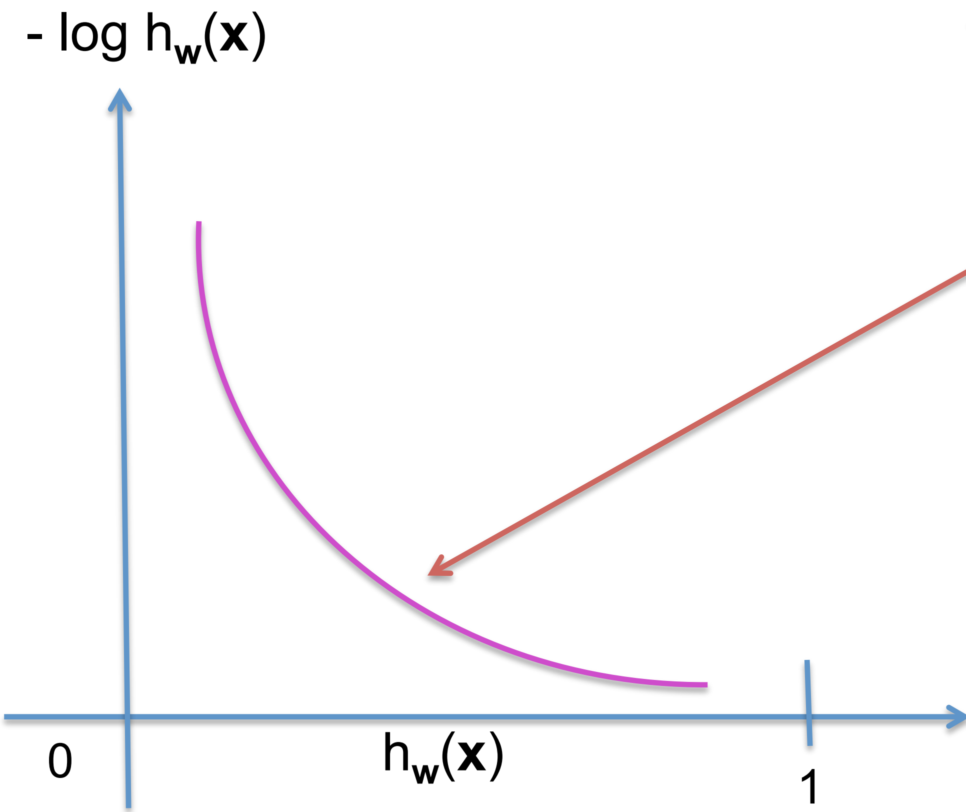
Algorithm looks identical to linear regression! But it is not!

Some intuition into cost function

$$Cost(h_{\boldsymbol{w}}(\boldsymbol{x}), y) = \begin{cases} -\log(h_{\boldsymbol{w}}(\boldsymbol{x})) \; if \; y = 1 \\ -\log(1 - h_{\boldsymbol{w}}(\boldsymbol{x})) \; if \; y = 0 \end{cases}$$

Consider y=1





As $h_w(\mathbf{x}) \to 1$, Cost $\to 0$
But as $h_w(\mathbf{x}) \to 0$, Cost $\to \infty$
Captures intuition that if $h_w(\mathbf{x}) = 0$, (predict $P_w(y=1 | \mathbf{x})=0$), we'll penalize learning algorithm by a very large cost.

Summary: Logistic regression cost function

$$J(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_{\boldsymbol{w}}\left(\boldsymbol{x}^{(i)}\right), y^{(i)})$$

$$= -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_{\boldsymbol{w}}\left(\boldsymbol{x}^{(i)}\right) + (1 - y^{(i)}) \log\left(1 - h_{\boldsymbol{w}}\left(\boldsymbol{x}^{(i)}\right)\right)\right]$$

To fit parameters **w**:

$$\min_{\boldsymbol{w}} J(\boldsymbol{w})$$

To make a prediction given new **x**:

Output $$h_{\boldsymbol{w}}(x) = \frac{1}{1 + e^{-\boldsymbol{w}\cdot\boldsymbol{x}}}$$