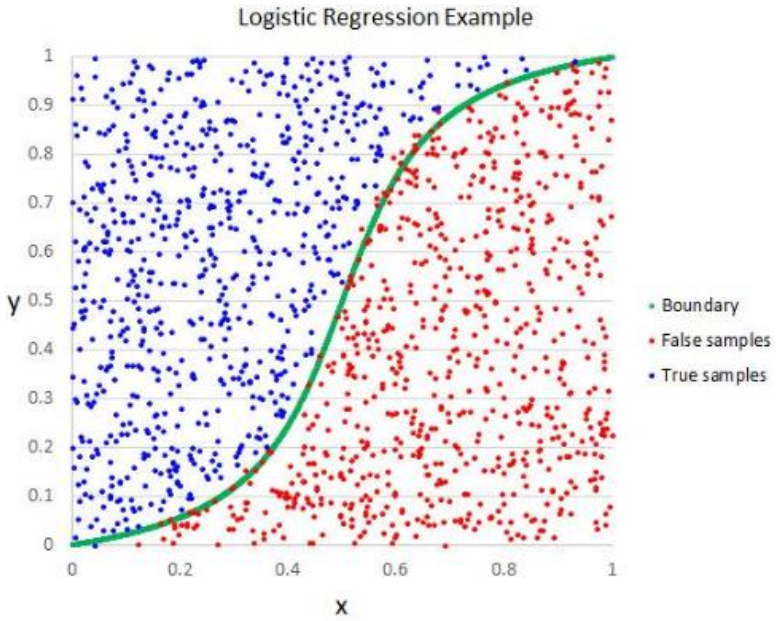
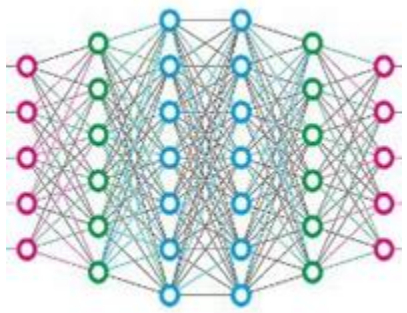
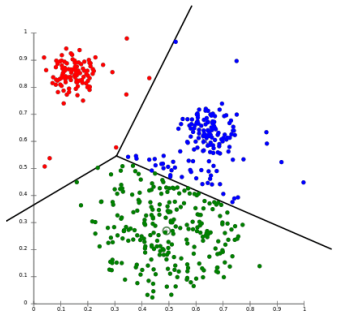
CS 445
Introduction to Machine Learning

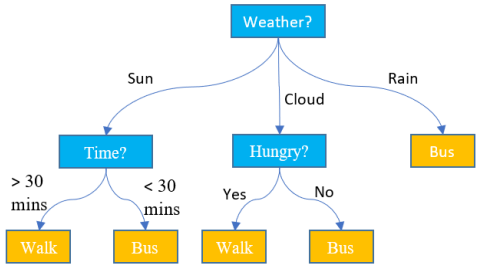
Logistic Regression



Dr. Kevin Molloy – Fall 2025







Review

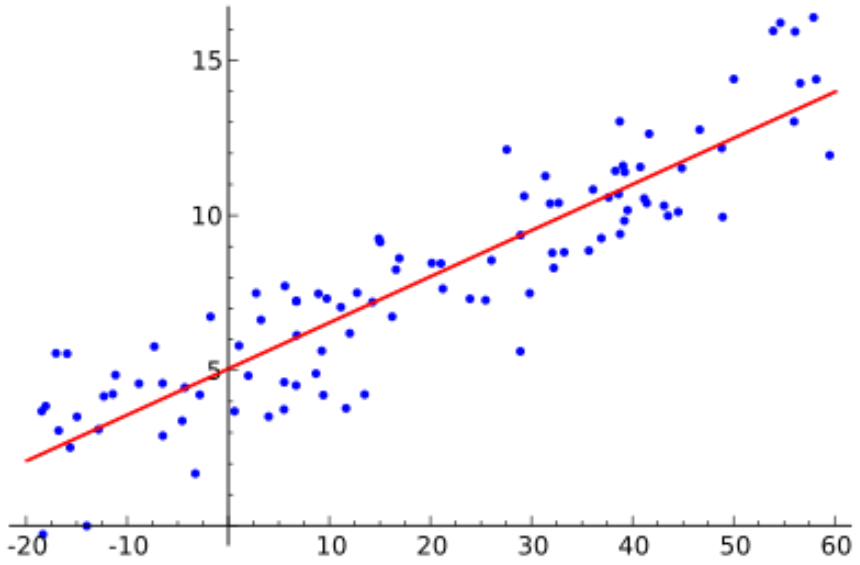
Linear regression

Finding the weights to assign to a polynomial so

that the resulting line minimizes the "loss".

$h(x_1, x_2, \ldots x_n) = w_0 + w_1 x_1 + \ldots + w_n x_n$

$h(x) = w^T x$

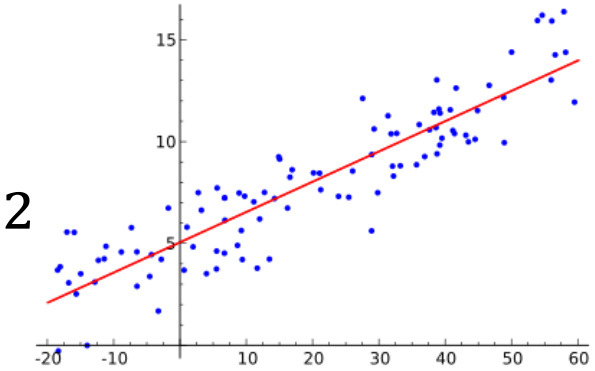


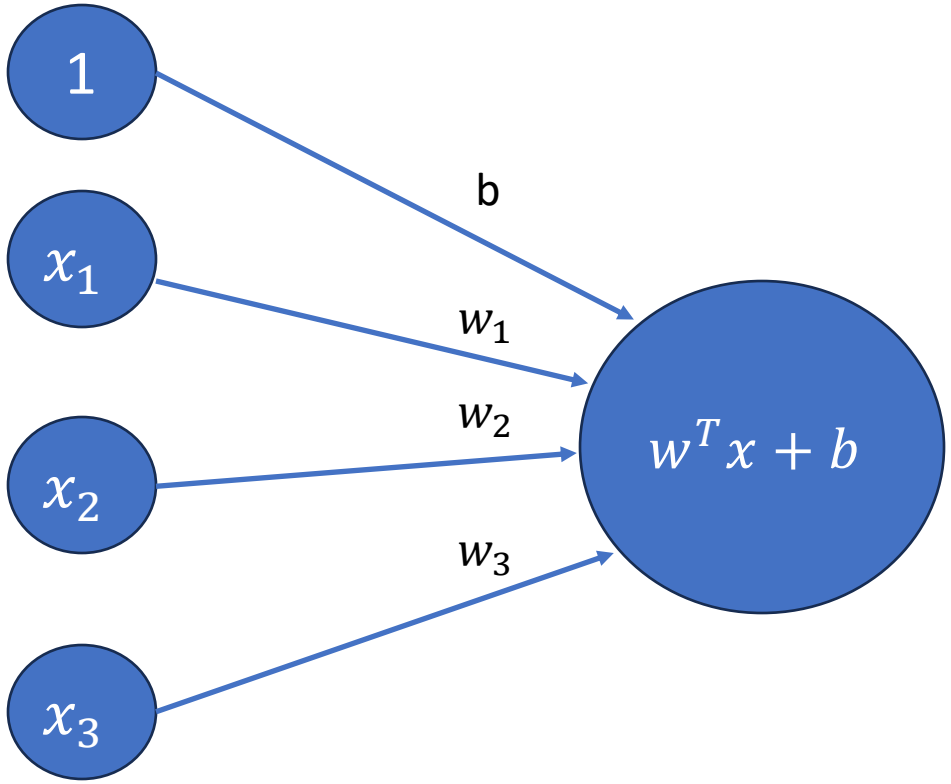
This function h(x) (hypothesis function) makes a real valued prediction (regression).

Linear Regression Loss $L(w) = \frac{1}{2}\sum_{(x_i,y_i)\in D}(y_i - w^t x_i)^2$

Approach for Linear Regression

Linear Regression Loss $L(w) = \frac{1}{2N}\sum_{(x_i,y_i)\in D}(y_i - w^t x_i)^2$



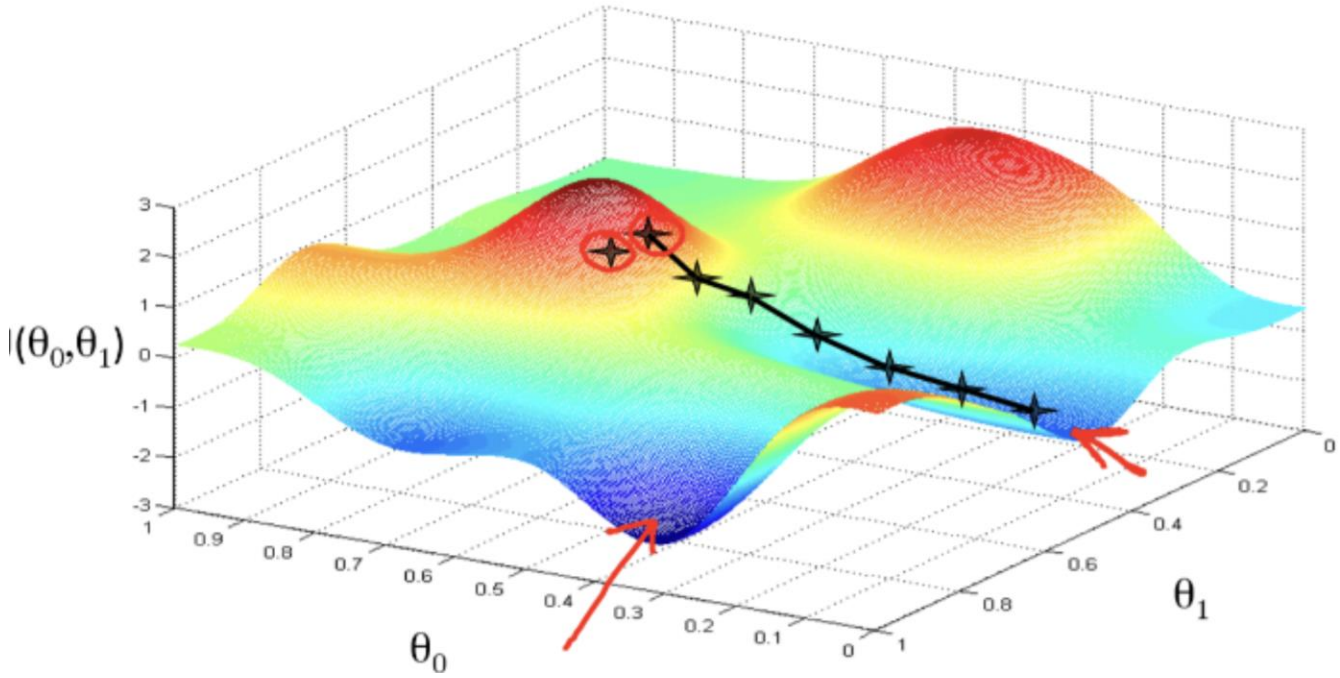


Optimize (find the min) of the loss function using the derivatives:

$$\frac{\partial \mathrm{L(w)}}{\partial \mathrm{w_i}} = \frac{1}{\mathrm{N}} \sum_{\mathrm{j=0..N}} x_j^{(i)} (\mathrm{y}_j - \mathrm{w^t x}_j)$$

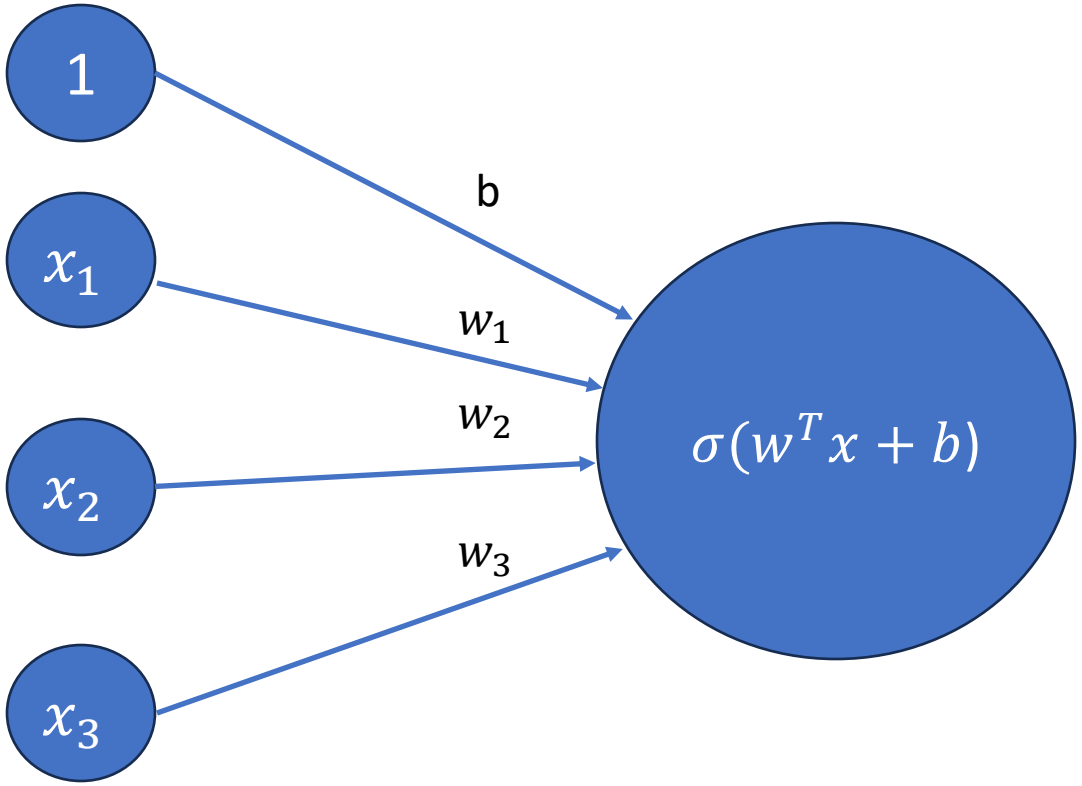
Linear Regression Algorithm

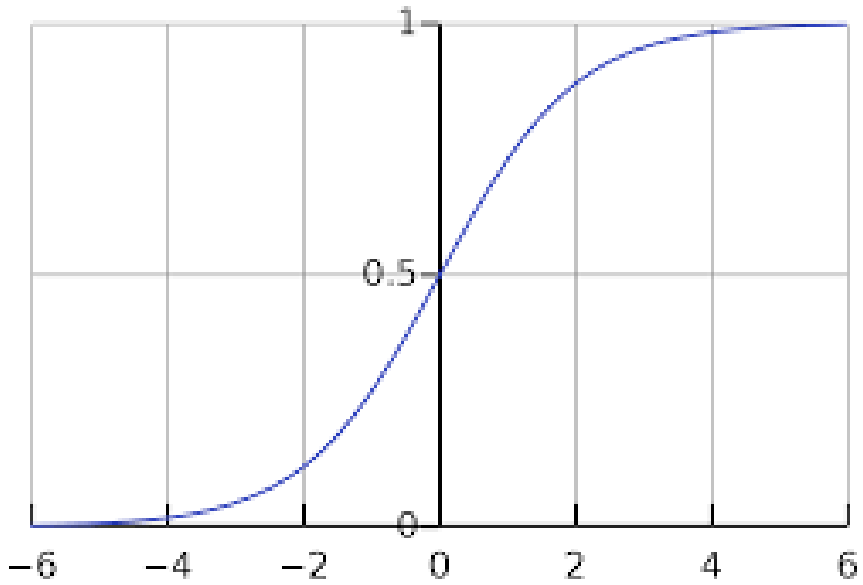
1. Make predictions using current w and compute loss
2. Compute derivative and update w's
3. When loss change between this and the last iteration is small, STOP. Otherwise, go back to 1.



Map Function to Values Between 0 and 1

Sigmoid (Logistic) (z) = $\sigma(z) = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-w^t x}}$



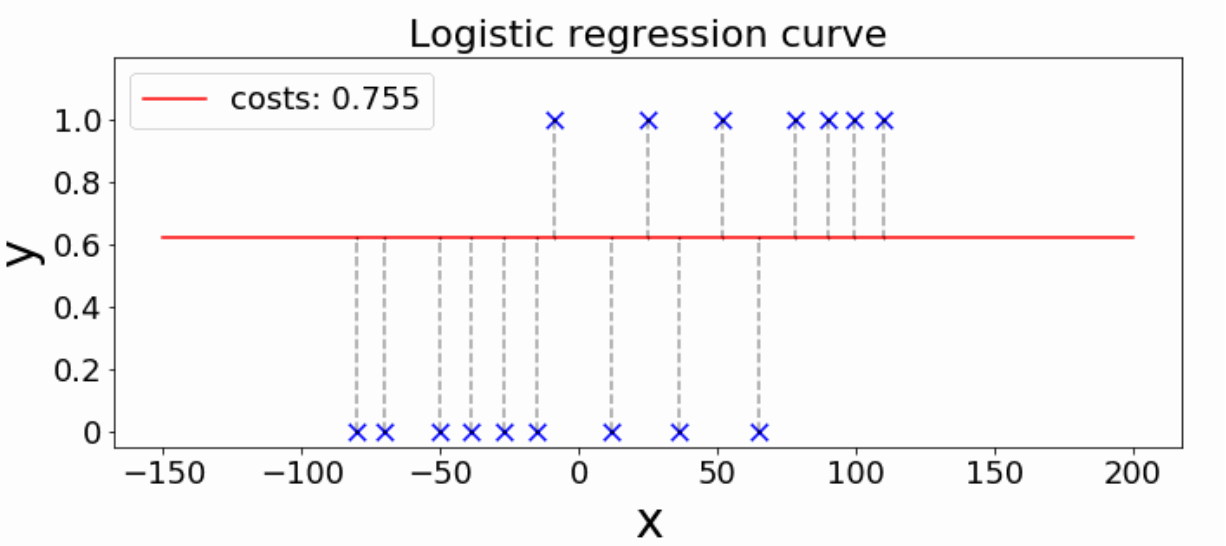


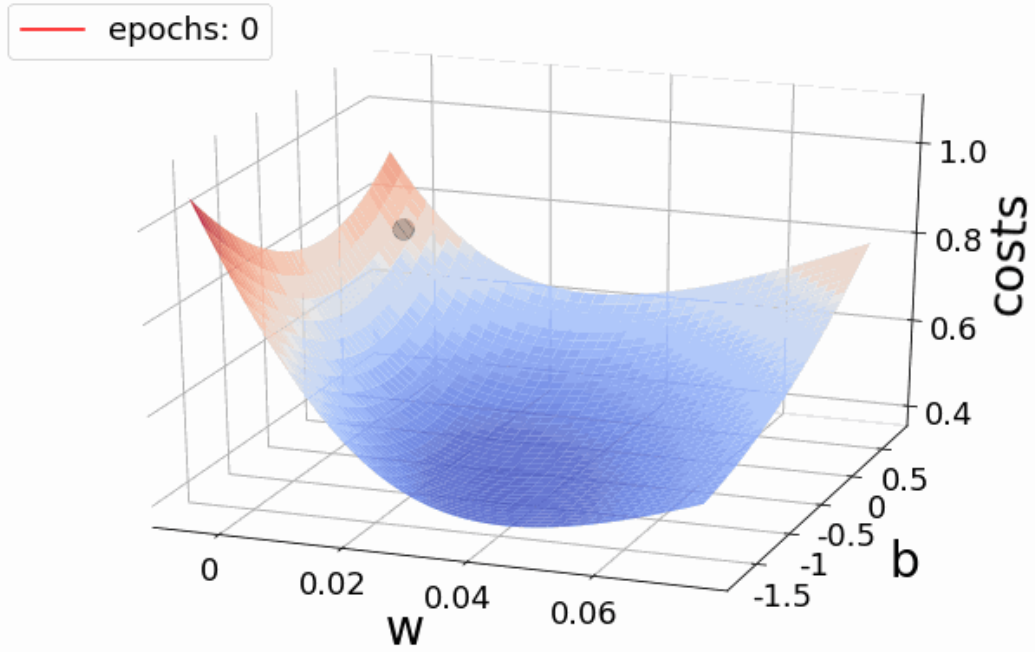
Note: When $w^t x + b > 0$, predict class 1 (that is what the odds tell us).

The plot shows $\sigma(z) > 0.5$ only when $z \geq 0$.

Question: How to learn the best w values so that the odds (posterior probabilities) reflect any training data x.

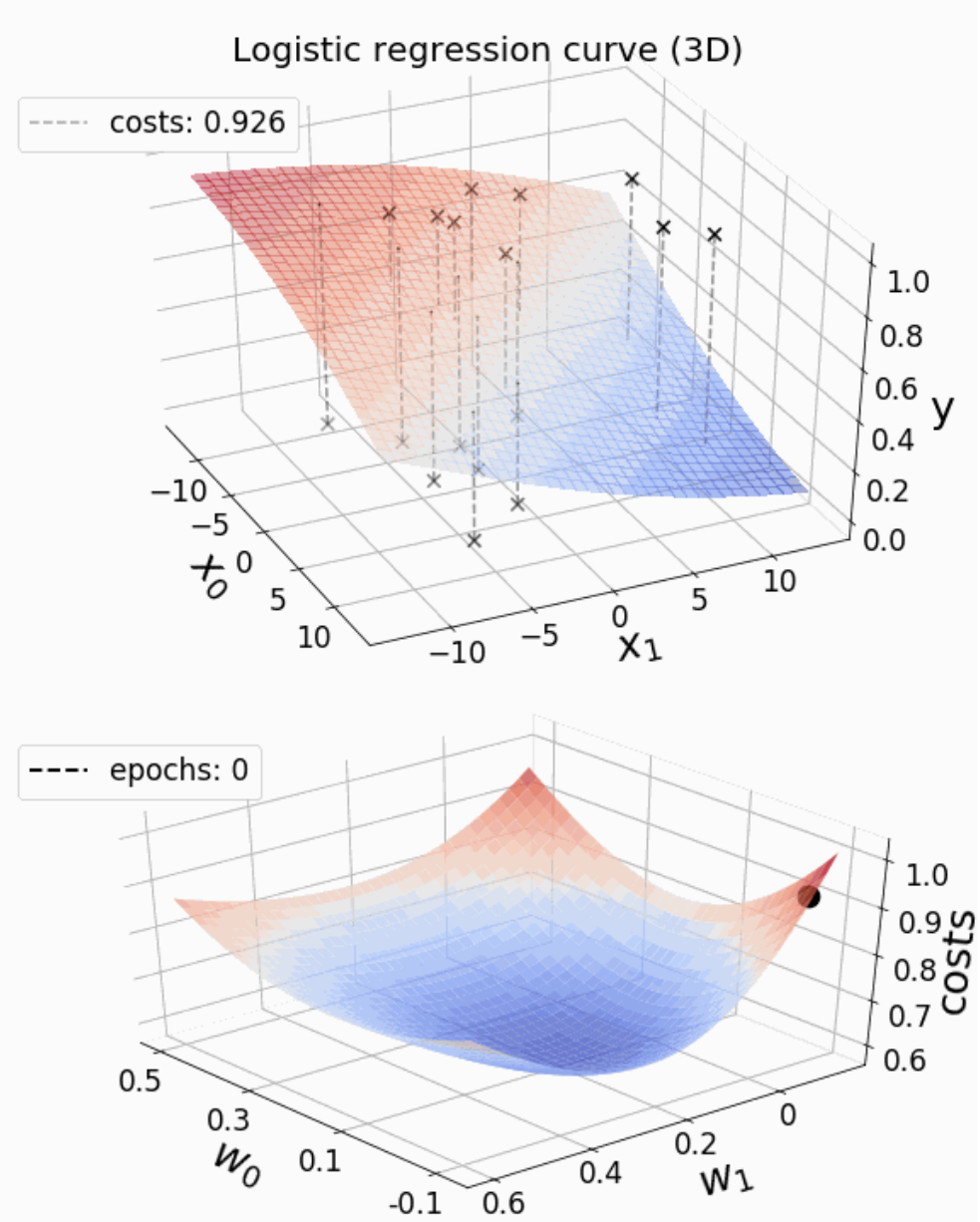
Logistic Regression Mechanics/Math





Citation to Tobias Roeschi

Logistic Regression Mechanics/Math



Citation to Tobias Roeschi

How about using this for Classification?

For binary classification, let's define the **odds** as $\frac{P(y=1\,|x)}{P(y=0\,|x)}$

When this is greater than 1, we predict class 1, otherwise we predict class 0.

Idea: Use linear regression model to compute the odds.

Note that $P(y=0|x)+P(y=1\,|x)\;=1$ Therefore: $P(y=0|x)=1-P(y=1|x)$

- $\mathrm{P(y=1|x,w)}=\;\sigma(w^t x+b)=\frac{1}{1+e^{-(w^T x+b)}}$
- $\mathrm{P(y=0|x,w)}=1\;-\sigma(w^t x+b)$

Maximum Likelihood

$$\theta_{ML} = argmax_{\theta} P(D\,|\theta) \qquad \mathcal{L}\,(\theta) = \mathrm{P}(\mathrm{D}\,|\theta) = \prod_{x_i \in D} P(x_i|\theta)$$

Said differently, what is the likelihood of the data GIVEN the parameters of the model.

For example, take a bias coin that has $\theta = 0.9$ to come up heads. What is the likelihood of seeing this data: [Heads, Tails Heads]?

Maximum Likelihood

$$\theta_{ML} = argmax_{\theta} P(D\,|\theta) \qquad \mathcal{L}\,(\theta) = \mathrm{P}(\mathrm{D}\,|\theta) = \prod_{x_i \in D} P(x_i|\theta)$$

Said differently, what is the likelihood of the data GIVEN the parameters of the model.

For example, take a bias coin that has $\theta = 0.9$ to come up heads. What is the likelihood of seeing this data: [Heads, Tails Heads]?

What is the θ that yields the maximum likelihood?

Maximum Likelihood

$$\theta_{ML} = argmax_{\theta} P(D\,|\theta) \qquad \mathcal{L}\,(\theta) = \mathrm{P}(\mathrm{D}\,|\theta) = \prod_{x_i \in D} P(x_i|\theta)$$

Said differently, what is the likelihood of the data GIVEN the parameters of the model.

Notice no priors.

Introducing Logistic Regression

World's **WORST** algorithm name (it is classification, not regression)

Likelihood function: $\mathcal{L}(w) = \prod_{i=1}^{n} \begin{cases} P(y=1|\boldsymbol{x_i}, \boldsymbol{w}), if\ y_i = 1 \\ P(y=0|\boldsymbol{x_i}, \boldsymbol{w})\ if\ y_i = 0 \end{cases}$

Nicer: $\mathcal{L}(w) = \prod_{i=1}^{n} P(y=1|\boldsymbol{x_i}, \boldsymbol{w})^{y_i} \times P(y=0|x_i, w)^{1-y_i}$

Finally, use logs for numeric stability:

$$\mathcal{LL}(w) = \sum_{i=1}^{n} y_i \log P(y=1|\,\boldsymbol{x_i}, \boldsymbol{w}) + (1-y_i) \log P(y=0\,|\boldsymbol{x_i}, \boldsymbol{w})$$

Maximizing the log likelihood is the same as minimizing the negative log likelihood:

$$-\mathcal{LL}(w) = -\sum_{i=1}^{n} y_i \log P(y=1|\,\boldsymbol{x_i}, \boldsymbol{w}) + (1-y_i) \log P(y=0\,|\boldsymbol{x_i}, \boldsymbol{w})$$

Cross Entropy

This log function is usually called *cross-entropy.*

$$-\mathcal{LL}\,(w) = -\sum_{i=1}^{n} y_i \log(\sigma(w^t x + b)) + (1 - y_i) \log(1 - \sigma(w^t x + b))$$

Quiz: What is the cross-entry loss when: $\sigma(w^t x + b)$ = 1 and

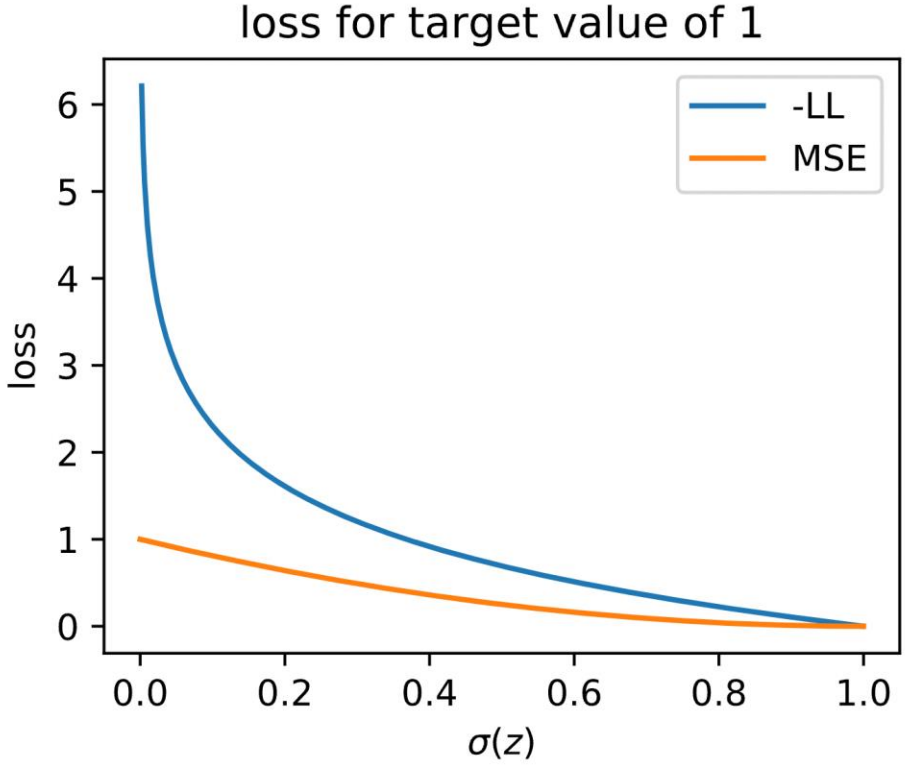
- $y_i = 0$
- $y_i = 1$

Recall that log(1) = 0 and log(0) = $-\infty$

Could we just use SSE and Make Life Easy(Easier)

Zero and one are just numbers, so, why not use the SSE we used for linear regression.

$$E(w) = \sum_{i=1}^{n} \left(y_i - \sigma(w^t x + b)\right)^2$$



(MSE is just mean squared error. Plot from Prof Sprague)

Minimize Cross-Entropy Loss

1. No closed form solution.
2. Could use gradient descent, but specialized solvers exists the use second derivatives (Hessian) to speed things along.
3. Good news! Logistic regression is guaranteed to have a single global minima.

Some nice animations of this are available here:

https://towardsdatascience.com/animations-of-logistic-regression-with-python-31f8c9cb420

Summary

Pros:

- Simple and explainable: individual coefficients indicate the relationship between the attribute and the class.
- Unique global minima means that we can be confident when algorithm converges
- Prediction is fast.
- Provides a real-valued score function that may be interpreted as a probability
- Key machinery for multi-layer neural networks

Cons:

- Linear decision boundary
- Subject to overfitting in high-dimensional settings