Logistic Regression

Classification

- **Learn**: h:**X**->Y
  - **X** – features
  - Y – target classes
- Suppose you know the distribution P(Y|**X**) exactly, how should you classify?
  - Bayes classifier:

$$y^* = h_{bayes}(x) = \arg\max_{y} P(Y = y \mid X = x)$$

- Why?

Generative vs. Discriminative Classifiers - Intuition

- Generative classifier, e.g., Naïve Bayes:
  - Assume some functional form for **P(X|Y), P(Y)**
  - Estimate parameters of P(X|Y), P(Y) directly from training data
  - Use Bayes rule to calculate P(Y|X=x)
  - This is 'generative' model
    - Indirect computation of P(Y|X) through Bayes rule
    - But, can generate a sample of the data, $P(X) = \sum_y P(y)P(X \mid y)$
- Discriminative classifier, e.g., Logistic Regression:
  - Assume some functional form for **P(Y|X)**
  - Estimate parameters of P(Y|X) directly from training data
  - This is the 'discriminative' model
    - Directly learn P(Y|X)
    - But cannot sample data, because P(X) is not available

The Naïve Bayes Classifier

- Given:
  - Prior P(Y)
  - n conditionally independent features X given the class Y
  - For each Xi, we have likelihood P(X_i|Y)
- Decision rule:

$$y^* = h_{NB}(x) = \arg\max_y P(y) P(x_1, \ldots, x_n \mid y)$$

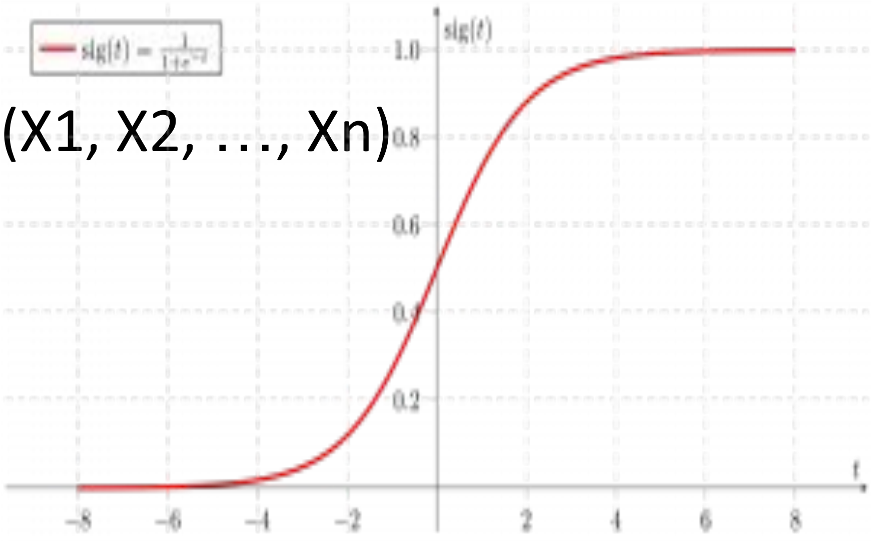
$$= \arg\max_y P(y) \prod_i P(x_i \mid y)$$

- If assumption holds, NB is optimal classifier!

Logistic Regression

- Let X be the data instance, and Y be the class label (0/1).

- Learn P(Y|X) directly
  - Let W = (W1, W2, … Wn), X=(X1, X2, …, Xn)
  - **WX** is the dot product
  - Sigmoid function:



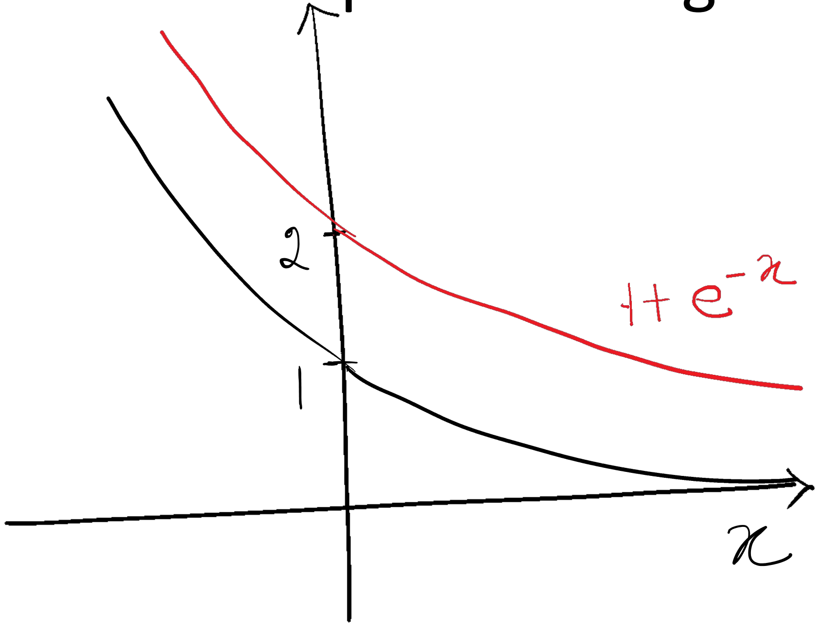
$$P(Y=1\,|\,\mathbf{X}) = \frac{1}{1+e^{-\mathbf{wx}}}$$

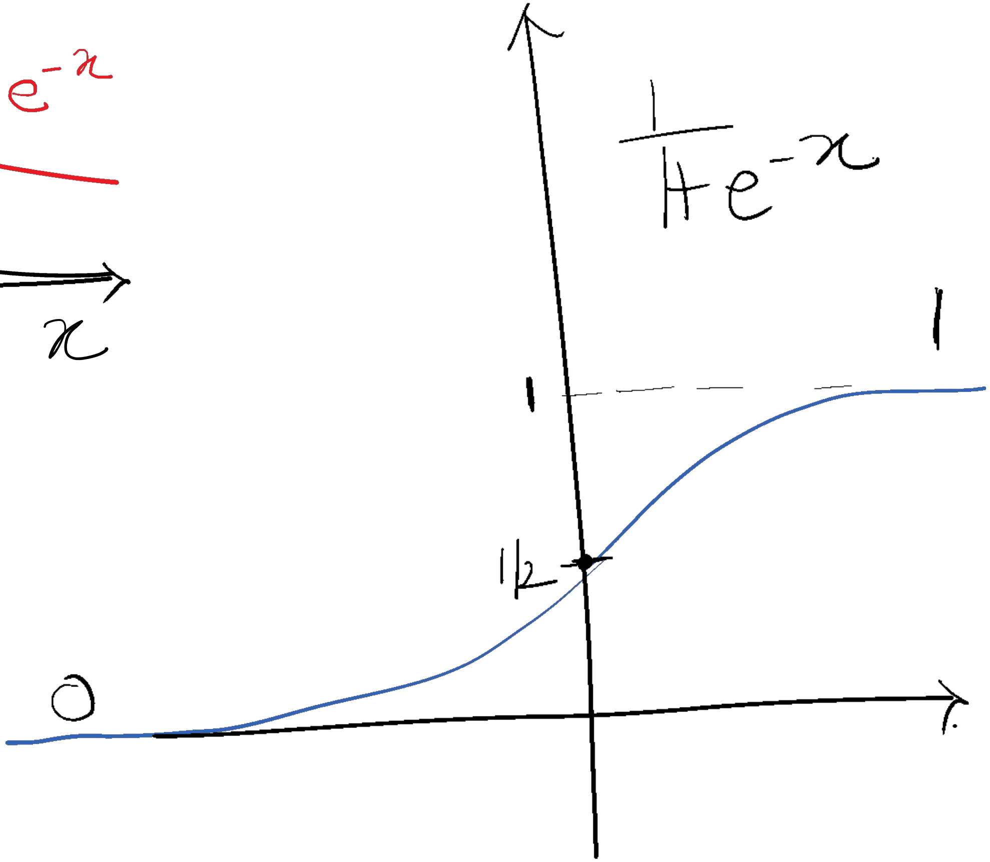
Regression or Classification

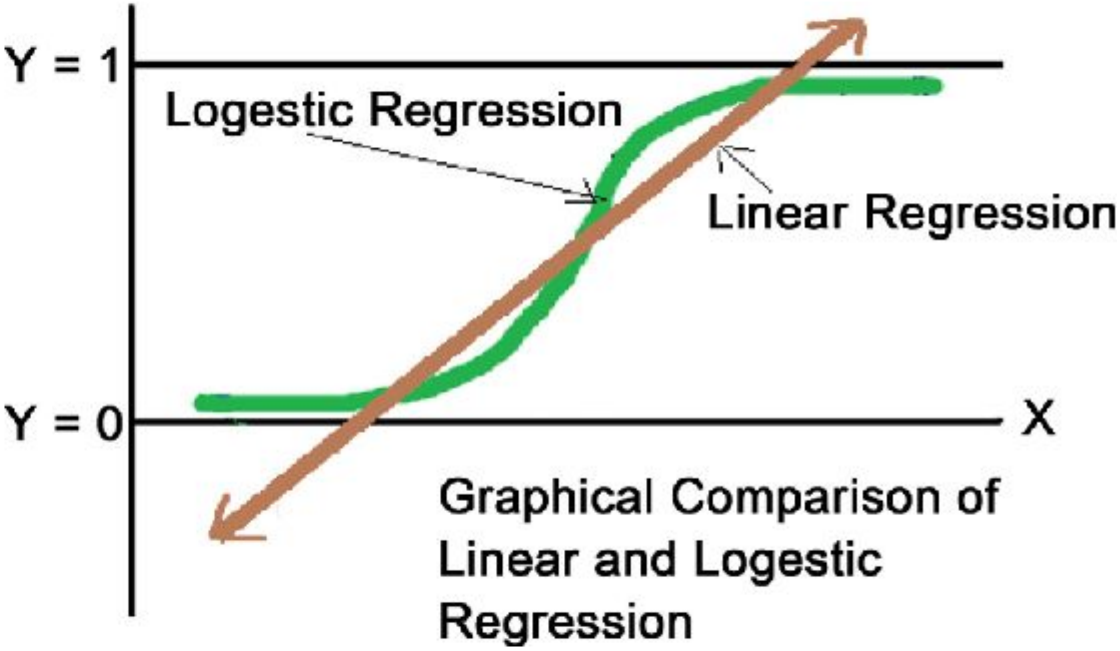
- Gives probability of a class (win/loss). Continuous output □ Regression

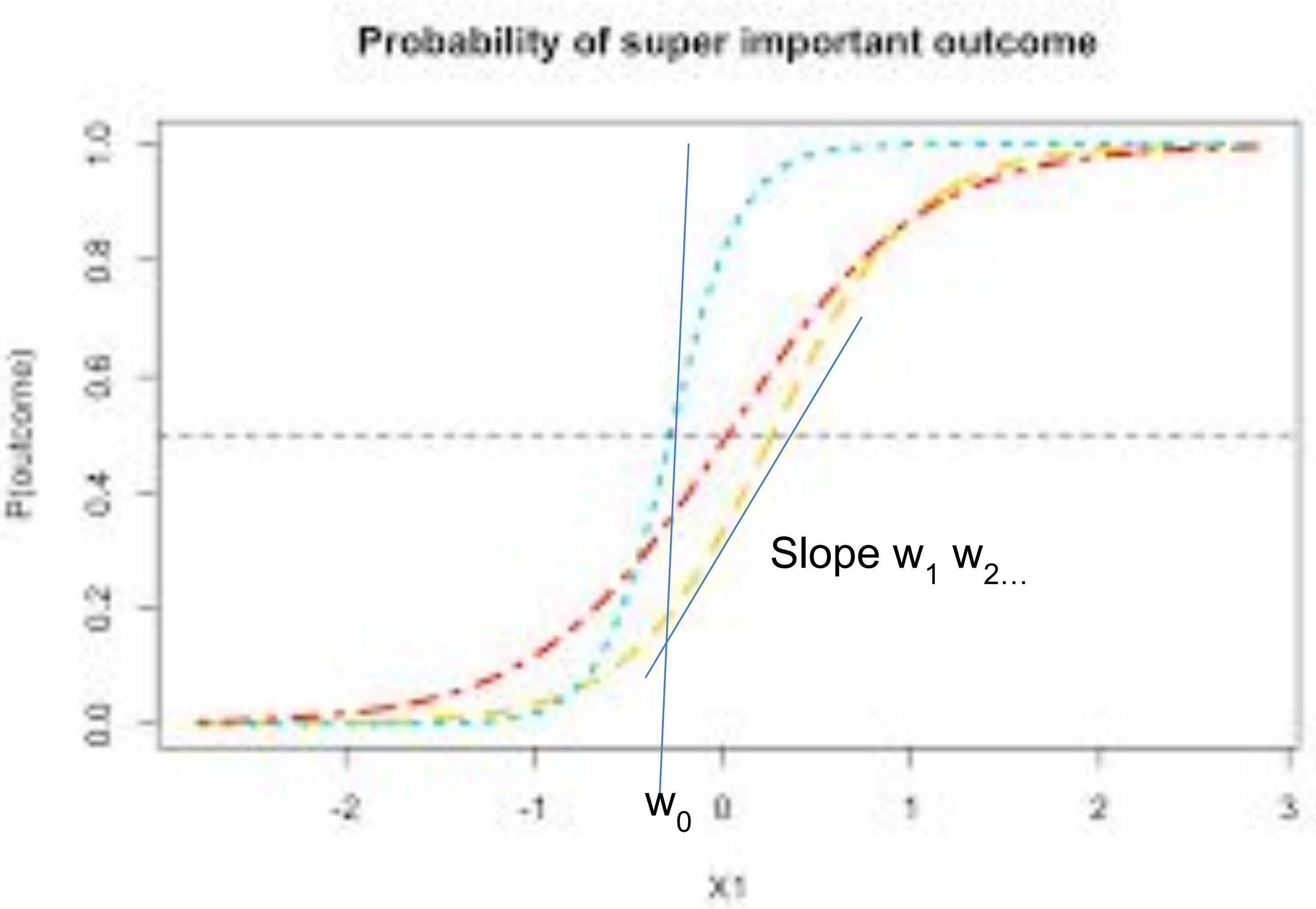
- Decide a threshold to decide outcome, becomes classification

Development of sigmoid as soft switch





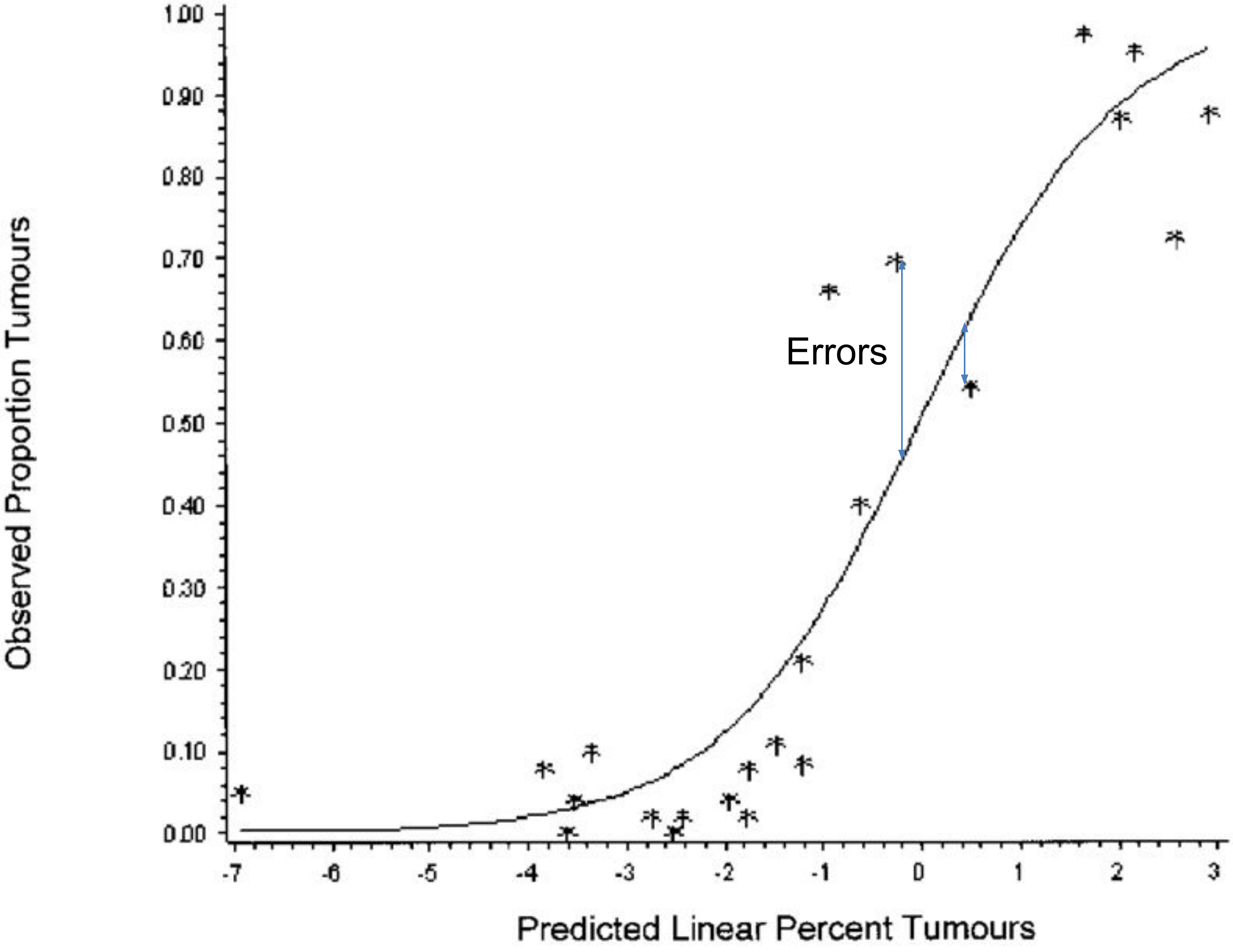

Y = 1
Logestic Regression
Linear Regression
Y = 0
X
Graphical Comparison of Linear and Logestic Regression



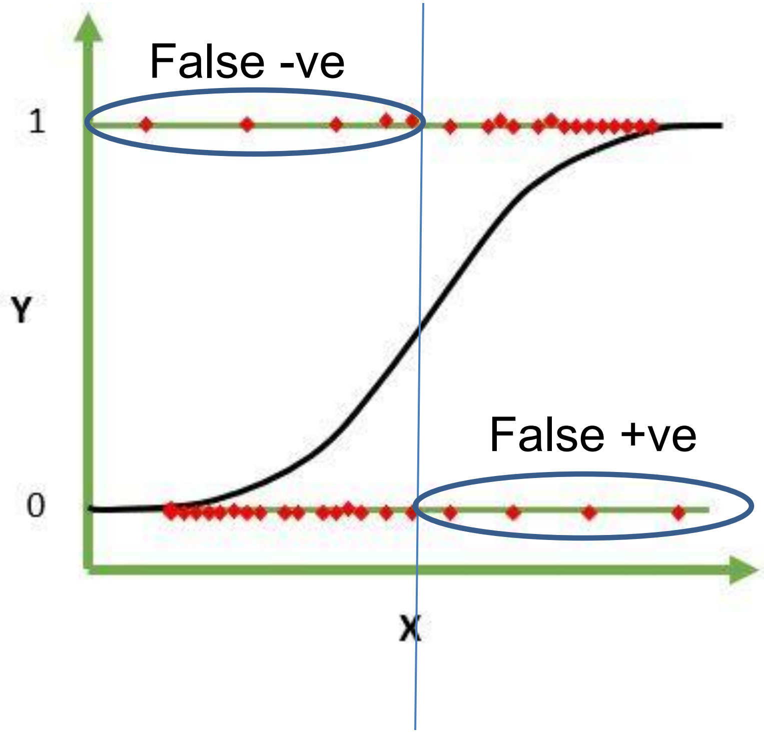
Slope w_1 $w_{2...}$

w_0

The Predicted Versus the Observed Proportion of Tumours (Fiber Number Injected, Median Fiber Length and IT-WT$_{1/2}$ L>20 μm)



Linear Predictor = Intercept + b1 * Length + b2 * Ln(Fib No) + b3 * T $_{1/2}$


False -ve
1
Y
False +ve
0
X

Logistic Regression

- In logistic regression, we learn the conditional distribution P(y|x)

- Let p_y(x;w) be our estimate of P(y|x), where w is a vector of adjustable parameters.

$$p\ (\mathbf{x};\mathbf{w}) = \frac{1}{1+e^{-\mathbf{w}\mathbf{x}}}$$

Log odds

- Assume there are two classes, y = 0 and y = 1 and

$$p_1(\mathbf{x};\mathbf{w}) = \frac{1}{1+e^{-\mathbf{wx}}} \qquad p_0(\mathbf{x};\mathbf{w}) = 1 - \frac{1}{1+e^{-\mathbf{wx}}}$$

- This is equivalent to $\mathrm{Log}_b \frac{p_1(\mathbf{x};\mathbf{w})}{p_0(\mathbf{x};\mathbf{w})} = \mathbf{wx}$

- That is, the **log odds of class 1 w.r.t class 2,** is a linear function of x

Log Odds, odds and Probability

With all x=0, When w_0=-2

- what is the log-odds of P_1 (Say Y?)
- What is the odds of P_1?
- What is the probability of P_1?
- Calculate for -3? 1? 0? 3?

With w_1= 1, x_1 increases by 1

- How much log-odds of P_1 increase?
- How much odds of P_1 increase?
- How much probability of P_1 increase?
- Calculate for w_1=2,3

Constructing a Learning Algorithm

- Q: How to find **W**?
- We choose parameters w that satisfy maximize of conditional probability:

$$\mathbf{w} = \arg\max_{\mathbf{w}} \prod_{l} P(y^{l} \mid \mathbf{x}^{l}, \mathbf{w})$$

- Maximum Likehood Estimation MLE.
- Note:
  - Here x^l and y^l are pre-determined from training data.
  - Intercept w_0 and coefficient w_i calculated so as to maximize probability
  - So, how many w should we try out – it is continuous? By what method?

Constructing a Learning Algorithm

- We take log of the conditional probabilities (why?):

$$\mathbf{w} = \arg\max_{\mathbf{w}} \sum_l \ln P(y^l \mid \mathbf{x}^l, \mathbf{w})$$

- We note that y^l can be either 1 or 0.

$$l(\mathbf{w}) = \sum_l y^l \ln P(y^l = 1 \mid \mathbf{x}^l, \mathbf{w}) + (1 - y^l) \ln P(y^l = 0 \mid \mathbf{x}^l, \mathbf{w})$$

Computing the Log-Likelihood

- We can re-express the log of the conditional likelihood as:

$$l(\mathbf{w}) = \sum_l y^l \ln P(y^l = 1 \mid \mathbf{x}^l, \mathbf{w}) + (1 - y^l) \ln P(y^l = 0 \mid \mathbf{x}^l, \mathbf{w})$$

$$= \sum_l y^l \ln \frac{P(y^l = 1 \mid \mathbf{x}^l, \mathbf{w})}{P(y^l = 0 \mid \mathbf{x}^l, \mathbf{w})} + \ln P(y^l = 0 \mid \mathbf{x}^l, \mathbf{w})$$

$$= \sum_l y^l (w_0 + \sum_{i=1}^n w_i x_i^l) - \ln(1 + \exp(w_0 + \sum_{i=1}^n w_i x_i^l))$$

- Need to maximize *l(w)*

Fitting LR by Gradient Ascent

- Unfortunately, there is no closed form solution to maximizing l(**w**) with respect to **w**. Therefore, one common approach is to use gradient ascent
- The i th component of the vector gradient has the form

$$\frac{\partial}{\partial w_i} l(\mathbf{w}) = \sum_l x_i^l (y^l - \hat{P}(y^l = 1 | \mathbf{x}^l, \mathbf{w}))$$

Fitting LR by Gradient Ascent

- Use standard gradient ascent to optimize **w**. Begin with initial weights = zero

$$w_i \leftarrow w_i + \eta \sum_l x_i^l (y^l - \hat{P}(y^l = 1 \mid \mathbf{x}^l, \mathbf{w}))$$

Regularization in Logistic Regression

- Overfitting the training data is a problem that can arise in Logistic Regression, especially when data has very high dimensions and is sparse.

- One approach to reducing overfitting is regularization, in which we create a modified "penalized log likelihood function," which penalizes large values of **w**.

$$\mathbf{w} = \arg\max_{\mathbf{w}} \sum_{l} \ln P(y^l \mid \mathbf{x}^l, \mathbf{w}) - \frac{\lambda}{2} \|\mathbf{w}\|^2$$

Regularization in Logistic Regression

- The derivative of this penalized log likelihood function is similar to our earlier derivative, with one additional penalty term

$$\frac{\partial}{\partial w_i} l(\mathbf{w}) = \sum_l x_i^l (y^l - \hat{P}(y^l = 1 \mid \mathbf{x}^l, \mathbf{w})) - \lambda w_i$$

- which gives us the modified gradient descent rule

$$w_i \leftarrow w_i + \eta \sum_l x_i^l (y^l - \hat{P}(y^l = 1 \mid \mathbf{x}^l, \mathbf{w})) - \eta \lambda w_i$$

Summary of Logistic Regression

- Learns the Conditional Probability Distribution P(y|x)
- Local Search.
  - Begins with initial weight vector.
  - Modifies it iteratively to maximize an objective function.
  - The objective function is the conditional log likelihood of the data – so the algorithm seeks the probability distribution P(y|x) that is most likely given the data.

What you should know LR

- In general, NB and LR make different assumptions
  - NB: Features independent given class -> assumption on P(X|Y)
  - LR: Functional form of P(Y|X), no assumption on P(X|Y)
- LR is a linear classifier
  - decision rule is a hyperplane
- LR optimized by conditional likelihood
  - no closed-form solution
  - concave -> global optimum with gradient ascent