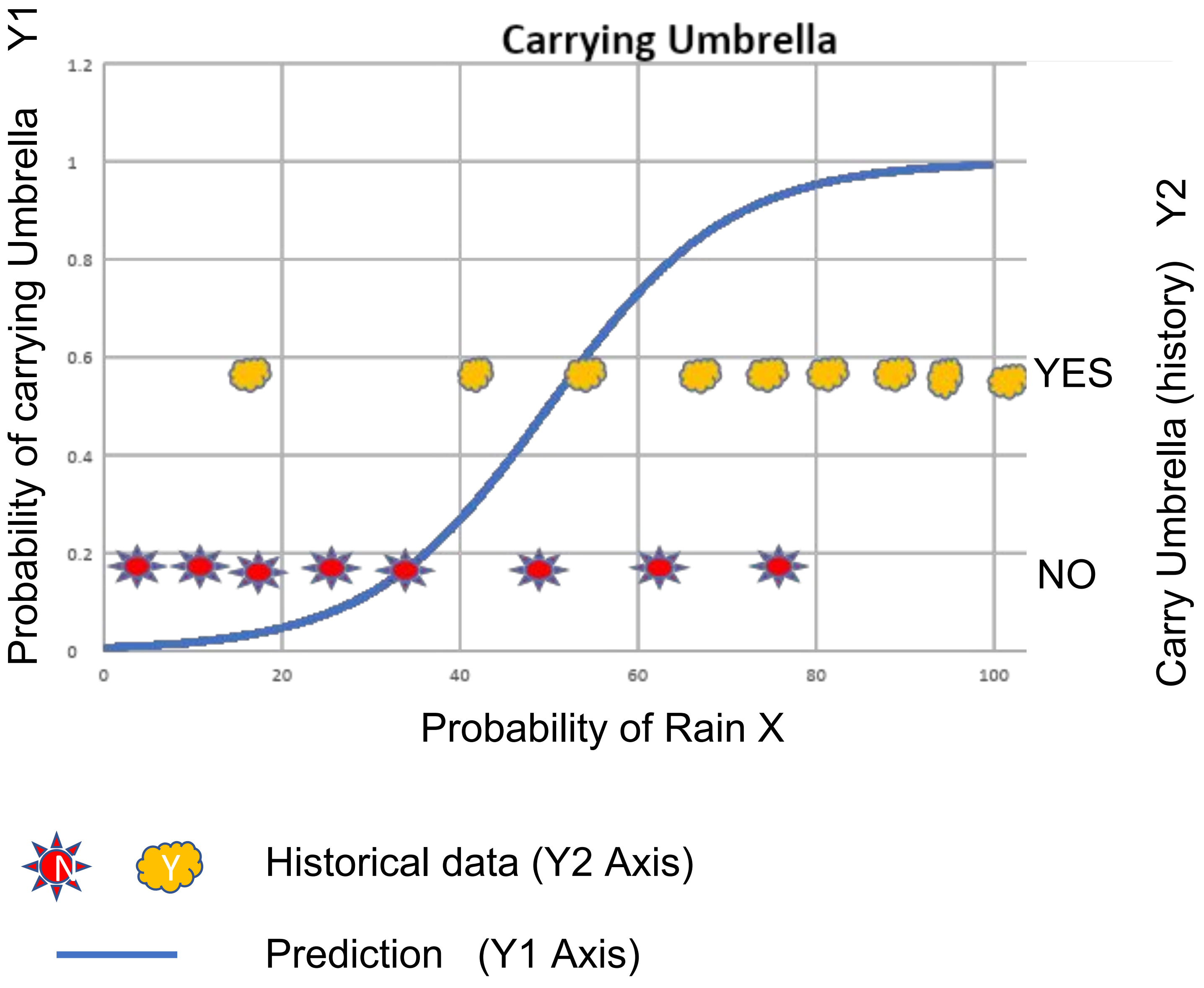
Logistic Regression

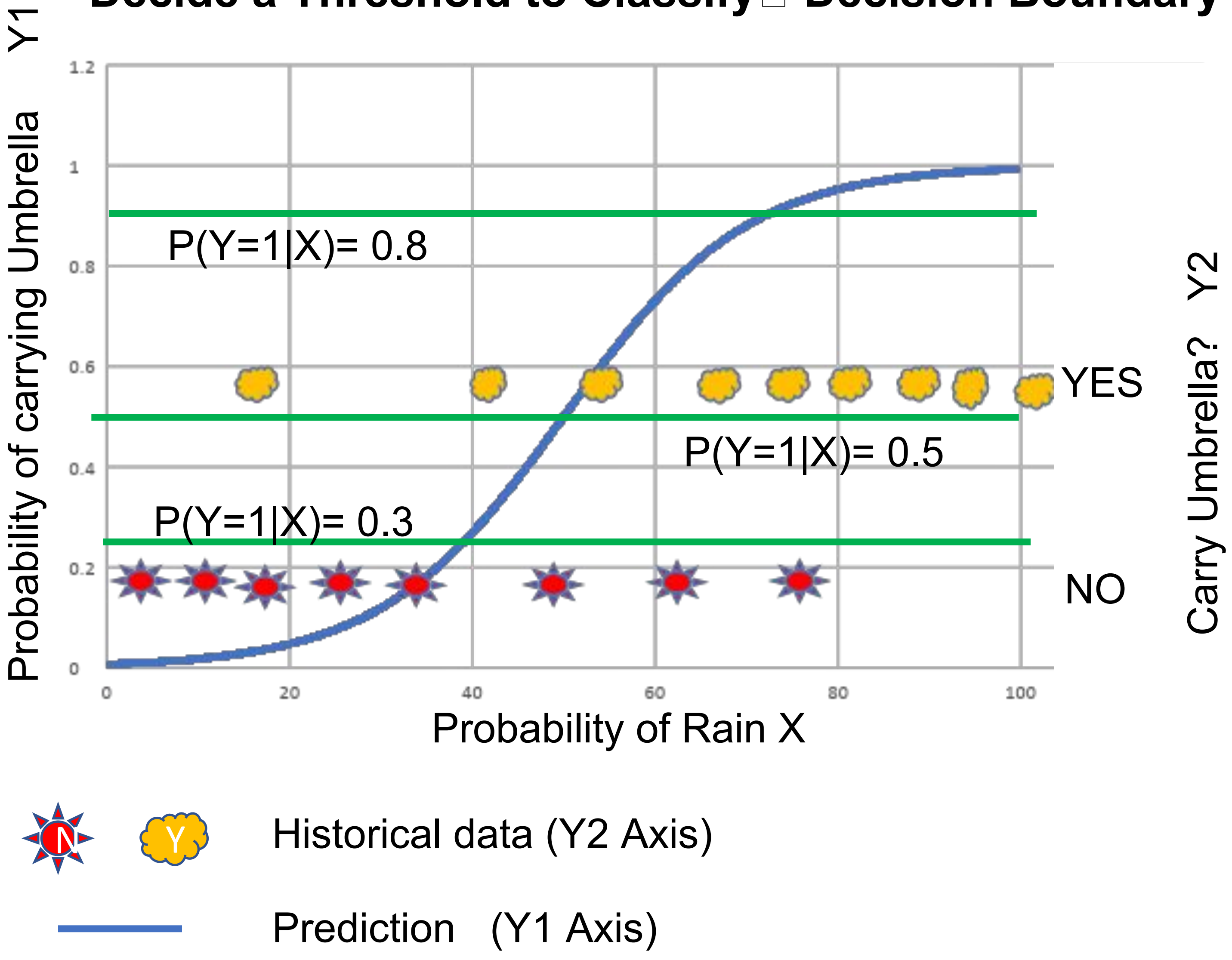
Probabilistic classification

- Most real-life prediction scenarios with discreet outputs, such as *Yes/No,* are probabilistic:
  - Will you carry an umbrella if it is raining?
  - Will you carry an umbrella if it is sunny?
  - Will you carry an umbrella if it drizzles?
- Logistic Regression gives the probability of an event occurring, given historical data to train-test the model



Carrying Umbrella
Y1
Probability of carrying Umbrella
1.2
1
0.8
0.6
0.4
0.2
0
0
20
40
60
80
100
Probability of Rain X
YES
NO
Y2
Carry Umbrella (history)
N
Y
Historical data (Y2 Axis)
Prediction (Y1 Axis)

Decide a Threshold to Classify□ Decision Boundary



Logistic Regression

- Logistic regression gives the *probability of an event occurring,* given historical data {y,X}:

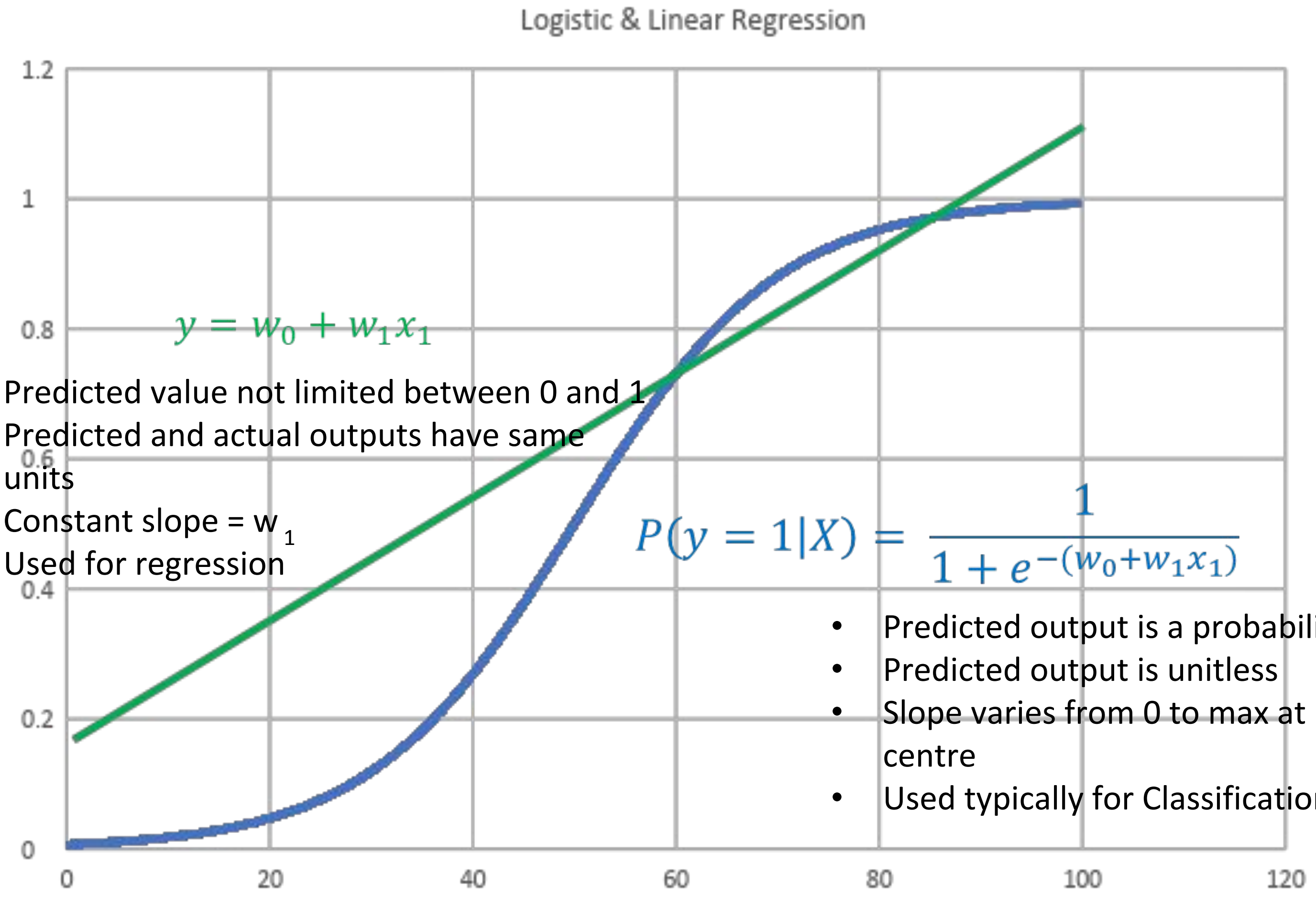
$$\boldsymbol{P}(\boldsymbol{y} = \mathbf{1}|\overrightarrow{\boldsymbol{X}}, \overrightarrow{\boldsymbol{w}}) = \frac{1}{1 + e^{-\vec{w}\cdot\vec{X}}}$$

- Where: $\vec{w} \cdot \vec{X} = w_0 + w_1 x_1 + \cdots \ldots + w_k x_k$

w are the adjustable weight parameters

- P(Y= ... sition
  - This is the Sigmoid function

A Comparison



$$y = w_0 + w_1 x_1$$

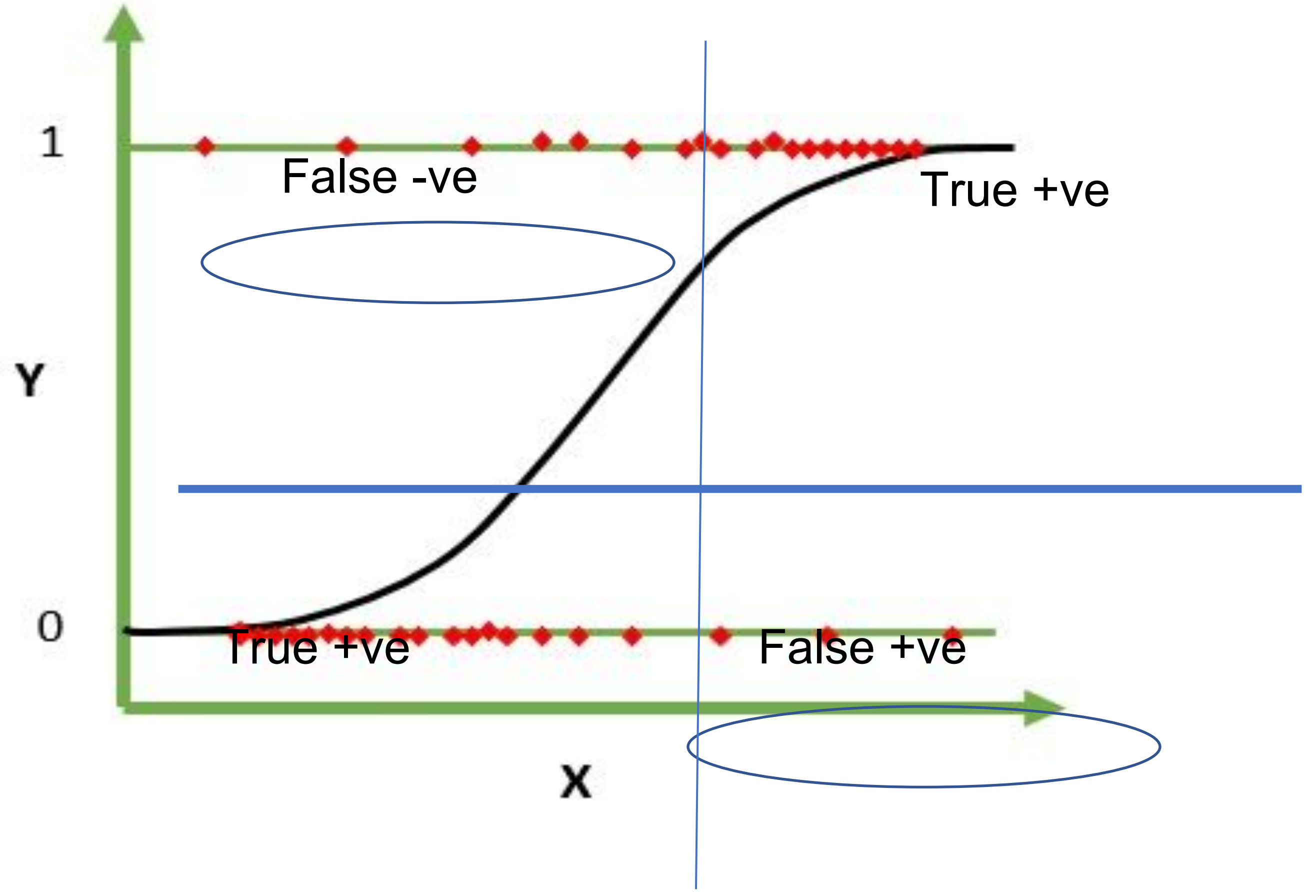
- Predicted value not limited between 0 and 1
- Predicted and actual outputs have same units
- Constant slope = w_1
- Used for regression

$$P(y = 1|X) = \frac{1}{1 + e^{-(w_0 + w_1 x_1)}}$$

- Predicted output is a probability,
- Predicted output is unitless
- Slope varies from 0 to max at centre
- Used typically for Classification

Midpoint & Slope

Performance Tallies



Log odds or Logit

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

$$p_1 = \frac{1}{1+e^{-\mathbf{wx}}} \qquad 1 - p_1 = 1 - \frac{1}{1+e^{-\mathbf{wx}}}$$

- Odds:

- Log Odds:

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

Model Fitting

Let p_1 be P(y=1|x,w)

Sequence n:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Actual Data y:	1	1	1	0	0	0	1	1	1	1	1	0	0	0	0	1	1	1
Prediction p:	p_1	p_1	p_1	$1-p_1$	$1-p_1$	$1-p_1$	p_1	p_1	p_1	p_1	p_1	$1-p_1$	$1-p_1$	$1-p_1$	$1-p_1$	p_1	p_1	p_1

Likelihood of a match?

Let p_1 be $P(y=1|x,w)$

Sequence n:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Actual Data y:	1	1	1	0	0	0	1	1	1	1	1	0	0	0	0	1	1	1
Prediction p:	p_1	p_1	p_1	$1-p_1$	$1-p_1$	$1-p_1$	p_1	p_1	p_1	p_1	p_1	$1-p_1$	$1-p_1$	$1-p_1$	$1-p_1$	p_1	p_1	p_1

Likelihood of a match? Note y_n can be either 1 or 0

$$\mathcal{L}(w) = \prod_n {p_1}^{y_n} (1-p_1)^{1-y_n}$$

Log Likelihood:

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

Training

- Maximum Likehood Estimation MLE.

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

- Note:
  - Here x^l and y^l are pre-determined from training data.
  - Intercept w_0, and coefficients w_i calculated so as to maximize probability
  - So, how many *w* should we try out?

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 LogR 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

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

Fitting LogR 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**.

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

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 LogR

- 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)
- LogR can be used as a linear classifier
  - decision rule is a hyperplane
- LogR optimized by conditional likelihood
  - no closed-form solution
  - concave -> global optimum with gradient ascent