TD2 Linear Classification:
Bayes Optimal & Naive Bayes, Logistic Regression

0: Decision Boundary

- Suppose you are given the following classification task: Predict the target $Y \in \{0, 1\}$ given two real valued features $X_1, X_2 \in R$
- After some training, you learn the following decision rule:

  Predict $Y = 1$ iff $w_1X_1 + w_2X_2 + w_0 \geq 0$ and $Y = 0$ otherwise,

  where $w_1 = 2$, $w_2 = 6$, and $w_0 = 12$

(a) Plot the decision boundary and label the region where we would predict $Y = 1$ and $Y = 0$

(b) Suppose that we learned the above weights using logistic regression

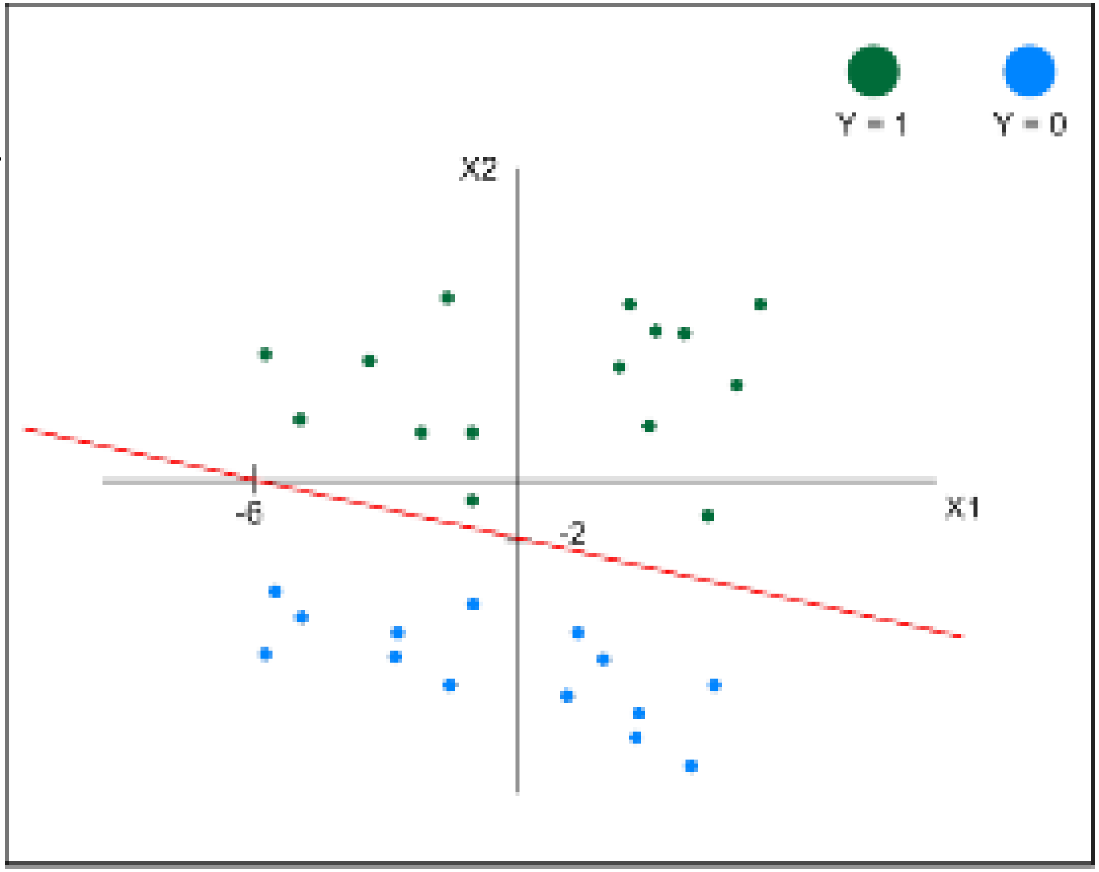
- Using this model, what would be our prediction for $P(Y = 1 \mid X_1, X_2)$?

Hint: You may want to use the sigmoid function $S(x) = 1/\,(1+e^{-x})$

0: Solution

(a) We know that iff $2X_1 + 6X_2 + 12 \geq 0$ then $Y = 1$

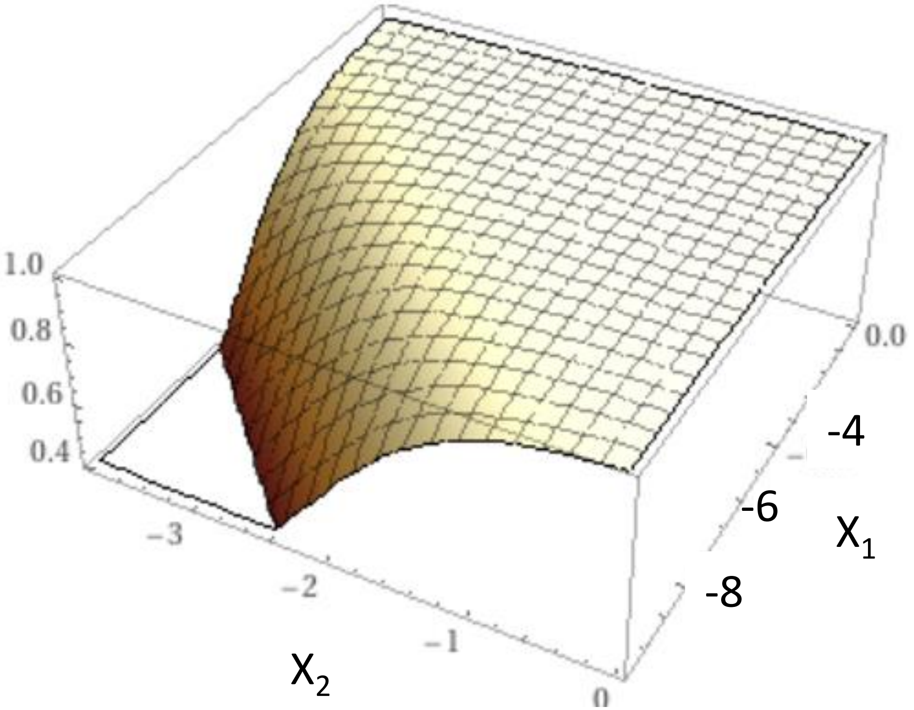
- To draw this linear boundary, we need to find X_1 and X_2 given the decision rule
- We first search for X_1 when $X_2 = 0$:
  $2X_1 = -12 \Rightarrow X_1 = -6$ and $X_2 = 0$
- We now search for X_2 when $X_1 = 0$:
  $6X_2 = -12 \Rightarrow X_2 = -2$ and $X_1 = 0$
- For the above values, the decision boundary is depicted in the left Figure, where the green points will be predicted as $Y = 1$ and the blue ones as $Y = 0$



0: Solution

(b) Given that we learned the weights using logistic regression, we need to calculate the following conditional probability for any given X_1 and X_2 using the sigmoid function as follows:

$$P(Y=1|X_1,X_2) = \frac{1}{1+\exp(2X_1+6X_2+12)}$$



- The above conditional probability will be our prediction

1: Bayes Optimal and Naïve Bayes Classifier

- Consider the following distributions used by a Naive Bayes classifier:
  - the joint probability distribution over 3 Boolean variables x_1, x_2, y given in Figure a
  - the marginal probabilities for this same distribution, given in Figures b, c, and d

(a) What is the decision rule used by the Bayes optimal classifier?

(b) Express $P_D(y=0 \mid x_1, x_2)$ in terms of $P_D(x_1, x_2, y = 0)$ and $P_D(x_1, x_2, y= 1)$

(c) Write out an expression for the value of $P(y = 1 \mid x_1 = 1, x_2 = 0)$ predicted by the Bayes optimal classifier

(d) Write out an expression for the value of $P(y = 1 \mid x_1 = 1, x_2 = 0)$ predicted by the naive Bayes classifier

(e) Explain why the expressions you wrote for (c) and (d) are unequal

Hint: Recall that Joint Pr. = Conditional Pr. × Marginal Pr.

x_1	x_2	y	$p_{\mathcal{D}}(x_1, x_2, y)$
0	0	0	.15
0	0	1	.25
0	1	0	.05
0	1	1	.08
1	0	0	.1
1	0	1	.02
1	1	0	.2
1	1	1	.15

(a) Joint distribution

Likelihood Tables

	$x_1 = 0$	$x_1 = 1$
$y = 0$	.4	.6
$y = 1$	.66	.34

(b) $P_{\mathcal{D}}(x_1 \mid y)$

	$x_2 = 0$	$x_2 = 1$
$y = 0$	.5	.5
$y = 1$	.54	.46

(c) $P_{\mathcal{D}}(x_2 \mid y)$

Prior Tables

y	$P_{\mathcal{D}}(y)$
$y = 0$	.5
$y = 1$	.5

(d) $p_{\mathcal{D}}(y)$

1: Solution

(a) if $P(y=1|x_1,x_2) > P(y=0|x_1,x_2)$ then $\hat{y}=1$, or else $\hat{y}=0$

(b) $$P_{\mathcal{D}}(y=0|x_1,x_2) = \frac{P_{\mathcal{D}}(x_1,x_2,y=0)}{P_{\mathcal{D}}(x_1,x_2,y=0)+P_{\mathcal{D}}(x_1,x_2,y=1)}$$

(c) $$P(y=1|x_1,x_2) = \frac{P_{\mathcal{D}}(x_1,x_2,y=1)}{P_{\mathcal{D}}(x_1,x_2,y=0)+P_{\mathcal{D}}(x_1,x_2,y=1)} = \frac{.02}{.1+.02}$$

(d) $$P(y=1|x_1,x_2) = \frac{P(x_1,x_2,y=1)}{P(x_1,x_2,y=0)+P(x_1,x_2,y=1)}$$

$$= \frac{P_{\mathcal{D}}(y=1)P_{\mathcal{D}}(x_1|y=1)P_{\mathcal{D}}(x_2|y=1)}{P_{\mathcal{D}}(y=0)P_{\mathcal{D}}(x_1|y=0)P_{\mathcal{D}}(x_2|y=0)+P_{\mathcal{D}}(y=1)P_{\mathcal{D}}(x_1|y=1)P_{\mathcal{D}}(x_2|y=1)}$$

(e) Bayes optimal classifier does not have the assumption of conditional independence!

1: Bayes Optimal and Naive Bayes Classifier

(f) Suppose you have a dataset involving five random variables: x_1, x_2, x_3, x_4 and y

- Variables x_i and x_j are conditionally independent given y, for all i and j except for the pair x_3 and x_4, which are not conditionally independent
- Therefore, you can't quite use Naive Bayes unless you extend it to handle the dependence between x_3 and x_4

- Write down the decision rule you would use in place of the Naive Bayes rule, to correctly model this data set
  - Hint: try rederiving the Naive Bayes decision rule but avoiding the conditional independence assumption for x_3 and x_4

1: Solution

(f)
$$\frac{P_{\mathcal{D}}(y=1)P_{\mathcal{D}}(x_1|y=1)P_{\mathcal{D}}(x_2|y=1)P_{\mathcal{D}}(x_3,x_4|y=1)}{P_{\mathcal{D}}(y=0)P_{\mathcal{D}}(x_1|y=0)P_{\mathcal{D}}(x_2|y=0)P_{\mathcal{D}}(x_3,x_4|y=0)}$$

- If this quality is larger than 1 then y^ = 1, or else y^ = 0

1: Bayes Optimal and Naive Bayes Classifier

(g) Suppose you know for fact that x_1, x_2, y are independent random variables

In this case is it possible for any other classifier (e.g., a decision tree) to do better than a naive Bayes classifier?

Hint: The dataset is irrelevant for this question

1: Solution

(g) The independency of x_1, x_2, y does not imply that they are independent within each class

- in other words, they are not necessarily independent when conditioned on y

- Therefore, naive Bayes classifier may not be able to model the function well, while a decision tree might

- For example, $y = x_1$ XOR x_2, is an example where x_1, x_2, might be independent variables, but a naive Bayes classifier will not model the function well since for a particular class (say, $y = 0$), x_1 andx_2, are dependent

2: Estimating Naïve Bayes Parameters

- Suppose you have the following training set with three Boolean inputs x, y and z, and a Boolean output U and you should predict U using a NB classifier

x	y	z	U
1	0	0	0
0	1	1	0
0	0	1	0
1	0	0	1
0	0	1	1
0	1	0	1
1	1	0	1

(a) After NB learning is completed what would be the predicted probability $P(U = 0 \mid x = 0, y = 1, z = 0)$?

(b) Using the probabilities obtained during the NB classifier training, what would be the predicted probability $P(U = 0 \mid x = 0)$

2: Solution

(a) The training of the NB is depicted in the following Tables

- Frequency Tables for the Three Variables

Variable **x**		
	U = 0	U = 1
X = 0	2	2
X = 1	1	2

Variable **y**		
	U = 0	U = 1
Y = 0	2	2
Y = 1	1	2

Variable **z**		
	U = 0	U = 1
Z = 0	1	3
Z = 1	2	1

- Likelihood Tables for the Three Variables

Variable **X**		
	U = 0	U = 1
X = 0	2/3	2/4
X = 1	1/3	2/4

Variable **Y**		
	U = 0	U = 1
Y = 0	2/3	2/4
Y = 1	1/3	2/4

Variable **Z**		
	U = 0	U = 1
Z = 0	1/3	3/4
Z = 1	2/4	1/4

2: Solution

(a) We calculate the prior probabilities by marginalizing for each sub-table of the Frequency Table for the columns U = 0 and column U = 1

- As a result, P(U = 0) = 3/7 and P(U = 1) = 4/7

- Then we calculate the following probability P(U = 0 | x = 0, y = 1, z = 0):

$$P(U=0|x=0,y=1,z=0) = \frac{P(U=0) * \prod_i P(X_i|U=0)}{\sum_j (P(U=u_j) * \prod_i P(X_i|U=u_j))} =$$

- To solve this Equation, we need to calculate the probabilities:

$$P(U=0) * P(x=0,y=1,z=0|U=0) = P(U=0) * P(x=0|U=0) * P(y=1|U=0) * P(z=0|U=0)$$

$$= \frac{3}{7} * \frac{2}{3} * \frac{1}{3} * \frac{1}{3} = \frac{2}{63} = 0.031$$

and

$$P(U=1) * P(x=0,y=1,z=0|U=1) = P(U=1) * P(x=0|U=1) * P(y=1|U=1) * P(z=0|U=1)$$

$$= \frac{4}{7} * \frac{2}{4} * \frac{2}{4} * \frac{3}{4} = \frac{3}{28} = 0.107$$

2: Solution

(a) Then we reformulate the equation for P(U = 0 | x = 0, y = 1, z = 0) as follows:

$$P(U=0|x=0,y=1,z=0) = \frac{P(U=0)*P(x=0,y=1,z=0|U=0)}{P(U=0)*P(x=0,y=1,z=0|U=0)+P(U=1)*P(x=0,y=1,z=0|U=1)}$$

$$= \frac{0.031}{0.031+0.107} = 0.225$$

= (2/63) / (2/63 + 3/28) = 8/35

(b) We would like to find the probability P(U = 0 | x = 0)

- In that case given that we have the value of x = 0 we can calculate the probability as:

$$P(U=0|x=0) = \frac{P(U=0)*P(x=0|U=0)}{P(U=0)*P(x=0|U=0)+P(U=1)*P(x=0|U=1)} =$$

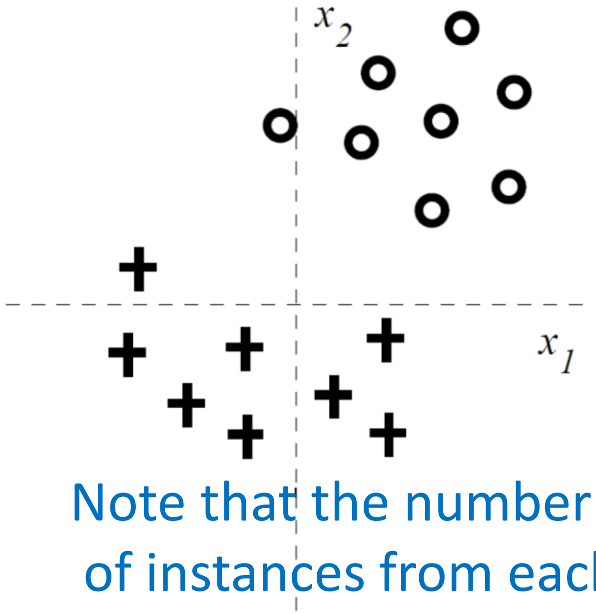
$$\frac{\frac{3}{7}*\frac{2}{3}}{\frac{3}{7}*\frac{2}{3}+\frac{4}{7}*\frac{2}{4}} = \frac{1}{2} = 0.5$$

3: Logistic Regression

- We consider here a discriminative approach for solving the classification problem illustrated in the next Figure
  - A 2-D labeled training set, where `+' corresponds to class y=1 and `O' corresponds to class y = 0

(a) We attempt to solve the binary classification task depicted in next Figure with the simple linear logistic regression model

$$P(y=1|\vec{x},\vec{w}) = g(w_0 + w_1x_1 + w_2x_2) = \frac{1}{1+exp(-w_0 - w_1x_1 - w_2x_2)}$$



Note that the number of instances from each class is the same

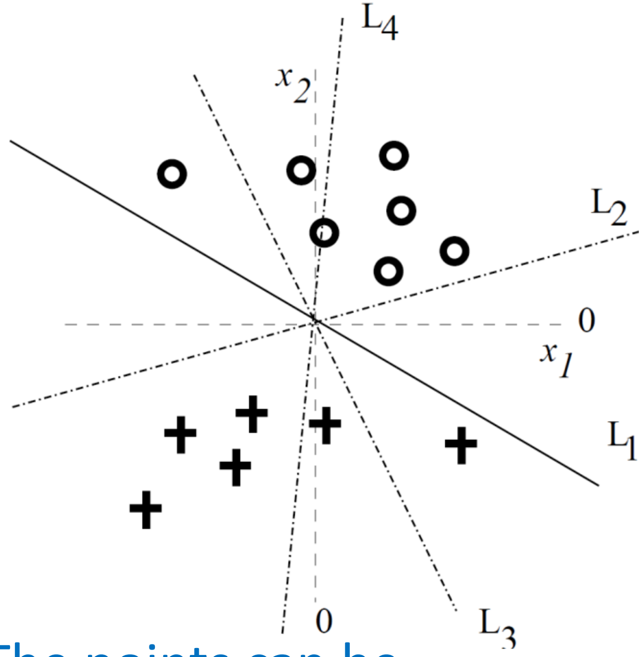
3: Logistic Regression

- Notice that the training data can be separated with zero training error with a linear separator (see L_1)

- Consider training regularized linear logistic regression models where we try to maximize for very large C

$$\sum_{i=1}^{n} \log\left(P(y_i|x_i, w_0, w_1, w_2)\right) - Cw_j^2$$

- The regularization penalties used in penalized conditional loglikelihood estimation are $-Cw^2_j$, where j = {0, 1, 2}
  - In other words, only one of the parameters is regularized in each case

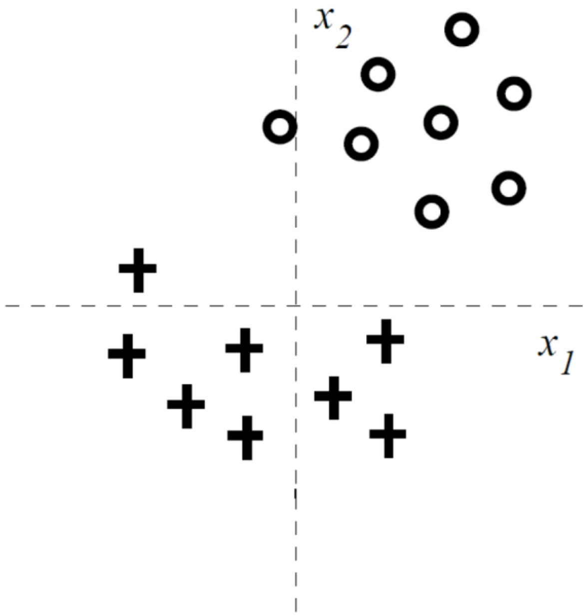


The points can be separated by L_1 (solid line) Possible other decision boundaries are shown by L_2,L_3,L_4

3: Logistic Regression

(a) Given the training data of the next Figure, how does the training error change with regularization of each parameter w_j?

$$\sum_{i=1}^{n} \log \left(P(y_i|x_i, w_0, w_1, w_2)\right) - Cw_j^2$$



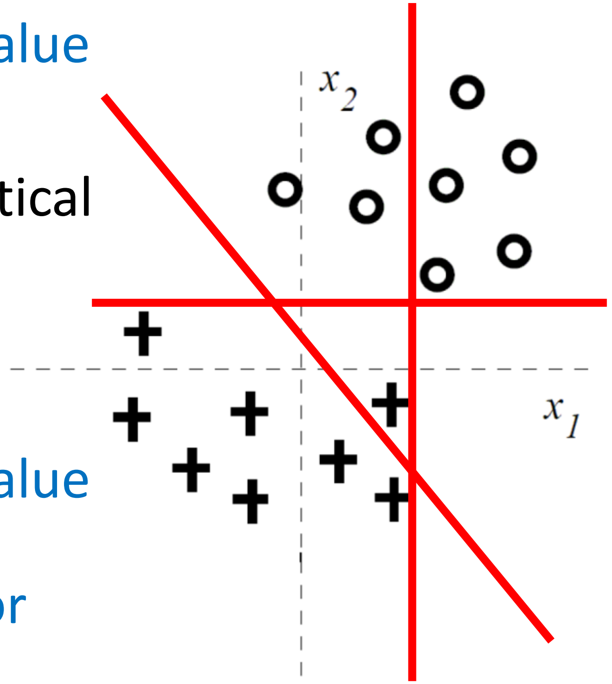
- State whether the training error increases or stays the same (zero) for each w_j for very large C

(a)

3:Solution

i) By regularizing w_2

- Increases
- When we regularize w_2, the resulting boundary can rely less and less on the value of x_2 and therefore becomes more vertical
  - For very large C, the training error increases as there is no good linear vertical separator of the training data



ii) By regularizing w_1

- Remains the same
- When we regularize w_1, the resulting boundary can rely less and less on the value of x_1 and therefore becomes more horizontal
  - For very large C, the training data can be separated with zero training error with a horizontal linear separator

iii) By regularizing w_0

- Increases
- When we regularize w_0, then the boundary will eventually go through the origin (bias term set to zero)
  - Based on the figure, we can not find a linear boundary through the origin with zero error: the best we can get is one error!

3:Logistic Regression

(b) If we change the form of regularization to L1-norm (absolute value) and regularize w_1 and w_2 only (but not w_0), we get the following penalized log-likelihood

$$\sum_{i=1}^{n} \log P(y_i|x_i, w_0, w_1, w_2) - C(|w_1| + |w_2|)$$

- Consider again the same dataset and the same linear logistic regression model

$$P(y = 1|\vec{x}, \vec{w}) = g(w_0 + w_1 x_1 + w_2 x_2)$$

i) As we increase the regularization parameter C which of the following scenarios do you expect to observe?

1. First w_1 will become 0, then w_2
2. First w_2 will become 0, then w_1
3. w_1 and w_2 will become zero simultaneously
4. None of the weights will become exactly zero, only smaller as C increases

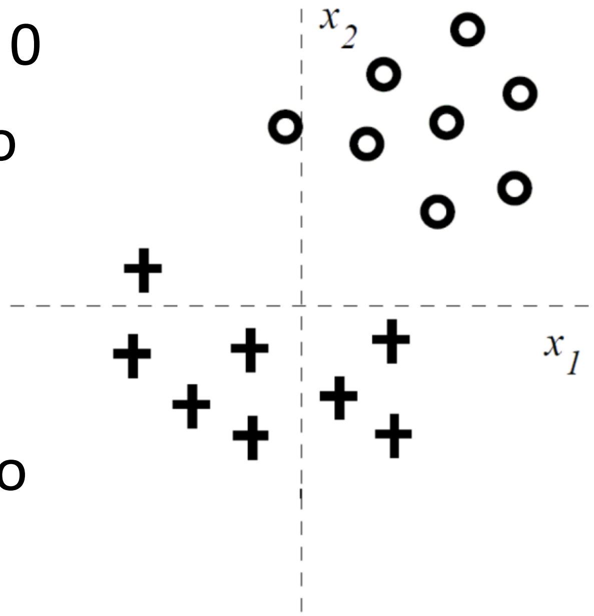
Choose only one and briefly explain your choice

3:Solution

(b)

i) 1. First w_1 will become 0, then w_2

- The data can be classified with zero training error and therefore also with high log-probability by looking at the value of x_2 alone, i.e., making $w_1 = 0$
  - Initially we might prefer to have a non-zero value for w_1 but it will go to zero rather quickly as we increase regularization
  - Note that we pay a regularization penalty for a non-zero value of w_1 and if it does not help classification why would we pay the penalty?
  - Also, the absolute value regularization ensures that w_1 will indeed go to exactly zero
- As C increases further, even w_2 will eventually become zero
  - We pay higher and higher cost for setting w_2 to a non-zero value
  - Eventually this cost overwhelms the gain from the log-probability of examples that we can achieve with a non-zero w_2



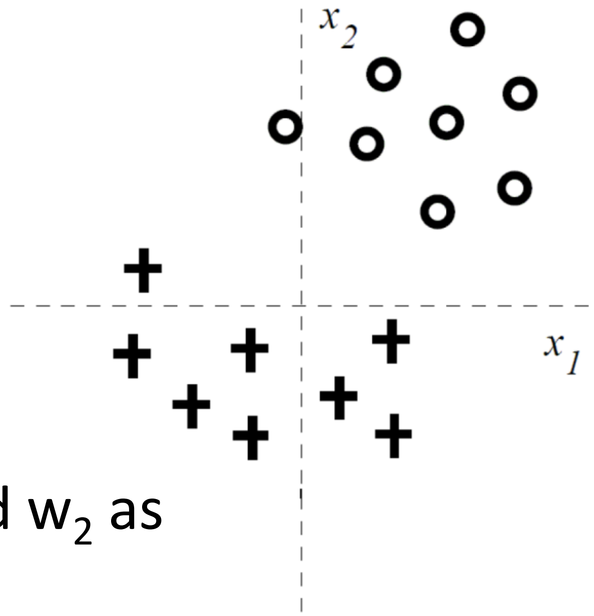
3:Logistic Regression

(b)

ii) Recall that the classification problem illustrated in the Figure is balanced: the number of examples from each class is the same

- For very large C, with the same L1-norm regularization for w_1 and w_2 as above, which value(s) do you expect w_0 to take? Explain briefly

Hint: You can give a range of values for w_0 if you deem necessary

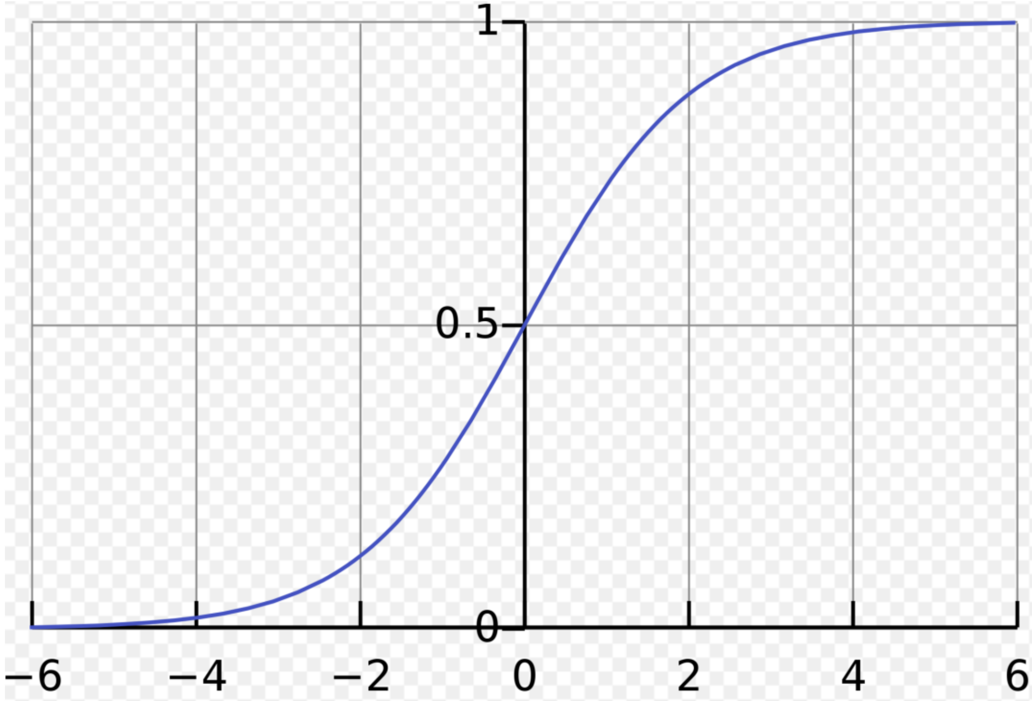


iii) Assume now that we obtain more examples from the `+' class that corresponds to y=1 so our classification problem become unbalanced

- Again, for very large C, with the same L1-norm regularization for w_1 and w_2 as above, which value(s) do you expect w_0 to take? Explain briefly

Hint: You can give a range of values for w_0 if you deem necessary

3:Solution



(b) For very large C, we argued that both w_1 and w_2 will go to zero

ii) With balanced classes, the number of examples in each class is the same n and so we would like to predict each one with the same probability $P(y=1|\vec{x},\vec{w})$=$P(y=0|\vec{x},\vec{w})$=0.5.

- Note that when $w_1 = w_2 = 0$, the log-probability of examples becomes a finite value, which is equal to n log(0.5), i.e., w_0=0 makes $P(y=1|\vec{x},\vec{w})$=0.5.

iii) With unbalanced classes, where the number of '+' examples are greater than that of 'o' examples, we want to have $P(y=1|\vec{x},\vec{w}) > P(y=0|\vec{x},\vec{w})$

- Hence, the value of w_0 should be greater than zero which makes $P(y=1|\vec{x},\vec{w}) > 0.5$.

4:Multi-class Logistic Regression

- One way to extend logistic regression to multi-class (say K class labels) setting is to consider (K-1) sets of weight vectors and define

$$P(Y = y_k|X) \propto \exp(w_{k0} + \sum_{i=1}^{d} w_{ki} X_i) \text{ for } k = 1, \ldots, K-1$$

(a) What model does this imply for $P(Y = y_K \mid X)$?

(b) What would be the classification rule in this case?

(c) Draw a set of training data with three labels and the decision boundary resulting from a multi-class logistic regression

- The boundary does not need to be quantitatively correct but should qualitatively depict how a typical boundary from multi-class logistic regression would look like

4: Solution

(a) Since all probabilities must sum to 1, we should have

$$P(Y = y_K|X) = 1 - \sum_{k=1}^{K-1} P(Y = y_k|X)$$

- Also, note that introducing another set of weights for this class will be redundant, just as in binary classification. We can define

$$P(Y = y_K|X) = \frac{1}{1 + \sum_{k=1}^{K-1} \exp(w_{k_0} + \sum_{i=1}^{d} w_{k_i} X_i)}$$

and for k = 1, …, K - 1

$$P(Y = y_k|X) = \frac{\exp(w_{k_0} + \sum_{i=1}^{d} w_{k_i} X_i)}{1 + \sum_{k=1}^{K-1} \exp(w_{k_0} + \sum_{i=1}^{d} w_{k_i} X_i)}$$

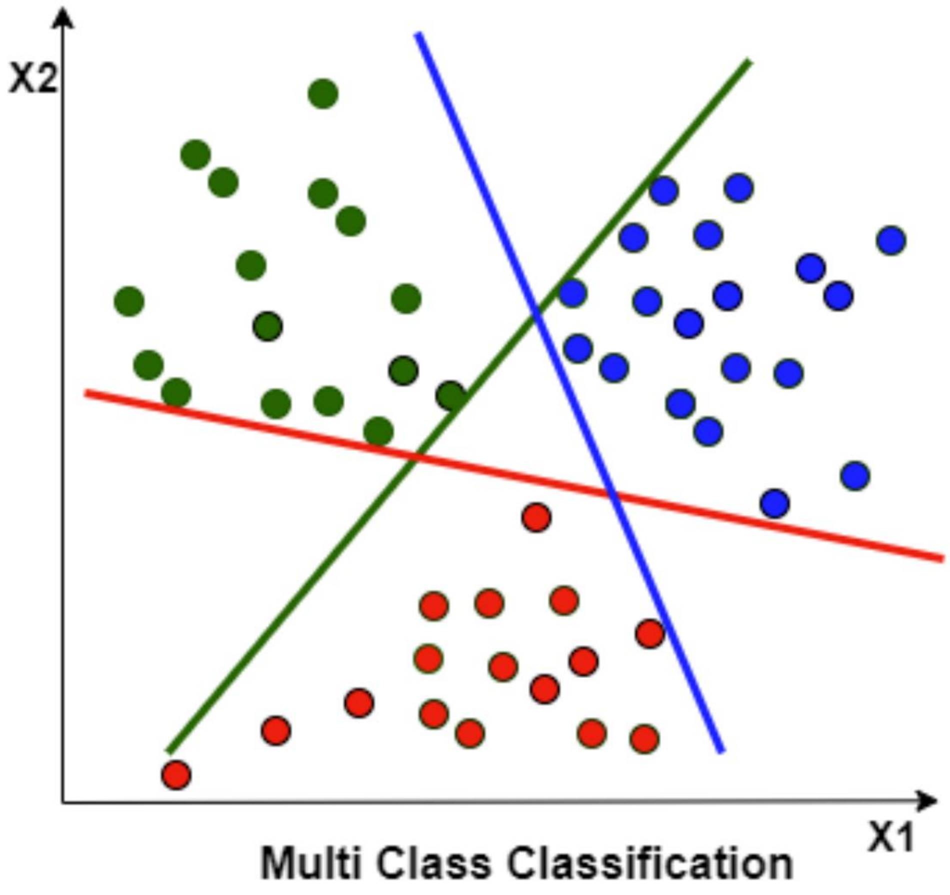
4: Solution

(b) The classification rule simply picks the label with highest probability:

$$y = y_{k^*} \text{ where } k^* = \arg \max_{k \in \{1,\ldots,K\}} P(Y = y_k | X)$$

(c) The decision boundary between each pair of classes is linear and hence the overall decision boundary is piece-wise linear

- Equivalently, since arg $\max_i \exp(a_i)$ = arg $\max_i a_i$ and max of linear functions is piece-wise linear, the overall decision boundary is piece-wise linear



https://satishgunjal.com/binary_lr

4 Bonus: Overfitting and Regularized Logistic Regression (20 pts)

- To prevent overfitting, we want the weights to be small. To achieve this, instead of maximum conditional likelihood estimation M(C)LE for logistic regression:

$$\max_{w_0,\ldots,w_d} \prod_{i=1}^{n} P(Y_i|X_i, w_0, \ldots, w_d)$$

- We can consider maximum conditional a posterior M(C)AP estimation:

$$\max_{w_0,\ldots,w_d} \prod_{i=1}^{n} P(Y_i|X_i, w_0, \ldots, w_d)P(w_0, \ldots, w_d)$$

where $P(w_0,\ldots,w_d)$ is a prior on the weights

- Assuming a standard Gaussian prior N(0, **I**) for the weight vector, derive the gradient ascent update rules for the weights