Logistic Regression (Prashant)

1. Overfitting and Regularized Logistic Regression [10 pts]

a) Plot the sigmoid function $1/(1+e^{-wX})$ vs. $X \in \mathbb{R}$ for increasing weight $w \in \{1, 5, 100\}$. A qualitative sketch is enough. Use these plots to argue why a solution with large weights can cause logistic regression to overfit.

b) 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:

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

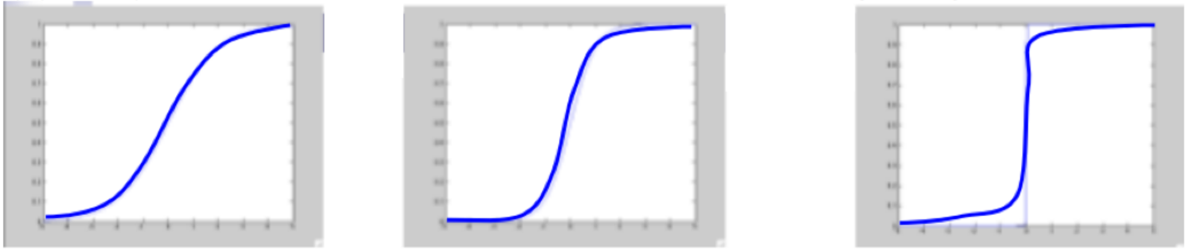
we can consider maximum conditional a posterior M(C)AP estimation:

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

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

Assuming a standard Gaussian prior $\mathcal{N}(0, \boldsymbol{I})$ for the weight vector, derive the gradient ascent update rules for the weights.

a) (3 points) This is a picture of how plots of the sigmoid change with $w \in \{1, 5, 100\}$ from left to right.



As we can see, the curve gets steeper as w gets larger. The steeper curve means that the model will be nearly completely sure of the class (almost 0 probability or almost 1 probability). With large weights, small changes in input can lead to a large change in probability of class, leading to easy flipping of the predicted output. This is the intuition behind why it overfits.
Note: Award credit if the answer has the required curves and presents some reasonable intuition for why large w leads to overfitting.

b) (7 points) The derivation for the second part of the question proceeds as follows. We start by writing out the log conditional posterior (akin to log conditional likelihood for the unregularized case). Here $\boldsymbol{w} = [w_0, \dots, w_d]^\top$.

$$L(\boldsymbol{w}) = \log(p(\boldsymbol{w}) \prod_{j=1}^{n} P(y^j | x^j, \boldsymbol{w}))$$

$$p(\boldsymbol{w}) = \prod_{i=0}^{d} \frac{1}{\sqrt{2\pi}} \exp(-\frac{w_i^2}{2})$$

Therefore, the M(C)AP estimate is

$$\boldsymbol{w}^* = \arg\max_{\boldsymbol{w}} L(\boldsymbol{w}) = \arg\max_{\boldsymbol{w}} \left[\sum_{j=1}^{n} \log(P(y^j|x^j,\boldsymbol{w})) - \sum_i \frac{w_i^2}{2}\right]$$

The gradient ascent update rule is:

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \left.\frac{\partial L(\boldsymbol{w})}{\partial w_i}\right|_t$$

The gradient of the log conditional posterior is

$$\frac{\partial L(\boldsymbol{w})}{\partial w_i} = \frac{\partial}{\partial w_i}\log\ p(\boldsymbol{w}) + \frac{\partial}{\partial w_i}\log(\prod_{j=1}^{n} P(y^j|x^j,\boldsymbol{w}))$$

The second term is same as derived in class for unregularized case. First term leads to an extra factor of

$$\frac{\partial}{\partial w_i}\log p(w) = -w_i$$

Therefore, the final update rule is

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta(-w_i^{(t)} + \sum_j x_i^j(y^j - P(Y=1|x^j, w^{(t)})))$$

Note: Award 3 points for the correct log conditional posterior expression or correct M(C)AP estimate expression, 2 points for the correct gradient and 2 points for the correct update step.

2. Multi-class Logistic Regression [10 pts]

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 $P(Y = y_K|X)$?

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.)

a) (3 points)
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 $k = 1, \ldots, K-1$

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

Note: Other solutions are possible too. As long as all probabilities sum to 1 and each $P(Y = y_k|X) \propto \exp(w_{k_0} + \sum_{i=1}^{d} w_{k_i}X_i)$, you can award 2 points. Award 1 more point (i.e. full 3 points) if the students realize that they don't need an extra set of weights.

b) (3 points) 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) (4 points)
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.
Note: Award 4 points to an image with a piecewise-linear decision boundary.

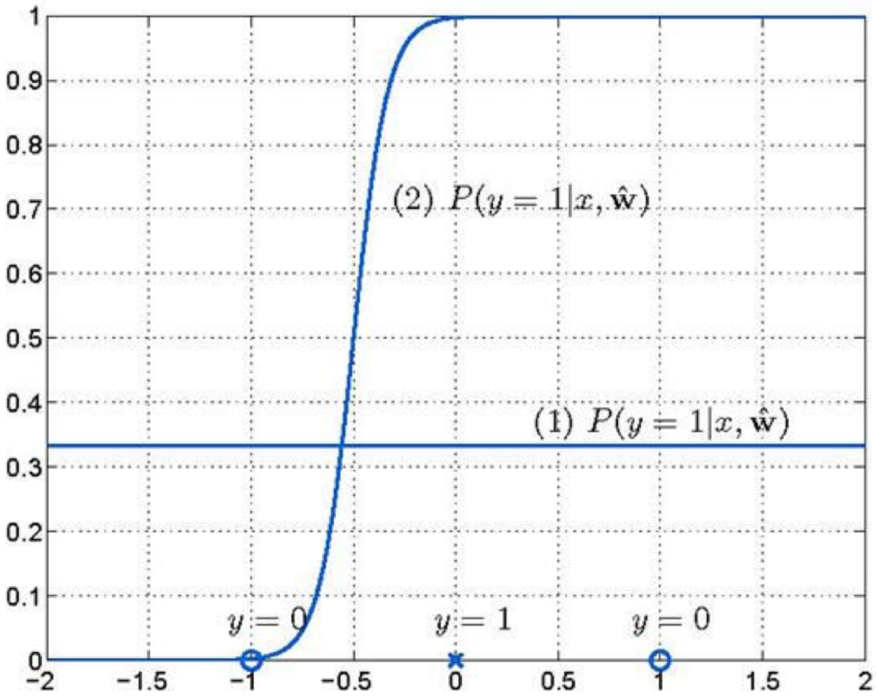
Question 4 – Logistic Regression (12 points)

Consider a simple one dimensional logistic regression model

$P\ (y = 1|x,\ \mathrm{w}) = g(w_0 + w_1 x)$

where g(z) = 1/(1+exp(z)) is the logistic function.

The following figure shows two possible conditional distributions P (y =1|x; w), viewed as a function of x, that we can get by changing the parameters w.



(a) (4 points) Please indicate the number of classification errors for each conditional given the labeled examples in the same figure.

Conditional (1) makes (1) classification errors

Conditional (2) makes (1) classification errors

(b) (4 points) One of the two classifiers corresponds to the maximum likelihood setting of the parameters w based on the labeled data in the figure, i.e. its parameters maximize the joint probability

P(y=0|x=-1; w) P(y=1|x=0; w) P(y=0|x=1; w)

Circle which one is the ML solution and briefly explain why you chose it:

Classifier 1 or Classifier 2

Answer: Class. 1 b/c it can't be classifier 2, for which P(y=0|x-1)=0

(c) (4 points) Would adding a regularization penalty $|w_1|^2/2$ to the log-likelihood estimation criterion affect your choice of solution (Y/N)? (Note that the penalty above only regularizes w_1, not w_0.)? Briefly explain why.

Answer: no, because w_1 is zero for Classifier 1, so no penalty is incurred. Therefore, if it was the ML solution before, it must still be the ML solution.

A. (3 points) Give a *one sentence* reason why:

- we might prefer Decision Tree learning over Logistic Regression for a particular learning task.

  ⋆ *Solution*: If we want our learner to produce rules easily interpreted by humans.
- we might prefer Logistic Regression over Naive Bayes for a particular learning task.

  ⋆ *Solution*: If we know that the conditional independence assumptions made by Naive Bayes are not true for our problem, and we have lots of training data.
- we choose parameters that minimize the sum of squared training errors in Linear Regression.

  ⋆ *Solution*: Because this corresponds to the MLE assuming that data is generated from a linear function plus Gaussian noise.

B. (3 points) Suppose we train several classifiers to learn $f : X \rightarrow Y$, where X is the feature vector $X =< X_1, X_2, X_3 >$. Which of the following classifiers contains sufficient information to allow calculating $P(X_1, X_2, X_3, Y)$? If you answer yes, give a brief sketch of how. If you answer no, state what is missing.

- Gaussian Naive Bayes

  ⋆ *Solution*: Yes, we can estimate $P(X_1, X_2, X_3, Y) = P(Y)P(X_1|Y)P(X_2|Y)P(X_3|Y)$.
- Logistic Regression

  ⋆ *Solution*: No, we cannot compute $P(X)$.
- Linear Regression

  ⋆ *Solution*: No, we cannot compute $P(X)$.

3. (**True or False**, 2 pts) Since classification is a special case of regression, logistic regression is a special case of linear regression.

Solutions: F

Naive Bayes vs logistic regression

1. [3 points] Provide descriptions of Naive Bayes and Logistic Regression algorithms for the dataset above, deriving

(a) P(Y = A|X1...X16) and P(Y = B|X1…X16)
(b) how to classify a new example (i.e. the classification rule)
(c) how to estimate the model parameters

a)

Naïve Bayes

$$P(Y=A|X_1 \dots X_{16}) = \frac{P(Y=A)P(X_1 \dots X_{16}|Y=A)}{P(Y=A)P(X_1 \dots X_{16}|Y=A) + P(Y=B)P(X_1 \dots X_{16}|Y=B)}$$

By conditional independence assumption of Naïve Bayes,

$$P(Y=A|X_1 \dots X_{16}) = \frac{P(Y=A)\prod_{i=1}^{16} P(X_i|Y=A)}{P(Y=A)\prod_{i=1}^{16} P(X_i|Y=A) + P(Y=B)\prod_{i=1}^{16} P(X_i|Y=B)}$$

Similarly for Y=B,

$$P(Y=B|X_1 \dots X_{16}) = \frac{P(Y=B)\prod_{i=1}^{16} P(X_i|Y=B)}{P(Y=A)\prod_{i=1}^{16} P(X_i|Y=A) + P(Y=B)\prod_{i=1}^{16} P(X_i|Y=B)}$$

Logistic Regression

$$P(Y=A|X_1 \dots X_{16}) = \frac{1}{1+\exp\{w_0 + \sum_{i=1}^{16} w_i x_i\}}$$

$$P(Y=B|X_1 \dots X_{16}) = 1 - P(Y=A|X_1 \dots X_{16})$$

Note: there are some alternative answers for this question, for example

$$P(Y=A|X_1 \dots X_{16}) = \frac{1}{1+\exp\{\mathrm{w}^T \mathrm{x}\}}, \text{where w} = \begin{bmatrix} w_0 \\ w_1 \\ \dots \\ w_{16} \end{bmatrix}, x = \begin{bmatrix} 1 \\ x_1 \\ \dots \\ x_{16} \end{bmatrix}$$

and

$$P(Y=A|X_1 \dots X_{16}) = \frac{1}{1+\exp -\{w_0 + \sum_{i=1}^{16} w_i x_i\}}$$

All of them are correct, but one needs to be careful that different definition of logistic regression would end up with different answers in the next several questions.

(b)

Decision rule

Generally speaking, for both cases, if we have $(Y = A|X_1 \dots X_{16}) \geq P(Y = B|X_1 \dots X_{16})$ we will choose A; else, we will choose B. But as a homework question, we prefer more formal form as follows.

$$\arg\max_{y \in \{A,B\}} P(Y = y|X_1 \dots X_{16})$$

Or for logistic regression (if we define $P(Y = A|X_1 \dots X_{16}) = (1 + \exp\{w_0 + \sum_{i=1}^{16} w_i x_i\})^{-1}$)

Since we choose A if

$$\frac{P(Y = A|X_1 \dots X_{16})}{P(Y = B|X_1 \dots X_{16})} = \exp\left\{w_0 + \sum_{i=1}^{16} w_i x_i\right\}^{-1} \geq 1$$

We choose:

$$A, \qquad w_0 + \sum_{i=1}^{16} w_i x_i \leq 0$$

$$B, \qquad w_0 + \sum_{i=1}^{16} w_i x_i > 0$$

Please note that if one defines the logistic regression as $P(Y = A|X_1 \dots X_{16}) = (1 + \exp -\{w_0 +$ *i=116wixi−1*, the result is just the opposite

(c)

Naïve Bayes

$$P(Y=y)_{y=\{A,B\}} = \frac{\#(Y=y)}{\#\text{samples}}$$

$$P(X_i = x_i|Y=y) = \frac{\#(X_i = x_i, Y = y_i)}{\#(Y=y)}$$

Logistic Regression

For purpose of simplicity, we will treat **Y=B as Y=1, and Y=A as Y=0**.
Hence the log likelihood of the data is: (where *m* is the number of training instances)

$$l(w) = \log\left(\prod_{j}^{m} \frac{exp^{\left(y^j\left(w_0+\Sigma_{i=1}^{16} w_i x_i^j\right)\right)}}{1+exp^{\left(w_0+\Sigma_{i=1}^{16} w_i x_i^j\right)}}\right)$$

$$l(w) = \sum_{j}^{m}\left(y^j\left(w_0 + \sum_{i=1}^{16} w_i x_i^j\right) - log\left(1 + exp^{\left(w_0+\Sigma_{i=1}^{16} w_i x_i^j\right)}\right)\right)$$

Maximize the log-likelihood function using gradient descend. The gradients of the log likelihood function are:

$$\frac{\partial l(w)}{\partial w_0} = \sum_{j}^{m}\left(y^j - \frac{exp^{\left(w_0+\Sigma_{i=1}^{16} w_i x_i^j\right)}}{1+ exp^{\left(w_0+\Sigma_{i=1}^{16} w_i x_i^j\right)}}\right)$$

$$\frac{\partial l(w)}{\partial w_0} = \sum_{j}^{m}(y^j - P(Y=1|X=x^j))$$

$$\frac{\partial l(w)}{\partial w_k} = \sum_{j}^{m} \left(y^j x_k^j - \frac{x_k^j exp^{\left(w_0 + \Sigma_{i=1}^{16} w_i x_i^j\right)}}{1 + \ exp^{\left(w_0 + \Sigma_{i=1}^{16} w_i x_i^j\right)}} \right)$$

$$\frac{\partial l(w)}{\partial w_k} = \sum_{j}^{m} x_k^j (y^j - \ P(Y = 1|X = \ x^j))$$

The weight-updating rule is: (repeat until convergence, where α is the step size):

$$w_0^{(t+1)} = w_0^{(t)} + \ \alpha \sum_{j}^{m} (y^j - \ P(Y = 1|X = \ x^j, w^{(t)}))$$

$$w_k^{(t+1)} = w_k^{(t)} + \ \alpha \sum_{j}^{m} x_k^j (y^j - \ P(Y = 1|X = \ x^j, w^{(t)}))$$

Note that if defining the logistic regression as

$$P(Y = A|X_1 = x_1 \dots X_{16} = x_{16}) = \left(1 + \exp - \left\{ w_0 + \sum_{i=1}^{16} w_i x_i \right\} \right)^{-1}$$

Then we will treat **Y=A as Y=1, and Y=B as Y=0.**

3 Generative-Discriminative Classifiers, 20 points

In class, we learned that when Y takes Boolean values and X is a n dimensional vector of $X = \langle X_1, X_2 \ldots X_n \rangle$ continuous variables, where each $X_i, i = 1 \ldots n$ is distributed normally (i.e. $P(X_i|Y = y_k) = N(\mu_{ik}, \sigma_i)$), then Logistic Regression is the discriminative equivalent of Naive Bayes under the Naive Bayes assumptions.

a. (14 points) Consider instead the case where $X = \langle X_1, X_2 \ldots X_n \rangle$ is a vector of *boolean* variables. Prove that even in this case, $P(Y|X)$ follows the same logistic function form (and hence that Logistic Regression is also the discriminative counterpart to a Naive Bayes classifier over boolean features). [Hint: see Exercise 3 in the Mitchell reading on Naive Bayes and Logistic Regression.]

SOLUTION: In the lecture we derived:

$$\begin{aligned} P(Y=1|X) &= \frac{1}{1+\exp\left(\ln \dfrac{P(X|Y=0)P(Y=0)}{P(X|Y=1)P(Y=1)}\right)} \\ &= \frac{1}{1+\exp\left(\ln \dfrac{PY=0)}{P(Y=1)} + \sum_i \ln \dfrac{P(X_i|Y=0)}{P(X_i|Y=1)}\right)} \end{aligned}$$

Prior for $P(Y = 1) = \pi$ and $P(Y = 0) = 1 - \pi$. Also, each X_i has binomial distribution:

$P(X_i|Y = 0) = \theta_{i0}^{X_i}(1 - \theta_{i0})^{(1-X_i)}$

$P(X_i|Y = 1) = \theta_{i1}^{X_i}(1 - \theta_{i1})^{(1-X_i)}$

Inserting this back to the equation:

$$\begin{aligned} P(Y=1|X) &= \frac{1}{1+\exp\left(\ln \dfrac{1-\pi}{\pi} + \sum_i \ln \dfrac{\theta_{i0}^{X_i}(1-\theta_{i0})^{(1-X_i)}}{\theta_{i1}^{X_i}(1-\theta_{i1})^{(1-X_i)}}\right)} \\ &= \frac{1}{1+\exp\left(\ln \dfrac{1-\pi}{\pi} + \sum_i X_i \ln \dfrac{\theta_{i0}}{\theta_{i1}} + (1-X_i) \ln \dfrac{(1-\theta_{i0})}{(1-\theta_{i1})}\right)} \\ &= \frac{1}{1+\exp\left(\ln \dfrac{1-\pi}{\pi} + \dfrac{(1-\theta_{i0})}{(1-\theta_{i1})} + \sum_i X_i \left[\ln \dfrac{\theta_{i0}}{\theta_{i1}} - \ln \dfrac{(1-\theta_{i0})}{(1-\theta_{i1})}\right]\right)} \end{aligned}$$

If we set:

$w_0 = \ln \frac{1-\pi}{\pi} + \sum_i \ln \frac{(1-\theta_{i0})}{(1-\theta_{i1})}$ and

$w_i = \ln \frac{\theta_{i0}}{\theta_{i1}} - \ln \frac{(1-\theta_{i0})}{(1-\theta_{i1})}$

then we can reach:

$$P(Y=1|X) = \frac{1}{1+\exp\left(\sum_i w_i X_i\right)}$$

which is equivalent to the LR formulation.

b. (2 points) Suppose the data satisfies the conditional independence assumption of Naive Bayes. As the number of training examples approaches infinity, which classifier produces better results, NB or LR? Justify your answer in one sentence.

SOLUTION: Under conditional independence assumptions, we showed that Logistic regression is discriminative counterpart of Naive Bayes. Therefore, if the data satisfies CI assumptions, Naive Bayes and Logistic Regression will produce equivalent results.

c. (2 points) Suppose the data does not satisfy the conditional independence assumption of Naive Bayes. As the number of training examples approaches infinity, which classifier produces better results, NB or LR? Justify your answer in one sentence.

SOLUTION: Logistic Regression will produce better results, since it doesn't assume that data satisfies conditional independence.

d. (2 points) Is it possible to calculate P(X) from the parameters estimated by Logistic Regression? Explain.

SOLUTION: No it is not, LR is a discriminative classifier, that estimates $P(Y|X)$, not $P(X|Y)$. In order to calculate P(X), we need to know $P(X|Y)$.

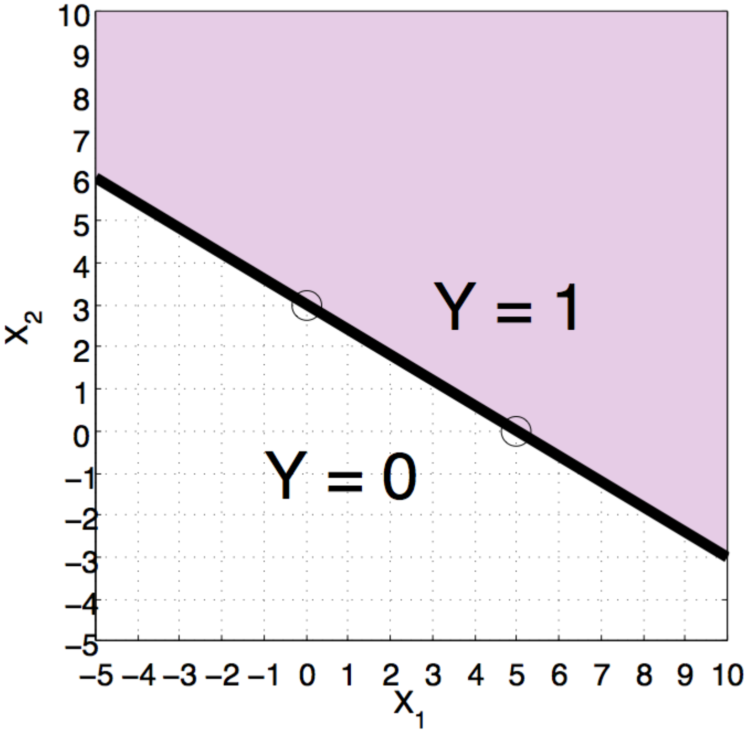
4 Logistic Regression [8 Points]

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

$$\textbf{Predict } Y = 1 \textbf{ iff } w_1X_1 + w_2X_2 + w_0 \geq 0 \textbf{ and } Y = 0 \textbf{ otherwise}$$

where $w_1 = 3$, $w_2 = 5$, and $w_0 = -15$.

1. **[Points: 6 pts]** Plot the decision boundary and label the region where we would predict $Y = 1$ and $Y = 0$.



★ **SOLUTION:** See above figure.

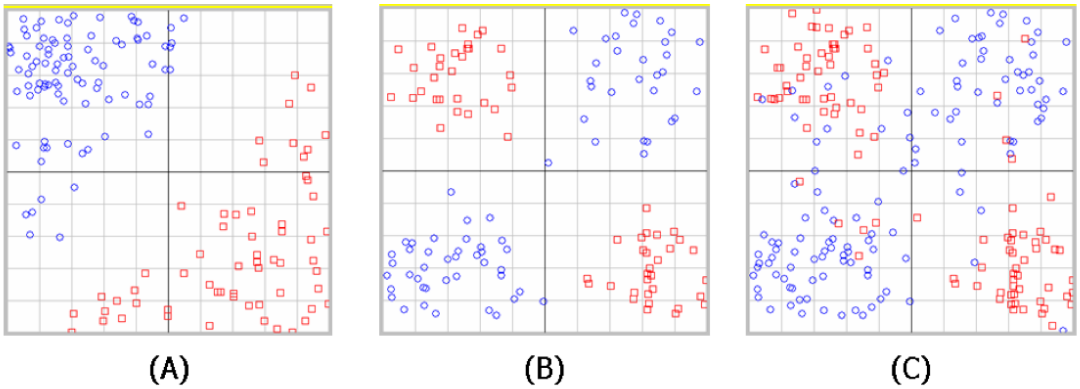
2. **[Points: 2 pts]** Suppose that we learned the above weights using logistic regression. Using this model, what would be our prediction for $P(Y = 1 \mid X_1, X_2)$? (You may want to use the sigmoid function $\sigma(x) = 1/(1 + \exp(-x))$.)

$$\mathbf{P}\left(Y = 1 \mid X_1, X_2\right) =$$

★ **SOLUTION:**

$$\mathbf{P}\left(Y = 1 \mid X_1, X_2\right) = \frac{1}{1 + \exp^{-(3X_1 + 5X_2 - 15)}}$$

Part IV: Understanding the performance of previously-studied learning methods (10 pts)



Consider the three datasets (A), (B), and (C) above, where the circles are positive examples (with two numeric features – the coordinate of the circle in 2D space) and the squares are negative examples. Now consider the following four learning methods: Naive bayes (NB), logistic regression (LR), decision trees with pruning (DT+p) , and decision trees without pruning (DT-p).

1. (5pts) For each of the three datasets, list the learning system(s) that you think will perform well, and say why.

   There was some freedom here as to how many methods you listed, and in which order, etc. The general idea, though, is that (A) is linearly separable and should be learnable by NB, LR, DT-p and DT+p. (B) is not linearly separable (and so can't be learned by NB or LR), but can be perfectly classified with two lines, ie, a decision tree like DT-p or DT+p. Finally, (C) is a noisy version of (B) and so we need DT+p's pruning to deal with the noise (DT-p would overfit the noise).

2. (5pts) For each of the three datasets, list the learning system(s) that you think will perform poorly, and say why.

   As above, all methods should do well on (A), NB and LR should fail (B) due to decision boundary form, and NB, LR and DT-p should fail (C) due to noise.

Regularized Logistic Regression

To train a logistic regression model for data points in figure below, we apply regularization to all the weights. The maximum conditional log-likelihood estimation is given by the following expression:

$$\max \sum_{i=1}^{n} \log(P(y_i \mid x^i, w_0, w_1, w_2)) - C_0 w_0^2 - C_1 w_1^2 - C_2 w_2^2$$

where i denotes the index of a training example.

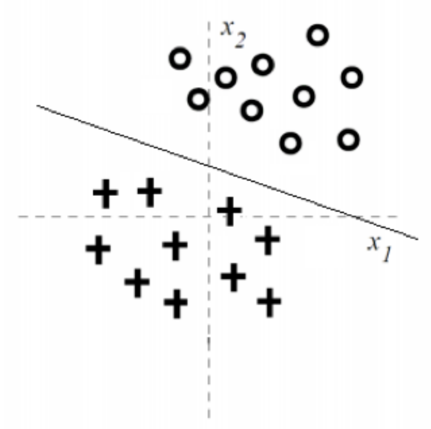


Figure 2: The 2-dimensional labeled training set, where '+' corresponds to class y=1 and 'O' corresponds to class y=0. The straight line corresponds to the trained decision boundary with no regularization.

How many errors would the resulting classifier have for each of the following settings of the regularization values:

A. No penalty $C_0 = C_1 = C_2 = 0$

B. Penalize w_0 $(C_0 = \text{inf}, C_1 = C_2 = 0)$

C. Penalize w_1 $(C_1 = \text{inf}, C_0 = C_2 = 0)$

D. Penalize w_2 $(C_2 = \text{inf}, C_0 = C_1 = 0)$

Solution:

(1) 0

(2) 2

(3) 0

7 Logistic Regression (10%)

We consider the following models of logistic regression for a binary classification with a sigmoid function $g(z) = \frac{1}{1+e^{-z}}$:

- Model 1: $P(Y = 1 \mid X, w_1, w_2) = g(w_1 X_1 + w_2 X_2)$
- Model 2: $P(Y = 1 \mid X, w_1, w_2) = g(w_0 + w_1 X_1 + w_2 X_2)$

We have three training examples:

$$\begin{array}{lll} x^{(1)} = [1,1]^T & x^{(2)} = [1,0]^T & x^{(3)} = [0,0]^T \\ y^{(1)} = 1 & y^{(2)} = -1 & y^{(3)} = 1 \end{array}$$

1. (5%) Does it matter how the third example is labeled in Model 1? i.e., would the learned value of $\mathbf{w} = (w_1, w_2)$ be different if we change the label of the third example to -1? Does it matter in Model 2? Briefly explain your answer. (Hint: think of the decision boundary on 2D plane.)

   (sol.) It does not matter in Model 1 because $x^{(3)} = (0,0)$ makes $w_1 x_1 + w_2 x_2$ always zero and hence the likelihood of the model does not depend on the value of $\mathbf{w}$. But it does matter in Model 2.

2. (5%) Now, suppose we train the logistic regression model (Model 2) based on the n training examples $x^{(1)}, \ldots, x^{(n)}$ and labels $y^{(1)}, \ldots, y^{(n)}$ by maximizing the penalized log-likelihood of the labels:

$$\sum_i \log P(y^{(i)} \mid x^{(i)}, \mathbf{w}) - \frac{\lambda}{2}||\mathbf{w}||^2 = \sum_i \log g(y^{(i)} \mathbf{w}^T x^{(i)}) - \frac{\lambda}{2}||\mathbf{w}||^2$$

   For large λ (strong regularization), the log-likelihood terms will behave as linear functions of $\mathbf{w}$.

$$\log g(y^{(i)} \mathbf{w}^T x^{(i)})) \approx \frac{1}{2} y^{(i)} \mathbf{w}^T x^{(i)}$$

   Express the penalized log-likelihood using this approximation (with Model 1), and derive the expression for MLE $\hat{\mathbf{w}}$ in terms of λ and training data $\{x^{(i)}, y^{(i)}\}$. Based on this, explain how $\mathbf{w}$ behaves as λ increases. (We assume each $x^{(i)} = (x_1^{(i)}, x_2^{(i)})^T$ and $y^{(i)}$ is either 1 or -1)

   (sol.)

$$\log l(\mathbf{w}) \approx \sum_i \frac{1}{2} y^{(i)} \mathbf{w}^T x^{(i)} - \frac{\lambda}{2}||w||^2$$

$$\frac{\partial}{\partial w_1} \log l(\mathbf{w}) \approx \frac{1}{2} \sum_i y^{(i)} x_1^{(i)} - \lambda w_1 = 0$$

$$\frac{\partial}{\partial w_2} \log l(\mathbf{w}) \approx \frac{1}{2} \sum_i y^{(i)} x_2^{(i)} - \lambda w_2 = 0$$

$$\therefore \quad \mathbf{w} = \frac{1}{2\lambda} \sum_i y^{(i)} \mathbf{x}^{(i)}$$

2. The correspondence between logistic regression and Gaussian Naïve Bayes (with identity class covariances) means that there is a one-to-one correspondence between the parameters of the two classifiers.

   False: Each LR model parameter corresponds to a whole set of possible GNB classifier parameters, there is no one-to-one correspondence because logistic regression is discriminative and therefore doesn't model $P(X)$, while GNB does model $P(X)$.

3 Logistic Regression [18 pts]

We consider here a discriminative approach for solving the classification problem illustrated in Figure 1.

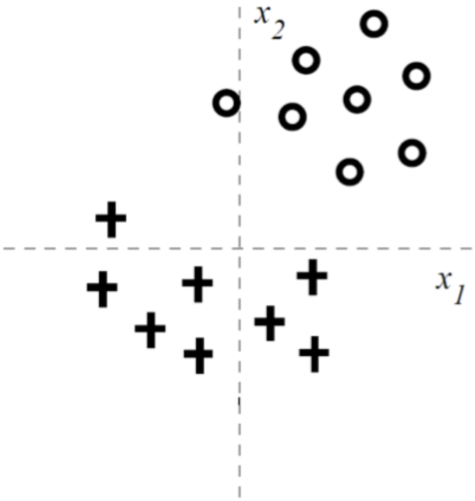


Figure 1: The 2-dimensional labeled training set, where '+' corresponds to class y=1 and 'O' corresponds to class $y = 0$.

1. We attempt to solve the binary classification task depicted in Figure 1 with the simple linear logistic regression model

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

Notice that the training data can be separated with *zero* training error with a linear separator.

Consider training *regularized* linear logistic regression models where we try to maximize

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

for very large C. The regularization penalties used in penalized conditional log-likelihood estimation are $-Cw_j^2$, where $j = \{0, 1, 2\}$. In other words, only one of the parameters is regularized in each case. Given the training data in Figure 1, how does the training error change with regularization of each parameter w_j? State whether the training error increases or stays the same (zero) for each w_j for very large C. Provide a brief justification for each of your answers.

(a) By regularizing w_2 **[2 pts]**

SOLUTION: 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.

(b) By regularizing w_1 **[2 pts]**

SOLUTION: 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 and the training data can be separated with *zero* training error with a horizontal linear separator.

(c) By regularizing w_0 **[2 pts]**

SOLUTION: 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.

2. 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 problem in Figure 1 and the same linear logistic regression model $P(y = 1|\vec{x}, \vec{w}) = g(w_0 + w_1x_1 + w_2x_2)$.

(a) **[3 pts]** As we increase the regularization parameter C which of the following scenarios do you expect to observe? (Choose only one) Briefly explain your choice:

( ) First w_1 will become 0, then w_2.

( ) First w_2 will become 0, then w_1.

( ) w_1 and w_2 will become zero simultaneously.

( ) None of the weights will become exactly zero, only smaller as C increases.

SOLUTION: 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 labels that we can achieve with a non-zero w_2.

(b) **[3 pts]** 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. (Note that the number of points from each class is the same.) (You can give a range of values for w_0 if you deem necessary).

SOLUTION: For very large C, we argued that both w_1 and w_2 will go to zero. Note that when $w_1 = w_2 = 0$, the log-probability of labels becomes a finite value, which is equal to n log(0.5), i.e. $w_0 = 0$. In other words, $P(y = 1|\vec{x}, \vec{w})$=$P(y = 0|\vec{x}, \vec{w})$=$0.5$. We expect so because the number of elements in each class is the same and so we would like to predict each one with the same probability, and w_0=0 makes $P(y = 1|\vec{x}, \vec{w})$=$0.5$.

(c) **[3 pts]** Assume that we obtain more data points from the '+' class that corresponds to y=1 so that the class labels 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. (You can give a range of values for w_0 if you deem necessary).

SOLUTION: For very large C, we argued that both w_1 and w_2 will go to zero. With unbalanced classes where the number of '+' labels are greater than that of 'o' labels, we want to have $P(y = 1|\vec{x}, \vec{w}) > P(y = 0|\vec{x}, \vec{w})$. For that to happen the value of w_0 should be greater than zero which makes $P(y = 1|\vec{x}, \vec{w}) > 0.5$.