CS6140

Final exam

April 20, 2017

Time: 1 hour 40min

Name (please print): ______________________________________

Show all your work and calculations. Partial credit will be given for work that is partially correct. Points will be deducted for false statements, even if the final answer is correct. Please circle your final answer where appropriate.

This exam is closed-book. You may consult one page with your hand-written notes. Calculators are permitted.

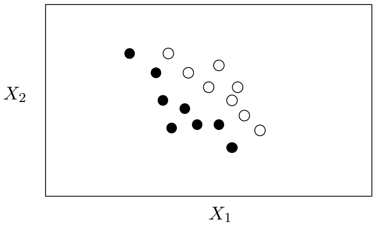
Honor code: I promise not to cheat on this exam. I will neither give nor receive any unauthorized assistance. I will not to share information about the exam with anyone who may be taking it at a different time. I have not been told anything about the exam by someone who has taken it earlier.

Signature: ____ Date: ____________

Question	Possible Points	Actual Points
1	12	
2	18	
3	18	
4	6	
5	30	
6	36	
Total	120	

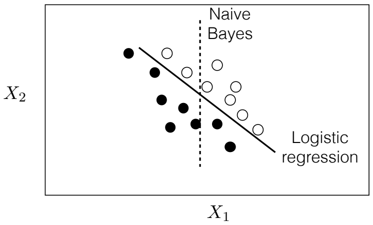
1. In this question we compare Naïve Bayes and logistic regression.

   (a) **(6 pts)** Consider the dataset below. Would you prefer logistic regression or Naïve Bayes for classification? On the figure sketch the decision boundary for each method and justify your answer.

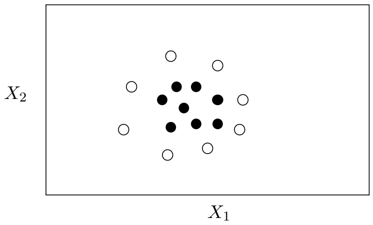


Answer:

Logistic regression. Logistic regression produces a linear decision boundary, that does not need to be parallel to the axes. The assumption of independence between dimensions of Naïve Bayes will result in a decision boundary that is parallel to one of the axes.

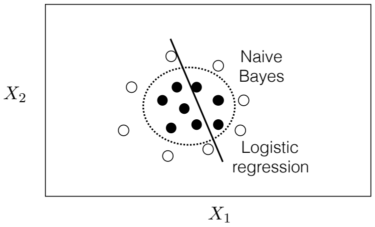


   (b) **(6 pts)** Consider the dataset below. Would you prefer logistic regression or Naïve Bayes for classification? On the figure sketch the decision boundary for each method and justify your answer.



Answer:

Naïe Bayes. Since Naïe Bayes does not assume Normal distributions, the decision boundary can be of any form (including a circle), provided that the underlying distributions do not have a correlation between the predictors. Logistic regression will not be able to separate this type of pattern, even if statistical interactions are included.



2. Consider the following data. X is the only predictor, and $Y \in \{0, 1\}$ is the response.

X	8	11	12	12	12	15	15	15	18	23
Y	0	0	0	0	1	0	0	1	1	1

(a) **(6 pts)** A decision tree is learned from this data by minimizing information gain. Circle the correct answer and explain.

True **False**

Answer:

False. Our goal is to make nodes as pure as possible based on $i(t)$. Therefore we maximize the information gain.

(b) **(6 pts)** What is the optimal split on X? Use Gini index as the impurity measure.

Answer:

The Gini index is:

$$i(t) = \sum_j p(j|t)(1 - p(j|t))$$

Since we only have two classes, $i(t) = p(0|t)p(1|t)$.

The Gini Index of root: $\frac{4}{10} \cdot \frac{6}{10} = \frac{24}{100} = 0.24$

Now we calculate the Gini indexes:

$X = 9 : i(t) = \frac{1}{10}(1 \times 0) + \frac{9}{10}(\frac{4}{9} \times \frac{5}{9}) = \frac{2}{9}$

$X = 11.5 : i(t) = \frac{2}{10}(1 \times 0) + \frac{8}{10}(\frac{4}{8} \times \frac{4}{8}) = \frac{2}{10} = \frac{1}{5}$

$X = 12 : i(t) = \frac{5}{10}(\frac{1}{5} \times \frac{4}{5}) + \frac{5}{10}(\frac{2}{5} \times \frac{3}{5}) = \frac{4}{50} + \frac{6}{50} = \frac{1}{5}$

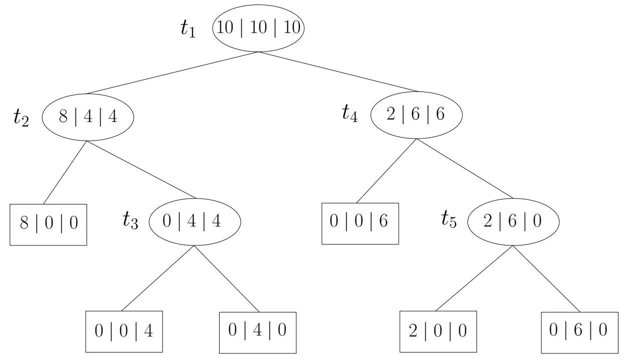
$X = 13.5 : i(t) = \frac{8}{10}(\frac{2}{8} \times \frac{6}{8}) + \frac{2}{10}(1 \times 0) = \frac{3}{20}$

$X = 18 : i(t) = \frac{9}{10}(\frac{3}{9} \times \frac{6}{9}) + \frac{1}{10}(1 \times 0) = \frac{18}{90} = \frac{1}{5}$

$X = 20.5 : i(t) = \frac{10}{10}(\frac{4}{10} \times \frac{6}{10}) + 0 = 0.24$

We can see that $X = 13.5$ has the lowest number ($\frac{3}{20}$), hence The impurity reduction is: $\Delta i = \frac{24}{100} - \frac{3}{20} = 0.09$

(c) **(6 pts)** The tree given below, denoted by T_{max}, has been constructed on the training sample for a 3-class classification problem. In each node, the number of observations with class 0 is given in the left part, the number of observations with class 1 in the middle part, and the number of observations with class 2 in the right part. The leaf nodes are drawn as rectangles.



Find the first internal node of the tree that needs to be pruned. Specifically, use resubstitution error, fill in the table below, circle the α and the label of the node that will be selected first for cost-complexity pruning.

	t_1	t_2	t_3	t_4	t_5
α					

Answer:

	t_1	t_2	t_3	t_4	t_5^*
α	$\frac{2}{15}$	$\frac{2}{15}$	$\frac{2}{15}$	$\frac{2}{15}$	$\frac{1}{15}$

The starred node corresponds to the smallest α, and should be pruned first.

$\alpha = \frac{R(t)-R(T_t)}{|T_t|-1}$

$\alpha(t_1) = \frac{\frac{20}{30}-0}{6-1} = \frac{2}{15}$

$\alpha(t_2) = \frac{\frac{16}{30}\cdot\frac{8}{16}-0}{3-1} = \frac{2}{15}$

$\alpha(t_3) = \frac{\frac{8}{30}\cdot\frac{4}{8}-0}{2-1} = \frac{2}{15}$

$\alpha(t_4) = \frac{\frac{14}{30}\cdot\frac{8}{14}-0}{3-1} = \frac{2}{15}$

$\alpha(t_5) = \frac{\frac{8}{30}\cdot\frac{2}{8}-0}{2-1} = \frac{1}{15}$

3. In this question we consider boosting.

   (a) **(6 pts)** True or false: Cross validation can be used to select the number of iterations in boosting. Circle the correct answer and explain.

   True **False**

   Answer:

   True. The number of iterations in boosting controls the complexity of the model. Therefore, a model selection procedure such as cross validation can be used to select the appropriate model complexity and reduce the possibility of overfitting.

   (b) **(6 pts)** True or false: The coefficients α assigned to the mis-classified observations by AdaBoost are always non-negative. Circle the correct answer and explain.

   True **False**

   Answer:

   True. From the update equation, the weights are $\alpha_m = log(\frac{1-err_m}{err_m})$*, where* err_m *is defined as*

$$err_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} w_i}$$

   Since $0 < err_m < 1$*, the weights are always positive.*

(c) **(6 pts)** True or false: In AdaBoost, weights of all the misclassified examples increase by the same multiplicative factor. Circle the correct answer and explain.

True **False**

Answer:

True. Follows from the update equation. We have $\alpha_m = log(\frac{1-err_m}{err_m})$ *and* err_m *is defined as,*

$$err_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} w_i}$$

The weights are then updated according to

$$w_{i+1} = w_i.exp[\alpha_m.I(y_i \neq G_m(x_i))], i = 1, , 2, ..., N$$

The indicator function is either one or zero. For the misclassified examples we have $w_{i+1} = w_i e^{\alpha_m}$*. Since at each step* m*,* α_m *is constant after we evaluate it, then* e^{α_m} *would be the same multiplicative factor for all the misclassified examples.*

4. **(6 pts)** True or False: Suppose our dataset has one predictor X and a continuous response Y. Locally weighted regression can be identical to ordinary least-squares regression, given a suitable kernel, for any data set. Circle the correct answer and explain.

True **False**

Answer:

True. Use the kernel $K(d) = c$ *for a very large c that covers the range of* X*. Then all examples are weighted equally and the algorithm behaves exactly like ordinary unweighted regression.*

5. Assume that your are training an SVM classifier.

(a) **(6 pts)** Recall parameters of the linear SVM classifier are learned by minimizing

$$\min_{\beta,\beta_0} \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{N} \xi_i \text{ s.t. } \xi_i \geq 0,\ y_i(x_i'\beta + \beta_0) \geq 1 - \xi_i,\ i = 1, \ldots, N$$

True or false: Very large values of C (i.e., $C \to \infty$) correspond to the large margin. Circle the correct answer and explain.

True **False**

Answer:

False. Large cost → few points inside the margin → small margin.

(b) **(6 pts)** True or false: When the data is not completely linearly separable, the linear SVM *without slack variables* returns $\beta = 0$. Circle the correct answer and explain.

True **False**

Answer:

False. If the data are not linearly separable, and slack variables are not allowed, then there is no solution.

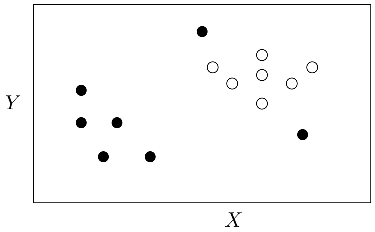
(c) **(6 pts)** True or false: Consider a point that is correctly classified and is distant from the decision boundary. If we remove this point, the decision boundary by both SVM and by logistic regression will be affected. Circle the correct answer and explain.

True **False**

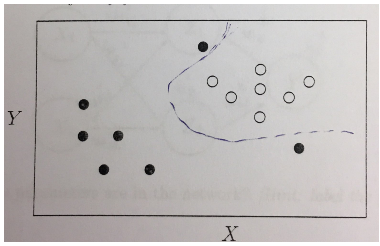
Answer:

False. Logistic regression will be affected, but not SVM. SVM is built on support vectors on the margin and a point far from the boundary will not affect the margin. Logistic regression will use the information of all the points in the data so removing a point will affect it.

(d) **(6 pts)** Assume that you are training an SVM classifier with a quadratic kernel (i.e., the kernel function is a polynomial of degree 2). You are given the data set in Figure below. Where would the decision boundary be for very large values of C (i.e., $C \to \infty$)? Sketch it on the figure and justify your answer.

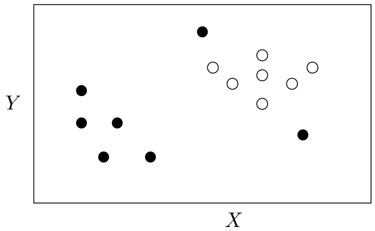


Answer:

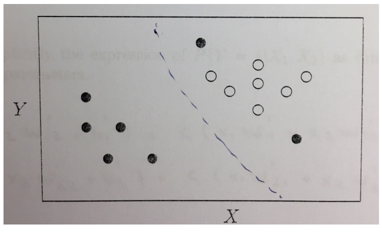


As $C \to \infty$, SVM don't allow miss-classification. The two classes will be perfectly separated by decision boundary. The decision boundary does not need to be linear, since this is a quadratic kernel.

(e) **(6 pts)** Assume that you are training an SVM classifier with a quadratic kernel (i.e., the kernel function is a polynomial of degree 2). You are given the data set in Figure below. Where would the decision boundary be for very small values of C (i.e., $C \approx 0$)? Sketch it on the figure and justify your answer.

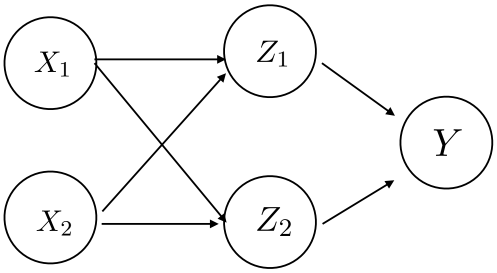


Answer:

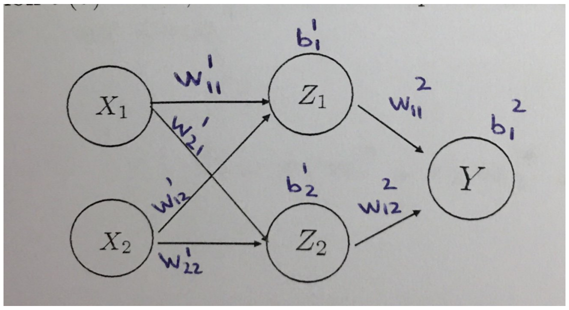


As $C \approx 0$, the constraint disappears from the minimization, and large values of slack variables are allowed. Therefore, points can be on the margin (or even other side of decision boundary), resulting in missclassification. The unconstrained optimization results in a nearly linear boundary.

6. Consider a neural network for binary classification, as shown below. The network uses a linear activation function $\sigma(v) = c \cdot v$, and a softmax output function.



(a) **(6 pts)** How many parameters are in the network? *[Hint: label the graph]*
Answer:



Number of input nodes = 2 + 1(bias)
Number of nodes in hidden layer = 2 + 1(bias)
Number of output nodes = 1
Total parameters $= 3 \times 2 + 3 \times 1 = 9$

(b) **(6 pts)** Write explicitly the expression of $P\{Y = 1|X_1, X_2\}$ as function of the input variables and the parameters.

Answer:

$Z_1 = \sigma(X_1W^1_{11} + X_2W^1_{12} + b^1_1) = C(X_1W^1_{11} + X_2W^1_{12} + b^1_1)$
$Z_2 = \sigma(X_1W^1_{21} + X_2W^1_{22} + b^1_2) = C(X_1W^1_{21} + X_2W^1_{22} + b^1_2)$

$$\begin{aligned} Y' &= \sigma(Z_1W^2_{11} + Z_2W^2_{12} + b^2_1) = C(Z_1W^2_{11} + Z_2W^2_{12} + b^2_1) \\ &= C[CW^2_{11}(X_1W^1_{11} + X_2W^1_{12} + b^1_1) + CW^2_{12}(X_1W^1_{21} + X_2W^1_{22} + b^1_2) + b^2_1] \\ &= C^2(X_1W^1_{11}W^2_{11} + X_2W^1_{12}W^1_{11} + W^2_{11}b^1_1 + X_1W^1_{21}W^2_{12} + X_2W^1_{22}W^2_{12} + b^1_2W^2_{12}) + Cb^2_1 \\ &= X_1(W^1_{11}W^2_{11}C^2 + W^1_{21}W^2_{12}C^2) + X_2(W^1_{12}W^2_{11}C^2 + W^1_{22}W^2_{12}C^2) + (C^2W^2_{11}b^1_1 + C^2b^1_2W^2_{12} + Cb^2_1) \end{aligned}$$

We will assign,
$W^*_1 = W^1_{11}W^2_{11}C^2 + W^1_{21}W^2_{12}C^2$

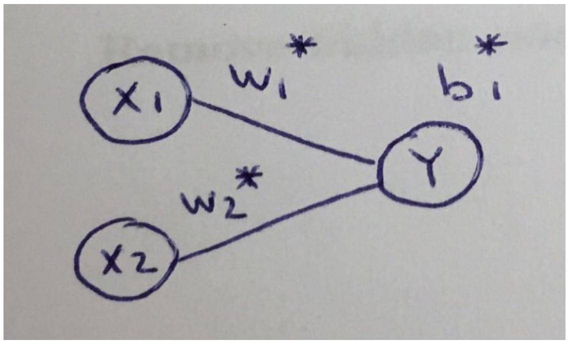
$W^*_2 = W^1_{12}W^2_{11}C^2 + W^1_{22}W^2_{12}C^2$

$b^* = C^2W^2_{11}b^1_1 + C^2b^1_2W^2_{12} + Cb^2_1$

Then we have,
$Y' = X_1W^*_1 + X_2W^*_2 + b^*,$
$Y = softmax(Y')$

(c) **(6 pts)** Below draw a neural net with no hidden layer, which is equivalent to the neural net in the figure above. Define the parameters of the new network in terms of the parameters of the original network.

Answer:



We have, $Y' = X_1W_1^* + X_2W_2^* + b_1^*$, $Y = softmax(Y')$. *We need to assign values to* W_1^*, W_2^*, b_1^* *so that* Y' *be equal to* Y' *in part (b). So, we have,*

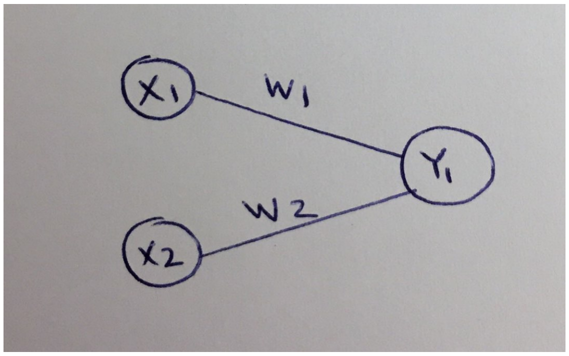
$$W_1^* = W_{11}^1 W_{11}^2 C^2 + W_{21}^1 W_{12}^2 C^2$$

$$W_2^* = W_{12}^1 W_{11}^2 C^2 + W_{22}^1 W_{12}^2 C^2$$

$$b_1^* = C^2 W_{11}^2 b_1^1 + C^2 b_2^1 W_{12}^2 + C b_1^2$$

(d) **(6 pts)** Suppose that instead of neural network you would like to use a simple logistic regression with two predictors. Draw the neural network that will represent logistic regression with two predictors. Label all the components of the network.

Answer:



$Y_1 = \sigma(X_1W_1 + X_2W_2 + b_1)$
$B_0 = b_1, B_1 = W_1, B_2 = W_2$
We have,

$$\hat{f}(X_1, X_2) = \frac{e^{B_0 + B_1X_1 + B_2X_2}}{1 + e^{B_0 + B_1X_1 + B_2X_2}}$$

(e) **(6 pts)** Suppose that you learn the network on a training dataset, and obtain the misclassification error of 1%. You then evaluate the performance on the model selection dataset, and obtain the misclassification error of 10%. What would be your next step in improving the performance? Circle the correct answer and explain.

Remove hidden nodes **Add hidden nodes**

Answer:

Remove hidden nodes

The network overfits the data because in training set we have 1% *missclassification error which means that it classifies training set almost perfectly. But it does not do the classification perfectly for the test set, hence we have to make it less non linear by removing hidden layer and making the network simpler.*

(f) **(6 pts)** Suppose that you learn the network on a training dataset, and obtain the misclassification error of 9%. You then evaluate the performance on the model selection dataset, and obtain the misclassification error of 10%. What would be your next step in improving the performance? Circle the correct answer and explain.

Remove hidden nodes **Add hidden nodes**

Answer:

Add hidden nodes

In this case the error rates on training and test set is almost the same, which means that if we examine another data set we do not expect a huge change in the error rate. We are sure that we can get a better result on the training set but we are only afraid of overfitting, so we add more hidden nodes and then again check the results for the training and test set to decide what to do. We have to be careful to avoid the situation that happened in part (e).