Solutions

10-701 Machine Learning **Name:** _____
Fall 2018
Midterm Exam
10/22/2018
Time Limit: 120 minutes **Andrew ID** _____

Instructions:

- Fill in your name and Andrew ID above. Be sure to write neatly, or you may not receive credit for your exam.
- This exam contains 25 pages (including this cover page and 2 blank pages at the end). There are 8 sections.
  The total number of points is 100.
- Clearly mark your answers in the allocated space. If your solution is messy please highlight the answer so it is clearly visible. If you have made a mistake, cross out the invalid parts of your solution, and circle the ones which should be graded.
- Look over the exam first to make sure that none of the 25 pages are missing. The problems are of varying difficulty, so you may wish to pick off the easy ones first.
- You may use a scientific calculator for this exam, however no other electronic devices can be used.
- Please write all answers in pen.
- You have 120 minutes to complete the exam. Good luck!

Topic	**Section**	**Points**
MLE	1	12
Decision Trees	2	14
Naive Bayes	3	10
Classification Algorithms	4	14
Neural Networks	5	10
SVM	6	16
Boosting	7	12
Hierarchical Clustering and KNN	8	12
—	**Total**	100

1 MLE [12 pts]

Below is the probability density function for the Rayleigh distribution.

$$f(x;\theta) = \frac{x}{\theta^2} exp\left(\frac{-x^2}{2\theta^2}\right)$$

1. **[4 pts]** Suppose Lord Rayleigh gathered data consisting of $X_1, X_2, \ldots, X_n$, which he deems are iid $Rayleigh(\theta)$ random variables and have the following the likelihood function, $lik(\theta) = f(x_1, \ldots, x_n \mid \theta)$ Find $\hat{\theta}_{MLE}$, the maximum likelihood estimate of Rayleigh's parameter.

   $lik(\theta) = \prod_{i=1}^{n} [\frac{x_i}{\theta^2} exp\left(\frac{-x_i^2}{2\theta^2}\right)]$

   $log(lik(\theta)) = [\sum_1^n log(x_i)] - 2nlog(\theta) - \frac{1}{\theta^2}\sum_1^n [x_i^2/2]$

   $\hat{\theta}_{MLE} = (\frac{1}{n}\sum_1^n [x_i^2/2])^{1/2}$

2. **[4 pts]** Suppose Lord Rayleigh lets $B = A^2 \sim Exponential(\lambda)$ and $A \sim Rayleigh(\theta)$, such that $\theta = \frac{1}{\sqrt{2\lambda}}$. Recall the probability density function of an Exponential random variable is given by $f(x;\lambda) = \lambda e^{-\lambda x}$. Suppose $\lambda \sim Gamma(\alpha, \beta)$, where the probability density function for a gamma distribution is $f(x;\alpha,\beta) = \frac{\beta^\alpha x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)}$. Recall that if a posterior and prior distribution are from the same family of distributions, we call the prior a conjugate prior of the likelihood. Show that if $x \sim Exponential(\lambda)$ and $\lambda \sim Gamma(\alpha, \beta)$, then $P(\lambda|x) \sim Gamma(\alpha', \beta')$; that is, find α' and β' in terms of α and β.

   $\alpha' =$_____

   $\beta' =$_____

   $\alpha' = \alpha + n$

   $\beta' = \beta + \sum_1^n x_i$

3. **[4 pts]** Determine which of the following represents $\hat{\lambda}_{MAP}$, the MAP estimate of λ, in terms of α and β.

   ○ $\frac{\beta+\sum_1^n x_i}{\alpha+n-1}$

   ○ $\frac{\alpha+n}{\beta+\sum_1^n x_i}$

- ◯ $\frac{\alpha+n-1}{\beta+\sum_1^n x_i}$
- ◯ $\frac{\beta+\sum_1^n x_i}{\alpha+n}$

3rd choice. $\frac{\alpha+n-1}{\beta+\sum_1^n x_i}$

Umang

2 Decision Trees [14 pts]

1. Suppose that (X_1, X_m) are categorical input attributes and Y is the categorical output attribute. We now plan to learn a decision tree without pruning, using the standard ID3 algorithm.

   (a) **[2 pts] True of false?** If X_i and Y are independent in the distribution that generated this dataset, then X_i will not appear in the decision tree.

   ◯ True

   ◯ False

   False (because the attribute may become relevant further down the tree when the records are restricted to some value of another attribute)

   (b) **[2 pts] True of false?** If the mutual information between Y and X_i is 0 according to the values of entropy and conditional entropy computed from the data, then X_i will not appear in the decision tree.

   ◯ True

   ◯ False

   False for same reason

2. You are in the mood to play tennis. However, you are not sure if your friend would be willing to play with you. You want to construct a decision tree that you will use to determine whether your friend is in the mood to play tennis or not. You know that willingness of your friend to play tennis is dependent on three binary attributes: Weather (which takes values Sunny/Rainy), whether he worked out that day (which takes values Yes/No), and whether he is injured (which takes values Yes/No). Here is the information for the last 8 times you tried contacting your friend to play tennis.

Weather	Worked Out?	Injured?	Played?
Sunny	Yes	Yes	No
Sunny	No	Yes	Yes
Sunny	No	No	Yes
Sunny	Yes	Yes	No
Rainy	Yes	Yes	Yes
Rainy	No	Yes	No
Rainy	Yes	No	Yes
Rainy	No	Yes	No

   (a) **[2 pts]** What is the initial entropy in the Played variable in the training dataset?

$H(played?) = H(4/8, 4/8) = H(0.5, 0.5) = 1bit$

(b) **[4 pts]** At the root of the tree, what is the mutual information offered about the label by each of the three attributes? Which attribute would ID3 place at the root?

$H(played?|weather) = 4/8H(2/4, 2/4) + 4/8H(2/4, 2/4) = 1$
$\implies I(played?; weather) = 0$
$H(played?|workedout?) = 4/8H(2/4, 2/4) + 4/8H(2/4, 2/4) = 1$
$\implies I(played?; workedout?) = 0$
$H(played?|injured) = 3/4log(3) - 1/2$
$\implies I(played?; injured?) = 3/2 - 3/4log(3)$

So Injured

(c) **[4 pts]** Will ID3 come up with the smallest possible tree (tree with the fewest nodes) that is consistent with this training data? If so, explain why. If not, show a smaller tree.

No. The smallest tree will compute an XOR between Weather and Worked out, using 3 nodes only (Weather in root and Worked out in both children, or vice versa). ID3s output will take more than 3 nodes.

Adithya

3 Naive Bayes [10 pts]

Recall that a Naive Bayes classifier assumes the features $x_1, x_2, \ldots$ are conditionally independent given the class label y, so that:

$$P(y|X = (x_1, \ldots, x_n)) \propto P(X, y) = P(X|y) \cdot P(y) = P(y) \cdot \prod_i P(x_i|y),$$

and consequently its prediction can be written as:

$$\arg\max_y P(y|X) = \arg\max_y P(y) \cdot \prod_i P(x_i|y). \tag{1}$$

Now, consider the following non-linear classifier with the classification rule of the following form:

$$\arg\max_y S(w_0 + \sum_{i=1}^{n} w_{y,i} X_i), \tag{2}$$

where S is the sigmoid function.

1. (a) **[6 pts]** Suppose the input features are binary i.e. $x_i \in \{0, 1\}$. Under this setting, rewrite the classification rule of the Naive Bayes classifier in Eqn. (1) to have the same form as that of the non-linear classifier in Eqn. (2) (**Hint:** Use the fact that X_i is binary).

$\arg\max_y P(y) \cdot \prod_i P(x_i|y) = \arg\max_y \log P(y) + \sum_{i=1}^{n} \log P(x_i|y)$
$= \arg\max_y \log P(y) + \sum_{i=1}^{n} [x_i \log P(X_i = 1|y) + (1 - x_i) \log P(X_i = 0|y))]$
$= \arg\max_y \log P(y) + \sum_{i=1}^{n} \log P(X_i = 0|y) + \sum_{i=1}^{n} x_i \log \frac{P(X_i=1|y)}{P(X_i=0|y)}$
$= \arg\max_y S\left(\log P(y) + \sum_{i=1}^{n} \log P(X_i = 0|y) + \sum_{i=1}^{n} x_i \log \frac{P(X_i=1|y)}{P(X_i=0|y))} \right)$

because S is a monotonically increasing function.

(b) **[4 pts]** Express the weights $w_0, w_{y,i}$ (for $i = 1, \ldots, n$) in terms of the Naive Bayes probabilities by using $P(y)$ and $P(x_i|y)$. Assume that all Naive Bayes probabilities are non-zero.

$$w_0 = \log P(y) + \sum_{i=1}^{n} \log P(X_i = 0|y)$$

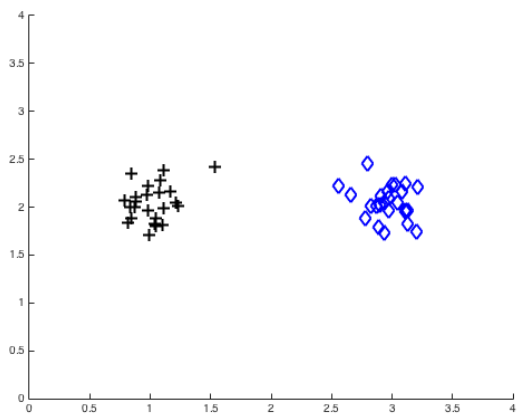
$$w_{y,i} = \log \frac{P(X_i = 1|y)}{P(X_i = 0|y)} \qquad \text{For } i = 1, \ldots n$$

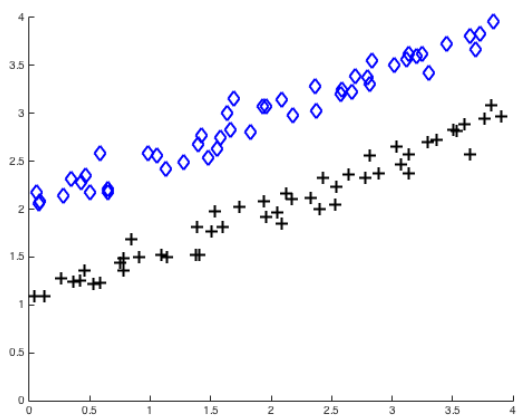
Jing

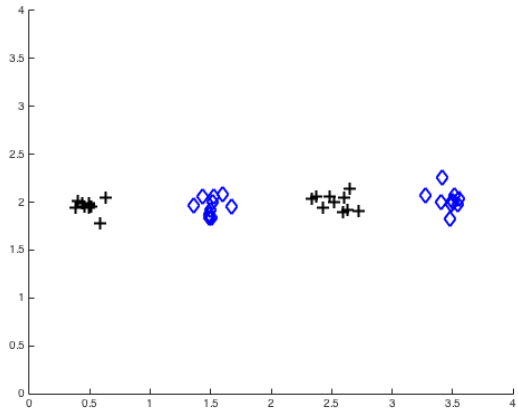
4 Classification Algorithms [14 pts]

1. [**9 pts**] Now we no longer restrict Naive Bayes classifier to have boolean features. Consider the problem of classifying the following 2-dimensional datasets into two classes (represented by 'plus' and 'diamond') using two classifiers: Logistic Regression and Gaussian Naive Bayes. For the Gaussian Naive Bayes, assume that both classes have equal variances in the y-dimension.
   For each of the three following datasets, **draw the Logistic Regression decision boundary** with a ***solid line*** and the **Gaussian Naive Bayes Boundary** with a ***dashed line***. If any of the classifiers cannot classify the data correctly, write **one sentence** explaining why not on the right of the figure.

   Note: You need to draw decision boundary only for the cases where the data can be classified correctly.



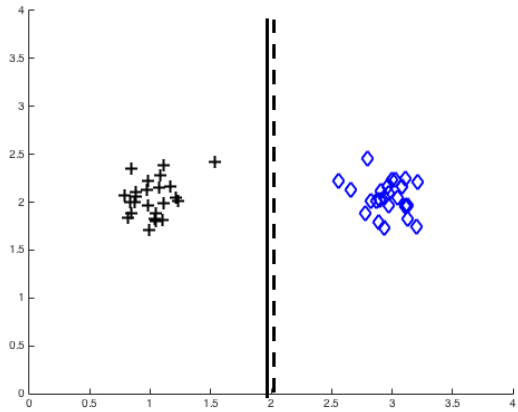


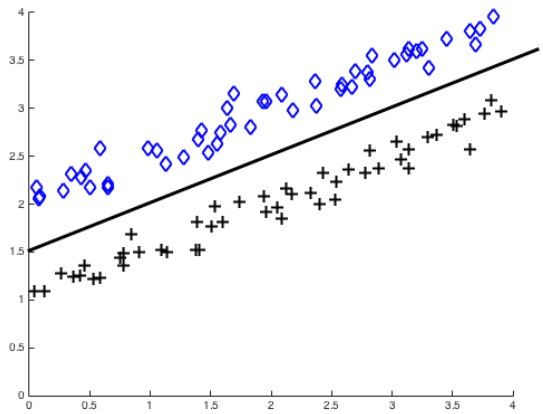


(a) Both GNB and LR can separate the data (Solid and dashed line $x = 2$)

(b) Only LR can separate the data (solid line $y = 1.5 + 0.5x$). GNB cannot represent any correlated Gaussian. The blue points around (0.25,2) will be closer to mean of black points. Similarly, black points around (3.75,2.5) will be closer to mean of blue points.

(c) Both GNB and LR can't separate the data. LR has linear boundary. GNB can't model a mixture of Gaussians.





Jing

2. (a) **[3 pts] True or False:** The weight vector estimated for logistic regression depends on the initialization of the weight vector at the beginning of gradient descent, assuming you run gradient descent for infinite iterations.

   ◯ True

   ◯ False

   True

   (b) **[2 pts] True or False:** Both linear regression and logistic regression are special cases of neural networks.

   ◯ True

   ◯ False

   True

Edward

5 Neural Networks [10 pts]

1. (a) **[3 pts]** Draw a fully connected Neural Network with the following specifications:

   - 4 Inputs x_1, x_2, x_3, x_4
   - 2 Hidden Layers
   - Hidden Layer 1 is made up of $a_j = \sum_i^4 W_{i,j}^1 x_i + \beta_j^1$ and $b_j = \sigma(a_j)$
   - Hidden Layer 2 is made up of $c_k = \sum_j^4 W_{j,k}^2 b_j + \beta_k^2$ and $d_k = \sigma(c_k)$
   - The output layer is made up of $e = \sum_k^3 W_k^3 d_k + \beta^3$ and $o = \sigma(e)$
   - Include the bias terms in all layers marking them as β^m with m as their layer number.

   Make sure you label your drawing.

   (b) **[2 pts]** For the tree created in part (a) what is back propagation derivative chain you needed to calculate $\frac{dL}{dW_{1,1}^2}$ Where L is the loss function. Use the variables given in part (a). Do not give any derivative which will be guaranteed 0. (You do not need to solve these derivatives, leave your answer as a product of $\frac{dX}{dY}$'s)

$$\frac{dL}{dW_{1,1}^2} = \frac{dL}{do}\frac{do}{de}\frac{de}{dd_1}\frac{dd_1}{dc_1}\frac{dc_1}{dW_{1,1}^2}$$

(c) **[3 pts]** For the tree created in part (a) what is back propagation derivative chain you needed to calculate $\frac{dL}{dW^1_{1,1}}$ Where L is the loss function. Use the variables given in part (a). Do not give any derivative which will be guaranteed 0. (You do not need to solve these derivatives, leave your answer as a product of $\frac{dX}{dY}$'s)

$\frac{dL}{dW^1_{1,1}} = \frac{dL}{do}\frac{do}{de}\sum_{k=1}^{3}\frac{de}{dd_k}\frac{dd_k}{dc_k}\frac{dc_k}{db_1}\frac{db_1}{da_1}\frac{da_1}{dW^1_{1,1}}$

(d) **[2 pts]** What can we say about this Neural Network if we replace the activation functions with $b_j = a_j$, $d_k = c_k$ and $o = e$. Specifically what other ML method can this be compared to? Explain your answer in one short sentence.

This is exactly Linear Regression. Since the product of weights can be written as new weights to get it in the form $o = w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4$

Daniel

6 SVM [16 pts]

We learned a two class linear SVM for a linearly separable input data. Let W and b be the parameters we obtained for the primal SVM formulation.

In the standard SVM formulation (SVM1) we use the following constraints for all x in class 1:

$$W^T x + b \geq 1$$

and for all x in class 0:

$$W^T x + b \leq -1$$

Assume that we learned a new SVM model (SVM2) using the following constraints instead, for all x in class 1:

$$W^T x + b \geq 0$$

and for all x in class 0:

$$W^T x + b < 0$$

1. **[2 pts]** If we compare the margin of SVM2 to that of SVM1 we can say that:

   ◯ The margin increased

   ◯ The margin decreased

   ◯ The margin stayed the same

   ◯ Impossible to tell

   Decrease (2). If we set the threshold at 0 then there will be no margin and since this is a linearly separable dataset the margin will decrease

2. Assume that we are using a new SVM, SVM3 which uses $\frac{W}{2}$ and $\frac{b}{2}$ where W and b are the parameters learned for SVM1. With these new parameters

   (a) **[2 pts]** Are we guaranteed that SVM3 would not make any mistakes on the training data? (recall that an SVM classifier determines the class based on the sign of $W^T x + b$ where x is the input).

   ◯ Yes

   ◯ No

   Yes. This is a linearly separable problem and everything that was higher than 0 before remains higher now and similarly for lower than 0.

   (b) **[2 pts]** How would the margin for SVM3 compare to the margin of SVM1?

   ◯ The margin would increase

   ◯ The margin would decrease

   ◯ The margin would stay the same

   ◯ Impossible to tell

The Margin would increase. The margin is $\frac{2}{\sqrt{(W^T W)}}$ and since W is divided by 2 it would increase.

(c) **[3 pts]** The number of support vectors for SVM3 compared to SVM1 would (recall that support vectors are those inputs that are either exactly on the +1 or -1 planes or those points that are between these planes and the decision boundaries)

- ◯ The number of support vectors would likely increase
- ◯ The number of support vectors would likely decrease
- ◯ The number of support vectors would likely stay the same
- ◯ Impossible to tell

The number of support vectors would likely increase. All previous support vectors now lie between the margin and the decision line and there could only be new support vectors added.

3. Now assume that we are using a new SVM, SVM4 which uses 2W and 2b where W and b are the parameters learned for SVM1. With these new parameters.

(a) **[2 pts]** How would the margin for SVM4 compare to the margin of SVM1?

- ◯ The margin would increase
- ◯ The margin would decrease
- ◯ The margin would stay the same
- ◯ Impossible to tell

The margin would decrease. Similar to 2.2 now the margin shrinks.

(b) **[2 pts]** The number of support vectors for SVM4 compared to SVM1 would (recall that support vectors are those inputs that are either exactly on the +1 or -1 planes or those points that are between these planes and the decision boundaries)

- ◯ The number of support vectors would likely increase
- ◯ The number of support vectors would likely decrease
- ◯ The number of support vectors would likely stay the same
- ◯ Impossible to tell

The number of support vectors would likely decrease. In fact, there will be no support vectors. All previous ones will now have a value of 2 or -2 and nothing has a lower value.

4. **[3 pts]** Assume that the number of support vectors for SVM1 is k. Can you give the exact number of support vectors for SVM3 or SVM4? Please choose *only one of them, either SVM3 or SVM4* and provide the number of support vectors for that classifier only (specify which one you chose). Its fine for the result to be a function of k (but not big O notation, so $2k$ is possible answer while $O(k^2)$ is not).

0 for SVM4 as noted above.

Ziv

7 Boosting [12 pts]

1. (a) **[2 pts]** Let $D = \{(X_i, Y_i)\}_{i=1}^n$ be a given dataset of n samples of distinct inputs $X_i \in \mathbb{R}^p$ and binary outputs $Y_i \in \{0, 1\}$. Let $\mathcal{H}$ be the set of binary classifiers specified by deep neural networks, with sigmoid units, and allowing for arbitrary depth, and arbitrary number of hidden units in each layer. What would be the smallest training error over D for the best classifier in the set $\mathcal{H}$?

   (b) **[2 pts]** Consider a weak learning algorithm that picks the best of the weak classifiers in set the $\mathcal{H}$ defined above, given a dataset, and suppose we run Adaboost using this weak learning algorithm. How many rounds of boosting will Adaboost undergo?

Solutions.

(a) Smallest training error is zero, given a DNN of sufficient complexity.

(b) Number of rounds is one. After the first round, the best weak learner already has training error zero, so the weights over samples for the next round will be zero.

2. Let $D = \{(X_i, Y_i)\}_{i=1}^n$ be a given dataset of n samples of distinct inputs $X_i \in \mathbb{R}^p$ and binary outputs $Y_i \in \{-1, 1\}$. Consider the set $\mathcal{H} := \{h_i\}_{i=1}^n$ of weak classifiers $h_i(x) = \mathbb{I}[k(x, X_i) > \delta]\, Y_i$, where $k(u, v)$ is some kernel function indicating similarity between u and v.

   (a) **[2 pts]** Consider a uniform ensembling of the classifiers above; what is the form of the resulting classifier, in terms of the data $\{(X_i, Y_i)\}_{i=1}^n$ and the kernel function k?

   (b) **[2 pts]** Consider a weak learning algorithm that picks the best of these weak

classifiers given a dataset, and suppose we run boosting using this weak learning algorithm. What is the form of the resulting classifier?

(c) **[2 pts]** Consider a classifier trained via Kernel SVM using the kernel function k. Recall that the form of the Kernel SVM classifier is similar to that of the standard linear SVM, except that Euclidean inner products $x_i^T x_j$ are replaced by $k(x_i, x_j)$. What is the form of this classifier?

Solutions. The forms of the classifiers are different if similar. Boosting would provide a classifier of the form:

$$f(x) = \text{sign}\left(\sum_i \alpha_i \mathbb{I}[k(x, X_i) > \delta_i]\, Y_i\right),$$

while Kernel SVM will provide:

$$f(x) = \text{sign}\left(\sum_i \alpha_i Y_i\, k(x, X_i)\right).$$

A uniform ensembling of the classifiers would be:

$$f(x) = \text{sign}\left(\sum_i \mathbb{I}[k(x, X_i) > \delta_i]\, Y_i\right).$$

The latter is also known as kernel regression.

3. **[2 pts]** Consider a binary classification problem with inputs $X \in \mathcal{R}^p$, and binary outputs $Y \in \{-1, 1\}$. Suppose we are given H distinct logistic regression models, and decide to take a weighted ensemble of these to make our prediction. Can you express the ensemble in terms of an artificial neural network?

Solutions. The overall ensemble is $f(x) = \text{sign}(\sum_{j=1}^{H} w_j \sigma(w_{hj} x + b_{h0}))$, which is precisely the form of a 2 layer NN.

Pradeep

8 Hierarchical Clustering and KNN [12 pts]

1. To save time in KNN calculations, we intend to use a binary hierarchical clustering tree to determine the nearest neighbors. After clustering the inputs, we store the average values for all features in internal nodes of the tree and for each input decide, at each level whether to go left or right until reaching a leaf which would be the neighbor selected. See Figure 1 for an example.

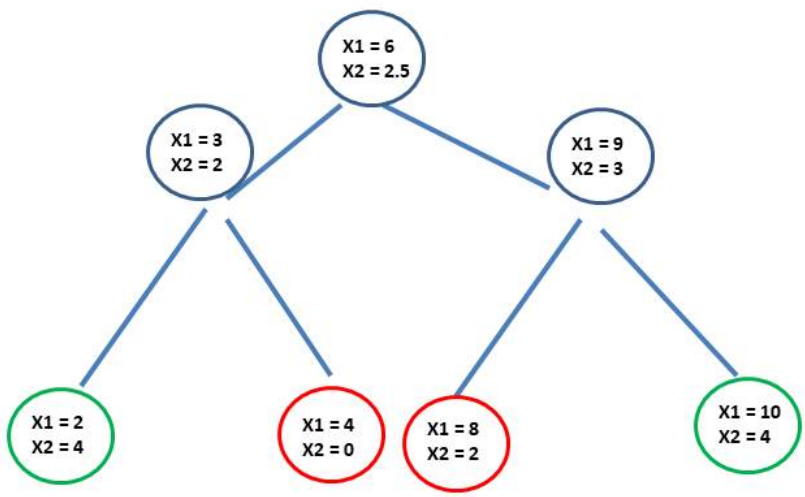


Figure 1: Figure 1: Sample KNN tree. Green nodes at the bottom are input samples from class 1. Red are inputs from class -1. Blue nodes are average values for nodes below them. A new sample is propagated down the tree and assigned one of the inputs to get the label. For example, if we need to classify a new input with values (x1=4, x2=4) we will first assign it to the left branch (left child of root) and then assign it to left again and so we predict green for this input.

The questions below assume that the tree has been constructed so they only focus on the time complexity and accuracy of classifying a new sample.

(a) **[3 pts]** For which tree will this lead to the largest run time saving? What would the run time of 1NN on such tree be?

A fully balanced tree (i.e. each internal node has either two internal node descendants or two lead node descendants). In that case the run time would be reduced from $O(n)$ to $O(\log n)$.

(b) **[3 pts]** Are there trees where such an approach would lead to an increase in run time compared to the standard 1NN? If so, which? If not, whats the worst run time in big O notations? (Explain your solution in one short sentence.)

- ◯ Yes
- ◯ No

No. The worst case scenario is a $O(N)$ length tree (each internal node has one internal and one leaf node descendent).

(c) **[3 pts]** We are now attempting to classify binary vectors. For this, we would like to use the hamming distance (number of positions in the vector in which the values are not the same). We construct a distance matrix based on hamming distance and build a hierarchical clustering tree using average linkage. Comparing the results of 1 nearest neighbors (1NN), with classification using the tree, is the tree guaranteed to identify the correct 1NN? Briefly explain your choice.

- ◯ Yes
- ◯ No

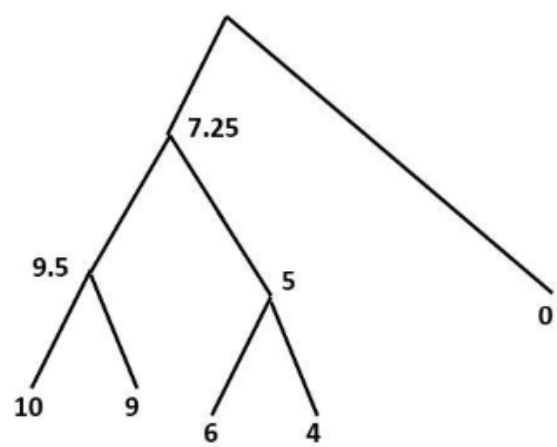
No. Since hierarchical clustering is greedy, the correct neighbor may end up in a cluster that at the very top level is very different from itself (see also below for an example).

(d) **[3 pts]** We are now interested in classifying binary vectors using a hierarchical clustering tree with average link. Here, the distance function is the difference in the number of 1 values between the vector (so for each vector we count how many 1's it has and the difference between these numbers is the number we use as the compute distance). Is the resulting tree guaranteed to identify the correct 1NN? Briefly explain

◯ Yes

◯ No

No. Assume in our training the NN is a sample with 4 ones. We also have a sample with 1 one, and a set of samples with 4,6,9,10 ones. In this case the tree will look like this:



Now, assume the input sample for classification has 3 ones. In that case it would be assigned to the 0 sample even though 4 is closer (since the first distance to the internal node of the 4 cluster is 4.25 which is bigger than 3, the distance to the 0 cluster).

Ziv

Do not remove this page! Use this page for scratch work.

Do not remove this page! Use this page for scratch work.