CIS 520, Machine Learning, Fall 2019
Homework 1
Due: Monday, September 16th, 11:59pm
Submit to Gradescope

Matthew Scharf

1 Non-Normal Norms

1. For the given vectors, the point closest to x_1 under each of the following norms is

   a) L_0:
   $x_2 = 1 + 1 + 1 + 1 = 4$
   $x_3 = 1 + 1 + 1 + 1 = 4$
   $x_4 = 1 + 1 + 0 + 1 = 3$
   x_4 is the closest.

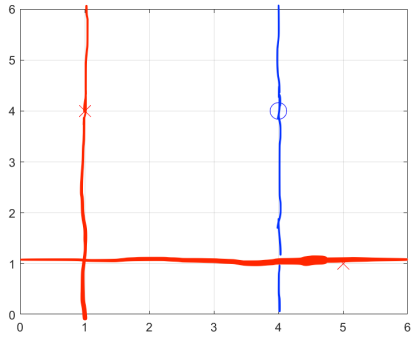
   b)L_1:
   $x_2 = 2.7 + .3 + 2.5 + .5 = 6$
   $x_3 = 3.8 + 1 + 2.1 + .7 = 7.6$
   $x_4 = 3.6 + 2.7 + 0 + 1.2 = 7.5$
   x_2 is the closest.

   c) L_2:
   $x_2 = \sqrt{2.7^2 + .3^2 + 2.5^2 + .5^2} = 3.73$
   $x_3 = \sqrt{3.8^2 + 1^2 + 2.1^2 + .7^2} = 4.51$
   $x_4 = \sqrt{3.6^2 + 2.7^2 + 0^2 + 1.2^2} = 4.66$
   x_2 is the closest.

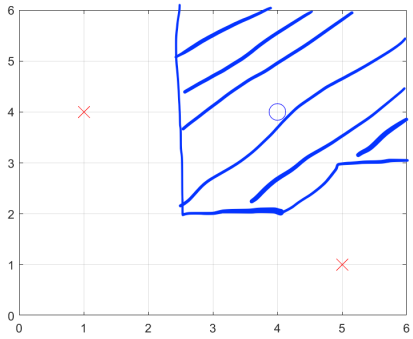
   d) L_{inf}:
   $x_2 = \max(2.7, .3, 2.5, .5) = 2.7$
   $x_3 = \max(3.8, 1, 2.1, .7) = 3.8$
   $x_4 = \max(3.6, 2.7, 0, 1.2) = 3.6$
   x_2 is the closest.

2. Draw the 1-Nearest Neighbor decision boundaries with the given norms and lightly shade the o region:

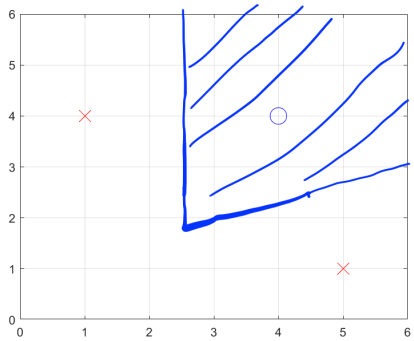
a) L_0



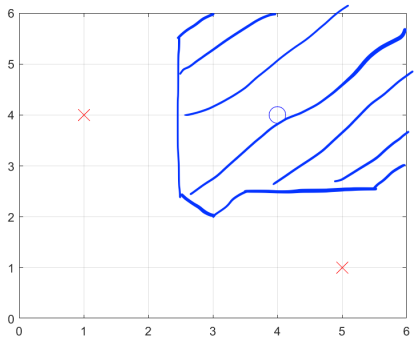
b) L_1



c) L_2



d) L_{inf}



2 Decision trees

1. Concrete sample training data.

(a)

$$\begin{aligned} H(Y) &= \frac{21}{40} log_2(\frac{40}{21}) + \frac{19}{40} log_2(\frac{40}{19}) \\ &= .488 + .510 \\ &= .998 \end{aligned}$$

(b)

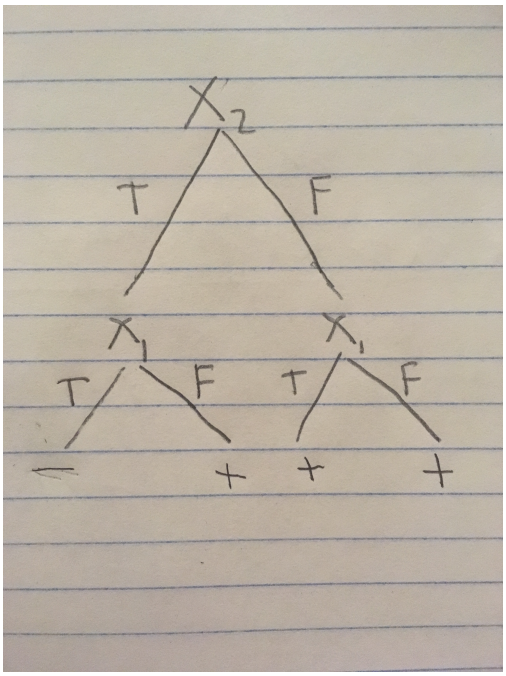
$$H(Y|X_1) = \frac{1}{5} log_2(\frac{9}{4}) + \frac{13}{40} log_2(\frac{22}{13}) + \frac{1}{4} log_2(\frac{9}{5}) + \frac{9}{40} log_2(\frac{22}{9}) = .983$$

$$IG(X_1) = .998 - .983 = .015$$

$$H(Y|X_2) = \frac{3}{20} log_2(\frac{5}{2}) + \frac{3}{8} log_2(\frac{5}{3}) + \frac{9}{40} log_2(\frac{5}{3}) + \frac{1}{4} log_2(\frac{5}{2}) = .970$$

$$IG(X_2) = .998 - .970 = .028$$

(c) The decision tree that would be learned is shown below.



2. Information gain and KL-divergence.

(a) If variables X and Y are independent, is $IG(x, y) = 0$? If yes, prove it. If no, give a counter example.

Yes, $IG(x, y) = 0$. Because X and Y are independent $p(x, y) = p(x)p(y)$. So, $\frac{p(x)p(y)}{p(x,y)} = 1$. So, $log(\frac{p(x)p(y)}{p(x,y)}) = 0$. So, all the terms in the sum are 0. So, $IG(x, y) = 0$.

(b) Prove that $IG(x, y) = H[x] - H[x \mid y] = H[y] - H[y \mid x]$, starting from the definition in terms of KL-divergence:

$$
\begin{aligned}
IG(x,y) &= KL\left(p(x,y)||p(x)p(y)\right) \\
&= -\sum_x \sum_y p(x,y) log(\frac{p(x)p(y)}{p(x,y)}) \\
&= -\sum_x \sum_y p(x,y)[log(p(x) + log(\frac{p(y)}{p(x,y)})] \\
&= [-\sum_x \sum_y p(x,y) log(p(x)] - [-\sum_x \sum_y p(x,y) log(\frac{p(x,y)}{p(y)})] \\
&= [-\sum_x p(x) log(p(x)] - [-\sum_x \sum_y p(x,y) log(p(x|y)] \\
&= H[x] - H[x \mid y]
\end{aligned}
$$

$= H[y] - H[y \mid x]$ by symmetry of the original equation for $IG(x, y)$.

3 High dimensional hi-jinx

1. Intra-class distance.

$$\begin{aligned}\mathbf{E}[(X - X')^2] &= E[X^2 - 2XX' + X'^2] \\ &= E[X^2] - 2E[X]E[X'] + E[X'^2] \\ &= (\mu^2 + \sigma^2) - (2\mu^2) + ((\mu^2 + \sigma^2) \\ &= 2\sigma^2\end{aligned}$$

2. Inter-class distance.

$$\begin{aligned}\mathbf{E}[(X - X')^2] &= E[X^2 - 2XX' + X'^2] \\ &= E[X^2] - 2E[X]E[X'] + E[X'^2] \\ &= (\mu_1^2 + \sigma^2) - (2\mu_1\mu_2) + ((\mu_2^2 + \sigma^2) \\ &= (\mu_1 - \mu_2)^2 + 2\sigma^2\end{aligned}$$

3. Intra-class distance, m-dimensions.

$$\mathbf{E}[\sum_{j=1}^{m}(X_j - X'_j)^2] = \sum_{j=1}^{m}\mathbf{E}[(X_j - X'_j)^2]$$

using the result from (1)

$$\begin{aligned}&= \sum_{j=1}^{m}[2\sigma^2] \\ &= 2m\sigma^2\end{aligned}$$

4. Inter-class distance, m-dimensions.

$$\mathbf{E}[\sum_{j=1}^{m}(X_j - X'_j)^2] = \sum_{j=1}^{m}\mathbf{E}[(X_j - X'_j)^2]$$

using the result from (2)

$$\begin{aligned}&= \sum_{j=1}^{m}[(\mu_{1j} - \mu_{2j})^2 + 2\sigma^2] \\ &= 2m\sigma^2 + \sum_{j=1}^{m}[(\mu_{1j} - \mu_{2j})^2]\end{aligned}$$

5. The ratio of expected intra-class distance to inter-class distance is: $\frac{2m\sigma^2}{2m\sigma^2+(\mu_{11}-\mu_{21})}$. As m increases towards ∞, this ratio approaches 1. This means that as $m \to \infty$, intra and inter class relationships become indistinguishable and the performance of the NN classifier gets worse and worse.

4 K-nearest neighbors Classification (Programming)

1. How does having a larger dataset might influence the performance of KNN?

   The accuracy will continue to improve but because the KNN must calculate the distance to each training data point, the run-time increases linearly with the size of the training set and the test set. This can become prohibitively slow for large datasets.

K	Norm	Accuracy (%)
3	L1	.94
3	L2	.95
3	L-inf	.94
5	L1	.94
5	L2	.93
5	L-inf	.94
7	L1	.93
7	L2	.92
7	L-inf	.93

Table 1: Accuracy for the KNN classification problem on the validation set

2. Tabulate your results in Table 1 for the **validation set**.

3. Finally, mention the best K and the norm combination you have settled upon from the above table and report the accuracy on the test set using that combination.

   The best hyper-parameters were $K = 3$ with the L_2 norm. With those, the accuracy on the test set was .88.

5 Decision Trees (Programming)

5.1 Part 1: Effects of Dataset Size on Performance

1. Report the training, validation, and test accuracies on the full and partial datasets below. Note that this portion will be graded by the Autograder.

Accuracy Scores		
	Full Dataset	Small Dataset
Training Accuracy	1.0	1.0
Validation Accuracy	0.92	0.91
Test Accuracy	0.9130434782608695	0.8840579710144928

2. Which dataset had a higher difference between training and test accuracy? Briefly explain why.

   The full dataset only had a difference of .087 while the partial dataset had a difference of .116. We can attribute this to overfitting in the case of the partial dataset because it's smaller.

5.2 Part 2: Effects of Dataset Size on Performance

1. Report the chosen hyperparameters for the complete and partial set below. Note that this section will be graded by the Autograder.

Grid Search Chosen Hyperparameters		
	Full Dataset	Small Dataset
Tree Depth	4	2
Max Leaf Nodes	5	3

2. Did the small dataset have higher or lower chosen hyperparameter values than the full dataset? Briefly explain why.
   The smaller dataset had lower chosen hyperparameters than the full dataset. This is because, as mentioned above, models trained on smaller datasets are more likely to suffer from overfitting. So, the smaller hyperparameters are to combat this tendency.

5.3 Part 3: Retrain Decision Tree and Plot Hyperparameter Search

1. Report the train, validation, and test accuracies after retraining the decision tree with the new hyperparameters. Also paste in the values for the training and validation scores lists when varying the max leaf node count hyperparameter.

Retrained Decision Tree Performance for Small Dataset	
	Score
Training Accuracy	0.955
Validation Accuracy	0.93
Test Accuracy	0.8985507246376812

Training and Validation List Values	
	List
Training	[0.945, 0.955, 0.96, 0.96, 0.96, 0.96, 0.96, 0.96, 0.96]
Validation	[0.91, 0.93, 0.92, 0.92, 0.92, 0.92, 0.92, 0.92, 0.92]

2. How did the training accuracy and testing accuracy change after tuning compared to before? Briefly explain why.

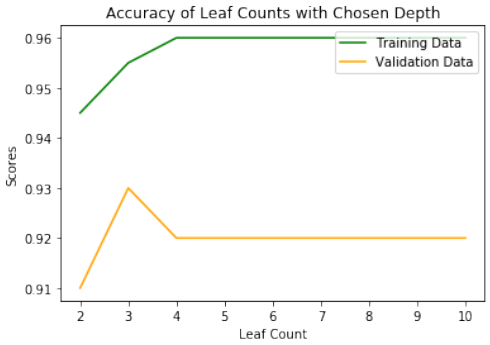
   The training accuracy decreased but the testing accuracy increased. This is because the model was overfitting before, but now with better regularization hyperparameters, the model is fitting the testing data better.

3. Paste the plot of training and validation scores with different leaf count values on the small dataset. Explain any trends or patterns with the plot within validation and training scores and briefly explain why.

   As the leaf count increases, so does the complexity of the model.

   The validation accuracy starts low (underfitting) peaks at 3 (properly fitting) and then decreases and flattens out (overfitting).

   The training data error is increasing and always higher than the validation data error. This is because as the complexity of the model increases, it will better represent the training data. However, it does cap out at .96 due to the complexity cap put in place by the max depth restriction.



6 Feature Scaling Effects (Programming)

1. Report the training and testing accuracies for unstandardized and standardized data for both Decision Trees and KNNs using their default hyperparameter values.

Scores for Unstandardized and Standardized Data				
	KNN Unscaled	KNN Scaled	DT Unscaled	DT Scaled
Training Accuracy	0.9475	0.9775	1.0	1.0
Test Accuracy	0.8986	0.9565	0.9130	0.9130

2. What happens to performance when we use standardization for data with decision trees? What about KNN? Briefly explain why each happened.

   For decision trees, standardization does not change the accuracy at all. This is because decision trees work by minimizing entropy which is not effected by rescaling.

   For KNN, standardization improves performance. KNN relies on a measure of distance and so the degree to which a dimension impacts the model is dictated by the scale of that dimension, which is arbitrary. So, by rescaling, we remove that noise, thereby improving the training and testing accuracy.