CS771 Homework 1

BHAVY KHATRI

TOTAL POINTS

100 / 100

QUESTION 1

1 Misclassification Rate vs Information Gain **10 / 10**

✓ **+ 3 pts Correctly calculated the misclassification rate**

+ **2 pts** Minor errors in calculating the misclassification rate

+ **1 pts** Major errors in calculating the misclassification rate

+ **0 pts** Totally incorrect calculation of misclassification rate, or not answered

✓ **+ 5 pts Correctly calculated the IG**

+ **3 pts** Minor errors in calculating IG (equtions correct, small calculation mistakes)

+ **1 pts** Major errors in calculating IG (equtions incorrect, numerous errors in calculations)

+ **0 pts** Totally incorrect calculation of IG, or not answered

✓ **+ 2 pts Sensible comparison of which of the two is better and why (should have mentioned about the IG/entropy taking into account the probabilities where as misclassification rate not depending on probabilities, or should have mentioned that the former better accounts for purity vs the latter does not)**

+ **1 pts** Misclassification rate vs IG comparison is not proper but still makes some sense

+ **0 pts** Misclassification rate vs IG comparison not given

+ **0 pts** Question not attempted.

+ **0 pts** Click here to replace this description.

QUESTION 2

2 Consistent or Not? **10 / 10**

✓ **+ 10 pts Correct explanation**

+ **8 pts** Mostly correct explanation but some minor erroneous points.

+ **6 pts** Uses a reasoning based on 1NN error rate not being worse than twice of Bayes optimal rate (which is zero in noise free setting). This only deserves partial credit because it just *uses* an existing fact.

+ **6 pts** Somewhat reasonable explanation but deviates too much from the theme of the question.

+ **0 pts** Incorrect explanation or not answered

QUESTION 3

3 Linear Regression meets Nearest Neighbors **10 / 10**

✓ **+ 8 pts Correctly plugged in x_* and combined x_* with (X'X)^{-1}X' term from the training data to expand and identify the prediction y_* as a weighted combination of N outputs in the training data**

+ **6 pts** Incomplete or errors in mathematical expansion step but show some elementary understanding

✓ **+ 2 pts Correct explanation - KNN uses only local neighbors with their weights whereas this prediction uses all the training examples (i.e., is global). Also, while in weighted KNN, the weights are typically the usual similarity (e.g.. Euclidean similarity or inverse Euclidean distance) between test and training points, here their similarities are modulated by the matrix Sigma = X'X (which depends on the all the training data).**

+ **1 pts** Explanation for local vs global not precise but shows some basic understanding

+ **0 pts** Not attempted/ Incorrect

QUESTION 4

4 Feature-Specific L2 Regularization **10 / 10**

✓ + 10 pts **Fully correct answer (correct expression, correct derivatives, and correct final solution)**

- 2 pts Minor errors in notation (e.g., not specifying the diagonal matrix properly in the regularizer) but overall correct otherwise.

- 4 pts Correct regularizer but minor errors in the derivatives (and incorrect final expression)

- 6 pts Correct regularizer but major errors in the derivatives (and incorrect final expression)

+ 0 pts Not attempted or incorrect.

QUESTION 5

5 Multi-output Regression with Reduced Number of Parameters 20 / 20

✓ + 15 pts **Correct derivation for S and correct final expression**

✓ + 5 pts **Correct explanation for the expression (X projected by B and then a standard multi-output regression to learn S on the transformed features)**

+ 2 pts Justification missing but only implied in the expression. (like writing the solution having (XB) form)

+ 12 pts Minor errors in derivatives etc but overall approach correct, or derivations mostly look correct but the final answer isn't correct.

+ 9 pts Major errors in derivatives etc but shows some elementary understanding of the problem

+ 5 pts Significant errors, incorrect expressions, but showed some steps

+ 0 pts Not attempted or incorrect.

+ 0 pts Unchecked

QUESTION 6

6 Programming Problem 40 / 40

✓ + 10 pts **Correctly computed the means of seen classes (common for both parts of the question)**

✓ + 10 pts **Correct implementation for convex conbination based method. Accuracy should be around 46% using naive approach (higher accuracies are possible using some hacks but 46% using the basic method is acceptable and deserves full marks)**

✓ + 15 pts **Correct implementation for regression based method. Accuracy should be around 73% for lambda = 10**

✓ + 5 pts **Code properly documented and easy to read/run**

+ 0 pts Not submitted

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 1

QUESTION 1

Student Name: Bhavy Khatri
Roll Number: 150186
Date: September 1, 2018

A vector symbol $\mathbf{b}$, a symbol in blackboard font $\mathbb{R}$, a symbol in calligraphic font $\mathcal{A}$, some colored text

1. Since, this is a classification problem and at each leaf node we have to predict any one of the class. We will predict by majority vote i.e. class that has maximum number of data points at the leaf node will be our prediction. In both the cases 200 data points will be classified incorrectly out of 800 points. The misclassification rate for both of the trees are equal and its value is $\frac{1}{4} = \mathbf{0.25}$.

2. Entropy for the set S with total C classes is given by: $H(S) = \sum_{c \in C} -p_c \log_2 p_c$ wherep_c is the fraction of inputs from the class with label c. Information gain is the difference between the entropy before and after the split.
   Entropy before split: $-\left(\frac{1}{2}\log\frac{1}{2} + \frac{1}{2}\log\frac{1}{2}\right)$
   Since the data points are divided into two parts after the split, the entropy is given by:
   Entropy after split = (fraction of points at leaf node 1)$\times$ Entropy at leaf node 1 + (fraction of points at leaf node 2)$\times$ Entropy at leaf node 2

   Entropy after split for tree A: $-\left(\frac{3}{4}\log\frac{3}{4} + \frac{1}{4}\log\frac{1}{4}\right) \times 2 \times \frac{1}{2}$

   Entropy after split for tree B: $-\left(\frac{1}{3}\log\frac{1}{3} + \frac{2}{3}\log\frac{2}{3}\right) \times \frac{3}{4}$

   $IG_A = 0.1887$, $IG_B = 0.3112$.
   Clearly $Tree_B$ has more "information gain" value than $Tree_A$.

3. Yes, the answer for both the part was different. The misclassification rate after the split was same for both the trees while information gain for Tree B was more than the A.
   The misclassification rate and entropy are two different notions. Misclassification rate tell us what fraction of data points will be misclassified whereas the information gain is the quantity to measure difference in entropy. Entropy is the direct measure of "purity" in the data. By purity we mean that how much non-uniform the distribution of the data point is. High entropy means that the data follows more like a uniform distribution. In case of decision trees our goal is to split the data in such a way that result in groups as pure as possible.

1 Misclassification Rate vs Information Gain **10 / 10**

- ✓ **+ 3 pts** **Correctly calculated the misclassification rate**
- **+ 2 pts** Minor errors in calculating the misclassification rate
- **+ 1 pts** Major errors in calculating the misclassification rate
- **+ 0 pts** Totally incorrect calculation of misclassification rate, or not answered
- ✓ **+ 5 pts** **Correctly calculated the IG**
- **+ 3 pts** Minor errors in calculating IG (equtions correct, small calculation mistakes)
- **+ 1 pts** Major errors in calculating IG (equtions incorrect, numerous errors in calculations)
- **+ 0 pts** Totally incorrect calculation of IG, or not answered
- ✓ **+ 2 pts** **Sensible comparison of which of the two is better and why (should have mentioned about the IG/entropy taking into account the probabilities where as misclassification rate not depending on probabilities, or should have mentioned that the former better accounts for purity vs the latter does not)**
- **+ 1 pts** Misclassification rate vs IG comparison is not proper but still makes some sense
- **+ 0 pts** Misclassification rate vs IG comparison not given
- **+ 0 pts** Question not attempted.
- **+ 0 pts** Click here to replace this description.

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 1

Student Name: Bhavy Khatri
Roll Number: 150186
Date: September 1, 2018

QUESTION 2

Proof by contradiction: Suppose the classifier is not consistent which means its error rate is bigger than bayes optimal error rate. Since the bayes optimal error rate is zero which suggests that there exists a point for which k-nearest-neighbour (KNN) label doesn't match with actual label. Suppose the correct label is p and the one predicted by KNN is q. But as the correct label is p and the there are infinite number of points which means in the sufficiently small neighbourhood there should be no points other that points with label p. Which is a contradiction as label of the point will be p but we started with q. This proves that the classifier is consistent.

2 Consistent or Not? **10 / 10**

✓ **+ 10 pts** **Correct explanation**

+ 8 pts Mostly correct explanation but some minor erroneous points.

+ 6 pts Uses a reasoning based on 1NN error rate not being worse than twice of Bayes optimal rate (which is zero in noise free setting). This only deserves partial credit because it just *uses* an existing fact.

+ 6 pts Somewhat reasonable explanation but deviates too much from the theme of the question.

+ 0 pts Incorrect explanation or not answered

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 1

QUESTION 3

Student Name: Bhavy Khatri
Roll Number: 150186
Date: September 1, 2018

- **Properties of Inverse and Transpose:**
  1. $(\mathbf{A}^T)^{-1} = (\mathbf{A}^{-1})^T$
  2. $(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$

- **Derivation of w_n**
  Note that,

$$
\begin{aligned}
f(\mathbf{x}_*) &= \hat{\mathbf{w}}^T\mathbf{x}_* \\
&= \mathbf{y}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_* \\
&= \sum_{n=1}^{N} y_n\mathbf{x}_n^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_* && \text{(Using property 1 \& 2)}
\end{aligned}
$$

  Clearly, $f(\mathbf{x}_*) = \sum_{n=1}^{N} w_n y_n$ where $w_n = \mathbf{x}_n^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_*$

- **How w_n's of Linear Regression differs from that of K-Nearest-Neighbours:**
  For KNN $f_{knn}(\mathbf{x}_*) = \sum_{n\in\mathcal{N}_k(\mathbf{x}_*)} \frac{1}{\|\mathbf{x}_n-\mathbf{x}_*\|} y_n$ where $\mathcal{N}_k(\mathbf{x}_*)$ is the set of K closest training examples from $\mathbf{x}_*$. Here w_n is inversely proportional to the euclidean distance between $\mathbf{x}_n$ and $\mathbf{x}_*$.
  - In case of **linear regression** w_n denotes the general form of inner product($\mathbf{a}^T\mathbf{Mb}$) of (x_n, x_*).
  - In case of **KNN**, it denotes the inverse of distance between $x_n \& x_*$.

3 Linear Regression meets Nearest Neighbors 10 / 10

✓ **+ 8 pts** Correctly plugged in x_* and combined x_* with (X'X)^{-1}X' term from the training data to expand and identify the prediction y_* as a weighted combination of N outputs in the training data

+ 6 pts Incomplete or errors in mathematical expansion step but show some elementary understanding

✓ **+ 2 pts** Correct explanation - KNN uses only local neighbors with their weights whereas this prediction uses all the training examples (i.e., is global). Also, while in weighted KNN, the weights are typically the usual similarity (e.g.. Euclidean similarity or inverse Euclidean distance) between test and training points, here their similarities are modulated by the matrix Sigma = X'X (which depends on the all the training data).

+ 1 pts Explanation for local vs global not precise but shows some basic understanding

+ 0 pts Not attempted/ Incorrect

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 1

Student Name: Bhavy Khatri
Roll Number: 150186
Date: September 1, 2018

QUESTION 4

Alternative objective function:
$\mathcal{L}(\mathbf{w}) = \sum_{n=1}^{N}(y_n - \mathbf{w}^T\mathbf{x}_n)^2 + \sum_{d=1}^{D}\lambda_d w_d^2 = \sum_{n=1}^{N}(y_n - \mathbf{w}^T\mathbf{x}_n)^2 + \mathbf{w}^T\ \mathbf{L}\ \mathbf{w}$, where,

$$L = \begin{bmatrix} \lambda_1 & 0 & 0 & \dots & 0 \\ 0 & \lambda_1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \lambda_n \end{bmatrix}$$

Closed form expression:

$$\begin{aligned} \hat{\mathbf{w}} &= argmin_w \mathcal{L}(\mathbf{w}) \\ \frac{\partial \mathcal{L}}{\partial \mathbf{w}} &= 0 \\ -2\sum_{n=1}^{N}(y_n - \mathbf{w}^T\mathbf{x}_n)\mathbf{x}_n + 2\mathbf{L}\mathbf{w} &= 0 \\ \sum_{n=1}^{N}\mathbf{x}_n\mathbf{x}_n^T\mathbf{w} + \mathbf{L}\mathbf{w} &= \sum_{n=1}^{N} y_n x_n \\ (\sum_{n=1}^{N}\mathbf{x}_n\mathbf{x}_n^T + \mathbf{L})\mathbf{w} &= \sum_{n=1}^{N} y_n x_n \\ \mathbf{w} &= (\mathbf{X}^T\mathbf{X} + \mathbf{L})^{-1}\mathbf{X}^T\mathbf{y} \end{aligned}$$

4 Feature-Specific L2 Regularization **10 / 10**

✓ **+ 10 pts** **Fully correct answer (correct expression, correct derivatives, and correct final solution)**

- 2 pts Minor errors in notation (e.g., not specifying the diagonal matrix properly in the regularizer) but overall correct otherwise.

- 4 pts Correct regularizer but minor errors in the derivatives (and incorrect final expression)

- 6 pts Correct regularizer but major errors in the derivatives (and incorrect final expression)

+ 0 pts Not attempted or incorrect.

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 1

Student Name: Bhavy Khatri
Roll Number: 150186
Date: September 1, 2018

QUESTION 5

- **Properties of trace:**
  1. $tr(\mathbf{A}+\mathbf{B}) = tr(\mathbf{A}) + tr(\mathbf{B})$
  2. $\frac{\partial}{\partial \mathbf{X}} tr(\mathbf{AX}) = \mathbf{A}^T$
  3. $\frac{\partial}{\partial \mathbf{X}} tr(\mathbf{X}^T\mathbf{A}) = \mathbf{A}$
  4. $\frac{\partial}{\partial \mathbf{X}} tr(\mathbf{X}^T\mathbf{AX}) = \mathbf{AX} + \mathbf{A}^T\mathbf{X}$

- **Derivation:**
  $\mathcal{L}(\mathbf{S}) = tr[(\mathbf{Y}-\mathbf{XBS})^T(\mathbf{Y}-\mathbf{XBS})] = tr(\mathbf{Y}^T\mathbf{Y}) - tr(\mathbf{Y}^T\mathbf{XBS}) - tr(\mathbf{S}^T\mathbf{B}^T\mathbf{X}^T\mathbf{Y}) + tr(\mathbf{S}^T\mathbf{B}^T\mathbf{X}^T\mathbf{XBS})$ (using property 1)
  We have to find $\hat{\mathbf{S}} = argmin\mathcal{L}(\mathbf{S})$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{S}} = 0$$
$$-\mathbf{B}^T\mathbf{X}^T\mathbf{Y} - \mathbf{B}^T\mathbf{X}^T\mathbf{Y} + \mathbf{B}^T\mathbf{X}^T\mathbf{XBS} + \mathbf{B}^T\mathbf{X}^T\mathbf{XBS} = 0 \quad \text{(by property 2,3 \& 4)}$$
$$(\mathbf{XB})^T(\mathbf{XB})\mathbf{S} = (\mathbf{XB})^T\mathbf{Y} \quad \text{By}(\mathbf{AB})^T = \mathbf{A}^T\mathbf{B}^T$$
$$\mathbf{S} = [(\mathbf{XB})^T(\mathbf{XB})]^{-1}(\mathbf{XB})^T\mathbf{Y}$$

In Problem 4 of the Practice Problem 1, we get the solution for $\hat{\mathbf{W}} = [(\mathbf{X})^T(\mathbf{X})]^{-1}(\mathbf{X})^T\mathbf{Y}$. Clearly, current equation uses $\mathbf{XB}$ as a transformed version of $\mathbf{X}$

5 Multi-output Regression with Reduced Number of Parameters 20 / 20

✓ **+ 15 pts** **Correct derivation for S and correct final expression**

✓ **+ 5 pts** **Correct explanation for the expression (X projected by B and then a standard multi-output regression to learn S on the transformed features)**

+ 2 pts Justification missing but only implied in the expression. (like writing the solution having (XB) form)

+ 12 pts Minor errors in derivatives etc but overall approach correct, or derivations mostly look correct but the final answer isn't correct.

+ 9 pts Major errors in derivatives etc but shows some elementary understanding of the problem

+ 5 pts Significant errors, incorrect expressions, but showed some steps

+ 0 pts Not attempted or incorrect.

+ 0 pts Unchecked

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 1

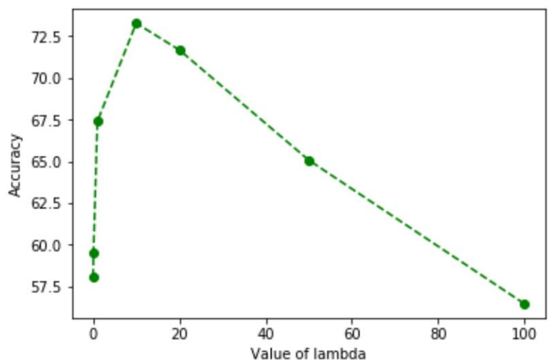
Student Name: Bhavy Khatri
Roll Number: 150186
Date: September 1, 2018

QUESTION 6

1. **Method 1:** For convex method the accuracy on the test data set was **46.89%**.

2. **Method 2:** The following λ vs accuracy table was obtained. The highest value of accuracy was obtained for $\lambda = 10$.

λ	**Accuracy (in %)**
0.01	58.09
0.1	59.54
1	67.39
10	73.28
20	71.68
50	65.08
100	56.47



6 Programming Problem 40 / 40

✓ **+ 10 pts** Correctly computed the means of seen classes (common for both parts of the question)

✓ **+ 10 pts** Correct implementation for convex conbination based method. Accuracy should be around 46% using naive approach (higher accuracies are possible using some hacks but 46% using the basic method is acceptable and deserves full marks)

✓ **+ 15 pts** Correct implementation for regression based method. Accuracy should be around 73% for lambda = 10

✓ **+ 5 pts** Code properly documented and easy to read/run

+ 0 pts Not submitted