# CSC 480/580 Homework 1

Due: 2/15 (Th) 5pm

**Instructions:**

- Submit your homework on time to gradescope. NO LATE DAYS, NO LATE SUBMISSIONS ACCEPTED.
- The submission must be one single PDF file (use Acrobat Pro from the UA software library if you need to merge multiple PDFs).
- Email your code to `csc580homeworks@gmail.com`.
  - You can use word processing software like Microsoft Word or LaTeX.
  - You can also hand-write your answers and then scan it. If you use your phone camera, I recommend using TurboScan (smartphone app) or similar ones to avoid looking slanted or showing the background.
  - Watch the video and follow the instruction: `https://youtu.be/KMPoby5g_nE` .
  - Points will deducted when you do not follow the instruction.
- Collaboration policy: do not discuss answers with your classmates. You can discuss HW for the clarification or any math/programming issues at a high-level. If you do get help from someone, please make sure you write their names down in your answer.
- If you cannot answer a problem, describing what efforts you have put in to solve the problem and where you get stuck will receive partial credit. Also, feel free to post your questions on Piazza.

**Problem 1.** Let $(X, Y)$ follow the distribution $\mathcal{D}$, which has the following joint probability table:

| | $X = 0$ | $X = 1$ | $X = 2$ | $X = 3$ |
|---|---|---|---|---|
| $Y = -1$ | $\frac{1}{54}$ | $\frac{1}{9}$ | $\frac{2}{9}$ | $\frac{4}{27}$ |
| $Y = +1$ | $\frac{2}{9}$ | $\frac{2}{9}$ | $\frac{1}{18}$ | $0$ |

(a) Let classifier $f$ to be such that $f(0) = f(1) = f(2) = -1$ and $f(3) = +1$. What is the error rate of $f$ on $\mathcal{D}$, $L_{\mathcal{D}}(f) = \mathbb{P}_{(x,y)\sim\mathcal{D}}(f(x) \neq y)$?

(b) What is $\mathcal{D}$'s Bayes optimal classifier $f_{BO}$?

(c) What is $\mathcal{D}$'s Bayes error rate $L_{\mathcal{D}}(f_{BO})$?

**Problem 2.** Generalized uncertainty measures.

(a) For $\mathbf{p} = (p_1, p_2, p_3)$, recall that the classification error-based uncertainty measure is defined as $v_1(\mathbf{p}) = 1 - \max_{k\in\{1,2,3\}} p_k$ and the Gini index-based uncertainty measure is defined as $v_2(\mathbf{p}) = 1 - \sum_{k=1}^{d} p_k^2$. Let $\mathbf{p} = (0.6, 0, 0.4)$ and $\mathbf{q} = (0.62, 0.19, 0.19)$. Is $v_1(\mathbf{p}) \geq v_1(\mathbf{q})$? Is $v_2(\mathbf{p}) \geq v_2(\mathbf{q})$? Justify your answer.

(b) Given a training dataset $S$, we would like to calculate the information score of feature $x_f$ using the entropy uncertainty score $u(T) = \sum_{y\in\mathcal{Y}} P_T(Y = y)\log_2(\frac{1}{P_T(Y=y)})$. Using the notation in the lecture slides, denote $S_L = \left\{(x, y) \in S : x_f = 0\right\}$, $S_R = \left\{(x, y) \in S : x_f = 1\right\}$, and $p_L, p_R$ as their respective proportion.

| | $x_f = 0$ | $x_f = 1$ | Total |
|---|---|---|---|
| $y = -1$ | 4 | 3 | 7 |
| $y = +1$ | 4 | 9 | 13 |

(b.1) Calculate $u(S)$, $u(S_L)$, $u(S_R)$ respectively. Express the results in decimals.

(b.2) Calculate $\mathrm{Score}(f, S) = u(S) - (p_L u(S_L) + p_R u(S_R))$. Is the score negative or negative? Does your calculation result match your intuition?

**Problem 3.** Decision trees with entropy uncertainty.

(a) Consider the entropy uncertainty $u(S) = \sum_{y \in \mathcal{Y}} P_S(Y = y) \log_2(\frac{1}{P_S(Y=y)})$ where $S$ is a labeled dataset and $P_S(Y = y)$ is the fraction of examples in $S$ with label $y$. Note that when $P_S(Y = y) = 0$, the term $V_y = P_S(Y = y) \log_2(1/P_S(Y = y))$ is undefined. In this case, which value should we use for $V_y$ to make sure $V_y$ is continuous w.r.t. $P_S(Y = y)$? *Hint: Remember L'Hopital's rule.*

(b) Implement the decision tree in Python as described in the book (handles only the binary features) but use the entropy instead of the classification error. Implement an option of `max_depth` so the trained tree will have depth at most `max_depth` (in our case, it corresponds to only considering at most `max_depth` features). Make sure to email your code to `csc580homeworks@gmail.com` so that I can run it.

Use the data in the book (Table 1) while taking the rating 2/1/0 as positive and -1/-2 as negative. Train your decision tree with your code with `max_depth` = 2. Report your tree along with the following information (in whatever form a person can reasonably comprehend)

- What are the branching questions at each node?
- What are the uncertainty scores at each node?
- Show the predicted label for each leaf node.

**Problem 4.** The $k$-Nearest Neighbor Classifier.

Let us define the data distribution $\mathcal{D}$ consisting of two-dimensional features $X \in \mathbb{R}^2$. Each dimension of $X = (X_1, X_2)^T$ is uniformly distributed on the unit interval: $X_1 \sim \mathsf{Uniform}[0,1]$ and $X_2 \sim \mathsf{Uniform}[0,1]$. Let $i(X)$ be 1 if $X_1 < 1/3$, 2 if $X_1 \in [1/3, 2/3)$, and 3 if $X_1 \geq 2/3$. Define $j(X)$ similarly for the second dimension $X_2$ (i.e., replace $X_1$ above by $X_2$). Furthermore, the labels are binary with $\mathbb{P}(Y = 1 \mid X = x) = A_{i(x),j(x)}$ and,

$$A = \begin{pmatrix} .1 & .2 & .2 \\ .2 & .4 & .8 \\ .2 & .8 & .9 \end{pmatrix}$$

($A_{i,j} \in \mathbb{R}$ is the entry of matrix $A$ at $i$-th row, $j$-th column) Throughout, we abuse notation and use $\mathbb{P}$ for both probability of events and the density function for continuous random variables.

Some preparations:

(1) Using the book as a guideline, implement the $k$-Nearest Neighbor algorithm with Euclidean distance in Python.

(2) Implement a function that draws $m$ i.i.d. samples from $\mathcal{D}$. Draw 10,000 points from $\mathcal{D}$ and call them a test set, but do this once and for all and use the same test set throughout the problems here.

Questions:

(a) What is $\mathcal{D}$'s Bayes optimal classifier and Bayes error rate?

(b) Plot the *learning curve* for nearest-neighbor classification. Let $k = 4$. Define $\mathcal{M} = \{10, 30, 100, 300, 1000, 3000\}$. Call the following one 'trial':

- Draw 3000 fresh data points from $\mathcal{D}$ and call it $S$.
- Then, for each $m \in \mathcal{M}$, choose the first $m$ data points from $S$, train a $k$-NN classifier with them, and then evaluate its test set error.

Perform 5 trials, compute the average test set error, and report the plot of 'test error rate' vs $m$. In the same plot, plot a horizontal line that shows the Bayes error rate so we know how close we get to the Bayes error rate.

(c) Let $\mathcal{K} = \{1, 2, 4, 8, 16, 32, 64\}$. Do (b) for every $k \in \mathcal{K}$. When $k \geq m$, simply force the code to set $k = m$.

**Problem 5.** ****CSC 580 Students Only**** Hyperparameter Tuning.

(a) Let us add hyperparameter tuning to the $k$-NN algorithm in the previous problem. For this, just perform one trial (instead of 5) for simplicity. For each $m \in \mathcal{M}$, try tuning $k$ by each of the following:

- training set error.
- 20% hold out from the training set.
- 5-fold cross validation.
- test set error (this is impossible in practice, but we just want to see what is the actually best one).

Implement each tuning method above and for each $m \in \mathcal{M}$ report:

- What is the tuned $k$?
- What is the test set error when you use the tuned $k$? How far are they from the actual best $k$ measured by the test set tuning?

Discuss your findings; e.g., does one perform better than the other? why? if there is a failing method, explain why.

(b) Let $\mathcal{A}$ be a learning algorithm that takes in a dataset $S$ and performs 5-fold cross validation with $k$-NN to choose the best $k \in \{1, 2, 4, 8, 16, 32, 64\}$, and then trains a $k$-NN classifier using $S$ with that chosen $k$. Plot the learning curve of $\mathcal{A}$. For each $m \in \mathcal{M}$, plot both the training set data points and the decision surface of the classifier obtained by $\mathcal{A}$ in one figure. (Total $|\mathcal{M}|$ plots.)

(c) Use the decision tree implementation of scikit-learn (`https://scikit-learn.org/stable/modules/tree.html#tree`) and perform the same procedure as (a), but now by tuning its `max_depth` $\in \{1, 2, 3, 4, 5, 6\}$. Compare the trained $k$-NN and decision tree classifiers side-by-side. Are there any differences? What might be the cause of the difference, if any? Are there any qualitative difference in the decision boundaries?