WEEK 1 STUDY GUIDE (SOLUTIONS)

10-301/10-601 Introduction to Machine Learning (Summer 2025)
http://www.cs.cmu.edu/~hchai2/courses/10601
Released: Monday, May 12th, 2025
Quiz Date: Friday, May 16th, 2025
TAs: Andy, Canary, Michael, Sadrishya, and Neural the Narwhal

Summary These questions are meant to prepare you for the upcoming quiz on Decision Trees, kNNs, Model Selection and Perceptrons. You'll first work through some information theory basics and terminology before "learning" a Decision Tree on paper. You'll also visually explore how to learn a Decision Tree on real-valued features. Then, you'll compare how decision trees and kNN relate, work through some simple kNN examples, think about how to correctly select the hyperparameter k and analyze the behavior of the perceptron learning algorithm in a variety of settings.

Note These questions are entirely optional; you do not need to submit your answers to these questions. However, at least 50% of the questions that will appear in your workshop quiz will be *identical or nearly identical* to questions in this document. Thus, we recommend you to at least attempt every question. Furthermore, we *highly encourage* you to work in groups to solve these questions: because you are not being directly assessed on your solutions, feel free to share solutions and discuss ideas with your peers.

We encourage you to work on this study guide regularly throughout the week; in particular, this study guide is organized in sections where each section corresponds to a particular day's lectures. Here is our recommended schedule for working on this study guide:

1. Mutual Information - after Monday's (5/12) lectures
2. Depth and Pruning - after Tuesday's (5/13) lectures
3. Our First Tree - after Tuesday's (5/13) lectures
4. Real-valued Decision Trees - after Tuesday's (5/13) lectures
5. Decision Trees and kNNs - after Wednesday's (5/14) lectures
6. k-Nearest Neighbors - after Wednesday's (5/14) lectures
7. Perceptron - after Thursday's (5/15) lectures

1 Mutual Information

1. $H(X) = -\sum_{x=1}^{6}(\frac{1}{6})\log_2(\frac{1}{6}) = \log_2(6)$
2. $I(X;Y) = H(X) - H(X \mid Y)$ so $I(X;Y)$ is 0 if and only if X and Y are independent.
   Mathematically, $H(Y \mid X) = H(Y)$ making $I(X;Y)$ go to 0.
   Intuitively, this is because if X and Y are independent, knowing one tells you nothing about the other and vice versa, so their mutual information is 0.

2 Depth and Pruning

1. The depth of the tree is 1
2. The depth of the tree- is 4. The depth of node X_1 is 0 and the depth of X_5 is 2.
3. A.

   The higher this threshold value is, the lesser nodes/smaller depth the decision tree contains.
4. B.

   Pruning tends to increase the training error and decrease the test error.

3 Our First Tree

1. H(Y) = -$\frac{6}{8} * \log_2(\frac{6}{8}) - \frac{2}{8} * \log_2(\frac{2}{8}) \approx 0.811$
2. - $H(Y \mid X_1 = sunny) = -[\frac{1}{3} * \log_2(\frac{1}{3}) + \frac{2}{3} * \log_2(\frac{2}{3})] \approx 0.918$
   - $H(Y \mid X_1 = rain) = 0$
   - $H(Y \mid X_1 = overcast) = 0$

   $\implies H(Y \mid X_1) = [\frac{3}{8} * 0.918 + \frac{3}{8} * 0 + \frac{2}{8} * 0] \approx 0.344$
   $\implies I(Y;X_1) \approx 0.811 - 0.344 = 0.467$
3. - $H(Y \mid X_2 = hot) = -[\frac{1}{3} * \log_2(\frac{1}{3}) + \frac{2}{3} * \log_2(\frac{2}{3})] \approx 0.918$
   - $H(Y \mid X_2 = cool) = 0$
   - $H(Y \mid X_2 = mild) = -[\frac{3}{4} * \log_2(\frac{3}{4}) + \frac{1}{4} * \log_2(\frac{1}{4})] \approx 0.811$

   $\implies H(Y \mid X_2) = [\frac{3}{8} * 0.918 + \frac{1}{8} * 0 + \frac{4}{8} * 0.811] \approx 0.75$
   $\implies I(Y;X_2) \approx 0.811 - 0.75 = 0.061$
4. - $H(Y \mid X_3 = high) = -[\frac{1}{2} * \log_2(\frac{1}{2}) + \frac{1}{2} * \log_2(\frac{1}{2})] = 1$
   - $H(Y \mid X_2 = normal) = 0$

   $\implies H(Y \mid X_3) = [\frac{4}{8} * 1.0 + \frac{4}{8} * 0] = 0.5$
   $\implies I(Y;X_3) \approx 0.811 - 0.5 = 0.311$
5. X_1, because this has the highest mutual information.

6. The sub-datasets $\mathcal{D}_{(X_1=rain)}$ and $\mathcal{D}_{(X_1=overcast)}$ are pure. So we need to split only on the sub-dataset $\mathcal{D}_{(X_1=sunny)}$.

   $H(Y_{(X_1=sunny)}) = -\frac{1}{3} * \log_2(\frac{1}{3}) - \frac{2}{3} * \log_2(\frac{2}{3}) \approx 0.918$

   For attribute X_2,

   - $H(Y_{(X_1=sunny)} \mid X_2 = hot) = 0$
   - $H(Y_{(X_1=sunny)} \mid X_2 = cool) = 0$
   - $H(Y_{(X_1=sunny)} \mid X_2 = mild) = -[\frac{1}{2} * \log_2(\frac{1}{2}) + \frac{1}{2} * \log_2(\frac{1}{2})] = 1$

   $\implies H(Y_{(X_1=sunny)} \mid X_2) = [\frac{2}{3} * 1.0 + \frac{1}{3} * 0] \approx 0.67$
   $\implies I(Y_{(X_1=sunny)}; X_2) \approx 0.918 - 0.67 \approx 0.25$

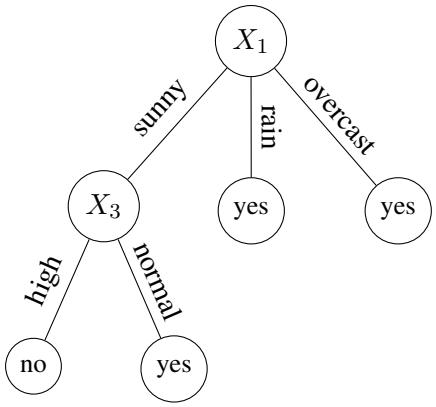
   For attribute X_3,

   - $H(Y_{(X_1=sunny)} \mid X_3 = high) = 0$
   - $H(Y_{(X_1=sunny)} \mid X_3 = normal) = 0$

   $\implies H(Y_{(X_1=sunny)} \mid X_3) = [\frac{2}{3} * 0 + \frac{1}{3} * 0] = 0$
   $\implies I(Y_{(X_1=sunny)}; X_3) \approx 0.918$

   Thus, we would split on attribute X_3.

7. The complete decision tree is shown below



4 Real-valued Decision Trees

1. 1
2. $1 - (\frac{8}{10} * (-\frac{5}{8} * \log_2(\frac{5}{8}) - \frac{3}{8} * \log_2(\frac{3}{8})) + \frac{2}{10} * 0) \approx 0.236$
3. 0.2; Achievable by a few different stumps e.g., $X_1 < 3.5$ or $X_1 < 2$
4. True

5 Decision Trees and *k*NNs

1. (a) Yes. One way is to choose each $t = \frac{x^{(i)}+x^{(j)}}{2}$ for consecutive i, j, forming a decision tree with $N - 1$ splits.

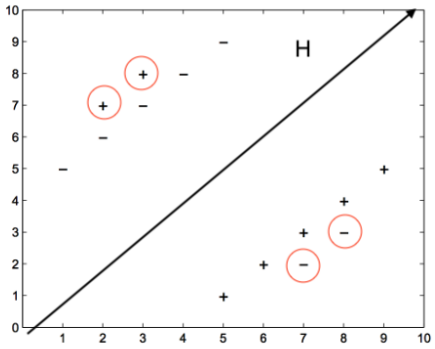
   (b) A, B, E. Any 1-NN with non-vertical or non-horizontal decision boundaries cannot have an equivalent decision tree. For instance, it is impossible for two points that aren't perfectly vertical or horizontal.

   A and B have vertical/horizontal boundaries. And 1-NN on E can be decomposed into vertical/horizontal boundaries as well.

6 *k* Nearest Neighbors

1. (a) 0. The training error rate of 1-nn model is always 0.

   (b) $\frac{4}{14}$. For knn with k = 6, the decision boundary looks roughly like H shown below, which points above H to be categorized as "-" and below as "+". In this case, 4 out of 14 training data is misclassified, meaning the training error rate is $\frac{4}{14}$.



2. (a) False.

   (b) A. It's better, because by merely lowering the training error does not always guarantee a generalized model. Instead, this may lead us with an overfitted model. In model selection, we prefer cross-validation technique to find better hyperparameters.

   (c) No, this is not a good idea. Test data should never be exposed to the machine learning model before test time. Using this data to tune the hyper-parameter might bias the model towards hyper-parameters that perform better on the test data than on generalized, unseen data.

3. A, B and D.

   1. True, in general larger k gives smoother decision boundary, because the testing data is less susceptible to individual points.

   2. True, to reduce the impact of noise or outliers is equivalent to prevent models from overfitting, which can be achieved by increasing k.

   3. False, if we make k too large, we could end up underfitting the data.

   4. True, cross-validation is a great way to determine hyperparameters, k in this case.

7 Perceptron

1. False. The perceptron makes two mistakes on the sequence

$$(-1, 2, -), (1, 1, +), (-1, 0, -)$$

but one mistake on the permutation

$$(1, 1, +), (-1, 2, -), (-1, 0, -)$$

2. A and B. The AND function and OR function are linearly separable - they can be learned by lines with slope -1. The XOR function, however, evaluates to negative at the first and third quadrant, and positive at the second and fourth quadrant. No linear decision boundary could be drawn in this situation.

3. $[18, 30, 63, 61]$, perceptron only updates on mistakes so add or subtract to w based on y-label and num mistakes.

4. C. According to the perceptron algorithm, we update the perceptron only if $y(\theta \cdot x) < 0$, and the update is

$$\theta^{(1)} = \theta + yx.$$

where $x = (x_1, x_2)$ such that $x_1 - x_2 = 0$ and $\theta = (3, 5)$. If we ignore the label of this dataset and only consider the set of covariates $S = \{(x_1, x_2) : x_1 - x_2 = 0\}$, we observe that for any $\theta^{(t)}$, we must have that $\theta^{(t)} - \theta \in S$.

$$\begin{aligned}(-1, 1) - (3, 5) &= (-4, -4) \in S \\ (4, 6) - (3, 5) &= (1, 1) \in S \\ (-3, 0) - (3, 5) &= (-6, 5) \notin S \\ (-6, -4) - (3, 5) &= (-9, -9) \in S\end{aligned}$$

Hence the answer is C.

5. The solution here is A and B. When the vector [-2,1] the positive side has 3 mistakes and the negative side has 2 mistakes (5 mistakes in total). For [2,-1] which is when the vector is pointing to the opposite direction, the positive side has 4 mistakes and the negative side has 4 mistakes yielding a total of 8 mistakes.

6. B. The perceptron mistake bound is given by $(R/\gamma)^2$, where R is radius of sphere that contains all points in the dataset. That is, R is the maximum distance from any point to the origin. Therefore we see that B is true.