

CS251 HW 10 | Mon May 6, 2019 | Week 14

Name:

Question 1: Decision Trees

a) Write a function to compute the entropy of an array of class counts. Test it using arrays that we would expect to give us the two extreme entropy values.

Implementation detail: You'll want to set a small non-zero minimum threshold on your probabilities otherwise you'll run into numerical stability issues when taking the logarithm.

b) Follow the ID3 algorithm to construct a **1R** decision tree by hand using the below movie viewing data. The output classification classes are in the `Will watch` column (i.e. will the customer watch movie i). Use your entropy function to help you with the calculations.

Movie	Genre	Longer than 1 hour	Twitter account mentions	Age	Will watch
1	Comedy	Yes	$10 \leq x < 20$	Old	Yes
2	Comedy	Yes	>20	Old	Yes
3	Comedy	Yes	<10	New	No
4	Comedy	No	>20	New	Yes
5	Drama	Yes	<10	New	No
6	Drama	No	$10 \leq x < 20$	New	No
7	Drama	No	<10	Old	No
8	Sci-fi	Yes	>20	Old	Yes
9	Sci-fi	Yes	<10	New	No
10	Sci-fi	No	$10 \leq x < 20$	New	Yes

c) Use your 1R tree to classify the following test data.

Movie	Genre	Longer than 1 hour	Twitter account mentions	Age	Will watch
1	Drama	No	$10 \leq x < 20$	New	No
2	Sci-fi	Yes	$10 \leq x < 20$	New	No
3	Comedy	Yes	< 10	New	No
4	Drama	No	> 20	Old	Yes
5	Drama	No	< 10	Old	Yes

d) What is the **training error**, defined as the proportion of cases misclassified in the *training set*?

e) What is the **generalization error**, defined as the proportion of cases misclassified in the *test set*?

Question 2: Neural Networks

a) Download the code for ADALINE from the CS251 website. Update the network to implement the Perceptron learning rule.

For comparison purposes, print out the raw error, which we will define as the L^2 norm (`np.linalg.norm`) of the difference between the net input (weighted sum of inputs) with the true classes (before you get the predicted output classes; this is not the misclassification rate). Test the updated network on an OR input pattern (error should converge/stabilize).

b) How does the choice of learning rule seem to affect convergence (i.e. number of iterations you run the network for the error to stabilize)?

c) Can the Perceptron learn NAND inputs (not AND)? Are any big differences compared to OR inputs?

d) Make sure that you understand the difference between ADALINE (Widrow-Hoff) and Perceptron learning rules.