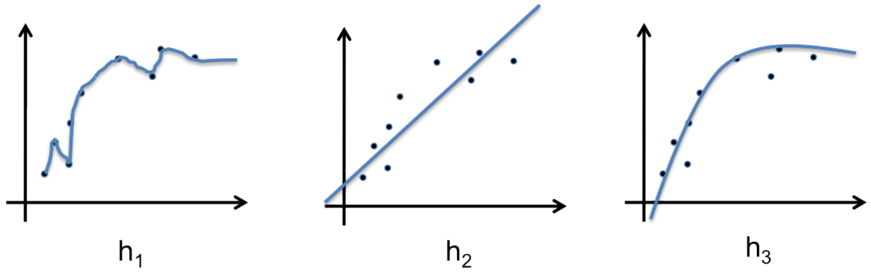
The purpose of this problem set is to gain (admittedly limited) experience with learning in AI. In the engineering design section you will train and test a decision tree classifier..

You can complete the exercises by either directly marking up this pdf, or by printing, completing, and scanning as a pdf. You should complete the Engineering Design Problems by writing the python code as instructed. The resulting pdf and python files should be uploaded to Canvas via the assignment tab by the due date and time.

Exercises

1. Consider a regression problem. Given the following training data and three hypotheses (curves).

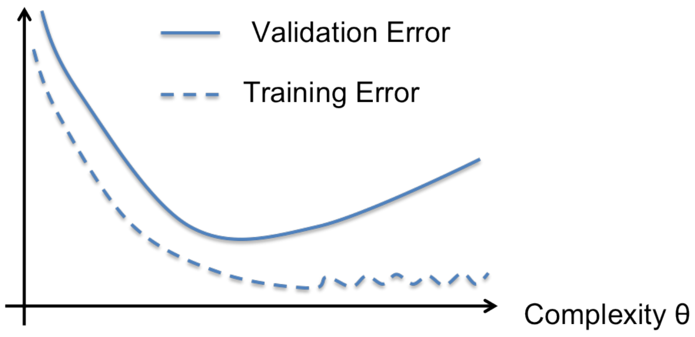


   (a) (2 points) Order the hypotheses according to their complexity.

   (b) (2 points) Order the hypotheses according to their training error.

   (c) (2 points) Which hypothesis do you think would generalize the best and why?

2. Given the following training curves for an unspecified learning algorithm with complexity parameter θ,



   (a) (2 points) Mark with an X at what point cross-validation would stop training.

   (b) (2 points) Mark with an O what complexity cross-validation would select.

3. Suppose during training of a binary decision tree there are two attributes, A_1 with possible values $0, 1$ and A_2 with possible values $1, 2, 3$. There are 10 examples with the following class labels and attribute values

class	A_1	$_2$
+1	0	2
+1	0	3
-1	1	2
+1	0	3
+1	1	2
-1	1	1
-1	1	1
-1	1	2
-1	0	1
-1	1	2

(a) (5 points) Compute the information gain for each attribute. Show your work.

(b) (5 points) Which attribute would be selected first and why?

4. Suppose a rover has an electric motor driving a wheel and a sensor that measures the rotation angle of the motor. The robot needs to learn a function that maps the measured rotation angle to a distance translated by the rover. Because of variations in manufacturing the radius of the wheel is unknown perfectly, but the specifications say it's mean is 3cm and has a standard deviation of 3mm.

   (a) (5 points) How could one cast this calibration problem as a learning problem?

   (b) (5 points) How could one design the robot to learn the mapping from rotation to translation?

Engineering Design Problems

5. (25 points) The purpose of this problem set is to gain experience with supervised learning. You will develop a python program to learn a classification rule using a decision tree given examples. This will require implementing the learning algorithm and cross-validation procedure.

Program Specification

Your program should be named class.py and implement the DECISION-TREE-LEARNING algorithm modified to build the tree breadth-first to a fixed depth, including an appropriate attribute splitting (see 18.3.6) and selection scheme (18.3.4). The program should take a variable number of command line arguments, the first being the 'mode'.

When the mode is 'train', indicating training mode, the second argument is the name of a data file containing attribute values and an associated class label in the format described below. The third argument is the name of an output test data file. The fourth argument is the name of the file storing your trained decision tree. E.g.

```
python class.py train somefile.dat test.dat tree.pickle
```

In training mode the program should read the input file and split it into a 2/3 training and 1/3 test set, having roughly the same proportion of positive and negative labels in both sets. The training set should be used to train your decision tree using leave-one-out cross-validation. The test set should be written to the specified file in the same format as the input. The decision tree should be serialized and written to the specified file. I recommend using json or pickle for this. Your program should also print to stdout the final training error rate.

When the mode is 'test', the second argument is the file name of a test data file and a file containing a decision tree, both generated in 'train' mode. E.g.

```
python class.py test test.dat tree.pickle
```

In test mode your program should load the test data file and the decision tree, then test the decision tree using the class label as the ground truth. The program should print the test error rate to stdout.

Datasets You will likely want to create simple test datasets for code debugging. The dataset we will use as a benchmark is the "Adult Data Set" from the UCI Machine Learning Repository (`http://archive.ics.uci.edu/ml/datasets/Adult`). The attributes of this dataset are both categorical and integer numbers derived from census data with some missing data. The data file (`adult.data`) is a csv file, one example per line as described on the website and in the file `adult.names`. The class label is the final field and is the income group to be predicted.

Analysis

Perform the train/test cycle on the dataset provided and summarize your performance results in a plain text file named README.txt.