COMPSCI 270 **Instructor:** Kris Hauser

Homework 5 **Due:** 04/12/2017

1 Instructions

Read this assignment carefully, complete the programming assignment and answer all of the written questions. Place all of your code in the `hw5.py` file and your written answers in the `hw5_answers.py` file in the appropriate locations. Alternatively, you may submit your answers to the written questions in a PDF file called `hw5_answers.pdf`. Submit all of these files on Sakai before 4:20pm on 04/12/2017.

1.1 Homework Policies

Homework and lab assignments **must be completed individually**. Students are permitted and even encouraged to discuss assignments. However, any attempt to duplicate work that is not your own – for example, in the form of detailed written notes, copied code, or seeking answers from online sources – is strictly prohibited and will be considered cheating.

Homework assignments are due by the end of class on the due date unless otherwise specified. **No extensions will be granted** except for extenuating circumstances. Extensions are granted at the instructor's discretion, and valid extenuating circumstances include, for example, a debilitating illness (with STINF), death in the family, or travel for varsity athletics. Extensions will not be granted for personal or conference travel, job interviews, or a heavy course load.

2 Overview

In this assignment, you will be creating and using decision trees and random forests for image classification problems. The dataset consists of images from Project Malmo of three different animal types (Pig, Cow, Sheep). It is your job to build a learning model to predict the animal type.

3 Preparation

For this assignment, you will need some additional Python packages that you might not have installed previously. Instructions for installing each package with the `pip` utility included with Python. These commands will also update the package if you already have an earlier version installed.

1. NumPy and SciPy are a scientific computing packages.
   (a) Linux/Mac

i. `pip install -U numpy`
ii. `pip install -U scipy`

(b) Windows

i. Visit `http://www.lfd.uci.edu/~gohlke/pythonlibs/#numpy` and download the numpy-1.12.1 file for Python 2.7 ("cp27") and the architecture which corresponds to your Python installation (32-bit vs. 64-bit).
ii. Visit `http://www.lfd.uci.edu/~gohlke/pythonlibs/#scipy` and download the appropriate scipy-0.19.0 file.
iii. Install NumPy from the file you downloaded by following the instructions at `https://pip.pypa.io/en/latest/user_guide/#installing-from-wheels` for "installing directly from a wheel archive".
iv. Install SciPy from the file you downloaded.

2. scikit-learn is a machine learning package

(a) `pip install -U scikit-learn`

Once you have done this, you can verify that these dependencies are installed correctly by running the `dependency_test.py` script provided with the assignment with `python dependency_test.py`. It will indicate which, if any, package(s) are not working.

4 Problems

4.1 Programming

For this section, you will be creating classifiers, comparing their effectiveness at classifying data, and observing how their performance changes with different parameters. You will be using a dataset containing many images captured from Minecraft. The driver module for this assignment is `hw5.py` and it can be run as follows (where problem can be *p0*, *p1*, *p2*, or *p3*):

- `python hw5.py <problem>`

4.1.1 Problem 1

The training data is given as a list of examples with class labels 0 (Pig), 1 (Cow), and 2 (Sheep). The `p0()` function already provides a decision tree classifier that automatically uses multi-class classification, and prints its classification accuracy on the test set. Complete this evaluation by computing the confusion matrix. Then, learn three new one-vs-one decision tree classifiers: the first to distinguish between labels 0 and 1, the second to distinguish between 0 and 2, and the third to distinguish between 1 and 2. Report the test set accuracy of each of these classifiers.

4.1.2 Problem 2

The raw images have a large number of pixels, so there are three different types of data preprocessing steps. First, a histogram of the pixels' (H,S) values in Hue-Saturation-Value (HSV) space is computed on a 16x16 grid, giving 256 features. Second, we have a version of the color image resized to 30x20, giving 1,800 features. Third, we have a grayscale version of the resized image,

giving 600 features. By default, the histogram features are used. In `p1()`, you should compare the effect of different features on the decision tree learner. Record the results.

4.1.3 Problem 3

Complete the `p2()` function. This function works similarly to p0() with one major difference - it fits the model to subsets of the full training set and checks the accuracies of the model for each size (50, 100, 150, 200, 250, 300).

4.1.4 Problem 4

Complete the `p3()` function. This function uses the whole training set for the digits and a random forest classifier instead of a decision tree classifier. It also tests how the number of estimators (trees) used in the random forest affects the accuracies of the model on the training and test sets.

4.2 Written

4.2.1 Problem 1

List the results from `p0`. What pair of classes was most often confused, in the confusion matrix? What pair of classes had greatest error in the one-vs-one tests? Explain the discrepancy in the results.

4.2.2 Problem 2

List the resulting accuracies of different feature types from `p1`. Which feature performs the best? Why do you think this is so?

4.2.3 Problem 3

Provide tables of the accuracy results from `p2` and `p3`. What trends do you notice in the accuracies for the different sizes of the training set in `p2`? What trends do you notice in the accuracies for the different numbers of estimators in `p3`?