COSC 311 - Lab 6

Dr. Joe Anderson

Due: 19 November

1 Objectives

In this lab, we will

1. Work on developing some simple supervised machine learning algorithms
2. Practice visualizing and presenting the behavior of a complex algorithm on data
3. Continue developing skills with Python, Jupyter notebooks, and data-processing libraries

2 Tasks

1. You may submit this lab in groups of one or two.
2. Begin by implementing the k-Nearest Neighbors (kNN) classification algorithm
   (a) Use a Python class with internal parameter k and methods `train` and `predict`.
      i. The `train` method should take in an array of pre-labeled data and store them to be used in the kNN calculation later.
      ii. The `predict` method should take in a single data point (of the same dimension as those used in training) and return the label of the kNN decision. Keep in mind that, in general, you may have to choose between multiple labels. In this case, ties are possible, but you can just decide to break them randomly or increase k to get more votes.
   (b) Test the algorithm on some simple hand-made data that you can easily visualize or inspect to check for correct behavior.
3. Write a python routines to split a given data set (randomly) into testing and training subsets. It's a good idea to parameterize this process with a percentage p so the data is split into sets of size np and $n(1-p)$; then you can "tune" p later for your specific training/testing procedure.
4. Using Python, implement the k-means algorithm and test this on the Iris dataset. Try using all of the columns versus only a subset. What works the best? How well does the algorithm do when setting $k = 3$, the true number of classes? Show the results visually (using only 2-d scatter plots) – be sure to indicate visually both the predicted cluster result and the true iris class.
5. Using the iris dataset, test your kNN algorithm to predict which class a given iris belongs to.
   (a) Show some confusion matrices for different size testing versus training datasets.
   (b) How does the accuracy change for different size training sets? Show how the accuracy changes within the training set and testing set separately.
   (c) What if you repeat this process, keeping the training fraction p the same? Does the accuracy change drastically based on the "quality" of the testing set? How might you quantify this behavior?

6. Try using the adult dataset to also do some prediction of income level (more or less than $50k).

   (a) How can you try to quantify the "distance" between two adults? Write some small python functions to try and transform this data into something that can be used by the kNN classifier. Try a couple different approaches, report their different behaviors and illustrate using learning curves, confusion matrices, and possibly some statistical measurements.

   (b) Are there other "labels" within the dataset that can be effectively learned by the kNN classifier?

3 Submission

Submit any notebooks, Python files, and documentation to the MyClasses assignment page.