Homework 4

Nishant Subramani

I.

Dataset is a univariate dataset (one predictor X1) with values 0, 1, 2.01.

Hierarchical clustering will always group the 0 and 1 together and keep the 2.01 in its own cluster due to the distance between 0 and 1 being the smallest. Sequential clustering will group them in the same manner, 0 and 1 together, keeping 2.01 in its own cluster with this ordering, but when reordered such that the attributes are given 2.01, 1, 0, since the threshold is 1.1, it will group 2.01 and 1 together, leaving 0 on its own. This shows that sequential clustering is very susceptible to ordering, while hierarchical clustering is not.

II.

a. K means clustering fails to work on nominal/categorical data for a few reasons. One since the sample space for categorical or nominal data is discrete, there is no natural origin or place to calculate a distance metric from. Given this, calculating Euclidean distance or any other kind of distance in a Cartesian space is relatively futile as there is not any sort of spaced distribution. Assigning a cluster mean as utilized in k-means that is between 0 and 1 tells us nothing about the cluster's characteristics or the cluster itself. If the attributes were categorical based on what airline you flew and United was 0, and Lufthansa was 1, a cluster value of 0.9 says absolutely nothing. Your plane was not 90% Lufthansa and 10% United. The other clustering algorithms group like examples and thus get around many of these characteristics and perform significantly better on nominal/categorical data.

b. The common strategies for adapting k-means for nominal data include using a different distance metric, such as an association metric (Jaccard or Tanimoto) and using modes instead of means to define a cluster which would entail finding the most common occurrence for each of the attributes. The mode strategy would mean the class would be defined by the most common occurrence which would mean something and signify some value that would embody the class. The similarity and association metrics would allow similar attributes to be classified close together. The Tanimoto metric allows the triangle inequality to hold true, such that two examples are similar if they are similar to a third example but not to each other. This allows for k-means to cluster nominal attributes in a better way.

III. First, SVMs use kernels. This allows the SVM algorithm to project onto a higher-dimensional space in which data may be better separable, while the perceptron assumes that in its current dimensional state, the training data is separable. Second, SVMs maximize margin.

This is helpful because SVMs find the best hyperplane that separates the data, which in turn, provides the best possible hypothesis given some data and ideally provides the best performance on the validation/test set. The perceptron simply finds a hyperplane that separates the data, so for an example where the true function is y = x, and data is only found in the II and IVth quadrants, the perceptron may just say that y = 0 perfectly splits the data. This would be a terrible hypothesis, while the SVM would see that something similar to y=x maximizes separation and thus would choose something similar to that.

IV. Description of the Dataset and how it was generated is provided in the comments in the .py file in the generate_num_4 function.

Algorithm	% Accuracy 10-Fold CV	Mean Absolute Error
IBk	72.4%	0.3049
J48	100%	0.0

Extra Credit:

Description of the Dataset and how it was generated is provided in the comments in the .py file in the generate_ec function.

Algorithm	% Accuracy 10-Fold CV	Mean Absolute Error
Multilayer Perceptron	100%	0.0
Naïve Bayes	40.2%	0.496