CS 440 Homework 8: k-Means
Extra Credit Due Date: April 30, 2017 11:59pm
Final Submission Due Date: May 2, 2017 11:59pm

k-means Algorithm

As discussed in class, k-means clustering is a simple but useful unsupervised learning algorithm. Given a dataset of unlabeled examples, a distance metric, and a positive integer k, the algorithm attempts to partition the examples into k clusters. A cluster is defined by the centroid (average according to the distance metric) of its constituent examples.

The k-Means algorithm

1) Select an integer k and a distance measure.
2) Select k initial locations for the cluster centroids.
3) Assign each example to the nearest centroid with ties broken randomly. This partitions the examples into clusters.
4) Re-compute each cluster centroid as the mean of its constituent examples.
5) Repeat steps 3) and 4) until either:
    a. some pre-specified number of iterations has been reached or
    b. the maximum distance moved by any centroid is less than some pre-specified tolerance value

Since the algorithm is greedy, the result will be only a local optimum. Note also that it is possible for a cluster to vanish.

For this homework, you will experiment with at least three different k's (k=3 and two others of your choice). For the distance, use Euclidean over the example features. You are free to select any strategy you like for the initial cluster locations. But this should involve some randomization (e.g., randomly selecting k of the examples as initial locations). You will stop after these pre-specified numbers of iterations: 5, 10, 20. Repeat each clustering four times. You will report each loss, but save only the best of the four. For a given k, the best clustering minimizes the total un-normalized variance given the clusters, SS_total: the sum of the squared distances from each example to its cluster's centroid. If a cluster vanishes, remove it and proceed as if k were reduced by one.

We will be clustering the observations in the *iris* dataset found here https://archive.ics.uci.edu/ml/datasets/Iris. You must read the description of the dataset before starting to code the algorithm. Briefly, each example in the dataset consists of 4 numeric features i.e. Sepal Length, Sepal Width, Petal Length, Petal Width. It also has a label for each example i.e. one out of Iris Sentosa, Iris Versicolour, Iris Virginica. We will ignore the label in running the k-means algorithm but will use it to evaluate the quality of your clusters in Part B.

Part A:

Since there known to be three clusters your two additional k choices must be greater than three. Thus, for k=3. You will repeat the k-means algorithm (with random initialization) four times, stopping the algorithm each time after 5 iterations. Record the four SS_total scores and save the best clustering of the four.
This process is then repeated with a stopping condition of 10 and 20. This will yield 12 SS_total scores and three saved clusterings (one for each stopping condition).
You will then repeat this entire procedure for your other two choices of k.

Present your SS_total results in one or more graphs (of your choice) for each k. Together they should clearly show how the range of scores and the best score for each k are affected by the choice of k and by the stopping condition.

Explain your findings.

Should we allow our preference for low SS_total to influence our choice of k? Explain.

Part B:
Because the class labels for this problem are known, your saved clusterings can be objectively evaluated. For this you will use the F1 score.
For each saved clustering, choose three *distinct* clusters as the "correct" ones (one for each iris class). You may do this any way you like but explain your procedure (greatest number of examples of that iris type, highest percentage of the iris type, etc.). Also explain and resolve any complications that you encounter in this.

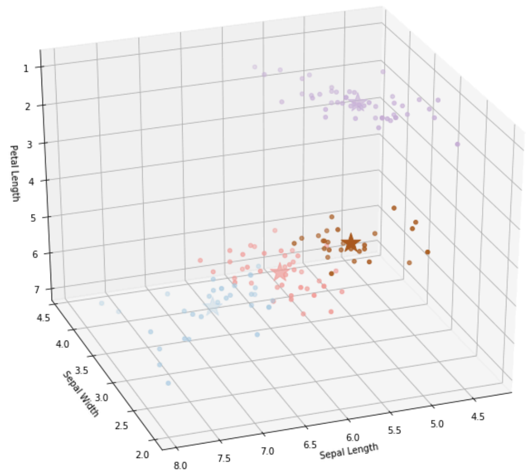
The F1 score is based on Recall and Precision. Recall captures how many of the desired items were found; Precision captures the dilution of the desired items by undesired ones. Given a clustering and an identification of the "correct" cluster, compute three F1 scores, one for each species of iris:
Recall = # correct species in the correct cluster / # that species in the database
Precision = # correct species in the correct cluster / # examples in that cluster
F1 is their harmonic mean:
F1 = 2 * Precision * Recall / (Precision + Recall)

For each saved clustering report each species' F1 score and the average F1 of the clustering: the species-specific F1's weighted by the proportion of each species in the database.

Explain your results.

Report your best average F1 clustering (i.e., give the centroids) and describe how you obtained it. Display the clustering with four 3D plots. You may wish to use
http://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html
or any plotting technique of your choice. Be sure to clearly label the axes. To differentiate between a regular data point and a cluster centroid, mark the centroid with a different symbol. Here the cluster is denoted by color with the centroid a star and all the data examples as a

circle. In this example, k=4. You may wish to use three different symbols for the different iris species to better understand the score of your clustering.



Each example has 4 features, we will be plotting them 3 at a time. Report your plots for the following:

a. Sepal Length vs Sepal Width vs Petal Length
b. Sepal Length vs Sepal Width vs Petal Width
c. Sepal Length vs Petal Length vs Petal Width
d. Sepal Width vs Petal Length vs Petal Width