# CS 470

# Data Mining

# Homework 3

Jason Ji

Collaborations: Didn't collaborate with any other students.

## Dataset Description

I used two data sets for this assignment. The first data set is the given 'iris.data', located at 'http://archive.ics.uci.edu/ml/datasets/Iris'. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. The specific attributes are shown below:

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm
5. class:
– Iris Setosa
– Iris Versicolour
– Iris Virginica

The second data set is 'buddymove_holidayiq.csv', located at 'https://archive.ics.uci.edu/ml/datasets/BuddyMove+Data+Set'. This dataset was populated from destination reviews published by 249 reviewers of holidayiq.com till October 2014. Reviews falling in 6 categories among destinations across South India were considered and the count of reviews in each category for every reviewer (traveler) is captured. The specific attributes are shown blow:

Attribute 1 : Unique user id
Attribute 2 : Number of reviews on stadiums, sports complex, etc.
Attribute 3 : Number of reviews on religious institutions
Attribute 4 : Number of reviews on beach, lake, river, etc.
Attribute 5 : Number of reviews on theatres, exhibitions, etc.
Attribute 6 : Number of reviews on malls, shopping places, etc.
Attribute 7 : Number of reviews on parks, picnic spots, etc.

## Data Pre-processing and Implementation

To only consider the numerical attributes, I first had to pre-process the data by removing the non-numerical attributes. For the 'iris.data', I removed the last column/attribute, which is the class attribute that is categorical. For the 'buddymove_holidayiq.csv' data, I removed the first column, which is the user id attribute. In addition, I had to remove the first row for the 'buddymove_holidayiq.csv' data because it is the heading for the data set. (data pre-processing for the second data set is commented out in the submitted code).

Here is my implementation of the K-means algorithm:
I first randomly selected k data points as my initial centroids. Then I assign each data entry in the original database to the nearest centroid based on euclidean distance, thus forming the initial random clusters. Then I used a while loop to repeat the process of calculating centroids and reassigning data entries until the centroids become constant. Throughout the process, I kept track of the data entries and the cluster to which they belong, as well as the centroid information for each cluster.

## Experiment and Results

| Dataset | iris.data | | | |
|---|---|---|---|---|
| Total number of data entries | 150 | | | |
| k | 3 | 5 | 7 | 9 |
| SSE | 78.95 | 49.92 | 37.34 | 35.7 |
| Average Silhouette coefficient | 0.55 | 0.46 | 0.31 | 0.27 |

| Dataset | buddymove_holidayiq.csv | | | |
|---|---|---|---|---|
| Total number of data entries | 250 | | | |
| k | 3 | 5 | 7 | 9 |
| SSE | 783419.90 | 565919.10 | 449346.33 | 374750.71 |
| Average Silhouette coefficient | 0.35 | 0.27 | 0.29 | 0.28 |

## Experience and Lessons Learned

During the experiment, I tested the effects of different initial k values on the final clustering results using the sum of squared error and average silhouette coefficient. And here are the insights we can draw from the experiment results:

1. As k increases, the sum squared error of the final clustering decreases. This is shown in both data sets as SSE for iris.data decreases from 78.95 to 35.7 as k increases from 3 to 9, and SSE for buddymove_holidayiq.csv decreases from 783419.9 to 374750.71 as k increases from 3 to 9. This makes sense because as the number of cluster increases, each cluster is smaller and each data point is closer to the centroid.

2. As k increases, the average silhouette coefficient decreases. We know that when the silhouette coefficient value of 0 approaches 1, the clusters are more compact, and each cluster is far away from other clusters, which is the preferable case. This means that for the two data sets examined, a smaller k is more preferable because it produces a larger average silhouette coefficient, meaning the resulting clusters are more compact and distant from other clusters. This could also be due to the fact that more clusters mean closer distances between clusters.

From this experience, I realized that it is very important to plan out the data structures before coding

by thinking about what information do we need to store throughout the process. In my code, I create the data id for each data entry, map each data id to its corresponding cluster id, and map the cluster id to the cluster centroid. However, I didn't realize that I would need to store all this information at the beginning, which resulted in me going back to the beginning and re-creating additional data structures.

In terms of the k means algorithm, I saw that different k values and initial clustering would produce distinct clustering results at the end. Therefore it is important for us to do experiments and examine the end result to determine the best k value, as well finding a way to choose the best initial clustering rather than choosing randomly (which is what I did for this assignment).