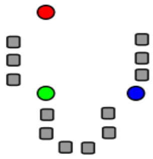
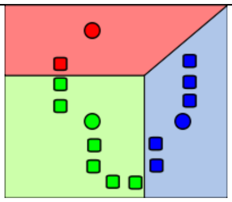
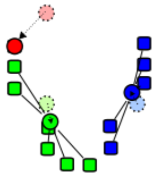
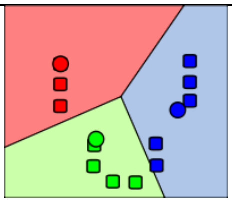
Adam Mischke
Dr. Phillips
CSCI 4350 – Intro to Artificial Intelligence - Open Lab 4
Mon. December 4, 2017

Unsupervised Learning

Introduction:

In this lab, we used an unsupervised learning method called **k-means clustering**. K-means attempts to learn the pattern structure of a data set without the use of its **classification or class label**. Without the labels of each class for each set, means that this algorithm learns only by the numbers surrounding it. To elaborate, k-means can be viewed as a data mining algorithm, iteratively mining useful hidden information out of the data that we give it. Specifically, K-means uses K **clusters means**, or any number of point averages, to determine the classification of each group. By moving the cluster means closer to the center of all of the points assigned in its own cluster, we see that each cluster mean will eventually stop moving and that gives us convergence. Wikipedia had excellent simple illustrations of this divided into 4 basic steps:

k-means clustering

 P1[1]	 P2[1]	 P3[1]	 P4[1]
Step 1: Out of our training data points, pick K points as our cluster mean. (1 – However many points we have) In this case, 3.	**Step 2:** Find the distance between the cluster means and every other point and assign color tags to the closest ones.	**Step 3:** Move the cluster means to be the average of the points with its associated color tags.	**Step 4:** Repeat step 2 and 3 until the cluster means stop moving.

Problem:

Now that we have a better idea of what k-means is about, let's use it! **T1:** First, we have a set of data on Tennessee's state flower, the Iris flower. The first training set consists of observations which contain 4 attributes: how long and wide the sepal and petal measures. The last column is the class label column and Iris has 3 classifications (what kind of Iris it is). That makes 5 Columns. We have 150 observations of the Iris data set each 50 evenly set across all 3 classifications. **T2:** Secondly, we have a more complicated Cancer set, consisting of 95 sets of Breast Cancer data. These sets have 9 attributes, most of which won't make sense unless you're a Biology major and a classification column that has 6 different possibilities of Breast Cancer. A breakdown of these numbers is below at **T0**:

T0

	# of observation and recorded entries (rows of training data)	# of attributes	# of classifications (class labels)	# of columns (# of attributes + one class label)
Iris data set	150	4	3	5
Cancer data set	105	9	6	10

Solution:

My Solution was to write a script to automate randomly picking rows to train and test on. We always tested on 10 and trained on however many we had left in the data set. By running K clusters (1 through to the # of training rows) each 100 times, we can calculate the average of the correct classifications out of 10. My algorithm implemented k-means in this relative fashion:

T1

Iris set attributes
1. Sepal length in cm
2. Sepal width in cm
3. Petal length in cm
4. Petal width in cm
5. Classifications: 1) Iris Setosa 2) Iris Versicolour 3) Iris Virginica

T2

Cancer set attributes
1. I0 Impedivity (ohm) at zero frequency
2. PA500 phase angle at 500 KHz
3. HFS high-frequency slope of phase angle
4. DA impedance distance between spectral ends
5. AREA area under spectrum
6. A/DA area normalized by DA
7. MAX IP maximum of the spectrum
8. DR distance between I0 and real part of the maximum frequency point
9. P length of the spectral curve
10. Classifications: 1. car(carcinoma) 2. fad (fibro-adenoma) 3. mas (mastopathy) 4. gla (glandular) 5. con (connective) 6. adi (adipose)

Algorithm:

```
$done = false$

$While\ done\ != true$:
    $For\ s\ in\ sets\ of\ attributes$:
        $For\ c\ in\ cluster$:
          $D = distance(c, s)$
          $if\ d < lowestD$:
              $lowestD = d$
    $//\ returns\ true\ if\ clusters\ don't\ move$
    $done = updateClusters()$
```

Data:

T3

K=1	K=2	K=3	K=4	K=5	K=6	K=7	K=8	K=9	K=10
Correct right in the Iris set in %									
18.9	59.8	83.8	86.6	87.8	90	91.4	93.5	93.9	94.9
Correct right in the Cancer set in %									
12	18.9	26.5	25.8	37.7	33.5	43.1	42.4	42.7	43.9

T3: Here, we can see a brief view of the averages of when using K clusters, where if K=1, that means we only used one cluster mean. We can immediately see a difference in success in classification just by looking at the first 10 cluster means- Iris is significantly better at classifying than the data set. Why is that? For one, the cancer set has a whopping 6 classifications to decide on whereas Iris has only 3. We can also see that the Iris data has 150 training rows, where the Cancer set only had 105. Cancer not only had less training, but it had to attempt to classify between twice the number of Iris classifications.

FIG A

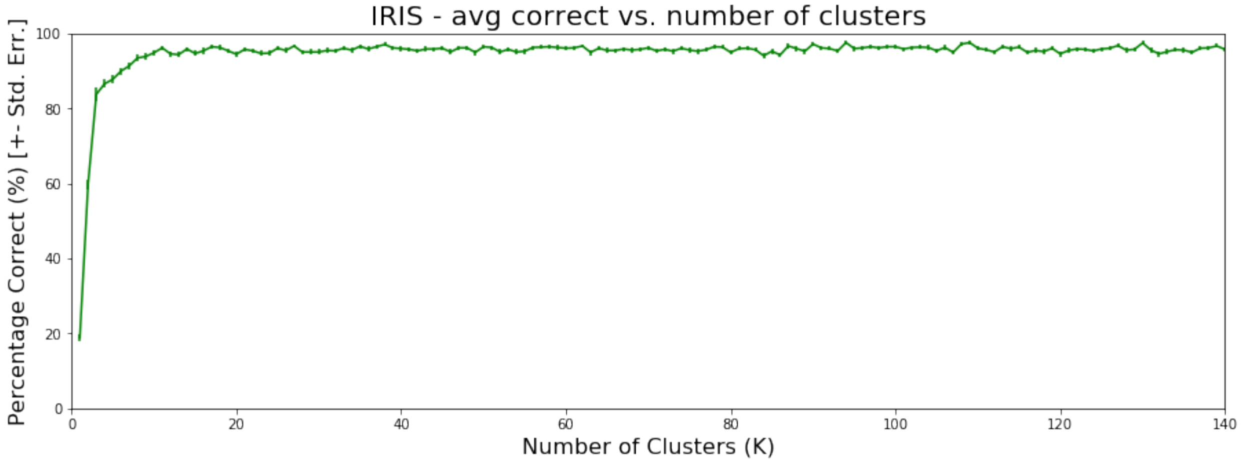


FIG B

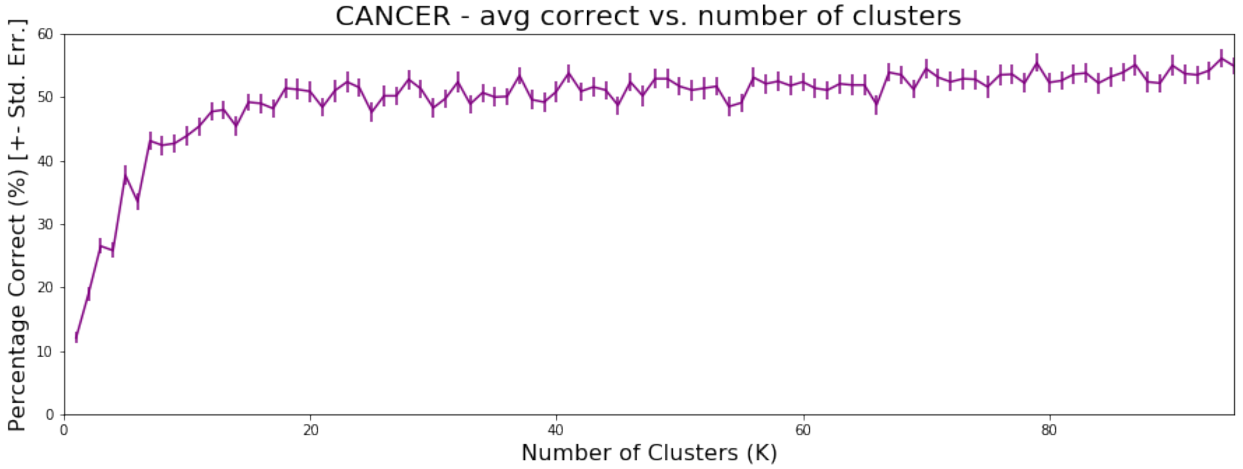
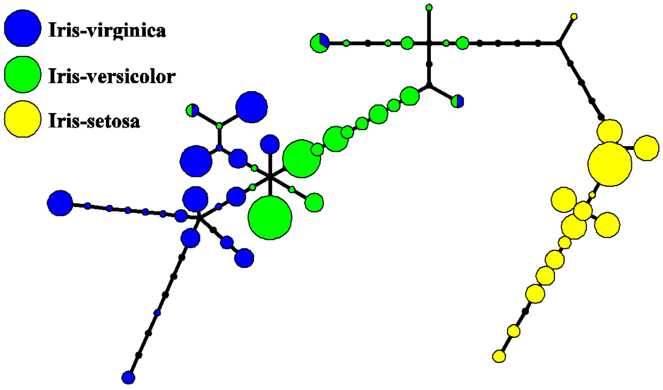


FIG A & B: In these two graphs, we can see all the way to the maximum clusters for the training set: Iris = (150-10) = 140 and Cancer = (105-10) = 95 K cluster means. This tells us that we can stop at about K=10 cluster means for the Iris set to get about 95% right classifications. The Cancer, however, looks as if it could continue to classify better if we gave it more clusters, but we can't because of our limited amount of data that we feed into it. Even at 95 cluster means, we only get around 65% right classifications between the 6 options of Breast Cancer.

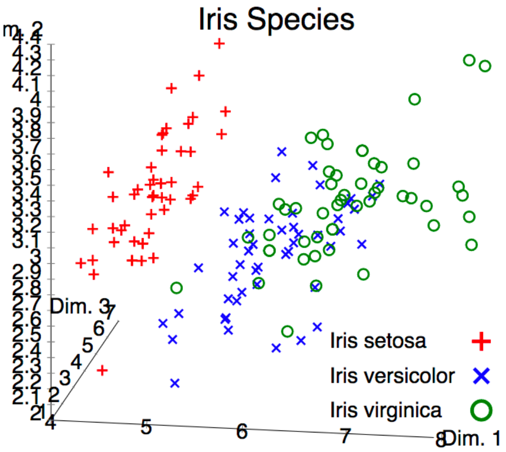
<u>Analysis:</u>

I found that in the Iris data set, out of the 3 classifications, Iris Setosa (0) classified almost perfectly. Contrastingly, Iris Versicolour (1) and Iris Virginica (2) went back and forth between classifying wrong. After some research, I found that this is part of Ronal Fisher's data set and that this a great model for showing, **P5 & P6,** which correlates with what we saw in the data above. Furthermore, we should be asking, how were we able to distinguish the clusters and assign them to their class labels? This was solved by going through the training data set (ours of which had the correct class labels) and taking a majority vote that the cluster with maximum correct in each cluster was the right class label.

P5[2]



P6[2]



As for our Breast Cancer data set, I think it is the antithesis of the Iris data set in that it's this set shows you how it may be hard to classify data. In this[3] article from the US Nation Library of Medicine National, they use different methods such as Manhattan and Pearson Distance (we used Euclidean), although Pearson is supposedly worse. They also had different methods in how you base the centroid, threshold, and split, which attributes to use and how many iterations. Using different cases for certain types, they achieved 92% average positive predictions of Breast Cancer in Wisconsin (with their special learning set).

Then you have these guys[4] from the Department of Systems Science and Industrial Engineering, State University of New York. They increased the Wisconsin data set from 92% correct to 97.38%. Their premise also included that radiologists and physicians used to predict tumor types. Unfortunately, 90% of the radiologists recognized fewer than 3% of cancers. This is why we use computers that can-do millions of clocks a second to recognize these patterns better than us. The article also goes into Support Vector Machines (SVM) which is a type of supervised learning method that they used to diagnose cancer. SVMs can cut out whole dimensions of features to speed up and make the learning more accurate by choosing which features to use. With their K-SVM (a hybrid of SVM & k-means), their classifier correctly classified between benign and malignant tumors with a 97.38% accuracy and 6 feature dimensions instead of their baseline, 30. This sped up the program from 15 seconds to 0.0004 seconds! I don't understand all of their terminology, but I can definitely read more into these papers.

Conclusion:

K-means is nice, but it can't solve everything. Not only that, but higher dimensions can cause the learning process to take a long time. The types of algorithms like in citation 3 and 4 will be able to accurately digest large sets of data and cut them to their core, slicing off all of the correlations and keeping only the causations. In other words, we trim all of the data we don't need to use to classify AND make it more accurate! As for the Iris data set, sometimes it all comes down to having the right data to work with.

Citations:

1	https://en.wikipedia.org/wiki/K-means_clustering	**P1 - P4**
2	https://en.wikipedia.org/wiki/Iris_flower_data_set	**P5, P6**
3	**https://www.ncbi.nlm.nih.gov/pubmed/27311823**	**92%**
4	**http://www.sciencedirect.com.ezproxy.mtsu.edu/science/article/pii/S0957417413006659**	**97%**