10.5.1 K-Means Clustering

May 18, 2018

0.1 10.5 Lab 2: Clustering 10.5.1 K-Means Clustering

```
In [1]: # conventional way to import pandas
        import pandas as pd
        # conventional way to import seaborn
        import seaborn as sns

        from sklearn.cluster import KMeans
        import numpy as np
        import matplotlib.pyplot as plt
```

Next we will set the random seed and generate the data.

```
In [2]: np.random.seed(0)
        X = np.random.standard_normal(size=(25,2))

        X.shape
```

```
Out[2]: (25, 2)
```

We will change our data a little so the first columne will have a mean of +3 and the secound is -4

```
In [3]: for row in X:
            row[0] = row[0]+3
            row[1] = row[1]-4
```

Now we will use the k-means algo. We will do two clusters and make it do 20 iterations. We display the labels, cluster_centers and the sum of distances of samples to their closest cluster center also called inertia.

```
In [4]: kmeans2 = KMeans(n_clusters=2, random_state=0,n_init=20).fit(X)
        print( kmeans2.labels_ )
        print( kmeans2.cluster_centers_ )
        print( kmeans2.inertia_ )
```

```
[0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 1 1 1 1 1 1]
[[ 3.91945474 -3.78502692]
 [ 1.85475744 -4.09063316]]
38.18401647381171
```

Now we will run the algo again with k=3 and that means find 4 clusters.

```
In [5]: kmeans3 = KMeans(n_clusters=3, random_state=0,n_init=20).fit(X)
        print( kmeans3.labels_ )
        print( kmeans3.cluster_centers_ )
        print( kmeans3.inertia_ )
```

```
[0 2 0 0 1 2 0 0 0 0 1 0 0 1 2 1 1 1 2 1 1 1 1 1 1]
[[ 4.19199303 -4.39210147]
 [ 2.149192   -4.01784344]
 [ 3.97146286 -2.40827367]]
27.556976178858864
```

Next we show the two plotted

```
In [6]: fig, (ax1, ax2) = plt.subplots(1,2, figsize=(14,5))

        ax1.scatter(X[:,0], X[:,1], s=40, c=kmeans2.labels_, cmap=plt.cm.prism)
        ax1.set_title('K-Means Clustering Results with K=2 with 20 iterations')
        ax1.scatter(kmeans2.cluster_centers_[:,0], kmeans2.cluster_centers_[:,1], marker='+', s

        ax2.scatter(X[:,0], X[:,1], s=40, c=kmeans3.labels_, cmap=plt.cm.prism)
        ax2.set_title('K-Means Clustering Results with K=3 with 20 iterations')
        ax2.scatter(kmeans3.cluster_centers_[:,0], kmeans3.cluster_centers_[:,1], marker='+', s

        plt.show()
```