08B-Clustering-III

October 3, 2017

1 Hierarchical Clustering

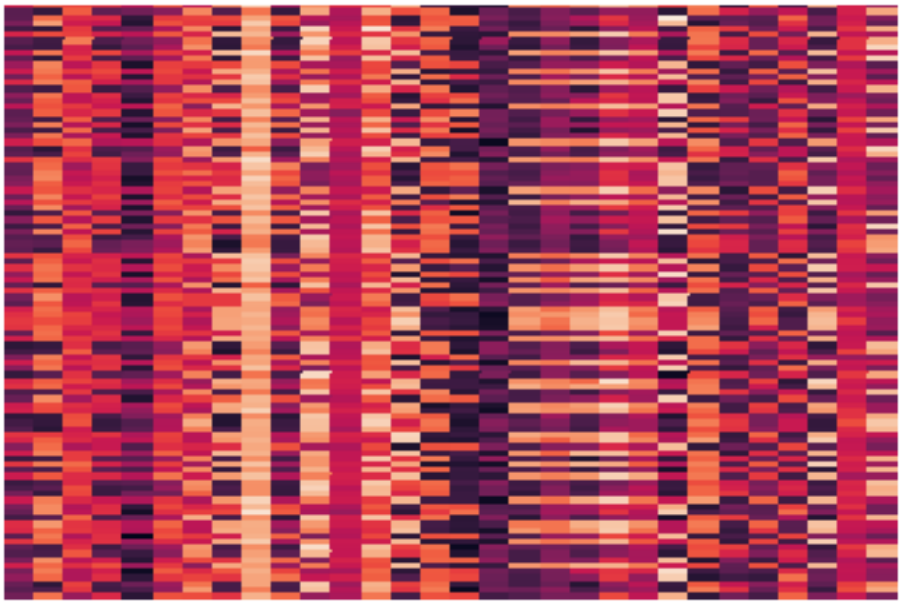
1.1 Synthetic data

We'll use the same synthetic data as we did in the k-means case -- ie., three "blobs" living in 30 dimensions.

```
In [102]: X, y = sk_data.make_blobs(n_samples=100, centers=3, n_features=30,
                                    center_box=(-10.0, 10.0),random_state=0)
```

As a reminder of the raw data here is the visualization.

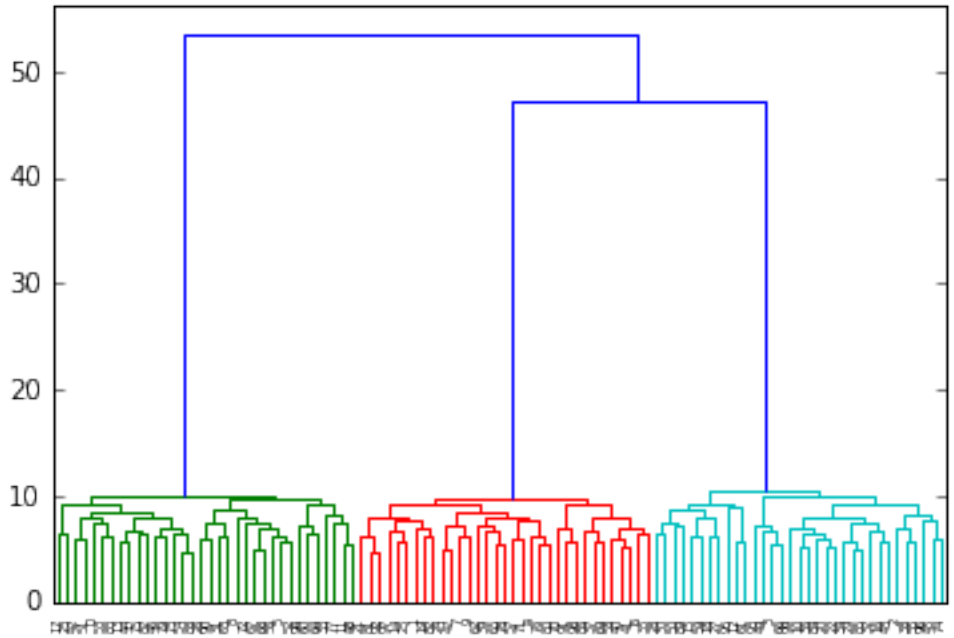
```
In [103]: _ = sns.heatmap(X, xticklabels=False, yticklabels=False, linewidths=0,cbar=False)
```



Hierarchical clustering is available in **sklearn**, but there is a much more fully developed set of tools in the **scipy** package and that is the one to use.

```
In [104]: import scipy.cluster
          import scipy.cluster.hierarchy as hierarchy
          import scipy.spatial.distance
          # linkages = ['single','complete','average','weighted']
          Z = hierarchy.linkage(X,method='complete')

In [105]: R = hierarchy.dendrogram(Z)
```



1.2 Working with real data

Once again we'll use the "20 Newsgroup" data provided as example data in sklearn. (http://scikit-learn.org/stable/datasets/twenty_newsgroups.html).

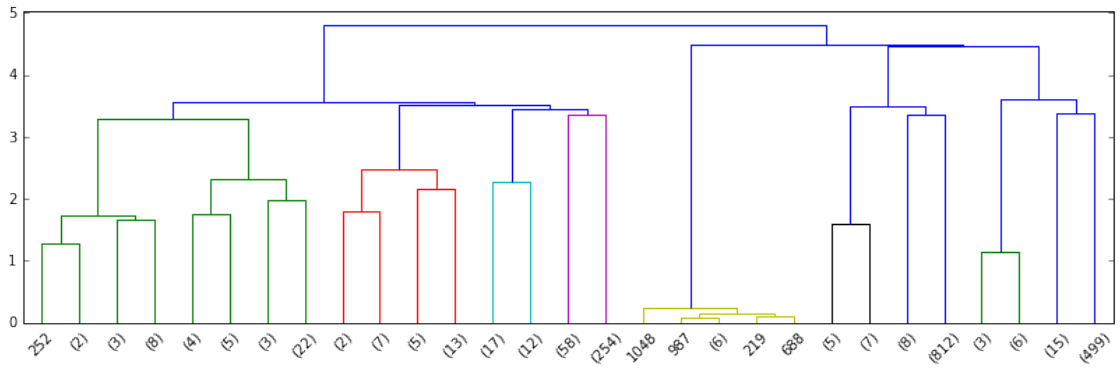
```
In [106]: from sklearn.datasets import fetch_20newsgroups
          categories = ['comp.os.ms-windows.misc', 'sci.space','rec.sport.baseball']
          news_data = fetch_20newsgroups(subset='train', categories=categories)

In [107]: from sklearn.feature_extraction.text import TfidfVectorizer
          vectorizer = TfidfVectorizer(stop_words='english', min_df=4, max_df=0.8)
          data = vectorizer.fit_transform(news_data.data).todense()
          data.shape

Out[107]: (1781, 9409)

In [108]: # metrics can be 'braycurtis', 'canberra', 'chebyshev', 'cityblock', 'correlation', 'c
          # 'dice', 'euclidean', 'hamming', 'jaccard', 'kulsinski', 'mahalanobis', 'matching',
          # 'minkowski', 'rogerstanimoto', 'russellrao', 'seuclidean', 'sokalmichener', 'sokalsn
```

```
        # 'sqeuclidean', 'yule'.
        Z_20ng = hierarchy.linkage(data, method='ward', metric='euclidean')
        plt.figure(figsize=(14,4))
        R_20ng = hierarchy.dendrogram(Z_20ng, p=4, truncate_mode='level', show_leaf_counts=Tru
```



1.2.1 Selecting the Number of Clusters

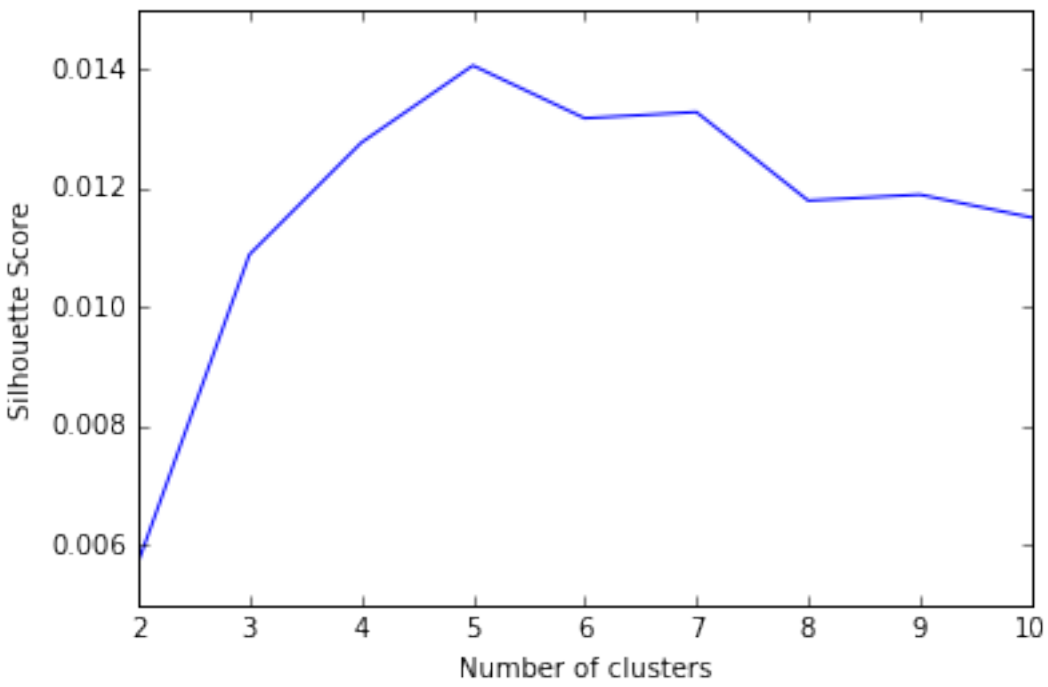
```
In [109]: clusters = hierarchy.fcluster(Z_20ng, 3, criterion='maxclust')
          print(clusters.shape)
          clusters
```

```
(1781,)
```

```
Out[109]: array([3, 3, 3, ..., 1, 3, 1], dtype=int32)
```

```
In [110]: max_clusters = 10
          s = np.zeros(max_clusters+1)
          for k in range(2,max_clusters+1):
              clusters = hierarchy.fcluster(Z_20ng, k, criterion='maxclust')
              s[k] = metrics.silhouette_score(data,clusters,metric='euclidean')
          plt.plot(range(2,len(s)),s[2:])
          plt.xlabel('Number of clusters')
          plt.ylabel('Silhouette Score')
```

```
Out[110]: <matplotlib.text.Text at 0x1191400b8>
```


0.014
0.012
0.010
0.008
0.006
Silhouette Score
2
3
4
5
6
7
8
9
10
Number of clusters