clustering_kmeans_gmm

January 30, 2019

```
In [1]: import numpy
        from sklearn.utils import shuffle
        from sklearn.model_selection import train_test_split
        X = numpy.loadtxt("./data/Train/X_train.txt")
        y = numpy.loadtxt("./data/Train/y_train.txt")

In [2]: from sklearn.cluster import KMeans
        kmeans = KMeans(n_clusters=12, n_init=10, max_iter=500, init='random').fit(X)

In [3]: y_pred = kmeans.predict(X)

In [4]: from sklearn.metrics.cluster import adjusted_rand_score
        adjusted_rand_score(y_pred, y)

Out[4]: 0.4101495469561107

In [5]: from sklearn import mixture
        from sklearn.metrics.cluster import adjusted_rand_score

In [6]: gmm = mixture.GaussianMixture(n_components=12, covariance_type='full', n_init=10).fit(X)
        y_pred = gmm.predict(X)
        adjusted_rand_score(y_pred, y)

Out[6]: 0.46936236627835504

In [9]: gmm = mixture.GaussianMixture(n_components=12, covariance_type='tied', n_init=10).fit(X)
        y_pred = gmm.predict(X)
        adjusted_rand_score(y_pred, y)

Out[9]: 0.5521806681791221

In [9]: gmm = mixture.GaussianMixture(n_components=12, covariance_type='diag', n_init=10).fit(X)
        y_pred = gmm.predict(X)
        adjusted_rand_score(y_pred, y)

Out[9]: 0.2668232086990351

In [9]: gmm = mixture.GaussianMixture(n_components=12, covariance_type='spherical', n_init=10).f
        y_pred = gmm.predict(X)
        adjusted_rand_score(y_pred, y)

Out[9]: 0.3745625437911099

In [ ]:
```