PCA_cluster_hw

March 31, 2022

1 PCA Homework

2021.4.13

	MNIST	PCA
train_X.csv:	mnist	data
train_y.csv:	mnist	target

1.0.1 1. sklearn mnist

conda install scikit-learn

[ ]:
```
import sklearn
from sklearn.datasets import fetch_openml
import numpy as np
import pandas as pd

# mnist=fetch_openml('mnist_784',version=1,cache=True)
X = pd.read_csv('./train_X.csv')
y = pd.read_csv('./train_y.csv')

X = np.array(X)
y = np.array(y)

X = X[:6000, :] #  6000
y = y[:6000]
y = np.squeeze(y)

(X.shape, y.shape)
```

[ ]:
```
((6000, 784), (6000,))
```

1.0.2 2. mnist PCA

sklearn PCA

[ ]:
```
## TODO
from sklearn.decomposition import PCA
```

```
from sklearn.preprocessing import StandardScaler
from sklearn import preprocessing
X=preprocessing.scale(X)
pca=PCA()
pca.fit(X)
X_pca=pca.transform(X)
```

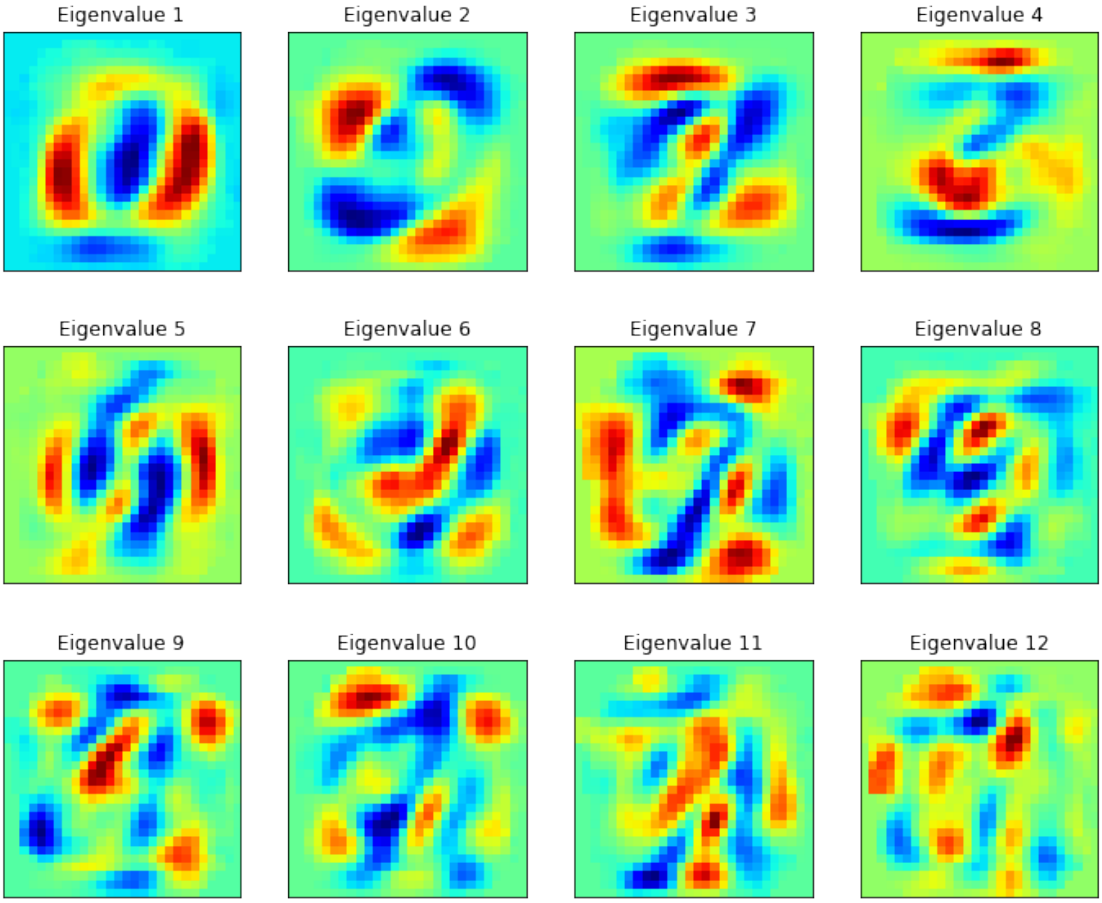
[ ]:
```
X_pca.shape
```

[ ]: (6000, 784)

1.0.3 3. 12 (28, 28)

PCA *shape*

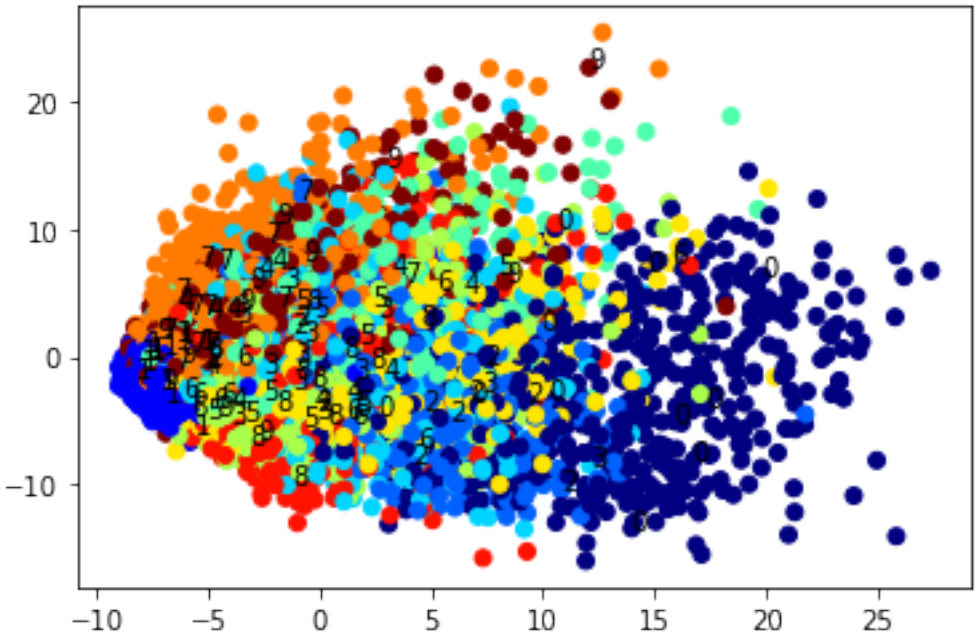
[ ]:
```
## TODO
import matplotlib.pyplot as plt
from matplotlib.colors import Colormap
eigenvalues=pca.components_
plt.figure(figsize=(12,10))
for i in range(12):
    plt.subplot(3,4,i+1)
    plt.imshow(eigenvalues[i].reshape(28,28),cmap='jet')
    plt.title('Eigenvalue '+str(i+1))
    plt.xticks(())
    plt.yticks(())
plt.show()
```



1.0.4 4. PCA 2 label

blog 50

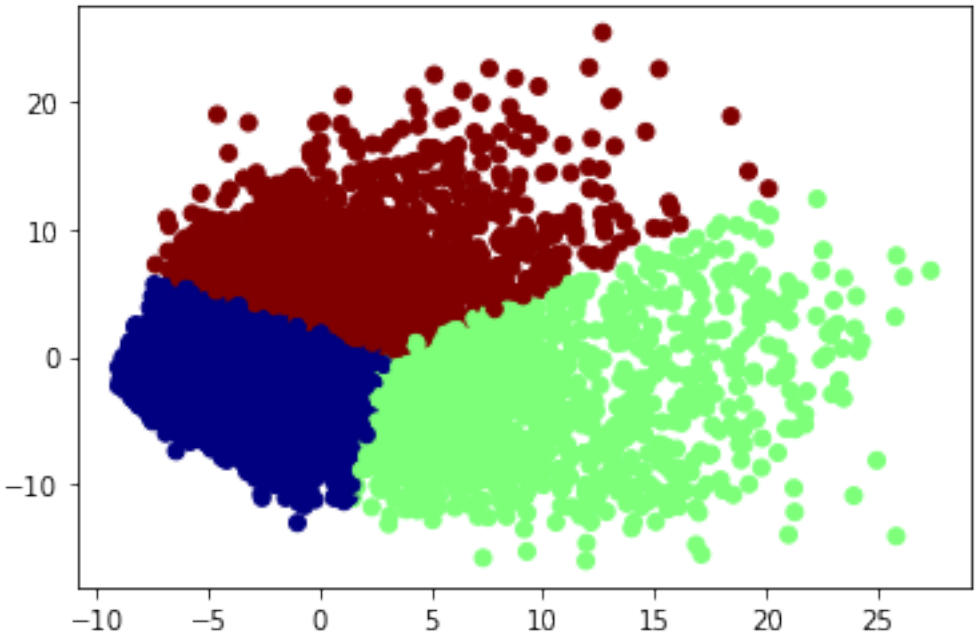
[ ]:
```
## TODO
X_2d=X_pca[:,:2]
fig,ax=plt.subplots()
ax.scatter(X_2d[:,0],X_2d[:,1],c=y,cmap='jet')
y_count=dict((i,0) for i in range(10))
for i in range(6000):
    y_count[y[i]]+=1
    if(y_count[y[i]]%50==0):
        ax.annotate(y[i],X_2d[i])
```



1.0.5 5. PCA 2 kmeans

3 ARI sklearn.metrics.adjusted_rand_score()

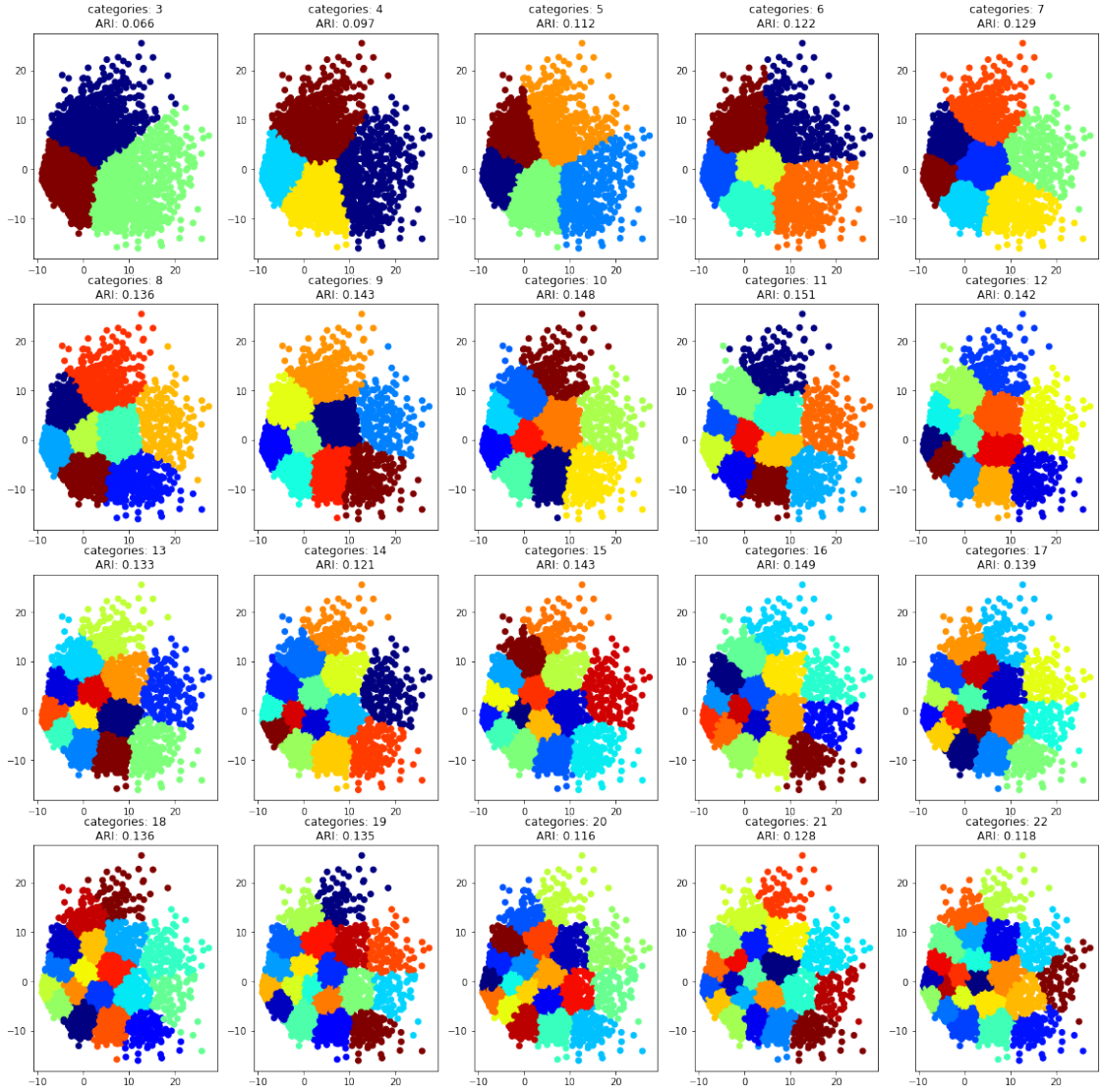
[ ]:
```
## TODO
from sklearn.cluster import KMeans
X_2d_K3=KMeans(n_clusters=3,init='k-means++').fit(X_2d)
fig,ax=plt.subplots()
ax.scatter(X_2d[:,0],X_2d[:,1],c=X_2d_K3.labels_,cmap='jet')
plt.show()
from sklearn.metrics import adjusted_rand_score
adjusted_rand_score(y,X_2d_K3.labels_)
```



[ ]: 0.06568206509202973

3 20 ARI

[ ]:
```
## TODO
Kmeans_3_20=[KMeans(n_clusters=i,init='k-means++').fit(X_2d) for i in␣
 ↪range(3,23)]
ARI_3_20=[adjusted_rand_score(y,Kmeans_3_20[i-3].labels_) for i in range(3,23)]
plt.figure(figsize=(20,20))
for i in range(20):
    plt.subplot(4,5,i+1)
    plt.scatter(X_2d[:,0],X_2d[:,1],c=Kmeans_3_20[i].labels_,cmap='jet')
    plt.title("categories: "+str(i+3)+"\nARI: %.3f"%ARI_3_20[i])
plt.show()
```


categories: 3
ARI: 0.066
categories: 4
ARI: 0.097
categories: 5
ARI: 0.112
categories: 6
ARI: 0.122
categories: 7
ARI: 0.129
categories: 8
ARI: 0.136
categories: 9
ARI: 0.143
categories: 10
ARI: 0.148
categories: 11
ARI: 0.151
categories: 12
ARI: 0.142
categories: 13
ARI: 0.133
categories: 14
ARI: 0.121
categories: 15
ARI: 0.143
categories: 16
ARI: 0.149
categories: 17
ARI: 0.139
categories: 18
ARI: 0.136
categories: 19
ARI: 0.135
categories: 20
ARI: 0.116
categories: 21
ARI: 0.128
categories: 22
ARI: 0.118