PrincipleComponentAnalysis

December 22, 2017

1 Principle Component Analysis

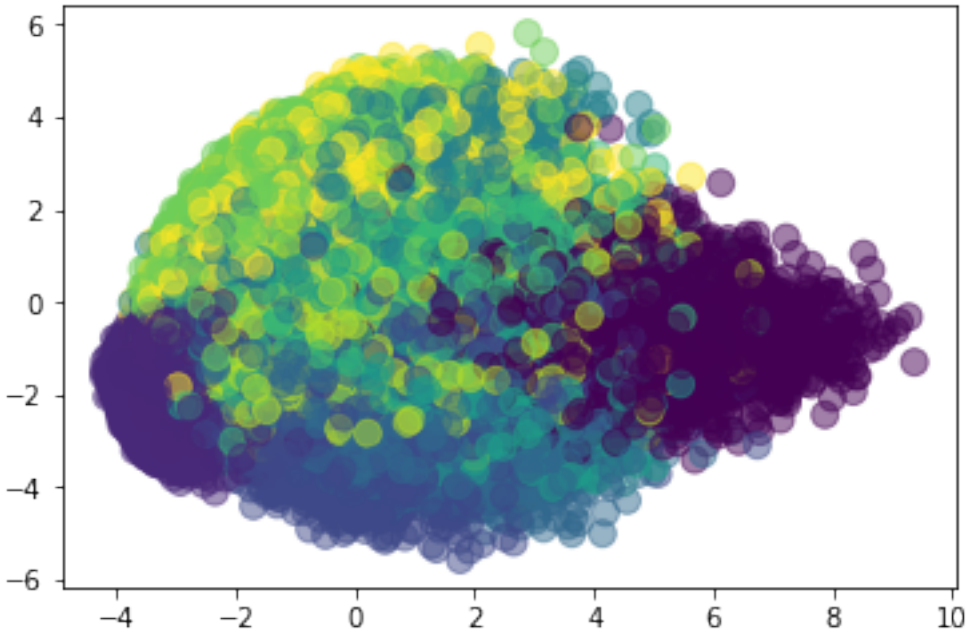
```
In [4]: from __future__ import print_function, division
        from builtins import range, input
        import numpy as np
        import matplotlib.pyplot as plt
        from sklearn.decomposition import PCA
        from util import getKaggleMNIST
        %matplotlib inline
```

Separating the training set and test set with function from util.

```
In [2]: Xtrain, Ytrain, Xtest, Ytest = getKaggleMNIST()
```

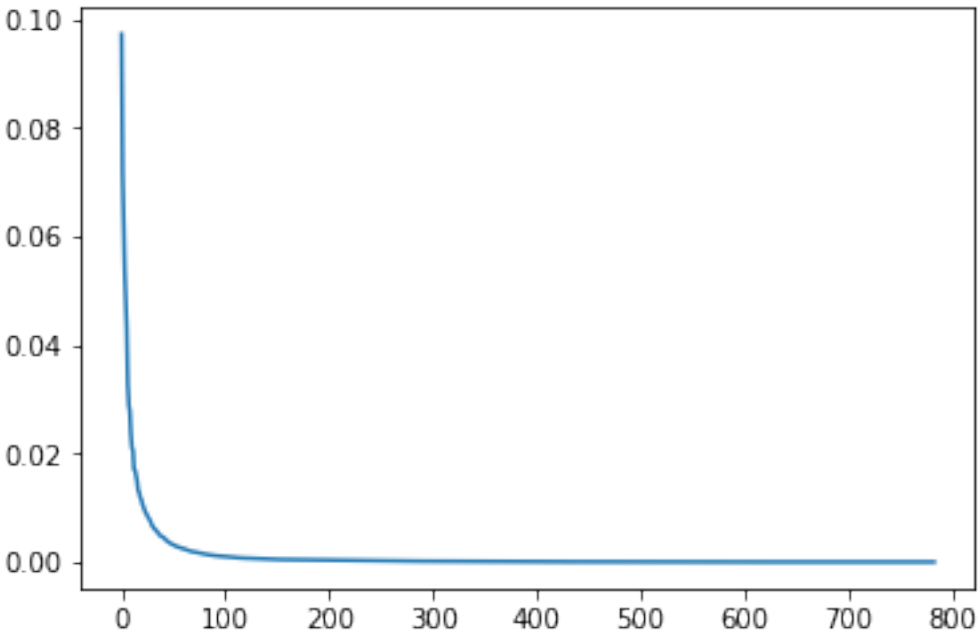
We create a pca object, and fit the training data to the recuce variable. Then we visualize the first two columns of the reduced data, colored by Ytrain.

```
In [3]: pca = PCA()
        reduced = pca.fit_transform(Xtrain)
        plt.scatter(reduced[:,0], reduced[:,1], s=100, c=Ytrain, alpha=0.5)
        plt.show()
```



Plotting the eigenvalues.

```
In [5]: plt.plot(pca.explained_variance_ratio_)
        plt.show()
```



cumulative variance
choose k = number of dimensions that gives us 95-99% variance

```
In [6]: cumulative = []
        last = 0
        for v in pca.explained_variance_ratio_:
            cumulative.append(last + v)
            last = cumulative[-1]
        plt.plot(cumulative)
        plt.show()
```