CS589 Machine Learning
Homework 5

Submitted by- Ravi Agrawal Due on- Dec 11nd, 2017

Submitted On- Dec 13th , 2017

1 a:

Maximizing variance :-

The variance is

$$\sigma^2_{\vec{w}} = \frac{1}{n} \sum_i (x_i \cdot w)^2$$

$$= \frac{1}{n} (xw)^T (xw)$$

$$= \frac{1}{n} w^T x^T x w$$

$$= w^T \frac{x^T x}{n} w$$

$$\sigma^2_{\vec{w}} = w^T V w$$

we want to choose unit vector $\vec{w}$ so as to maximize $\sigma^2_{\vec{w}}$, to do this constraint is $|w| = 1, w^T w = 1$ and we can do this by introducing a new variable, a Lagrange multiplier d,

so,

$$L(w, d) = \sigma^2_w - d(w^T w - 1)$$

$$\frac{\partial L}{\partial d} = w^T w - 1$$

$$\frac{\partial L}{\partial w} = 2Vw - 2dw$$

Setting derivative to zero.

$$w^T w = 1$$

$$Vw = dw$$

thus, w is an eigen vector of covariance V, and the maximizing vector will be the one associated with the largest eigenvalue d.

1 b:

As see in the 1a, we get the maximum variance is generated by the k maximum eigen value, it is very simple to proof that subspace of dimension 2 that maximizes the variance is associated with the eigenvalue of the first two maximum eigen value.

We know:- $VW = \lambda W$ gives us maximum variance of the data,

The 1 principle component that covers the maximum variance is represented as $VW1 = \lambda 1W1$. Using the above equations. The Same rule applies for the two dimension $VW2= \lambda 2W2$.

1 C: The $X_p = \sum \alpha_i x_i \ \forall\ 1 \ldots p - 1$ solving this equation for p = 2 and p = 3 gives:

$$X_2 = \alpha_1 x_1 \qquad \text{Equation 1}$$
$$X_3 = \alpha_1 x_1 + \alpha_2 x_2 \qquad \text{Equation 2}$$

Here:

We can update X_2 in Equation 2 to Equation 1

$$X_3 = \alpha_1 x_1 + \alpha_2 \alpha_1 x_1 \qquad \text{Equation 3}$$

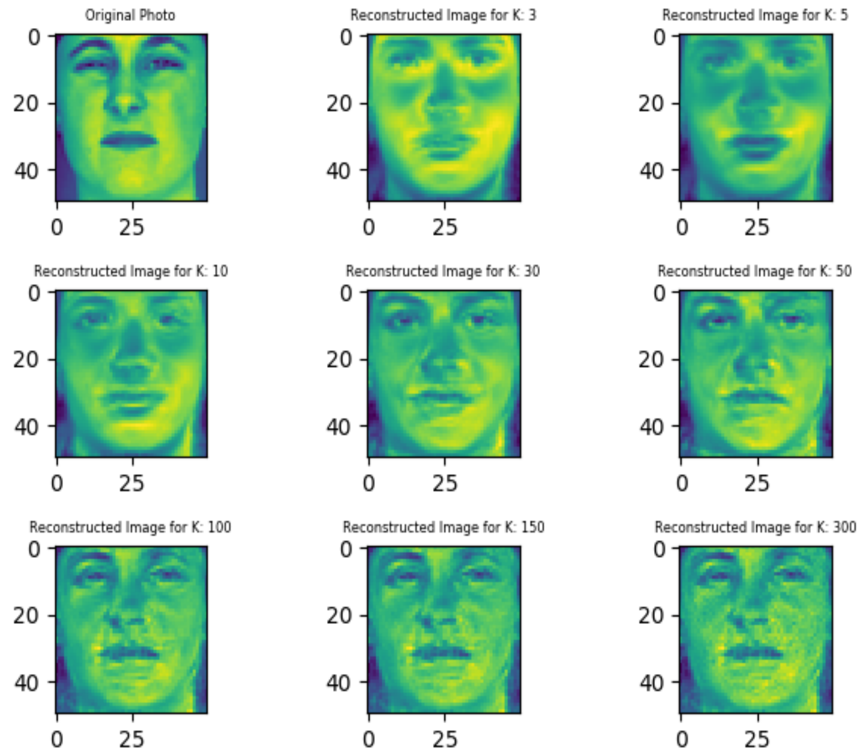
Same is true for the higher dimension.

Say P = 4:

$$X_4 = \alpha_1 x_1 + \alpha_2 \alpha_1 x_1 + \alpha_3(\alpha_1 x_1 + \alpha_2 \alpha_1 x_1) \qquad \text{Equation 4}$$

So, it gets clear from the above equation that most of the variance is captured by the first Eigen vector of the estimated covariance matrix. The reason this Happens is very clear from the Equation 3 and Equation 4. When we start moving to the higher dimension, the equation follows the linear relation with first Eigen vector.

1 d 1:



1 d 2 / 4:

c	Reconstruction Error	Compression Rate
3	44.263	0.031
5	40.743	0.052
10	35.653	0.104
30	22.825	0.312
50	18.698	0.520
100	14.544	1.040
150	14.449	1.560
300	14.056	3.120

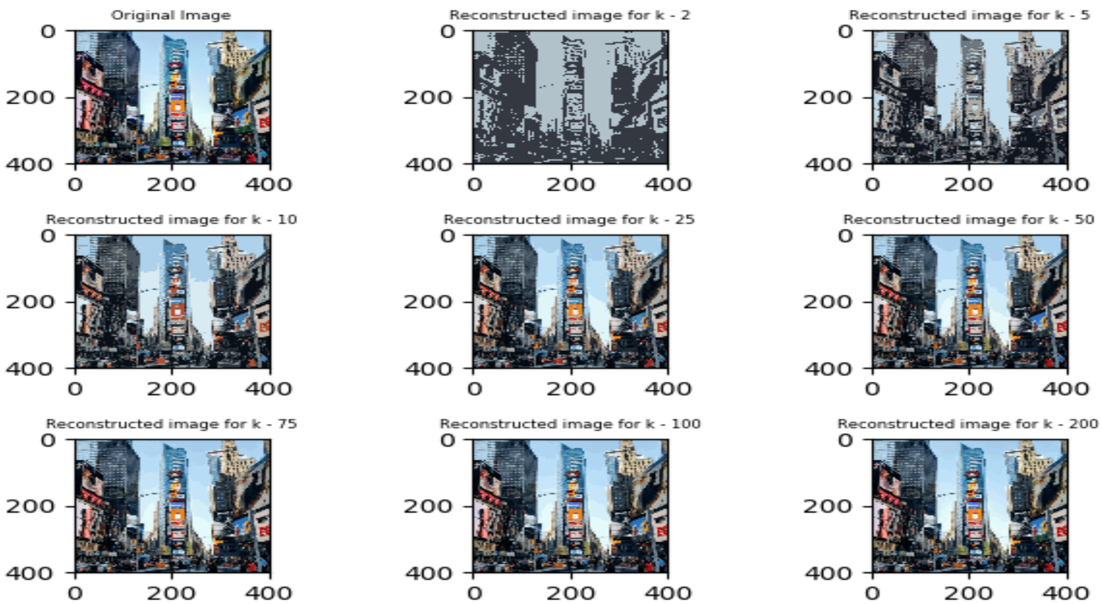
The PCA is very efficient method for the image compression, As performing the PCA in the 100 images took fraction of time as compared to the performing the k-means on a single image with the k as low as 2. But the PCA have high reconstruction Error and with the increase in the k, the compression rate also increases and in case of really high k the compression rate becomes 1.5 to 3 time of the original data set.

2 a. The elbow method in the k mean clustering is way to find the optimal number of k for the k-mean clustering. The main idea of the elbow method is to run the k-mean clustering method for the range of values of k and for each value of the k calculate the sum of

squared error (SEE). When plotting the SSE for the values of the k, it looks like an arm and the elbow of that arm provide the best value of the k.

2 b: The K-mean++ is an extension to the k-mean clustering algorithm. The approach leads to the constant factor improvement. The typical approach is to choose a data example as the center for cluster, choose another data example which is at least ε distance away from the first example as center for cluster.

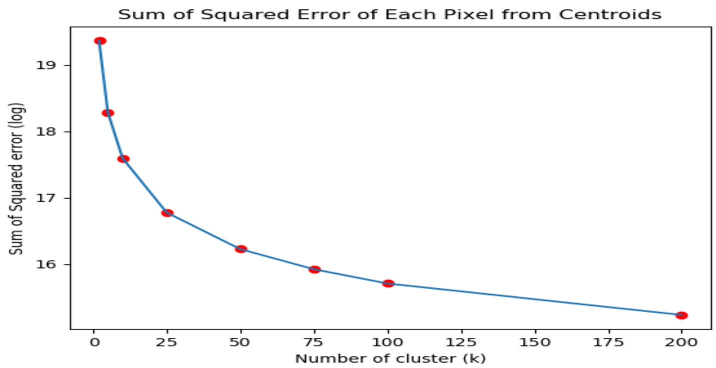
2 c 1.



2 c 2/ 3.

K	Reconstruction Error	Compression Rate
2	0.157	0.041
5	0.091	0.096
10	0.064	0.138
25	0.043	0.193
50	0.032	0.235
75	0.028	0.260
100	0.025	0.277
200	0.019	0.329

2 c 4:



The K-means is very powerful tool for the image compression, but finding the right k can be troublesome. If the K-means is performed with very small k then we may end up with very high Reconstruction error, Sum of Squared error (SSE) or very low compression rate. If we choose k very high then the algorithms will work very well on compression without losing information but will take a long time to compress the image. The elbow graph as shown above is a great technique to found the right k to get the fast compression, without losing information.