$$\forall \mathbf{x} \in \mathbb{R}^n : \quad \mathbf{x}^\top \mathbf{A} \mathbf{x} \geq 0,$$

שאלה 1

תהי $A \in \mathbb{R}^{nxn}$, נתון פירוק ה-SVD הקומפקטי שלה: $A = U_r \Sigma_r V_r^T$. הוכיחו את הטענות הבאות:

1. אם A מוגדרת אי שלילית אז $\forall i = 1,2, \dots$, מתקיים: $A^i = U_r \Sigma_r^i U_r^T$.
2. אם A לא סינגולרית (כלומר $rank(A) = n$) אז $A^{-1} = V_r \Sigma_r^{-1} U_r^T$.

1) A אי שלילית ⇐ $\forall x \in \mathbb{R}^n : x^T \cdot A \cdot x \geq 0$ ומכאן מתקיים כי בפירוק SVD

הוא $U_r \cdot \Lambda_i U_r^T$ ו- $\forall i = 1,2,\dots : \lambda_i(A) \geq 0$

בפרט, גם מתקיים כי $\lambda_i = \sigma_i$ ו- $u_i = v_i$ ⇐ $U_r = V_r$, ו- A היא סימטרית

$$A^i = \sum_{j=1}^{n} (U \cdot \Sigma \cdot V^T)_{j,i} = U_r \cdot \begin{pmatrix} \sigma_1^i & & 0 \\ & \ddots & \\ 0 & & \sigma_r^i \end{pmatrix} \cdot V_r^T = \boxed{U_r \cdot \Sigma_r^i \cdot V_r}$$

פירוק SVD

לכן

2) $rank(A) = n$, $V_R = V$ ולכן $U_R = U$

$$A^{-1} = (A)^{-1} = (U_r \cdot \Sigma_r \cdot V_r^T)^{-1} = (U \Sigma V^T)^{-1} = (V^T)^{-1} \cdot \Sigma^{-1} \cdot U^{-1}$$

$$= V \cdot \Sigma^{-1} \cdot U = \boxed{V_r \cdot \Sigma_r^{-1} \cdot U_r}$$

V, U אורתוגונליות

שאלה 2

(2

תהי A מטריצה ממשית $m \times n$ ונתבונן בבעית האופטימיזציה הבאה:

$$\max_{B \in R^{m \times n}: \|B\|_2 \le 1} Tr(AB^\top)$$

נסמן את פירוק ה-SVD הקומפקטי של A: $A = U_r \Sigma_r V_r^T$.

הוכיחו כי הערך האופטימלי של בעיית האופטימיזציה הנ״ל הוא $Tr(\Sigma_r)$ וכי הוא מתקבל עבור המטריצה $B^* = U_r V_r^\top$.

<u>תזכורת:</u> ראינו בתרגיל בית כי עבור $A \in \mathbb{R}^{mxn}$ מתקיים:

$$\max_{u \in \mathbb{R}^m, v \in \mathbb{R}^n, \|u\|_2 = \|v\|_2 = 1} u^T A v = \sigma_1(A)$$

$$\max_{B \in R^{m \times n}: \|B\|_2 \le 1} Tr(AB^T) = \max_{B \in R^{m \times n}: \|B\|_2 \le 1} Tr(U_r \cdot \Sigma_r \cdot V_r^T \cdot B^T)$$

נבחר $B^* = U^r V^T$

מתקיים ש-

$$Tr(A(B^*)^T) = Tr(U_r \Sigma_r V_r^T U_r^T V_r) \underset{\substack{\text{V אורתונורמלי} \\ \text{associative} \\ \text{Tr של}}}{=} Tr(U_r \Sigma_r U_r^T)$$

$$\underset{\substack{\text{associative} \\ \text{Tr של} \\ \text{U אורתונורמלי}}}{=} Tr(\Sigma_r)$$

נסתכל על $Tr(AB^T)$:

$$Tr(A \cdot B^T) = Tr\left(\sum_{i=1}^{r} \sigma_i(A) \cdot u_i v_i^T \cdot B^T\right) = \sum_{i=1}^{r} Tr(\sigma_i(A) u_i v_i^T B^T) =$$

$$= \sum_{i=1}^{r} \sigma_i(A) \cdot Tr(u_i v_i^T B^T) \underset{\substack{\text{תכונת} \\ \text{ציקליות} \\ \text{trace של}}}{=} \sum_{i=1}^{r} \sigma_i(A) \cdot Tr(v_i^T \cdot B^T \cdot u_i) \underset{\text{אזכורת}}{\le} \sum_{i=1}^{r} \sigma_i(A) \cdot \sigma_1(B^T)$$

$$= \sigma_1(B^T) \cdot \sum_{i=1}^{r} \sigma_i(A) \underset{\sigma_1(B^T) \le 1}{\le} \sum_{i=1}^{r} \sigma_i(A) = Tr(\Sigma_r)$$

לכן מתקיים שהוא חסום ע"י $Tr(\Sigma_r)$.

לכן קיבלנו כי הערך האופטימלי הוא $Tr(\Sigma_r)$.

3

נפריך את הטענה.

$n=m=2$

$$X=\begin{pmatrix}1&1\\1&-1\end{pmatrix},\quad \bar{x}=\begin{pmatrix}1\\0\end{pmatrix}$$

$$\tilde{X}=\begin{pmatrix}0&0\\1&-1\end{pmatrix}$$

$$XX^T=\begin{pmatrix}1&1\\1&-1\end{pmatrix}\begin{pmatrix}1&1\\1&-1\end{pmatrix}=\begin{pmatrix}2&0\\0&2\end{pmatrix}=2I$$

$$\tilde{X}\tilde{X}^T=\begin{pmatrix}0&0\\1&-1\end{pmatrix}\begin{pmatrix}0&1\\0&-1\end{pmatrix}=\begin{pmatrix}0&0\\0&2\end{pmatrix}$$

$$\underline{\lambda_1(\tilde{X}\tilde{X}^T)=2}:\begin{pmatrix}0&0\\0&2\end{pmatrix}\begin{pmatrix}u_1^1\\u_1^2\end{pmatrix}=\begin{pmatrix}0\\2u_1^2\end{pmatrix}=\begin{pmatrix}2u_1^1\\2u_1^2\end{pmatrix}\Rightarrow u_1=\begin{pmatrix}0\\1\end{pmatrix}$$

$$\underline{\lambda_{1,2}(XX^T)=2}:\quad v_1=\begin{pmatrix}1\\0\end{pmatrix},\ v_2=\begin{pmatrix}0\\1\end{pmatrix}$$

$\underline{k=1}:$

$$u_1u_1^T\tilde{x}_1=\begin{pmatrix}0\\1\end{pmatrix}\begin{pmatrix}0&1\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}=\begin{pmatrix}0&0\\0&1\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}=\begin{pmatrix}0\\1\end{pmatrix}$$

$$z_1-\bar{z}=v_1v_1^Tx_1-\frac{1}{m}\sum_i v_1v_1^Tx_i=v_1v_1^T\cdot(x_1-\bar{x})=$$

$$=\begin{pmatrix}1\\0\end{pmatrix}\begin{pmatrix}1&0\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}=\begin{pmatrix}1&0\\0&0\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}=\begin{pmatrix}0\\0\end{pmatrix}\neq\begin{pmatrix}0\\1\end{pmatrix}$$

⇐ מצאנו X עבורו כך ש- $u_1u_1^T\tilde{x}_1\neq z_1-\bar{z}$

⇐ מצאנו דוגמא נגדית.

ולכן הטענה אינה נכונה.

Shitot HW04 Q4

Rank of original Photo: 1200,

for k = 5 the relative error is: [0.005105578357907279, 0.003963948517002348, 0.00571569483351151]

for k = 20 the relative error is: [0.0014383270710896577, 0.0012398712244860092, 0.0016262032426646277]

for k = 50 the relative error is: [0.000552146070697173, 0.0004367929525544366, 0.0005844667493007953]

for k = 70 the relative error is: [0.00034836949542746285, 0.0002876235837437606, 0.0003806913037311242]

for k = 80 the relative error is: [0.00029327531918666805, 0.0002462184863804964, 0.000320776712959371]

for k = 100 the relative error is: [0.00021024596901793945, 0.00018316303070982972, 0.00023085245272585838]

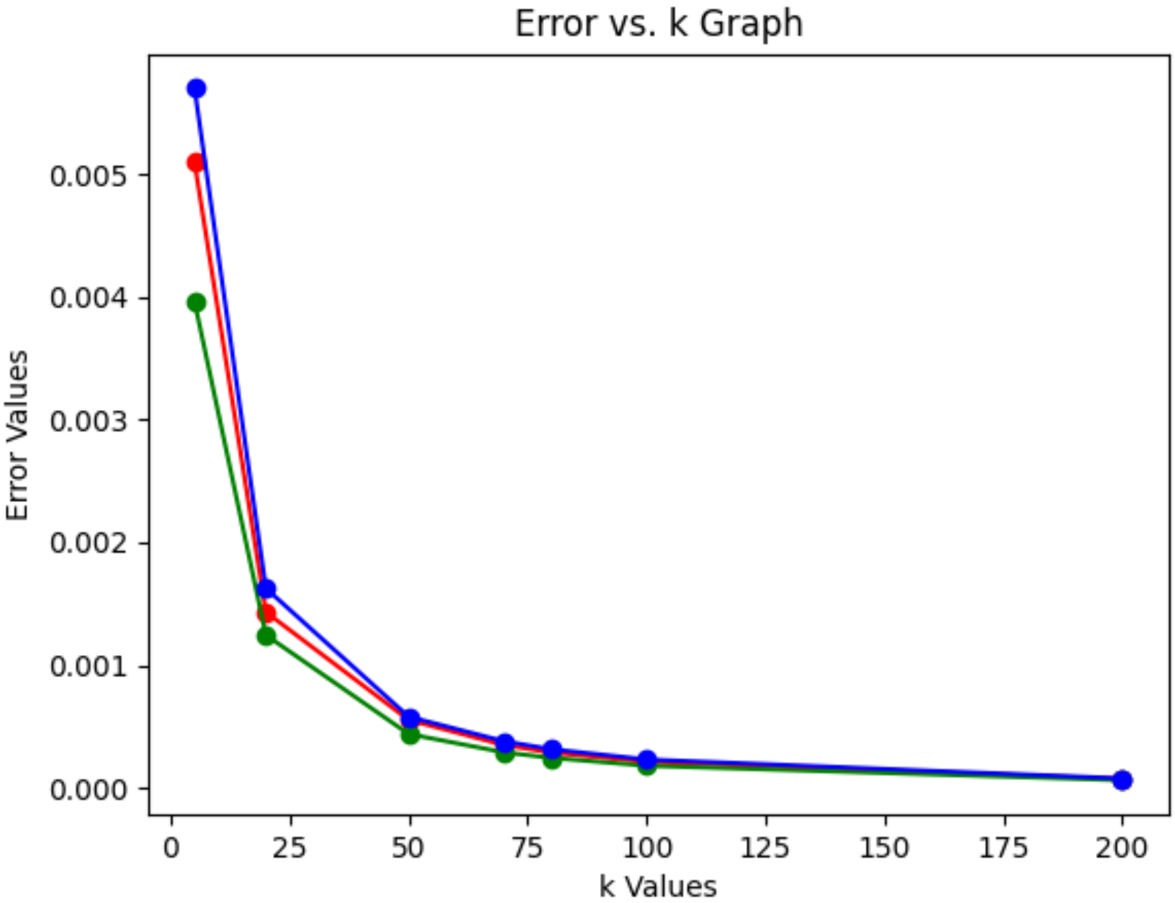
for k = 200 the relative error is: [7.365990714309867e-05, 6.34732538080542e-05, 7.848161378655681e-05]

for K = 50 we can see the picture relatively clearly. The ratio of this minimal value to the original rank is $\frac{5}{120}$

Does the relative errors agree with the minimal k values?
We can see that for higher k value we use the lower the relative error get's so it makes sense that the resolution of the image will get better. On the other hand, the error for k=20 the photo is pretty blurry and hard to see what it shows. Furthermore, the error rate for $k = 5$ is less than 0.6% per each of RGB matrices but the image received is super blurry and we can't see what it is.

here are graphs where we can see a sharp decrease in error values next to $k = 50$, **thus it agrees with the minimal k value**:

for each of the RGB: matrices separately according to the errors above and different k values.



Report for Q5.

Testing Errors.

	S=5	**S=10**	**S=50**	**S=100**	**S=500**	**S=1000**	**ORIGINAL**
K=5	0.7777	0.7007	0.6529	0.6712	0.7023	0.7065	0.7066
K=10	0.748	0.6789	0.6489	0.6684	0.7038	0.7097	0.7096
K=50	0.7145	0.6489	0.6615	0.6863	0.7176	0.7198	0.7198
K=100	0.7113	0.6592	0.6765	0.6969	0.7227	0.7249	0.725
K=200	0.7158	0.6703	0.6944	0.7159	0.7385	0.7399	0.7399

Training Errors.

The train set size was really big for the computer to do the computations, so we chose 1000 first values in the train set to calculate the train error.

	S=5	**S=10**	**S=50**	**S=100**	**S=500**	**S=1000**	**ORIGINAL**
K=5	0.545	0.459	0.471	0.482	0.544	0.548	0.547
K=10	0.605	0.52	0.547	0.553	0.599	0.607	0.607
K=50	0.661	0.6	0.621	0.644	0.684	0.678	0.678
K=100	0.681	0.654	0.654	0.696	0.713	0.712	0.712
K=200	0.693	0.696	0.696	0.71	0.741	0.742	0.742

From the tables that we get, we can see that for S=500 (less than half of the original dimension!), both training and testing errors are very close to the those of the original data, independent of the values of k.

To conclude, we think that PCA is effective for the KNN algorithm no matter what k we choose, as we can see that for any k, the error gets closer to the original one as s increases.