Collaborator: Satchel

Math 123 **Problem Set 3** **Due: Friday 3/12 at 11:59 p.m. EST**

Instruction: Read the homework policy. For problems 6 and 7, include printed copies of your code with your final homework submission. You should submit a PDF copy of the homework and any associated codes on Canvas. Your PDF must be a single file, not multiple images.

1. Given n data points $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$ lying in $\mathcal{R}^d$, the covariance matrix is defined as follows:

$$\mathbf{\Sigma} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T$$

(a) Argue that $\mathbf{\Sigma} = \frac{1}{n}\mathbf{X}^T\mathbf{X}$ where $\mathbf{X}$ is the n by d data matrix defined as

$$\mathbf{X} = \begin{pmatrix} \dots & \mathbf{x}_1^T & \dots \\ \dots & \mathbf{x}_2^T & \dots \\ & \vdots & \\ \dots & \mathbf{x}_n^T & \dots \end{pmatrix}$$

(b) Prove that the covariance matrix is positive semi-definite. [**Remark**: A matrix $\mathbf{A} \in \mathcal{R}^{n\times n}$ is positive semi-definite if $\mathbf{y}^T\mathbf{A}\mathbf{y} \geq 0$ for all $\mathbf{y} \in \mathcal{R}^n$.]

(c) Prove that all eigenvalues of the covariance matrix are non-negative. [**Hint**: Consider the quadratic form.]

(d) The covariance matrix is not necessarily positive definite. Let $d = 4$. Describe or give an example of data for which the covariance matrix is positive semi-definite but not positive definite.

(e) Let $d = 4$. Describe or give an example of data for which the covariance matrix is positive definite.

[**Remark**: For parts (d) and (e), data point must be not trivial e.g. $(0, 0, 0, 0)$].

2. Consider n data points $\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n$ lying in $\mathcal{R}^d$. The mean of the data points is defined as $\mu = \frac{1}{n}\sum_{i=1}^{n} \mathbf{y}_i$. Let the points $\{\mathbf{x}_1 = \mathbf{y}_1 - \mu, \mathbf{x}_2 = \mathbf{y}_2 - \mu, ..., \mathbf{x}_n = \mathbf{y}_n - \mu\}$ denote the centered points i.e. the points have zero mean (centered at origin). Let $\mathbf{v}$ be a unit vector in $\mathcal{R}^d$ and let $\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, ..., \hat{\mathbf{x}}_n$ denote respectively the projection of the centered points onto the vector $\mathbf{v}$. The representation of the projections with respect to the coordinate $\mathbf{v}$ is $c_1 = (\mathbf{x}_1)^T\mathbf{v}, c_2 = (\mathbf{x}_2)^T\mathbf{v}, ..., c_n = (\mathbf{x}_n)^T\mathbf{v}$.

(a) Show that the variance of $\{c_1, c_2, ..., c_n\}$ is given by

$$V = \frac{1}{n}\sum_{i=1}^{n} \left(\mathbf{v}^T\mathbf{x}_i\right)^2$$

(b) Show that the variance of $\{\mathbf{y}_1^T\mathbf{v}, \mathbf{y}_2^T\mathbf{v}, ..., \mathbf{y}_n^T\mathbf{v}\}$ is given by

$$V = \frac{1}{n}\sum_{i=1}^{n} \left(\mathbf{v}^T\mathbf{x}_i\right)^2$$

(c) What is the implication of the results in (a) and (b) to finding the first principal component of the data? Briefly interpret your result.

3. Consider n data points $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$ lying in $\mathcal{R}^d$. Let $\mathbf{v}_1, \mathbf{v}_2$ and $\mathbf{v}_3$ denote the first three principal components of the data.

(a) Let $\hat{\mathbf{x}}_i$ denote the projection of a data point $\mathbf{x}_i$ onto the subspace spanned by the first three principal components. What is the coordinate of $\hat{\mathbf{x}}_i$ with respect to the basis $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$?

(b) Assume $d \gg 3$. What is the low dimensional embedding of the data points?

(c) Let $\mathbf{y}$ be a point in $\mathcal{R}^d$ and let $\hat{\mathbf{y}}$ be the projection of $\mathbf{y}$ onto the subspace spanned by the first three principal components. Prove that $\hat{\mathbf{y}} = \mathbf{V}\mathbf{V}^T\mathbf{y}$ where

$$\mathbf{V} = \begin{bmatrix} \vdots & \vdots & \vdots \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \\ \vdots & \vdots & \vdots \end{bmatrix}$$

4. Let $\mathbf{A}$ be an $n \times n$ symmetric matrix.

(a) Prove that $\lambda_{\min}(\mathbf{A}) \leq \min\limits_{1 \leq i \leq n} \mathbf{A}_{i,i}$ i.e. the minimum eigenvalue of $\mathbf{A}$ is upper bounded by the the minimum value in the diagonal.

(b) Prove that $\lambda_{\max}(\mathbf{A}) \geq \max\limits_{1 \leq i \leq n} \mathbf{A}_{i,i}$ i.e. the maximum eigenvalue of $\mathbf{A}$ is lower bounded by the the maximum value in the diagonal.

[**Hint**: Use Rayleigh quotient.]

5.[**Bonus**: 5 pts] Let $\mathbf{A}$ be a $2n \times 2n$ symmetric matrix of the following form

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_3 \\ \mathbf{A}_3 & \mathbf{A}_2 \end{bmatrix}$$

where the blocks $\mathbf{A}_1, \mathbf{A}_2$ and $\mathbf{A}_3$ are each $n \times n$ matrices and $\mathbf{A}_1$ and $\mathbf{A}_2$ are symmetric matrices. Prove that $\lambda_{\min}(\mathbf{A}) \leq \min\left(\lambda_{\min}(\mathbf{A}_1), \lambda_{\min}(\mathbf{A}_2)\right)$. [**Hint**: Use Rayleigh quotient.]

6. In this problem, we implement the principal component analysis algorithm and test it on 2-dimensional datasets.

(a) Given n points in $\mathcal{R}^2$, implement an algorithm that takes the data points as input and returns the principal components.
Remark: No credit will be given to using an inbuilt PCA function. However, you can use any inbuilt function to compute eigenvalues and eigenvectors. For example, in MATLAB, you can use the *eig* command.

(b) Load the data `gaussian_noisy` available in the HW3 folder. Compute and display the first 2 principal components of the data. What is the percentage of variance for the first principal component? What is the percentage of variance for the second principal component? Summarize and explain your observations.

(c) Load the data `uniform_noisy` available in the HW3 folder. Compute and display the first 2 principal components of the data. Compute the percentage of variance for the each of the first two principal components. Summarize and explain your observations.

7. In this problem, we consider the latent features obtained by the principal component analysis and explore the relationship between the number of principal components and average reconstruction error. The dataset we consider is the MNIST dataset which is a database of handwritten digits.

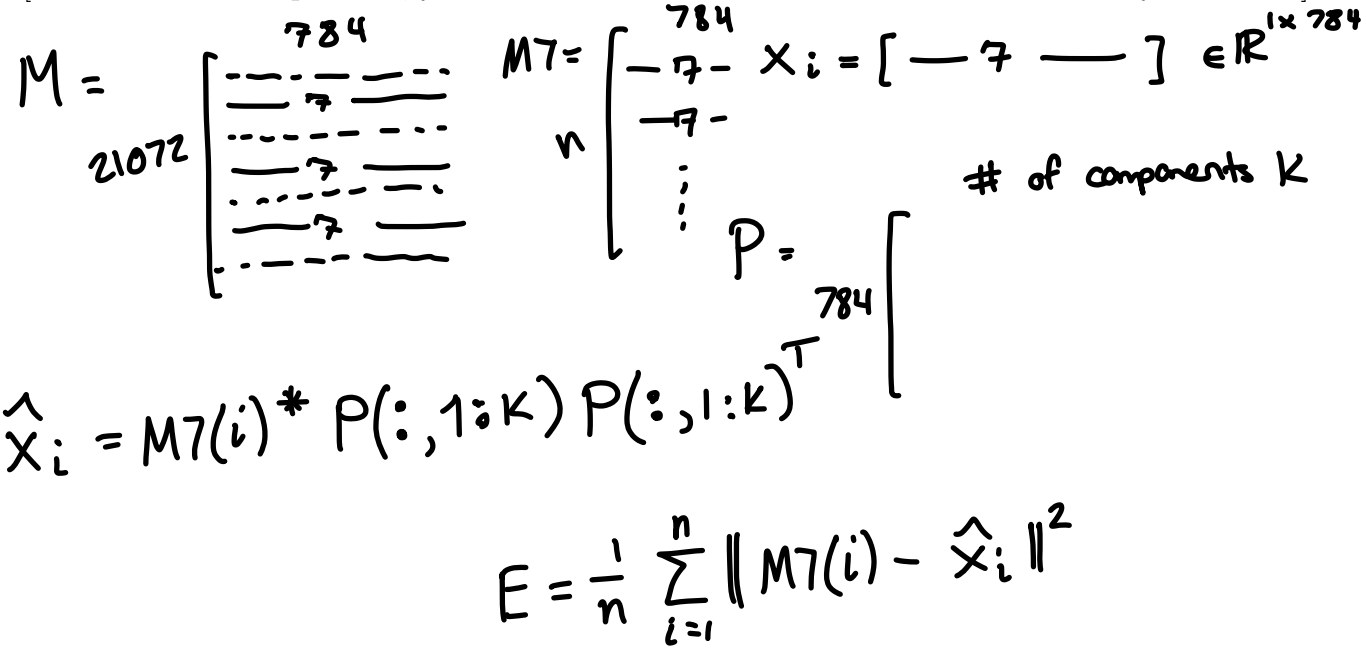
(a) Load the data `mnist_067` available in the HW3 folder. The included data is a subset of the MNIST data consisting of the digits $0, 6$ and 7. There are 21072 digits and each digit is an image of size 28×28. With that, the data matrix is a matrix of size 21072×784. Note that each image is mapped into a 784 dimensional vector with each entry informing the gray pixel intensity. Project the data points onto the first two principal components of the data. Show the representation of each data point in the coordinate of the principal components i.e. plot the latent representation of the data in $\mathcal{R}^2$. Use different colors to label the latent features according to the label of the digit. Is this a good low dimensional representation? Interpret your results.

(b) We now consider the digit 7 from the provided dataset. Define the average reconstruction error as follows

$$E = \frac{1}{n} \sum_{i=1}^{n} ||\mathbf{x}_i - \hat{\mathbf{x}}_i||^2$$

where n is the number of digits of label 7, $\mathbf{x}_i$ is the input representation of the digit and $\hat{\mathbf{x}}_i$ is the reconstructed digit using K principal components of the data. Plot the average reconstruction error as a function of the number of principal components. Let the array of the number of principal components be $\{1, 10, 20, ..., 300\}$. Briefly discuss your results.

[**Remark**: For this problem, you can use an inbuilt PCA function from a solver of your choice.]



1

data: $x_1, \dots, x_n \in \mathbb{R}^d$

$$\Sigma := \frac{1}{n} \sum_{i=1}^{n} (\vec{x}_i)(\vec{x}_i)^T$$

(a) show $\Sigma = \frac{1}{n} X^T X$ where $X = \begin{pmatrix} — & x_1^T & — \\ & \vdots & \\ — & x_n^T & — \end{pmatrix}$

$$(\vec{x}_i)(\vec{x}_i)^T = \begin{pmatrix} x_{i_1}^2 & x_{i_1}x_{i_2} & \cdots & x_{i_1}x_{i_d} \\ x_{i_2}x_{i_1} & x_{i_2}^2 & & \\ \vdots & & \ddots & \\ x_{i_d}x_{i_1} & & & x_{i_d}^2 \end{pmatrix} \in \mathbb{R}^{d \times d}$$

$i \in \{1, 2, \dots, n\}$

$$(\vec{x}_1)(\vec{x}_1)^T + \cdots + (\vec{x}_n)(\vec{x}_n)^T = \sum_{i=1}^{n} (\vec{x}_i)(\vec{x}_i)^T$$

$$= \begin{pmatrix} x_{1_1}^2 + x_{2_1}^2 + \cdots + x_{n_1}^2 & \cdots & x_{1_1}x_{1_d} + \cdots + x_{n_1}x_{n_d} \\ \vdots & \ddots & \\ x_{1_d}x_{1_1} + \cdots + x_{n_d}x_{n_1} & & x_{1_d}^2 + x_{2_d}^2 + \cdots + x_{n_d}^2 \end{pmatrix} \in \mathbb{R}^{d \times d}$$

$$= \underbrace{\begin{pmatrix} | & & | \\ x_1 & \cdots & x_n \\ | & & | \end{pmatrix}}_{d \times n} \underbrace{\begin{pmatrix} — & x_1^T & — \\ & \vdots & \\ — & x_n^T & — \end{pmatrix}}_{n \times d} = X^T X \in \mathbb{R}^{d \times d}$$

$$\Longrightarrow \Sigma = \frac{1}{n} X^T X$$

(b) Prove that Σ is positive semi-definite

by (a), $\Sigma = \frac{1}{n} X^T X$ so let $y \in \mathbb{R}^d$

then $(n)\, y^T \Sigma y = y^T X^T X y = (Xy)^T (Xy)$

$= \|Xy\|_2^2 \geq 0$

so Σ is positive semi-definite.

(c) Prove all eigenvalues of Σ are non-negative.

let y be an eigenvector with eigenvalue λ,

then $y^T \Sigma y = y^T \lambda y = \lambda \|y\|^2 \geq 0$ by (b)

but $\|y\|^2$ is always ≥ 0, so $\lambda \geq 0$

and since λ was arbitrary, all eigenvalues are non-negative.

(d) Give an example of data for which the covariance matrix is positive semi-definite not positive definite

let $X = \begin{pmatrix} 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \end{pmatrix}$, then $\Sigma = \frac{1}{2} X^T X = \begin{pmatrix} 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \end{pmatrix}$

using SVD on MatLab,

$u_2 = \begin{pmatrix} .8668 \\ -.2887 \\ .2887 \\ .2887 \end{pmatrix}$, $\Sigma u_2 = 0 \Rightarrow \|\Sigma u_2\| = 0$

$\Rightarrow \Sigma$ is only positive semi-definite

(e) give an example of data for which the covariance matrix is positive definite.

if the data matrix X gives rise to a covariance matrix Σ which is of full rank, then $\ker(\Sigma) = \{0\}$ and no eigenvalues of Σ are 0,

thus $\|\Sigma y\| > 0$ <u>strictly</u> for all $y \in \mathbb{R}^d$, $y \neq 0$

so Σ would be positive definite.

2 Consider $y_1, y_2, \ldots, y_n \in \mathbb{R}^d$

$\mu = \frac{1}{n}\sum_{i=1}^{n} y_i$ the mean, and $x_i = y_i - \mu$

$v \in \mathbb{R}^d$ of length 1

let $\hat{x}_1, \ldots, \hat{x}_n$ be the projections of $x_1, \ldots, x_n$ onto v.

$c_1 = x_1^T v$, $c_2 = x_2^T v$, $\ldots$, $c_n = x_n^T v$

(a) show the variance of $\{c_i\}_{i=1}^n$ is given by

$$V = \frac{1}{n}\sum_{i=1}^{n} (v^T x_i)^2$$

$V = \frac{1}{n}\sum_{i=1}^{n} (c_i - \mu)$ where μ is the mean of $\{c_i\}_{i=1}^n$

since the data composing c_i is centered, $\mu = 0$

hence $V = \frac{1}{n}\sum_{i=1}^{n} (c_i)^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i^T v)^2$ by definition of c_i

$$= \frac{1}{n}\sum_{i=1}^{n} (x_i \cdot v)^2 = \frac{1}{n}\sum_{i=1}^{n} (v \cdot x_i)^2$$

$$= \frac{1}{n}\sum_{i=1}^{n} (v^T x_i)^2$$

(b) Show that the variance of $\{y_1^T v, y_2^T v, \ldots, y_n^T v\}$ is given by

$$V = \frac{1}{n}\sum_{i=1}^{n}\left(v^T x_i\right)^2$$

$$\mu = \frac{1}{n}\sum_{i=1}^{n} y_i \Rightarrow \mu^T = \frac{1}{n}\sum_{i=1}^{n} y_i^T \Rightarrow \mu^T v = \frac{1}{n}\sum_{i=1}^{n} \mu^T v$$

$$V = \frac{1}{n}\sum_{i=1}^{n}\left(y_i^T v - \mu^T v\right)^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\left(y_i^T - \mu^T\right)v\right)^2$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left(\left(x_i^T\right)v\right)^2 = \frac{1}{n}\sum_{i=1}^{n}\left(v^T x_i\right)^2$$

by (a)

(c) Summarize the results of (a) and (b)

we get the same exact variance from the centered data and the non-centered data, hence centering does not impact variance and thus we don't lose any important information by doing so.

3

$x_1, x_2, \ldots, x_n \in \mathbb{R}^d$

v_1, v_2, v_3 are the first 3 principal components of the data

(a) let $\hat{x}_i$ denote the projection of x_i onto the subspace spanned by $\{v_1, v_2, v_3\}$

what is the coordinate of $\hat{x}_i$ wrt v_1, v_2, v_3?

$V = \begin{bmatrix} | & | & | \\ v_1 & v_2 & v_3 \\ | & | & | \end{bmatrix}$, then $\hat{x}_i = VV^T x_i$

$= \langle v_1, x_i \rangle v_1 + \langle v_2, x_i \rangle v_2 + \langle v_3, x_i \rangle v_3$

(b) $d >> 3$, what is the low-dimensional embedding?

if $X = \begin{bmatrix} — & x_1^T & — \\ & \vdots & \\ — & x_n^T & — \end{bmatrix}$, then decompose X via SVD to obtain $X = U \Sigma V^T$

truncate at $k=3$ to get low dimensional representation

OR $\hat{X} = XVV^T$ gives the same answer (I think....)

(c) Prove $\hat{y} = VV^T y$,

by (a), $\hat{y} = \langle v_1, y \rangle v_1 + \langle v_2, y \rangle v_2 + \langle v_3, y \rangle v_3$

$= y v_1^T v_1 + y v_2^T v_2 + y v_3^T v_3$

$= VV^T y$

4

Let A be an $n \times n$ symmetric matrix

(a) Prove $\lambda_{min}(A) \leq \min_{1 \leq i \leq n} A_{i,i}$

$\lambda_{min} = \min_{v \neq 0} \frac{\langle v, Av \rangle}{\langle v, v \rangle}$ by Courant-Fischer

but notice that for any diagonal entry $A_{i,i}$,

$A_{i,i} = \frac{\langle e_i, Ae_i \rangle}{\langle e_i, e_i \rangle}$ where e_i is the standard i^{th} basis vector,

hence λ_{min} must be smaller than or equal to all $A_{i,i}$

(b) Prove $\lambda_{max}(A) \geq \max_{1 \leq i \leq n} A_{i,i}$

$\lambda_{max} = \max \frac{\langle v, Av \rangle}{\langle v, v \rangle}$ by Courant-Fischer

and by the same exact argument as (a),

$A_{i,i} = \frac{\langle e_i, Ae_i \rangle}{\langle e_i, e_i \rangle}$, hence λ_{max} must be greater than or equal to all $A_{i,i}$

Since λ_{min} and λ_{max} minimize and maximize the rayleigh quotient and since the diagonal entries of A can all be represented by a rayleigh quotient using an appropriate vector (e_i), we reach the result.

5 (Bonus)

$$A = \begin{bmatrix} A_1 & A_3 \\ A_3 & A_2 \end{bmatrix} \in \mathbb{R}^{2n \times 2n}$$

Prove that $\lambda_{min}(A) \leq \min\left(\lambda_{min}(A_1), \lambda_{min}(A_2)\right)$

By Courant-Fischer, this must be the case since $\lambda_{min}(A)$ is the minimum of the rayleigh quotient $\frac{\langle v, Av \rangle}{\langle v, v \rangle}$.

Suppose for contradiction and WOLOG that $\lambda_{min}(A_1) < \lambda_{min}(A)$

then take the eigenvector $v_1 \in \mathbb{R}^n$ and extend it by 0's so that it is in $\mathbb{R}^{2n}$

then $\frac{\langle v_1, Av_1 \rangle}{\langle v_1, v_1 \rangle} = \lambda_{min}(A_1) < \lambda_{min}(A)$

but since $\lambda_{min}(A_1)$ can be expressed as a rayleigh quotient of A, it cannot be smaller than the minimum rayleigh quotient, which by Courant-Fischer is $\lambda_{min}(A)$.

6

(a) see MatLab implementation of PCA Algorithm on Gaussian_noisy and Uniform_noisy.

(b) From code execution:

Principal components are $V_2 = \begin{pmatrix} -.8667 \\ .4989 \end{pmatrix}$ and $V_1 = \begin{pmatrix} .4989 \\ .8667 \end{pmatrix}$

with respective eigenvalues $\lambda_2 = .0623$ and $\lambda_1 = 14.9475$

therefore V_2 has $\frac{\lambda_2}{\lambda_1 + \lambda_2} \times 100 \approx .5\%$ of the variance

and V_1 has $\frac{\lambda_1}{\lambda_1 + \lambda_2} \times 100 \approx 99.5\%$ of the variance

from this we can conclude that there is far more variance in the second principal component, which is approximately in the positive, positive direction, hence the two features are highly correlated.

(c) From code execution:

Principal components are $V_2 \begin{pmatrix} -1 \\ -.0028 \end{pmatrix}$ and $V_1 = \begin{pmatrix} -.0028 \\ 1 \end{pmatrix}$

with respective eigenvalues $\lambda_2 = .0718$ and $\lambda_1 = .0892$

so V_2 has $\approx 44\%$ of the variance and V_1 has $\approx 56\%$

from this we conclude there is around equal variance for each component, but the directions are basically along the axes indicating no correlation.

7

credit to Satchel for help w/ my code. and Chat GPT for syntax questions.

(a) see MatLab implementation to view low-dimensional representation of the data.

This representation is actually quite good as we can see 3 fairly distinct clusters corresponding to the digits 0, 6, and 7.

we can also see that 6 and 0 are close together which makes sense since they both share an important feature: a closed loop.

(b) see MatLab implementation for plot

results look exactly as expected, the more components we consider, the smaller the error, but of use to us is the observation that error gets small very quickly, so we don't need all 784 components to get back most of the information.