Homework 2

Part 1

PCA

1. PCA can be explained from two different perspectives. What are the two perspectives?

2. The first principal direction is the direction in which the projections of the data points have the largest variance in the input space. We use λ_1 to represent the first/largest eigenvalue of the covariance matrix, w_1 to denote the corresponding principal vector/direction (w_1 has unit length i.e., L2 norm is 1), μ to represent the sample mean, and x to represent a data point. The deviation of x from the mean μ is $x - \mu$.

$y = PCA(x)$ is implemented in sk-learn with "whiten=True", and the number of components/elements of y is usually less than the number of components/elements of x

(1) what is the scalar-projection of x in the direction of w_1 ?

(2) what is the scalar-projection of the deviation $x - \mu$ in the direction of w_1?

(3) what is the first component of y ?

note: compute y using w_1 , x, μ, and λ_1

(4) assuming y only has one component, then we do inverse transform to recover the input

$$\tilde{x} = PCA^{-1}(y)$$

compute $\tilde{x}$ using μ, y, λ_1 and w_1: $\tilde{x}$ =???

(5) assuming x and y have the same number of elements, and we do inverse transform to recover the input

$$\tilde{x} = PCA^{-1}(y)$$

what is the value of $x - \tilde{x}$?

3. Show that PCA is a linear transform of $x - \mu$

Note: must use the definition on http://mathworld.wolfram.com/LinearTransformation.html

Maximum Likelihood Estimation and NLL loss
(This is a general method to estimate parameters of a PDF using data samples)

4. Maximum Likelihood Estimation when the PDF is an exponential distribution.
Suppose we have *N* i.i.d. (independently and identically distributed) data samples $\{x_1, x_2, x_3, \ldots, x_N\}$ generated from a PDF which is assumed to be an exponential distribution. $x_n \in \mathcal{R}^+ \; for \; n = 1 \; to \; N$, which means they are positive scalars. This is the PDF:

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & for\; x \geq 0 \\ 0 & otherwise \end{cases}$$

Your task is to build an NLL (negative log likelihood) loss function to estimate the parameter λ of the PDF from the data samples.

(1) write the NLL loss function: it is a function of the parameter λ
(2) take the derivative of the loss with respect to λ, and set the result to 0.
After some calculations, you will obtain an equation about $\lambda =$******
Hint: read NLL in the lecture of GMM

5. Maximum Likelihood Estimation when the PDF is histogram-like.

A histogram-like PDF $f(x)$ is defined on a 1-dimensional (1D) space that is divided into fixed regions/intervals. So, $f(x)$ takes constant value h_i in the i-th region. There are K regions. Thus, $\{h_1, h_2, \dots, h_K\}$ is the set of (unknown) parameters of the PDF. Also, $\sum_{i=1}^{K} h_i \Delta_i = 1$, where Δ_i is the width of the i-th region.

Now, we have a dataset of N samples $\{x_1, x_2, x_3, \dots, x_N\}$, and N_i is the number of samples in the i-th region. The task is to find the best parameters of the PDF using the samples.

(1) write the NLL loss function: it is a function of the parameters
Note: it is a constrained optimization problem, we need to use Lagrange multiplier to convert constrained optimization to unconstrained optimization. Thus, add $\lambda(\sum_{i=1}^{K} h_i \Delta_i - 1)$ to the loss function, where λ is the Lagrange multiplier.

(2) take the derivative of the loss with respect to h_i, set it to 0, and obtain the best parameters along with the value of λ.

Part 2

Complete the task in H2P2T1.ipynb and H2P2T2.ipynb

Note: It is very time consuming to fit a GMM to high dimensional data, and therefore PCA + GMM is the "standard" approach.

Grading: the number of points

	Undergraduate Student	Graduate Student
1 (PCA)	2	2
2 (PCA)	15	10
3 (PCA)	3	3
4 (NLL)	5 bonus points	5
5 (NLL)	N.A.	5 bonus points
H1P2T1	15	15
H2P2T2	15	15
Total number of points	50 +5	50 + 5

Extra Reading

PCA is widely used in many applications. Do a google scholar search with PCA + some field, e.g., PCA +bioinformatics or PCA + finance, you will find relevant papers.

https://www.nature.com/articles/s41467-018-04608-8

There are many variants of PCA, such as sparse PCA and kernel PCA that are implemented in sk-learn.

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.72.7798&rep=rep1&type=pdf

https://www.di.ens.fr/sierra/pdfs/icml09.pdf

https://www.di.ens.fr/~fbach/sspca_AISTATS2010.pdf

Which one is good for your application? Test different algorithms and find the best. Remember that machine learning is more like an experimental science: you need to run lots of experiments.