**PCA**

**Homework 2**

**Part 1**

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

* Maximum Variance
  + Find various w’s such that each w maximizes the variance of all the data points (C) in a certain dimension M, so therefore each w is an eigenvector of C. By finding the maximum variances they can each be scaled accordingly to better round out and generalize the data in all its dimensions.
* Minimum Reconstruction
  + By finding the mean object mu, use a series of sums of modifications (w) – otherwise called components - with certain weights (B) such that the error between reconstructed objects in this way and the actual object it is trying to reconstruct is minimized aiming to use as few components as possible.

1. The first principal direction is the direction in the input space, along which the projections of the data points have the largest variance. We use 𝜆1 to represent the first/largest eigenvalue of the covariance matrix,

𝑤1 to denote the corresponding principal vector/direction (𝑤1 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 − 𝜇.

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

𝑦 is usually less than the the number of components/elements of 𝑥

1. What is the scalar-projection of 𝑥 in the direction of 𝑤1 ?
   * 𝜆1
2. what is the scalar-projection of the deviation 𝑥 − 𝜇 in the direction of 𝑤1?
   * B­1 = 𝑤1­T(𝑥1 – 𝜇)
3. what is the first component of 𝑦 ? (whiten=True) hint: it is a function of 𝑤1 , 𝑥, 𝜇, and 𝜆1
   * [𝑤1­T(𝑥1 – 𝜇)] / 𝜆1
4. assuming 𝑦 only has one component, then we do inverse transform to recover the input

𝑥̃ = 𝑃𝐶𝐴−1(𝑦)

compute using 𝜇, 𝑦, 𝜆1 and 𝑤1: 𝑥̃ =? ? ?

* + 𝑥̃ = 𝑦\*𝜆1 \* ­𝑤1 + 𝜇

1. assuming 𝑥 and y have the same number of elements, and we do inverse transform to recover the input

𝑥̃ = 𝑃𝐶𝐴−1(𝑦)

what is the value of 𝑥 − 𝑥̃ ?

* + 𝑥 − 𝑥̃ = 0 since K = M so x = 𝑥̃

1. **(Really 4 but Word changed the counting of the questions)**

Maximum Likelihood Estimation when the PDF is an exponential distribution.

Suppose we have *N* i.i.d. (independently and identically distributed) data samples {𝑥1, 𝑥2, 𝑥3, … , 𝑥𝑁} generated from a PDF which is assumed to be an exponential distribution. 𝑥𝑛 ∈ ℛ+ 𝑓𝑜𝑟 𝑛 = 1 𝑡𝑜 𝑁, which means they are positive scalars. This is the PDF:

𝑓(𝑥) = {𝜆𝑒−𝜆𝑥 𝑓𝑜𝑟 𝑥 ≥ 0

0 𝑜𝑡ℎ𝑒𝑟𝑤𝑖𝑠𝑒

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.

−(xn)

1. 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 𝜆 =∗∗∗∗∗∗

Hint: read NLL in the lecture of GMM

**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, 2 (PCA) | 10 | 10 |
| 3 (PCA) | 5 bonus points | 5 |
| 4 (NLL) | 10 | 10 |
| 5 (NLL) | N.A. | 5 bonus points |
| H1P2T1 | 15 | 10 |
| H2P2T2 | 15 | 15 |
| Total number of points | 50 +5 | 50 + 5 |

Note: If you want to get A+, you need the bonus points.

**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](http://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](http://www.di.ens.fr/sierra/pdfs/icml09.pdf)

https://[www.di.ens.fr/~fbach/sspca\_AISTATS2010.pdf](http://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.