

# Homework 1 for CSI 431

Due: Tuesday, Sep 29, 23:59pm

**All homeworks are individual assignments. This means: write your own solutions and do not copy code/solutions from peers or online. Should academic dishonesty be detected, the proper reporting protocols will be invoked (see Syllabus for details).**

Instructions: Submit two files. One should be a write-up of all solutions and observations, as *LastnameFirstnameSolution.pdf*. The second should be an archive *LastnameFirstnameCode.zip* containing code and any results files.

## 1 [20 pts.] Data analysis

In a statistical survey of 2000 families on ownership of cars 30 families responded that they do not own a car, 470 families owned 1, 850 families owned 2, 490 families owned 3, 150 owned 4, and there were 10 families which owned 10 cars each.

**(a)[5 pts]** Plot the sample-based Probability Mass Function (PMF) and Cumulative Distribution Function (CDF) of the random variable corresponding to **number of cars per family**.

**(b)[5 pts]** Calculate the expected **number of cars per family** and its variance. Show the steps of your calculation.

**(d)[5 pts]** Calculate the expected value of **number of cars per family** if you exclude the families with 10 cars. Is the mean (as compared to part b) stable or sensitive to the removal of these data points? What other statistical measures are there to estimate the average behavior? Are they less or more stable with regards to the outliers? Justify your answer by computing those alternative measures for the original sample.

**(e)[5 pts]** Create a box plots for the **number of cars per family** for the variables  $X_{all}$  (cars per family including all observations) and  $X_{-10}$  (cars per family after excluding the 10-car families). (don't use python libraries for this - just draw it by hand in your submission). Look here: <http://www.physics.csbsju.edu/stats/box2.html> for examples of box plots. You should have two boxes one for  $X_{all}$  and one for  $X_{-10}$  with their specific statistics: mean, median, ranges and inter-quantile ranges (see slides on numeric attributes for details).

## 2 [10 pts.] Irreducible data example

In class we discussed that not all datasets' dimensionality can be successfully reduced using PCA.

(a)[2 pts] Discuss the cases when PCA will fail.

(b)[3 pts] How do we quantify that it fails?

(c)[5 pts] Provide an example dataset of 2D points (specify the points as vectors of numbers) in which PCA will not work well for dimensionality reduction. Explain why. *Hint: Think of 2D points and reduction to 1D.*

## 3 [50 pts] Dimensionality reduction

For this question you will use the Cloud Dataset from the UCI ML repository: <https://archive.ics.uci.edu/ml/datasets/Cloud>. Read about it to get familiar with what is measured. Within the data, there are two datasets: DB #1 and DB #2. For this homework, just use the 1024 vectors in DB #1. Use python for all your programming. You will have to submit your code in LastnameFirstnameCode.zip together with the relevant write-up in the main solution file LastnameFirstnameSolution.pdf.

(a)[5 pts] Load the data into a python program and center it. Note: there should be a function called *center()* in your code that achieves this.

(b)[5 pts] Compute the covariance matrix of the data  $\Sigma$ . *Hint: by using the definition of sample covariance, as a matrix product or as a sum of outer products. See book for details.* Use Numpy for linear algebra computations (<https://docs.scipy.org/doc/numpy-1.13.0/reference/routines.linalg.html>). As a result you should have a function *covar()* in your code which does not use the built-in covariance functions.

(c)[5 pts] Compute the eigenvectors and eigenvalues of  $\Sigma$ . The numpy linear algebra module referenced above has a function that can help.

(d)[10 pts] Determine the number of principal components (PCs)  $r$  that will ensure 90% retained variance? How did you compute this? Provide a function in your code that determines  $r$  based on an arbitrary percentage  $\alpha$  of retained variance.

(e)[10 pts] Plot the first two components in a figure with horizontal axis (x) corresponding to the dimensions and vertical axis (y) corresponding to the magnitude of the component in this dimension. There will be 2 traces with d points

in this figure. Include the figure in your LastnameFirstnameSolution.pdf. Also save the top two components in a text file "Components.txt", with each component on a separated line and represented as  $d$  comma separated numbers (i.e. the file should have two lines with  $d$  numbers separated by commas). Include "Components.txt" in your LastnameFirstnameCode.zip.

(f)[10 pts] Compute the reduced dimension data matrix  $A$  with two dimensions by projection on the first two PCs. Plot the points using a scatter plot (two dimensional diagram that places each sample  $i$  according to its new dimensions  $a_{i1}, a_{i2}$ ). Discuss the observations. Are there clusters of close-by points? What is the retained variance for  $r = 2$ ? Argue for or against whether these are sufficient dimensions.

(g)[5 pts] Study the PCA implementation in python's sklearn library <https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html#sklearn.decomposition.PCA>. Do PCA using the library on the same data. Do the eigenvalues approximately match to what you computed above?

## 4 [20 pts.] Kernel methods

Consider the problem of finding *the most dissimilar diametric pair (MDDP)*: this is a pair of data points that are dissimilar from the mean and also dissimilar from each other. Below is an algorithm that would find such a pair given a data matrix  $D$ :

---

### Algorithm 1: MDDP( $D$ )

---

**Result:**  $a, b$  - the most dissimilar diametric points in  $D$

Compute the data mean  $\mu = \text{mean}(D)$ ;

$s = +\infty$ ;

**for**  $i$  **in**  $(1 \dots n)$  **do**

**for**  $j$  **in**  $(i + 1 \dots n)$  **do**

$\text{temp} = x_i^T \mu + x_j^T \mu + x_i^T x_j$ ;

**if**  $\text{temp} < s$  **then**

$s = \text{temp}$ ;

$a = x_i$ ;

$b = x_j$ ;

**end**

**end**

**end**

---

The algorithm computes the sum of inner products  $x_i^T \mu + x_j^T \mu + x_i^T x_j$  for each pair of points and returns the pair with the lowest such quantity.

(a)[5 pts] Demonstrate the execution of this algorithm on the following data

matrix of 2D instances:  $D = \begin{pmatrix} 0 & 1 \\ 1 & 3 \\ 5 & 0 \\ 2 & 4 \end{pmatrix}$ . Show the steps and the resulting MDD pair of points.

**(b)[15 pts]** As we discussed in class sometimes we would like to kernelize methods to handle non-linearity in data. Provide a pseudo-code for a kernel version of the MDDP algorithm above. The goal is to kernelize the algorithm for an arbitrary kernel **Hint: Assume that you can compute the kernel matrix  $K$ , corresponding to some mapping  $\phi()$  and then use the basic kernel operations we discussed in class and also in the book, to derive the steps of MDDP in terms of elements in  $K$ .**