

# Homework #1

Foundations of Computer and Data Science CS-596

**Problem 1:** Consider the matrix

$$A = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 + \epsilon \end{bmatrix}.$$

a) Find the eigenvalues/eigenvectors of  $A$  assuming  $\epsilon \neq 0$ . Force your eigenvectors to have unit norm. b) Diagonalize  $A$  using the eigenvalues/eigenvectors you computed. c) Start now sending  $\epsilon \rightarrow 0$ . What do you observe is happening to the matrices you use for diagonalization as  $\epsilon$  becomes smaller and smaller? So what do you conclude when  $\epsilon = 0$ ?

**Problem 2:** Let  $A, B$  be two matrices of the same dimensions  $k \times m$ . a) With direct computation show that  $\text{trace}(AB^T) = \text{trace}(B^T A) = \text{trace}(BA^T) = \text{trace}(A^T B)$ . b) Use question a) to compute  $E[\mathbf{x}^T A \mathbf{x}]$  where  $E[\cdot]$  denotes expectation,  $A$  is a constant matrix and  $\mathbf{x}$  is a random vector for which we know that  $E[\mathbf{x}\mathbf{x}^T] = Q$ . *Hint: The trace of a scalar is the scalar itself. Trace and expectation are linear operations, so we can change their order of application.* c) Using the previous properties show that for any matrix  $A$  of dimensions  $k \times k$  we have  $\text{trace}(A) = \text{trace}(U A U^{-1})$  for any nonsingular matrix  $U$  of dimensions  $k \times k$ . In other words that the trace does not change if we apply a similarity transformation. d) Use question c) to prove that if matrix  $A$  of dimensions  $k \times k$  is diagonalizable then its trace is equal to the sum of its eigenvalues (actually this is true even if the matrix is not diagonalizable). e) Regarding question d) how do you explain this equality given that when  $A$  is real the trace is also real whereas the eigenvalues can be complex? f) Using again question d) what can you say about the coefficient  $c_{k-1}$  of the characteristic polynomial  $\lambda^k + c_{k-1}\lambda^{k-1} + \dots + c_0$  of  $A$ . We recall that we already know that  $c_0 = (-1)^k \lambda_1 \dots \lambda_k = (-1)^k \det(A)$ .

**Problem 3:** A matrix  $A$  is called *nilpotent* if  $A^r = 0$  for some integer  $r > 1$ . a) Show that all eigenvalues of  $A$  must be equal to 0. b) Is such a matrix diagonalizable? c) Give an example of a  $2 \times 2$  matrix which is nilpotent. d) If we apply a similarity transformation to a nilpotent is the resulting matrix nilpotent? Apply the previous observation to your example in c) to obtain a second example of a nilpotent matrix.

**Problem 4:** There is clock and at each period of the clock you obtain a new pair  $(y_t, X_t)$  where  $y_t$  is scalar and  $X_t$  vector. We are interested in modeling  $y$  as a cross product of  $X$  with a vector  $A$

$$y_n = X_n^T A + e_n, \text{ for } n = 1, 2, \dots, t$$

where  $\{e_n\}$  denotes the error term between the model  $X_n^T A$  and the target value  $y_n$ . At each time  $t$  we have available the data  $(y_t, X_t), (y_{t-1}, X_{t-1}), \dots, (y_1, X_1)$  and we would like to find the best  $A$  that minimizes the sum of the squared errors, namely perform the minimization

$$\min_A \sum_{n=1}^t (y_n - X_n^T A)^2 \tag{1}$$

a) Show that if  $A_t$  denotes the solution, at time  $t$ , of the optimization in (1) then

$$A_t = R_t^{-1} U_t, \text{ where } R_t = \sum_{n=1}^t X_n X_n^T, \quad U_t = \sum_{n=1}^t y_n X_n. \tag{2}$$

b) At the next time instant  $t + 1$  when we receive the new pair  $(y_{t+1}, X_{t+1})$  we need to reapply the formula in (2) to compute the new estimate  $A_{t+1}$ . Express the computational complexity in terms of the size  $k$  of the vector  $X_t$ .

c) Show that we can compute  $A_{t+1}$  by using the previous  $A_t$  and applying the following updating formulas

$$e_{t+1} = y_{t+1} - X_{t+1}^\top A_t, \quad K_{t+1} = Q_t X_{t+1}, \quad \gamma_{t+1} = 1 + X_{t+1}^\top K_{t+1},$$

$$A_{t+1} = A_t + e_{t+1} K_{t+1}, \quad Q_{t+1} = Q_t - \frac{1}{\gamma_{t+1}} K_{t+1} K_{t+1}^\top,$$

where  $Q_t = R_t^{-1}$ . What is the complexity of the algorithm in terms of  $k$ ? This recursive algorithm is known as *Recursive Least Squares (RLS)* and is one of the most famous learning algorithms in the literature. *Hint: To prove the validity of RLS you need to apply the matrix inversion lemma onto  $R_{t+1} = R_t + X_{t+1} X_{t+1}^\top$  in order to compute  $Q_{t+1}$  in terms of  $Q_t$ . From (2) you must also use the fact that  $U_{t+1} = U_t + y_{t+1} X_{t+1}$  and  $A_{t+1} = Q_{t+1} U_{t+1}$ .*

**There is a meeting on Friday, September 27, at 5PM, in CoRE-101 to discuss YOUR questions. Mr. Anastasios Stathopoulos, our TA, will answer them.**

**Your reports, in *hard copy*, are due on Monday, September 30, in CBIM, between 10:00 and 11:00AM. Mr Stathopoulos will collect them.**

**Please STRICTLY respect the indicated time period because Mr. Stathopoulos has many other tasks, besides TAing. Late/Early submission will NOT be accepted neither by the TA nor by me.**