UTSA

CS 6243 Machine Learning

EE 6363: Advanced Machine Learning

Assignment: HW1

I. Assignment Instructions

Solve all given problems.

Submit your report in the dedicated folder on CANVAS.

Submit your report by the deadline. Delayed reports will not be graded.

Deadline: 9/19/2023, 11:59pm.

Reports must be typed and uploaded in PDF format. Handwritten reports will not be graded.

Any requested code must be presented at the end of your report in a dedicated appendix, titled “Code.” All codes must have adequate comments.

All display equations must be numbered. All figures must be numbered and captioned.

You can use any source (notes, books, online), but it must be cited in a References List at the end of your report.

Do not outsource this assignment (or parts of it) to another intelligent entity (whether human or AI). Except for what you explicitly cite, what you submit must be your own intellectual work.

Grade: This assignment corresponds to 6% of your final grade. You will be graded based on 1) correctness, 2) completeness, 3) clarity (equal weight).

Teams: Work in teams. Each team member must submit the exact same report on CANVAS (only students that submit a report will receive a grade). It is expected that all team members will put equal effort into this assignment.

The instructor may determine each student's grade on this assignment by individual examination during course office hours.

Terminology: Present = just show result. Derive = show all math steps. Implement in python = write python code and presented it in the dedicated appendix. Discuss/Compare = discuss/compare in words (no need for math/code).

II. Problems

Problem 1: (40%) – Probability and Random Variables

Task 1:

Consider random events $\{E_i\}_{i\in[3]}$ and given: $P(E_i)\ \forall i \in [3]$, $P\big(E_i \cap E_j\big)\ \forall (i,j) \in [3]^2$, and $P\big(\cap_{i\in[3]} E_i\big)$.

- Present: What is the formula for the probability of the union of the three events? *Notation*: $[n]$ is short notation for $\{1,2,\dots,n\}$.

Task 2:

Consider random variable (RV) X uniformly distributed in $[-2,2]$ and RV $Y = X^2$.

- Derive:
  - Are X and Y uncorrelated?
  - Are they independent?

Task 3:

Consider deterministic variables $\{A_i\}_{i\in[N]}$, RVs $\{X_i\}_{i\in[N]}$ that are zero-mean with covariance $E\{X_i X_j\} = C_{i,j}\ \forall (i,j) \in [N]^2$, and RV $Y = \sum_{i\in[N]} A_i X_i$.

- Derive:
  - The mean of Y.
  - The variance of Y (in terms of $\{A_i\}_{i\in[N]}$ and $\{C_{i,j}\}_{(i,j)\in[N]^2}$).

    Notation: $[n]^k = \{(i_1, i_2, \dots, i_k) : i_j \in [n] \forall j\}$.

Task 4:

Consider RVs $\{X_i\}_{i\in[N]}$ that are zero-mean, independent, and each has variance 1. Consider random vector $x = [X_1,\ X_2, \dots, X_N]^{\mathrm{T}}$.

- Derive:
    - The covariance matrix of x.
    - The covariance matrix of $x + a$, in terms deterministic vector a.

Problem 2: (20%) – Linear Algebra

Task 1:

Consider vector $y = [Y_1,\ Y_2, \dots, Y_N]^{\mathrm{T}}$ and matrices $A = y\,1_M^{\mathrm{T}}$ and $B = A + A^{\mathrm{T}}$.

- Discuss:
    - What is the rank of A?
- Derive the highest singular value of A (in terms of $\{Y_i\}_{i\in[N]}$).
- Derive the third highest singular value of B.

Task 2:

Consider positive-semidefinite symmetric $N \times N$ matrix R with SVD: $R = U\Lambda U^T$, $\Lambda = diag(\lambda_1,\lambda_2, \dots, \lambda_N)$, and $\lambda_i \geq \lambda_j \geq 0$ for $i < j$.

- Derive the highest singular value of $R + I_N$.

Problem 3: (40%) – Least Squares Optimization

Task 1: Stabilized SVD

- Implement in python a function that given generic $m \times n$ matrix A and constant $\epsilon > 0$, calculates the SVD $A_{m\times n} = US_{m\times n}V^T$, where $S_{i,i} \geq S_{j,j}$ for any $j \geq i$, and returns the maximum k such that $\frac{S_{k,k}}{S_{1,1}} \geq \epsilon$. Then, it returns the "stabilized" SVD $A \approx U_s S_s V_s^T$ where $U_s = [U]_{:,1:k}$, $S_s = [S]_{1:k,1:k}$, $V_s = [V]_{:,1:k}$. Use the above stabilized-SVD on $A = \begin{bmatrix} 3 & 1 & 2 \\ 1 & 3 & 1 \\ 2 & 1 & 3 \end{bmatrix}$.
- Present and compare the obtained singular vectors and singular values to those of standard SVD.
- Discuss:
    - What is the role of ϵ?

- What if we set $\epsilon = 0$?
- Why is this a “stabilized” SVD?

Task 2: LS Single-Shot Solution

Consider the least squares problem $\min_{x \in \mathbb{R}^3} \|y - Ax\|_2^2$, where matrix A is as in Task 1 and $y = [2,3,4]^T$.

- Implement in python the optimization by means of the stationarity condition. Solve any SLE by means of inversion.
- Then implement again the optimization by means of the stationarity condition, this time solving any SLE by means of the SVD-based approach presented in the class. Perform SVD by means of the stabilized-SVD that you implemented in Task 1, for appropriately chosen ϵ.
- Present and compare the optimal points obtained by the two approaches.
- Discuss:
    - What is the value of the metric at the optimal point?
    - Why does this make sense?

Task 3: LS GD Solution

- Implement in python GD iterations that solve the above LS problem. Initialize GD iterations at $x = 0_3$ and perform 300 updates. Implement GD (a) with constant step size $\gamma = 0.01$ and (b) with adaptive step that minimizes the metric at the next GD update.
- Present the plots of the metric vs GD iteration for both (a) (red) and (b) (blue) in the same figure.
- Discuss:
    - How do the two curves compare?
    - What is the metric value upon convergence?
    - What is x at the last update for (a) and (b) and how does it compare with the single-shot solution of Part 2?

Task 4: LS with Low-Rank Coefficients

- Repeat parts 2-3 for $A = [3,1,2; 9,3,6; 2,1,3]$.
- Discuss:

- Is the inversion-based solution possible now?
- Is the value of the metric at the closed form solution same as before? If not, why?
- Did you have to change constant step-size for GD to converge? Did (b) converge faster in part 3 or part 4?