UTSA
CS 6243, EE 4463, EE 5573
Spring 2025
Assignment: HW-1
Topic: Unsupervised Machine Learning: Dimensionality Reduction with SVD

1 Assignment Instructions

Submission: Submit your report in the designated drop box on CANVAS by the corresponding deadline.

Format: Reports must be typed and uploaded as a PDF. Handwritten reports will not be graded. Any requested code should be presented at the end of your report in an appendix titled "Code." All code must include appropriate comments. Display equations and figures must be numbered, labeled, and captioned accordingly.

References: You may use any source (notes, books, online), but all sources must be cited in a "Reference List" at the end of your report.

Originality: You are not allowed to outsource this assignment (or parts of it) to another (naturally or artificially) intelligent entity (this includes ChatGPT and similar tools). Except for explicitly cited content, by submitting your work you verify and commit that it is your own intellectual work.

Grading: Your assignment will be graded based on three equally weighted factors: Correctness, Completeness, and Clarity. For each problem, each criterion will be rated as follows: 100% (Excellent), 90%, 70%, 50% (Fair Effort), 30%, 10%, or 0% (Missing Effort). Example: A correct final result (100%), with an almost complete method (90%), and almost clear presentation (90%) would receive $100\% \times 90\% \times 90\% = 81\%$. If the total points for the problem are 18, this would earn 14.6 points (rounded to the nearest first decimal).

Teams: Work in your determined project/assignment teams (see CANVAS). Each team member must individually submit a copy of the same team report on CANVAS. Only team members who make a submission before the deadline will receive a grade. All team members are expected to contribute equally and in general will receive the same grade, unless there are indications of imbalanced efforts, in which case the instructor may assign individual grades based individual examination of each student.

Task Terminology:

- **Present:** Show only the final result in math.
- **Derive:** Show all the necessary mathematical steps leading to the final result.
- **Compute:** Write the Python code and present it.
- **Plot:** Write Python code that computes and plots; present both the code and the plots.
- **Discuss:** Discuss in words and reason/justify in detail (no need for math or code).

2 Problems

2.1 Problem 0 (0 points)

Use the MNIST dataset (`from tensorflow.keras.datasets import mnist`) and extract only the digits 0, 1, and 9 for analysis. Vectorize all training images and append them as columns to a matrix X. Ensure that the data type of X is set to `float32`.

2.2 Problem 1: Norms (5 points)

Calculate the average vector (centroid) of each of the 3 digits in X.

Compute: Compute the 1-norm, 2-norm, and 3-norm of each centroid vector.

2.3 Problem 2: Singular Value Decomposition (SVD) (5 points)

Perform SVD on X using `np.linalg.svd` and decompose it into matrices U, S, V.

Plot: In Fig. 1 of your report, plot the singular values S in decreasing order.

Discuss: What do the singular values represent in terms of the structure of the data? What does the singular value profile reveal in terms of redundancy in the data?

2.4 Problem 3: Low-Rank Approximation (18 points)

Construct low-rank approximations Y_K for $K \in K_{\text{set}} := \{1, 51, 101, \ldots, 784\}$. Let Q_K contain the singular vectors in U corresponding to the K highest singular values. Define:

$$Y_K = Q_K Q_K' X.$$

Compute: Compute the approximation error:

$$e(X, Y_K) = \|X - Y_K\|_F.$$

Plot: In Fig. 2 of your report, plot $e(X, Y_K)$ versus K.

Discuss: What do you observe in terms of low-rank approximation of data?

2.5 Problem 4: Storage Efficiency (18 points)

Compute: Compute the number of elements needed to store X and the minimum number of elements needed to store each low-rank approximation Y_K.

Plot: In Fig. 3, plot storage requirements for Y_K versus K. Include a horizontal benchmark line for X.

Discuss: What do you observe in terms of low-rank storage savings? Compare Fig. 2 and Fig. 3.

2.6 Problem 5: Visualizing Low-Rank Projections (18 points)

Project the data on the span of the two dominant singular vectors.

Compute: Compute the centroid of the subspace representations for each class.

Plot: In Fig. 4, create a scatter plot of projections, color-coded by digit class. Include centroids as star markers.

Discuss: What do you observe in terms of low-rank visualization?

Discuss: Propose a classification method using the orthonormal basis and centroids.

2.7 Problem 6: Low-Rank Denoising (18 points)

Create a noisy version X_n by corrupting each pixel independently with probability 5%, replacing it with 0 or 255.

Perform SVD on X_n and construct low-rank approximations:

$$Y_K = Q_K Q_K' X_n.$$

Compute: Compute errors $e(X, Y_K)$ and $e(X_n, Y_K)$.

Discuss: What do these errors capture? How are they different?

Plot: Plot $e(X_n, Y_K)$ and $e(X, Y_K)$ versus K, including a benchmark $e(X, X_n)$.

Discuss: Are the curves monotonically decreasing? Explain.

Discuss: How does low-rank approximation perform denoising?

2.8 Problem 7: Solving Linear Equations (18 points)

Consider the following matrices and vectors:

$$A = \begin{bmatrix} -2.74125009 & 2.24215689 & -0.60553211 & -0.16755625 \\ -0.34868395 & 0.29538923 & -0.45259498 & 0.50015934 \\ 2.49664208 & 0.27798324 & 2.00739274 & 0.2197803 \end{bmatrix}$$

$$y_A = \begin{bmatrix} 0.61339829 \\ 0.11012282 \\ -0.06426754 \end{bmatrix}$$

$$B = \begin{bmatrix} -2.74125009 & -0.34868395 & 2.49664208 \\ 2.24215689 & 0.29538923 & 0.27798324 \\ -0.60553211 & -0.45259498 & 2.00739274 \\ -0.16755625 & 0.50015934 & 0.2197803 \end{bmatrix}$$

$$y_B = \begin{bmatrix} 0.66761214 \\ 0.35931116 \\ 0.74289966 \\ 0.02979187 \end{bmatrix}$$

$$y_{B2} = \begin{bmatrix} 0.24982762 \\ -0.45768269 \\ 0.22778277 \\ 0.6341392 \end{bmatrix}$$

$$C = \begin{bmatrix} 0.31997336 & 0.43316234 & -0.33457014 & -0.34017903 \\ 1.12969075 & 1.52931319 & -1.18122581 & -1.20102843 \\ 0.2008776 & 0.27193705 & -0.21004138 & -0.21356262 \end{bmatrix}$$

$$y_C = \begin{bmatrix} 0.1421664 \\ 0.50192948 \\ 0.08925132 \end{bmatrix}$$

$$y_{C2} = \begin{bmatrix} -1.01480112 \\ 0.4115211 \\ -0.45229071 \end{bmatrix}$$

$$D = \begin{bmatrix} 0.07999334 & 0.28242269 & 0.0502194 \\ 0.10829058 & 0.3823283 & 0.06798426 \\ -0.08364254 & -0.29530645 & -0.05251035 \\ -0.08504476 & -0.30025711 & -0.05339065 \end{bmatrix}$$

$$y_D = \begin{bmatrix} 0.41615372 \\ 0.56336601 \\ -0.43513813 \\ -0.44243299 \end{bmatrix}$$

$$y_{D2} = \begin{bmatrix} 0.47277025 \\ -0.64357627 \\ 1.30059591 \\ 1.426948 \end{bmatrix}$$

Consider also the following cases.

- **Case 1:** $G = A$, $f = y_A$
- **Case 2:** $G = B$, $f = y_B$
- **Case 3:** $G = B$, $f = y_{B2}$
- **Case 4:** $G = C$, $f = y_C$
- **Case 5:** $G = C$, $f = y_{C2}$
- **Case 6:** $G = D$, $f = y_D$
- **Case 7:** $G = D$, $f = y_{D2}$

For each of the cases above, perform the following tasks.

Compute: Compute the size and rank of matrix G.

Compute: Compute the projection matrix for the span of G and determine if f is in the span of G.

Discuss: Is the system $f = Gw$ overdetermined or underdetermined?

Discuss: Does $f = Gw$ have none, one, or infinitely many solutions?

Compute: Compute a solution if one exists.