HOMEWORK 8

HAMSALEKHA PREMKUMAR

Instructions: Although this is a programming homework, you only need to hand in a pdf answer file. There is no need to submit the latex source or any code. You can choose any programming language, as long as you implement the algorithm from scratch.
Use this latex file as a template to develop your homework. Submit your homework on time as a single pdf file to Canvas. Please check Piazza for updates about the homework.

1 Principal Component Analysis [50 pts]

Download three.txt and eight.txt. Each has 200 handwritten digits. Each line is for a digit, vectorized from a 16x16 gray scale image.

1. (5 pts) Each line has 256 numbers: they are pixel values (0=black, 255=white) vectorized from the image as the first column (top down), the second column, and so on. Visualize the two gray scale images corresponding to the first line in three.txt and the first line in eight.txt.

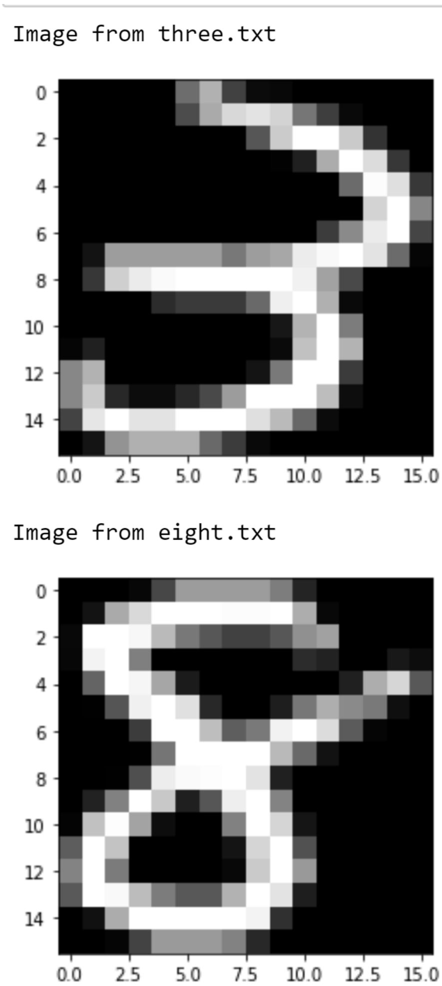


Figure 1: Gray scale images corresponding to the first line in three.txt and the first line in eight.txt

2. (5 pts) Putting the two data files together (threes first, eights next) to form a $n \times D$ matrix X where $n = 400$ digits and $D = 256$ pixels. Note we use $n \times D$ size for X instead of $D \times n$ to be consistent with the convention in linear regression. The ith row of X is $x_i^\top$, where $x_i \in \mathbb{R}^D$ is the ith image in the combined data set. Compute the sample mean $y = \frac{1}{n}\sum_{i=1}^{n} x_i$. Visualize y as a 16x16 gray scale image.

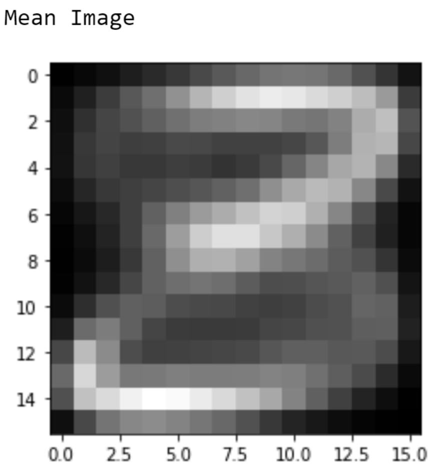


Figure 2: Mean Image

3. (10 pts) Center X using y above. Then form the sample covariance matrix $S = \frac{X^\top X}{n-1}$. Show the 5x5 submatrix $S(1 \ldots 5, 1 \ldots 5)$.

```
S Submatrix
[[  59.16729323  142.14943609   28.68201754   -7.17857143  -14.3358396 ]
 [ 142.14943609  878.93879073  374.13731203   24.12778195  -87.12781955]
 [  28.68201754  374.13731203 1082.9058584   555.2268797    33.72431078]
 [  -7.17857143   24.12778195  555.2268797  1181.24408521  777.77192982]
 [ -14.3358396   -87.12781955   33.72431078  777.77192982 1429.95989975]]
```

Figure 3: 5x5 submatrix $S(1 \ldots 5, 1 \ldots 5)$

4. (10 pts) Use appropriate software to compute the two largest eigenvalues $\lambda_1 \geq \lambda_2$ and the corresponding eigenvectors v_1, v_2 of S. For example, in Matlab one can use eigs(S,2). Show the value of λ_1, λ_2. Visualize v_1, v_2 as two 16x16 gray scale images. Hint: their elements will not be in [0, 255], but you can shift and scale them appropriately. It is best if you can show an accompany "colorbar" that maps gray scale to values.

```
Two largest Eigen values: [237155.24629049 145188.35268683]
```

Figure 4: Two largest eigenValues

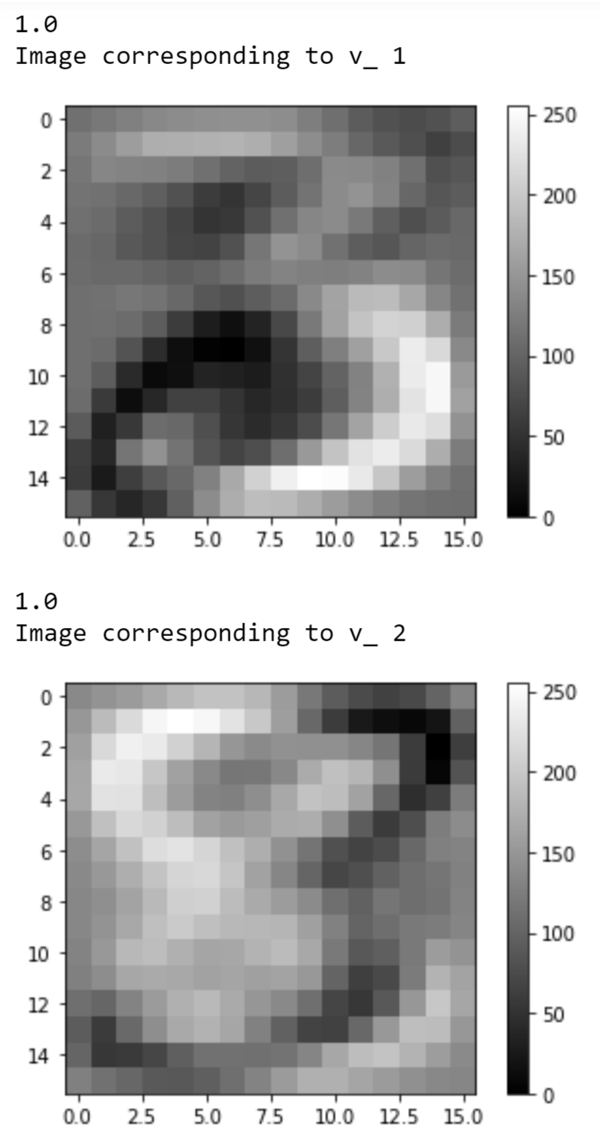


Figure 5: Eigen Vectors as images

missing

5. (5 pts) Now we project (the centered) X down to the two PCA directions. Let $V = [v_1 v_2]$ be the $D \times 2$ matrix. The projection is simply XV. Show the resulting two coordinates for the first line in three.txt and the first line in eight.txt, respectively.

```
Two coordinates for:
The first line in three.txt [ 136.20872784 -242.62848028]
The first line in eight.txt [-312.68702792  649.57346086]
```

Figure 6: Coordinates of 2 projected datapoints in two PCA directions

6. (5 pts) Now plot the 2D point cloud of the 400 digits after projection. For visual interest, color points in three.txt red and points in eight.txt blue. But keep in mind that PCA is an unsupervised learning method and it does not know such class labels.

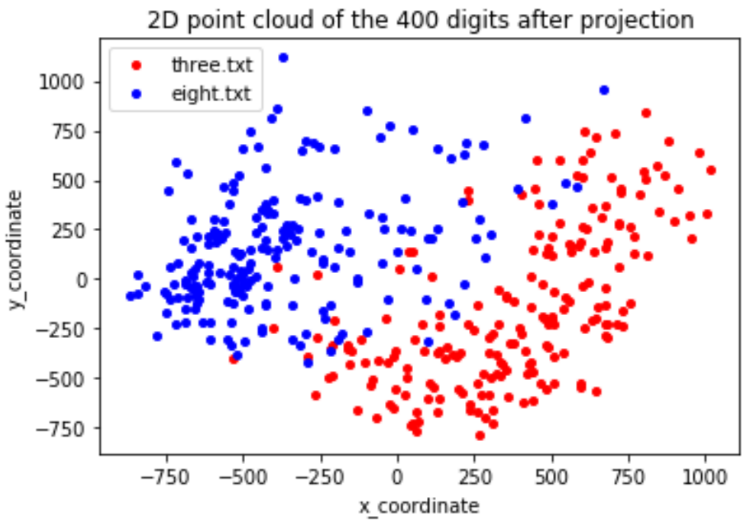
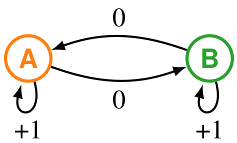


Figure 7: 2D point cloud of the 400 digits after projection

2 Q-learning [50 pts]

Consider the following Markov Decision Process. It has two states s. It has two actions a: move and stay. The state transition is deterministic: "move" moves to the other state, while "stay' stays at the current state. The reward r is 0 for move, 1 for stay. There is a discounting factor $\gamma = 0.9$.



The reinforcement learning agent performs Q-learning. Recall the Q table has entries $Q(s, a)$. The Q table is initialized with all zeros. The agent starts in state $s_1 = A$. In any state s_t, the agent chooses the action a_t according to a behavior policy $a_t = \pi_B(s_t)$. Upon experiencing the next state and reward s_{t+1}, r_t the update is:

$$Q(s_t, a_t) \Leftarrow (1 - \alpha)Q(s_t, a_t) + \alpha \left(r_t + \gamma \max_{a'} Q(s_{t+1}, a') \right).$$

Let the step size parameter $\alpha = 0.5$.
For the follow solutions, this format is used for both reward and Q table

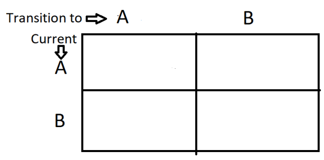


Figure 8: Format of table used for below solutions

1. Run Q-learning for 200 steps with a uniformly random behavior policy: $\pi_B(s_t) =$ move or stay with 1/2 probability for any s_t. Show the Q table at the end.

```
----------
round: 199
Transition: 1 -> 0
new Q value at 1 0 is: 8.2270658493042
Q table at  199 th step:
[[9.150261 8.284402]
 [8.227066 9.295782]]
----------
```

Figure 9: Uniformly random behavior policy

2. Reset and repeat the above, but with an ϵ-greedy behavior policy: at each state s_t, with probability $1-\epsilon$ choose what the current Q table says is the best action: $\arg\max_a Q(s_t, a)$; Break ties arbitrarily. Otherwise (with probability ϵ) uniformly chooses between move and stay. Use $\epsilon = 0.5$.

```
Q table at  199 th step:
[[9.760233  8.8141165]
 [8.721388  9.8348465]]
----------
```

Figure 10: Greedy behavior policy

3. Reset and repeat the above, but with a deterministic greedy behavior policy: at each state s_t use the best action $a_t \in \arg\max_a Q(s_t, a)$ indicated by the current Q table. If there is a tie, prefer move.

```
Q table at  199 th step:
[[0. 0.]
 [0. 0.]]
----------
```

Figure 11: Deterministic greedy behavior policy

4. Without doing simulation, use Bellman equation to derive the true Q table induced by the MDP.

$$V^*(s) \leftarrow E\left[r(s,\pi^*(s))\right] + \gamma E_{s'|s,\pi^*(s)}\left[V^*(s')\right]$$

$$Q(s,a) \leftarrow E\left[r(s,a)\right] + \gamma E_{s'|s,a}\left[V^*(s')\right]$$

Figure 12: Bellman Equations

$$\pi^*(s) \leftarrow \arg\max_a Q(s,a) \qquad V^*(s) \leftarrow \max_a Q(s,a)$$

Figure 13: Bellman Equations

We need to find Q function for every state to generate the Q table. For this, we need $V^*(s)$ for every state. We will find it for state A and B.

For State A,the choices are to stay or move. $V^*(A) = max_a(Q(s,a))$ We need to check the Q value for both move and stay option and pick the max of the two.
WKT, $Q(s,a) = r(s,a) + \gamma V(s',a')$

For stay action, $Q(A,'stay') = 1 + \gamma V^*(A)$
For move action, $Q(A,'move') = 0 + \gamma V^*(B)$

Due to symmetry in the graph, A and B occur with equal probability, $V^*(A) = V^*(B)$ Hence $V^*(A)$ is assigned the max of the two Q values, ie.,$Q(A,'stay')$
Thus, $V^*(A) = \frac{1}{1-\gamma}$
Given $\gamma = 0.9$, $V^*(A) = 10$
Similarly, $V^*(B) = 10$

Now to construct, Q table transitions AA,AB,BA,BB, we use the above Q(s,a) equation
For A to A: $1 + 0.9 * 10 = 10$
For A to B: $0 + 0.9 * 10 = 9$
Similarly, B to B: $= 10$ and
B to A: $= 9$

Thus the Q Table looks like $\begin{bmatrix} 10 & 9 \\ 9 & 10 \end{bmatrix}$

3 Extra Credit: VC dimension [10 pts]

Let the input $x \in X = \mathbb{R}$. Consider $F = \{f(x) = \text{sgn}(ax^2 + bx + c) : a, b, c \in \mathbb{R}\}$, where $\text{sgn}(z) = 1$ if $z \geq 0$, and 0 otherwise. What is $VC(F)$? Prove it.

4 Extra Credit: VC-dimension of Linear Separators [10 pts]

In this problem, you will prove that the VC-dimension of the class H_n of halfspaces (another term for linear threshold functions $f_{w,b}(x) = \text{sign}(w^\top x + b)$) in n dimensions is $n + 1$. We will use the following definition: The convex hull of a set of points S is the set of all convex combinations of points in S; this is the set of all points that can be written as $\sum_{x_i \in S} \lambda_i x_i$, where each $\lambda_i \geq 0$, and $\sum_i \lambda_i = 1$. It is not hard to see that if a halfspace has all points from a set S on one side, then the entire convex hull of S must be on that side as well.
(a) [lower bound] Prove that VC-dim$(H_n) \geq n + 1$ by presenting a set of $n + 1$ points in n-dimension space such that one can partition that set with halfspaces in all possible ways, i.e., the set of points are shattered by H_n. (And, show how one can partition the set in any desired way.)
(b) [upper bound part 1] The following is Radon's Theorem, from 1920's.

Theorem 1. Let S be a set of $n + 2$ points in n dimensions. Then S can be partitioned into two (disjoint) subsets S_1 and S_2 whose convex hulls intersect.

Show that Radon's Theorem implies that the VC-dimension of halfspaces is at most $n + 1$. Conclude that VC-dim$(H_n) = n + 1$.
(c) [upper bound part 2] Now we prove Radon's Theorem. We will need the following standard fact from linear algebra. If $x_1, \ldots, x_{n+1}$ are $n + 1$ points in n-dimensional space, then they are linearly dependent. That is, there exist real values $\lambda_1, \ldots, \lambda_{n+1}$ not all zero such that $\lambda_1 x_1 + \ldots + \lambda_{n+1} x_{n+1} = 0$. You may now prove Radon's Theorem however you wish. However, as a suggested first step, prove the following. For any set of $n + 2$ points $x_1, \ldots, x_{n+2}$ in n-dimensional space, there exist $\lambda_1, \ldots, \lambda_{n+2}$ not all zero such that $\sum_i \lambda_i x_i = 0$ and $\sum_i \lambda_i = 0$. (This is called affine dependence.)