Homework 8

>>NAME HERE<<
>>ID HERE<<

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.

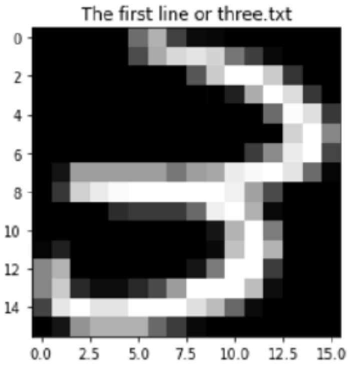


Figure 1: The first image of three.txt

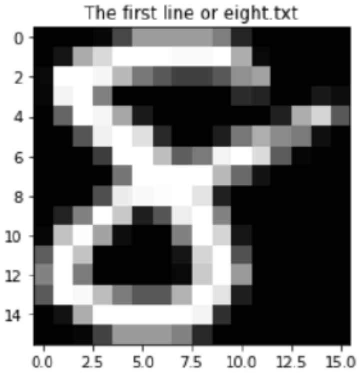


Figure 2: The first image of eight.txt

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.

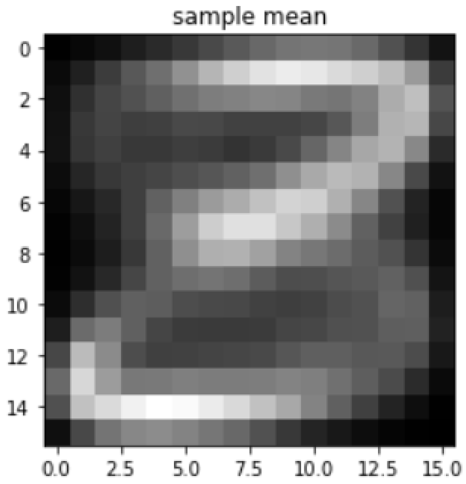


Figure 3: The sample mean

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 $S(1\ldots5, 1\ldots5)$.

$$S(1\ldots5,1\ldots5) = \begin{pmatrix} 59.16729323 & 142.14943609 & 28.68201754 & -7.17857143 & -14.3358396 \\ 142.14943609 & 878.93879073 & 374.13731203 & 24.12778195 & -87.12781955 \\ 28.68201754 & 374.13731203 & 1082.9058584 & 555.2268797 & 33.72431078 \\ -7.17857143 & 24.12778195 & 555.2268797 & 1181.24408521 & 777.77192982 \\ -14.3358396 & -87.12781955 & 33.72431078 & 777.77192982 & 1429.95989975 \end{pmatrix}$$

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.

λ_1 is 2.3716×10^5
λ_2 is 1.4519×10^5

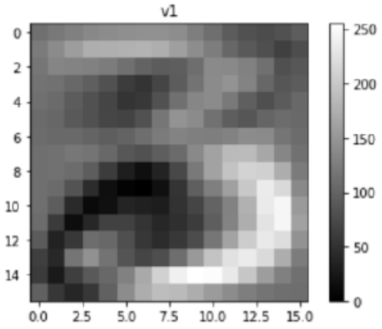


Figure 4: v_1

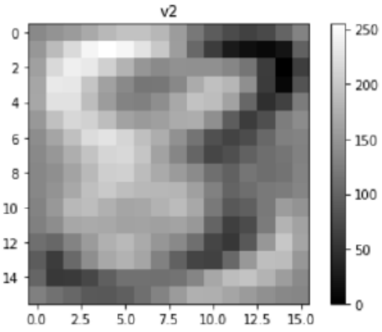


Figure 5: v_2

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.

The resulting coordinates for the first line of three.txt is 1.3233×10^2 and 2.1766×10^2
The resulting coordinates for the first line of eight.txt is 4.2583×10^1 and -1.0712×10^1

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.

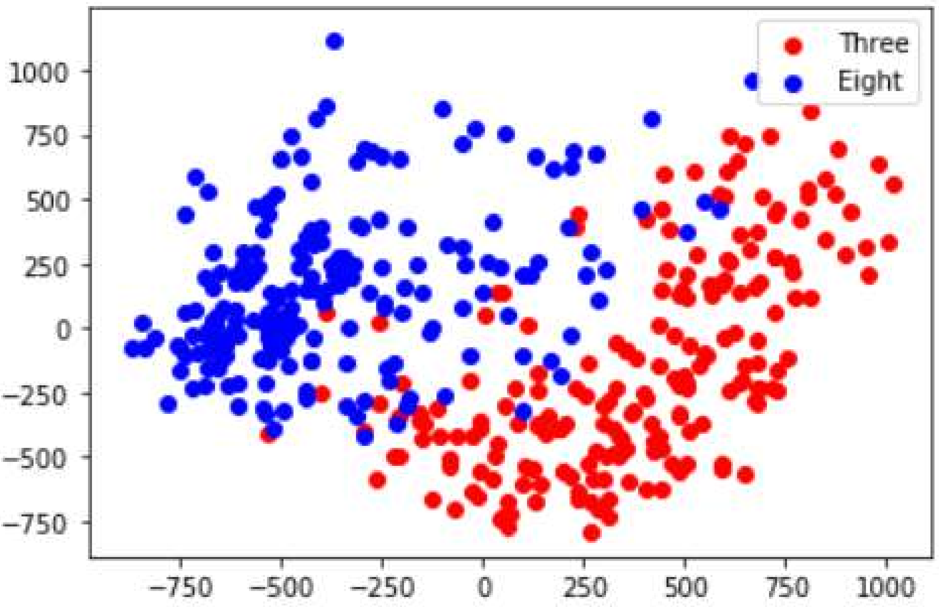
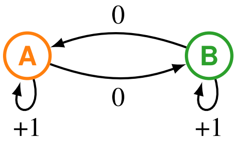


Figure 6: Projected point clouds of ”Three” and ”Eight”

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 $\alpha = 0.5$.

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.

   Q table
   {(1, 1): 9.473172269609766, (1, -1): 8.170509729824643, (-1, 1): 9.124895030917292, (-1, -1): 8.456275464965469}

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
   {(1, 1): 9.758571633656855, (1, -1): 8.87289205865943, (-1, 1): 9.906060535258245, (-1, -1): 8.658865889671446}

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
   {(1, 1): 9.999649473337515, (1, -1): 0, (-1, 1): 0, (-1, -1): 0}

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

   Bellman equation: $E[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} ...]$
   $Q(1,1) = 1 + 1 * 0.9 + 1 * 0.9^2 ... = \frac{1}{0.1} = 10$
   $Q(1,-1) = 1 * 0.9 + 1 * 0.9^2 ... = \frac{0.9}{0.1} = 9$
   $Q(-1,1) = 1 + 1 * 0.9 + 1 * 0.9^2 ... = \frac{0.9}{0.1} = 10$
   $Q(-1,-1) = 1 * 0.9 + 1 * 0.9^2 ... = \frac{0.9}{0.1} = 9$

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.
The VC dimension is 3.
The feature here is x. If we sort the data according to the x, we have 1 or 0 on each different x_i. That means , $ax^2 + bx + c$ could be either positive or negative. So let's analyze what kind of data distribution can be shattered by this F.

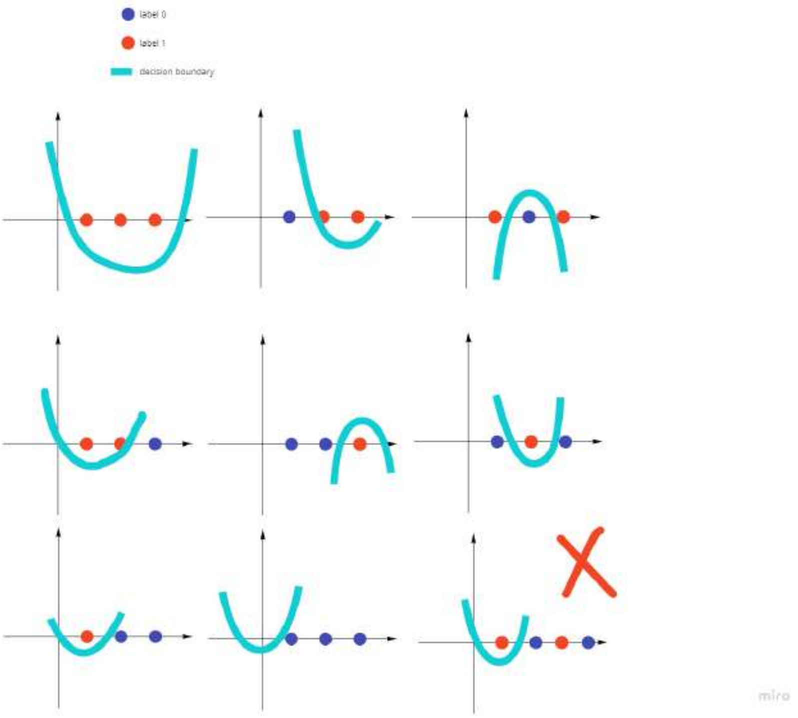


Figure 7: The cases this F can shatter or cannot

From this figure we can see this classifier can shatter three points in all the cases, but it cannot shatter some four points case. Thus, the $VC(F)$ is 3.

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.)

VC dimension is a measure of the maximum number of points that can be shattered in this hypothesis. Assume we have $n+1$ points in a n-dimension space. Each point in this space may be classified into either S_1 or S_2. The VC dimension of this hypothesis is $n+1$, so it can shatter $n+1$ points at most.

So assuming the i_{th} point has a feature $x_i = (x_{i0}, X_{i1}, ...x_{in})$, the output in this hypothesis is sign$(w^\top x + b)$, where $(w^\top x + b) = \sum_{j=0}^{n} w_j x_{ij} + b$. To shatter $n+1$ points, assume the label of x_i is negative while others are positive.

$\sum_{j=0}^{n} w_j(x_{ij} - x_{kj}) < 0, \forall k \neq i$

We have $n \times n$ variables $(x_{ij} - x_{kj})$ and n coefficients (w_j). We can always find a $n \times 1$ vector y where $y_j = w_j(x_{ij} - x_{kj}) < 0$ thus we have n unknown coefficients and n equations, and there is always a solution in this case. Equivalently, we can always separate any point into another set by this hypothesis.

In conclusion, it is always possible to shatter the points set.

(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$(H_n) = n+1$.

Assuming we have $n+2$ points in n-dimension space. Any point in this set can be represented by a linear combination of n of the other $n+1$ points.

Assuming the set S is shattered into S_1 and S_2. The Radon's Theorem indicates that if there are $n+2$ points in n-dimensions, we cannot find a way to partition it into S_1 and S_2 that both their convex hulls are completely on their own sides respectively.

Assuming we have a point $x_p \in S_1$ at this intersection.

$$w^\top x_j + b > 0, \forall x_j \in S_1$$

$$w^\top x_j + b \leq 0, \forall x_j \in S_2$$

Hence we have $x_p = \sum_j \lambda_j x_j, \forall x_j \in S_1$. $w^\top \sum_j \lambda_j x_j + b \leq (\sum_j \lambda_j)(\max(w^\top x_j + b), x_j \in S_2) \leq 0$. But $w^\top x_p + b > 0$, thus contradicted, and this point cannot be shattered correctly. Thus 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.)

In n-dimensional space, we can select n points which are linear independent. Then for the $n+1$ point we have $\sum_{i=1}^{n+1} \lambda_{i,1} x_i = 0$, and for the $n+2$ point we have $\sum_{i=1}^{n} \lambda_{i,2} x_i + \lambda_{n+2,2} = 0$. Sum them up, and we have

$$\sum_{i=1}^{n} (\lambda_{i,1} + \lambda_{i,2}) x_i + \lambda_{n+1,1} x_{n+1} + \lambda_{n+2,2} x_{n+2} = 0$$

If we set $\sum_i \lambda_i = 0$, we have $\lambda_{n+2} = -\sum_i^{n+1} \lambda_i$. Then we have $\sum_{i=1}^{n} (\lambda_{i,1} + \lambda_{i,2}) x_i + \lambda_{n+1,1} x_{n+1} - \sum_i^n (\lambda_{i,1} + \lambda_{i,2}) x_{n+2} - \lambda_{n+1,1} x_{n+2}$. Then this question turns out to be if there exists λ to validate this following equation.

$$\sum_{i=1}^{n} (\lambda_{i,1} + \lambda_{i,2})(x_i - x_{n+2}) + \lambda_{n+1,1}(x_{n+1} - x_{n+2}) = 0$$

$$\sum_{i=1}^{n} (\lambda_{i,1} + \lambda_{i,2})(x_i - x_{n+2}) = -\lambda_{n+1,1}(x_{n+1} - x_{n+2})$$

Now there are totally $n \times n$ matrix whose i_{th} column is exactly $x_i - x_{n+2}$, and also n targets where the j_th one is exactly $-\lambda_{n+1,1}(x_{n+1,j} - x_{n+2,j})$. So for any non-zero $\lambda_{n+1,1}$, we can find a deterministic $\lambda_{i,1} + \lambda_{i,2}$, which can just be notated as λ_i. As $\lambda_{n+1,1}$ is a variable,then there are **infinite** solutions in this case.

Therefore, the first step is proved.
To prove Radon's Theorem, we need to find a point which is in both the convex hull of S_1 and S_2. So, assume we have a point p in the convex hull intersection, so $x_p = \sum_i \lambda_i x_i, \forall x_i \in S_1$, s.t. $\quad \lambda_i \geq 0, \quad \sum_i \lambda_i = 1$.
Let's assume there are k points in S_1, notated as $x_1, x_2, ...X_k$, so we have $x_p = \sum_{i=1}^{k} \lambda_{i,1} x_i$
Because x_p is also in the S_2, so $x_p = \sum_{i=k+1}^{n+2} \lambda_{i,2} x_i$

$$\sum_{i=k+1}^{n+2} \lambda_{i,2} x_i - \sum_{i=1}^{k} \lambda_{i,1} x_i = 0 \tag{1}$$

$$\sum_{i=1}^{k} \lambda_{i,1} = 1, \sum_{i=k+1}^{n+2} \lambda_{i,2} = 1 \tag{2}$$

Recapping the first step in this proof, we see it is always able to find the λ which satisfying $\sum_{i=1}^{n+2} \lambda_i x_i = 0$ and $\sum_{i=1}^{n+2} \lambda_i = 0$. So, we just let $\lambda_i = -\lambda_{i,1}, \forall 1 \leq i \leq k$ and $\lambda_i = \lambda_{i,2}, \forall k \leq i \leq n+2$, and $\sum_{i=1}^{n} +2 = -1 + 1 = 0$.
Thus we can just see the equations above are always solvable. That means we are able to find this point p in the convex hulls of both S_1 and S_2.