Homework 4

1. [25 points] Consider a dataset of n observations $\mathbf{X} \in \mathbb{R}^{n \times d}$, and our goal is to project the data onto a subspace having dimensionality p, $p < d$. Prove that PCA based on projected variance maximization is equivalent to PCA based on projected error (Euclidean error) minimization.

   Solution:
   Suppose $\mathbf{X}$ has been centralized. Let $\mathbf{V} \in \mathbb{R}^{d \times p}$ represent the projected matrix and $\mathbf{V}^T\mathbf{V} = \mathbf{I}$. Then we have two different optimization goals. The one based on projected variance maximization is

   $$\text{maximize} \quad \|\mathbf{X}\mathbf{V}\mathbf{V}^T\|_F^2 = Tr(\mathbf{V}\mathbf{V}^T\mathbf{X}^T\mathbf{X}\mathbf{V}\mathbf{V}^T) = Tr(\mathbf{X}^T\mathbf{X}\mathbf{V}\mathbf{V}^T)$$

   , the other one based on projected error minimization is

   $$\text{minimize} \quad \|\mathbf{X} - \mathbf{X}\mathbf{V}\mathbf{V}^T\|_F^2 = Tr((\mathbf{X} - \mathbf{X}\mathbf{V}\mathbf{V}^T)^T(\mathbf{X} - \mathbf{X}\mathbf{V}\mathbf{V}^T))$$

   And we have

   $$\begin{aligned} Tr((\mathbf{X} - \mathbf{X}\mathbf{V}\mathbf{V}^T)^T(\mathbf{X} - \mathbf{X}\mathbf{V}\mathbf{V}^T)) &= Tr((\mathbf{I} - \mathbf{V}\mathbf{V}^T)^T\mathbf{X}^T\mathbf{X}(\mathbf{I} - \mathbf{V}\mathbf{V}^T)) \\ &= Tr(\mathbf{X}^T\mathbf{X}(\mathbf{I} - \mathbf{V}\mathbf{V}^T)) \\ &= Tr(\mathbf{X}^T\mathbf{X}) - Tr(\mathbf{X}^T\mathbf{X}\mathbf{V}\mathbf{V}^T) \end{aligned}$$

   Since $Tr(\mathbf{X}^T\mathbf{X})$ is a constant value, so projected variance maximization is equivalent to projected error minimization.

2 Maximum Margin Classifier

Consider a data set of n d-dimensional sample points, $\{X_1, \dots, X_n\}$. Each sample point, $X_i \in \mathbb{R}^d$, has a corresponding label, y_i, indicating to which class that point belongs. For now, we will assume that there are only two classes and that every point is either in the given class ($y_i = 1$) or not in the class ($y_i = -1$). Consider the linear decision boundary defined by the hyperplane

$$\mathcal{H} = \{x \in \mathbb{R}^d : x \cdot w + \alpha = 0\}.$$

The *maximum margin classifier* maximizes the distance from the linear decision boundary to the closest training point on either side of the boundary, while correctly classifying all training points.

(a) An in-class sample point is correctly classified if it is on the positive side of the decision boundary, and an out-of-class sample is correctly classified if it is on the negative side. Write a set of n constraints to ensure that all n points are correctly classified.

Solution: We can begin by writing the set of constraints

$$\begin{cases} X_i \cdot w + \alpha \geq 1 & \text{if } y_i = 1 \\ X_i \cdot w + \alpha \leq -1 & \text{if } y_i = -1 \end{cases} \quad \text{for } \ i = 1, \dots, n.$$

Note that we could replace ± 1 in the inequalities above with $\pm c$, where c is any non-negative constant. We can combine these two sets of constraints into the n constraints

$$y_i(X_i \cdot w + \alpha) \geq 1 \text{ for } i = 1, \dots, n.$$

(b) The maximum margin classifier aims to maximize the distance from the training points to the decision boundary. Derive the distance from a point X_i to the hyperplane $\mathcal{H}$.

Solution: Let $\hat{X}_i$ denote the projection of point X_i onto the hyperplane, $\mathcal{H}$. We know that $\hat{X}_i$ must lie on the hyperplane, so $\hat{X}_i \cdot w + \alpha = 0$. We also know that the vector $(X_i - \hat{X}_i)$ must be perpendicular to the hyperplane. Because w is the normal vector for $\mathcal{H}$, $(X_i - \hat{X}_i)$ must lie in the direction of w. Therefore, there exists some scalar $\eta \in \mathbb{R}$ such that $(X_i - \hat{X}_i) = \eta w$. With these two observations, we can determine the value of η as follows.

$$\begin{aligned} (X_i - \hat{X}_i) \cdot w &= (\eta w) \cdot w \\ X_i \cdot w - \hat{X}_i \cdot w &= \eta (w \cdot w) \\ X_i \cdot w + \alpha &= \eta \|w\|^2 \\ \eta &= \frac{X_i \cdot w + \alpha}{\|w\|^2}. \end{aligned}$$

The distance from the point X_i to the hyperplane $\mathcal{H}$ is thus

$$d_i = \|X_i - \hat{X}_i\| = \|\eta w\| = |\eta| \, \|w\| = \left| \frac{X_i \cdot w + \alpha}{\|w\|^2} \right| \|w\| = \frac{|X_i \cdot w + \alpha|}{\|w\|}.$$

(c) Assuming all the points are correctly classified, write an inequality that relates the distance of sample point X_i to the hyperplane $\mathcal{H}$ in terms of only the normal vector w.

Solution: From part (a), we know that if the points are correctly classified,

$$y_i(X_i \cdot w + \alpha) \geq 1 \text{ for } i = 1, \ldots, n.$$

Because y_i is either 1 or -1, these inequalities imply that

$$|X_i \cdot w + \alpha| \geq 1 \text{ for } i = 1, \ldots, n.$$

Therefore, we obtain the following inequality for the distance of X_i to the hyperplane.

$$d_i = \frac{|X_i \cdot w + \alpha|}{\|w\|} \geq \frac{1}{\|w\|}.$$

(d) For the maximum margin classifier, the training points closest to the decision boundary on either side of the boundary are referred to as *support vectors*. What is the distance from any support vector to the decision boundary?

Solution: A support vector X_+ in the given class (i.e. a positive sample) must satisfy

$$X_+ \cdot w + \alpha = 1.$$

A support vector X_- not in the given class (i.e. a negative sample) must satisfy

$$X_- \cdot w + \alpha = -1.$$

Therefore, every support vector X_i must satisfy

$$|X_i \cdot w + \alpha| = 1.$$

Hence the distance from the closest point on either side of the decision boundary is

$$d_i = \frac{|X_i \cdot w + \alpha|}{\|w\|} = \frac{1}{\|w\|}.$$

(e) Using the previous parts, write an optimization problem for the maximum margin classifier.

Solution: The distance of any point to the hyperplane can never be less than $\frac{1}{\|w\|}$. Therefore, to maximize the margin, we want to maximize $\frac{1}{\|w\|}$, which is equivalent to minimizing $\|w\|$. This leads us to the maximum margin classification problem

$$\min_{w,\alpha} \ \|w\| \ \text{ subject to } \ y_i(X_i \cdot w + \alpha) \geq 1, \ \ \forall i \in \{1, \ldots, n\}.$$

We prefer a smooth objective function, and $\|w\|$ is not smooth. (It is pointed at $w = 0$.) So we equivalently express this problem as

$$\min_{w,\alpha} \ \|w\|^2 \ \text{ subject to } \ y_i(X_i \cdot w + \alpha) \geq 1, \ \ \forall i \in \{1, \ldots, n\}.$$

Q3. [16 pts] Performing PCA by Hand

Let's do principal components analysis (PCA)! Consider this sample of six points $X_i \in \mathbb{R}^2$.

$$\left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 2 \end{bmatrix} \right\}.$$

(a) [3 pts] Compute the mean of the sample points and write the centered design matrix $\dot{X}$.

The sample mean is

$$\mu = \frac{1}{6} \sum_{i=1}^{6} X_i = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

By subtracting the mean from each sample, we form the centered design matrix

$$\dot{X} = \begin{bmatrix} -1 & -1 \\ -1 & 0 \\ 0 & -1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}.$$

(b) [6 pts] Find all the principal components of this sample. Write them as **unit** vectors.

The principal components of our dataset are the eigenvectors of the matrix

$$\dot{X}^\top \dot{X} = \begin{bmatrix} 4 & 2 \\ 2 & 4 \end{bmatrix}.$$

The characteristic polynomial of this symmetric matrix is

$$\begin{aligned} \det(sI - X^\top X) = \det \begin{bmatrix} s-4 & -2 \\ -2 & s-4 \end{bmatrix} &= (s-4)(s-4) - (-2)(-2) \\ &= s^2 - 8s + 12 = (s-6)(s-2). \end{aligned}$$

Hence the eigenvalues of $\dot{X}^\top \dot{X}$ are $\lambda_1 = 2$ and $\lambda_2 = 6$. With these eigenvalues, we can compute the eigenvectors of this matrix as follows. (Or you could just guess them and verify them.)

$$\begin{bmatrix} \lambda_1 - 4 & -2 \\ -2 & \lambda_1 - 4 \end{bmatrix} v_1 = \begin{bmatrix} -2 & -2 \\ -2 & -2 \end{bmatrix} v_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow v_1 = \begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix}$$

$$\begin{bmatrix} \lambda_2 - 4 & -2 \\ -2 & \lambda_2 - 4 \end{bmatrix} v_2 = \begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix} v_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow v_2 = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$$

(c) [4 pts]

- Which of those two principal components would be preferred if you use only one? [1 pt]
- What information does the PCA algorithm use to decide that one principal components is better than another? [1 pt]
- From an optimization point of view, why do we prefer that one? [2 pts]

We choose $v_2 = [1/\sqrt{2} \quad 1/\sqrt{2}]^\top$ first.

PCA picks the principal component with the largest eigenvalue.

We prefer it because it maximizes the variance of the sample points after they are projected onto a line parallel to v2.

Q4. Backpropagation on an Arithmetic Expression

Consider an arithmetic network with the inputs a, b, and c, which computes the following sequence of operations, where $s(\gamma) = \dfrac{1}{1 + e^{-\gamma}}$ is the logistic (sigmoid) function and $r(\gamma) = \max\{0, \gamma\}$ is the hinge function used by ReLUs.

$$d = ab \quad e = s(d) \quad f = r(a) \quad g = 3a \quad h = 2e + f + g \quad i = ch \quad j = f + i^2$$

We want to find the partial derivatives of j with respect to every other variable a through i, in backpropagation style. This means that for each variable z, we want you to write $\partial j / \partial z$ in two forms: (1) in terms of derivatives involving each variable that *directly* uses the value of z, and (2) in terms of the inputs and intermediate values $a \ldots i$, as simply as possible but with no derivative symbols. For example, we write

$$\frac{\partial j}{\partial i} = \frac{\mathrm{d} j}{\mathrm{d} i} = 2i \quad \text{(no chain rule needed for this one only)}$$

$$\frac{\partial j}{\partial h} = \frac{\partial j}{\partial i}\frac{\partial i}{\partial h} = 2ic \quad \text{(chain rule, then backprop the derivative expressions)}$$

(a) [15 pts] Now, please write expressions for $\partial j/\partial g$, $\partial j/\partial f$, $\partial j/\partial e$, $\partial j/\partial d$, $\partial j/\partial c$, $\partial j/\partial b$, and $\partial j/\partial a$ as we have written $\partial j/\partial h$ above. If they are needed, express the derivative $s'(\gamma)$ in terms of $s(\gamma)$ and express the derivative $r'(\gamma)$ as the indicator function $\mathbf{1}(\gamma \geq 0)$. (*Hint: f is used in two places and a is used in three, so they will need a multivariate chain rule. It might help you to draw the network as a directed graph, but it's not required.*)

$$\frac{\partial j}{\partial g} = \frac{\partial j}{\partial h}\frac{\partial h}{\partial g} = 2ic$$

$$\frac{\partial j}{\partial f} = \frac{\partial j}{\partial h}\frac{\partial h}{\partial f} + \frac{\mathrm{d} j}{\mathrm{d} f} = 2ic + 1$$

$$\frac{\partial j}{\partial e} = \frac{\partial j}{\partial h}\frac{\partial h}{\partial e} = 4ic$$

$$\frac{\partial j}{\partial d} = \frac{\partial j}{\partial e}\frac{\partial e}{\partial d} = 4ic\, s(d)(1 - s(d))$$

$$\frac{\partial j}{\partial c} = \frac{\partial j}{\partial i}\frac{\partial i}{\partial c} = 2ih$$

$$\frac{\partial j}{\partial b} = \frac{\partial j}{\partial d}\frac{\partial d}{\partial b} = 4ica\, s(d)(1 - s(d))$$

$$\frac{\partial j}{\partial a} = \frac{\partial j}{\partial d}\frac{\partial d}{\partial a} + \frac{\partial j}{\partial f}\frac{\partial f}{\partial a} + \frac{\partial j}{\partial g}\frac{\partial g}{\partial a} = 4icb\, s(d)(1 - s(d)) + (2ic + 1) \cdot \mathbf{1}(a \geq 0) + 6ic$$