HOMEWORK 5

Ryan Yee
9074025223

Instructions: Use this latex file as a template to develop your homework. We are changing our reproducibility policy on code submissions going forward. **Instead of uploading it on GitHub, please submit a separate zip file that contains your code. You will submit two files to Canvas, one is your pdf, and the other one is a zip file.** Late submissions may not be accepted. You can choose any programming language (i.e. python, R, or MATLAB). Please check Piazza for updates about the homework.

This homework is more difficult than previous homework. The total amount of points for this homework is **150**. The extra credit reflects the level of difficulty.

1 Clustering

1.1 K-means Clustering (14 points)

1. **(6 Points)** Given n observations $X_1^n = \{X_1, \ldots, X_n\}$, $X_i \in \mathcal{X}$, the K-means objective is to find k ($< n$) centres $\mu_1^k = \{\mu_1, \ldots, \mu_k\}$, and a rule $f : \mathcal{X} \to \{1, \ldots, K\}$ so as to minimize the objective

$$J(\mu_1^K, f; X_1^n) = \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{1}(f(X_i) = k) \|X_i - \mu_k\|^2 \tag{1}$$

   Let $\mathcal{J}_K(X_1^n) = \min_{\mu_1^k, f} J(\mu_1^K, f; X_1^n)$. Prove that $\mathcal{J}_K(X_1^n)$ is a non-increasing function of K.

   Prove via induction. Suppose $K = 1$. Now add one additional center so $K = 2$. If $\mu_2 = \mu_1$ then every X_i is equal and $\mathcal{J}_1(X_1^n) = \mathcal{J}_2(X_1^n)$. Otherwise, if $\mu_2 \neq \mu_1$ then at least one $X_i \in \mathcal{X}$ is closer to μ_2 than μ_1, thus $\mathcal{J}_1(X_1^n) > \mathcal{J}_2(X_1^n)$. So $\mathcal{J}_K(X_1^n)$ is a non-increasing function of K when $K = 1$. Now assume $\mathcal{J}_K(X_1^n)$ is a non-increasing function of K for $K = 1, 2, \ldots, k$. Now consider $K = k + 1$. If $\exists i$ s.t. $\mu_i = \mu_{k+1}$, $i = 1, \ldots, k$, then every X_i is equal to some μ_i and $\mathcal{J}_k(X_1^n) = \mathcal{J}_{k+1}(X_1^n)$. Otherwise, if $\mu_{k+1} \neq \mu_i \forall i = 1, \ldots, k$ then at least one $X_i \in \mathcal{X}$ is closer to μ_{k+1} than any other μ_i, thus $\mathcal{J}_k(X_1^n) > \mathcal{J}_{k+1}(X_1^n)$. So $\mathcal{J}_K(X_1^n)$ is a non-increasing function of K for any K.

2. **(8 Points)** Consider the K-means (Lloyd's) clustering algorithm we studied in class. We terminate the algorithm when there are no changes to the objective. Show that the algorithm terminates in a finite number of steps.

   First, prove the objective function in Lloyd's algorithm is strictly decreasing. Indeed, much like the EM algorithm, at each iteration Lloyd's algorithm finds the expected group of each observation and then maximizes the likelihood by moving the center to the optimal point among the points in its custer. This maximization step guarantees the objective function will decrease in the next iteration of the algorithm will terminate. Next, prove the Lloyd's algorithm will either terminate after finitely many steps or loop indefinitely. There are a finite number of data points, therefore, there are a finite number of ways observations can be partitioned. It follows that Lloyd's algorithm much terminate after finitely many steps or loop indefinitely. But since the objective function is strictly decreasing, the algorithm cannot loop indefinitely. Indeed, say at step t the value of the objective function is k_t. Then, $k_{t+q} < k_t$ for $q = 1, 2, \ldots$. But if the algorithm loops after q iterations, then $k_t = k_{t+q}$ which is a contradiction. So Lloyd's algorithm cannot loop indefinitely, so it must terminate after finitely many iterations.

1.2 Experiment (20 Points)

In this question, we will evaluate K-means clustering and GMM on a simple 2 dimensional problem. First, create a two-dimensional synthetic dataset of 300 points by sampling 100 points each from the three Gaussian distributions

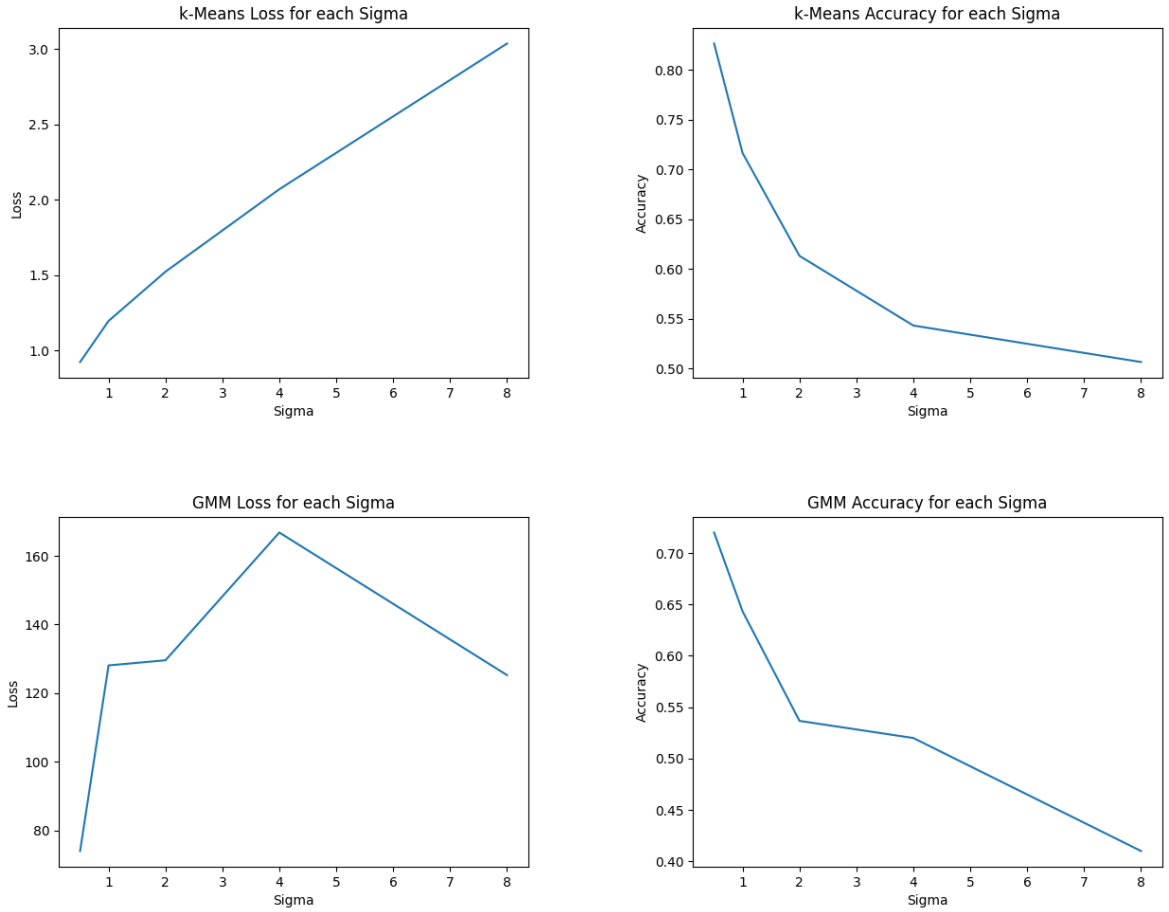
shown below:

$$P_a = \mathcal{N}\left(\begin{bmatrix}-1\\-1\end{bmatrix}, \sigma\begin{bmatrix}2, & 0.5\\0.5, & 1\end{bmatrix}\right), \quad P_b = \mathcal{N}\left(\begin{bmatrix}1\\-1\end{bmatrix}, \sigma\begin{bmatrix}1, & -0.5\\-0.5, & 2\end{bmatrix}\right), \quad P_c = \mathcal{N}\left(\begin{bmatrix}0\\1\end{bmatrix}, \sigma\begin{bmatrix}1 & 0\\0, & 2\end{bmatrix}\right)$$

Here, σ is a parameter we will change to produce different datasets.

- First implement K-means clustering and the expectation maximization algorithm for GMMs. Execute both methods on five synthetic datasets, generated as shown above with $\sigma \in \{0.5, 1, 2, 4, 8\}$. Finally, evaluate both methods on *(i)* the clustering objective (1) and *(ii)* the clustering accuracy. For each of the two criteria, plot the value achieved by each method against σ.
- Both algorithms are only guaranteed to find only a local optimum so we recommend trying multiple restarts and picking the one with the lowest objective value (This is (1) for K-means and the negative log likelihood for GMMs). You may also experiment with a smart initialization strategy (such as kmeans++).
- To plot the clustering accuracy, you may treat the 'label' of points generated from distribution P_u as u, where $u \in \{a, b, c\}$. Assume that the cluster id i returned by a method is $i \in \{1, 2, 3\}$. Since clustering is an unsupervised learning problem, you should obtain the best possible mapping from $\{1, 2, 3\}$ to $\{a, b, c\}$ to compute the clustering objective. One way to do this is to compare the clustering centers returned by the method (centroids for K-means, means for GMMs) and map them to the distribution with the closest mean.

Points break down: 7 points each for implementation of each method, 6 points for reporting of evaluation metrics.



2 Linear Dimensionality Reduction

2.1 Principal Components Analysis (10 points)

Principal Components Analysis (PCA) is a popular method for linear dimensionality reduction. PCA attempts to find a lower dimensional subspace such that when you project the data onto the subspace as much of the

information is preserved. Say we have data $X = [x_1^\top; \ldots; x_n^\top] \in \mathbb{R}^{n \times D}$ where $x_i \in \mathbb{R}^D$. We wish to find a d ($< D$) dimensional subspace $A = [a_1, \ldots, a_d] \in \mathbb{R}^{D \times d}$, such that $a_i \in \mathbb{R}^D$ and $A^\top A = I_d$, so as to maximize $\frac{1}{n} \sum_{i=1}^n \|A^\top x_i\|^2$.

1. **(4 Points)** Suppose we wish to find the first direction a_1 (such that $a_1^\top a_1 = 1$) to maximize $\frac{1}{n} \sum_i (a_1^\top X_i)^2$. Show that a_1 is the first right singular vector of X.

   Let a_1 be the first right singular vector of X. Then $X^T X a_1 = \lambda_1 a_1$ where λ_1 is an eigenvalue of $X^T X$ and $\lambda_1 \geq \lambda_i$ for $i = 2, 3, \ldots, D$ by definition. Then:

   $$X^T X a_1 = \lambda_1 a_1$$
   $$a_1^T X^T X a_1 = \lambda_1 a_1^T a_1$$
   $$\sum_i (a_1^\top X_i)^2 = \lambda_1$$

   Since λ_1 is the largest eigenvalue of $X^T X$, the function is maximized if a_1 is a right singular vector of X. a_1 must be a right singular vector of X for the equation to hold. Thus, the function is maximized when a_1 is the first right singular vector of X.

2. **(6 Points)** Given $a_1, \ldots, a_k$, let $A_k = [a_1, \ldots, a_k]$ and $\tilde{x}_i = x_i - AA^\top x_i$. We wish to find a_{k+1}, to maximize $\frac{1}{n} \sum_i (a_{k+1}^\top \tilde{x}_i)^2$. Show that a_{k+1} is the $(k+1)^{th}$ right singular vector of X.

   In matrix form, we have $\tilde{X} = X - XAA^T$ since AA^T is symmetric. Then:

   $$\begin{aligned} \tilde{X}^T \tilde{X} &= (X - XAA^T)^T (X - XAA^T) \\ &= X^T X - X^T XAA^T - AA^T X^T X + AA^T X^T XAA^T \end{aligned}$$

   So:

   $$\begin{aligned} \tilde{X}^T \tilde{X} A &= (X^T X - X^T XAA^T - AA^T X^T X + AA^T X^T XAA^T) A \\ &= X^T XA - X^T XA - AA^T X^T XA + AA^T X^T XA \\ &= \mathbf{0} \end{aligned}$$

   So $a_1, \ldots, a_k$ do not maximize $\sum_i (a_{k+1}^\top \tilde{x}_i)^2 = a_{k+1}^T \tilde{X}^T \tilde{X} a_{k+1}$. Now consider a^* which is any singular vector of X that is not in A. Then:

   $$\begin{aligned} \tilde{X}^T \tilde{X} a^* &= (X^T X - X^T XAA^T - AA^T X^T X + AA^T X^T XAA^T) a^* \\ &= X^T X a^* \\ &= \lambda^* a^* \end{aligned}$$

   since a^* is orthogonal to all the vectors in A. Therefore, a_{k+1} is the first right singular vector of $\tilde{X}$. So by the arguments in (1), the function is maximized when a_{k+1} is the $(k+1)^{th}$ singular vector of X.

2.2 Dimensionality reduction via optimization (22 points)

We will now motivate the dimensionality reduction problem from a slightly different perspective. The resulting algorithm has many similarities to PCA. We will refer to method as DRO.
As before, you are given data $\{x_i\}_{i=1}^n$, where $x_i \in \mathbb{R}^D$. Let $X = [x_1^\top; \ldots x_n^\top] \in \mathbb{R}^{n \times D}$. We suspect that the data actually lies approximately in a d dimensional affine subspace. Here $d < D$ and $d < n$. Our goal, as in PCA, is to use this dataset to find a d dimensional representation z for each $x \in \mathbb{R}^D$. (We will assume that the span of the data has dimension larger than d, but our method should work whether $n > D$ or $n < D$.)
Let $z_i \in \mathbb{R}^d$ be the lower dimensional representation for x_i and let $Z = [z_1^\top; \ldots; z_n^\top] \in \mathbb{R}^{n \times d}$. We wish to find parameters $A \in \mathbb{R}^{D \times d}$, $b \in \mathbb{R}^D$ and the lower dimensional representation $Z \in \mathbb{R}^{n \times d}$ so as to minimize

$$J(A, b, Z) = \frac{1}{n} \sum_{i=1}^n \|x_i - Az_i - b\|^2 = \|X - ZA^\top - \mathbf{1}b^\top\|_F^2. \tag{2}$$

Here, $\|A\|_F^2 = \sum_{i,j} A_{ij}^2$ is the Frobenius norm of a matrix.

1. **(3 Points)** Let $M \in \mathbb{R}^{d \times d}$ be an arbitrary invertible matrix and $p \in \mathbb{R}^d$ be an arbitrary vector. Denote, $A_2 = A_1 M^{-1}$, $b_2 = b_1 - A_1 M^{-1} p$ and $Z_2 = Z_1 M^\top + \mathbf{1}p^\top$. Show that both (A_1, b_1, Z_1) and (A_2, b_2, Z_2) achieve the same objective value J (2).

$$\begin{aligned} J &= \|X - Z_2 A_2^\top - \mathbf{1} b_2^\top\| \\ &= \|X - (Z_1 M^\top + \mathbf{1} p^\top)(A_1 M^{-1})^\top - \mathbf{1}(b_1 - A_1 M^{-1} p)^\top\| \\ &= \|X - Z_1 M^\top M^{-\top} A_1^\top - \mathbf{1} p^\top M^{-\top} A_1^\top - \mathbf{1} b_1^\top + \mathbf{1} p^\top M^{-\top} A_1^\top\| \\ &= \|X - Z_1 A_1^\top - \mathbf{1} b_1^\top\| \end{aligned}$$

Therefore, in order to make the problem determined, we need to impose some constraint on Z. We will assume that the z_i's have zero mean and identity covariance. That is,

$$\bar{Z} = \frac{1}{n}\sum_{i=1}^{n} z_i = \frac{1}{n} Z^\top \mathbf{1}_d = 0, \qquad S = \frac{1}{n}\sum_{i=1}^{n} z_i z_i^\top = \frac{1}{n} Z Z^\top = I_d$$

Here, $\mathbf{1}_d = [1, 1 \ldots, 1]^\top \in \mathbb{R}^d$ and I_d is the $d \times d$ identity matrix.

2. **(16 Points)** Outline a procedure to solve the above problem. Specify how you would obtain A, Z, b which minimize the objective and satisfy the constraints.

   Hint: The rank k approximation of a matrix in Frobenius norm is obtained by taking its SVD and then zeroing out all but the first k singular values.

   Note that in the best case $X - ZA^\top - \mathbf{1}b^\top = 0$. First, solve for b:

   $$\begin{aligned} X - ZA^\top - \mathbf{1}b^\top &= 0 \\ X - ZA^\top &= \mathbf{1}b^\top \\ X^\top - AZ^\top &= b\mathbf{1}^\top \\ X^\top \mathbf{1} - AZ^\top \mathbf{1} &= bn \\ \frac{1}{n} X^\top \mathbf{1} &= b \\ \frac{1}{n}\sum_{i=1}^{n} x_i &= b \end{aligned}$$

   Next, find $Q = ZA^\top$:

   $$\begin{aligned} X - Q - \mathbf{1}b^\top &= 0 \\ Q &= X - \mathbf{1}b^\top \end{aligned}$$

   Note that the rows of Q have mean zero. Using the *hint* from above, the rank k approximation of Q is its SVD with all but the first k singular values singled out. So, let $A_k = [a_1, \ldots, a_k]$ where a_i is the ith right singular vector of Q.

3. **(3 Points)** You are given a point x_* in the original D dimensional space. State the rule to obtain the d dimensional representation z_* for this new point. (If x_* is some original point x_i from the D–dimensional space, it shoud be the d–dimensional representation z_i.)

   $z_i = A^\top (x_i - b)$

2.3 Dimensionality reduction via a generative model (42 points)

We will now study dimensionality reduction via a generative model. We will refer to method as DRLV. We will assume a $d(< n)$ dimensional latent space and the following generative process for the data.

$$\begin{aligned} z &\sim \mathcal{N}(\mathbf{0}, I), \quad z \in \mathbb{R}^d \\ x|z &\sim \mathcal{N}(Az + b, \eta^2 I), \quad x \in \mathbb{R}^D \end{aligned}$$

The model says that we first sample a d dimensional Gaussian with zero mean and identity variance. Then we map it to D dimensions by computing $Az + b$. Finally, we add some spherical Gaussian noise with variance η^2 on each dimension.

We will use an expectation maximization (EM) procedure to learn the parameters A, b, η. So far we have only studied EM with discrete latent variables. In this problem, we will look at EM with a continuous latent variable which has a parametric distribution. The following results will be useful.

Fact 1 (Conditional of a Gaussian). *Say* $(Y_1, Y_2), Y_i \in \mathbb{R}^{d_i}$ *is Gaussian distribued.*

$$\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} = \mathcal{N}\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{12}^\top & \Sigma_{22} \end{bmatrix} \right)$$

Then, conditioned on $Y_1 = y_1$ *the distribution for* Y_2 *is*

$$Y_2|Y_1 = y_1 \sim \mathcal{N}(\mu_2 + \Sigma_{12}^\top \Sigma_{11}^{-1}(y_1 - \mu_1),\ \Sigma_{22} - \Sigma_{12}^\top \Sigma_{11}^{-1} \Sigma_{12})$$

Fact 2 (Some Matrix Derivatives). *Let* $X \in \mathbb{R}^{r \times c}$, *and* $u \in \mathbb{R}^r$, $v, w \in \mathbb{R}^c$.

$$\nabla_X v^\top X^\top u = uv^\top$$
$$\nabla_X v^\top X^\top X w = X(vw^\top + wv^\top)$$

1. **(10 Points)** Assuming some given values for A, b, and η^2, write down the joint distribution of (z, x). Use this to derive the marginal distribution of x and the conditional distribution $z|x$.

$$\begin{aligned}
\mathbb{P}(x, z) &= \mathbb{P}(x|z)\mathbb{P}(z) \\
&= (2\pi)^{-D/2} \det(\eta^2 I)^{-1/2} \exp\left\{ -\frac{1}{2}(x - Az - b)^\top (\eta^2 I)^{-1}(x - Az - b) \right\} (2\pi)^{-d/2} \exp\left\{ -\frac{1}{2} z^\top z \right\} \\
&= (2\pi)^{-\frac{D+d}{2}} \det(\eta^2 I)^{-1/2} \exp\left\{ -\frac{1}{2} \left((x - Az - b)^\top (\eta^2 I)^{-1}(x - Az - b) + \eta^2 z^\top z \right) \right\}
\end{aligned}$$

We can assume x follows a Gaussian distribution. It is straightforward to see $\mathbb{E}[X] = b$. Then

$$\begin{aligned}
\text{Var}(X) &= \mathbb{E}[(X - b)^\top (X - b)] \\
&= \mathbb{E}[(AZ + \epsilon)^\top (AZ + \epsilon)] \\
&= \mathbb{E}[Z^\top A^\top AZ + 2Z^\top A^\top \epsilon + \epsilon^\top \epsilon] \\
&= A^\top A + \eta^2 I
\end{aligned}$$

So, $x \sim \mathcal{N}(b, A^\top A + \eta^2 I)$. Then:

$$\begin{aligned}
Cov(x, z) &= \mathbb{E}[X^\top Z] \\
&= \mathbb{E}[(AZ + b + \epsilon)^\top Z] \\
&= \mathbb{E}[Z^\top A^\top Z + b^\top Z + \epsilon^\top Z] \\
&= A^\top
\end{aligned}$$

Using **Fact 1** we have $z|x \sim \mathcal{N}(A(A^\top A + \eta^2 I)^{-1}(x - b), I - A(A^\top A + \eta^2 I)^{-1} A^\top)$

2. **(4 points)** Write the log likelihood in terms of parameters A, b, and η^2.

3. **(4 Points)** First obtain the Maximum Likelihood Estimate for b. This does not require EM.

4. **(10 Points)** To apply the EM algorithm, let $Q(z_i)$ denote some distribution over z_i for each z_i. Obtain a lower bound on the log likelihood via Jensen's inequality.

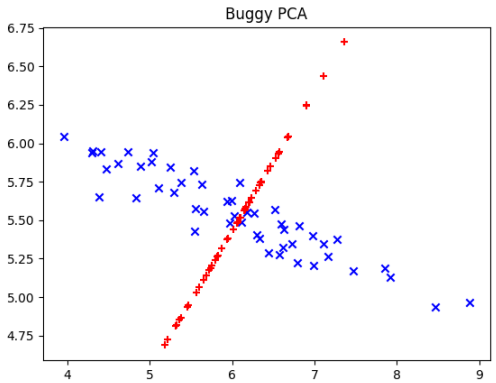
5. **(4 Points)** Recall, from the lectures, that we chose $Q(z_i) = \mathbb{P}(z_i|x_i)$ in the E-step to obtain the tightest possible lower bound for the log likelihood. Here, $\mathbb{P}(z_i|x_i)$ is the conditional distribution of z_i given x_i under the current estimates for A, b, and η. Write down the E-step update for the next iteration.
N.B: Unlike in GMMs, where the latent variable was discrete, here the latent variable is continuous. Fortunately, it has a parametric form we can represent $Q(z_i)$ using a finite number of parameters.
(Hint: See part 1)

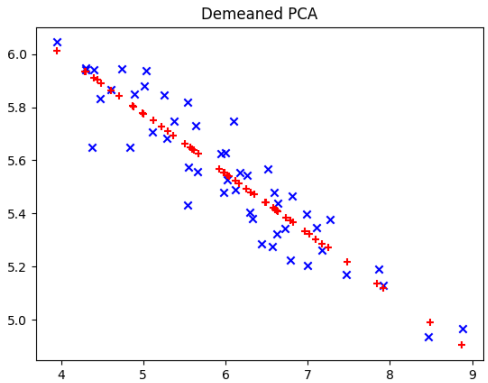
6. **(10 Points)** Now write down the M-step update for parameters A and η, obtained by maximizing the lower bound obtained from parts 3 and 4.

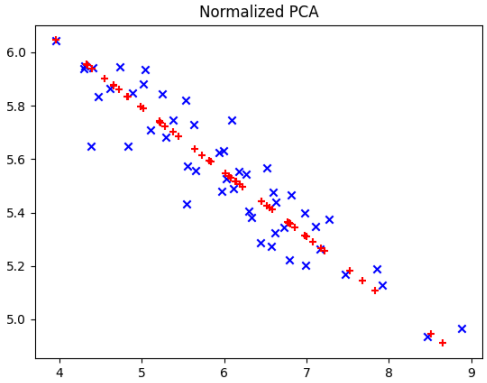
2.4 Experiment (42 points)

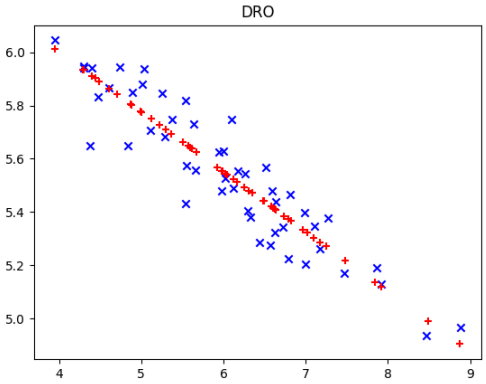
Here we will compare the above three methods on two data sets.

- We will implement three variants of PCA:
  1. "buggy PCA": PCA applied directly on the matrix X.
  2. "demeaned PCA": We subtract the mean along each dimension before applying PCA.
  3. "normalized PCA": Before applying PCA, we subtract the mean and scale each dimension so that the sample mean and standard deviation along each dimension is 0 and 1 respectively.
- One way to study how well the low dimensional representation Z captures the linear structure in our data is to project Z back to D dimensions and look at the reconstruction error. For PCA, if we mapped it to d dimensions via $z = Vx$ then the reconstruction is $V^\top z$. For the preprocessed versions, we first do this and then reverse the preprocessing steps as well. For DRO we just compute $Az + b$. For DRLV, we will use the posterior mean $\mathbb{E}[z|x]$ as the lower dimensional representation and $Az + b$ as the reconstruction.
  We will compare all four methods by the reconstruction error on the datasets.
- Please implement code for the five methods: Buggy PCA (just take the SVD of X) , Demeaned PCA, Normalized PCA, DRO, DRLV. In all cases your function should take in an $n \times d$ data matrix and d as an argument. It should return the the d dimensional representations, the estimated parameters, and the reconstructions of these representations in D dimensions. For DRLV, use the values obtained from DRO as initializations for A. Set η based on the reconstruction errors of DRO. Use 10 iterations of EM.
- You are given two datasets: A two Dimensional dataset with 50 points `data2D.csv` and a thousand dimensional dataset with 500 points `data1000D.csv`.
- For the $2D$ dataset use $d = 1$. For the $1000D$ dataset, you need to choose d. For this, observe the singular values in DRO and see if there is a clear "knee point" in the spectrum. Attach any figures/ Statistics you computed to justify your choice.
- For the $2D$ dataset you need to attach the a plot comparing the orignal points with the reconstructed points for all five methods. For both datasets you should also report the reconstruction errors, that is the squared sum of differences $\sum_{i=1}^{n} \|x_i - r(z_i)\|^2$, where x_i's are the original points and $r(z_i)$ are the D dimensional points reconstructed from the d dimensional representation z_i.



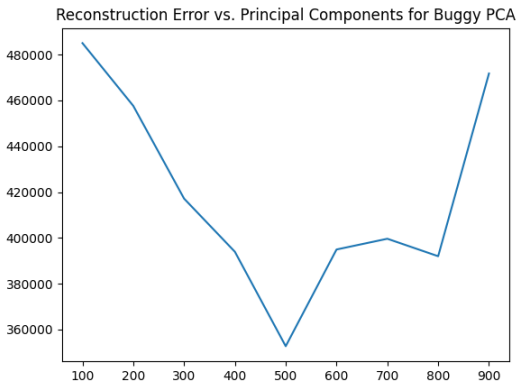


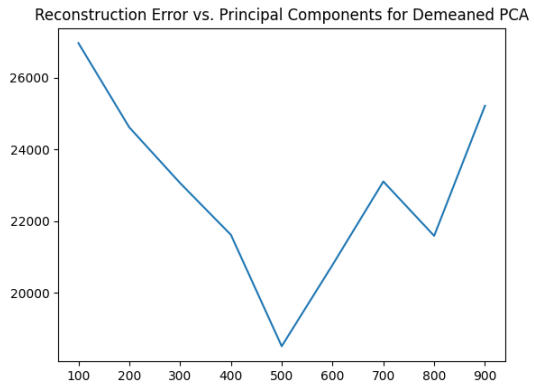


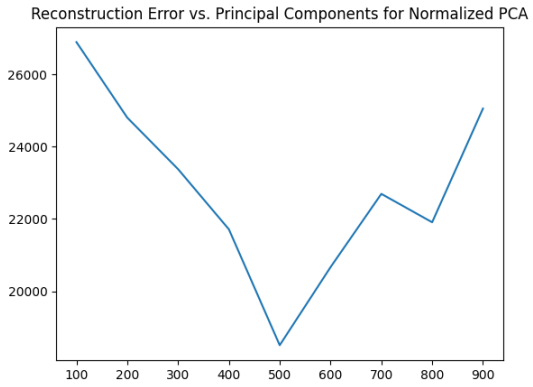


2D Data

Method	Reconstruction Error
Buggy PCA	0.886903
Demeaned PCA	0.010006
Normalized PCA	0.049472
DRO	0.010006
DRLV	







The "knee-point" seems to be at $d = 500$ for all the PCA methods. This makes sense, because there are 500 observations in the 1000D data ($n = 500$), which means its SVD has 500 singular values. I will use $d = 500$ for all the methods.

1000D Data

Method	Reconstruction Error
Buggy PCA	352741
Demeaned PCA	18509
Normalized PCA	18503
DRO	18509
DRLV	

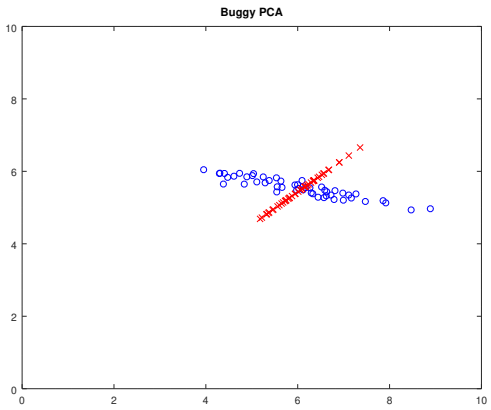
- **Questions:** After you have completed the experiments, please answer the following questions.

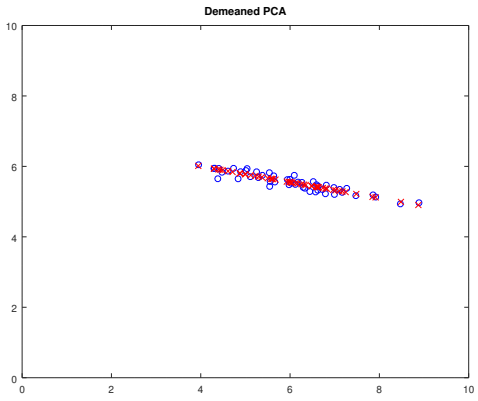
  1. Look at the results for Buggy PCA. The reconstruction error is bad and the reconstructed points don't seem to well represent the original points. Why is this?
     Hint: Which subspace is Buggy PCA trying to project the points onto?
     Buggy PCA is trying to project onto a subspace that passes through the origin. In the other PCA methods, we account for this by subtracting the mean from each dimension so the mean of each dimension is zero.
  2. The error criterion we are using is the average squared error between the original points and the reconstructed points. In both examples DRO and demeaned PCA achieves the lowest error among all methods. Is this surprising? Why?
     This is not surprising. Demeaned PCA is equivalent to DRO. In problem 2.2, we derived DRO such that the reconstruction error was minimized.

- Point allocation:
  - Implementation of the three PCA methods: **(10 Points)**
  - Implementation of DRO and DRLV: **(20 points)**
  - Implementing reconstructions and reporting results: **(5 points)**
  - Choice of d for $1000D$ dataset and appropriate justification: **(3 Points)**
  - Questions **(4 Points)**

Partial answers: These were our errors on all methods for the $2D$ dataset and the reconstructions obtained for Buggy PCA and Demeaned PCA. We have provided them to cross-check with your solution. Our implementation may have bugs so if your answer does not tally, first double check with your peers and then speak to the TA/Instrutor.

```
Reconstruction Errors:
Buggy PCA: 0.886903
Demeaned PCA: 0.010006
Normalized PCA: 0.049472
DRO: 0.010006
DRLV: 0.010081
```





The blue circles are from the original dataset and the red crosses are the reconstructed points.