Homework 5

Yuhan Wang
ywang2558

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.

Github link: https://github.com/hanhuangv587/CS760hw5

1 Clustering

1.1 K-means Clustering (14 points)

1. (6 Points) Given n observations $X_1^n = \{X_1, \dots, X_n\}$, $X_i \in \mathcal{X}$, the K-means objective is to find k $(< n)$ centres $\mu_1^k = \{\mu_1, \dots, \mu_k\}$, and a rule $f : \mathcal{X} \to \{1, \dots, K\}$ 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 $\mathcal{J}_K(X_1^n) = \min_{\mu^K, f} J(\mu_1^K, f; X_1^n)$. Prove that $\mathcal{J}_K(X_1^n)$ is a non-increasing function of K.

We can easily see that $\mathcal{J}_K(X_1^n)$ is a non-increasing function of K by induction. For the base case, we have $\mathcal{J}_1(X_1^n) = \sum_{i=1}^n \|X_i - \mu_1\|^2$. For the inductive step, we have $\mathcal{J}_{K+1}(X_1^n) = \min_{\mu_1^{K+1}, f} J(\mu_1^{K+1}, f; X_1^n)$ $= \min_{\mu_1^{K+1}, f} \sum_{i=1}^n \sum_{k=1}^{K+1} \mathbb{1}(f(X_i) = k)\|X_i - \mu_k\|^2 \leq \min_{\mu_1^K, f} \sum_{i=1}^n \sum_{k=1}^{K} \mathbb{1}(f(X_i) = k)\|X_i - \mu_k\|^2 = \min_{\mu_1^K, f} J(\mu_1^K, f; X_1^n) = \mathcal{J}_K(X_1^n)$. Therefore, $\mathcal{J}_K(X_1^n)$ is a non-increasing function of K. The less than or equal is hold since we can always set $\mu_1^{K+1} = \{\mu_1^K, 0\}$ and keep the same f to achieve $\mathcal{J}_K(X_1^n)$. And by optimizing μ_{K+1}, we can achieve $\mathcal{J}_{K+1}(X_1^n)$ less than or equal to $\mathcal{J}_K(X_1^n)$.

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 notice that for a given number of K, the possible choices of μ^k is finite. For each dimension of μ^k, there are at most $\sum_{i=1}^n C_n^i = 2^n$ possible centers. So the possible value of (1) is finite for a given dataset.
And from the t step to the $t+1$ step, the objective function is always decreasing unless it reach the optimal(since both the center calculation and regroup will result a decreasing in objective function unless no change). Therefore, an algorithm optimize a finite valued function and grantee to decrease in each step until stop will terminate in a finite number of steps.

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.

  See Fig1.png and Fig2.png below. For code and data, please see hw5q1.ipynb.

Sigma	Accuracy	Objective
0.5	0.80	315
1	0.70	514
2	0.63	868
4	0.53	1623
8	0.50	3188

Table 1: K-means results

Sigma	Accuracy	Objective
0.5	0.65	550
1	0.56	891
2	0.50	1441
4	0.44	2950
8	0.40	6081

Table 2: EM results

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

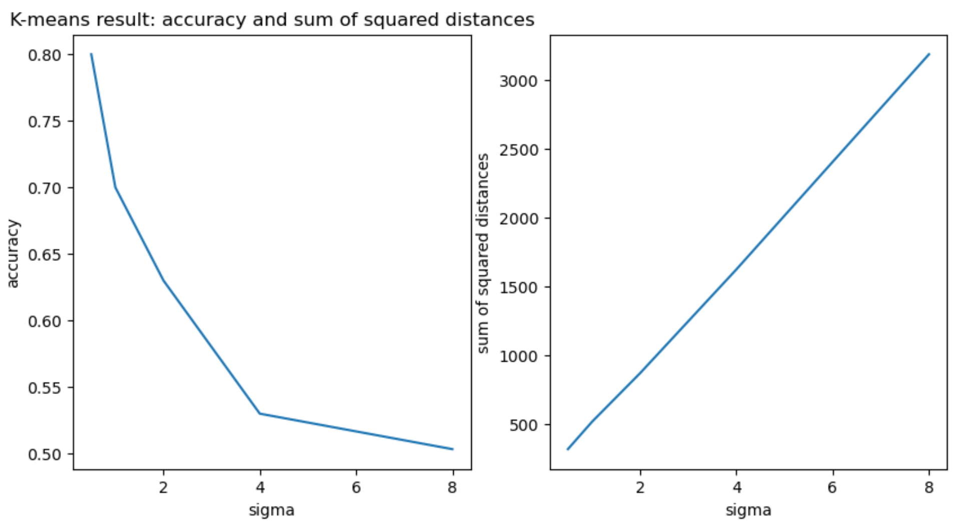


Figure 1: K-means result: accuracy and sum of squared distances

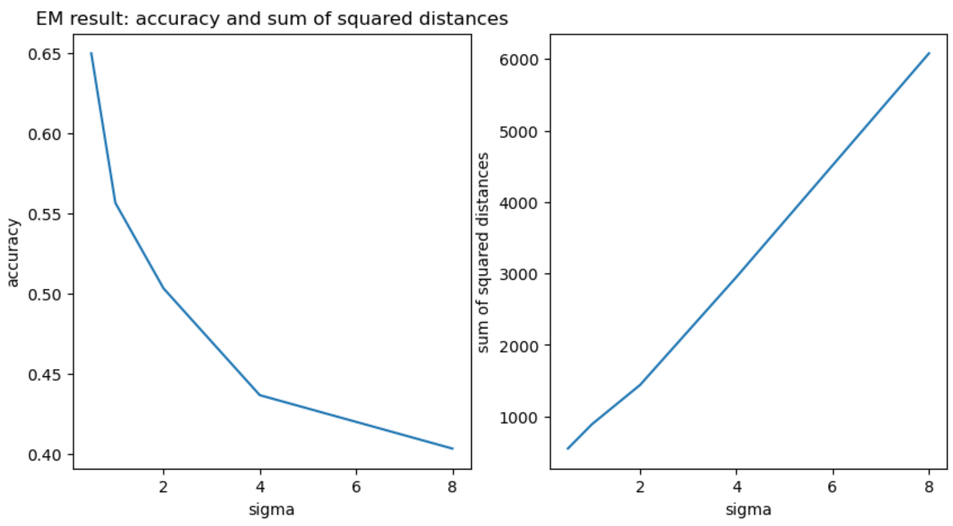


Figure 2: EM result: accuracy and negative log likelihood

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 $X = U\Sigma V^\top$ be the SVD of X. To find the first principal component, we need to find the direction in which the data varies the most, which means a_1 should be the solution of following optimization problem:

$$\max_{a_1} \operatorname{Var}(Xa_1) = (Xa_1)^T(Xa_1)/n = a_1^T\Sigma^T\Sigma a_1/n$$
$$\text{s.t. } a_1^\top a_1 = 1$$

   The solution of this problem will give the eigenvector associated with the largest eigenvalue of $\Sigma^T\Sigma$, which is the first right singular vector of X, to see that, we can exam that $X^\top X = V\Sigma^\top U^\top U\Sigma V^\top = V\Sigma^\top\Sigma V^\top$, and Σ is a diagonal matrix with decreasing entries. Thus, the solution a_1 is the first right singular vector of X.

2. (6 Points) Given $a_1, \dots, a_k$, let $A_k = [a_1, \dots, a_k]$ and $\tilde{x}_i = x_i - A_kA_k^\top x_i$. 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.

   We can show it by induction. For $k = 1$, we have already shown that a_1 is the first right singular vector of X. For k, A_k is the first k right singular vectors of X. We want to find a_{k+1} to solve the following optimization problem:

$$\max_{a_{k+1}} \operatorname{Var}(\tilde{X}a_{k+1}) = (\tilde{X}a_{k+1})^T(\tilde{X}a_{k+1})/n = a_{k+1}^T\tilde{\Sigma}^T\tilde{\Sigma}a_{k+1}/n$$
$$\text{s.t. } a_{k+1}^\top a_{k+1} = 1$$

Where $\tilde{X} = [\tilde{x}_1^\top; \dots; \tilde{x}_n^\top] = (I - V_kV_k^\top)X$ and $I - V_kV_k^\top$ is a diagonal matrix with first k entries equal to 0 and remains equal to 1. Then $\tilde{\Sigma} = (I - V_kV_k^\top)\Sigma$ is a diagonal matrix with first k entries equal to 0 and remains equal to Σ.
So problem is equivalent to finding the first eigenvactor of $\tilde{X}^\top\tilde{X}$, which is the $(k+1)^{th}$ eigenvactor of $X^\top X$ (since the first k eigenvalue in Σ are set to be 0). So, the solution a_{k+1} is the $(k+1)^{th}$ right 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 $X = [x_1^\top; \dots x_n^\top] \in \mathbb{R}^{n\times D}$. 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 $Z = [z_1^\top; \dots; z_n^\top] \in \mathbb{R}^{n\times d}$. 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, $A_2 = A_1M^{-1}$, $b_2 = b_1 - A_1M^{-1}p$ and $Z_2 = Z_1M^\top + \mathbf{1}p^\top$. Show that both (A_1, b_1, Z_1) and (A_2, b_2, Z_2) achieve the same objective value J (2).

$J(A_2, b_2, Z_2) = \|X - Z_2A_2^\top - \mathbf{1}b_2^\top\|_F^2 = \|X - (Z_1M^\top + \mathbf{1}p^\top)(A_1M^{-1})^\top - \mathbf{1}(b_1 - A_1M^{-1}p)^\top\|_F^2 = \|X - Z_1A_1^\top - \mathbf{1}b_1^\top\|_F^2 = J(A_1, b_1, Z_1)$.
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,

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

Here, $\mathbf{1}_d = [1, 1 \dots, 1]^\top \in \mathbb{R}^d$ 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.

   We first subtract the mean of X along each dimension to obtain a zero-mean matrix $\bar{X}$, which will allow us to enforce the constraint that Z has zero mean. Then we perform SVD on $\bar{X}$ to obtain $U, \Sigma, V^\top$ such that $\bar{X} = U\Sigma V^\top$. For rank d approximation, we take $U_d = U[1:n, 1:d], \Sigma_d = \Sigma[1:d, 1:d]$ and $V_d^T = V^T[1:d, 1:D]$. Now, we can calculate Z and A from the truncated SVD:

$$Z = \sqrt{n}U_d$$
$$A = V_d\Sigma_d/\sqrt{n}$$

   Since we subtracted the mean of X before performing the SVD, Z will have zero mean. Moreover, because U_d is orthogonal with n rows, the covariance matrix of Z will be an identity matrix.

Now that Z has zero mean and identity covariance, we can calculate the bias term b as the mean of the original matrix X along each dimension:

$$b = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{1}{n} X^\top \mathbf{1}_n$$

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.)
   $z_* = A^\top(x_* - b) = \Sigma_d V_d^T(x_* - b)/\sqrt{n}$

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.

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

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 $z|x$.
   The joint distribution of (z, x) is given by:
   $$\begin{pmatrix} z \\ x \end{pmatrix} = \mathcal{N}\left(\begin{pmatrix} 0 \\ b \end{pmatrix}, \begin{bmatrix} I_d & A^T \\ A & AA^T + \eta^2 I_D \end{bmatrix}\right)$$
   The marginal distribution of x is given by:
   $$x \sim \mathcal{N}(b, AA^T + \eta^2 I_D)$$
   The conditional distribution of z given x is given by:
   $$z|x \sim \mathcal{N}(A^\top(AA^T + \eta^2 I_D)^{-1}(x - b), I_d - A^\top(AA^T + \eta^2 I_D)^{-1}A)$$

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

   Define $C = AA^T + \eta^2 I_D$ and $S = \frac{1}{n}\sum_{i=1}^{n}(x_i - b)(x_i - b)^\top$. Then the log likelihood is given by:

$$\begin{aligned}\log p(x) &= \log \prod_{i=1}^{n} p(x_i)\\ &= \sum_{i=1}^{n} \log p(x_i)\\ &= -\frac{n}{2}(D\log(2\pi) + ln \det C + tr(C^{-1}S))\end{aligned}$$

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

   The MLE for b is given by:

$$b = \frac{1}{n}\sum_{i=1}^{n} x_i$$

   It can be derived from maximize the log likelihood of $p(x)$ since it only appears in S. And to maximize the log likelihood of $p(x)$ we need to find b that minimizes S, which is the mean of x in each dimension.

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.

   For our log likelihood, we have:

$$\sum_{i=1}^{n} \log p(x_i) = \sum_{i=1}^{n} \log \int Q(z_i)\frac{p(x_i, z_i)}{Q(z_i)} dz_i$$

   By Jensen's inequality, we have:

$$\sum_{i=1}^{n} \log p(x_i) \geq \sum_{i=1}^{n} \int Q(z_i) \log \frac{p(x_i, z_i)}{Q(z_i)} dz_i = \mathbb{E}_{Q(z)}\left[\log \frac{p(x, z)}{Q(z)}\right]$$

   So for any probability distribution $Q(z)$ satisfying $1 > Q(z) > 0$, this lower bound is valid.

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)

   For the E-step we need to be able to compute the expected log-likelihood with respect to the conditional distribution of our latent variables z given x and our current estimates of the parameters (E-step):

$$Q(A, \eta|A^t, \eta^t) = \mathbb{E}_{z|x,A^t,\eta^t}\left[\log p(x|z, (A, \eta))\right]$$

This can be derived by setting $Q(z_i) = \mathbb{P}(z_i|x_i)$ to the lower bound we got in last question. We can keep going to get:

$$\begin{aligned}
Q(A,\eta) &= \mathbb{E}_{z|x,A,\eta}\left[\log p(x|z,(A,\eta))\right] \\
&= -\frac{1}{2}\sum_{i=1}^{n}\mathbb{E}_{z|x,A,\eta}\left[(x_i - b - Az_i)^\top(\eta^2 I_D)^{-1}(x_i - b - Az_i)\right] - n^2\log\eta + constant \\
&= -\frac{1}{2\eta^2}\sum_{i=1}^{n}\mathbb{E}_{z|x,A,\eta}\left[\bar{x}_i^\top\bar{x}_i - 2\bar{x}_i^\top Az_i + z_i^\top A^\top Az_i\right] - n^2\log\eta + constant \\
&= -\frac{1}{2\eta^2}\sum_{i=1}^{n}\left[\bar{x}_i^\top\bar{x}_i - 2\bar{x}_i^\top A\mathbb{E}_{z|x,A,\eta}[z] + \mathbb{E}[z^\top A^\top Az]\right] - n^2\log\eta + constant \\
&= -\frac{1}{2\eta^2}\sum_{i=1}^{n}\left[\bar{x}_i^\top\bar{x}_i - 2\bar{x}_i^\top A\mathbb{E}_{z|x,A,\eta}[z] + tr(A^\top A\mathbb{E}_{z|x,A,\eta}[zz^\top])\right] - n^2\log\eta + constant
\end{aligned}$$

Before M-step we need to compute the expected value of z and $zz^\top$, which can get from Q2.3.1: Let $\beta = A^\top C = A^\top(AA^\top + \eta^2 I)$

$$\mathbb{E}_{z|x,A,\eta}[Z] = \beta(X - b\mathbf{1}_n^\top)$$
$$\mathbb{E}_{z|x,A,\eta}[ZZ^\top] = I_d - \beta A + \beta XX^\top\beta^\top$$

Where X and Z is the data matrix with each column being a data point, each row being a feature.

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.

Derivative w.r.t. A:

$$\frac{\partial Q(A,\eta)}{\partial A} = -\frac{1}{2\eta^2}\sum_{i=1}^{n}\left[-2\bar{x}_i\mathbb{E}[z]^\top + 2A\mathbb{E}[zz^\top]\right]$$

Setting it to 0, we can solve for A:

$$A^* = (X\mathbb{E}_{z|x,A,\eta}[Z]^\top)/(\mathbb{E}_{z|x,A,\eta}[ZZ^\top])$$

Derivative w.r.t. η:

$$\frac{\partial Q(A,\eta)}{\partial\eta} = -\frac{1}{2\eta^3}\sum_{i=1}^{n}\left[-2\bar{x}_i^\top\bar{x}_i + 2\bar{x}_i^\top A\mathbb{E}[z] + 2tr(A^\top A\mathbb{E}[zz^\top])\right] - n^2/\eta$$

Setting it to 0, we can solve for η:

$$\eta^{2*} = \frac{1}{nD}tr(XX^\top - A^*\mathbb{E}_{z|x,A,\eta}[Z]X^\top)$$

Bring in the $\mathbb{E}_{z|x,A,\eta}[Z]$ and $\mathbb{E}_{z|x,A,\eta}[ZZ^\top]$ from Q2.3.5, we can get new A^* and η^*.

2.4 Experiment (42 points)

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

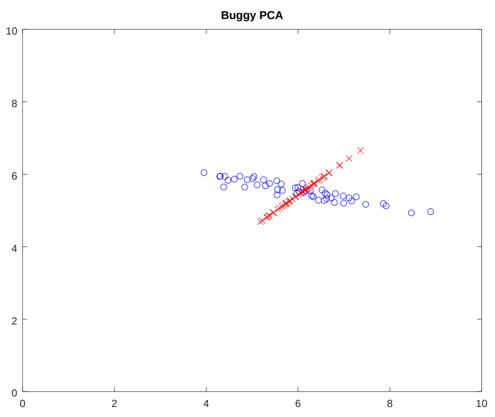
- We will implement three variants of PCA:
  1. "buggy PCA": PCA applied directly on the matrix X.

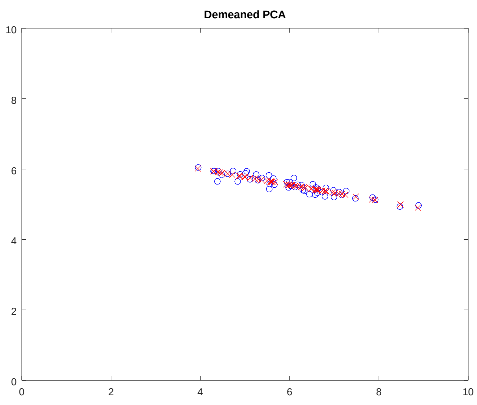
2. "demeaned PCA": We subtract the mean along each dimension before applying PCA.
3. "normalized PCA": Before applying PCA, we subtract the mean and scale each dimension so that the sample mean and standard deviation along each dimension is 0 and 1 respectively.

- 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.
- 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?
  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?
- 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.
From Fig3.png we can see the knee point is 30, so we choose $d = 30$. In DRLV I set $A_0 = A$ from DRO,

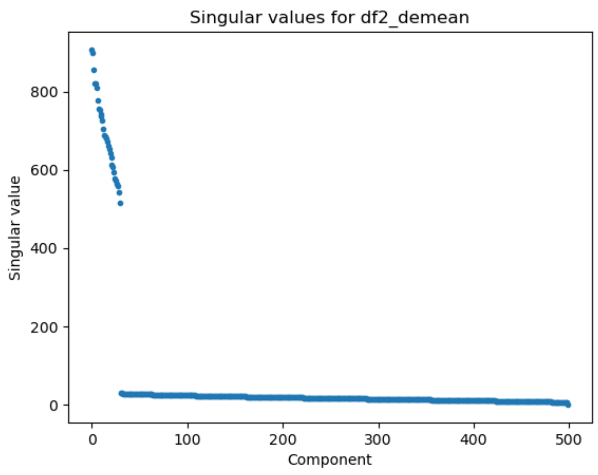


Figure 3: Knee point in the singular values of data1000D

and set $\eta_0^2 = \frac{1}{D} tr(cov(X) - AA^\top)$.

Reconstruction Report for data1000D.csv:
Buggy PCA: 802.7313986203584
Demeaned PCA: 273.0459589786028
Normalized PCA: 273.62858099762326
DRO: 273.0459589786028
DRLV: 313.69333446428493

Q1: Buggy PCA is trying to project the points onto a line that passes through the origin. However, the original points are not seem to on a line passes through the origin, so the reconstruction error is high.
Q2: No, because these two methods are equivalent to each other. The process of them are both demean, then do SVD, and then project the points onto the first d principal components, finally add the mean back. So the reconstruction error is the same. Their cluster results z are only different in the scale, but the reconstruction results $r(z)$ are the same.