

HOMEWORK 5

Mingcong Cao
9084259218

Instructions: Use this latex file as a template to develop your homework. Submit your homework on time as a single pdf file. Please wrap your code and upload to a public GitHub repo, then attach the link below the instructions so that we can access it. Answers to the questions that are not within the pdf are not accepted. This includes external links or answers attached to the code implementation. 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. It is ok to share the experiments results and compare them with each other.
https://github.com/Mingcong-Cao/ECE760_Homework5.git

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 $f: \mathcal{X} \rightarrow \{1, \dots, K\}$ so as to minimize the objective

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

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 .

Lemma: For a fixed partition of clusters, the best representations under the objective function are the centers (proof omitted).

Proof: For any $m < k < n$, there is at least one cluster with at least two elements, then name that cluster h . Take one element out of h to form a single element cluster $m+1$, and name the new rule f' , the new center for cluster h μ'_h . We have

$$\begin{aligned} & \sum_{i=1}^n \mathbb{1}(f(X_i) = h) \|X_i - \mu_h\|^2 \\ & \geq \sum_{i=1}^n \mathbb{1}(f'(X_i) = h) \|X_i - \mu_h\|^2 \quad (\text{one element is taken out}) \\ & \geq \sum_{i=1}^n \mathbb{1}(f'(X_i) = h) \|X_i - \mu'_h\|^2 \quad (\text{by lemma}) \end{aligned}$$

For the single element cluster $m+1$, the distance would be 0. Therefore, for the new partition f' , we have $\mathcal{J}_{m+1}(X_1^n) \leq J(\mu_1^{m+1}, f'; X_1^n) \leq \mathcal{J}_m(X_1^n)$.

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 consider when the clusters between two steps are different. The clusters are different only when some points find a closer center. Thus, the distance of each point to its corresponding center will never increase, and that distance for at least one point will strictly decrease. Denote the partition at iteration t as f^t and centers as $\mu_1^{K,t}$, when the clusters are different, by lemma we have:

$$J(\mu_1^{K,t+1}, f^{t+1}; X_1^n) \leq J(\mu_1^{K,t+1}, f^t; X_1^n) < J(\mu_1^{K,t}, f^t; X_1^n)$$

If the clustering of two iterations is the same, then the algorithm terminates. When the algorithm iterates before convergence, it cannot encounter the same clustering twice because J decreases strictly and the same clustering can only have the same value J . As the number of possible clusters is finite, the algorithm 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 .

Guidelines:

- 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.

Results for GMM are shown below:

σ	0.5	1	2	4	8
eval metrics	932.16	1044.28	1214.04	1399.71	1588.82
Acc	0.853	0.563	0.57	0.513	0.42

Results for K-means are shown below:

σ	0.5	1	2	4	8
eval metrics	297.78	492.48	901.13	1687.53	3176.81
Acc	0.82	0.74	0.573	0.533	0.527

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 $\frac{1}{n} \sum_{i=1}^n \|A^\top x_i\|^2$.

- (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 .

We are finding $a_1 \in \mathbb{R}^D$ and $a_1^\top a_1 = 1$ that minimize $\|X a_1\|^2$

Let the SVD of X be

$$X = \sum_{j=1}^r \sigma_j \mathbf{u}_j \mathbf{v}_j^\top$$

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > \sigma_{r+1} = \dots = \sigma_d = 0$$

where r is the rank of X . Complete \mathbf{v}_j into an orthonormal basis of \mathbb{R}^d . Let $\langle a, b \rangle = ab^\top$. We have:

$$\begin{aligned} \|Xa_1\|^2 &= \|X(\sum_{i=1}^d \langle a_1, v_i \rangle v_i)\| \\ &= \|\sum_{i=1}^d \langle a_1, v_i \rangle Xv_i\| \\ &= \|\sum_{i=1}^d \langle a_1, v_i \rangle \sigma_i u_i\| \\ &= \sum_{i=1}^d \langle a_1, v_i \rangle^2 \sigma_i^2 \\ &\leq \sum_{i=1}^d \langle a_1, v_i \rangle^2 \sigma_1^2 \\ &= \sigma_1^2 \end{aligned}$$

In the second last step, we use the fact that a_1 has norm 1, and v_i from an orthonormal basis in space \mathbb{R}^D . When $a_1 = v_1$, the upper bound is taken. Thus we have v_1 optimize the objective.

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

\tilde{x}_i equals to x_i minus the projection of x_i to the space $\text{span}\{a_1, a_2, \dots, a_k\}$, thus \tilde{x}_i doesn't have components in that space. Then we could assume a_{k+1} is perpendicular to $\text{span}\{a_1, a_2, \dots, a_k\}$. With this constraint, the objective could be transformed into $\frac{1}{n} \sum_i (a_{k+1}^\top x_i)^2$, which is the same one in the last question. By following the same procedure we have:

$$\begin{aligned} \|Xa_{k+1}\|^2 &= \|X(\sum_{i=1}^d \langle a_{k+1}, v_i \rangle v_i)\| \\ &= \|\sum_{i=1}^d \langle a_{k+1}, v_i \rangle Xv_i\| \\ &= \|\sum_{i=1}^d \langle a_{k+1}, v_i \rangle \sigma_i u_i\| \\ &= \sum_{i=1}^d \langle a_{k+1}, v_i \rangle^2 \sigma_i^2 \\ &\leq \sigma_{k+1}^2 \end{aligned}$$

because a_{k+1} is perpendicular to $\text{span}\{a_1, a_2, \dots, a_k\}$, namely $\text{span}\{v_1, v_2, \dots, v_k\}$, the first k summation will be zero. When $a_{k+1} = v_{k+1}$, the upper bound is taken. Thus we have v_{k+1} optimize the objective.

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

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

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).

$$\begin{aligned} & \|X - Z_2 A_2^\top - \mathbf{1} b_2^\top\|_F^2 \\ &= \|X - (Z_1 M^\top + \mathbf{1} p^\top)(M^\top)^{-1} A_1^\top - \mathbf{1}(b_1^\top - p^\top (M^\top)^{-1} A_1^\top)\|_F^2 \\ &= \|X - Z_1 M^\top (M^\top)^{-1} A_1^\top - \mathbf{1} p^\top (M^\top)^{-1} A_1^\top - \mathbf{1} b_1^\top + \mathbf{1} p^\top (M^\top)^{-1} A_1^\top\|_F^2 \\ &= \|X - Z_1 A_1^\top - \mathbf{1} b_1^\top\|_F^2 \end{aligned}$$

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,

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

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.

First we find the optimal solution for b . Let $\tilde{X} = Z A^\top$. Then $J(A, b, Z) = \sum_{i,j} (X_{ij} - \tilde{X}_{ij} - b_j)^2$. Set its derivative w.r.t b to zero:

$$\begin{aligned} \frac{\partial J}{\partial b_j} &= \sum_i -2(X_{ij} - \tilde{X}_{ij} - b_j) = 0 \\ b_j &= \frac{1}{n} \sum_i (X_{ij} - \tilde{X}_{ij}) \\ b &= \frac{1}{n} (X^\top \mathbf{1}_n - \tilde{X}^\top \mathbf{1}_n) = \frac{1}{n} (X^\top \mathbf{1}_n - A Z^\top \mathbf{1}_n) = \frac{1}{n} X^\top \mathbf{1}_n \end{aligned}$$

Then we will optimize the objective by rank k approximation of matrix $X - \mathbf{1} b^\top$. By zeroing out all but the first d singular values, we have the approximation $U \Sigma V^\top$, where $U \in \mathbb{R}^{n \times d}$, $\Sigma \in \mathbb{R}^{d \times d}$ and $V \in \mathbb{R}^{D \times d}$. It is obvious that $Z = \sqrt{n} U$ and $A^\top = \frac{1}{\sqrt{n}} \Sigma V^\top$ is a solution to this optimization problem. Now we only need to show the constraints are satisfied. First we prove the identity covariance:

$$Z^\top Z = n U^\top U = n I_d$$

For the zero mean, it's easy to verify that $(X - \mathbf{1} b^\top)^\top \mathbf{1}_n$ is zero. Let the SVD of $(X - \mathbf{1} b^\top)$ without zeroing out be $(X - \mathbf{1} b^\top) = \sum_{j=1}^{rank} \sigma_j \mathbf{u}_j \mathbf{v}_j^\top$. Then

$$(X - \mathbf{1} b^\top)^\top \mathbf{1} = \sum_{j=1}^{rank} \sigma_j \mathbf{v}_j \mathbf{u}_j^\top \mathbf{1} = 0$$

Because \mathbf{v}_j are linearly independent, so we have $\mathbf{u}_j^\top \mathbf{1} = 0$, namely $U^\top \mathbf{1} = 0$

$$Z^\top \mathbf{1}_n = \sqrt{n} U^\top \mathbf{1}_n = 0$$

To summarize, we have $b = \frac{1}{n} X^\top \mathbf{1}_n$, $Z = \sqrt{n} U$ and $A^\top = \frac{1}{\sqrt{n}} \Sigma V^\top$

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 should be the d -dimensional representation z_i .)

We need to minimize the objective: $\|x_*^T - z^T A^T - b^T\|_F^2 = \|x_* - Az - b\|_F^2$. This is solved by letting Az be the projection of $x_* - b$ into the column space of A . Because the columns of A are orthogonal,

$$\text{Projection of } x_* \text{ into } \text{Col}(A) = A(\Sigma^{-1})^2 A^T (x_* - b) = Az$$

$$z = A^T A (\Sigma^{-1})^2 A^T (x_* - b) = (\Sigma^{-1})^2 A^T (x_* - b)$$

2.3 Experiment (34 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 $V^T z$. For the preprocessed versions, we first do this and then reverse the preprocessing steps as well. For DRO we just compute $Az + b$. We will compare all methods by the reconstruction error on the datasets.
- Please implement code for the methods: Buggy PCA (just take the SVD of X), Demeaned PCA, Normalized PCA, DRO. In all cases your function should take in an $n \times d$ data matrix and d as an argument. It should return the d dimensional representations, the estimated parameters, and the reconstructions of these representations in D dimensions.
- 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 original points with the reconstructed points for all 4 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: **(6 Points)**
 - Implementation of DRO: **(6 points)**
 - Plots showing original points and reconstructed points for 2D dataset for each one of the 4 methods: **(10 points)**
 - Implementing reconstructions and reporting results for each one of the 4 methods for the 2 datasets: **(5 points)**

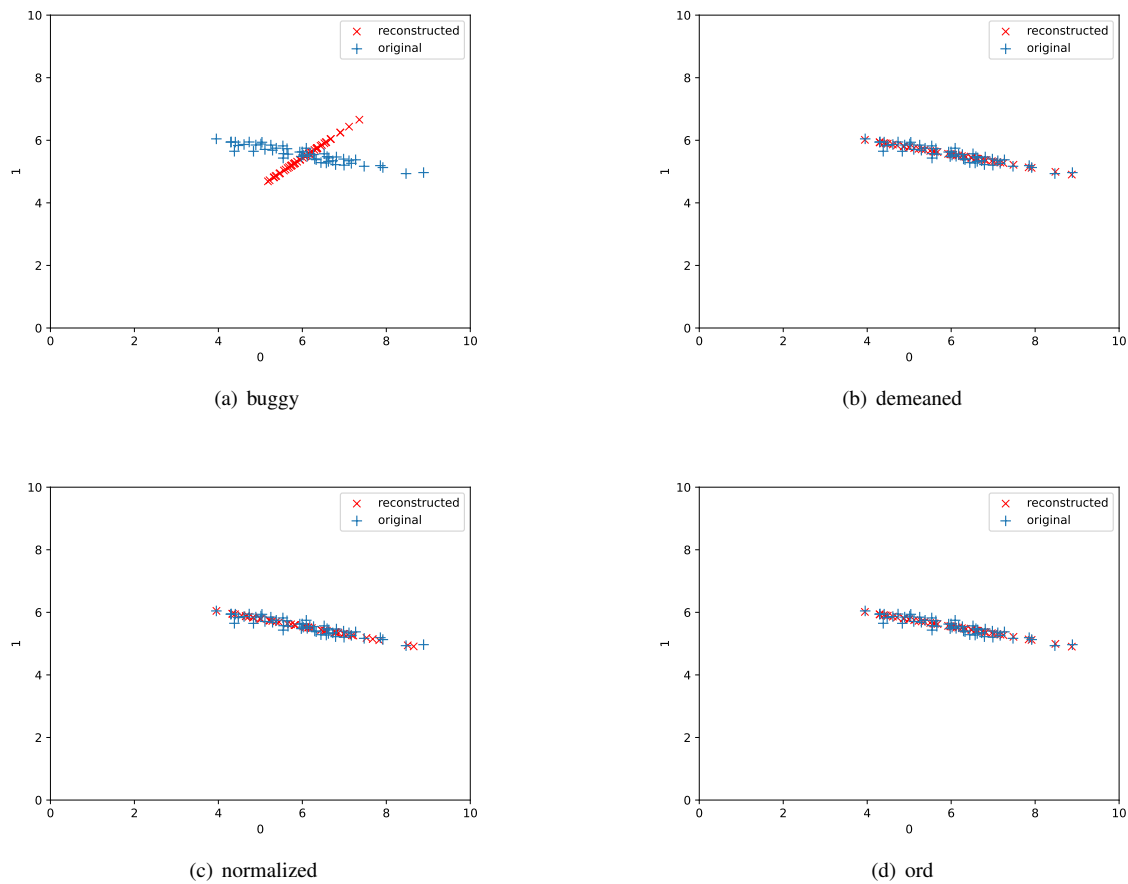


Figure 1: PCA 2d

- Choice of d for $1000D$ dataset and appropriate justification: **(3 Points)**
- Questions **(4 Points)**

Answer format:

The graph below is an example of how a plot of one of the algorithms for the 2D dataset may look like:

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

And this is how the reconstruction error may look like for Buggy PCA for the 2D dataset: 0.886903

We should set $d=30$ for the $1000D$ dataset as can be seen from figure 2.

The reconstruction error of the 2d dataset is shown in the table below:

buggy	demeaned	normalized	ord
44.34	0.5003	2.4736	0.5003

The reconstruction error of the 1000d dataset is shown in the table below:

buggy	demeaned	normalized	ord
401365.70	136522.97	136814.29	136522.97

Q1: The buggy PCA tries to project the original dataset onto the subspace of every single line that passes through the origin. Because the X is not demeaned or normalized, the cluster is very far away from the origin, resulting in a buggy PCA.

Q2: Not surprising. We can observe that the b in ord is actually the mean of X , and the A is just PCA in a different scale. Thus those two methods are equivalent.

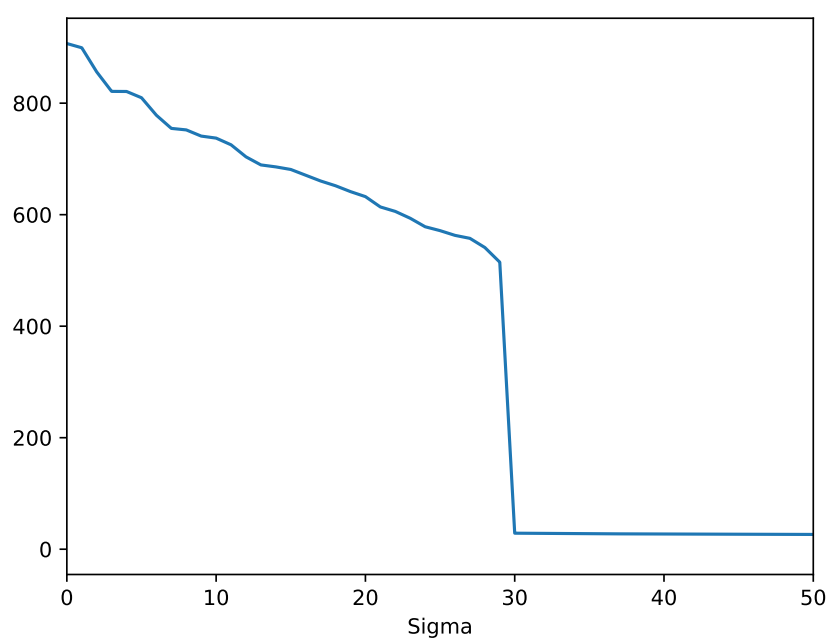


Figure 2: Singular values of 1000D dataset