Stat601(Section001): Homework #3

Due on Feb. 23, 2021 at 11:59pm

Instructor:Ziwei Zhu

Tiejin Chen **tiejin@umich.edu**

Problem 1

We can get:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} y_i = \bar{Y}$$

Followed by the formula of condition pdf of $x_i|y_i$,we can get:

$$x_i|y_i \sim N((I + \Lambda^T\Phi^{-1}\Lambda)^{-1}\Lambda^T\Phi^{-1}(y_i - \mu), (I + \Lambda^T\Phi^{-1}\Lambda)^{-1})$$

We have $y_i|x_i \sim N(\mu + \Lambda x, \Phi)$,we have a similar result in E-step:

$$\sum_{i=1}^{n} logp(y_i|x_i, \theta) = C - \frac{N}{2}log|\Phi| - \frac{N}{2}Tr[\Phi^{-1}E_{X|Y,\theta}(S)]$$

Where C is a constant and S here is:

$$S := \frac{1}{N}\sum_{i=1}^{N}(y_i - \mu - \Lambda x_i)(y_i - \mu - \Lambda x_i)^T$$

And:

$$E_{X|Y}(S) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \mu)(y_i - \mu)^T - 2(y_i - \mu)E(x_i^T|y_i)\Lambda^T + \Lambda E(x_i x_i^T|y_i)\Lambda^T$$

Now, we consider M-step.Taking derivate of it, we can get:

$$\frac{\partial \sum_{i=1}^{n} logp(y_i|x_i, \theta)}{\partial \mu} = 0 \to \mu = \frac{1}{n}\sum_{i=1}^{n} y_i = \bar{Y}$$

$$\frac{\partial \sum_{i=1}^{n} logp(y_i|x_i, \theta)}{\partial \Lambda} = 0 \to \Lambda = [\sum_{i=1}^{n}(y_i - \mu)E(x_i^T|y_i)][\sum_{i=1}^{n} E(x_i x_i^T|y_i)]^{-1}$$

$$\frac{\partial \sum_{i=1}^{n} logp(y_i|x_i, \theta)}{\partial \Phi} = 0 \to \Phi = E(S|Y)$$

Let$(I + \Lambda^T\Phi^{-1}\Lambda)^{-1} = A$, we have:

$$E(x_i x_i^T|y_i) = Var(x_i|y_i) + E(x_i|y_i)E(x_i|y_i)^T = A + A\Lambda^T\Phi^{-1}(y_i - \mu)(A\Lambda^T\Phi^{-1})^T(y_i - \mu)^T$$

And we can get:

$$\Lambda_{t+1} = [\sum_{i=1}^{N}(y_i - \mu)(A\Lambda_t^T\Phi_t^{-1}y_i)^T][\sum_{i=1}^{N}(A + A\Lambda_t^T\Phi_t^{-1}(y_i - \mu)(A\Lambda_t^T\Phi_t^{-1})^T(y_i - \mu)^T)]^{-1}$$

$$\Phi_{t+1} = diag[E(S|Y, \theta^t]) = diag(\frac{1}{N}\sum_{i=1}^{N}(y_i - \mu)(y_i - \mu)^T - 2(y_i - \mu)E(x_i^T|y_i)\Lambda^T + \Lambda E(x_i x_i^T|y_i)\Lambda^T)$$

Basically, we just get expection of $l_{X|Y}(\theta|X, Y)$ in E-step. And calulate the MLE of θ in M-step.

Problem 2

We have model assumption for Probabilistic PCA are:

$$y_i = \mu + \Lambda x_i + w_i, x_i \sim N_p(0, I_p).w_i \sim N_q(0, \sigma^2 I)$$

Problem 2 continued on next page...

We can see that it is very close to the factor analysis where the only difference is the variance ofw_i. In fact, we can let $\Phi = \sigma^2 I$. Then all the process in E-step will be same as factor analysis. Actually, the only thing difference here will we need to calculate $\frac{\partial E_{X|Y} l(\theta|X,Y)}{\partial \sigma^2}$ instead of $\frac{\partial E_{X|Y} l(\theta|X,Y)}{\partial \Phi^{-1}}$. And we have:

$$\frac{\partial E_{X|Y} l(\theta|X,Y)}{\partial \sigma^2} = (\frac{\partial \Phi^{-1}}{\partial \sigma^2})^T \frac{\partial E_{X|Y} l(\theta|X,Y)}{\partial \Phi^{-1}}$$

And we have:

$$\Phi = \sigma^2 I, \Phi^{-1} = \frac{1}{\sigma^2} I$$

Thus:

$$\frac{\partial \Phi^{-1}}{\partial \sigma^2} = -\sigma^4 I$$

Thus:

$$\frac{\partial E_{X|Y} l(\theta|X,Y)}{\partial \sigma^2} = -\sigma^4 I(\frac{N}{2}\Phi - \frac{N}{2}E(S|Y)) = 0$$

Hence we can get the update relu in M-step:

$$\Lambda_{t+1} = [\sum_{i=1}^{N}[\sum_{i=1}^{n}(y_i - \mu)E(x_i^T|y_i)][\sum_{i=1}^{n} E(x_i x_i^T|y_i)]^{-1}$$

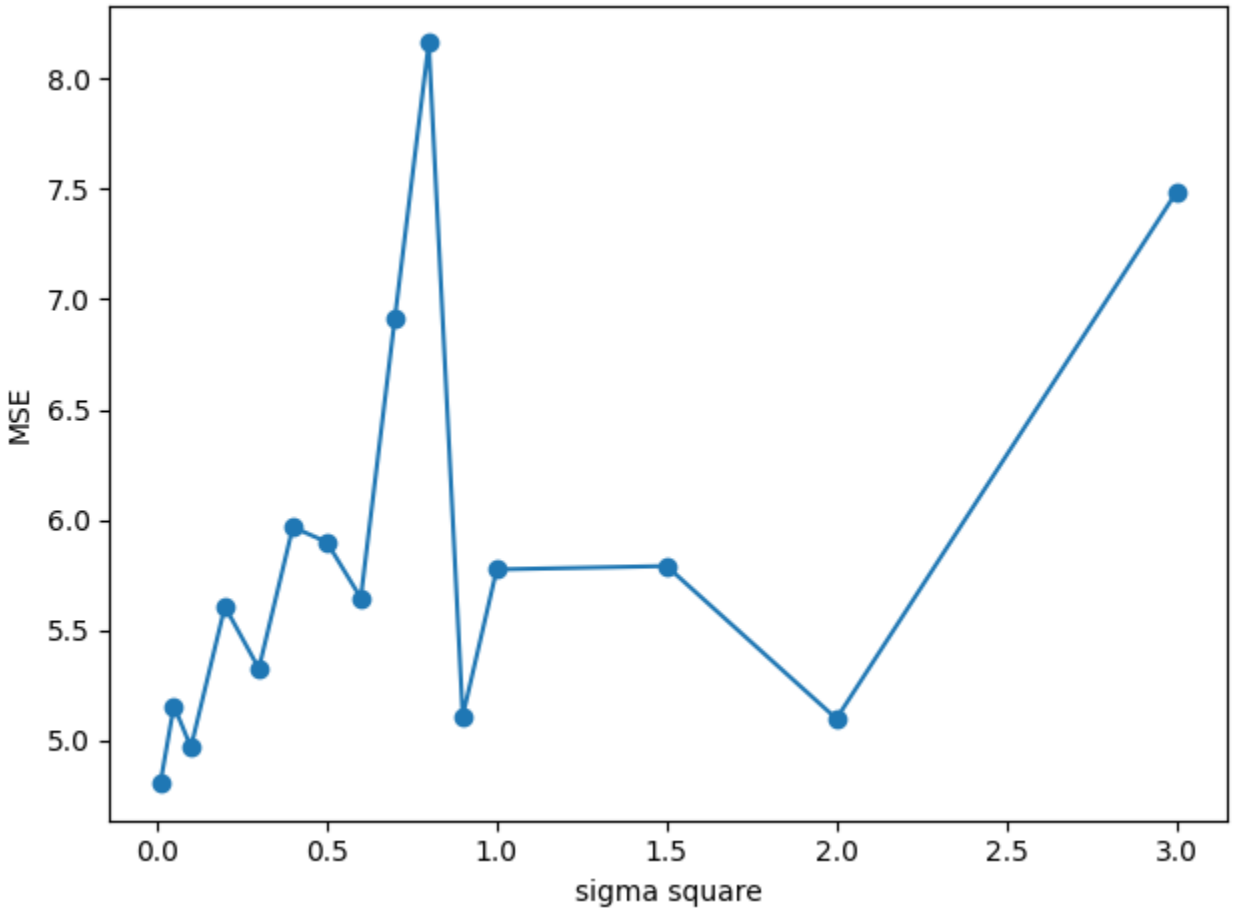
$$\sigma_{t+1}^2 = \frac{1}{q} sum(diag[E(S|Y))]$$

Problem 3

(a)

In this part, we will use mse of $\hat{X}$ and X to measure the result. If we can have smaller F-norm value, then the estimator is more accurate. And we will pick σ^2 from $(0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.5, 2, 4)$. For every σ, we will have initialization of $\Phi = I$. To make the results as accurate as possible, we will run 20 experiment and take mean result for every σ , and for every experiment, we will run 75 iterations. And we plot the figure:

Problem 3 continued on next page...



In general, we can see, with the increasing of σ^2, the F-norm increase. However, we can see that it is a trend of fluctuation. I think that is because EM is very sensitive to the initial value and we do not have enough experiment times, or maybe it just because the initialization of the Λ leads us to an orthornormal transformation of Λ thus make X different here.

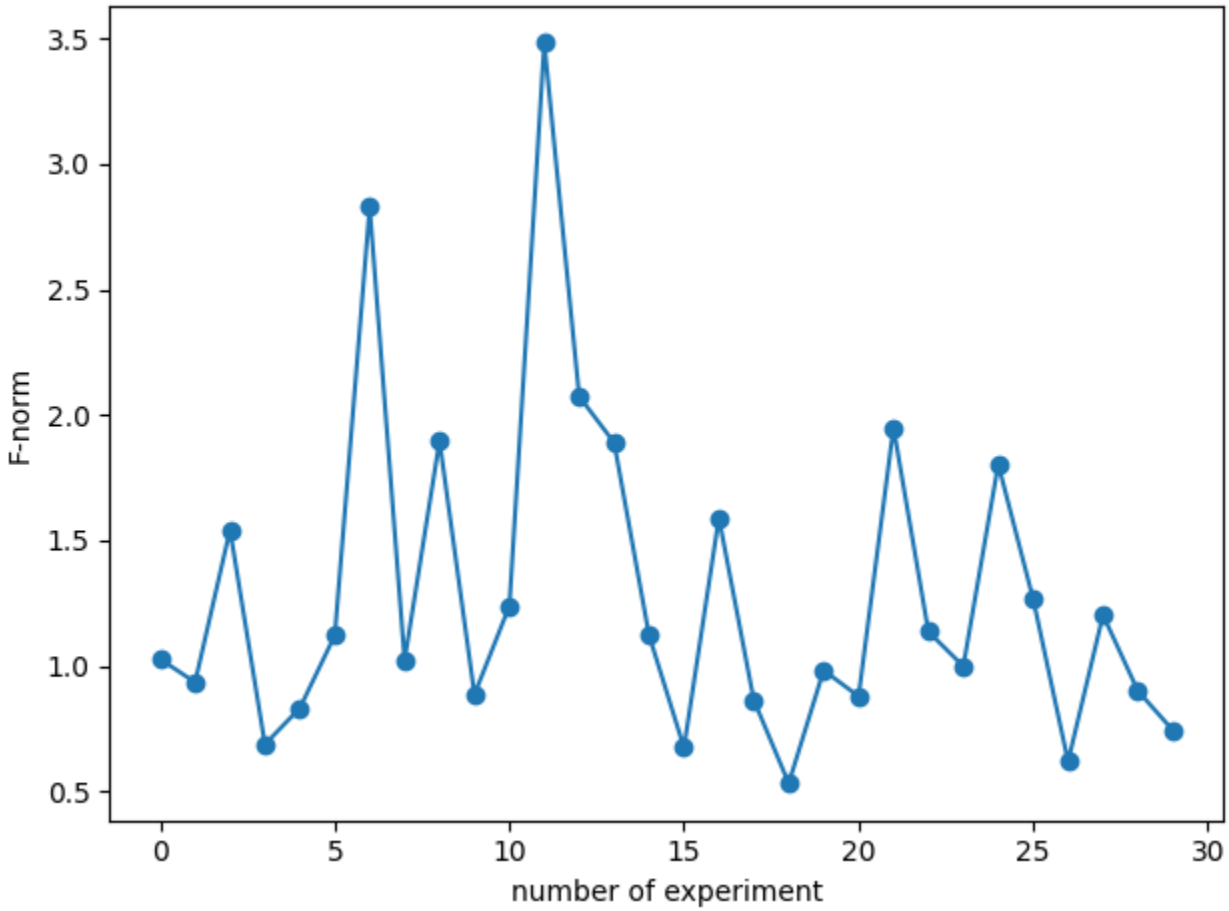
(b)

We will again use $\left\|\hat{\Lambda}\hat{\Lambda}^T - \Lambda^*\Lambda^{*T}\right\|_F$ to get the measure. Because if we do an orthornormal transformation to $\hat{\Lambda}$, we have:

$$\Lambda = \hat{\Lambda}O \rightarrow \Lambda\Lambda^T = \hat{\Lambda}OO^T\hat{\Lambda}^T = \hat{\Lambda}\hat{\Lambda}^T$$

They have same value. Thus we will use this measure. We run 30 experiments and every time we random initialize Λ. We get the result:

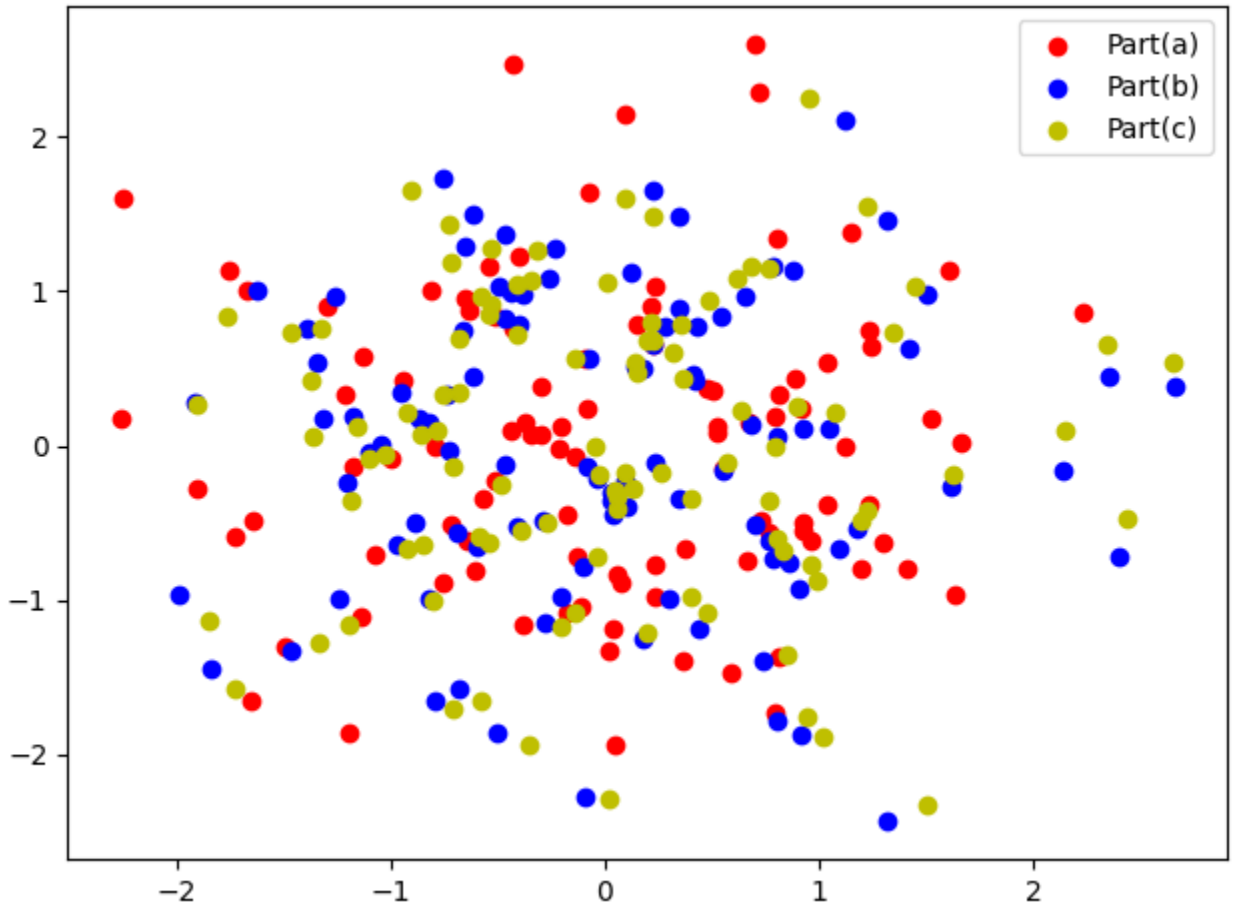
Problem 3 continued on next page...



We can find that the different initialization of Λ affect the performance of EM. And some of them are better, some of them are worse. And the gap between them are very big, Hence, we think that is beacuse EM is very sensitive to the initial value.

(c)

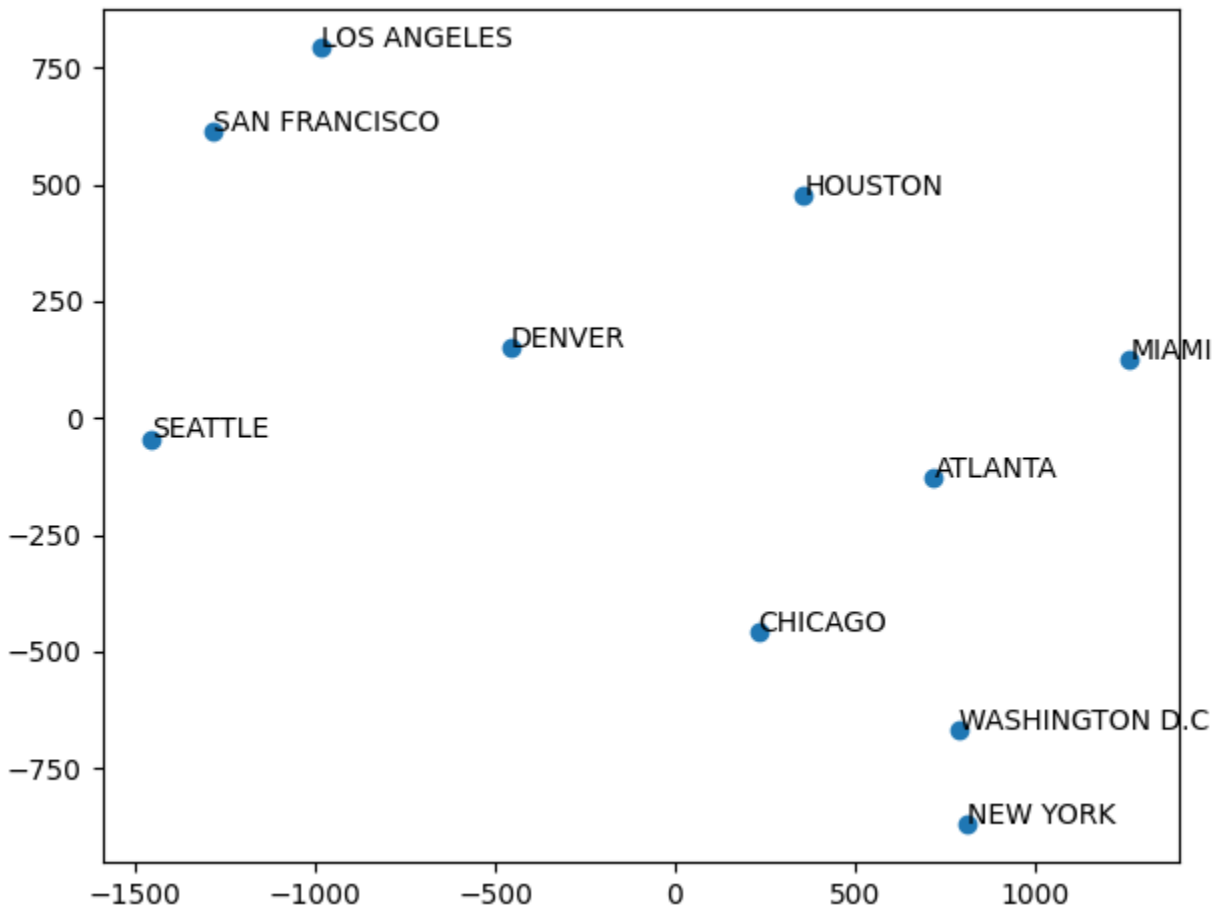
We will compare PPCA with part(a) model with $\sigma^2 = 0.4$ and one random initialization from part(b). And for PPCA, we will use the same initialization with part(b). And we get the plot:



We can see the Probabilistic PCA result is very close to part(a) and part(b), especially with part(b). And that makes sense since in this problem, since $\Phi = 0.4 * I$. And the reason why the result of PPCA is especially similar to part(b) is we use the same initialization.

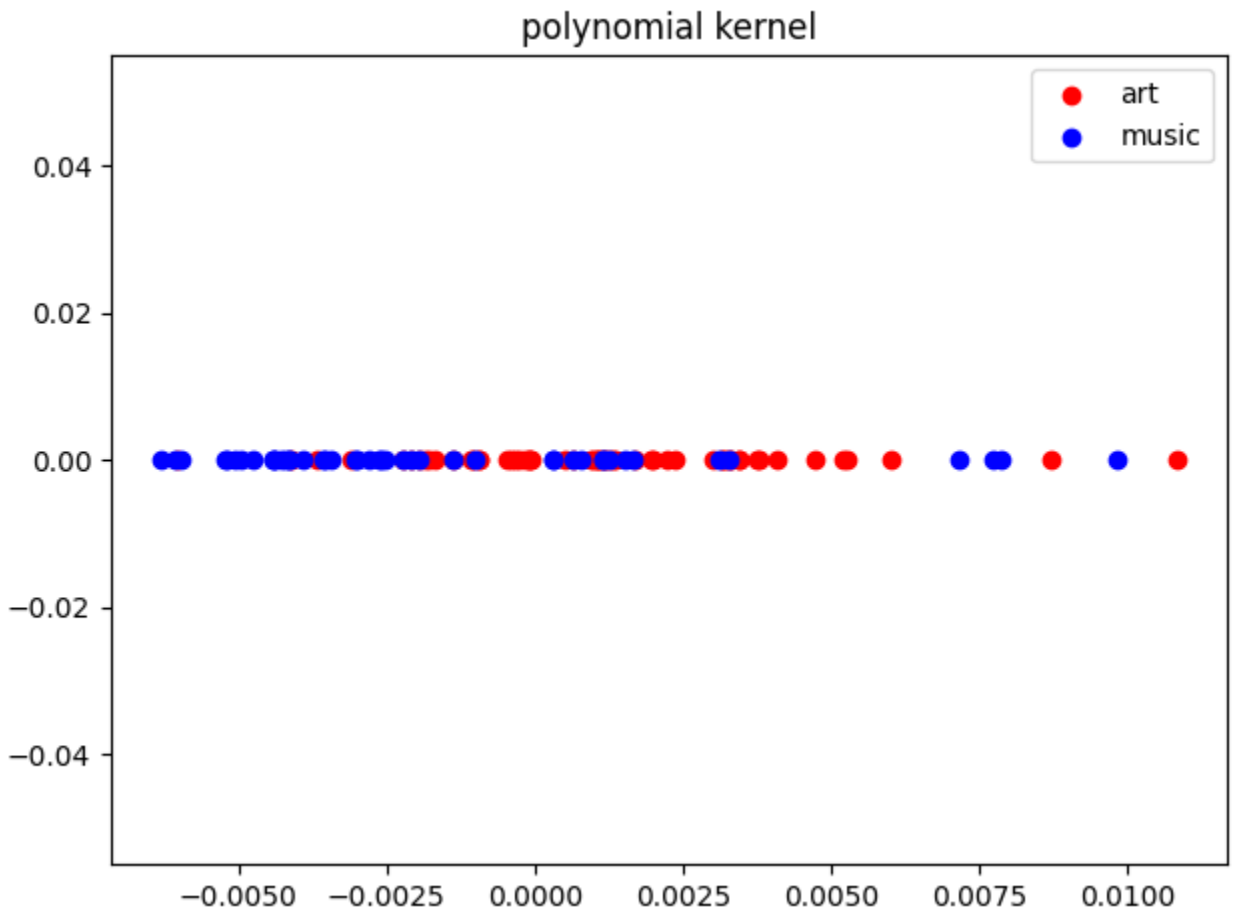
Problem 4

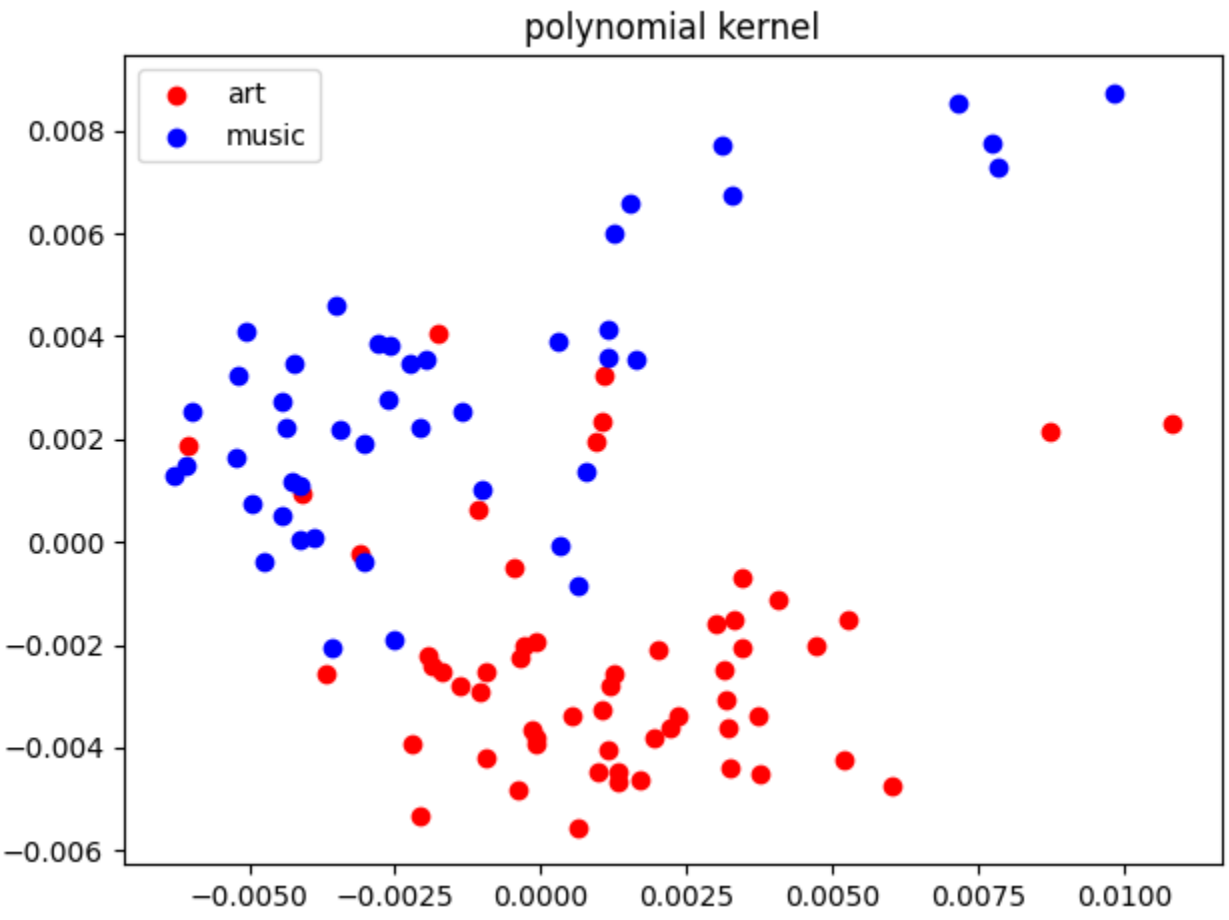
In this problem, we only have the distance matrix, which is YY^T. And under Euclidean distance, we know the only thing we need to do is just calculate the spectral decomposistion of YY^T. However, the second largest eigenvalue is a negative number, which causes our result bad. By experiment, we find using sklearn will get a much better result, we can get the map:

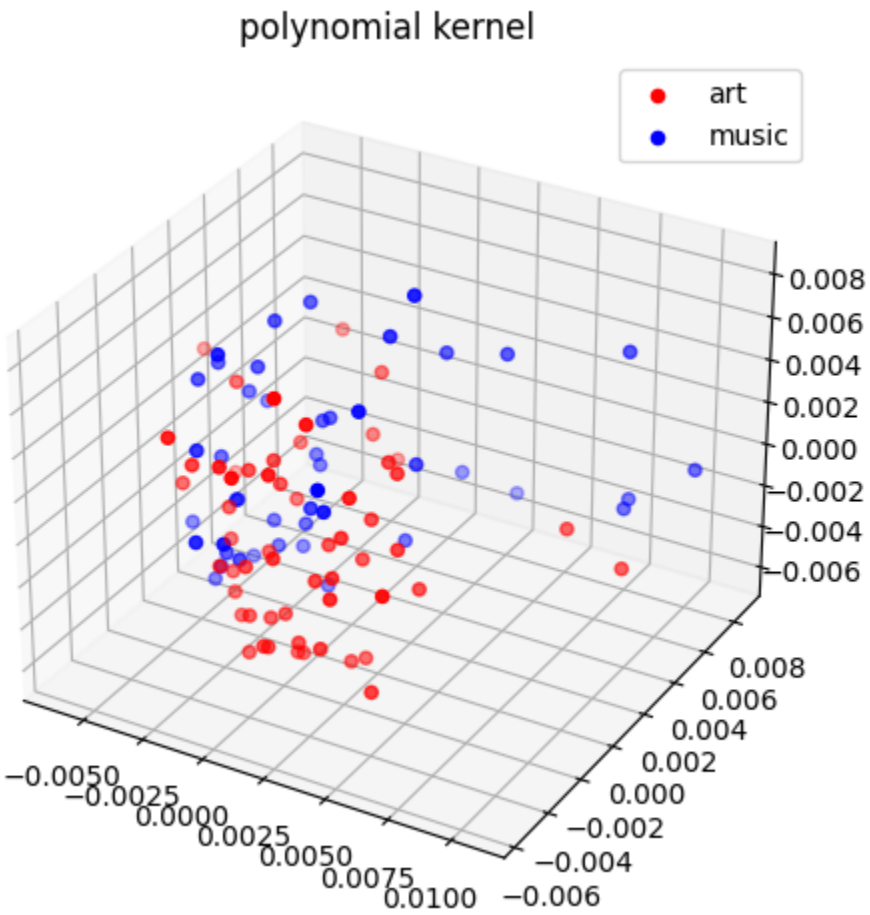


Problem 5

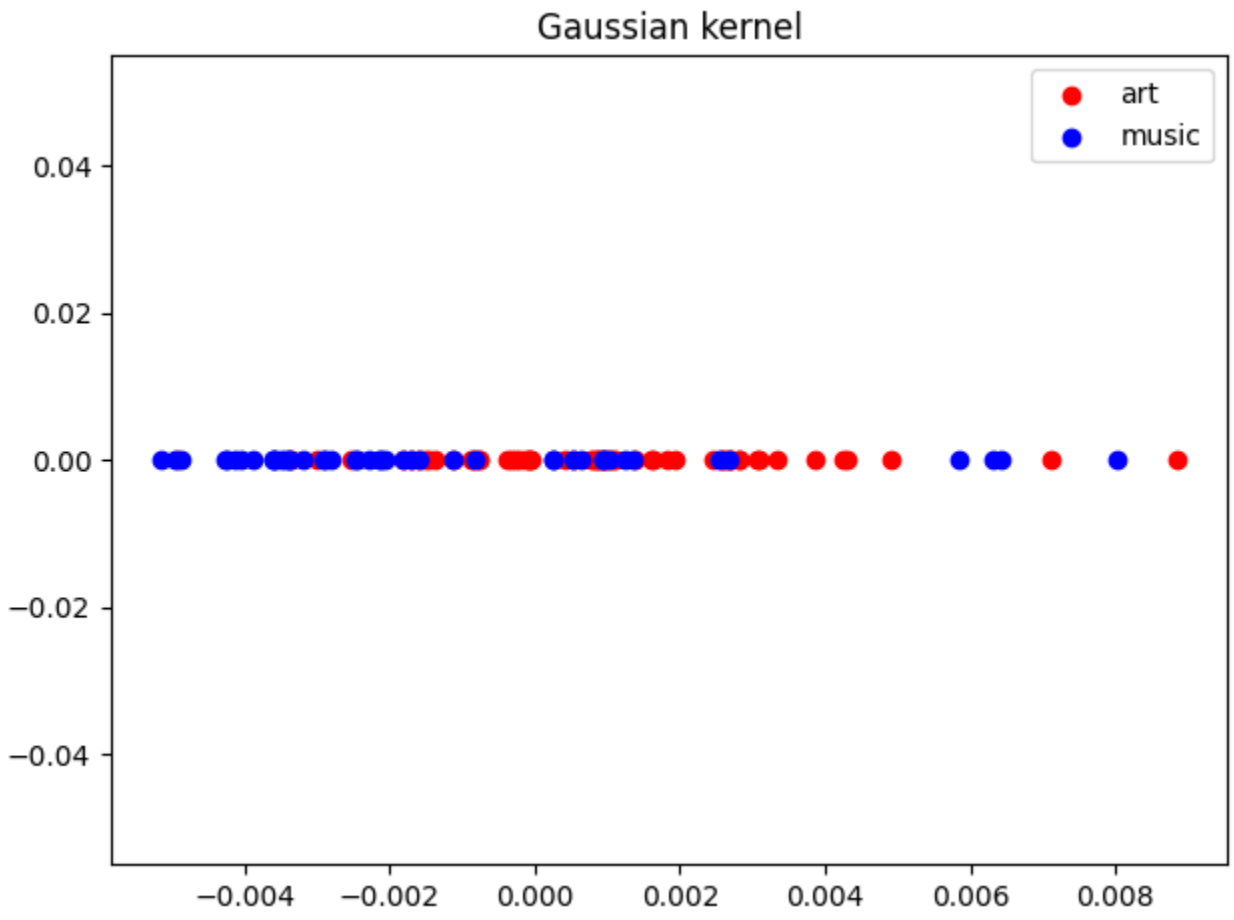
In this problem, the degree of polynomial kernel is 3, which is the default setting in sklearn. For polynomial kernel, we have the following figures:


polynomial kernel
art
music
0.04
0.02
0.00
−0.02
−0.04
−0.0050
−0.0025
0.0000
0.0025
0.0050
0.0075
0.0100

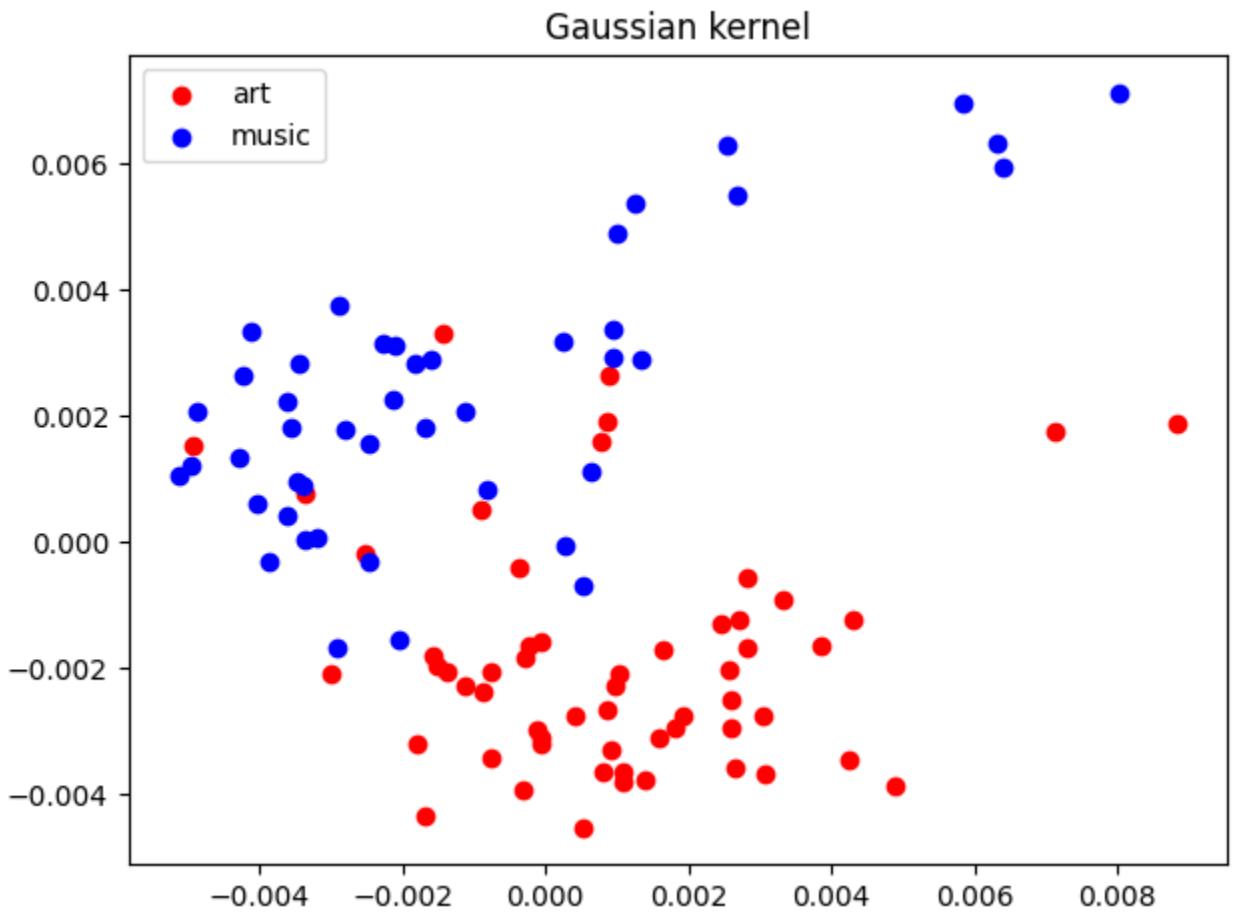

polynomial kernel
art
music
0.008
0.006
0.004
0.002
0.000
−0.002
−0.004
−0.006
−0.0050
−0.0025
0.0000
0.0025
0.0050
0.0075
0.0100



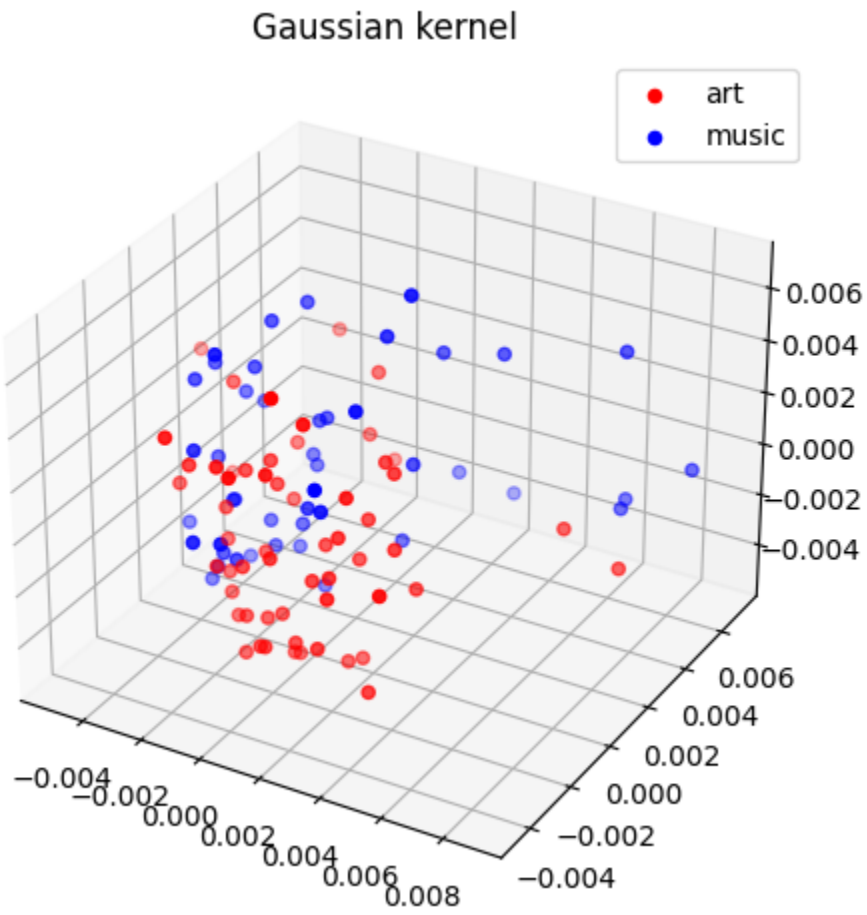
For Gaussian kernel, we have the following figures:



Problem 5 continued on next page. . .



Problem 5 continued on next page...



We can see that for both kernel, using 2-dim can separate the data well. Compared them with HW2, it is easy to see there is nearly no difference between each kernel and linear result except the detail value of PCA scores. I think that is because data martix is a sparse matrix and kernel only have a little affect.