Fundamentals of Machine Learning

Assignment - 5

CS5590

Submitted By

Tushar K Raysad

[BM21MTECH14004]

1 December 2021

1. Hierarchical Clustering:

Given below is the distance matrix for 6 data pointsNon-Uniform Weights in Linear Regression:

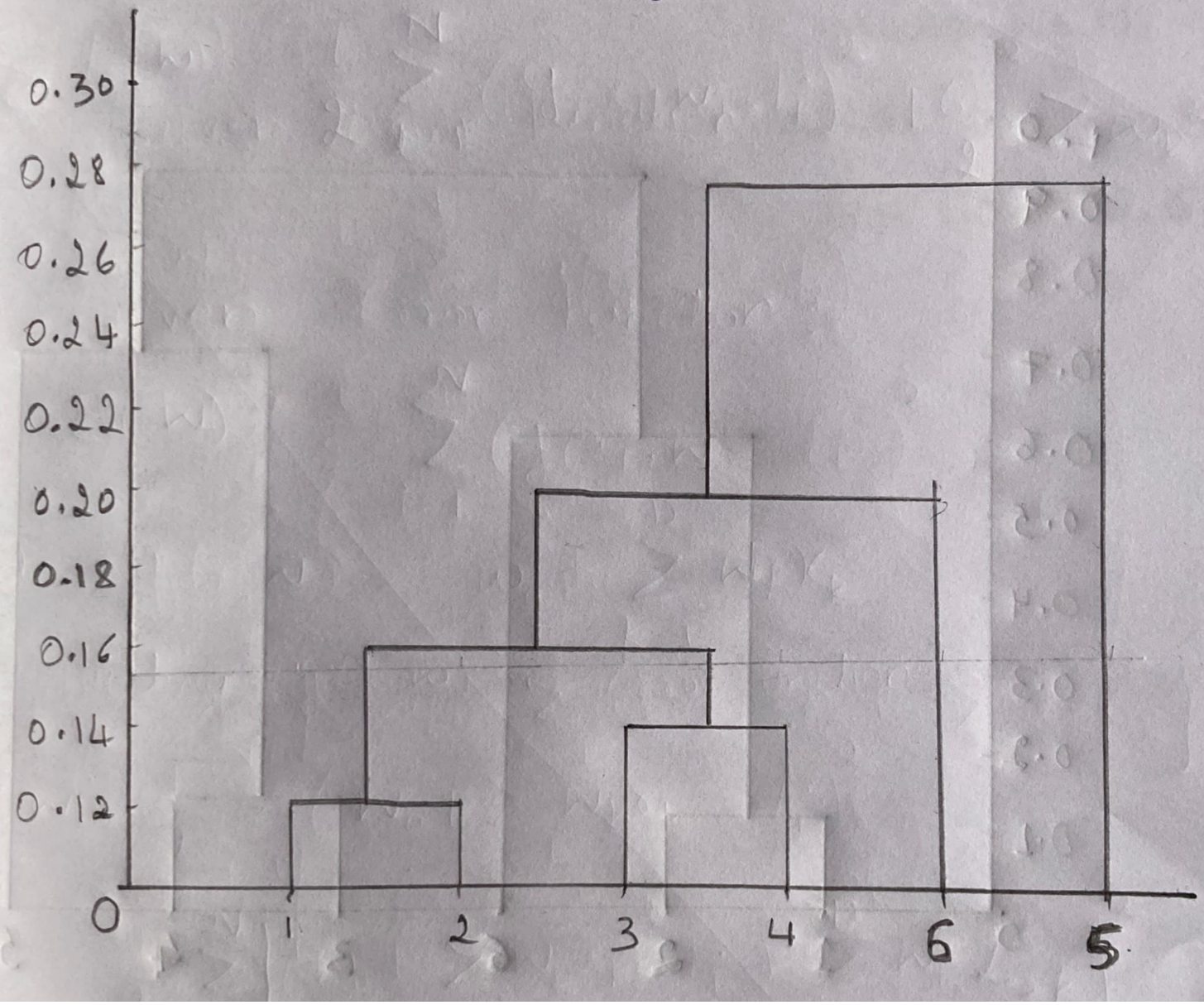
	x_1	x_2	x_3	x_4	x_5	x_6
x_1	0					
x_2	0.12	0				
x_3	0.51	0.25	0			
x_4	0.84	0.16	0.14	0		
x_5	0.28	0.77	0.70	0.45	0	
x_6	0.34	0.61	0.93	0.20	0.67	0

a) Draw a dendrogram for the final result of hierarchical clustering with single link.

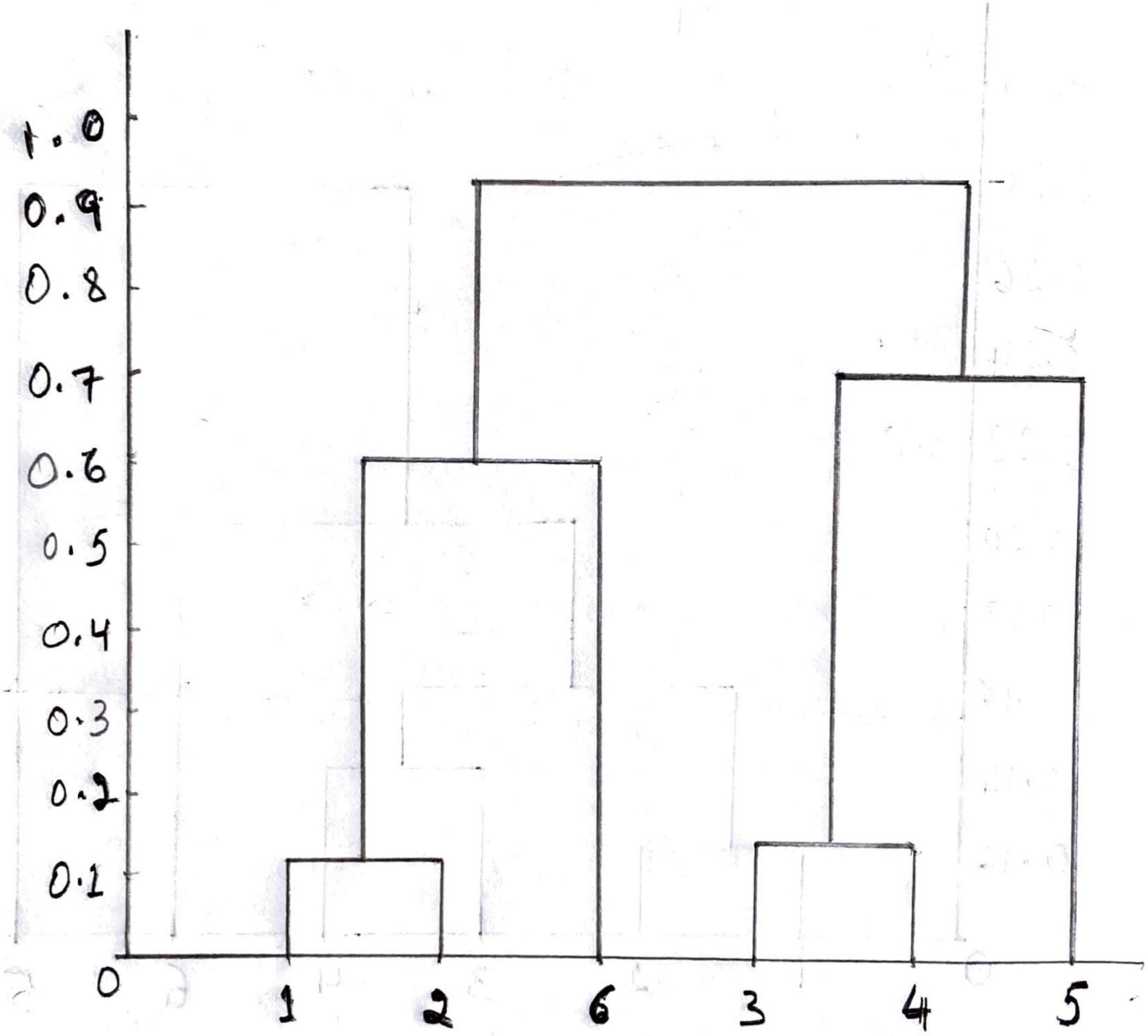
(b) Draw a dendrogram for the final result of hierarchical clustering with complete link.

(c) Change two values from the matrix so that the answer to the last two questions is same.

a] Dendrogram for the final result of heirarchial clustering with single link:



b] Dendrogram for the final result of heirarchial clustering with complete link:



C] The key difference between complete link and single link clustering is where $x_1 x_2$ and x_6 are grouped together by distance$(x_2, x_6) = 0.61$. To make both the clustering links identical we should have dist(x_4, x_4) to be less than 0.61 which may be 0.52.

After this, we can change the value of distance$(x_1 x_2 x_3 x_4, x_6)$ = distance(x_3, x_6) as we should keep this value to be the smallest so that we can group $x_1 x_2 x_3 x_4$ and x_6 together. We can change the value from 0.93 to 0.63.

After changing these two values we can have both clustering with single link and clustering with complete link to be identical.

2. **PCA:** ***(7 marks)***

(a) ***(2 marks)*** Let $\mathbf{X}' = [\mathbf{X_1}, \mathbf{X_2}, ..., \mathbf{X_p}]$ have covariance matrix $\mathbf{\Sigma}$, with eigenvalue-eigen vector pairs $(\lambda_1, e_1), (\lambda_2, e_2), ...(\lambda_p, e_p)$, where $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p \geq 0$.
Let $\mathbf{Y_1} = \mathbf{e_1'X}, \mathbf{Y_2} = \mathbf{e_2'X}, ..., \mathbf{Y_p} = \mathbf{e_p'X}$ be the principal components. Prove that:

$$\sigma_{11} + \sigma_{22} + ... + \sigma_{pp} = \sum_{i=1}^{p} Var(\mathbf{X_i}) = \lambda_1 + \lambda_2 + ... + \lambda_p = \sum_{i=1}^{p} Var(\mathbf{Y_i})$$

(b) ***(5 marks)*** Suppose the random variables $\mathbf{X_1}, \mathbf{X_2}, \mathbf{X_3}$ have covariance matrix

$$\Sigma = \begin{bmatrix} 1 & -2 & 0 \\ -2 & 5 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

It may be verified that the eigenvalue-eigenvector pairs are

$$\lambda_1 = 5.83, \mathbf{e_1'} = [0.383, -0.924, 0]$$
$$\lambda_2 = 2.00, \mathbf{e_2'} = [0, 0, 1]$$
$$\lambda_3 = 0.17, \mathbf{e_3'} = [0.924, 0.383, 0]$$

i. Find out the principal components $\mathbf{Y_1}, \mathbf{Y_2}, \mathbf{Y_3}$. ***(1 mark)***
ii. Do you think $\mathbf{X_3}$ is a principal component? If so, why? ***(0.5 mark)***
iii. Demonstrate $Var(\mathbf{Y_i}) = \lambda_i, i = 1, 2, 3$, and $Cov(\mathbf{Y_i}, \mathbf{Y_k}) = 0, i \neq k$. ***(2 marks)***
iv. Do you think any of the principal components could be ignored/eliminated? Give reasons. ***(1.5 marks)***

→ 9] Given: Σ is a covariance matrix.

Therefore, Sum of diagonal elements = Trace.

So, we can write,

$$\sigma_{11} + \sigma_{22} + \sigma_{33} + \dots \sigma_{pp} = Tr(\Sigma). \quad \text{—— (1)}$$

We have, $\Sigma = PDP^{-1}$ where D → diagonal matrix.

Also we have $P_1 = [e_1, e_2, e_3 \dots e_p]$

We have $PP^{-1} = P^{-1}P = I$.

Therefore,

$$Tr(\Sigma) = Tr(PDP^{-1}) = Tr(DP^{-1}P) = Tr(DI) = Tr(D)$$

$$\therefore Tr(\Sigma) = \lambda_1 + \lambda_2 + \lambda_3 + \dots \lambda_p.$$

Hence from (1),

$$\sigma_{11} + \sigma_{22} + \dots + \sigma_{pp} = \lambda_1 + \lambda_2 + \dots \lambda_p.$$

which gives, $\sum_{i=1}^{p} Var(X_i) = \sum_{i=1}^{p} Var(Y_i)$

Hence proved.

2. b]. For the given data principal components can be written as,

(i) $Y_1 = e_1' x = 0.383 x_1 - 0.924 x_2$

$Y_2 = e_2' x = x_3$

$Y_3 = e_3' x = 0.924 x_1 + 0.383 x_2$

(ii). Yes, x_3 is a principal component. Because x_3 is not correlated with the other two variables i.e., x_1 and x_2.

(iii) $\text{Var}(Y_1) = \text{Var}(0.383 x_1 - 0.924 x_2)$

$= (0.383)^2 (1) + 0.854 (5) - 0.708 (-2)$

$= 5.83$

$\because \text{Var}(Y_1) = \lambda_1$.

$\text{Var}(Y_2) = \text{Var}(x_3) = 2 = \lambda_2$.

$\text{Var}(Y_3) = \text{Var}(0.924 x_1 + 0.383 x_2)$

$= 0.171$

$\text{Var}(Y_3) = \lambda_3$.

And,

$$Cov(Y_1, Y_2) = Cov(0.383x_1 - 0.924x_2, x_3)$$
$$= 0. \quad [\text{As } x_1, x_2 \text{ and } x_3 \text{ are independent.}]$$

$$Cov(Y_2, Y_3) = Cov(x_3, 0.924x_1 + 0.383x_2)$$
$$= 0.$$

$$Cov(Y_1, Y_3) = Cov(0.383x_1 - 0.924x_2, 0.924x_1 + 0.383x_2)$$
$$= 0.$$

iv) The ratio of 1st principal component to total variance $= \frac{\lambda_1}{\lambda_1+\lambda_2+\lambda_3} = 0.73$ — (2)

The ratio of 2nd principal component to total variance
$= \frac{\lambda_2}{\lambda_1+\lambda_2+\lambda_3} = 0.25$ — (3).

The ratio of 3rd principal component to total variance
$= \frac{\lambda_3}{\lambda_1+\lambda_2+\lambda_3} = 0.02$ — (4).

Hence, (1) + (2) = 0.98.

The first two principal components account for 0.98 part of the total population variance. Hence Y_1 and Y_2 could replace the three variables with some amount of loss of information.

3. **EM application:** ***(9 marks)*** Consider the following problem. There are P papers submitted to a machine learning conference. Each of R reviewers reads each paper, and gives it a score indicating how good he/she thought that paper was. We let $x^{(pr)}$ denote the score that reviewer r gave to paper p. A high score means the reviewer liked the paper, and represents a recommendation from that reviewer that it be accepted for the conference. A low score means the reviewer did not like the paper.
   We imagine that each paper has some "intrinsic" true value that we denote by μ_p, where a large value means it's a good paper. Each reviewer is trying to estimate, based on reading the paper, what μ_p is; the score reported $x^{(pr)}$ is then reviewer r's guess of μ_p.
   However, some reviewers are just generally inclined to think all papers are good and tend to give all papers high scores; other reviewers may be particularly nasty and tend to give low scores to everything. (Similarly, different reviewers may have different amounts of variance in the way they review papers, making some reviewers more consistent/reliable than others.)

We let ν_r denote the "bias" of reviewer r. A reviewer with bias ν_r is one whose scores generally tend to be ν_r higher than they should be.
All sorts of different random factors influence the reviewing process, and hence we will use a model that incorporates several sources of noise. Specifically, we assume that reviewers' scores are generated by a random process given as follows:

$$y^{(pr)} \sim \mathcal{N}(\mu_p, \sigma_p^2) \tag{1}$$

$$z^{(pr)} \sim \mathcal{N}(\nu_p, \tau_r^2) \tag{2}$$

$$x^{(pr)}|y^{(pr)}, z^{(pr)} \sim \mathcal{N}(y^{(pr)} + z^{(pr)}, \sigma^2) \tag{3}$$

The variables $y^{(pr)}$ and $z^{(pr)}$ are independent; the variables (x, y, z) for different paper reviewer pairs are also jointly independent. Also, we only ever observe the $x^{(pr)}$'s; thus, the $y^{(pr)}$'s and $z^{(pr)}$'s are all latent random variables.

We would like to estimate the parameters μ_p, σ_p^2, ν_r, τ_r^2 If we obtain good estimates of the papers' "intrinsic values" μ_p these can then be used to make acceptance/rejection decisions for the conference.

We will estimate the parameters by maximizing the marginal likelihood of the data $\{x^{(pr)}; p = 1, ..., P, r = 1, ..., R\}$ This problem has latent variables $y^{(pr)}$ and $z^{(pr)}$, and the maximum likelihood problem cannot be solved in closed form. So, we will use EM. Your task is to derive the EM update equations. Your final E and M step updates should consist only of addition/subtraction/multiplication/division/log/exp/sqrt of scalars; and addition/subtraction/multiplication/inverse/determinant of matrices. For simplicity, you need to treat only $\{\mu_p, \sigma_p^2; p = 1, ..., P\}$ and $\{\nu_r, \tau_r^2; r = 1, ..., R\}$ as parameters. I.e. treat σ^2 (the conditional variance of $x^{(pr)}$ given $y^{(pr)}$ and $z^{(pr)}$) as a fixed, known constant.

(a) we will derive the E-step:
   i. The joint distribution $p(y^{(pr)}, z^{(pr)}, x^{(pr)})$ has the form of a multivariate Gaussian density. Find its associated mean vector and co-variance matrix in terms of the parameters μ_p, σ_p^2, ν_r, τ_r^2 and σ^2.
   ii. Derive an expression for $Q_{pr}(y^{(pr)}, z^{(pr)}) = p(y^{(pr)}, z^{(pr)}|x^{(pr)})$ (E-step), using the rules for conditioning on subsets of jointly Gaussian random variables

(b) Derive the M-step updates to the parameters { μ_p, σ_p^2, ν_r, τ_r^2 }. [Hint: It may help to express the lower bound on the likelihood in terms of an expectation with respect to $(y^{(pr)}, z^{(pr)})$ drawn from a distribution with density $Q_{pr}(y^{(pr)}, z^{(pr)})$]

⟶ Let us take ψ as the whole set of parameters that we are estimating on. The methods for the problem are:

a] E-step :-

For each p, r set

$$Q_{pr}(y^{(pr)}, z^{(pr)}) = P(y^{(pr)}, z^{(pr)} / x^{(pr)}; \psi).$$

b] M-step :-

$$\text{Set } \psi = \text{argmax}_{\psi} \sum_{p=1}^{P} \sum_{r=1}^{R} E_{Q_{pr}(Y^{(pr)}, Z^{(pr)})} \log p(x^{(pr)}, Y^{(pr)}, Z^{(pr)}; \psi)$$

a] For E-step,

If we ought to use Baye's rule to compute $p(y^{(pr)}, z^{(pr)} / x^{(pr)})$, then it will be very difficult to compute integrals of Guassian Denominator.

Therefore,

$$p(y^{(pr)}, z^{(pr)}, x^{(pr)}) = p(y^{(pr)}, z^{(pr)}) p(x^{(pr)} / y^{(pr)}, z^{(pr)})$$

$$= p(y^{(pr)}) p(z^{(pr)}) p(x^{(pr)} / y^{(pr)}, z^{(pr)}).$$

As the above equation is product of 3 Guassian densities, it is multivariate Guassian density. So, the distribution $p(y^{(pr)}, z^{(pr)}, x^{(pr)})$ is one type of normal distribution so we can use it for computing the conditional. To get a form for joint density

we will use that Gaussian density is fully parametrized by its mean vector and co-variance matrix.

To compute mean vector, we have

$x^{(pr)} = y^{(pr)} + z^{(pr)} + e^{(pr)}$

where $e^{(pr)} \sim N(0, \sigma^2)$ - independent Gaussian noise.

Then $E[y^{(pr)}] = \mu_p$, $E[z^{(pr)}] = \nu_r$.

So, $E[x^{(pr)}] = E[y^{(pr)} + z^{(pr)} + e^{(pr)}]$

$= E[y^{(pr)}] + E[z^{(pr)}] + E[e^{(pr)}]$

$= \mu_p + \nu_r + 0 = \mu_p + \nu_r$.

To compute co-variance matrix, we have

$Var(y^{(pr)}) = \sigma_p^2$, $Var(z^{(pr)}) = \tau_r^2$

$\therefore Cov(y^{(pr)}, z^{(pr)}) = Cov(z^{(pr)}, y^{(pr)}) = 0$. [as $y^{(pr)}$ and $z^{(pr)}$ are independent.]

Therefore,

$Var(x^{(pr)}) = Var(y^{(pr)}) + Var(z^{(pr)}) + Var(e^{(pr)})$

$= \sigma_p^2 + \tau_r^2 + \sigma^2$.

So, $Cov(y^{(pr)}, x^{(pr)}) = Cov(x^{(pr)}, y^{(pr)})$

$= Cov(y^{(pr)} + z^{(pr)} + e^{(pr)}, y^{(pr)})$.

$= Cov(y^{(pr)}, y^{(pr)}) + Cov(z^{(pr)}, y^{(pr)}) + Cov(e^{(pr)}, y^{(pr)})$

$= \sigma_p^2 + 0 + 0 = \sigma_p^2$.

Similarly we can show that

$$Cov(z^{(pr)}, x^{(pr)}) = Cov(x^{(pr)}, z^{(pr)}) = \tau_r^2.$$

Therefore, we can write,

$$y^{(pr)}, z^{(pr)}, x^{(pr)} \sim \mathcal{N}\left(\begin{bmatrix} \mu_p \\ \nu_r \\ \mu_p + \nu_r \end{bmatrix}, \begin{bmatrix} \sigma_p^2 & 0 & \sigma_p^2 \\ 0 & \tau_r^2 & \tau_r^2 \\ \sigma_p^2 & \tau_r^2 & \sigma_p^2 + \tau_r^2 + \sigma^2 \end{bmatrix}\right)$$

Therefore from the Factor Analysis we can use standard results for conditioning on subsets of variables for Gaussians. Thus we can write:

$$Q_{pr}(y^{(pr)}, z^{(pr)}) = \mathcal{N}\left(\begin{bmatrix} \mu_{pr,Y} \\ \mu_{pr,Z} \end{bmatrix}, \begin{bmatrix} \Sigma_{pr,YY} & \Sigma_{pr,YZ} \\ \Sigma_{pr,ZY} & \Sigma_{pr,ZZ} \end{bmatrix}\right)$$

where,

$$\mu_{pr} = \begin{bmatrix} \mu_{pr,Y} \\ \mu_{pr,Z} \end{bmatrix} = \begin{bmatrix} \mu_p + \dfrac{\sigma_p^2}{\sigma^2 + \sigma_p^2 + \tau_r^2}\left(x^{(pr)} - \mu_p - \nu_r\right) \\ \nu_r + \dfrac{\tau_r^2}{\sigma^2 + \sigma_p^2 + \tau_r^2}\left(x^{(pr)} - \mu_p - \nu_r\right) \end{bmatrix} \quad \text{---- (1)}$$

and

$$\Sigma_{pr} = \begin{bmatrix} \Sigma_{pr,YY} & \Sigma_{pr,YZ} \\ \Sigma_{pr,ZY} & \Sigma_{pr,ZZ} \end{bmatrix} = \frac{1}{\sigma^2 + \sigma_p^2 + \tau_r^2} \begin{bmatrix} \left(\sigma_p^2(\tau_r^2 + \sigma^2)\right) & -\sigma_p^2\tau_r^2 \\ -\sigma_p^2\tau_r^2 & \left(\tau_r^2(\sigma_p^2 + \sigma_2^2)\right) \end{bmatrix}$$

---- (2)

b] For M-step,

We have Q_{pr} distribution defined in terms of ψ^t, while we want to choose parameters for the next step i.e., ψ^{t+1}. Thus parameters of the Q_{pr} distribution are constant in terms of the parameter we wish to maximize. Maximizing the expected log likelihood.

We have,

$$\psi = \text{argmax}_\psi \sum_{p=1}^{P}\sum_{r=1}^{R} E_Q \log p(x^{(pr)}, y^{(pr)}, z^{(pr)}; \psi)$$

$$= \text{argmax}_\psi \sum_{p=1}^{P}\sum_{r=1}^{R} E_Q \log \left[\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(x^{(pr)} - y^{(pr)} - z^{(pr)})^2} \cdot \frac{1}{\sqrt{2\pi}\sigma_p} e^{-\frac{1}{2\sigma_p^2}(y^{(pr)} - \mu_p)^2} \cdot \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\tau_r^2}(z^{(pr)} - \nu_r)^2} \right]$$

$$= \text{argmax}_\psi \sum_{p=1}^{P}\sum_{r=1}^{R} \left[E_Q \left[\log \frac{1}{(2\pi)^{3/2}\sigma\sigma_p\tau_r} - \frac{1}{2\sigma^2}(x^{(pr)} - y^{(pr)} - z^{(pr)})^2 - \frac{1}{2\sigma_p^2}(y^{(pr)} - \mu_p)^2 - \frac{1}{2\tau_r^2}(z^{(pr)} - \nu_r)^2 \right] \right]$$

$$= \text{argmax}_\psi \sum_{p=1}^{P}\sum_{r=1}^{R} E_Q \left[\log \frac{1}{\sigma_p\tau_r} - \frac{1}{2\sigma_p^2}(y^{(pr)} - \mu_p)^2 - \frac{1}{2\tau_r^2}(z^{(pr)} - \nu_r)^2 \right]$$

$$= \arg\max_{\psi} \sum_{p=1}^{P}\sum_{r=1}^{R} E_Q\left[\log \frac{1}{\sigma_p \tau_r} - \frac{1}{2\sigma_p^2}\left((y^{(pr)})^2 - 2y^{(pr)}\mu_p + \mu_p^2\right) - \frac{1}{2\tau_r^2}\left((z^{(pr)})^2 - 2z^{(pr)}\nu_r + \nu_r^2\right)\right]$$

$$= \arg\max_{\psi} \sum_{p=1}^{P}\sum_{r=1}^{R}\left[\log \frac{1}{\sigma_p \tau_r} - \frac{1}{2\sigma_p^2}\left(E_Q[(y^{(pr)})^2] - 2E_Q[y^{(pr)}]\mu_p + \mu_p^2\right) - \frac{1}{2\tau_r^2}\left(E_Q[(z^{(pr)})^2] - 2E_Q[z^{(pr)}]\nu_r + \nu_r^2\right)\right]$$

$$= \arg\max_{\psi} \sum_{p=1}^{P}\sum_{r=1}^{R}\left[\log \frac{1}{\sigma_p \tau_r} - \frac{1}{2\sigma_p^2}\left(\Sigma_{pr,yy} + \mu_{pr,y}^2 - 2\mu_{pr,y}\mu_p + \mu_p^2\right) - \frac{1}{2\tau_r^2}\left(\Sigma_{pr,zz} + \mu_{pr,z}^2 - 2\mu_{pr,z}\nu_r + \nu_r^2\right)\right]$$

[As $E_Q[y^{(pr)}] = \mu_{pr,y}$ and $E_Q[(y^{(pr)})^2]$

$= \left(E_Q[(y^{(pr)})^2] - E_Q[y^{(pr)}]^2\right) + E_Q[y^{(pr)}]^2$

$= \Sigma_{pr,yy} + \mu_{pr,y}^2$. and Similarly for $E_Q[z^{(pr)}]$ and $E_Q[(z^{(pr)})^2]$.]

Getting the derivatives w.r.t parameters $\mu_p, \nu_r, \sigma_p, \tau_r$ to 0.

We get,

$$-\frac{1}{2\sigma_p^2}\sum_{r=1}^{R}(2\mu_p - 2\mu_{pr,Y}) = 0 \Rightarrow \mu_p = \frac{1}{R}\sum_{r=1}^{R}\mu_{pr,Y} \quad \text{—(3)}$$

$$-\frac{1}{2\tau_r^2}\sum_{p=1}^{P}(2\nu_r - 2\mu_{pr,Z}) = 0 \Rightarrow \nu_r = \frac{1}{P}\sum_{p=1}^{P}\mu_{prZ} \quad \text{—(4)}$$

$$\sum_{r=1}^{R}\left[-\frac{1}{\sigma_p} + \frac{1}{\sigma_p^3}\left(\Sigma_{pr,YY} + \mu_{pr,Y}^2 - 2\mu_{pr,Y}\mu_p + \mu_p^2\right)\right] = 0$$

$$\Rightarrow \sigma_p^2 = \frac{1}{R}\sum_{r=1}^{R}\left(\Sigma_{pr,YY} + \mu_{pr,Y}^2 - 2\mu_{pr,Y}\mu_p + \mu_p^2\right) \quad \text{—(5)}$$

$$\sum_{p=1}^{P}\left[-\frac{1}{\tau_r} + \frac{1}{\tau_r^3}\left(\Sigma_{pr,ZZ} + \mu_{pr,Z}^2 - 2\mu_{pr,Z}\nu_r + \nu_r^2\right)\right] = 0.$$

$$\Rightarrow \tau_r^2 = \frac{1}{P}\sum_{p=1}^{P}\left(\Sigma_{pr,ZZ} + \mu_{pr,Z}^2 - 2\mu_{pr,Z}\nu_r + \nu_r^2\right)$$

— (6).

Using the results (1), (2), (3), (4), (5) and (6) we can restate E and M steps in terms of actual computations.

a) For each p, r compute μ_{pr}, Σ_{pr} using (1) & (2).

b) For M-step compute μ_p, ν_r, σ_p^2, τ_r^2 using (3), (4), (5) and (6).

Questions: Programming

1. **Clustering *(7 marks)*:** DBSCAN, as we discussed in class, is a density-based clustering algorithm. In this problem, you need to implement your own DBSCAN algorithm. You can read more about it from paper that proposed this method [link].

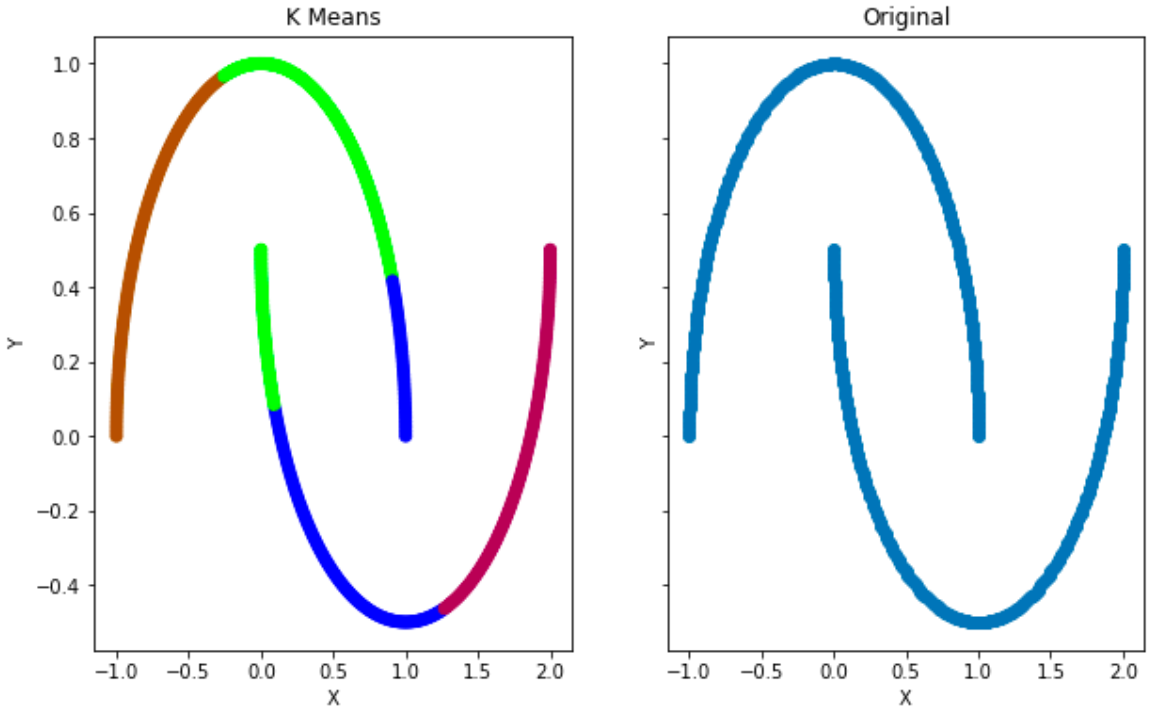
   (a) Use the Kmeans clustering algorithm from sklearn and find the number of clusters in dataset1 shared with you. Plot the data points with different colors for different clusters. [*1 mark*]

   (b) Implement your own DBSCAN algorithm on the same dataset and plot the data points. [*3 marks*]

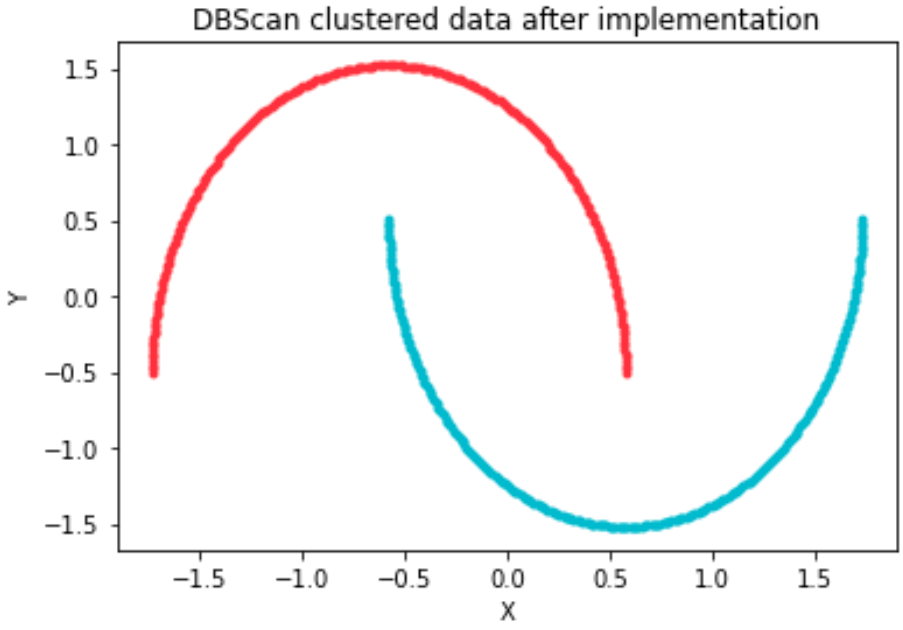
   (c) What differences do you see between the DBSCAN and k-means methods, and why? [*1 mark*]

   (d) Consider the dataset2 (also shared with you) with three clusters. Use (a) and (b) for dataset2, and compare the performance. List your observations clearly, and make conclusions on pros and cons of DBSCAN and k-means. [*2 marks*]

a] K-means clustering algorithm from sklearn for the dataset1 gave the following plot:



b] DBScan algorithm on dataset1 gave following plot:



c] Differences between K-means and DBScan:

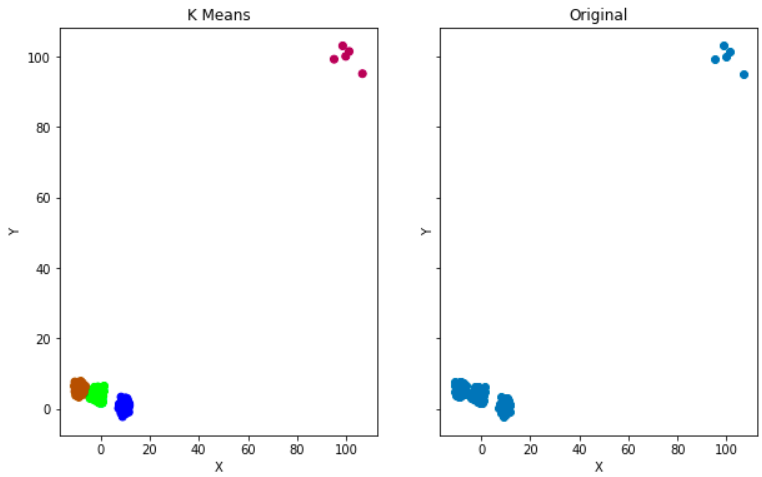
- The Key difference between K-means and DBScan algorithm is that K-means algorithm is a centroid based or partition based clustering algorithm while DBScan is a density based clustering technique.

- The clusters formed in this algorithm are more or less spherical or convex in shape while in DBScan clusters formed are arbitrarily shaped and may not have feature size.

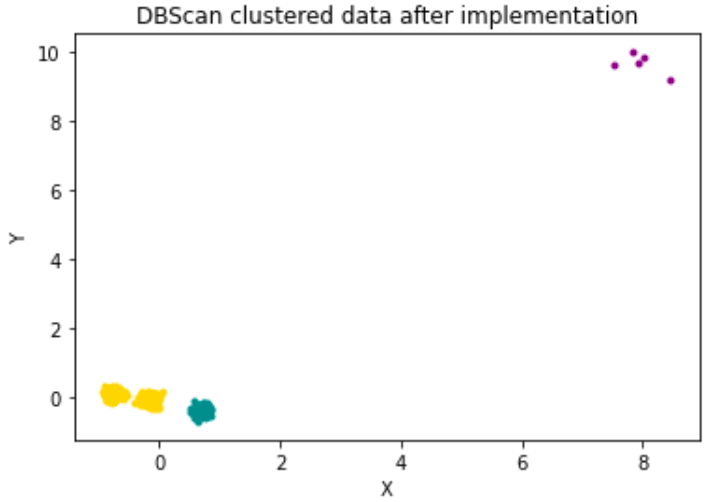
- K-means algorithm doesn't work well with outliers and noisy data while DBScan works with good efficiency.

- No prior knowledge of numbers of clusters is required for DBScan whereas it is required for K-means.

d] K-means clustering algorithm from sklearn for the dataset2 gave the following plot:



DBScan algorithm on dataset1 gave following plot:



Pros of K-means and DBScan:

1.K-means works faster compared to DBScan

2. There is no need to specify the number of clusters for DBScan.

Cons of K-means and DBScan:

1. Dbscan doesn't work well with varying cluster densities.

2. K-means require specification of number of clusters.

3. DBScan's parameters should be carefully selected.

2. **t-SNE:** ***(3 marks)*** In this question, use the handwritten digits dataset which has images of size 8×8 (64 dimensional). Use sklearn's t-SNE algorithm and reduce this 64-dimensional data to 2 dimensions (2-D).

   - For dataset, you can use 'load_digits' from sklearn.datasets
   - You can use numpy, scipy for any mathematical operations and seaborn for scatterplot
   - Implement the algorithm taking perplexity = 30 and number of iterations = 1000
   - It is recommended to take degrees of freedom as (dimensions of reduced space - 1) which is 2-1 = 1 here.
   - Visualize the reduced 2-D data as a scatter plot

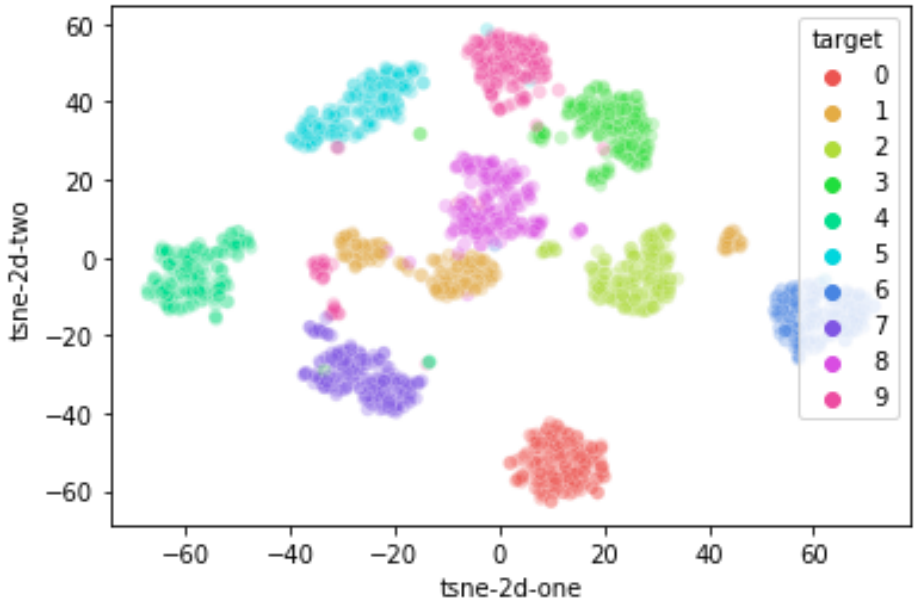
   (a) Fix perplexity to 30, repeat the algorithm with number of iterations = 100 and 2000. Observe the scatter plots. Write your observation on how scatter plots varies with no. of iterations (100, 1000 and 2000) and give reasons. (For example, if you say that experiments with no. of iterations = 1000 and 2000 produces the same results, give reasons on why you think that happens and so on)

   (b) It is observed that, for some datasets, different runs of t-SNE algorithm with the same hyperparameters produce different results. Why do you think it happens?

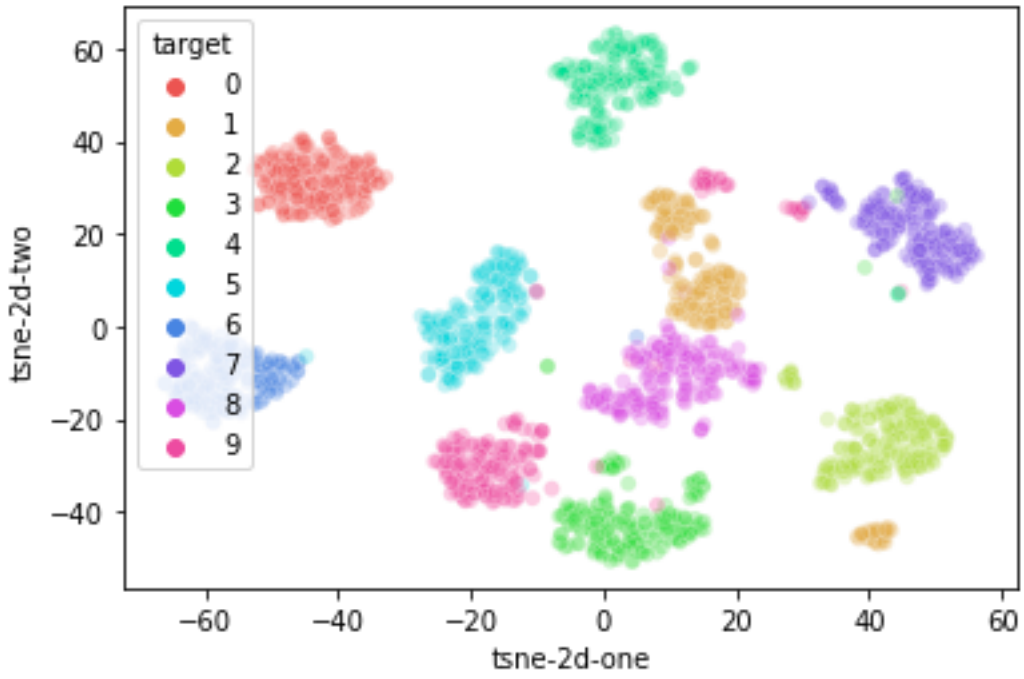
   Deliverables:

   - Code (Preferably an ipynb file)
   - Brief report (PDF) with your answers to the above questions

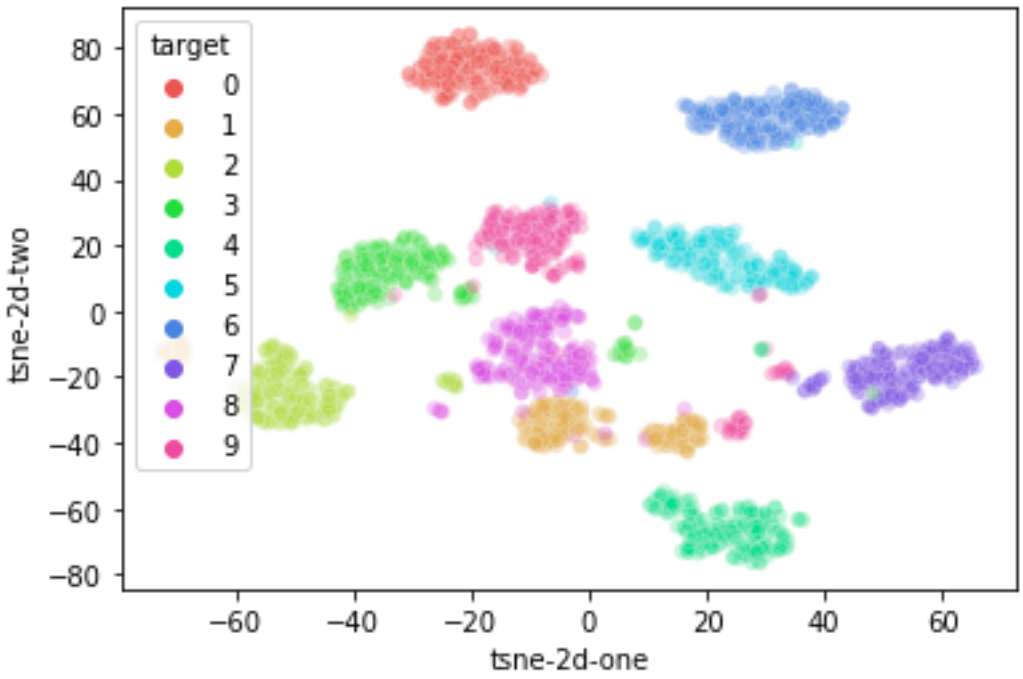
a] The plot t-SNE algorithm gave for Number of iterations =250, Perplexity =30 :



The plot t-SNE algorithm gave for Number of iterations =1000, Perplexity =30 :

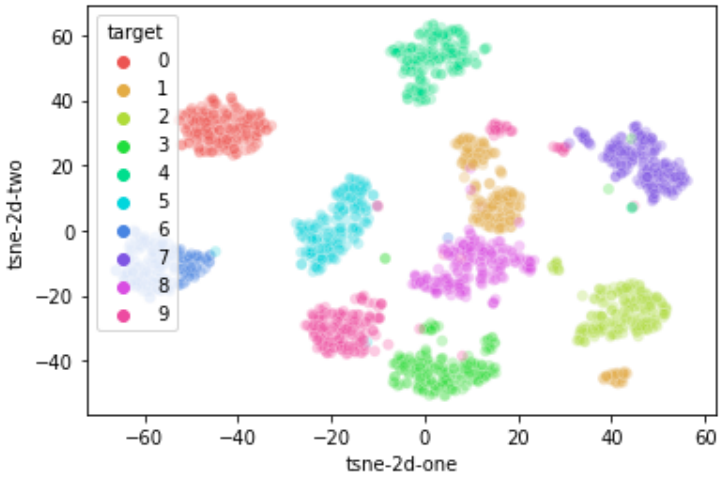


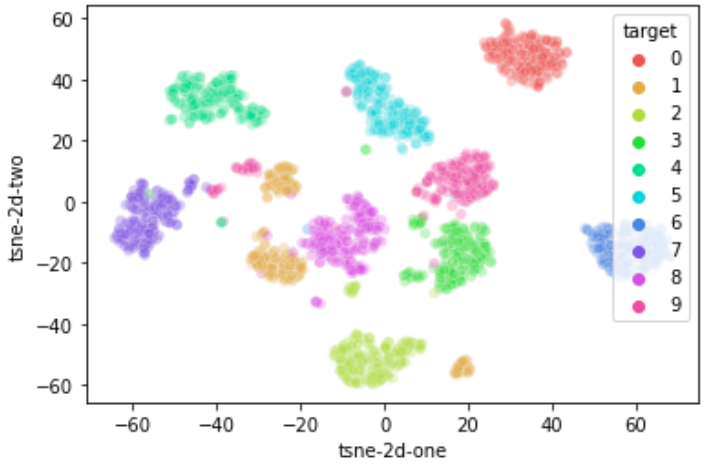
The plot t-SNE algorithm gave for Number of iterations =2000, Perplexity =30 :



- t-SNE is an unsupervised, randomised algorithm, used only for visualisation. It preserves the local structure of the data using student t-distribution to compute the similarity between two points in lower-dimensional space. t-SNE uses a heavy-tailed Student-t distribution to compute the similarity between two points in the low-dimensional space rather than a Gaussian distribution, which helps to address the crowding and optimization problems.In the digits dataset, t-SNE separated clusters of each digit class. And we get different plots for different number of iterations in t-SNE.

b] Two different plots obtained for the same hyper parameters when compiled more than once :





- The Reason for obtaining different plots for same hyper parameters when compiled more than once:

  t-SNE has a non-convex objective function. The objective function is minimized using a gradient descent optimization that is initiated randomly. As a result, it is possible that different runs give you different solutions.