Part I: Clustering Algorithm

Problem 1

Steps for KMeans algorithm

Solution Steps for KMeans algorithm

1. Choose the number of clusters k
2. Select k random points from the data as centroids
3. Assign all the points to the closest cluster centroid
4. Recompute the centroids of newly formed clusters
5. Repeat step 3 and step 4 until one of of those stopping conditions met:
    - Maximum number of iterations is reached
    - Newly formed clusters do not change compared to the previous loop (or the centroids of newly formed clusters do not change)

Problem 2

Given the pairs of (x, y) with coordinates as follows: (0.5; 0.5), (1; 0.5), (1; 1.5), (1.5; 1), (2.6; 2), (3; 2), (2.4; 2.5), (2.5; 3).

- Apply the k-means algorithm with 2 centroids.
- Initialize the centroid centers with random coordinates, run the k-means algorithm for 3 iterations, and return the coordinates of the 2 centroids.

Solution Two randomly initialized centroids: (0.5, 0.5) and (1, 1.5)

- Loop 1:
    + Calculate the distance between all points and each centroid

Centroid	(0.5, 0.5)	(1, 1.5)
(0.5, 0.5)	0	1.118
(1, 0.5)	0.5	1
(1, 1.5)	1.118	0
(1.5, 1)	1.118	0.707
(2.6, 2)	2.58	1.676
(3, 2)	2.915	2.06
(2.4, 2.5)	2.7586	1.72
(2.5, 3)	3.2	2.121

    + Newly formed clusters and updated centroids:

Cluster	Old Centroids	Data points	New Centroids
1	(0.5, 0.5)	(0.5, 0.5), (1, 0.5)	(0.75, 0.5)
2	(1, 1.5)	(1, 1.5), (1.5, 1), (2.6, 2), (3, 2), (2.4, 2.5), (2.5, 3)	(2.2, 2)

- Loop 2:

+ Calculate the distance between all points and each centroid

Centroid	(0.75, 0.5)	(2.2, 2)
(0.5, 0.5)	0.25	2.267
(1, 0.5)	0.25	1.92
(1, 1.5)	1.03	1.3
(1.5, 1)	0.9	1.22
(2.6, 2)	2.38	0.4
(3, 2)	2.7	0.8
(2.4, 2.5)	2.59	0.54
(2.5, 3)	3.05	1.044

+ Newly formed clusters and updated centroids:

Cluster	Old Centroids	Data points	New Centroids
1	(0.75, 0.5)	(0.5, 0.5), (1, 0.5), (1, 1.5), (1.5, 1)	(1, 0.875)
2	(2.2, 2)	(2.6, 2), (3, 2), (2.4, 2.5), (2.5, 3)	(2.625, 2.375)

- Loop 3:
  + Calculate the distance between all points and each centroid

Centroid	(1, 0.875)	(2.625, 2.375)
(0.5, 0.5)	0.625	2.834
(1, 0.5)	0.375	2.48
(1, 1.5)	0.625	1.845
(1.5, 1)	0.515	1.7766
(2.6, 2)	1.956	0.376
(3, 2)	2.2947	0.53
(2.4, 2.5)	2.145	0.257
(2.5, 3)	2.6	0.637

  + Newly formed clusters and updated centroids:

Cluster	Old Centroids	Data points	New Centroids
1	(1, 0.875)	(0.5, 0.5), (1, 0.5), (1, 1.5), (1.5, 1)	(1, 0.875)
2	(2.625, 2.375)	(2.6, 2), (3, 2), (2.4, 2.5), (2.5, 3)	(2.625, 2.375)

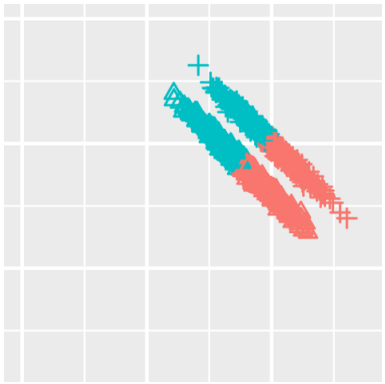
After 3 iteration, the coordinates of the two centroids are (1, 0.875) and (2.625, 2.375).

Problem 3

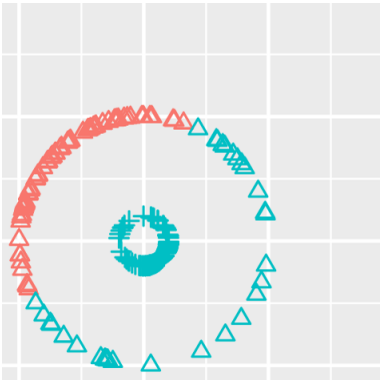
- Draw 3 different scenarios with datasets where clustering poorly performs when using k-means. Explain the reasons.
- Propose a suitable algorithm for that dataset and explain why that algorithm is appropriate.

Solution Examples of bad clustering using Kmeans.

Case 1: different sizes



Case 2: non-spherical shapes



Case 3: coincident means

Recommended clustering algorithm for each case:

- Case 1: GMM (can handle data with varying sizes)
- Case 2: GMM (can handle non-spherical data, as long as it has elliptical shape)
- Case 3: DBScan (density based methods, no assumption for data shape and size)

Problem 4

GMM Algorithm:

- Write the Loss function for this algorithm.
- From the loss function, indicate the approach of the GMM algorithm to optimize this loss (E-M in slides). Explain the significance of the E-step and M-step and what they are used for (Using mathematical formulas).

Solution Loss function for GMM algorithm:

$$l(\theta) = \sum_{n=1}^{N} log \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k) \right)$$

To optimize this loss function, we will take the derivative of $l(\theta)$ with respect to each parameter and set them equal to 0. Then we have:

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n$$

$$\Sigma_k = \frac{1}{N} \sum_{n-1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T$$

$$\pi_k = \frac{N_k}{N}$$

Notice that all three parameters depend on $\gamma(z_{nk}) \rightarrow$ iterative scheme with EM (Expectation - Maximization) algorithm can be used.

- E-step: Compute the probability each data point belongs to each Gaussian (cluster)

$$\gamma(z_{nk}) = \frac{p(z_{nk} = 1) p(x_n | z_{nk} = 1}{\sum_{j=1}^{K} p(z_{nj} = 1) p(x_n | z_{nj} = 1)}$$

- M-step: Re-estimate parameters (μ_k, Σ_k, π_k) of each cluster using $\gamma(z_{nk})$.

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n$$

$$\Sigma_k = \frac{1}{N} \sum_{n-1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T$$

$$\pi_k = \frac{N_k}{N}$$

Problem 5

Steps for GMM algorithm

Solution

1. Define number of clusters
2. Initialize parameters, which include means (μ_k), covariances (Σ_k) and mixing coefficients (π_k)
3. EM (Expectation - Maximization) algorithm:
    - E-step: Compute the probability each data point belongs to each Gaussian (cluster)
    - M-step: update parameters (μ_k, Σ_k, π_k) of each cluster based on the data point assignment in E-step. Maximize Likelihood.
4. Repeat step 2 until reaching max iteration or convergence (if the parameter or the log-likelihood does not change significantly)

Problem 6

Draw a comparison table between k-means and DBSCAN for each aspect.

Solution Comparison table:

Aspects	KMeans	DBScan
Based idea	distance based	density based
Type	hard cluster	soft cluster
Number of clusters	required to be defined beforehand	automatically determined based on density
Outliers handling	sensitive to outliers	flexible when dealing with outliers by assigning them as noise points
Complexity	lower	higher
Assumptions of data	spherical shape and equal size	arbitrary shapes and sizes
Predefined parameters	number of clusters	radius and min points
Initialization	requires initialization of centroids, which can affect the performance of the algorithm	no initialization required

Problem 7

Explain how DBSCAN operates based on your understanding (you may use additional mathematics to explain or not)

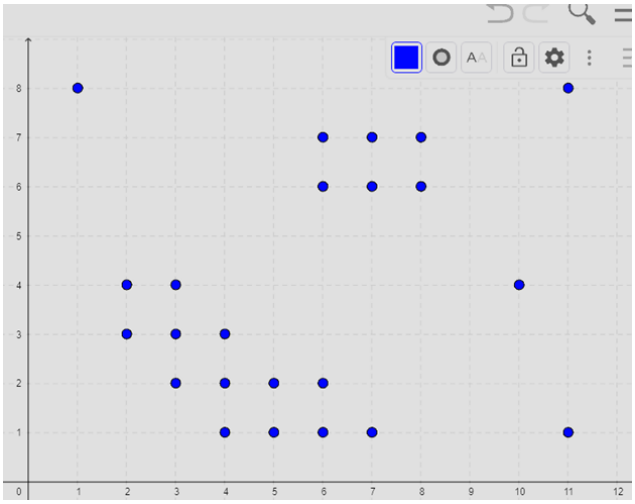
Solution

1. Define the radius (Eps) and the minimum points (MinPts) in an Eps-neighbourhood of a point so that it can be classified as a core point.
2. For every point p with null label:
    - Draw a circle with Eps radius of around this data point.
    - If the density round p (the number of point within the radius Eps) is less than MinPts, it is not a core point, assign a null label to it.
    - If the density around p reach the minimum point MinPts, p is a core point and a new cluster is formed. Assign that new cluster label for p and all the point in its density. Then, from p, find all the point density reachable and put them in the same cluster with p. (Put all the point in the Eps-neighborhood of p in a queue, going through each point, draw a circle around it, put their neighbours in p's cluster and define their type. Keep putting their Eps-neighbourhood points in the queue if they are core points and repeat that process until no more point of p's cluster can be found) *Notes: Reassign the zero labels but not the others*
3. Repeat step 2 until all the points have been visited

4. Assign noise points type to all zero labels after step 3.

Problem 8

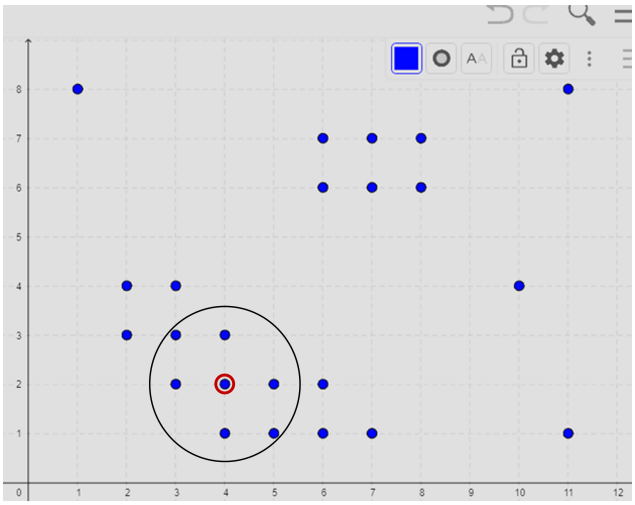
Given some data points in the following Oxy coordinate:



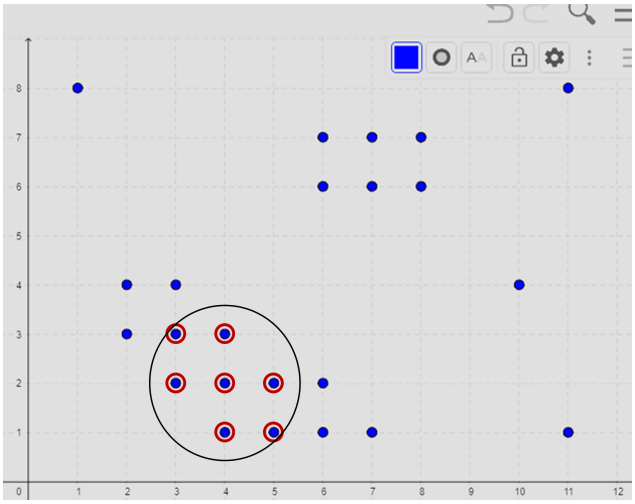
Apply DBScan algorithm with radius (Eps) = 1.5 and min points (MinPts) = 3. Determine possible clusters and noise points.

Solution Given: Eps = 1.5 and MinPts = 3

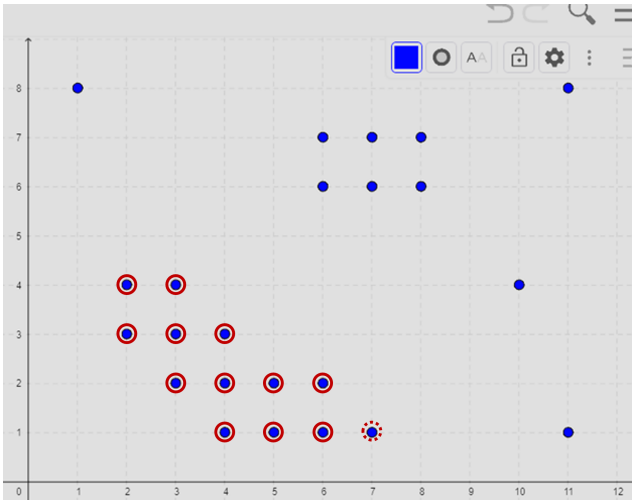
1. For every null label point:
    - Draw a circle around them



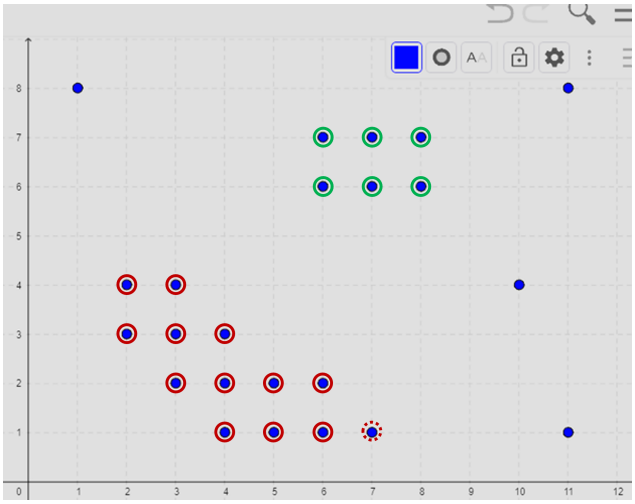
    - Leave it as null label if density less than MinPts, otherwise, it is a core point.
    - If it is a core point, a new cluster it form.



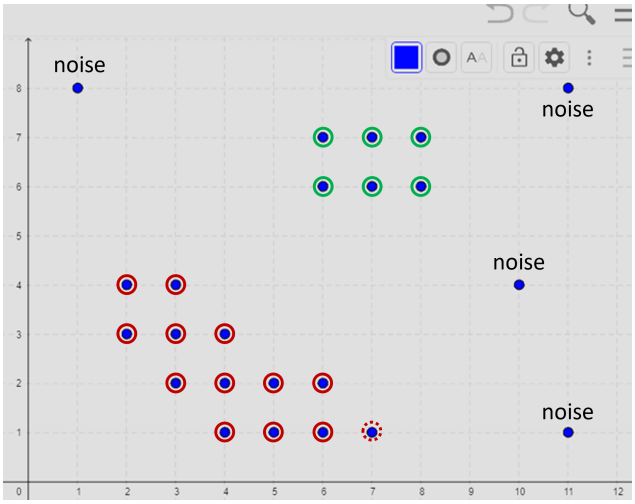
Then find all points density-reachable from p and classify them in the cluster. (Solid line represents core points, dot line represents boder points)



2. Repeat step 2 until all the points have been visited. Different colors represent different clusters.



3. Assign noise point type to all null label points left after step 2.



Part II: Dimensionality Reduction Algorithm

Problem 1

What are the steps of PCA? Give your reason(s) for each step if possible.

Solution

Steps	Actions	Reasons
1	standardization by subtracting the mean and diving by the standard deviation for each feature	to make the data unit free (all features are in the same scale) and avoid biased results
2	Compute the covariance matrix of the data	to see if there is any relationship between features
3	Calculate the eigenvalues and eigenvectors of the covariance matrix	to identify the principal components (eigenvectors are the direction of the axes where there is most variance, or principal components, and eigenvalues give the amount of variance carried in each PC)
4	Select the number of principal components based on the explained variance captured in each component	The first PC captures most information, then the second and so on. Hence, the optimal number of components needed to be chosen to avoid overfitting while still capturing as much variance as possible
5	Data transformation in new dimensional space by multiplying the original data by the selected number of previously computed eigenvectors	to obtain newly low-dimensional data

Problem 2

How to choose the the number of dimensions to reduce when building PCA?

Solution To choose the number of dimensions to reduce when building PCA, we can draw and look at two types of chart:

- A scree plot that illustrates the amount of explained variance captured in each principal component
- An elbow curve that shows the cumulative explained variance as the number of principal components increase

After that, choosing the number of dimensions might depend on specific purposes:

- For example, if you want to keep 90% variance of the data, choose the number of components that keep those amount based on the elbow curve. In common, people often choose to keep about 90 to 90% variance from the dataset.

- Another instance when not having any idea, the optimal number of components can be defined by estimating the trade-off point on the elbow curve (trade-off between information loss and number of dimensions after reducing).

Problem 3

What are the limitations of PCA?

Solution

- PCA assumes linear relationships between variables (data contains variability)
- The trade-off between information loss and the number of dimensions reduced
- Difficult to tell which are the most important features of the original data after computing principal components.
- ...

Problem 4

Why do we have to use t-distribution in t-SNE?

Solution

- Crowding problem: High dimensional space have more rooms than low dimensional one. Hence, after reducing dimensions, sometimes there might be not enough room to accommodate all neighbors.
- t-distribution also has bell shape as gaussian distribution does but with longer tail, so that by using t-distribution in t-SNE, far points can also be calculated probability.

Problem 5

Reconstruct PCA.

Solution

1. Define the problem: Why is PCA (or dimensionality reduction in general) necessary?
    - Reduce the size of the data:
        + Better for storage
        + Better for visualization
        + Reduce computaional cost and the training time
        + Reduce the complexity of the model $\rightarrow$ improve model performance
    - Removing irrelevant features, capturing the most important ones
        + Reduce overfitting
        + The new data contains more concise information, making it easier to compare.
2. Idea of PCA:
    - Transform correlated data into uncorrelated data to construct newly relevant features from the dataset
    - Try to maximize variance to maintain as much information as possible
3. Steps for PCA:
    - Standardize the data
    - Compute covariance matrix of the data
    - Calculate eigenvalues and eigenvectors of the covariance matrix to identify the principal components
    - Choose the number of components
    - Transform the data to obtain low dimensional data