**Project 3 Questions**

* 1. What are the assumptions that the k-means algorithm makes of the input data?
     1. *Clusters are spherical/ spread out uniformly from centroids*
     2. *Clusters are separate/ have clear boundaries between them*
     3. *Clusters are similarly sized and have a similar density*
     4. *Clusters separated correctly by Voronoi partitioning*
  2. Each input file has the original generating clusters in the filename before the “.dat” extension. For example, bullseye2.dat was generated with 2 clusters. For each input file provided, run *k*-means 3 times with the k value indicated by the filename.

Record the WCSS of each run in the Columns B-D of the given p3-results.xlsx Excel file. Be sure to visualize the results for each run. For which files does *k*-means clustering appear to succeed always/sometimes/never in your sample of 3 runs?

* + 1. *Always: easygaussian2, easygaussian3, hardgaussian2, hardgaussian3*
    2. *Sometimes: easygaussian4, easygaussian5, easygaussian6, easygaussian7, easygaussian8, easygaussian9, hardgaussian4, hardgaussian5, hardgaussian6,*
    3. *Never: bullseye2, diffdensity2, diffstddev2, easygaussian1, hardgaussian1, hardgaussian7, hardgaussian8, hardgaussian9, stretched2*
  1. For those problems where *k*-means appears to always fail, which assumptions (if any) of the *k*-means algorithm are violated?
     1. *Bullseye2: 1 and 4*
     2. *Diffdensity2: 3*
     3. *Diffstddev2: 3*
     4. *Easygaussian1: only one cluster so 2, 3, and 4 are not applicable*
     5. *Hardgaussian1: only one cluster so 2, 3, and 4 are not applicable*
     6. *Hardgaussian7: 3*
     7. *Hardgaussian8: 3*
     8. *Hardgaussian9: 3*
     9. *Stretched2: 1 and 4*
  2. Is it possible for *k*-means to fail if no assumptions are violated? Why or why not?
     1. *Yes:* 
        1. *If centroids converge on a local minimum. This can happen due to the random initialization (Forgy initialization) choosing bad centroids, or k-means not going through enough iterations.*
        2. *If K does not match the number of clusters. If K is too high, groups can get split. If K is too low, groups can be merged.*
  3. Why should iterated use of the *k*-means algorithm help in some cases with the quality of output clusters?

The repeated application of the k-means algorithm improves the quality of the output clusters, because k-means is sensitive to the initial starting position of the centroids. When k-means algorithm is run, the centroids are initialized randomly, meaning k-means might produce poor local minima, if the initial centroids are poorly placed. By running k-means a number of times, using different initializations of the centroids, then selecting the run with the lowest overall, it is more likely that the resulting clusters will fall within a better global minimum. The more overlap there is within the dataset (overlapping clusters; noise) or the more poorly separated the clusters are, the more important the location of the initial centroids is to the outcome.

* 1. For each input file provided, run iterated *k*-means one time each. Enter the resulting WCSS in Column H of the Excel sheet. For which files does *k*-means clustering appear to succeed in your iterated sample run?

The files where k-means clustering appears to succeed are:

* **bullseye2.dat**: The min WCSS is 22.54196633, which suggests reasonable separation of clusters even though it's not perfect.
* **diffdensity2.dat**: The min WCSS is 9.797249415, which indicates clear separation of clusters despite density differences.
* **diffstddev2.dat**: The min WCSS is 1.12140865, which is very low and indicates excellent clustering.
* **easygaussian1.dat**: The min WCSS is 0.336387617, which is extremely low, indicating perfect clustering.
* **easygaussian2.dat**: The min WCSS is 0.559716829, which is low and indicates good clustering.
* **hardgaussian1.dat**: The min WCSS is 0.291958811, which is very low, indicating accurate clustering.
* **hardgaussian2.dat**: The min WCSS is 0.542170273, which is low, indicating reasonable clustering.
* **hardgaussian3.dat**: The min WCSS is 0.738544702, which is moderately low and indicates reasonable clustering.
  1. For which types of clustering problems does/doesn’t this iterated approach help?
     1. **Does Help:**
        1. When clusters are clearly separated, and errors that arise from initialization are the only problem.
        2. With datasets where there is some overlap, but clear centroids can be identified.
        3. Non-high-dimensional datasets and low-dimensional datasets are easy to measure in terms of distance.
     2. **Doesn’t help**
        1. When clusters are non-convex or have complex shapes.
        2. When clusters are significantly overlapping or are distributed in terms of density.
        3. When the dataset is in a very high-dimensional space, and it is very difficult to determine clear areas of separation.
  2. For each input file provided, run your program 8 times with *kMin*=2 and *kMax*=10 and *iter*=10. From each run, record the number of clusters *k* that yielded the maximum gap statistic in Columns J-Q of the spreadsheet.
  3. For which data set(s) does this technique consistently succeed in discerning the correct number of clusters?
* **diffdensity2:** All runs have the same WCSS and k value (9.797249415).
  + Justification: The dataset has well-separated clusters with clear boundaries.
* **diffstddev2:** All runs produce the same WCSS and k value (1.12140865).
  + Justification: The clusters are well-separated and uniformly shaped.
* **easygaussian2:** No variation in k values across all runs.
  + Justification: Simple and clearly defined clusters.
* **hardgaussian2:** Consistent k values with no deviation.
  + Justification: Clear separation of clusters with low overlap.
* **hardgaussian7:** All runs return the same k value of 1.688931093.
  + Justification: Dataset contains well-separated, spherical clusters.
  1. For which data set(s) does the discerned number of clusters vary from run to run?

 **bullseye2:** Slight variation in k values between 22.3 and 22.6.

* Justification: Slight instability due to complex cluster shapes.

 **easygaussian1:** Minor variations around 0.34.

* Justification: Simple clusters but slight sensitivity to initialization.

 **easygaussian3:** Frequent changes between 6.38 and 6.84.

* Justification: Overlapping clusters make k-means less stable.

 **easygaussian4:** High variability with some extreme values (18.29).

* Justification: Likely due to overlapping or poorly separated clusters.

 **easygaussian5:** Fluctuations between 8.7 and 10.57.

* Justification: High overlap or density differences affect consistency.

 **easygaussian6:** Fluctuation between 6.84 and 7.33.

* Justification: The algorithm struggles with non-uniform clusters.

 **easygaussian7:** Some variation, mostly stable but with occasional spikes.

* Justification: Slight instability due to overlapping clusters.

 **easygaussian8:** Minor variations around 6.19.

* Justification: Mostly stable but minor fluctuations occur.

 **easygaussian9:** Consistently fluctuating around 7.6.

* Justification: Minor overlaps or density variations.

 **hardgaussian1:** Small changes in k values, but mostly consistent.

* Justification: Clear clusters with minor WCSS differences.

 **hardgaussian3:** Small variation between 0.734 and 0.738.

* Justification: Mostly consistent but with slight fluctuations.

 **hardgaussian6:** Significant variations between 3.18 and 4.03.

* Justification: High overlap or non-convex shapes create instability.

 **hardgaussian8:** Variations between 2.31 and 3.73.

* Justification: Dataset has complex clustering structures.

 **hardgaussian9:** One run shows an outlier value (5.91).

* Justification: Unstable detection of clusters with high overlap.
* **stretched2:** Mostly consistent k values except one outlier, indicating stability.
  + Justification: Clusters are elongated but consistently detected.
  1. For which data set(s) does this technique consistently return the incorrectnumber of clusters? What do you observe about the nature of the data in these case(s)?

These datasets consistently detect incorrect k values due to complex shapes, overlapping clusters, or density variations.

* **easygaussian5:** Wide range of k values (8.7 to 10.57).
  + Justification: Overlapping clusters cause instability in clustering results.
* **hardgaussian4:** Constantly fluctuating between 1.48 and 2.12.
  + Justification: High overlap or complex structures reduce the algorithm's reliability.
* **hardgaussian6:** Large variation between 3.18 and 4.03.
  + Justification: Non-convex clusters or uneven density affects detection.