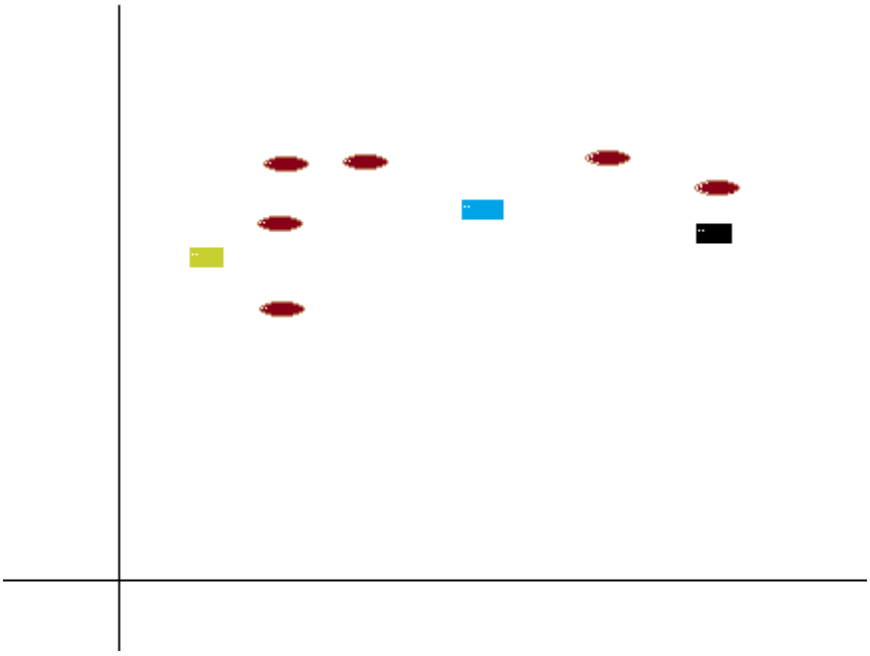
1) Find an example of a small set of points and three initial centroids so that kMeans with k=3 converges to a clustering with an empty cluster. Note that the initial centroids do not have to be members of the set of points. Explain your example in detail in your own words. Zero points without detailed explanation.

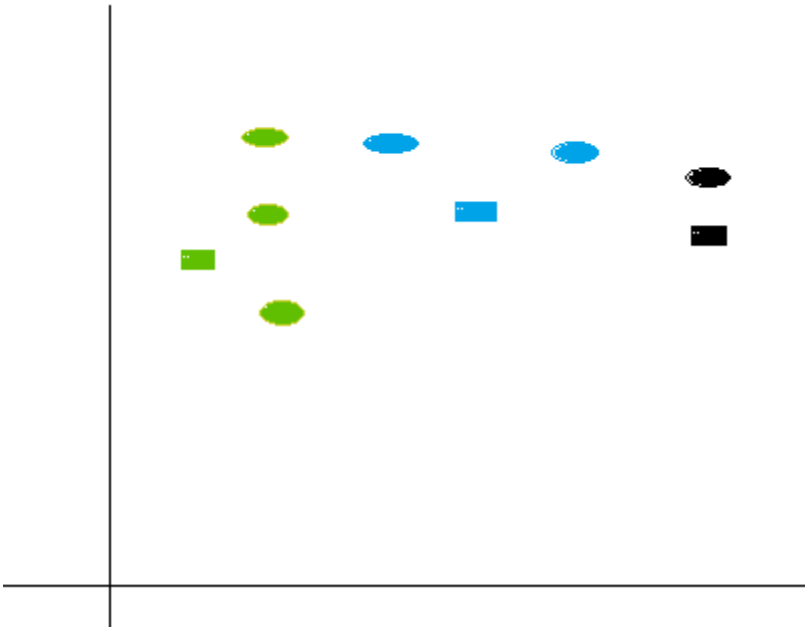
Solution:

kMeans is a method used in cluster analysis. The following example shows the how empty clusters are generated. The example considers a small point set and generates an empty cluster for k=3.

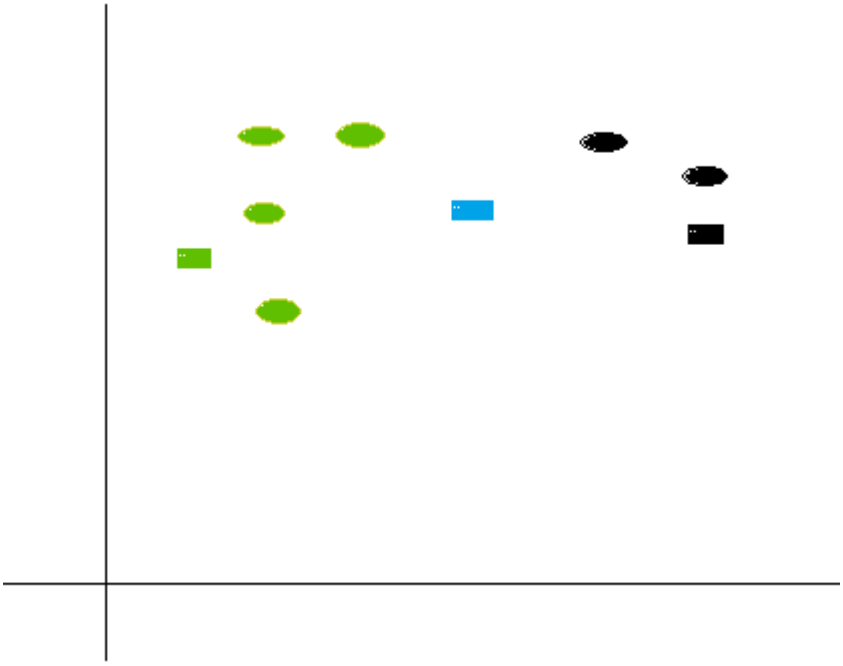
Initially we assume 6 points for cluster analysis (oval shaped) with 3 initial centroid denoted with green, blue and black rectangle. The centroids are selected randomly but effective methods for centroid selection can be helpful.



Iteration 1: The centroid selected use the distance vectors to form clusters using kMeans algorithm. The 3 clusters created are shown with green, blue and black. The point belonging to the same cluster are shown with same color as the centroid of the cluster.



Iteration 2: After the clusters in iteration 1 are formed, the centroid is adjusted and the clustering is done again on the same points. Due to change in centroid the two points that belonged to the cluster with blue centroid change to different cluster.



After adjusting the centroid we see that the cluster with the blue centroid is empty, thus kMeans algorithm may generate empty clusters.

2) We use SSE (RSS) as the measure of cluster quality and kMeans minimizes it. If there is an empty cluster, can that clustering be the global minimum solution based on RSS? Show all details of you arguments. Use your own words.

Solution:

- SSE is the measure of quality of the cluster and is calculated by finding the difference between each point to the closet centroid and then adding the error for each point to find the sum of squared errors.
- The lower the SSE value the better the points are clustered. The major goal of the kMeans is reducing the SSE value for better clustering.
- Thus kMeans uses a local minimum approach to get the optimal solution. But the kMeans algorithm can generate an empty cluster.
- Thus, a solution with empty cluster may be a global minimum solution.

#3) kMeans with soft cluster assignment
Computes the fractional membership of a document in a cluster as a function of the distance D from its centroid. That function is monotonically decreasing, e.g., as e^(1/d)
Wrote very detailed pseudocode of kMeans using this soft version. Provide clear comments for each line of code, in your own words.

Solution:

Step 1: Documents are assigned to each cluster.

Step2: The existence of the document in cluster depends on relevance measure.

Step 3: k iteration are decided for the execution and threshold t for cluster membership.

Step 4: Calculate relevance to cluster by probability measure p

Step 5: if probalility p < Threshold t

Step 6: Spilt cluster.

Step7: Calculate centroid

Step 8: Else
Step 9: if the cluster has documents less than than the decided threshold for each cluster.

Step 10: Else repeat Step6 &7;

Step 11 :If empty Cluster

Step 12: Assign new centroid.

Step 13: End

#4) MMDS Read section 7.2.4 Hierarchical clustering in non-Euclidian spaces
- Bonus points: Exercise 7.2.6. Submitting the solution is optional. But think about the solution as preparation for the final.

5) MMDS Exercise 7.4.1

Solution:

The solution uses a non-Euclidean distance parameter for this so the point is known to be in same centroid so if the point belongs to the same cluster

For the outer rings

If (0.8 * o -0.8 * i) < d then the points belong to same cluster .

For inner ring

If(0.8 * i – 0.8 * c) < d then the point belongs to the same cluster.

#7) Weka Experiments

Use two datasets: iris, vote
Use the cluster tab. Use the option “Classes to cluster evaluation”

Run SimpleKMeans, DBScan.

Don't assume you know the number of clusters for each data set, experiment with different numbers of clusters parameter, pick the best based on the evaluation methods that we discussed, compare to the true number of clusters. Remember the relation between SSE and the number of clusters when you compare different clusterings - make sure your comparison is meaningful.

1) Use the preprocessing tab and the visualize tab to explore and visualize the attributes. Analyze what you see and how you think it can affect clustering.

2) Cluster the data, use the default values, report the results for both algorithms. Explain how you evaluate the cluster performance in Weka, what measures do you see in the output tab, what do they mean.

Explain all parameters for each algorithm. What parameters should be changed for the experimentations and what parameters should be used in the default values and why.

3) Select some attributes based on your analysis in 1) and use only them during clustering. Analyze the results. How are the results different from 2)? Is it what you expected based on your analysis of the data?

4) Use 3-4 different sets of parameters for each algorithm (such as number of clusters for kMeans; epsilon and minPoints for DBScan). Experiment until you get much better results than with default. Explain what parameter values gave you best performance and why do you think those values were based based on your understanding of the data and the algorithm.

5) What are your conclusions? What did you learn? How do these two algorithms perform on different data sets? etc.

Try to make the report informative, concise, use summary tables etc where appropriate. Show what you learned during the semester in terms of data analysis, experiments, and conclusions.

Solution:

i) Iris Dataset

Visualization and Preprocess:

- Preprocess tab shows that the dataset has 5 attributes namely, sepal length, sepal width, petal length, petal width and class. The preprocess tab also shows the maximum and minimum values of the attributes. The visualize option in preprocess tabs shows the graph of each attribute with the count of class they belong. The attribute values can help in clustering of the Iris dataset. The max and min values can help in determining outliers which affect clustering.
- The visualize tab consist of 5*5 graphs which plot the relation between on the attributes. Each graph in these 25 grids consists of 2 attributes and the plot of points for each value. The different colors represent the different value of class to which the attribute belong. The graphs can be useful when clustering is to be done on only a few attributes.

a) kMeans Iris Dataset.

A) kMeans with default parameters

```
kMeans
======

Number of iterations: 7
Within cluster sum of squared errors: 12.143688281579722
Missing values globally replaced with mean/mode

Cluster centroids:
                         Cluster#
Attribute      Full Data         0          1
                   (150)      (100)       (50)
=============================================
sepallength       5.8433      6.262      5.006
sepalwidth         3.054      2.872      3.418
petallength       3.7587      4.906      1.464
petalwidth        1.1987      1.676      0.244



Time taken to build model (full training data) : 0.01 seconds

=== Model and evaluation on training set ===

Clustered Instances

0      100 ( 67%)
1       50 ( 33%)
```

- The above example runs SimplekMeans with the default values and the resultant values are given in the above table.
- The algorithm runs for 7 instances and with 2 clusters forming a cluster with 67%(100) of values in one cluster and the rest 33% with cluster #1.
- The above table shows the centroid value for each attribute of the relation. The visualize option in the result buffer plots the different values with respect to attributes.
- The Sum of Squared error, which determines the effectiveness of the kMeans algorithm comes out to 12.14 .

```
Class attribute: class
Classes to Clusters:

  0  1  <-- assigned to cluster
  0 50 | Iris-setosa
 50  0 | Iris-versicolor
 50  0 | Iris-virginica

Cluster 0 <-- Iris-versicolor
Cluster 1 <-- Iris-setosa

Incorrectly clustered instances :       50.0     33.3333 %
```

- The clustering based on class yields the above result, which shows that the incorrectly clustered instances are 50 as Iris-vericolor and Iris-virginica belong to the same cluster.

B) Parameters in kMeans.

- The kMeans algorithm takes into account the following parameters
    - DisplayStdDevs :This attribute display standard deviation for numeric attributes and count for nominal attribute default values is false. If set to true then the centroid table displays the std deviation value. Output is not affected by this parameter.
    - Distancefunction: Distance function is used to compare two instances. The default value is Euclidean distance , Manhattan distance also works with kMeans. On selection of Manhattan distance as the distance function. Manhattan distance uses sum of within cluster distance for performance measurement. The cluster center changes and the clustering percentage also varies due to this function.
    - MaxIteration: The kMeans is an iterative approach and maxIteration specifies the value of the maximum iteration kmeans can take. If this values is reduced to a very small value (5) then the clustering is affected to a very small extent.
    - numCluster: It specifies the number of clusters in which the data can exist. Default value is set to 500.Large number cluster for Iris dataset ensures that the instances are spread evenly across the cluster, the SSE is drastically reduced due to large number of clusters.
    - preserveInstanceOrder:This attribute helps to perverse the order of the instances.The default value is false and output is not affected by this parameter.

- Seed: This is a random seed value ,default value is 10.Change in seed doesn't affect the result

C) Parameters affecting output:

- Distance Function: Distance function affects the output of the cluster. The simplekMeans algorithm has two distance factors Euclidean and Manhattan distance, each uses different evaluation variables for performance.
- Number of cluster: The number clusters affect the output and higher number clusters may result in very low value of SSE or very high value (depends on the distribution of data).

D) Attributes that affect the simplekMeans:

- Petal Length and Petal width

```
kMeans
======

Number of iterations: 6
Within cluster sum of squared errors: 5.179687509974783
Missing values globally replaced with mean/mode

Cluster centroids:
                           Cluster#
Attribute      Full Data          0          1
                   (150)      (100)       (50)
=============================================
petallength       3.7587      4.906      1.464
petalwidth        1.1987      1.676      0.244



Time taken to build model (full training data) : 0 seconds

=== Model and evaluation on training set ===

Clustered Instances

0      100 ( 67%)
1       50 ( 33%)
```

- If the sepal length and width are discarded and the clustering is run on the instances then the SSE error value is drastically reduced and efficient solution is generated.

- Sepal width and sepal length

  If we discard the petal width and petal length then the output generated is the following

```
kMeans
======

Number of iterations: 4
Within cluster sum of squared errors: 7.081076555902943
Missing values globally replaced with mean/mode

Cluster centroids:
                          Cluster#
Attribute      Full Data          0          1
                   (150)       (84)       (66)
=============================================
sepallength       5.8433     5.2333     6.6197
sepalwidth         3.054     3.1286     2.9591



Time taken to build model (full training data) : 0.01 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       84 ( 56%)
1       66 ( 44%)
```

  The SSE is high in this case as compared to kMeans performed on petal length and width. The clustering is also different as 56% is one cluster and 44% is in another.

E) Conclusion

- The Iris dataset works well with 3 clusters as there are 3 class labels in the Iris data set and hence the incorrectly clustered instances are less if there are 3 clusters.
- Besides, if the sepal length and width are ignored the output formed by simplekMeans has very small value of SSE.

b) DBSCAN on Iris dataset

A) DBSCAN with default values:

```
Clustered Instances

0      150 (100%)


Class attribute: class
Classes to Clusters:

  0  <-- assigned to cluster
 50 | Iris-setosa
 50 | Iris-versicolor
 50 | Iris-virginica

Cluster 0 <-- Iris-setosa

Incorrectly clustered instances :       100.0    66.6667 %
```

- The DBSCAN uses a metric wherein its groups together all the points in the high density region and the outliers are present in the low density region .It uses nearest neighbor technique for clustering.
- The above shows the result of running the DBSCAN algorithm using default parameter. The 150 instances are classified in a single cluster.
- The result shows that only a single cluster is required if epsilon value is 0.9 and min points are 6.
- The above result also shows that 100 out of 150 are incorrectly classified as they belong to the same single cluster.

B) Parameters in DBSCAN.

- Database_type: this parameter specifies the database used in the DBSCAN method. The database is provided is weka.
- Database_distance_type: This indicates the distance measure used by the DBSCAN algorithm. Default is the Euclidean distance.
- Epsilon: This the epsilon radius for range queries which is used calculate the radius from core points. The default value is 0.9
- minPoints: These are the minimum number of points that are required to be in the radius of epsilon or in the epsilon range query.

C) Parameters in DBSCAN that affect the output:

- Epsilon: Epsilon specifies the radius of range queries and hence an alteration may cause some instances to enter or be removed from the cluster. If the value is reduced to 0.1 or lower the number of clusters

increases to 3 as there is low range radius. The incorrectly classified clusters also reduce drastically. The epsilon value if increased above 0.7 then it only generates one cluster. Thus, for Iris dataset if the epsilon value increases above 0.7 it only generates one cluster.

- minPoints : The default value is 6,but in this case the minimum points doesn't affect the output of clustering.

D) Attributes effect on DBSCAN

- DBSCAN is not affected if perform the processes without affecting the result of the clustering.

E) Conclusion

- The DBSCAN on Iris data set produces the best result when the epsilon value is reduced to below 0.5.This results in instances with less noise.

ii) Vote Dataset

Visualization and Preprocess:

- Preprocess tab shows that the dataset has 17 attributes with a class attribute with nominal values 'y' and 'n'. There are 3% missing values out of total 435 instances. The visualize tab also shows the same 17* 17 grid graphs as seen in the Iris data set.

a) simplekMeans

A) simplekMeans on vote with default parameter

```
kMeans
======

Number of iterations: 3
Within cluster sum of squared errors: 1449.0
Missing values globally replaced with mean/mode

Cluster centroids:
                                                Cluster#
Attribute                               Full Data       0       1
                                            (435)   (207)   (228)
=============================================================================
handicapped-infants                            n       n       y
water-project-cost-sharing                     y       y       n
adoption-of-the-budget-resolution              y       n       y
physician-fee-freeze                           n       y       n
el-salvador-aid                                y       y       n
religious-groups-in-schools                    y       y       n
anti-satellite-test-ban                        y       n       y
aid-to-nicaraguan-contras                      y       n       y
mx-missile                                     y       n       y
immigration                                    y       y       y
synfuels-corporation-cutback                   n       n       n
education-spending                             n       y       n
superfund-right-to-sue                         y       y       n
crime                                          y       y       n
duty-free-exports                              n       n       y
export-administration-act-south-africa         y       y       y
```

- The simplekMeans algorithm takes 3 iteration and the sum of squared errors (SSE) is 1449 which is very high as compared to the value for other dataset.
- The Centroid table is as shown above which shows that two clusters are present – cluster #0 and #1.

```
=== Model and evaluation on training set ===

Clustered Instances

0      207 ( 48%)
1      228 ( 52%)


Class attribute: Class
Classes to Clusters:

   0   1  <-- assigned to cluster
  50 217 | democrat
 157  11 | republican

Cluster 0 <-- republican
Cluster 1 <-- democrat

Incorrectly clustered instances :       61.0     14.023  %
```

- The above results shows the number of instances in each cluster 0(48%) and cluster1 (52%).
- The class to cluster table shows the mapping of class democratic and republican to the clusters. Based on these values the incorrectly classified instances are calculated.
- The incorrectly classified instances are 61 out of 435 which gives 14.02% incorrectly classified instances.

B) Parameters that affect output

- maxIteration : If we decrease the value of max iteration the result are not affected .The SSE and incorrectly clustered instances are not affected and hence the result are same as the previous run for default parameters.
- numClusters: If the maxIteration are kept constant the increase in the number of clusters causes the SSE value to decrease , but on the other hand the incorrectly clustered instances increase.

C) SimplekMeans with changed attributes:

- If we discard the attributes adoption-of-budget-resolution and physician-fee-freeze the output of the simplekMeans algorithm is drastically affected.
- The SSE of the algorithm reduces to 1275.0 and the error increases to 17.93% .

D) Conclusion:

- Thus, from the above results we can conclude that simplekMeans algorithm provides good quality clustering when the number of clusters are reduced to 2.
- The vote dataset mainly consists of two class labels namely republican and democrat and hence the clusters only two clusters.
- If we restrict the maxIteration two values like 5 and the result can be even more optimal.

b) DBSCAN

A) DBSCAN with default parameters:

```
Unclustered instances : 308

Class attribute: Class
Classes to Clusters:

  0  1  2  3  4  5  6  7  8  9 10 11 12 13  <-- assigned to cluster
  0  1 14  0  0  0  0  0  0  9  9  9  6  7 | democrat
 14 13  0 10  8  8  6  7  6  0  0  0  0  0 | republican

Cluster  0 <-- republican
Cluster  1 <-- No class
Cluster  2 <-- democrat
Cluster  3 <-- No class
Cluster  4 <-- No class
Cluster  5 <-- No class
Cluster  6 <-- No class
Cluster  7 <-- No class
Cluster  8 <-- No class
Cluster  9 <-- No class
Cluster 10 <-- No class
Cluster 11 <-- No class
Cluster 12 <-- No class
Cluster 13 <-- No class

Incorrectly clustered instances :       99.0     22.7586 %
```

- The output for DBSCAN clustering consists of 13 clusters with 2 clusters that belong to democrat and republic class , the rest of the cluster belong to no class.
- The unclustered instances are 308 and incorrectly clustered instances are 99 i.e 22.75% of the total dataset instances.

B) Parameters that affect the output

- Epsilon: If we increase the epsilon value to 1.5 only one cluster is formed .The total noise in the data is also reduced drastically to 6.But the number of incorrectly classified instances increases .If we further increase epsilon the output remains same.
- minPoints :The default value is 6,but if the value is decreased then the output is drastically affected and the number of clusters and noise points also increase incorrectly clustered instances. But if minpoint is increased to 10 the incorrectly clustered attribute decrease but the noise instances also increase.

C) Attribute change in DBSCAN

- After excluding adoption-of-budget-resolution and physician-fee-freeze the DBSCAN output remains the same.

D) Conclusion:

- DBSCAN algorithm uses two parameters epsilon and minpoints for the process of clustering.
- If the value of epsilon and min points is chosen above 1.5 and 20 respectively the result formed contains very low amount of Noise instances.

iii) Conclusion after clustering on Iris and Vote dataset

- kMeans clustering algorithm is centroid based algorithm that uses sum of squared errors(SSE) as performance measure to determine the clustering efficiency.
- kMeans algorithm works efficiently when the number of cluster equals to the number of class labels.
- In vote dataset the algorithm produces least noise instances when the number of clusters are 2 (one for republican, one for democrat).In Iris dataset the algorithm gives best result when there are 3 clusters.
- DBSCAN uses a density based approach to form the clusters and the it uses the nearest neighbor concept for the same.
- The min points and epsilon are the two parameters that affect the DBSCAN, the core points are assumed as center and epsilon is the radius of the nearest neighbors.
- The DBSCAN algorithm uses higher value of epsilon and min points in case of vote dataset as the instances are scattered and are large in number.

- The DBSCAN uses lower value of epsilon and min points as there are lesser instances for Iris dataset as compared to vote dataset.