VILNIUS UNIVERSITY
FACULTY OF MATHEMATICS AND INFORMATICS
MASTER'S STUDY PROGRAMME
Data Science

A Comparison of Clustering Methods
Master's thesis

Author:
Shanice Kilaire

Supervisor:
Viktor Skorniakov

Vilnius
2024

Abstract

In this paper, a comparison of the three recently published novel clustering methods: Parameter free Clustering based K means, A Criterion for Deciding the Number of Clusters in a Dataset and Graph-based Data clustering via Multiscale Community Detection is undertaken. Each of the these methods are replicated on both real and synthetic data sets for which their performance is evaluated with the regards to the relative error between the number of outputted clusters and the true cluster number for each data set. It was found that the Graph-based Data clustering via Multiscale Community Detection was the best performing and was able to determine the cluster number for each data set with complete accuracy with regards to the number of partitions. Further evaluation of each method considers the quality of the output in more depth as well as evaluates the strengths and limitations of each respective method.

Contents

1 Introduction

Clustering is a data mining technique used to process large data in the fields of data analysis and machine learning. Clustering has many purposes that range from dimensionality reduction to uncovering meaningful information about a datas structure. There are a wide range of clustering techniques used today for which each of them aim to maximise the similarity of points within a cluster and the dissimilarity between alternate clusters for which uncover patterns in data that are not readily accessible. Clustering can be applied to a multitude of data sets that are not limited to numerical, such as images and graph networks which has given rise a wide range of techniques. The majority of commonly used clustering algorithms focus on applying distance metrics to measure the closeness of data points. Despite the efficacy of these methods, there are still persistent challenges in the field of clustering with regards to parameter sensitivity, initialisation processes and dealing with outliers and noisy data. The challenges that arise with mining high dimensional data are referred to as the curse of dimensionality, which is a limitation to even the most robust clustering algorithms.

In this paper, three recent novel clustering methods are compared and contrasted with regards to the best performing method. They each stem from individual areas where one proposes a parameter free clustering algorithm, the second a parameter estimation algorithm and the final method introduces a novel framework applied to a graph based multi scale community detection method. Each of these studies are replicated on a selection of both real and synthetic data that varies in attributes and dimensionality that were not covered in the original analysis.

2 Literature Review

In this section, a comprehensive overview of established clustering methods is detailed with regards to the relative field of each clustering method used in the comparison.

One of the most commonly used clustering methods is the k means clustering algorithm. This is an unsupervised learning method, meaning that the algorithm does not have knowledge regarding details of the data set such as labels, upfront. The algorithm works by segregating data into k partitions by calculating the distance between each data object and a cluster center such that the similarity of data objects within a cluster is maximised, and the dissimilarity between clusters is maximised. Therefore data objects are assigned to the centers that they are closest to. The process of selecting the initial k cluster centers is random. The distance between each data object and cluster center is calculated at each iteration and the mean of all the data objects within a cluster is assigned to that cluster k. The process continues until there is no longer a change in the mean cluster values [**6**] The random initialisation process conveys that the results may not always be stable, and considering pairwise distances and each data object in isolation yields to outlier sensitivity. The method is therefore sensitive to the parameter k which means that the selection of k needs to be a well considered process. The criterion method based on data depth[**2**] is a novel method that estimates the number of clusters k with relevance to the importance of this metric.

Another commonly used clustering method is the density based spatial clustering of applications with noise known as the DBSCAN method. In this method, data objects are clustered based on their density, where clusters are determined by high density regions and outliers are conveyed as noise. This method is therefore robust to outliers and effective in determining clusters of varying shape [**28**].For each data point in a cluster it has an ϵ neighbourhood value which is a radius value defined by a distance metric and a 'MinPts' value which is the minimum number of points that are within the this ϵ radius [**4**]. Although this algorithm is able to cluster data of varying shape and does not include an initial input parameter for the number of clusters, it is not able to cluster data sets that have significant difference in densities which renders the parameter selection unsuitable for all clusters of this kind. Additionally the method is not deterministic meaning is does not produce consistent results as the algorithm has a random starting point [**32**].

Another common branch of clustering regards a data visualisation technique to determine similarities known as hierarchical clustering. The data visualisation concerns diagrams that have tree based structures but are mostly reliant on interpretation of a dendrogram. A dendrogram is a diagram that conveys the relations between data objects in a data set. It behaves as almost like a tree like structure where the branches convey the clusters. The vertical component represents the distance between clusters defined by some distance metric and the horizontal component represents the data points [**24**]. To compute this dendrogram there are two main methods. One is agglomeritve clustering which works by starting with n groups where a single point considered as a cluster features in each group. The algorithm is said to work as a ' bottom up' approach to clustering data, where the two groups of the highest similarity are joined to form one group and the process repeats until there is a single group that is inclusive of all the data objects. The vertical component or height of each group conveys the dissimilarity between the groups being joined [**31**]. The second approach is known as divisive clustering which is the converse of agglomeritive clustering. Instead all data objects feature in a single cluster and the data is divided iterative according to the dissimilarity between objects. This method is effective in identifying large clusters whereas the agglomerative method better identifies smaller clusters [**8**].

Regarding graph based approaches to clustering, one of the main techniques considered is spectral clustering. Spectral clustering focuses on extracting communities from graphs or networks in the form of nodes. The matrix representations of the graph are extracted such as the adjacency matrix A which denotes the connections between the nodes and the degree matrix D conveying the number of edges attached to each node, are then extracted from the graphical representation of the data set where the eigenvectors and eigenvalues are analysed. An eigenvector is a non zero vector that when multiplied a matrix A, the result is a multiple of that matrix A, and an eigenvalue is the scalar multiple corresponding the eigenvector such that $Ax = \lambda x$ [**13**]. Using the information given by the eigenvectors and eigenvalues the variations of the graph Laplacian matrices, of the form $L = D^{-1}A$ can be deduced which are the main focus of spectral clustering for the purpose of identifying the data's inherit structure by considering it in its entirety [**20**] These Laplacian matrices are used to identify natural clusters in the data but the significance of spectral clustering is the use of eigenvectors and eigenvalues to uncover the networks structure. The multi scale community detection algorithm replicated in this paper is a branch of graph based clustering that uses a novel framework applied to the matrix representation of the data extracted from the graph network.

3 Theoretical Description of Methods

In this section, each study implemented for the comparison is detailed with regards to their theoretical background. As these are novel clustering methods published within the past 3-4 years, these descriptions focus on the methodology and key definitions used to implement each respective algorithm.

3.1 A Parameter free Clustering Algorithm based K Means

In this study, the authors Said Slaoui and Zineb Dafir, propose in their clustering study [**29**], a novel method that aims to determine the optimal number of clusters in a data set as well as the initial cluster centres without any prior initialisation of the number of clusters. Their methods consist of developing a new clustering heuristic that combines and existing E – transitive heuristic adapted to quantitative data as well as adaptations of the traditional k means clustering algorithm denoted as PFK-means. The mathematical concepts behind their E- transitive method and its derivation are outlined in detail below alongside Slaoui and Defir's variations of their parameter free methods.

3.1.1 Mathematical Concepts

The theoretical concepts behind Slaoui and Defirs defined in their work [**29**], resembles traditional clustering ideology: to partition data into distinct clusters such that the union of all clusters is equal to the data set, and that there exists no data element that is present in more than one cluster, thus maximising the dissimilarity across clusters.

The cluster centres are defined as follows:

$$c_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$$

Where c_j is the cluster centre of each respective cluster C_j, therefore the center is defined as the sum of all data points divided by the size of the cluster which denoted the center as the clusters location within the data set.

The distance between the clusters centre c_i and a data object x_i is determined using the Euclidean distance metric. The Euclidean distance between a pair of points (x_1, y_1) and (x_2, y_2) is denoted by

$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$. This conveys the difference between the x and y coordinates in each respective coordinate pair for which the horizontal and vertical distance components are obtained. Summing and square rooting the addition of these distances yields the distance between the coordinates which stems from the Pythagorean theorem of calculating missing triangle lengths. The Euclidean distance is therefore used to calculate the closest distance between a pair of points [**3**]. Applying this ideology, Slaoui and Defir, denote the quality metric of a cluster C_j as the sum of squared error between all data objects c_i in a cluster C_j and the cluster centre c_j. To do this, the difference between the expected value of the distance function and the average value of the distance function is computed. This suffices as a quality metric as it provides an understanding of the dispersion of the cluster centres and the overall spread of the data which therefore provides an insight to towards the datas structure and the reliability of this shape in terms of these distances. The expected value of the cluster distances is defined as follows:

$$E = \sum_{i=1}^{n} \sum_{j=1}^{k} dist(x_i, c_j)^2, x_i \in C_j$$

Which is simply the squared distances of the data objects and the respective cluster centres. The average distance of all the data objects in the entire data set denoted as X is given by:

$$MeanDist(X) = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j+1}^{n} dist(x_i, x_j)$$

Which computes a normalised output of the distances occurring between 0 and 1. New cluster centres are therefore computed as follows:

$$dist(c_{new}, c_j) = Max(dist(x_i, c_j))$$

Where j and k are bounded by p, the number of existing cluster centres, in the following manner: $1 \leq j \leq p$, $p \leq k$, and c_j is the j^{th} cluster center. [**29**].

3.1.2 The Transitive Heuristic

The PFK heuristic measure as stated previously, is a combination of the E- transitive heuristic and a representative of a parameter free clustering algorithm. The E transitive heuristic is an adaptation of the transitive heuristic which aims to cluster data without the use of parameters or specifying the number of clusters, and instead clusters are based the similarities between data objects. The transitive heuristic regards clustering categorical data via an iterative process for which the pipeline is outlined in the flow chart below that was detailed in their previous alternate work [**30**]

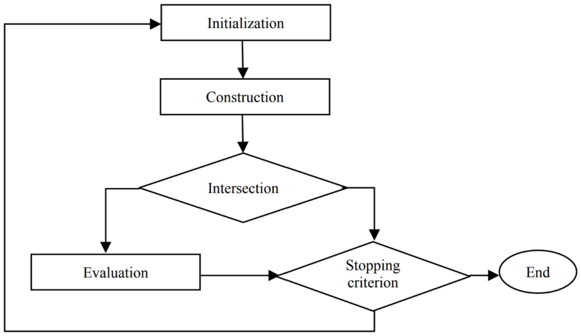


Figure 1: Transitive heuristic process

The transitive clustering process commences with an initialisation stage where data objects are randomly selected as the representative of a cluster. The construction stage formulates clusters by assigning similar data objects to this cluster representative using a metric known as the Condorcet's criterion. The Condorcet method applied to statistics concerns a branch of relational analysis for which the relationship between categorical variables is explored with regards to their similarity .The ideology behind the Condorcet method is stemmed from voting theory where candidates were elected based on their ranking relative to other candidates, so the ideal candidate was selected via a process of pairwise comparisons [**38**] .The ideal or winning candidate was identified as having the highest ranking of preferential interest compared to all other candidates. The Condorcet ideology has been applied to the transitive method of clustering in the form of a Condorcet Criterion which is not an exact measure but can be expressed as such..As stated by Ah-Pine and Marcotorchino [**1**], to first apply the methodology of relational analysis, the relationship between variables needs to be encoded as a binary relation, where 1 represents a relation and 0 no relation, for which the values are stored in matrices known as 'relational matrices'. In this context the Condorcet criterion is a mathematical expression that conducts pairwise comparisons between relational matrices to output a numerical value that determines the magnitude of the relation that exists between the pairwise objects, a higher value thus constitutes to a higher similarity between variables. Slaoui and Dafir [**30**] have expressed this metric with regards to the mathematical encoding of the pairwise similarities between data objects. To first define the Condorcet criterion C_{ij} is defined which is the collective relational matrix based on the individual relational matrices. The individual matrix for the individual attributes a_l is defined as C^l_{ij}, where $a_l(d_i)$ of the attribute a_l for the data object i:

$$C^l_{ij} = \begin{cases} 1 & if a_l(d_i) = a_l(d_j) \\ 0 & otherwise \end{cases}$$

$$C_{ij} = \sum_{l=1}^{k} C^l_{ij}$$

The Condorcet criterion therefore makes pairwise comparisons between these matrices and X, the cluster collective that conveys the partition that needs to be optimised:

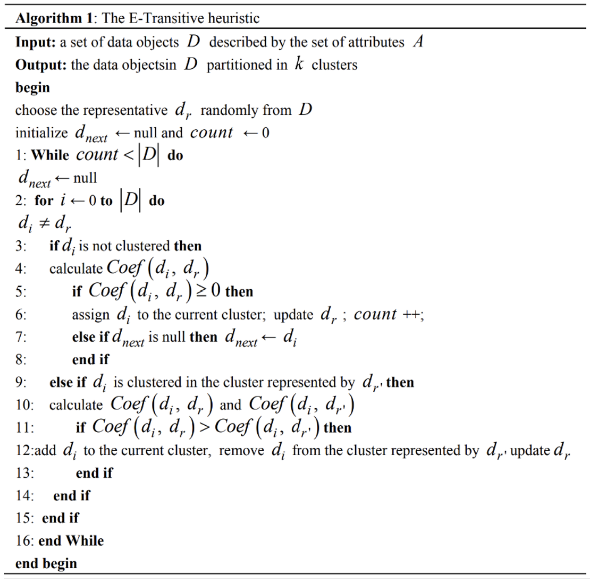
$$\underset{X}{MaxCondorcet}(C, X) = \sum_{i=1}^{N}\sum_{j=1}^{N} C_{ij}X_{ij} + \sum_{i=1}^{N}\sum_{j=1}^{N} \bar{C}_{ij}\bar{X}_{ij}$$

The above formula conveys the maximum value for the Concert criterion with respect to the cluster collectives regarding partitions and relational matrices. The compliments $\bar{C}$ and $\bar{x}$ for the cluster collective and collective and relational matrix collective represent their converse values. The Condorcet criterion therefore computes the similarity metric by considering not only the relationship between the pairs considers in each comparison, but the dissimilarity amongst them to output an overall similarity metric to best decide the maximum similarity.

The next step in the transitive clustering process is known as the intersection stage. The purpose of performing intersections between all clusters is to identify shared data objects, as an object can be assigned to more than one cluster. The subsequent evaluation step therefore decides which cluster that these shared objects would be best suited or assigned to. To do this the contribution for each of the shared data objects toward each cluster is considered and thus the shared data objects are assigned to the cluster that constitutes towards the highest contribution. Once all data objects are clustered the process terminates and if not, the process reiterates from the initialisation stage.

3.1.3 E Transitive

The Enhanced transitive heuristic (E – transitive) is an augmentation of the transitive heuristic that is also applied to large data sets that have categorical features. The main improvements of this algorithm are found in different stages of the proposed transitive heuristic. For the E transitive process, the initialisation phase differs in the selection of the cluster representatives. As opposed to all representatives being chosen randomly, instead only the first cluster representative is randomly selected and each subsequent data object undergoes a pairwise comparison to the abstaining cluster centre using the Condorcet criterion. The data point that has a contribution or positive Condorcet criterion value to this current cluster centre is added to this cluster and labelled as clustered. In the case of a data point having a negative Condorcet criterion value that conveys dissimilarity, it is noted as not clustered will be saved as the new cluster representative for the next iteration. The only remaining change in this method regards data points that are labelled clustered. When a new cluster centre is determined, the previous clustered data point is compared to the new forgoing cluster centre via its Condorcet criterion value and its contribution to the new cluster is higher than its current cluster, then



```
Algorithm 1: The E-Transitive heuristic
Input: a set of data objects $D$ described by the set of attributes $A$
Output: the data objectsin $D$ partitioned in $k$ clusters
begin
choose the representative $d_r$ randomly from $D$
initialize $d_{next} \leftarrow$ null and $count \leftarrow 0$
While $count < |D|$ do
  $d_{next} \leftarrow$ null
  for $i \leftarrow 0$ to $|D|$ do
  $d_i \neq d_r$
    if $d_i$ is not clustered then
      calculate $Coef(d_i, d_r)$
        if $Coef(d_i, d_r) \geq 0$ then
          assign $d_i$ to the current cluster; update $d_r$ ; $count$ ++;
        else if $d_{next}$ is null then $d_{next} \leftarrow d_i$
        end if
    else if $d_i$ is clustered in the cluster represented by $d_{r'}$ then
      calculate $Coef(d_i, d_r)$ and $Coef(d_i, d_{r'})$
        if $Coef(d_i, d_r) > Coef(d_i, d_{r'})$ then
add $d_i$ to the current cluster, remove $d_i$ from the cluster represented by $d_{r'}$, update $d_r$
        end if
    end if
  end if
end While
end begin
```

H

Figure 2: E Transitive Heuristic

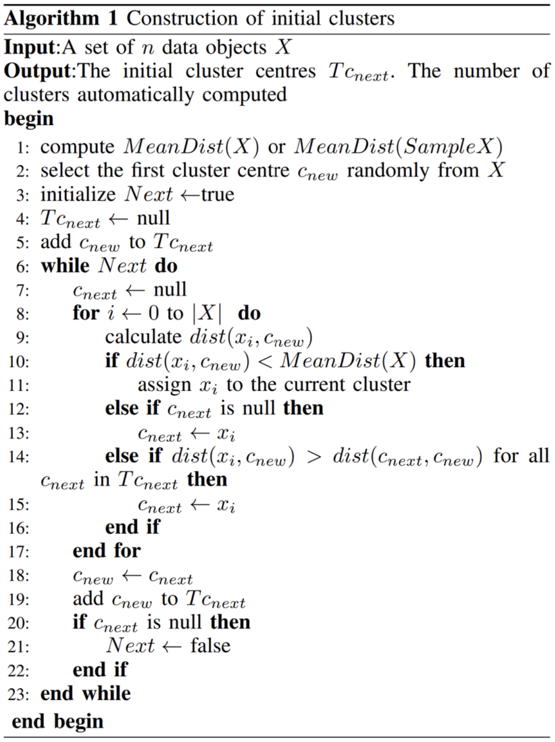
it is removed from the old cluster. This process repeats until all data points are assigned to cluster, therefore the cluster centres themselves are formed iteratively without being defined upfront. This method is an improvement of the existing transitive method as it eliminates the intersection stage. This was achieved as the status of each data object is labelled as clustered or not clustered which avoid data objects belonging to the multiple clusters, hence the construction and evaluation stage form a singular stage in the process. Lastly, each cluster representative is updated using the maximum likelihood estimation metric each time a new data object is added to the cluster. Slaouri and Dafir define this in their pseudo code below [**30**]:

The input of the algorithm outlined above takes in a set of data objects or points denoted as D and encodes them as a set of attributes A where $A = a_1, a_2, \ldots, a_q$ where q denotes a set of categorical attributes.

3.1.4 The PFK Heuristic

In Slauoi and Dafir,s recent work, [**29**] they derived a PFK method which is a combination of the E -transitive method but applied to quantitative data and the k means clustering method devised in two stages. The first stage or initialisation stage makes use of the E transitive method to iteratively form the cluster centres without the use of initial parameters or specifying k, the number of clusters. The average distance between each of the data objects is first computed using the $MeanDist(X)$ and then a cluster centre is randomly selected. The forgoing process to assign data objects to a cluster therefore uses the Euclidean distance as the similarity metric as opposed to the Condorcet Criterion since the data to be processed is no longer categorical. New data objects are assigned to the cluster by computing the Euclidean distance between the current cluster centre and all remaining data objects in the data set. The data objects that have a Euclidean distance less than this average distance computed earlier, are assigned to the cluster. To form a new cluster centre, the data point that is the least similar to the current cluster centre is selected for the purpose of dissimilarity. To do this the maximum Euclidean distance defined in equation 3, is used to select the new cluster representative and subsequent data objects are assigned to it in the same iterative manner. The process is continued until all data objects are assigned to a cluster thus the initial cluster centres are defined, the algorithm for this initial stage is outlined below:

The second stage in the PFK method is to implement the traditional k means clustering algorithm using the output of Algorithm 1 as the input parameters. The aim of this method as stated by Slaoui and Dafir is to automatically detect the number of clusters k using the initial cluster centres from the previous stage. Therefore, the k means clustering method is applied to the initially clustered data set with established cluster centres as opposed to raw data. The Euclidean distances are calculated between



Algorithm 1 Construction of initial clusters

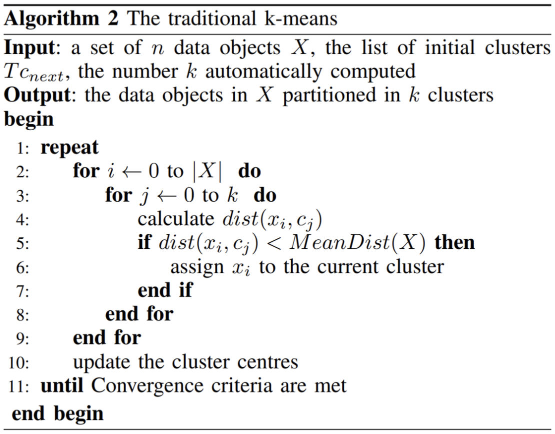
Input:A set of n data objects X
Output:The initial cluster centres Tc_{next}. The number of clusters automatically computed

```
begin
compute $MeanDist(X)$ or $MeanDist(SampleX)$
select the first cluster centre $c_{new}$ randomly from $X$
initialize $Next$ ←true
$Tc_{next} \leftarrow$ null
add $c_{new}$ to $Tc_{next}$
while $Next$ do
    $c_{next} \leftarrow$ null
    for $i \leftarrow 0$ to $|X|$ do
        calculate $dist(x_i, c_{new})$
        if $dist(x_i, c_{new}) < MeanDist(X)$ then
            assign $x_i$ to the current cluster
        else if $c_{next}$ is null then
            $c_{next} \leftarrow x_i$
        else if $dist(x_i, c_{new}) > dist(c_{next}, c_{new})$ for all $c_{next}$ in $Tc_{next}$ then
            $c_{next} \leftarrow x_i$
        end if
    end for
    $c_{new} \leftarrow c_{next}$
    add $c_{new}$ to $Tc_{next}$
    if $c_{next}$ is null then
        $Next \leftarrow$ false
    end if
end while
end begin
```

H

Figure 3: Pseudo Code for Algorithm 1

each data object and each of these initial cluster centres and are then assigned to the cluster that corresponds to the smallest Euclidean distance smaller than the $MeanDist(X)$ calculated previously. . Once these data objects have been re assigned, the mean distance of all data points in each cluster and assigned to its cluster centre or representative k. The process repeats until there are no changes in the k mean values. Slaoui and Dafir note that applying the k means algorithm to the initialised cluster data set enables a rapid convergence to an optimal solution and convey its application to Algorithm 1 in the outline of Algorithm 2 below.



Algorithm 2 The traditional k-means

Input: a set of n data objects X, the list of initial clusters Tc_{next}, the number k automatically computed
Output: the data objects in X partitioned in k clusters

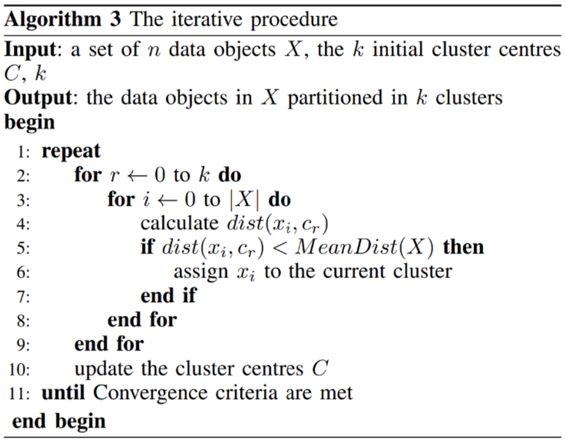
```
begin
repeat
    for $i \leftarrow 0$ to $|X|$ do
        for $j \leftarrow 0$ to $k$ do
            calculate $dist(x_i, c_j)$
            if $dist(x_i, c_j) < MeanDist(X)$ then
                assign $x_i$ to the current cluster
            end if
        end for
    end for
    update the cluster centres
until Convergence criteria are met
end begin
```

Figure 4: Pseudo code for Algorithm 2

Slaoui and Defir proposed variants of their PFK method to further investigate their method. The variants regard the initialisation process with two main approaches called the overlapping PFK means and the hard-PFK means.

3.1.5 Overlapping PFK means

The aim of this variant is to resolve the issue of a data object belonging to multiple clusters in the initialisation stage which is denoted as an overlapping cluster. Thus the result of this method yields to overlapping clusters as opposed to distinct ones. The overlapping clusters not only maintain the integrity of the data's structure but are also useful in clustering data objects that do not necessarily contribute strongly to a single distinct cluster and is still able to produce a partitioned data set into k clusters. This process is denoted as Algorithm 3 in the pseudo code below.



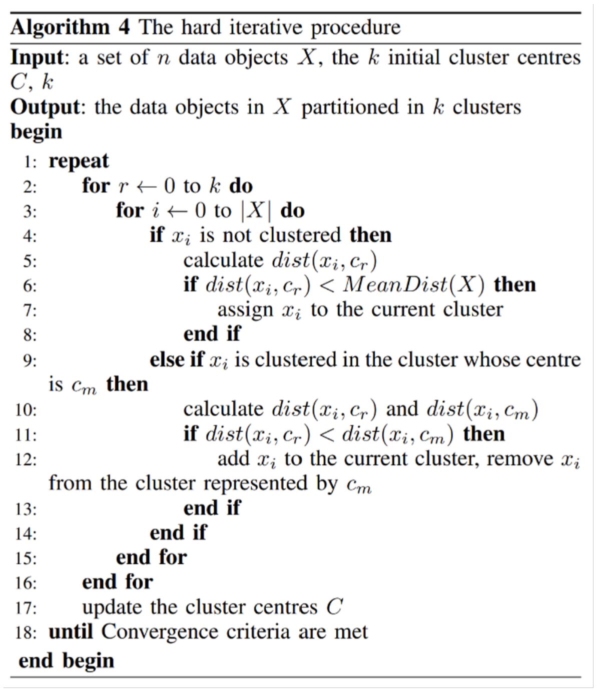
```
Algorithm 3 The iterative procedure
Input: a set of $n$ data objects $X$, the $k$ initial cluster centres $C$, $k$
Output: the data objects in $X$ partitioned in $k$ clusters
begin
repeat
    for $r \leftarrow 0$ to $k$ do
        for $i \leftarrow 0$ to $|X|$ do
            calculate $dist(x_i, c_r)$
            if $dist(x_i, c_r) < MeanDist(X)$ then
                assign $x_i$ to the current cluster
            end if
        end for
    end for
    update the cluster centres $C$
until Convergence criteria are met
end begin
```

Figure 5: Pseudo code for Algorithm 3

The input of Algorithm 3 takes in the clustered data with initial cluster centres C obtained from the same initialisation stage defined in Algorithm 1. The singular main difference in this approach is the absence of the $T_{c_{next}}$ notation which encodes each cluster centre distinctively. In the case where the Euclidean distance between the initial cluster centre and an unlabelled data object is not less than the $MeanDist(X)$, it is added to another cluster currently being formed in the form of a facet almost, and not a separate cluster which creates an overlapping cluster. The data object can therefore belong to more than one cluster and the cluster numbers are stored in the variable C. The cluster centres are then updated via the k means process where the mean value of the distances between all data points and their respective clusters is calculated, and assigned to the cluster representative in an iterative process that terminates when the mean value no longer changes [**29**]

3.1.6 The hard PFK means procedure version 1

This variant works almost as a combination of the PFK means and the overlapping PFK means methods in terms of producing distinct clusters but not via defining dissimilar points as cluster centres. The initialisation stage defined in Algorithm 1 is applied to initialise the cluster numbers k and the centres C. In this process, each data point is assigned to an appropriate cluster using the same criteria of : $dist(x_i, c_r) < MeanDist(X)$. In the case where the data point is not clustered it is immediately added to the current cluster being formed, and in the case where data objects are already assigned to a cluster, the Euclidean distance between its current cluster centre and the forgoing cluster centre is observed and the data object is assigned to a single cluster that constitutes towards the highest similarity. So the clusters are formed in an overlapping manner where the intersections between data objects and multiple cluster centres are removed in parallel in each iteration. The cluster centres are updated after each iteration by calculating the mean value of the data points in each respective cluster, for which the process terminates when the mean values have been stabilised. The outline of this procedure is detailed in the pseudo code in Algorithm 4:



```
Algorithm 4 The hard iterative procedure
Input: a set of $n$ data objects $X$, the $k$ initial cluster centres $C$, $k$
Output: the data objects in $X$ partitioned in $k$ clusters
begin
repeat
    for $r \leftarrow 0$ to $k$ do
        for $i \leftarrow 0$ to $|X|$ do
            if $x_i$ is not clustered then
                calculate $dist(x_i, c_r)$
                if $dist(x_i, c_r) < MeanDist(X)$ then
                    assign $x_i$ to the current cluster
                end if
            else if $x_i$ is clustered in the cluster whose centre is $c_m$ then
                calculate $dist(x_i, c_r)$ and $dist(x_i, c_m)$
                if $dist(x_i, c_r) < dist(x_i, c_m)$ then
                    add $x_i$ to the current cluster, remove $x_i$ from the cluster represented by $c_m$
                end if
            end if
        end for
    end for
    update the cluster centres $C$
until Convergence criteria are met
end begin
```

Figure 6: Psuedo Code or Algorithm 4

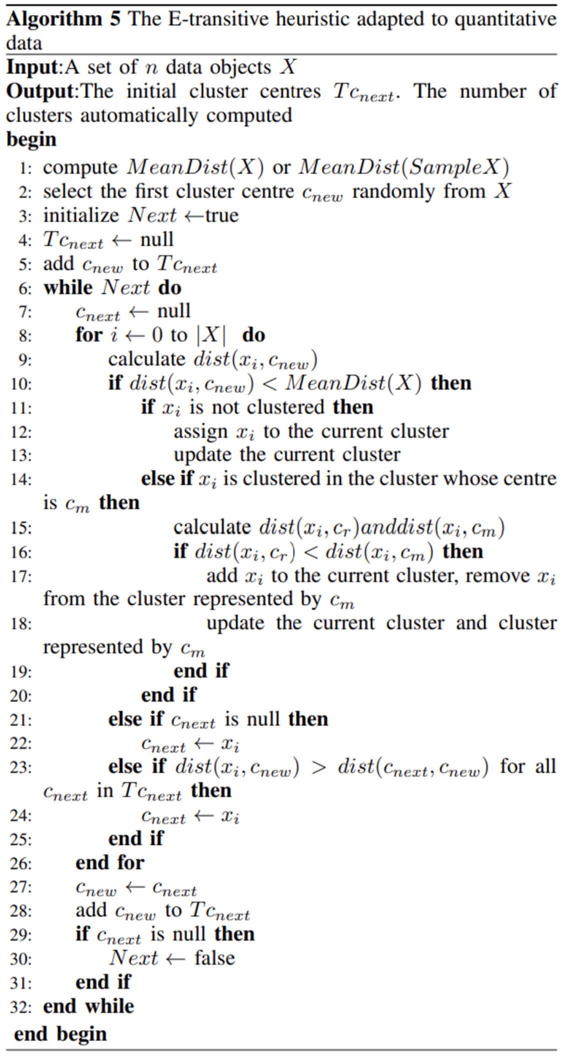
From figure it can be seen that data objects that are both clustered and not clustered are compared to the cluster being formed c_r in terms of their Euclidean distance. The data object in question x_i is added to c_r if this distance is smaller than the mean distance. If x_i is already clustered, then the distance between its assigned cluster \$c_m\$ and the current cluster c_r is compared, and x_i is removed from the c_m if the distance to $c_r$is smaller than the distance to c_m. Whereas Algorithm 3 only considered the distance between x_i and the current cluster being formed. In Algorithm 4, overlapping clusters are still produced hence the notation C as opposed to Tc_{next}, since data objects can be assigned to more than one cluster. This is because the removal of x_i from c_m is only considered is this distance is greater than $x_i's$ distance to c_r. If the Euclidean distance between x_i and c_m is greater than or equal to the Euclidean distance to between x_i and c_r then x_i remains clusters or assigned to both c_m and c_r.

3.1.7 The hard PFK means version 2

The original version of the PFK means method consists of applying the initialisation stage (Algorithm 1) with the traditional k means method (Algorithm 2). In the overlapping variant and the hard PFK means variant, they combine Algorithm 1 with Algorithms 3 and 4 respectively without applying the traditional \$k\$ means algorithm. This is where the hard PFK version 2 differs from the previous two variations. The hard PFK means method combines the initialisation stage (Algorithm 1) with Algorithm 4 and Algorithm 2, meaning that the output of the first hard iterative procedure is used as the input for the k means algorithm.

3.1.8 The E transitive heuristic applied to quantitative data

The intuition behind the E transitive method regards choosing the first cluster centre randomly and using a similarity metric to make pairwise comparisons to cluster data objects and define new cluster centres via their dissimilarity to previous cluster centres. In this application to quantitative data the initialisation stage (Algorithm 1) is applied to the data set and removes the intersections between the overlapping clusters via labelling data objects as clustered or not clustered and only enabling them to belong to one single cluster. Saori and Dafir defined their application to this method to quantitative data in the pseudo code below denoted as Algorithm 5:



Algorithm 5 The E-transitive heuristic adapted to quantitative data

Input:A set of n data objects X

Output:The initial cluster centres Tc_{next}. The number of clusters automatically computed

```
begin
 1: compute $MeanDist(X)$ or $MeanDist(SampleX)$
 2: select the first cluster centre $c_{new}$ randomly from $X$
 3: initialize $Next$ ←true
 4: $Tc_{next}$ ← null
 5: add $c_{new}$ to $Tc_{next}$
 6: while $Next$ do
 7:     $c_{next}$ ← null
 8:     for $i \leftarrow 0$ to $|X|$ do
 9:         calculate $dist(x_i, c_{new})$
10:         if $dist(x_i, c_{new}) < MeanDist(X)$ then
11:             if $x_i$ is not clustered then
12:                 assign $x_i$ to the current cluster
13:                 update the current cluster
14:             else if $x_i$ is clustered in the cluster whose centre is $c_m$ then
15:                 calculate $dist(x_i, c_r) and dist(x_i, c_m)$
16:                 if $dist(x_i, c_r) < dist(x_i, c_m)$ then
17:                     add $x_i$ to the current cluster, remove $x_i$ from the cluster represented by $c_m$
18:                     update the current cluster and cluster represented by $c_m$
19:                 end if
20:             end if
21:         else if $c_{next}$ is null then
22:             $c_{next} \leftarrow x_i$
23:         else if $dist(x_i, c_{new}) > dist(c_{next}, c_{new})$ for all $c_{next}$ in $Tc_{next}$ then
24:             $c_{next} \leftarrow x_i$
25:         end if
26:     end for
27:     $c_{new} \leftarrow c_{next}$
28:     add $c_{new}$ to $Tc_{next}$
29:     if $c_{next}$ is null then
30:         $Next$ ← false
31:     end if
32: end while
end begin
```

Figure 7: Psuedo code for Algorithm 5

By viewing the progressive steps of Algorithm 5 it can be deduced that the E transitive heuristic method applies the initialisation stage and the hard iterative procedure to produce distinct cluster centres where the clusters are updated within each iteration after they are modified. Steps 1 to 12 of Algorithm 5 convey the steps of Algorithm 1 with constructing the initial clusters iteratively assigning one data point to each cluster. In steps 11-19, the Algorithm 5 has integrated steps 9-13 from Algorithm 4 where the clustered data object's pairwise distance to the current cluster is compared and assigned accordingly. If the Euclidean distance to the new cluster is smaller than the Euclidean distance to the cluster that its currently assigned to c_m then it is removed from c_m. The clusters are then updated immediately after data objects have been assigned or reassigned. The Algorithm then concludes by implementing steps 12-23 from Algorithm 1 where the data objects not assigned to the current cluster or removed from a previous cluster are used to form the new cluster centres and the process terminates once all data objects have been clustered distinctively. This method therefore does not produce overlapping clusters or apply the k means algorithm after the initialisation stage, as the cluster updates are performed within each iteration.

3.2 A Criterion for Deciding the Number of Clusters in a Dataset Based on Data Depth

In this paper, the number of clusters in a data set are determined using a concept of 'data depth'. Baidari and Patil [**2**]define data depth as 'a statistical function that calculates the deepness or the centrality of a point in a data cloud'. A depth value of a large magnitude is indicative of a points deepness within a data cloud, whereas the converse demonstrates a points degree of being an outlier. Besides conveying a data object's position in a data cloud, depth functions can also be used to gain further insights to a data set such as shape and spread via depth regions.

Definitions In this section the metrics regarding depth and distance functions used in Baidari and Patils work are outlined in full.

3.2.1 Depth

The depth function used in this analysis in an adapatoin of the Mahalanobis depth (MD) function. This was selected for determining the depth of all data points x_i in a data set X. The original MD function is based on the Mahalanobis distance of a point x, with respect to a dataset X and was defined by P.C Mahalanobis [**25**] as:

$$MD(x \mid X) = \left[1 + (x - \bar{x})^T Cov(X)^{-1} (x - \bar{x})\right]^{-1}$$

As seen above, the MD function is estimated using the sample mean $\bar{x}$ of a point x ,and covariance matrix $Cov(X)$ of a data set X. As the mean estimate is sensitive to outliers this renders $\bar{x}$, $Cov(X)$ and hence the MD to be a non robust depth function [**2**]. Baidari and Patil therefore substitute the mean and covariance with X_i: Rousseeuw's minimum covariance determinant (MCD), and C: minimum volume ellipsoid (MVE) estimates respectively, in their alternative depth function:

$$MD(x; X) = \left[1 + (x - X_i)^T C^{-1} (x - X_i)\right]^{-1}$$

Regarding the MVE, this estimates the center and scatter of where he elements in the data set have the highest concentration, thus making it robust to the influence of outliers [**36**]

The MCD estimates the mean and covariance of a multivariate random variable, and works by assuming that a dataset of n observations has known $n - h$ outliers. The h observations are defined by covariance matrices that have the lowest determinant.The outliers $n - h$ can therefore be determined and excluded in the process of finding the mean and covariance estimates, which thus concludes the MCD robustness [**25**]

The depth function outlined in the above equation is therefore used to calculate the depth of all data points x_i in the kth cluster which corresponds to $D(x_i|k)$.

3.2.2 Depth median

The depth median μ_k is denoted by:

$$\mu_k = max_{1 \leq i \leq n_k} D(x_i|k)$$

where n_k represents the number of points within a k cluster. The depth median is used to denote the maximum depth value in a cluster.

3.2.3 Within-Cluster Depth

The within- cluster depth is calculated by taking the average of the differences between μ_k and $D(x_i|k)$. In this context,μ_k represents the centroid of a cluster k and $D(x_i|k)$ is the data depth (measure of centrality) of a point in a cluster k.

Thus the within- cluster depth was defined by Baidari.I and Patil.C as the average difference between each point in a data set and the center of a cluster k:

$$WD(k) = \frac{1}{n_k} \sum_{i=1}^{n_k} (\mu_k - D(x_i|k))$$

where n_k is the number of points within-cluster k.

3.2.4 Between-Cluster Depth

The between cluster depth, which is also referred to as the minimum between cluster depth, was defined as:'the minimum value of the average distance between every object in cluster k, and the center of every other cluster'. The calculation of the between-cluster depth is quite similar $WD(k)$, with the main contrast that every other cluster j is considered as opposed to a cluster k:

$$BD(k) = \min_{1\leq j<\leq m, j\neq k} \left[\frac{1}{n_k} \sum_{i=1}^{n_k} (\mu_j - D(x_i|k)) \right]$$

Depth Difference The depth difference is simply the difference between the within and between cluster depths outlined as follows:

$$DD(k) = WD(k) - BD(k)$$

3.2.5 Validity Index Value

The validity index value is used to define the optimal number of clusters and is defined by the average depth difference of k clusters:

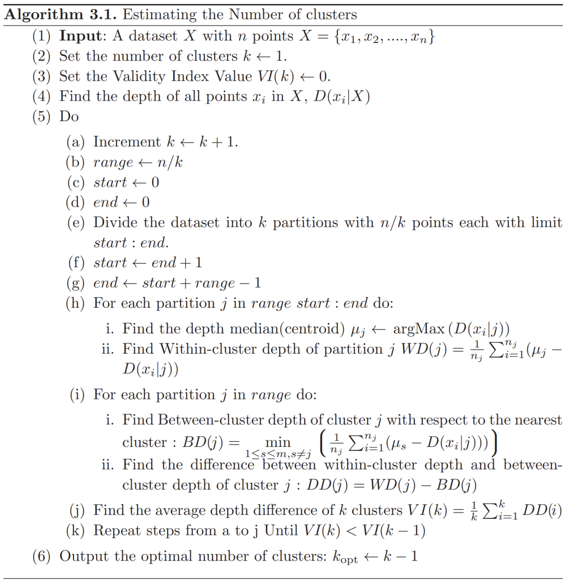
$$VI(k) = \frac{1}{k} \sum_{i=1}^{k} DD(i)$$

It can be seen from the definitions of the previous metrics that the final validity index is determined via the accumulation of the previous depth measures.

3.2.6 Method

To estimate the number of clusters in a data set, Baidari and Patil begin their algorithm [**2**] with a single input data set X containing n instances. The entirety of this data set is considered to be a single cluster, since no partitioning has taken place, therefore the initial value of k is set to 1. As there exists just one cluster, the $DD(K)$ value will simply be 0, so the VI index value of one cluster is set to 0.

The algorithm then proceeds to calculate the depth values $D(x_i|k)$ of each point x_i in the data set X as seen is step 4 in Figure 8.



```
Algorithm 3.1. Estimating the Number of clusters
  (1) Input: A dataset $X$ with $n$ points $X = \{x_1, x_2, ...., x_n\}$
  (2) Set the number of clusters $k \leftarrow 1$.
  (3) Set the Validity Index Value $VI(k) \leftarrow 0$.
  (4) Find the depth of all points $x_i$ in $X$, $D(x_i|X)$
  (5) Do
      (a) Increment $k \leftarrow k + 1$.
      (b) $range \leftarrow n/k$
      (c) $start \leftarrow 0$
      (d) $end \leftarrow 0$
      (e) Divide the dataset into $k$ partitions with $n/k$ points each with limit
          $start : end$.
      (f) $start \leftarrow end + 1$
      (g) $end \leftarrow start + range - 1$
      (h) For each partition $j$ in $range$ $start : end$ do:
          i. Find the depth median(centroid) $\mu_j \leftarrow \text{argMax}\,(D(x_i|j))$
          ii. Find Within-cluster depth of partition $j$ $WD(j) = \frac{1}{n_j}\sum_{i=1}^{n_j}(\mu_j -$
              $D(x_i|j))$
      (i) For each partition $j$ in $range$ do:
          i. Find Between-cluster depth of cluster $j$ with respect to the nearest
             cluster : $BD(j) = \min_{1\leq s\leq m, s\neq j} \left( \frac{1}{n_j}\sum_{i=1}^{n_j}(\mu_s - D(x_i|j))) \right)$
          ii. Find the difference between within-cluster depth and between-
              cluster depth of cluster $j$ : $DD(j) = WD(j) - BD(j)$
      (j) Find the average depth difference of $k$ clusters $VI(k) = \frac{1}{k}\sum_{i=1}^{k} DD(i)$
      (k) Repeat steps from a to j Until $VI(k) < VI(k-1)$
  (6) Output the optimal number of clusters: $k_{\text{opt}} \leftarrow k - 1$
```

Figure 8: Pseudo code for the estimation of k algorithm

Regarding the process outlined in step 5,the k value is increased by 1 as the data set is divided into k partitions, each with n/k points. The number of points in each partition is dictated by the 'start' and 'end' limits outlined in steps a)-g). The depth median for each j partition μ_j is set to the maximum value of $D(x_i|k)$, so that the subsequent within cluster depth is can be calculated for each partition. The depth medians are used to determine the nearby partitions for each j which is then used to calculate the between cluster depth $BD(j)$ for each cluster j with respect to the nearest cluster outlined in step (i)i. The depth difference $DD(j)$ is calculated and its average with respect to k clusters is found, which yields the value for the VI index for each k cluster. If this value is less than the value of the previous $k-1$ clusters, the algorithm terminates and $k-1$ is defined as the optimal cluster number. This condition is outlined as $VI(k) < VI(k-1)$, so the value $k-1$ indicates that average validity index of the k clusters has dropped from the previous $k-1$ clusters. The output k_{opt} therefore denotes the optimal cluster number.

3.3 Graph-based Data Clustering via Multiscale Community Detection

In this paper, Liu and Barahona adopt a graph-based method to clustering[**17**] where five methods of graph construction are considered to encompass both local and global features of a data set, which are then used to determine the number of clusters. Liu and Barahana apply a framework using Markov logic to model the connectivity of the graphs structure in matrix form, for which a multiscale community detection framework is applied to obtain a solution as an output of an algorithm. The multi scale element of their Markov stability framework enables to scan for communities across different levels of resolutions to determine robust clustering that are not sensitive to the parameters used in the graph constructions. An outline of their graph-based clustering approach to identify clusters is seen below:

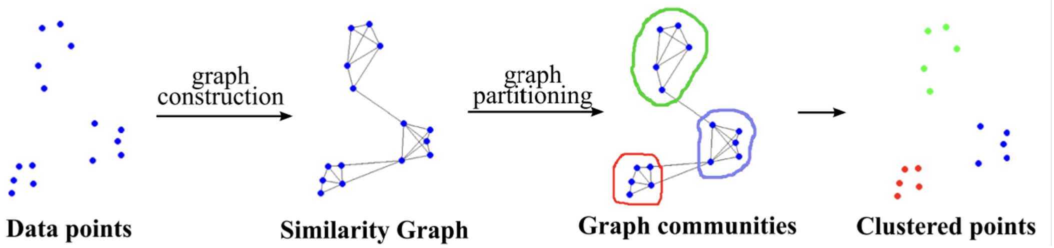


Figure 9: Diagram of the framework for graph based clustering

3.3.1 Graph construction

To construct each graph, each data point is encoded as a vector, where n samples each correspond to $\{\mathbf{y}_i\}_{i=1}^n$. A pairwise dissimilarity metric is assumed between the samples, as the local distances are considered only, as $d(i,j) \leq 0$. For graphs considered in this analysis the Euclidean distances are used for the dissimilarity metric. The Euclidean distance between two points \$(x_1,y_1)\$ and (x_2, y_2) is denoted by $d = \sqrt{\{(x\_2 - x\_1)^2 + (y\_2 - y\_1)^2\}}$. This conveys the difference between the x and y coordinates in each respective coordinate pair for which the horizontal and vertical distance components are obtained. Summing and square rooting the addition of these distances yields the distance between the coordinates which stems from the Pythagorean theorem of calculating missing triangle lengths. The Euclidean distance is therefore used to calculate the closest distance between a pair of points [**3**] .The main methods of graph construction utilised in Liu and Barahona's work consists of neighbourhood-based graphs, k nearest neighbour graphs and spanning trees.

3.3.2 Minimum spanning tree based methods PMST and RMST

The minimum spanning tree (MST) method of graph construction focuses on capturing the global features of the data's structure and therefore the geometry of the overall data set is considered..The intuition behind the minimum spanning tree approach is to connect all nodes of in the network to form an acyclic graph that spans across the entire data set[**17**].From this, a subset of the graph is extracted that is also of an acyclic nature, such that the sum of the weighted edges is minimised. To do this, the matrix of all pairwise distances $d(i,j)$ is regarded as the adjacency matrix, so the actual distances are considered as apposed to the matrix of 0's and 1's conveying the just the connectivity. Therefore this approach considers a weighted graph that is fully connected and the sum total of all these weights is minimises, meaning the graph connects all possible points using the shortest distances in the network, obtained a minimal global distance. The two minimum based spanning tree algorithms considered in Liu and Barahona's work considers a union of an MST graph and a kNN graph with a small value for k. This is done as a means of ensuring global connectivity of the data structure.

3.3.3 Perturbed minimum spanning tree (PMST)

The perturbed minimum spanning (PMST) method considers each individual data point y_i and induces a small amount of noise controlled by a parameter r, where d^k denotes an estimation of the local noise and the amount of noise is defined as a standard deviation of $s_i = rd^k(i)r \in [0, 1]$. d^k is the average distance of the k nearest neighbours to the selected data point [**39**]. For each aspect of the perturbed data, an MST is constructed and the PMST graph is therefore a combination of the all of the perturbed MST graphs alongside the original MST. The reason for this noise induction is to identify the robust global features of the graph as they will remain immutable or minimally changed in the presence of noise which therefore obtains the most important edges of the graph representative of the solid global features. Liu and Barahona note that the computational demand for this algorithm is of a heavy nature due to the fact that the distance matrix and MST are computed for each realisation of the perturbed data, so small fixed values for r and k are used. The following alternative to the PMST method is therefore proposed as a means of alleviating the computational cost of this MST based method.

Restricted mean survival time (RMST)

The RMST graph works by connecting a pair of nodes using the rule $d(i, j) < d^{max}_{path(i,j)} + \gamma(d^k(i) + d^k(j))$ where $d^{max}_{path(i,j)}$ denotes the longest edge between a pair of nodes and γ is a parameter that puts a weight on the local density a global property, meaning the average distance to the \$k\$ nearest neighbours of a pair of points is measured against the maximum distance between the pair of points. The γ parameter therefore controls the sparsity of the network in the same manner that the δ parameter in the CkNN method. This method suffices as good description of data according to Liu and Barahona [**17**] due to less computational time and the fact that it reconstructs data that is not evenly distributed across the entire network.

Regarding the following constructed graphs, these methods are neighbourhood based which effectively capture the data's inherent structure. The constructed graphs in question, involve the use of local distances for the purpose of capturing local and global measures.

3.3.4 Neighbourhood based methods

The neighbourhood-based methods used in this paper [**17**]are the ϵ-ball graph, the k- nearest neighbour graph (kNN) and the continuous k- nearest neighbours graph (CkNN). These methods construct graphs based on the local node distances, meaning that the shortest pairwise distance, in this case the Euclidean distance, between two nodes $d(i, j)$ is considered. The ϵ ball and k-nearest neighbour graphs are the two simplest neighbourhood-based methods as they connect nodes relative to the parameters ϵ and k. For the ϵ ball method, the Euclidean distance between each data sample \$x_i\$ and all other samples of data x_j is calculated. All distances that are less than the ϵ are used to establish connections between the respective nodes and are therefore considered as nearest neighbours using this metric. [**19**]The ϵ parameter is determined upfront and represents the radius of a ball, so each of the nodes are connected to the closest node that resides within the ball bound by ϵ. This method works best for data that is uniformly distributed [**35**]

Regarding the k^{th} nearest neighbour method, the Euclidean distances are calculated between a data point x_i and all other points in the data set. The set of the first k points that x_i is closest to which have the smallest Euclidean distances are selected and then connected to the point x_i. In contrast to the ϵ ball method, the kNN method connects a single point to a selection of its closest neighbours which is determined by the parameter k. [**19**]The drawbacks to these methods stated by Liu and Barahona are that they are sensitive to the ϵ and k parameters as well as that they convey mainly the local structure of the data. The parameters that these methods are heavily reliant on are also determined by the density of the data at hand, however the majority of data sets are not always uniformly distributed which would could thus render these parameters unreliable.

Therefore the continuous k nearest neighbour method (CkNN) was considered as a means to resolve this issue. The CkNN method was proposed by Berry and Sauer [**35**]as a means to capture the structure of complex data in the form of a manifold. A manifold in the context of clustering regards the surface in which data is found on. Manifolds refer to several geometric surfaces that can exist in multiple dimensions, and take a variety of forms such as a curved and warped, or surfaces that even have holes in them. [**39**]. Therefore the characteristics of manifolds will vary depending on the vantage point in which they are observed. So the reference to the local and global features of the data structure in Berry and Sauer's work concern the specific surfaces of the manifold where the neighbouring data points will resemble some structure, whereas the entirety of the data will vary with each manifold surface but the overall data's structure will have an inherit shape in terms of the density in which the nodes are arranged. The difficulties in interrogating large data confirming to complex manifolds are therefore apparent as manifolds do not always conform to a closed shape with definitive boundaries as the overall shape can be amorphous. Therefore the CkNN method is a novel method of constructing a single unweighted graph [**35**]from large data sets that maintain the integrity of the data's structure.

The method can be thought of as a combination of both the ϵ ball method and the kNN method in terms of connecting a single node to multiple nodes and using a pre defined tuning parameter as a threshold to determine which nodes are to be connected, with regards to their Euclidean distance. The advantages of this method are highlighted as producing a single graph that captures topological features simultaneously across multiple scales [**39**] where topology refers to the characteristics of the graph that remain invariant under continuous changes.

To determine the CkNN graph construction, the points x, y are connected if:

$d(x, y) < \delta\sqrt{d(x, x_k)d(y, y_k)}$

Where $d(x, y)$ is the Euclidean distance between the nodes $x$ and $y$ and x_k, and y_k,the respective k nearest neighbours of \$x\$ and \$y\$. The δ parameter is a continuous, unitless scale parameter that is best suited to determining graph structures that focus primarily on conserving topology and produces consistent geometry. The δ metric can be considered as the replacement of ϵ length parameter in the ϵ ball method [**35**]in terms of setting a threshold for which determines the number of connections in the network.

The delta parameter is a metric that conveys the density of the nodes across the entirety of the network. To connect a pair of nodes using the above formula, the Euclidean distance between each node $x$ and $y$ and their respective nearest neighbours x_k and y_h is calculated. The dot product of these two distances is then computed as a means of determining the similarity between the points, which is then normalised via the square root operation and then multiplied by the δ metric which scales this value with regards to the sparsity of the original network, thus maintaining the data's structure. If the Euclidean distance between the original nodes $d(x, y)$ is less than this value, then a pair of nodes is connected. So each node will be connected to a selection of its k nearest neighbours which considers the global features of the network upfront to ensure that the connections between nodes will construct a graph that is consistent with the data's existing structure.

The formula for constructing the CkNN network is based on the function of 'self -tuning' kernels [**35**] A kernel is a function that is able to quantify information regarding the similarity between a pair of subjects [**27**]in this context, a pair of nodes in an unweighted network. The kernel function takes the input as a coordinate as each node in the Euclidean space has an (x, y) component which is thus a vector, and computes the dot product between these points for which the output is a scalar numerical value. The magnitude of this scalar is used to determine the similarity between a pair of points, which is defined by the Euclidean distance between the nodes, for which a smaller value that is outputted by the kernel function will indicate that the points are more similar. which is therefore used to determine whether a pair of nodes should be connected.

The self-tuning kernel function that the CkNN formula is derived from is as follows:

$K(x, y) = \exp\left(-\frac{d(x,y)^2}{d(x,x_k)d(y,y_k)}\right)$

This kernel measure applies to weighted networks,by removing the exponential element and substituting an 'indicator' function, the kernel for unweighted networks is derived:

$$K(x, y) = I\frac{d(x, y)^2}{d(x, x_k)d(y, y_k)} < 1$$

These functions were introduced by L. Zelnick-Manor and P.Perona [**14**] to describe the similarity between a pair of points. In their paper, they introduced the prospect of using local scaling parameters that can self tune the between node distances according to the local features of the neighbourhoods surrounding these nodes. The selection of the scaling parameter is achieved by considering the local statistics of the neighbourhood of a point \$x\$, in this context the points $x$ and $y$ are considered and their local neighbourhood statistics are defined by $d(x, x_k)d(y, y_k)$. Zelnick-Manor and Perona demonstrate in their paper that the benefits of local scaling result in high similarities within clusters and low similarities across clusters which maximises the dissimilarity across clusters and maximises the similarities within clusters.

Berry and Sauer [**35**]built on this concept by applying the use of a self tuning scaling parameter to their method of graph construction where the considered data's structure was not limited to its local features. Instead their parameter δ considers the global features across the entire data set whilst incorporating the local features using the Zelnick-Manor and Perona's local similarity parameter [**14**]represented by the respective distances of each point and its kth nearest neighbour. Introducing the δ parameter to the above kernel function as their indicator function and rearranging the kernel function to consider the Euclidean distance between the nodes, yields Berry and Sauers formula for determining the connection between a pair of nodes:

$$K(x, y) = \{I\}\frac{d(x, y)^2}{d(x, x_k)d(y, y_k)} < 1$$

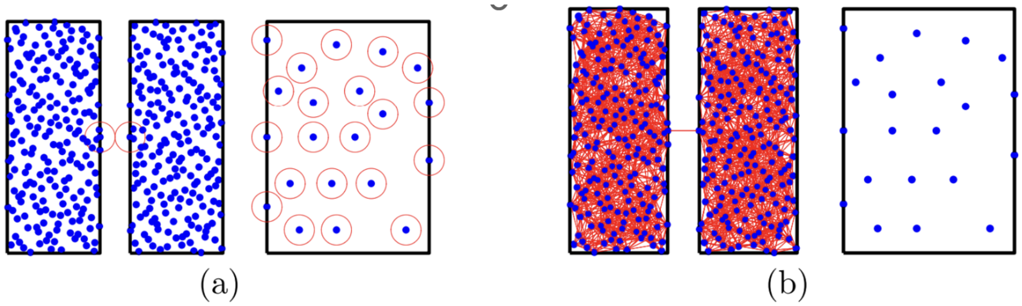
$$K(x, y) = d(x, y) < \{I\}\sqrt{d(x, x_k)d(y, y_k)}$$

$$d(x, y) < \delta\sqrt{d(x, x_k)d(y, y_k)}$$

The CkNN method is also noted as a multi scale method of graph construction as the kNN nearest neighbour element in the process suffices a simple density estimator as it underpins the local features of the graph that satisfies the following for small values of k:

$$\|x - x\_k| \propto q(x)^{-\frac{1}{m}}$$

where $q(x)$ represents the sampling density and m is the dimension of the data. In common data practices these characteristics may not always be known upfront, yet the CkNN method is able to incorporate the dimension component $\frac{1}{m}$ without needed to have an estimate for m in the underlying manifold [**35**] A simple visual representation given by Berry and Sauer of how the CkNN method is able to construct a single weighted graph whilst maintain the topological integrity of the data is illustrated below with clear comparison to how it outperforms the ϵ ball method and kNN method.



(a)

(b)

Figure 10:

In the above example, there are three rectangular spaces that represent true clusters of a data set, each containing two densely sampled data components and one sparsely sampled data component. The value for ϵ is indicated in figure 10 a) by the red ball surrounding the data points and used consistently for each cluster. Figure 10b conveys that the value is insufficient in connecting the sparse data and thus no connections are established as ϵ is too small, whereas in each of the dense data components, all data points within the radius ofϵ establishes false connections between the nodes and incorrectly connected the two distinct clusters which is further not representative of maximising the dissimilarity between clusters, a fundamental characteristic of clustering large data. Therefore, the data is either overly connected or under connected, and the ϵ value cannot be tuned to provide a fixed value that will correctly connect the nodes within and across these data clusters [**35**]

The same illustration is used in figure 11 below to demonstrate the kNN method of graph construction.

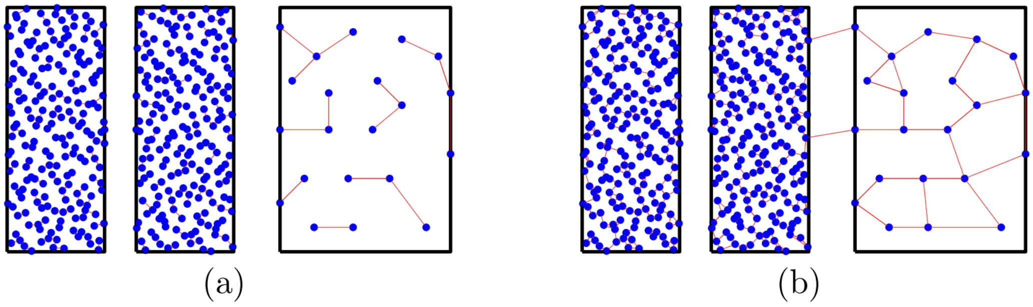


Figure 11:

In figure 2a. the nodes are connected to its nearest neighbour with $k = 1$, and it can be seen that it is unable to consider the topology of the entire data set as it fails to establish connections between all the regions in both the sparse and densely sampled data. In figure 11 b, each node is connected to two of its closest k nearest neighbours with $k = 2$ which establishes false connections between the data clusters and incorrectly connects the nodes. For values of $k > 2$, Berry and Sauer state that this problem will remain and is thus not able to correctly construct a network for the case of non-uniformly sampled data.

Using the same visual explanation, figure 13 conveys the CkNN method to the same data clusters and explicitly shows in this simple example how the CkNN method is able to construct single network that is inclusive of the correct topology of the data, across each cluster in both the dense and sparsely connected cluster spaces.

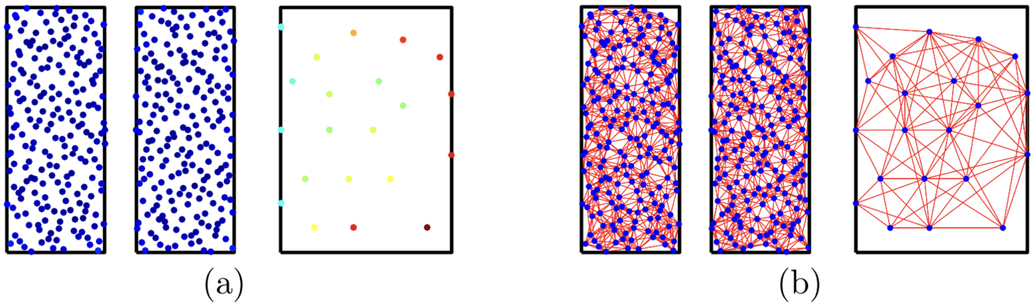


Figure 12:

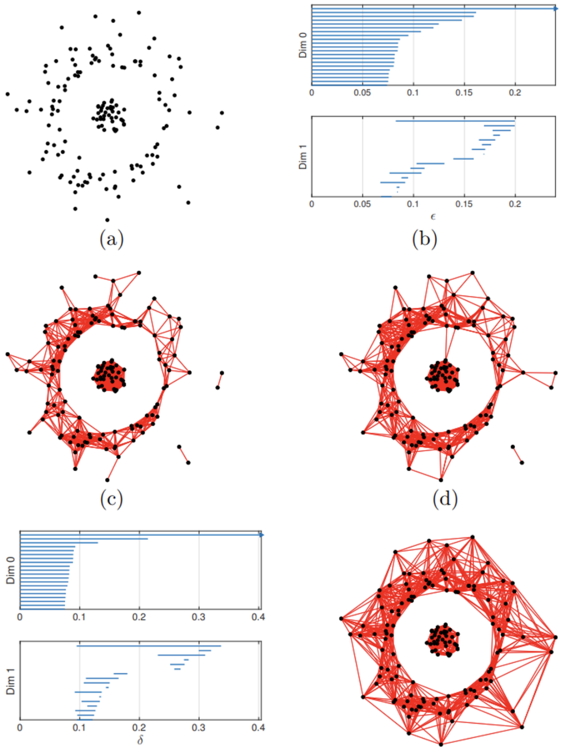
In figure 13a, the colouring represented in the sparse data sample represents the relative values of the bandwidth function $d(x, x_k)$, with a value for 10 for k and an optimal tuning of the δ parameter where blue conveys a low value and red a high value. Using the formula for connecting a pair of nodes $K(x, y) = d(x, y) < I\sqrt{d(x, x_k)d(y, y_k)}$

each of the nodes in the respect data samples are fully connected and triangulated, preserving the data's topology in each of the clusters, evident in figure 13b.

Regarding Berry and Sauers research, they were able to determine via rigorous mathematical proof that the existence of δ is guaranteed which constitutes to an unweighted graph preserving the manifold in the large data limit [**35**].This proof centers around the CkNN graph Laplacian convergence to the Laplace Beltrami operator, where the graph Laplacian refers to the matrix representation of the graph in terms of the connectivity between the nodes. The theory underpinning the δ metric states that the Laplacian convergence is achieved for any compact Riemannian manifold where a Riemannian manifold is a manifold with the Riemannian metric g that determines the lengths of the curves in the manifold. This g metric is therefore used to define the distance between the points in the network that are not limited to straight lines but also encompasses curved lengths.

Convergence of the graph Laplacian to the Laplacian beltrami operator means that the Laplacian beltrami operator is able to determine the Riemannian metric g , so that the entire topology of the Riemannian manifold i.e the networks structure, can be determined [**35**] The Laplacian beltrami operator is a generalisation of the Laplacian operator and can be considered as such when interpreting Riemannian manifolds in the Euclidean space [**13**]. The Laplacian operator itself is differential operator that acts on functions to describe the variation of properties with respect to their rate of change, in this context, the shape of the manifold. So convergence of the matrix representation of the CkNN graph to this Laplacian operator conveys that the structure of the data and the data's shape is known, therefore the δ metric can be determined to represent the entire graphs structure. Berry and Sauer outlined this via technical mathematical proofs which can be referred to in their work.

To derive the δ metric Berry and Sauer used a persistence diagram to visually convey the data's topology and identify the delta metric accordingly. A persistence diagram is a method of data representation that enables to interpret data across a multitude of scales without the need to rely on any singular specific scale. The persistence diagram conveys the connectivity between the data points as well as the 'holes' in the data's structure using a simple barcode type representation. These two features are referred to as dimension 0 and dimension 1 respectively. In order to construct the persistence diagram, Berry and Sauer first construct a graphical representation of the data using the Vietoris-Rips (VR) complex. The Vietoris rips are a means representing the topology of a data set in terms o# of the distance between each of the points using triangles. Berry and Sauer state that the 'VR complex is constructed inductively' meaning each triangle is added to each subsequent triangle until all of the triangles are connected to each other. [**35**]This complex or graphical structure then suffices as a visual portrayal of how each of the points in the data set are connected, preserving the structure of the data. In the example below, Berry and Sauer apply the ϵ ball and δ metric to a persistence diagram conveying the structure of some sampled data.



(a) (b)

(c) (d)

(e) (f)

Figure 13:

In figure 13a) the data's original structure is conveyed alongside the persistence diagram in figure 13b) where the ϵ metric is used to define the overall structure of the data. It is evident that the data's inherent structure consists of two connected components isolated by an annulus region which is referred to as a hole in the structure.

The persistence diagram in figure 13b) conveys the connections between the data points in 'Dim 0' and the holes in the data by 'Dim1' that resemble a barcode format. The length of these lines convey the ‘lifespan’ or persistence of each of these connections or holes which conveys how these features last or persist in the data over time. The length of the lines are used to determine the significance of these data aspects as the longer lines convey features that have higher persistence, which are stable features that do not distort with the presence of noise or fluctuations in the data and are therefore representative of established characteristics of the data. So regarding the results figure 3b. there is no evidence of a persistence region that is representative of these two connected components and a hole across both diagrams. Figure 3c) conveys the structure of the graph with an ϵ value of 0.135, which does not successfully connect the outliers whereas ϵ value of 0.16, will connect the outliers but not distinguish between the connected components by bridging the annulus region and will therefore remain in the large data limit [**35**]. Regarding the CkNN method in figures 3.e and 3.f the correct structure of the graph is conveyed with a δ metric in the range of 0.180 <δ ¡ 0.215. The graph construction conveyed in 3.f constitutes to a δ metric of 0.2. It is evident from figure 3.e that this δ range is representative of a region across both dimensions encompassing the connections and the holes in the graph that constitute towards two distinct connected components and the hole conveyed by the annulus region.

These persistence diagrams therefore enable a simple interpretable mean of selecting a δ value that accurately represents the data’s structure alongside a visual representation using the VR complex. After this method is applied, δ is then used to construct the CkNN graph using: $d(x, y) < \delta\sqrt{d(x, x_k)d(y, y_k)}$.

3.3.5 The Markov Process

The Markov stability method outlined in this study, refers to the use of random processes to determine the strength or quality of the communities constructed via graph construction. These communities are conveyed as a set of connected nodes where the connections within communities are dense and the connections between different communities are sparse [**21**]

The Markov process involved in this method regards a random walk which is a special type of Markov chain. A Markov chain is simply a model that is used to describe the random motion of an object in a distinct set of locations [**17**]which in this case is the undirected graphs constructed previously. The random walk itself is understood as a series of steps that an object takes randomly in a direction, where each subsequent step is not dependent on any steps taken previously.The direction of each step that the object takes is determined probabilistically [**9**] by finding its probability distribution function, a mathematical expression that describes likelihood that the object will step in a certain direction. [**33**].This model can be observed in under both discrete and continuous time scales, where discrete time (t)involves times scales observed at distinct intervals such as 2,4,6 and 8 seconds for example, and a continuous time scale is considered at all times of t that are above zero [**17**] . In this study, these random walks are used to uncover the structure of these communities [**15**] help determine the quality metric of these graph partitions and are observed in continuous time for optimisation purposes.

3.3.6 Graph encoding

These graphs to be encoded are unweighted and undirected, meaning that edges only correspond to a connection between the nodes. The connections are encoded using an adjacency matrix A, where the entries of the matrix A_ij} consist of 0’s and 1’s, with 1 corresponding to a edges that connect nodes i and,j and 0 to nodes that are unconnected. The degree of the nodes conveys the number of edges it is connected to and is summarised using the vector **d** where $d_i = \sum_j^n A_{ij}$, the sum of all the connections, and using a diagonal degree matrix D, where the diagonal entries correspond to $D_{ii} = d_i$ and all other entries are 0. The total number of edges in the network is denoted by $m = \sum_{I,j}(A_{ij}/2)$ [**17**].

3.3.7 Markov Stability

As mentioned previously, the Markov process involves a continuous time random walk for which the main statistics are the number of steps taken and the times that they occur. Using this information, the probability that a random walker will visit a specific node at a given time can be defined by a Poisson process. The Poisson process is a model for random events that occur independently, counted in integer form, where the time between events is unknown and no two events can occur at the same time, i.e the random walker cannot visit two nodes at the same time. The probability distribution associated with the Poisson model is of an exponential form, for which the time between traversed nodes is a random variable and distributed exponentially with an associated parameter of \lambda that controls the time scale. This \lambda value is normalised to 1, denoting the sum of all probabilities [**21**].

3.3.8 Markov Stabiltiy Defined as Vector Partioning

Liu and Barahona have defined the dynamics of their Markov stability framework by redefining the markov stability optimisation problem as a 'max -sum' partitioning of vectors that was established in the previous paper: multiscale community detection and vector partitioning [**18**]. In their previous work, they demonstrated that the Markov stability optimisation can be represented by an equivalent geometric interpretation for community detection using vector partitioning, where the time parameter is referred to as a resolution scale. By using this vector representation of the Markov process, it enables the optimised number of communities to be interpreted as an output of an algorithm and also incorporates a quality metric that additionally defines the strength of these communities which thus leads to better interpretability of the results. This vector partitioning approach was proposed by Liu and Barahona as means of community detection as it resembles the objective of the minimum cut problem for graph partitioning [**15**]. The minimum cut problem regards partitioning a graph into disjoint subsets such that the dissimilarity between these subsets is maximised, which is defined as the total weight of the removed edges In this case, the graphs in question are unweighted so the minimum cut concerns the minimum number of edges removed to achieve this partition. In short, the Markov process can be translated to vector notation to model the Markov process in the same manner that matrices are used to encode connections of the graph.

To describe the dynamics of the network, a $1 \times n$ row vector $\mathbf{p}$ is defined on the nodes or vertices of the graph, which constitutes to a numerical representation of the network's nodes. To define the vector p, the network is represented numerically via a Laplacian matrix. The Laplacian matrix L, is simply the difference of the degree matrix and the adjacency matrix: $D - A$, which constitutes to a matrix representation of the graph. The normalised form of the Laplacian is used in this analysis and is obtained by multiplying the degree matrix inverse by the adjacency matrix: $D^{-1}A$ which is denoted by M in Liu and Barahona's framework [**17**]. The dynamics that they have defined for this process is as follows:

$$\frac{d\mathbf{p}}{dt} = -p[D^{-1}L]$$

where $L = D - A$
therefore

$$\frac{d\mathbf{p}}{dt} = -p[D^{-1}[D - A]]$$

$$= -p[I - D^{-1}A]$$

where I is the identity matrix

Denoting M as $D^{-1}A$, the derivative simplifies to

$$\frac{d\mathbf{p}}{dt} = -p(I - M)$$

In this vector representation of the nodes, M is conveys the transition matrix which is a square matrix of the probabilities that the random walker will transition between the nodes, where 1 represents a certain transition and 0, no probability of transitioning. Each row of M represents an initial state (node) and the column, the state (node)that it transitions to. (I-M) is therefore the 'random- walk normalised Laplacian).

The vectors that are representative of the network's nodes are derived from the eigenvalue decomposition of the Laplacian matrix. The Laplacian matrix contains important information about the graphs structure which is encoded in the eigenvalues and eigenvectors which is necessary for the network partitioning. According to Luxburg, [**20**] the Laplacian matrix has the lowest eigenvalue of 0 which corresponds to an eigenvector of 1, so when a graph can be partitioned into p non overlapping communities, the amount of the 0 eigenvalues correspond to the number of components of p/ Whereas when a graph has a less distinctive layout with p communities, there exists only one eigenvalue equal to 0, $p-1$ eigenvalues larger than 0 and the remaining eigenvalues with a small deviation from 0. Therefore the eigenvalue decomposition of the Laplacian enables critical information of the graphs structure, [**14**] regarding the eigenvectors related to eigenvalues slightly greater than 0, which enables an effective mathematical encoding of the network.

Regarding the parameters of the Markov process defined on the node vectors, since the likelihood of a transition between node i to node j, is not dependent on any previous transition, therefore transition probabilities between the nodes in the undirected network are uniform across all neighbouring nodes The Markov process is also defined as ergodic which means that the process is non periodic and visits each node with a 'non zero frequency' over a long period of time, which therefore converges to a stationary distribution [**36**]. A stationary distribution of a Markov chain is a probability distribution that remains unchanged over time . It is usually denoted by the vector π of probabilities summing to 1, where solving the following will yield to the unique stationary distribution of the Markov chain: $\pi = \pi\mathbf{P}$ [**25**] Where $\mathbf{P}$ represents the transition matrix, for this process the transition matrix is represented by $\mathbf{M}$ .The analytical form of the stationary distribution of this Markov process is given by $\pi_i = d_i/2L$ where $2L = \sum_{j\epsilon N} d_j$ which denotes the total outgoing connection of node i and $d_i = \sum_{j\epsilon N} A_{ij}$ denoting two times the total weight of the outgoing connections in the network of node j [**17**]. Liu and Barahona have applied this to their framework by denoting the stationary distribution for their Markov process as follows: $\pi = \mathbf{d}^T/2m$ where m is the sum of all connections in the network and d is the degree of each node. So this stationary distribution that describes the likelihood of each node being traversed, is expressed as a ratio of each nodes degree to the sum total of the networks edges.

The autocovariance function is another parameter defined by Liu and Barahona that conveys more information about the random walk's behaviour. Autocovariance itself in this context, regards the covariance (variance between two variables) and describes the behaviour of the random walk. The autocovariance function is given by $B(t) = \prod P(t) - \boldsymbol{\pi}^T\boldsymbol{\pi}$ where $\prod = D/2m$ denotes the stationary distribution, and $P(t) = exp(-t(I-M))$ is an encoding of the transition matrix. The exponential component in the definition of $P(t)$ refers to the Markov processes evolving over a continuous time period and expressing the transition matrix as an exponential function of t, using the identity matrix and M, this definition of $P(t)$ conveys the changes in states across the network, with I representing the initial state and \$M\$ the matrix of transition probabilities. Using this definition the time parameter t is therefore understood as 'Markov time' [**18**]. By defining the autocovariance function with respect to the transition probabilities and stationary distribution, the dynamics of Markov process over continuous time can be analysed, which in the context of community detection, can be used to help determine the stability of the network.

Liu and Barahona [**17**] use their autocovariance definition to define a quality function that gauges the stability of partitions of their network. A partition of nodes is denoted as g, into c into non -overlapping communities, and the markov stabilty is evaluated using the following metric:

$$r(t,g) = \sum_{s=1}^{c} \sum_{i,j\epsilon g_s} B(t)_{ij}$$

In this definition, $B(t)_{ij}$ conveys how the interactions between nodes i and j vary with t, and that these interactions are observed across all pairwise nodes in each community.

Liu and Barahona state that a high value of g, constitutes towards the most robust and optimised partitions over time and are able to convey the multiscale community structure of the network [**18**], thus finding partitions with a high values of Markov stability are highly sought after. A higher value of Markov stability conveys a high probability that a random walker at a time t is observed within the same community as it was at $t = 0$ [**17**] as a function of t, Markov stability can be expressed as:

$$r*(t)\underset{g}{max}r(t,g)$$

given by the partition $g*(t)$ .Observing how the Markov process unfolds over time, gives valuable information about the graphs structure. For large values of t, global properties of the networks structure are uncovered as few communities are present, whereas at smaller values of t, more communities can be detected to convey more local information about the graph. To optimise the stability of each partition at each time interval, Liu and Barahona adopt Vincent D Blondel's [**5**]version of the Louvain algorithm.

3.3.9 The Louvain Algorithm

In Vincent D Blondel et al's paper: 'Fast unfolding of communities in large networks' [**5**] they developed an algorithm that determines the community structure of large networks, which was adopted by Liu and Barahona in their method to uncover the most robust and optimised partitions.

This method is of a heuristic nature that optimises a quality metric called modularity. The modularity metric Q measures the quality of a partition by measuring the density of the links within the communities on a scale between -1 and 1.

$$Q = 1/2m \sum_{i,j} [A_{i,j} - \frac{k_i k_j}{2m}] \partial(c_i, c_j)$$

Where $A_{i,j}$ is the weight of the edge between i and j and $k_i = \sum_j A_{i,j}$ is the sum of the weights for the edges connected to vertex i, $m = 1/2 \sum_{i,j} A_{i,j}$, c_i represents the community that i is assigned to and the ∂ function $\partial(u, v)$ is 1 where $u = v$ and 0 otherwise. In the case of undirected networks each edge is assigned to a weight of 1 [**21**] , therefore the number of edges that a node is connected to is considered only.

The Louvain algorithm proposed by Vincent D Blondel [**5**] suffices as an improvement to existing algorithms that have attempted to optimise modularity by finding high modularity based partitions of large networks in a short computational time which also reveal complete hierarchical community structures of the network, and enables a range of resolutions of community detections. Previous methods such as greedy algorithms, output partitions of lower values of modularity which are therefore not optimised, and other alternative methods output 'super-communities' or networks consisting nodes that have no significant contribution towards its structure.

Blondel achieved this by developing an algorithm that is to be run iteratively over a network, consisting of two phases. The first stage involves assigning a community to each node i, in the network. In order to build communities in the network, the algorithm begins by considering the neighbouring nodes j of i, and i is placed in the community of j. The modularity of this community shift is then evaluated in terms of gain, and the node i is placed in the community that constitutes towards the maximum, positive gain in modularity. In cases where no positive gain is observed, the node i remains in its its original community. This process is repeated until there are no more individual node transitions that will yield to a gain in modularity, and the first phase is completed. Regarding the output of the first phase, Blondel notes that the order in which the nodes are processed has no impact on the modularity but seems to affect the computational time for which this is worth further investigation [**5**]

As mentioned previously, the Louvain algorithm is an efficient approach to modularity optimisation which is largely attributed to the simplicity and efficiency of how modularity gain is computed. The modularity gain metric used to build communities is defined below as ΔQ, representing the change in modularity.

$$\Delta Q = \left[\frac{\sum_{in} + 2k_{i,in}}{2m} - \left(\frac{\sum_{tot} + k_i}{2m} \right)^2 \right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m} \right)^2 - \left(\frac{k_i}{2m} \right)^2 \right]$$

In this formula, the $=\sum_{in}$ conveys the sum of the weighted edges inside the community C and $\sum_{tot}$ the sum of weights of the edges that are incident to the nodes in C [**5**] k_i is sum of weighted edges from node \$i\$ to all nodes in C, m is the sum of weights of all edges of the network. When i is removed from its community and then moved to a neighbouring community, a similar expression is used to evaluate this modularity change. The efficiency of this method is therefore attributed to this simplicity of calculating ΔQ as it just considers the summations of respective edge weights.

The second phase of the algorithm regards constructing a new network where the nodes are the communities determined from the first phase. To construct this new network, the sum of the edge weights between two communities are used to construct the edge weights between new nodes of this network. Regarding nodes that are in the same community, their edges become 'self-loops', and thus the second phase of the algorithm is completed. The first phase of the algorithm is then applied to network outputted from the second phase and this combination of the two phases is denoted as a 'pass'.

Blondel states that these passes are iterated until there no longer exists a change of modularity and the maximum modularity is therefore attained. A visual interpretation of these phases of the Louvain algorithm are seen below:

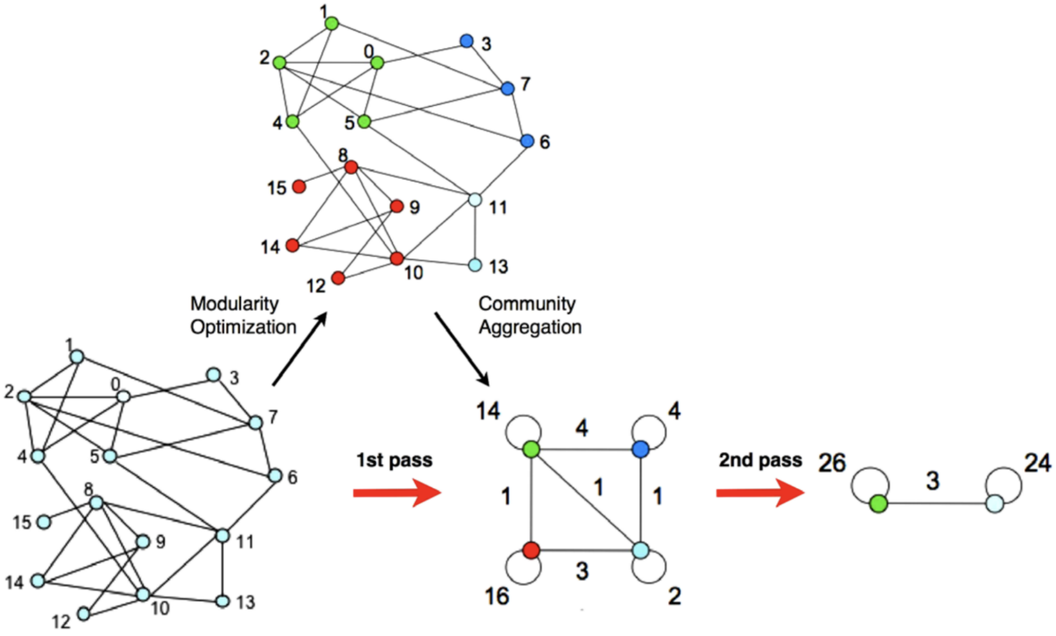


Figure 14: Visual representation of the Louvain Algorithm

In this above example taken from Blondel's fast unfolding communities paper [**5**], a visualisation of both phases and each pass of the algorithm are observed to denote its simplicity. The first phase of the algorithm is described as modularity optimisation, where each singular node is assigned to an individual community and then transitioned to others with regards to a positive, maximum gain in modularity. The second phase that uses the communities determined from the first phase to create nodes of a new network, is summarised as community aggregation. The passes are then visualised with where these new nodes are assigned to others with regards to modularity gain, and the self-loops indicate nodes that remain in their original community that are already representative of maximum modularity. The output of the second pass in this example conveys a network of communities connected by a weighted edge where no further adjustments can be made to increase the modularity of the communities in the network.

Blondel outlines that the advantages of their algorithm derive from the intuitive steps that are easy to implement, its fast computational time, simple calculations regarding modularity gain and change as well as minimal passes that need to be iterated in the final stage, as the weight of the computational time is concentrated on the first few iterations. It is noted that modularity optimisation is not sufficient in identifying communities that are smaller than a certain scale, therefore the introduction of a resolution limit on the detected community is advised. In Liu and barahona's work, [**17**] they have used time as a resolution parameter regarding their multi scale community approach.

As mentioned previously, the Markov process is observed in continuous time so their approach involves running the Louvain algorithm multiple times at each time t which constitutes to their method of computational maximisation of the Markov stability each partition g as of r. As a result, optimised partitions $g * (t)$ are detected at all time scales. Their method was able to determine the most robust partitions at robust scales which involved optimising Markov stability across long intervals of Markov time.

In order to achieve this, the dissimilarity between each detected partition was computed at times t and t'. the metric used to define the partition dissimilarity, is the variation of information (VI).

3.3.10 Variation of information

The variation of information metric is a theoretical criterion used to compare clustering's or partitions of the same data set. This criterion was invented by Marina Meilia in her paper:' comparing clusterings by the varition of information' [**40**]. Thus this comprehensive summary of the variation of information is an overview of the metric's derivation and its defintions denoted by Melia. In her reasearch, she states that the variation of information measures the loss and gain of information from changing between clusterings *C* and *C'*. The relationship between two clusterings is considered as an exchange of information, similar to how the quality of each partition was determined via the change in modularity as each node shifted between communities.

The information in each cluster and the information that a cluster gives about another is defined using the concept of entropy. At first, the probability of each node being assigned to a cluster C_k, is defined as $P(k) = \frac{n_k}{n}$ which conveys a uniform probability across all nodes. Meila defines this as a 'discrete random variable taking K values, uniquely associated to a clustering *C*'. The measurement of uncertainty that a node will be in a given cluster C is equal to the entropy of the random variable K which is defined as:

$$H(C) = -\sum_{k=1}^{k} P(k) log P(k)$$

Here $H(C)$ refers to the entropy of the clustering C which always takes a non negative value. A zero value of $H(C)$ denotes no uncertainty that a node will be assigned to a cluster C, meaning that there exists only one cluster that that node is to be assigned to. It is also noted that this uncertainty depends on the relative proportions of the clusters.

Regarding the information that one clustering contains about the other, this is defined as the mutual information between two clusterings. Meila uses probability theory to also define this metric as follows:

$$I(C, C') = \sum_{k=1}^{K}\sum_{k=1}^{K} = P(k, k') log \frac{P(k, k')}{P(k)P'(k')}$$

In this formula, the $P(k)$ and $P(k')$ represent the random variables associated with the clusterings C and C' respectively. $P(k, k')$ is therefore the joint probability distribution of the random variables in each of the two clusterings and is defined as:

$$P(k, k') = \frac{C_k \cap C'_k}{n}$$

This demonstrates the joint probability that a point in k and k' belong to a clustering C in C_k and C' in C'_k. To define the variation of information metric, the above definitions of the associated entropy of a clustering C, $H(C)$ and the mutual information between two clusterings $I(C, C')$ are used with regards to their properties. $I(C, C')$ can be considered as the 'reduction of uncertainty, averaged over all points'. Meila's explanation of $I(C, C')$ as uncertainty reduction stems from the idea that, if it is known upfront that a point is to be assigned the cluster C, then the uncertainty about the cluster C' is reduced, and when averaged across all points s equivalent to $I(C, C')$.

Given that the $I(C,C')$ is always non negative, never greater than the total uncertainty in a clustering and that the entropies of associated with each clustering: $H(C)$ and $H(C')$ are equal when one clustering is determined by the other, a quantity metric to compare the clusterings C and C' can be deduced as follows:

$$VI(C, C') = H(C) + H(C') – 2I(C, C')$$

Upon closer inspection the $VI(C, C')$ metric can be understood as the sum of the differences between the entropy of each respective clustering and the mutual information between them:

$$VI(C, C') = [H(C) - I(C, C')] + [H(C') - I(C, C')]$$

Thus denotes the variation of information between two clusterings, where the first term represents the loss of information from a clustering C whereas the second term reflects the gain of information regarding the clustering C', when a point is assigned to clustering C' from C. Meila also clarifies in this formula that each respective term denotes the conditional entropies $H(C|C')$ and $H(C'|C)$. These conditional entropies regard the entropy of each respective clustering based on or given the entropy of the alternative clustering [**40**]

3.3.11 Summary

The pipeline of Liu and Barahona's multiscale community detection method via Markov stability was outlined in one of their examples involving synthetic data. For this data set, they first constructed a weighted Cknn graph and optimised the Markov stability of the g nodes of each partition into c communities across a Markov time frame of $t \in [1, 1000]$. At each t the Louvain algorithm was run $n_L = 500$ times and the partition with the highest Markov Stability $g*(t)$ was recorded. Then the $VI(t)$ is used to compute the dissimilarity metric of the partitions in each n_L optimisations he values for which are stored in a matrix that convey the dissimilarity of the optimal partitions. The robustness of the partitions were further analysed by plotting graphs of all partitions uncovered by Louvain optimisations at each Markov time and one dimensional embedding of the VI distances. [**17**]

To summarise this method, a graph is first constructed from a data set for which the adjacency and degree matrices are extracted and subjected to calculations that correspond to a Markov stability framework. In these calculations the transition matrix for the respective Markov process is determined along with its auto covariance function in order to determine the $r(t, g)$ metric that quantifies the stability of each partition. The VI metric is also calculated alongside this to determine partitions that are the most dissimilarity. The Louvain algorithm is used to detect these partitions at different time scales to determine the most robust partitions. The robustness is also measured by the stability of the output of the Louvain algorithm in terms of the partition number.

Data Description

3.4 Data from Studies

In the studies outlined previously, both real and synthetic data sets were used for the parameter free clustering and criterion method, whereas the multi scale community detection method only used UCI benchmark real data sets. In the parameter free clustering method, the real data sets had clusters that varied from 2-4 with the exception of the letter recognition data set which had 26 clusters (see appendix). The synthetic data sets for this method were constructed with regards to varying the size of the data, the attributes and the cluster number but each in isolation. Therefore, variations with both a high cluster and attribute number were not considered.

Similarly for the criterion method, the number of clusters estimated only ranged from 2-4 with maximum considered attributes of 16. The data range considered up to 5000 values which established that the size of the data was not a limitation to the algorithm's performance, however this method did not explore high dimensional data in terms of cluster number and attributes. Lastly, the multi scale community detection method did not consider synthetic data and experimented on real data sets with up to 10 clusters.

So regarding the established performance of each algorithm, the number of attributes has not been a limitation, however the selection of clusters and attributes has conveyed a limited understanding of each methods performance.

3.5 Data Selection

To further investigate the performance of each method, the selection of data used in this study considered high dimensional data with regards to cluster number and attributes for both real and synthetic data sets, denoted as S1, S2, S3 and S4. The main parameters considered for variation across all data sets were the cluster number, attributes and the cluster separation/ data overlap. As each method is reliant on distance metrics to determine the number of clusters, the sparsity of the synthetic data was varied to explore its influence over the performance (refer appendix for data visualisation) . The selection of data are detailed in the table below.

Dataset	No. of Instances	No. of Attributes	No. of Clusters
Iris	150	4	3
Breast Cancer	569	30	2
Digits	1797	64	10
Forest Cover Type Subset	350	54	3
S1	700	10	8
S2	400	50	16
S3	900	40	25
S4	1200	12	8

Table 1: Details of the Real and Synthetic Data Sets

The aim of this analysis is therefore to explore more complex data structures and analyse each algorithms relative suitability. The forest covert type data featured in this analysis is a subset of the original data set consisting of 581012 instances (see appendix). The original number of 54 attributes is sustained but the number of classes has been reduced to 3 (see appendix). This was done to explore high dimensionality with regards to attributes only as the cluster number and instances are small, and the overlap is minimal. Regarding the number of instances, this parameter was not a key interest in the experiments as the studies have already established that the size of the data did not convey a limitation with regards to the algorithms performance and simply confirmed that it constituted to a higher computational time.

4 Results

In this section the results of each algorithms performance on the eight data sets are conveyed both individually and as a collective. As each method is unique to one another, the evaluation metric used to compare the performance of all three methods is their relative error:

$$RE = \frac{|k - k_{est}|}{k}$$

where k is the true number of clusters for each data set. The performance evaluation criteria therefore concerns each method's ability to output the correct number of clusters. The relative error metric was used from the criterion method as this is an estimation of the number of clusters k and does not have an actual cluster output with regards to cluster centers and their relative assigned data objects. The graph based multi scale community detection algorithm also conveys an output partition with regards to its Markov stability and variation of information between relative partitions which also does not convey an interpretable cluster output in terms of centers or partitioned data.

4.1 Parameter-free clustering Method

The PFK method and its variants were tested on the 8 data sets for which each algorithm was run five times. The following results represent the best result out of the five runs to replicate Slaouri and Defir's experiments. The evaluation of each method was performed with regards to the sum of squared errors and the root mean squared error in tables 2 and 3 respectively. Additionally,the silhouette score was calculated for each output in order to evaluate the cluster quality for which the results can be observed in table 4.

4.1.1 Clustering Performance

The PFK methods were only able to correctly cluster the Iris and Digits Data sets and failed to detect the correct cluster number for each of the remaining data sets. This suggests that the algorithms are not well suited to high dimensional data as it was unable to cluster data with larger cluster numbers. Additional evidence to support this was the fact that the method over estimated the number of clusters for the breast cancer and forest covert type subset data sets. Despite these data sets being low in classes and relatively low in instance number, this suggests that the high attributes are the key factor regarding dimensionality that limits the performance of these algorithms. This could potentially be due to the level of overlapping data objects in the data set, yet the synthetic data sets S1-S4 have well separated clusters will minimal to moderate overlap. Regarding the the Musk and Letter Recognition data sets (see figures 6 and 7 in appendix) used in the original study, they are high dimensional in terms of clusters and attributes as well as consisting of both minimal considerable data overlap. So it does not appear that data overlap is a limiting factor to the algorithms performance.

4.1.2 Evaluation of Individual PFK Variants

From tables 2 and 3, the lowest SSE and RMSE are highlighted in bold and are reported for the original PFK method consisting of algorithms (1+2) and the Hard iterative procedure consisting of algorithms (1+4). In each of these methods the output of the initialisation algorithm is used as the input to the algorithms that produce distinct clusters allowing each data object to belong to one cluster only. This suggests that the data objects in each data set are well separated and the clusters are distinct overall.

Regarding the quality of the cluster output, the silhouette score was calculated to summarise the compactness of the data objects to their cluster centers and detailed in table 4. The largest silhouette score: 6.93E-01 can be observed for the second and third variants of the PFK method for the Iris data set, with the lowest value at -7.52E-03 for the third PFK variant for the digits data. This conveys a moderate to poor cluster quality which is interesting as its representative of the only two data sets that these methods were able to cluster correctly. This also indicates that the iris data set results are reliable and too a good quality whereas the digits data set was clustered very poorly, rendering the result unreliable.

Moreover the second variant of the PFK method was not able to cluster the digits, S2 and S3 data sets at all which produced no output for the silhouette score. Upon further inspection of these results, it was found that the algorithm produced empty clusters, resulting in an output of a single cluster. Additionally, the S1 and S4 data sets for the second PFK variant produced empty clusters, and the output result for each data set was inclusive of 2-3 clusters for each run. As these data sets constitute towards the highest dimensional data in terms of clusters and attributes, this suggests that PFK method and its variants are not well suited to high dimensional data regardless of good cluster separation demonstrated in data sets S1 and S2

Lastly regarding the S1 and S2 data sets, the E transitive method was the best performing in terms of their respective silhouette score , which suggests that the E transitive method is best suited to well separated data as the S1 and S2 data sets have distinct clusters with very minimal overlap. However since it was still not able to cluster the data correctly, this suggests that the E transitive heuristic method is not well suited to high dimensional data in terms of cluster number.

The final results of these methods are represented in table 5 which details the cluster number relative to each data sets best performing algorithm and the RE between the actual and outputted cluster number.

Dataset	PFK Means	Overlapping PFK	Hard PFK1	Hard PFK2	Hard PFK3	E- transitive
Iris	1.13E+02	1.65E+02	**8.77E+01**	1.44E+02	1.90E+02	1.76E+02
BC	3.05E+07	4.90E+07	**2.74E+07**	6.26E+07	4.90E+07	4.87E+07
Digits	1.84E+06	3.01E+06	**1.81E+06**	2.97E+06	3.02E+06	3.03E+06
FCT	**5.07E+08**	1.13E+09	5.40E+08	1.36E+09	1.14E+09	1.01E+09
S1	4.04E+04	7.68E+04	**3.86E+04**	2.26E+05	7.88E+04	6.62E+04
S2	**4.79E+05**	7.64E+05	6.04E+05	9.08E+05	8.01E+05	7.39E+05
S3	7.71E+05	1.38E+06	**6.14E+05**	1.73E+06	1.41E+06	1.38E+06
S4	**1.16E+05**	2.10E+05	1.93E+05	5.30E+05	2.18E+05	2.35E+05

Table 2: Sum of Squared Errors (SSE)

Dataset	PFK Means	Overlapping PFK	Hard PFK1	Hard PFK2	Hard PFK3	E- transitive
Iris	8.61E-01	1.04E+00	**7.59E-01**	9.86E-01	1.13E+00	1.09E+00
BC	2.30E+02	2.93E+02	**2.19E+02**	3.32E+02	2.91E+02	2.92E+02
Digits	3.19E+01	4.08E+01	**3.17E+01**	4.06E+01	4.09E+01	4.11E+01
FCT	**1.19E+03**	1.79E+03	1.29E+03	1.97E+03	1.80E+03	1.76E+03
S1	7.56E+00	1.04E+01	**7.42E+00**	1.80E+01	1.06E+01	9.72E+00
S2	**3.42E+01**	4.32E+01	3.84E+01	4.76E+01	4.21E+01	4.30E+01
S3	**2.91E+01**	4.08E+01	3.21E+01	4.41E+01	3.92E+01	3.92E+01
S4	**9.80E+00**	1.32E+01	1.27E+01	2.10E+01	1.34E+01	1.40E+01

Table 3: Root Mean Squared Error (RMSE)

Dataset	PFK Means	Overlapping PFK	Hard PFK1	Hard PFK2	Hard PFK3	E- Transitive
Iris	5.38E-01	5.19E-01	**4.20E-01**	6.93E-01	6.93E-01	5.12E-01
BC	5.01E-01	4.66E-01	**2.77E-01**	6.61E-01	4.41E-01	4.90E-01
Digits	-1.01E-02	-2.80E-02	**-2.29E-02**	null	-7.52E-03	-6.42E-02
FCT	**3.10E-01**	2.88E-01	9.65E-02	2.77E-01	3.04E-01	1.94E-01
S1	6.34E-01	6.38E-01	6.35E-01	3.38E-01	6.34E-01	**6.75E-01**
S2	1.63E-02	1.20E-01	-8.65E-02	null	9.41E-02	**1.69E-01**
S3	1.17E-01	8.27E-02	**2.41E-01**	null	4.70E-02	1.09E-01
S4	**1.09E-01**	1.96E-01	-2.98E-02	3.58E-02	1.19E-01	7.73E-02

Table 4: Silhouette Score (SIL)

Dataset	PFK Algorithm	Output Cluster	Exact cluster	Relative Error
Iris	Hard PFK1	3	3	0.00
Breast Cancer	Hard PFK1	8	2	-3.00
Digits	Hard PFK1	10	10	0.00
Forest Covert Type Subset	PFK - Means	5	3	0.29
S1	Hard PFK1	6	8	0.25
S2	PFK - Means	10	16	0.38
S3	Hard PFK1	14	25	0.44
S4	PFK - Means	8	12	0.33

Table 5: Final Best Method Relative Results

4.2 Criterion for for Deciding the Number of Clusters Data Depth

The data depth based criterion method was applied to each of the eight data sets where the number of clusters k is estimated using distance metrics based on a robust variation of the Mahalanobis depth function. The criterion algorithm segregates the data into partitions of size n/k and calculates the a series of depth distance metrics relative to these partitions in order to derive a value for k that maximises the separation between these clusters. The separation is gauged via minimum values of the validity index, and the number of clusters k increases until there is no longer a decrease in the validity index, conveying that additional clusters does not improve the data separation. The algorithm therefore does not output clusters with assigned centers, and instead just provides and estimation for k which is results to the relative error RE as the only suitable evaluation metric for this method and consequently for comparison of across all three methods.

Out of the eight data sets, this method was only able to correctly output a value for k for the breast cancer data set. The details for each of the clusterings are conveyed in table 6 below.

Dataset	Output Cluster	Exact cluster	Relative Error
Iris	2	3	0.33
Breast Cancer	2	2	0.00
Digits	3	10	0.70
Forest Covert Type Subset	2	3	0.33
S1	2	8	0.75
S2	3	16	0.81
S3	3	25	0.88
S4	4	12	0.67

Table 6: Summary of the number of estimated clusters k for each data set

Given that this method under estimated the number of clusters for data with class numbers greater than 2, this suggests that the criterion method is not suited to high dimensional data, however it was still unable to correctly estimate k for the iris data set which is well separated and low in dimensionality. Regarding the digits data set, when computing the Rousseeuw minimum covariance determinant, this was not possible as the data was not full rank. There were some attributes that were linearly independent which resulted in the matrix not being invertible which highlighted a key limitation of this method. In an attempt to overcome this, principal component analysis (PCA) was applied which reduced the data set to two components. This PCA transformed digits data was therefore inputted into the criterion method. However the method was still not able to correctly identify the number of clusters despite the reduction in dimensionality. This therefore indicates that high dimensionality, significant data overlap, cluster separation and data sparsity are limitations to this method as well as data that is not full rank. Interestingly high dimensional data with regards to attributes is not implied to be a limiting factor as the breast cancer data set was correctly clustered and the forest covert type data subset was underestimating by 1 cluster, constituting to the lowest RE value in the experiments. Since the original study had a much higher success rate with clustering 16 out of the 20 data sets, for data with clusters that ranged from 2-4 and attributes that ranged up to 16, it can be deduced that the number of classes or clusters in a data set it a key limiting factor for this method.

4.3 Graph-based data Clustering via Multiscale Community Detection

In this method, graphs are constructed from the eight data sets for which their respective adjacency and degree matrices are extracted and subjected to the matrix calculations, used to determine the metrics needed to apply the Markov stability framework. Using the Markov stability framework, the transition matrix $P(t) = (-exp(I - M))$,where $M = D^{-1}A$ is the one step random walk transition matrix that corresponds to the random walk associated with the respective data set. The $P(t)$ matrix was used to construct a network which was used as the input of the Louvain algorithm As mentioned previously, time is used a resolution parameter in the Louvain algorithm to scan across the graph network and detect the most robust partitions. The optimal partition is determined from a high value of Markov stability $r(t, g) = \sum_{s=1}^{c} \sum_{i,j \in g} B(t)$ where $B(t)$ conveys the auto covariance matrix for the relative Markov process. and a low value of variation of information between g partitions at different times t and t': $VI(t, t') = VI(g^*(t)g^*(t'))$, which conveys the dissimilarity between partitions at different times. The graphs constructed for each of the data sets were the CkNN graphs with parameters $k = 7$ and $\delta = 1$ as the Markov stability framework was only applied to this network in the original study. Liu and Barahona established in their work that the four additionally considered networks were not well suited to cluster separation [**17**] With regards to replication of this method, it was not explicit in their study at which Markov times or the time frames for which the networks were scanned across for each data set, therefore in these experiments Markov time scales were considered from $0 \leq t \leq 3500$ on average for each data set. From this range, the selected Markov time that constitutes to the correct data partition with regards to a low $VI(t, t')$ value and a high $r(t, g)$ value was run 10 times through the Louvain algorithm. The stability of the output result was what determined the final cluster number. Overall this multi scale community detection algorithm was able to successfully detect the correct number of partitions for all data sets.

4.3.1 Iris data

Below is the CkNN network constructed from the Iris data set:

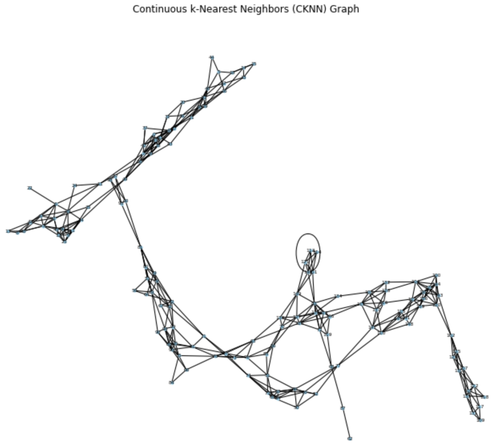


Figure 15: CkNN graph for Iris data

It is evident from this network that the shape of the Iris data is unstable. The delta values were adjusted in attempt to better capture the data shape, however there did not exists a range or value for delta that was able to graph the data that formed distinct clusters. Given that the method was not able to cluster the data accurately, it is evident that the method is sensitive to the graph construction which is dependent on the shape of the data. Therefore this clustering algorithm is not sensitive to the data's attributes or cluster number but geometric shape of the data , as the iris data set is quite low in dimensionality.

Markov time in seconds	No. of Partitions	Markov stability	Variation of information
$0 \leq t \leq 50$	7-3	$0.31 \leq 0.39$	$0 \leq 3.09$
$500 \leq t \leq 1000$	2	$0.12 \leq 0.20$	8.81E-16
$1000 \leq t \leq 1500$	2	$0.08 \leq 0.12$	8.81E-16
$1500 \leq t \leq 2000$	2	$0.05 \leq 0.07$	8.81E-16
$3000 \leq t \leq 4000$	3	-0.04 ≤ -0.03	1.11E-15 ≤ 0.11
$4000 \leq t \leq 5000$	5-4	-0.07 ≤ -0.04	$0.07 \leq 0.48$

Table 7: Markov time frames for Iris data

From Table 7, it can be seen that the Louvain algorithm is able to detect the correct number of partitions at both low and high Markov times but the reliable partitions exist in the $0 \leq t \leq 50$ with regards the high Markov stability, however this does not constitute to the lowest variation of information which conveys that the partitions are not robust. At time t close to zero the number of partitions should be large as this detects local graph features, and at higher Markov times the number of partitions should decrease as the global features are detected. The varying VI as well as the varying number of partitions suggests that the network does not fully stabilise across time which evident from Figure 1.

The selected Markov time was therefore selected at $t = 35$ which was run 10 times on the Louvain algorithm which produced a stable output in terms of partition value and Markov stability for which the output can be seen in Ta

Run	Partition	Markov Stability
1	3	0.33325513999113693
2	3	0.33325513999113693
3	3	0.333255139991137
4	3	0.3271600852073318
5	3	0.3302069666165575
6	3	0.33325513999113693
7	3	0.3302069666165575
8	3	0.33325513999113693
9	3	0.3302069666165575
10	3	0.333255139991137

Table 8: Final Output Result

4.3.2 Breast Cancer Data

The CkNN graph for the Breast cancer data is given below

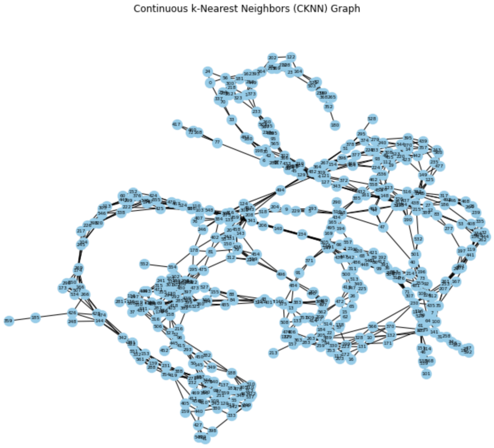


Figure 16: CkNN Graph for Breast Cancer Data

The Markov times that the network was scanned are cross are given in table 9.

Markov time in seconds	No. of Partitions	Markov stability	Variation of information
$0 \leq t \leq 100$	563-4	0.33-0.41	0.18-4.5
$500 \leq t \leq 1000$	3	0.22-0.26	0.06-0.94
$1250 \leq t \leq 1300$	2	0.25	-6.61E-16

Table 9: Markov time frames for Breast Cancer data

It is evident from the above table that the network stabilises quite quickly as at $t = 0$,the number of partitions detected was 563, but within 100 seconds the number of partitions detected was 4. For the correct number of partitions the Markov stability and VI index stabilises, each of which constitutes to the highest and lowest respective stable values. The selected Markov time was $t = 1266$ for which the number of partitions and Markov stability stabilises, conveying a robust output.

Run	Partition	Markov Stability
1	2	0.24914527076519183
2	2	0.24914527076519183
3	2	0.24914527076519183
4	2	0.24914527076519183
5	2	0.24914527076519183
6	2	0.24914527076519183
7	2	0.24914527076519183
8	2	0.24914527076519183
9	2	0.24914527076519183
10	2	0.24914527076519183

Table 10: Final Output Result

4.3.3 Digits Data

Regarding the original digits data set, the Louvain algorithm originally was only able to partition the data into 3 clusters. From previous analysis it was found that the digits data set has linearly independent attributes, therefore the PCA transformed digits data was used for the analysis as a means to determine the correct cluster number for which the algorithm was able to do so accurately. This conveys that PCA is a sucessful correction method to enable the algorithms perforamnce and data that is not full rank is a limitation to this method The original and PCA transformed CkNN networks are given below.

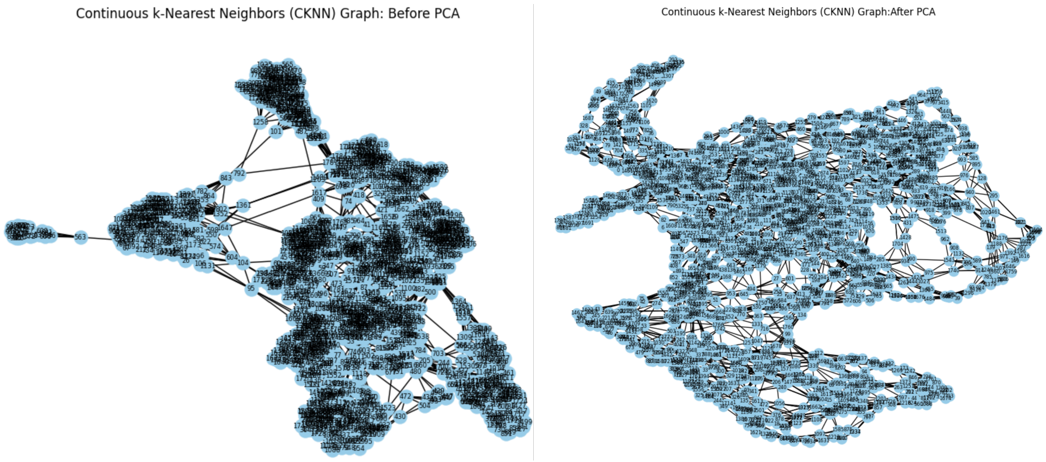

Continuous k-Nearest Neighbors (CKNN) Graph: Before PCA
Continuous k-Nearest Neighbors (CKNN) Graph:After PCA

Figure 17: CkNN graphs for the digits data set before and after PCA

The Markov times frames ran on the Louvain algorithm and their respective outputs are given below.

Markov time in seconds	No. of Partitions	Markov stability	Variation of information
$50 \leq t \leq 500$	5-3	0.19-0.33	0.08-5.73
$1000 \leq t \leq 1200$	3	0.12	0.01-0.15
$20 \leq t \leq 100$	8-5	0.30-0.37	0.76-5.45
$2 \leq t \leq 10$	21-11	0.40-0.45	0.60-4.50
$10.5 \leq t \leq 15$	11-9	0.39-0.40	0.79-1.14
$10.3 \leq t \leq 13$	11-9	0.40	0.74-1.24
$11.05 \leq t \leq 11.5$	11-10	0.40	0.61-1.14

Table 11: Markov time frames for Digits data

From table 11, it is clear that the network does not stabilise in terms of the number of partitions and variation of index values, however it does stabilise in terms of Markov stability. This means that the output partitions are reliable but not robust and therefore do not constitute to the best possible partitioning due to the unstable structure of the data's shape as seen in Figure 3.

Run	Partition	Markov Stability
1	11	0.40208789166528774
2	10	0.4031791393004087
3	11	0.40124292707189185
4	11	0.3995928457208694
5	10	0.40133550252109645
6	10	0.4056037387295903
7	10	0.4074591245174318
8	10	0.4053638469076779
9	11	0.39819366120833677
10	10	0.40546433782819086

Table 12: Final Output Result

From table 12 it can be deduced that the output for the Louvain algorithm that constitutes to 10 partitions is unstable. Therefore, the datas shape can be seen as a limiting factor to the algorithm's performance.

4.3.4 Forest Covert Type Data Subset

Figure 4 below shows the CkNN network for the forest covert data subset.

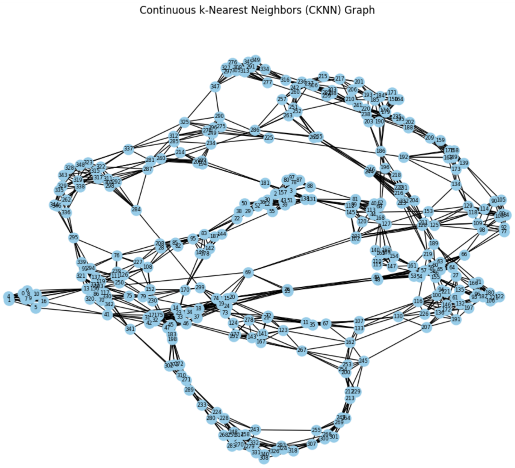


Figure 18: CkNN graph for forest covert type data subset

The Markov times frames ran on the Louvain algorithm and their respective outputs are given below.

Markov time in seconds	No. of Partitions	Markov stability	Variation of information
$0 \leq t \leq 1000$	4-2	0.12-0.25	0.02-4.78
$1000 \leq t \leq 1500$	4-3	0.09-0.11	0.07-0.67
$1500 \leq t \leq 2000$	4-2	0.09-0.12	0.03-1.02
$2000 \leq t \leq 3000$	3-2	0.12	0.08-0.57
$2000 \leq t \leq 2500$	3	0.11-0.13	6.66E-16-0.42
$2500 \leq t \leq 3000$	3	0.11-0.12	8.88E-16-0.39

Table 13: Markov time frames for Forest Covert Type Data Subset

Table 13 shows that the network stablises at a higher time probably due the high dimensionality of the data. The correct number of partitions is stablised after $t = 2000$ seconds. The interval where the markov time was selected was between $2000 \leq t \leq 3000$ as this constitutes to the lowest variation of information.

The selected time was $t = 2055$ seconds as this corresponds to the lowest outputted variation of information at 8.88E-16, for which the stability of the results can be observed below.

Run	Partition	Markov Stability
1	3	0.12525964635785905
2	3	0.11200846919714065
3	3	0.1176728570106732
4	3	0.11200846919714065
5	3	0.11200846919714065
6	3	0.11200846919714065
7	3	0.11200846919714065
8	3	0.11200848735961529
9	3	0.11200848735961529
10	3	0.11200846919714065

Table 14: Forest Covert Data Final Output Result

4.3.5 S1 data

Figure 5 conveys the CkNN network for the first synthetic data set.

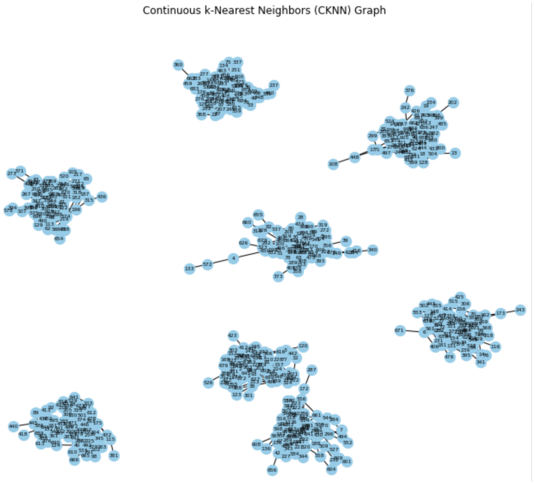


Figure 19: CkNN graph for S1 data

It can be deduced from Figure 4 that the CkNN method is able to capture the shape of the data for well separated and distinct clusters.

Markov time in seconds	No. of Partitions	Markov stability	Variation of information
$0 \leq t \leq 500$	8	0.47	1.78E-15 -4,46
$500 \leq t \leq 1000$	8	0.47	1.78E-15
$1200 \leq t \leq 1220$	8	0.47	8.88E-16-1.78E-15
$1200 \leq t \leq 1201$	8	0.47	8.88E-16-1.78E-15
$1000 \leq t \leq 1100$	8	0.47	8.88E-16-1.78E-15

Table 15: Markov time frames for S1 Data

Table 15 conveys that the network stabilises quite quickly and consistently detects 8 partitions evident from the distinct communities constructed from the CkNN graph. The Markov stability metric shows that the partitions are stable throughout the network yet the variation of information varies between exactly two values: 8.88E-16-1.78E-15. This shows that the algorithm is finding the same two partitions sets each time which again is representative of the data's well separated clusters.

There therefore exists multiple times for which the algorithm will output stable partitions.

Run	Partition	Markov Stability
1	8	0.46869749491153634
2	8	0.4686974949115362
3	8	0.4686974949115362
4	8	0.46869749491153634
5	8	0.46869749491153634
6	8	0.46869749491153634
7	8	0.46869749491153634
8	8	0.4686974949115363
9	8	0.46869749491153634
10	8	0.46869749491153634

Table 16: S1 Final Output Result

4.3.6 S2 Data

The CkNN graph below conveys S2 data. Although the CkNN method is able to distinguish between the clusters, is it not able to account for the minor overlap or the true sparsity of the clusters and therefore is not able to accurately represent the data.

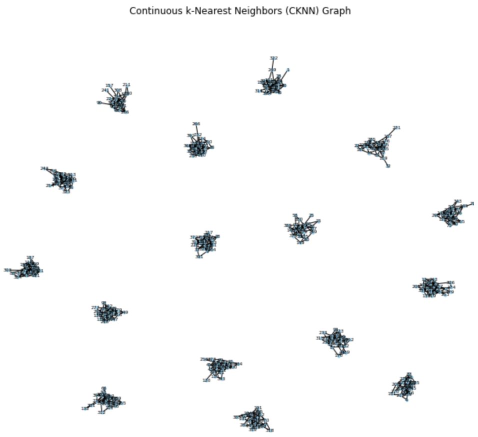


Figure 20: CkNN for S2 Data

Varying the δ parameter was not useful in producing a more reliable CkNN graph, as it either did not convey edges within the cluster partitions or conveyed an even wider cluster disparity. This therefore shows that the CkNN method is sensitive to data sparsity, not in terms of overlap but in terms of separation, even in the case of well separated data. The Louvain algorithm was run at $t = 920$ for which the output is given below.

Markov time in seconds	No. of Partitions	Markov stability	Variation of information
$0 \leq t \leq 100$	16	0.48	0-3.20
$0 \leq t \leq 10$	16	0.48	-8.88E-16-3.20
$0 \leq t \leq 1000$	16	0.48	0-3.20
$1200 \leq t \leq 1220$	16	0.48	0
$1000 \leq t \leq 2000$	16	0.48	0

Table 17: Markov time frames for S2 Data

From Table 17, its clear that the network stabilises quite quickly with consistency across the partition, Markov stability and Variation of information value which stabilises to 0. This means that the algorithm is outputting the same partitions each time as the Markov's time increases across the network, which also conveys that the graph and data shape is stable. However CkNN graph does not accurately capture the data's true structure and completely dismisses the overlap and does not accurately show the cluster separation. Therefore it is likely that these partitions do not constitute to accurate clusters despite being robust, and data objects have been assigned to the incorrect clusters.

The following output values constitute to the Markov time $t = 11$ but due to the stability of the network, this output will exist at multiple times after $t = 11$.

Run	Partition	Markov Stability
1	16	0.4843182108105294
2	16	0.4843182108105293
3	16	0.4843182108105294
4	16	0.4843182108105293
5	16	0.4843182108105293
6	16	0.4843182108105293
7	16	0.4843182108105293
8	16	0.4843182108105294
9	16	0.4843182108105293
10	16	0.4843182108105292

Table 18: S2 Final Output Result

4.3.7 S3 Data

The output for the bets CkNN graph for the S3 data is given below. The performance of this graph construction was identical to the S2 data in terms of not capturing the true disparity between cluster for considering the data overlap. The δ parameter was varied for which the output graphs can be seen in figures 4-8 in the appendix, and shows that the method either under connects or over connects the data. The CkNN method was not able to accurately represent this data.

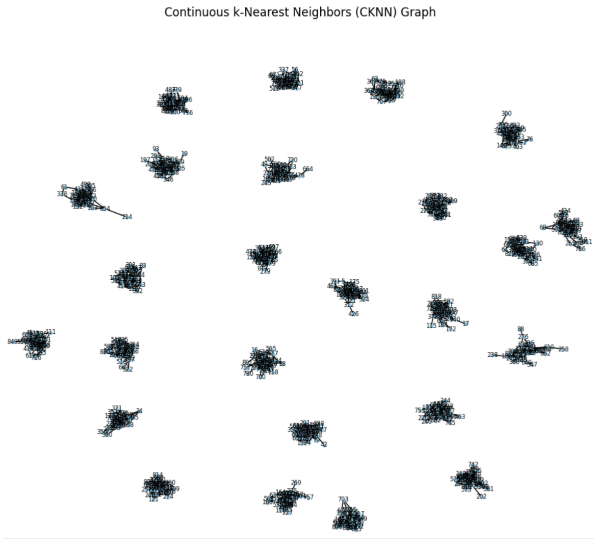


Figure 21: CkNN Graph for S3 Data

Markov time in seconds	No. of Partitions	Markov stability	Variation of information
$0 \leq t \leq 100$	25	0.49	0-3.7
$0 \leq t \leq 50$	25	0.49	-8.88E-16-3.55
$500 \leq t \leq 1000$	25	0.49	0-8.88E-16
$1200 \leq t \leq 1200$	25	0.49	0-8.88E-16
$1000 \leq t \leq 2000$	25	0.49	0-8.88E-16

Table 19: Markov time frames for S3 Data

Table 19 is almost identical to Table 16 in terms of output consistency and the variation of information range. The network stabilises quickly the output is consistent in the terms of partitions and Markov stability. The variation of information fluctuations between 0 and 8.88E-16, again show that the algorithm is ouputting the same partitions each time.

The output detailed below is taken at a Markov time of $t = 600$, but again this stability exists at multiple times across the network.

Run	Partition	Markov Stability
1	25	0.48996638035170514
2	25	0.48996638035170514
3	25	0.4899663803517052
4	25	0.4899663803517052
5	25	0.4899663803517052
6	25	0.48996638035170514
7	25	0.48996638035170514
8	25	0.4899663803517052
9	25	0.48996638035170526
10	25	0.48996638035170514

Table 20: S3 Final Output Result

4.3.8 S4 Data

The CkNN graph for the S4 data is given below. In this instance, the CkNN was about to better account for the data disparity, was inclusive of some overlap and partitioned the data into the correct number of clusters.

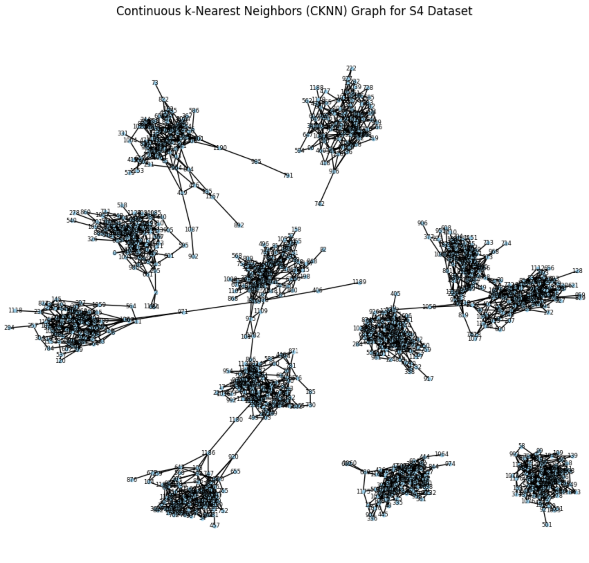


Figure 22: CkNN graph for S4 data

The stability of the output at different Markov times is given in Table 19 below.

Markov time in seconds	No. of Partitions	Markov stability	Variation of information
$0 \leq t \leq 100$	12	0.48	-8.88E-16 - 4.59
$0 \leq t \leq 1000$	12	0.48	0-8.88E-16
$1000 \leq t \leq 2000$	12	0.48	-8.88E-16-0
$2000 \leq t \leq 3000$	12	0.48	-8.88E-16-0

Table 21: Markov time frames for S4 Data

From table 21, it can be seen that again the network stabilises quite quickly with regards to the partition number and Markov stability. The variation of information again fluctuates exactly between two values 0 and -8.88E-16. This conveys that the network is stable and Louvain algorithm outputs the same set of partitions across increasing Markov time.

The stability of the output from 10 runs is given below at a value of $t = 50$. Due to the networks stability this output will be stable across a wide range of Markov times.

Run	Partition	Markov Stability
1	12	0.47913185273232006
2	12	0.47913185273232006
3	12	0.47913185273232006
4	12	0.47913185273232006
5	12	0.47913185273232006
6	12	0.47913185273232006
7	12	0.4791318527323202
8	12	0.47913185273232006
9	12	0.47913185273232006
10	12	0.47913185273232006

Table 22: S4 Final Output Result

4.4 Results Comparison Across all Methods

The performance of each clustering method with regards to the relative error for each outputted number of clusters is detailed below, with MSCD referring to the multi scale community detection method

Data Set	Cluster number	RE: PFK Means	RE: Criterion Method	RE:MSCD
Iris	3	0.00	0.33	0.00
Breast Cancer	2	-3.00	0.00	0.00
Digits	10	0.00	0.70	0.00
Forest Covert Type Subset	3	0.29	0.33	0.00
S1	8	0.25	0.75	0.00
S2	16	0.38	0.81	0.00
S3	25	0.44	0.88	0.00
S4	12	0.33	0.67	0.00

Table 23: RE across all methods

To summarise the performance of each method, the multi scale community detection is the best performing across all 3 methods with zero relative error for each cluster output. It should be noted that the reliability of the output for the Iris, and synthetic data sets is not Representative of the best cluster partition as the CkNN graphs were not able to accurately capture the data's integrity. The zero relative error for the digits data set in the parameter free clustering method also does not constitute a robust output as this had a very poor silhouette score for the cluster quality.

5 Discussion

The analysis of each of algorthim's individual performance is outlined below.

5.1 PFK Means Method

The parameter free clustering method and its variants performance was quite poor with regards to the selected data sets as it was only able to cluster the Iris data set reliably. In both the original and replicated results, the best out of five runs was reported which conveys that the algorithms do not produce stable or consistent results and its possible that the algorithm's will not output the correct result at all. This suggests that the random initialisation process is poor and and that even correct results could be representative of a convergence to the local optimum and not the best possible clustering. Although the PFK method and its variants are not dependent on initial parameters, they are however sensitive to the data disparity due to the mean euclidean distance metric being the threshold to determine the clusters in each method. As the pairwise distance between two data objects are iterative compared and assigned to clusters, this method could be sensitive to outliers and unsuitable for high dimensional and complex data which was established in the results of this study. Moreover these PFK methods that consider the entirety of the data set in the algorithm's input become computationally expensive in of large data even if it is of low dimensionality. So overall the PFK algorithm is not suited to high dimensional or large data sets with regards to large cluster numbers and attributes. The key limitation in these methods is the sparsity of the data as it was not able to correctly cluster well separated distinctively clustered data, which again renders this unsuitable for complex data. Overlapping data objects do not seem to be a limitation for this method as the original data sets used in the study had considerate overlap as well as the digits data set that was able to output the correct cluster number. Overall it is unclear what factors have the strongest influence or limitations to the algorithms performance and further investigation of this methods performance should elaborate on the influence of: data disparity, random initialisation of the first cluster center and sensitivity to the mean euclidean distance metric. Improvements towards this method should factor in the quality of the cluster output as well as outputting consistent stable results. The second variant of the hard iterative procedure constitutes to the greatest area of improvement/ further investigation as it outputs empty clusters in the case of high dimensional data. Overall this method is able to produce a reliable and correct output in the low dimensional data limit and is better suited to small data sets for which the data objects are low in sparsity, and have minimal to no outliers.

5.2 Criterion Method based on Data Depth

The results of this study as well as the original study convey that this method is not well suited to high dimensional data with regards to cluster size, as it was not able to determine the correct k value for a cluster number greater and than 4, and even had difficulty with determining k for 2 to 3 clusters. The number of attributes and size of the data set were not a limitation to this method however since this method takes in the input as a data collective it is computationally expensive in the case of large data. As the main distance metric was inclusive of a robust covariance matrix determinant, the algorithm is not able to compute an output on data that is not full rank as the matrix inverse is not attainable, which was uncovered when processing the digits data set. Principal component analysis was not an effective means to overcome this limitation so further exploration into this area is needed or its possible that this algorithm will not work on data that has linearly independent attributes. The fact the performance of the algorithm underestimated the number of clusters suggests that this method is best suited to low dimensional data in terms of cluster number. Other limitations of this method could be attributed to the initialisation process in terms of the data partitioning. Depending on the data set it may not always be possible to be partitioned into equal n/k sizes which would results in data objects wrongly assigned to clusters. The size and shape of clusters could also be an issue as the algorithm assumes that they can be equally partitioned which may not always be the case. This could also mean that the algorithm is not robust noisy data and outliers. Overall this method is not well suited to data that is partitioned into more than 4 clusters or large in size and areas for improvement should consider the initialisation process and data suitability to this method. It can however suffice as a good estimator for low dimensional data but it is difficult to gain a comprehensive evaluation of the performance when limited a single relative error metric. The criterion method is therefore thought to be sensitive to particular data structures and does not consider the global data structure.

5.3 Graph-based Data Clustering via Multi scale Community Detection

This method was very well suited to high dimensional data and produced robust, reliable outputs. Both the global and local features of the data set are accessible and the method considered the integrity of the entire data set and accounted for its shape. The main limitations of this method lie in the graph construction as the data extracted from this determines the algorithms performance. The CkNN graph was sensitive to the δ parameter which was not always able to be tuned to produce an accurate depiction of the data's structure in case of the first two synthetic data sets. This method struggled the most with data that was both sparse and overlapping as it was not able to capture the cluster separation accurately. it is also worth noting that it can be computationally expensive with regards to large data which is why the Louvain algorithm was run 10 times as opposed to n_L times in these experiments. While the method was able to determine the correct number of clusters in each case, they are not always representative of the most robust or accurate partition. Therefore regarding areas for improvement, the main area would lie in the graph construction to uncover graphs that are suitable to varying data shapes and sparsity. It could also be possible that certain data shapes may not be suitable for this method as this is what the parameters are most sensitive to.So investigation into the type of data that this method is best suited to can help to guarantee reliable results. There could also be advancements in the interpretability of the output as it is difficult to visualise the detected partitions. Certain metrics such as the silhouette score are not well suited to this method so investigation to evaluation criteria will be useful. Although the ARI index was used in the original study, this was only applicable to data sets for which the number of classes was known upfront, hence why there was no inclusion of synthetic data. This is not ideal as the ground truth is not always known upfront when clustering data.

6 Conclusion

To summarise the findings of this study, the comparison of the performance of the parameter free clustering , the criterion based on data depth and the graph based community detection methods deduced that the graph based multi scale community detection method was the best performing. It is thought that the previous two methods were not well suited to high dimensional data as they do not consider the entirety of the data's integrity, they are computationally expensive in the large data limit and are sensitive to their parameter initialisation processes. Despite their drawbacks, these methods still have proven efficacy for low dimensional data with a cluster number less than 4. Given the performance of the multi scale community detection algorithm it can also be deduced that these methods are not robust to outliers and are sensitive to the shape of the data, however these were also drawbacks in the graph based method. Overall despite these novel clustering methods and enhancements in the field of data mining, parameter sensitivity and high dimensional data still prove to be persistent difficulties when developing clustering algorithms. This study found that the sparsity of clusters and data objects are also a significant draw back to methods that are able to successfully cluster high dimensional data, as these algorithms as well as existing methods are mainly developed around distance metrics. Further enhancements should continue to focus on developing algorithms for data of specific shapes in the large data limit, where the global structure of the data is taken into consideration.

7 Appendix

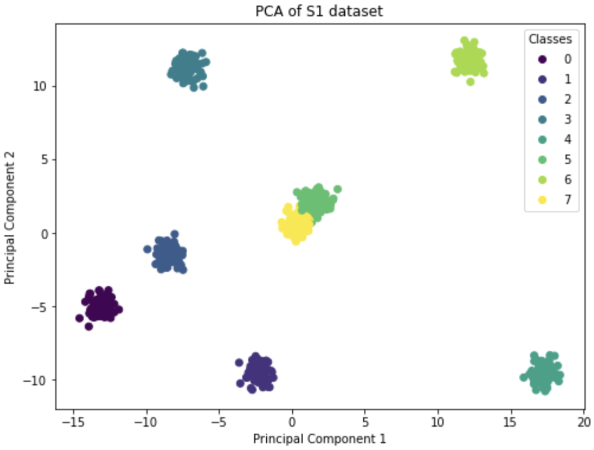


Figure 23: Scatter graph of S1 data

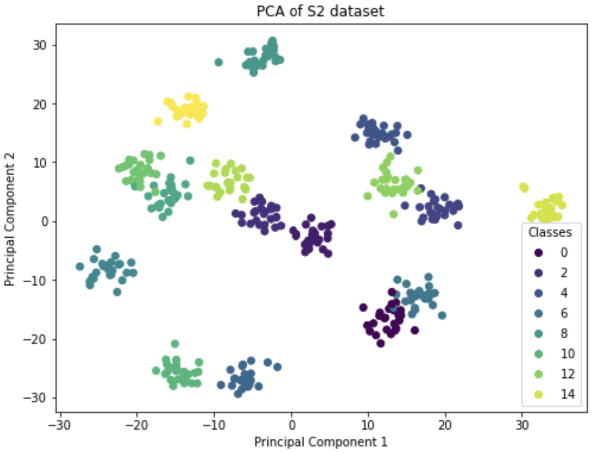


Figure 24: Scatter graph of S2data

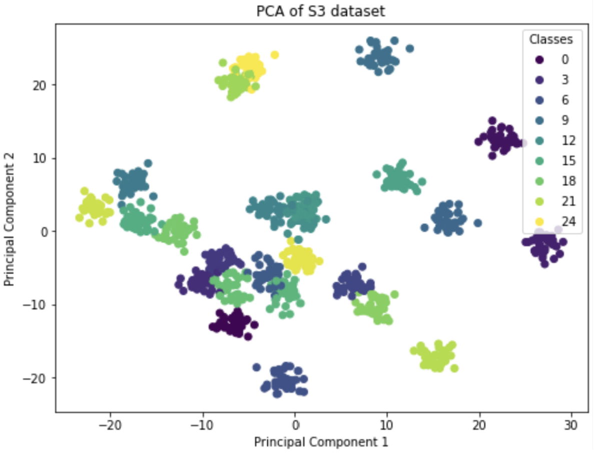


Figure 25: Scatter graph of S3 data

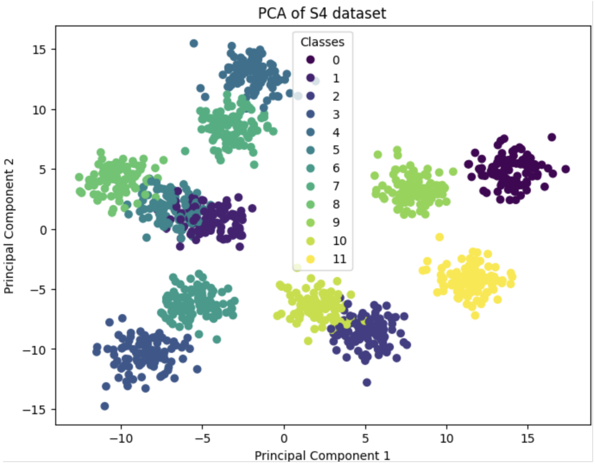


Figure 26: Scatter graph of S4 data

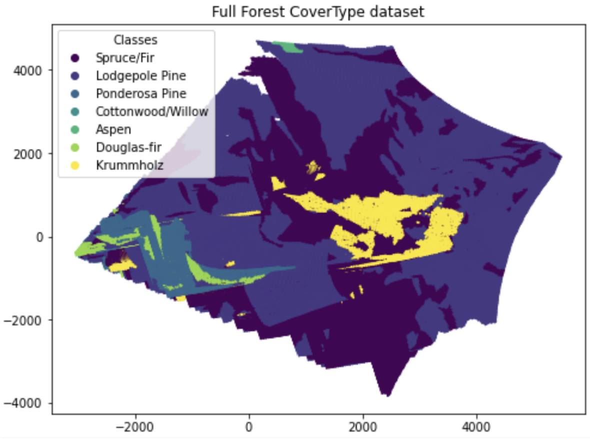


Figure 27: Full Forest Covert Type data

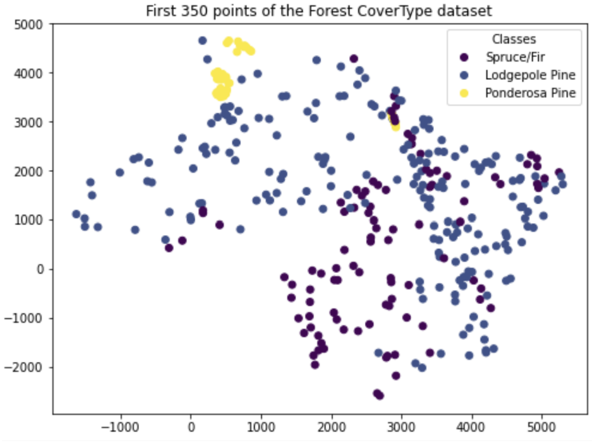


Figure 28: Forest Covert Type data Subset st

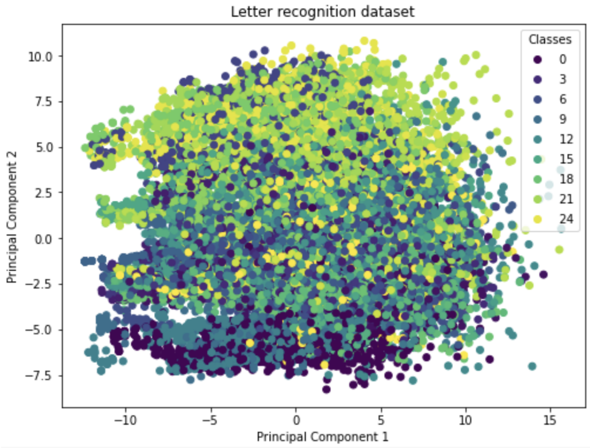


Figure 30: Letter recognition data set from original PFK study

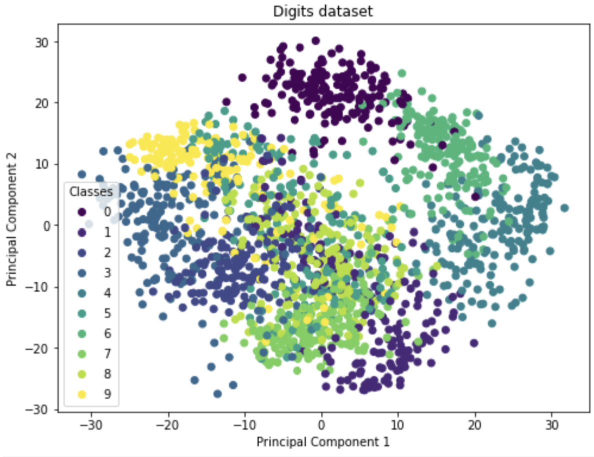


Figure 29: Digits data set

References

[1] Ah-Pine, Julien, and J.-F. Marcotorchino.Overview of the Relational Analysis approach in Data-Mining and Multi-criteria Decision Making Z.-U.-H. 2010. Usmani (eds) Web Intelligence and Intelligent Agents,InTech.

[2] Baidari,I and Patil,C., A Criterion for Deciding the Number of Clusters in a Dataset Based on Data Depth. Vietnam Journal of Computer Science2020.Vol. 7 (4), pp. 417–431.

[3] Banerjee, W. Role of Distance Metrics in Machine Learning - Analytics Vidhya - medium. *Medium.* (2021) https://medium.com/analytics-vidhya/role-of-distance-metrics-in-machine-learning-e43391a6bf2e

[4] Birant, D., \ Kut, A. ST-DBSCAN: An algorithm for clustering spatial–temporal data.Data \ Knowledge Engineering 60, 2006,pp 208–221.

[5] Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, 2008(10). https://doi.org/10.1088/1742-5468/2008/10/p10008

[6] Dabbura, I. K-Means Clustering: algorithm, applications, evaluation methods, and drawbacks. Medium. https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a. 2022

[7] Dafir, Z and Slaoui, S. C,. A Parameter-free Clustering Algorithm based K-means. International Journal of Advanced Computer Science and Applications, 12(3). 2021.

[8] Datanovia.Divisive Hierarchical Clustering - Datanovia.https://www.datanovia.com/en/lessons/divisive-hierarchical-clustering/. 2018

[9] Datta, S.What is a random walk? — Baeldung on Computer Science. Baeldung on Computer Science.2023.https://www.baeldung.com/cs/random-walk#:~̃text=A%20random%20walk%20can%20be,work%20is%20a%20random%20process.

[10] Delvenne, J.-C., Yaliraki, S. N., and Barahona, M. Stability of graph communities across time scales. Proceedings of the National Academy of Sciences, 2010, 107(29), pp. 12755–12760.

[11] Ester,M.,Kreigel,H.,Sander,J and Xu,X.Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications. Data Mining and Knowledge Discovery, 2, 1998,p169–194.

[12] Fleshman, W.Spectral Clustering - towards data science. Medium.https://towardsdatascience.com/spectral-clustering-aba2640c0d5b. 2022.

[13] Hua, J., Hu, J., & Zhong, Z. Near-isometric motion analysis using spectral geometry.2020 In *Elsevier eBooks* (pp. 29–44). https://doi.org/10.1016/b978-0-12-813842-7.00012-7

[14] L. Zelnick-Manor and P. Perona, Self-tuning spectral clustering, Adv. Neur. Inf. Proc. Sys. 2005

[15] Lambiotte, R., Delvenne, J., & Barahona, M. Laplacian dynamics and multiscale modular structure in networks. *arXiv (Cornell University)*.2008. http://arxiv.org/abs/0812.1770

[16] Lambiotte, R., Delvenne, J.-C., & Barahona, M.Random walks, Markov processes and the multiscale modular organization of Complex Networks. *IEEE Transactions on Network Science and Engineering*, *1*(2), 76–90.2014. https://doi.org/10.1109/tnse.2015.2391998

[17] Liu, Z. and Barahona, M. Graph-based data clustering via multiscale community detection. Applied Network Science, 2020, 5(1), pp.1-20

[18] Liu, Z., and Barahona, M. Geometric Multiscale Community Detection: Markov stability and vector partitioning. *Journal of Complex Networks*, *6*(2), 157–172.2017. https://doi.org/10.1093/comnet/cnx028

[19] Lu, H., Halappanavar, M., & Kalyanaraman, A.Parallel heuristics for scalable community detection.2014. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1410.1237

[20] Luxburg,U.,A Tutorial on Spectral Clustering. Statistics and Computing, 17 (4), 2007.

[21] Masuda, N., Porter, M. A., & Lambiotte, R. Random walks and diffusion on networks. *Physics Reports*, *716–717*, 1–58. 2017. https://doi.org/10.1016/j.physrep.2017.07.007

[22] Nikaj, K. and Ifti, M. Markov Stability Analysis and Community Structure in Social Networks. In Journal of Physics: Conference Series (Vol. 1548, No. 1, p. 012003).2020

[23] Nikaj, K., & Ifti, M.Markov stability analysis and community structure in social networks. IOP Publishing. *Journal of Physics: Conference Series*, *1548*(1), 012003.2020. https://doi.org/10.1088/1742-6596/1548/1/012003.

[24] Pai, P.Hierarchical clustering explained.Towards Data Science.Medium.https://towardsdatascience.com/hierarchical-clustering-explained-e59b13846da8. 2021

[25] Rand R. W,. Some Multivariate Methods, chapter 6. Introduction to Robust Estimation and Hypothesis Testing (Fifth edition) 2021, Pages 253-350

[26] Schaid, DJ. Genomic similarity and kernel methods I: advancements by building on mathematical and statistical foundations. Hum Hered. 2010;70(2):109-31. doi: 10.1159/000312641. Epub 2010 Jul 3.

[27] Schaub, M. T., Delvenne, J., Yaliraki, S. N., & Barahona, M.Markov Dynamics as a zooming lens for multiscale community detection: non Clique-Like communities and the Field-of-View limit. *PLOS ONE*, *7*(2), e32210.2012. https://doi.org/10.1371/journal.pone.0032210

[28] Sharma, A. {How to master the popular DBSCAN Clustering algorithm for machine Learning}. Analytics Vidhya.2023. https://www.analyticsvidhya.com/blog/2020/09/how-dbscan-clustering-works/\:\ :text=DBSCAN\

[29] Slaoui, S. C., & Dafir, Z. A Parameter-free Clustering Algorithm based K-means. *International Journal of Advanced Computer Science and Applications*, *12*(3). 2021.https://doi.org/10.14569/ijacsa.2021.0120372

[30] Slaoui, S. C., Dafir, Z., & Lamari, Y. E-Transitive: an enhanced version of the Transitive heuristic for clustering categorical data. Procedia Computer Science, *127*, 26–34. 2018.https://doi.org/10.1016/j.procs.2018.01.094

[31] Subasi.,A. Practical Machine Learning for Data Analysis Using Python. 2020

[32] Tanner, G.{Density-Based Spatial Clustering of Applications with Noise (DBSCAN)}. Machine Learning Explained. https://ml-explained.com/blog/dbscan-explained. 2020.

[33] The Editors of Encyclopaedia Britannica.Markov process.Stochastic Process, Probability Theory & Random Walks. Encyclopedia Britannica.2023. https://www.britannica.com/science/Markov-process

[34] The Editors of Encyclopaedia Britannica.*Random walk*.Stochastic Process, Probability & 2023.Diffusion. Encyclopedia Britannica. https://www.britannica.com/science/random-walk

[35] Tyrus,B Sauer,T. Consistent manifold representation for topological data analysis. 2016. *Foundations of Data Science*. V

[36] an Aelst, S., & Rousseeuw, P. J. Minimum volume ellipsoid. WIREs Computational Statistics, 1(1),2009 pp71–82.

[37] Walrand, J., and Varaiya, P. *High-Performance Communication Networks* (2nd ed). 1999, http://ci.nii.ac.jp/ncid/BA46653957

[38] Young, H. P.Condorcet's Theory of Voting The American Political Science Review, vol. 82, no. 4, 1988, pp. 1231–44. JSTOR, https://doi.org/10.2307/1961757

[39] Zemel, R. S. and Carreira-Perpiñán, M. Á. Proximity graphs for clustering and manifold learning. *Neural Information Processing Systems*, *17*,2004,pp 225–232.

[40] Meilă, M.Comparing clusterings by the variation of information. In *Lecture Notes in Computer Science*. 2003,pp. 173–187.