Using Sequential Deviation to Dynamically Determine the Number of Clusters Found by a Local Network Neighbourhood Artificial Immune System

A. J. Graaff[a,1,*], A. P. Engelbrecht[a,1]

[a] *Computational Intelligence Research Group (CIRG), Department of Computer Science, University of Pretoria, Lynnwood Road, Hillcrest, Pretoria, 0002, South Africa*

Abstract

Many of the existing network theory based artificial immune systems have been applied to data clustering. The formation of artificial lymphocyte (ALC) networks represents potential clusters in the data. Although these models do not require any user specified parameter of the number of required clusters to cluster the data, these models do have a drawback in the techniques used to determine the number of ALC networks. This paper discusses the drawbacks of these techniques and proposes two alternative techniques which can be used with the local network neighbourhood artificial immune system. The end result is an enhanced model that can dynamically determine the number of clusters in a data set.

Keywords: dynamic clustering, sequential deviation detection, immune networks, clustering performance measures

1. Introduction

A challenge in data clustering is to determine the optimal number of clusters in the data set. An approach to validate the number of clusters formed is to visually present the clustering results. In multidimensional problems where the number of dimensions is greater than three, visualization of the formed clusters becomes difficult [1, 2]. Another approach to determine the optimal number of clusters is to execute the clustering algorithm multiple times, each time with a different number of clusters and validating the clustered data set with a cluster validity index. The cluster validity index is then plotted as a function of the number of clusters obtained for each execution of the algorithm. The number of clusters generated from the input parameters with the highest (or lowest) cluster validity index is then selected as the optimal number of clusters [3, 4]. A drawback of the multiple execution approach is that the technique is computationally expensive and time consuming. Therefore a clustering technique or model which can dynamically determine the number of clusters in a data set and which is computationally inexpensive will have an added advantage. Section 2 gives a formal definition of data clustering, the performance

[*] Corresponding author. Tel:. +2712 680 6360; Fax:. +2712 680 7388

Email addresses: agraaff@cs.up.ac.za (A. J. Graaff), engel@cs.up.ac.za (A. P. Engelbrecht)

[1] http://cirg.cs.up.ac.za

measures used to measure the quality of the clusters and how these performance measures can be used to determine the optimal number of clusters in a data set.

Many of the existing network based artificial immune systems (AIS) for data clustering do not require any user specified parameter of the number of required clusters to cluster the data [5, 6, 7, 8, 9]. These models are inspired by the network theory of immunology which can be defined as the formation of self-organizing lymphocyte network structures [10]. These network structures are a result of the co-operation and co-stimulation between lymphocytes in response to invading antigens. The number of artificial lymphocyte (ALC) networks formed in existing network based AIS models represent the number of potential clusters in the data. Thus, each ALC network structure represents a potential cluster in the data.

There are different techniques used by existing network based AIS models to determine the number of ALC networks. The first is to use a network affinity threshold with a proximity matrix of network affinities between the ALCs in the population [5, 6, 8]. A pair of ALCs with a network affinity below the threshold is linked to form a network. The specified network affinity threshold determines the number of ALC networks. Therefore, specifying the correct network affinity threshold to obtain the correct or required number of clusters can be a formidable task. Another technique to determine the number of ALC networks is to take a hybrid-approach by clustering the ALC population into sub-nets [6, 7, 9]. A drawback to a hybrid-approach is the user specified parameter of the number of required clusters. Another potential drawback to a hybrid-approach is that the formed sub-nets might not always contain ALCs with a good or generic representation of the data. Furthermore, both of these techniques are computationally expensive.

Graaff and Engelbrecht proposed the local network neighbourhood AIS (LNNAIS) with a different ALC network topology [11, 12]. In LNNAIS an ALC's neighbours are not determined by network affinity, but by their individual indices in the population of ALCs. A revised version of the model is discussed in section 4. An ALC in LNNAIS can only link to its immediate neighbours to form an ALC network and there is no need for a proximity matrix of network affinities (with a network affinity threshold) or the need to take a hybrid-approach to determine the number of ALC networks. The number of required clusters, K, is determined by pruning the K lowest calculated network affinities between the ALCs. Even though this technique in LNNAIS is less computationally expensive than the above discussed proximity matrix and hybrid-approaches, it shares a mutual drawback of the user specified parameter of the number of required clusters.

In order to address the user specified parameter of the number of required clusters in LNNAIS, this paper proposes two techniques which can be used with LNNAIS to dynamically determine the number of clusters in a data set and does not investigate the effect of different network neighbourhood topologies in LNNAIS. The first technique utilises cluster validity indices and is similar to the multiple execution approach, though less computationally expensive. The second technique is based on sequential deviation outlier detection and is discussed in section 5. With both techniques, the end result is an enhanced LNNAIS model that can dynamically determine the number of clusters in a data set. Furthermore this paper is more focused on the comparison between the proposed techniques for LNNAIS and the cluster validity indices used in a multiple execution of a clustering algorithm to dynamically determine the number of clusters in a data set. Experimental results of K-means clustering using the multiple execution technique are compared with the results of the proposed LNNAIS techniques and presented in section 7. The experimen-

tal results of LNNAIS and K-means clustering are not compared to determine the better of the two algorithms but to show that the proposed techniques in LNNAIS succeed in finding the optimal number of clusters as good as a multiple execution approach using cluster validity indices. The paper is concluded in section 8 with future work on LNNAIS.

2. Data Clustering and Performance Measures

Clustering of a data set can be defined as the partitioning of the data set in such a way that patterns or feature vectors within the same partition are more *similar* compared to patterns across different partitions. Each partition is referred to as a cluster of patterns and is represented by a *centroid* [13]. The most general measure of *similarity* or *dissimilarity* between feature vectors is based on the distance between these vectors. The Euclidean distance is the most commonly used *similarity* measure, and is defined as

$$\delta_2 \left(\mathbf{p}_i, \mathbf{p}_j \right) \quad = \quad \left\| \mathbf{p}_i - \mathbf{p}_j \right\|^2 \tag{1}$$

where $\mathbf{p}_i$ and $\mathbf{p}_j$ are feature vectors. Partitioning of these feature vectors optimises a specific objective function [14]. The objective function is optimised such that the *inter-cluster* distance is maximised and the *intra-cluster* distance minimised. The *inter-cluster* distance measures the separation between clusters and is calculated as

$$J_{inter} = \frac{2}{K \times (K-1)} \sum_{k=1}^{K-1} \sum_{j=k+1}^{K} \delta \left(\mathbf{c}_k, \mathbf{c}_j \right) \tag{2}$$

where K is the number of clusters and $\mathbf{c}_k$ and $\mathbf{c}_j$ are the centroids of the k-th and j-th clusters, respectively. The *intra-cluster* distance measures the *compactness* of the clusters and is calculated as

$$J_{intra} = \frac{\sum_{k=1}^{K} \sum_{\forall \mathbf{p} \in C_k} \delta \left(\mathbf{p}, \mathbf{c}_k \right)}{|P|} \tag{3}$$

where C_k is the cluster (partition) of patterns grouped with the k-th centroid and P is the data set. The classical K-means clustering algorithm [15] is an example of a clustering method which partitions a data set into a number of clusters by means of optimising a specific objective function. K-means clustering initialises K centroids. A feature vector, $\mathbf{p}$, is assigned to a centroid, $\mathbf{c}$, if $\mathbf{p}$ is most similar to $\mathbf{c}$. Similarity is measured using equation (1). Thus the subset of feature vectors assigned to a centroid forms a cluster. After all the feature vectors in data set P have been assigned to a centroid, the centroid of each cluster is recalculated using

$$\mathbf{c}_k = \frac{1}{|C_k|} \sum_{\forall \mathbf{p} \in C_k} \mathbf{p} \tag{4}$$

Algorithm 1 lists the pseudo-code of a basic K-means clustering algorithm [14]. The *stopping criterion* for K-means in this paper is based on a specified number of iterations, t_{max}.

Since the identified number of groups (clusters) and the partitioning of data patterns between these groups may differ among different clustering algorithms, the quality of the partitioning needs to be evaluated. The quality of the clusters can be validated with a cluster validity index.

Algorithm 1 Basic K-means

1: Randomly initialise K centroids
2: **while** some stopping condition(s) not true **do**
3: **for** each feature vector $\mathbf{p}_i \in P$ **do**
4: Calculate the *similarity* between $\mathbf{p}_i$ and $\mathbf{c}_k, k = 1, \ldots, K$
5: Assign $\mathbf{p}_i$ to centroid $\mathbf{c}_k$ with which $\mathbf{p}_i$ has the highest *similarity*
6: **end for**
7: Recalculate the centroid of each cluster using equation (4)
8: **end while**

Ray and Turi proposed a validity index which is based on the ratio of *intra-clustering* distance to the minimum *inter-clustering* distance [3]. The proposed index is calculated as [3]

$$Q_{ratio} = \frac{intra}{inter_{min}} \tag{5}$$

where $intra$ is defined in equation (3), $inter_{min}$ is calculated as

$$inter_{min} = \min_{\substack{k=1,\ldots,K-1 \\ j=k+1,\ldots,K}} \{\delta(\mathbf{c}_k, \mathbf{c}_j)\} \tag{6}$$

and δ is the Euclidean distance as defined in equation (1). In the above definition of *intra*, the average *compactness* of the clusters is calculated by averaging over all the distances between each cluster's centroid and the feature vectors within that cluster. The definition of *inter*$_{min}$ simply calculates the smallest distance between the centroids of the clusters to determine the smallest separation between clusters. The *intra* function needs to be minimised for more compact clusters and the *inter*$_{min}$ needs to be maximised for more separated clusters. Thus, the defined ratio validity index, Q_{ratio}, needs to be minimised to have optimal clustering. Therefore the optimal number of clusters, K, minimises the value of Q_{ratio}.

Davies and Bouldin (DB) proposed a cluster validity index that measures the average similarity between each cluster and the cluster most similar to it [16]. The DB-index is calculated as [17]

$$Q_{DB} = \frac{1}{K} \sum_{k=1}^{K} \max_{\substack{j=1,\ldots,K \\ j \neq k}} \left\{ \frac{\frac{1}{2}\varsigma(C_k) + \frac{1}{2}\varsigma(C_j)}{\sigma(\mathbf{c}_k, \mathbf{c}_j)} \right\} \tag{7}$$

where $Q_{DB} \in [0, \infty)$, K is the number of clusters, σ is the Euclidean distance as defined in equation (1) and ς is the cluster centroid diameter, defined as [18]

$$\varsigma(C_k) = 2 \left[\frac{\sum_{\forall \mathbf{p} \in C_k} \sigma(\mathbf{p}, \mathbf{c}_k)}{|C_k|} \right] \tag{8}$$

where $|C_k|$ is the number of feature vectors in cluster C_k and $\mathbf{c}_k$ is the centroid of cluster C_k.

In the above definition, Q_{DB} has a small value when the distance between centroids $\mathbf{c}_k$ and $\mathbf{c}_j$ is large and the corresponding clusters C_k and C_j of these centroids are compact. Thus, an

optimal number of K clusters minimises the value of Q_{DB}.

Another approach to determine the optimal number of clusters is to execute the clustering algorithm multiple times, each time with a different number of clusters and validating the clustered data set with a cluster validity index like Q_{ratio} or Q_{DB}. The cluster validity index is then plotted as a function of the number of clusters obtained for each execution of the algorithm. The number of clusters generated from the input parameters with the lowest cluster validity index (in the case of Q_{ratio} and Q_{DB}) is then selected as the optimal number of clusters [3, 4]. Since the Q_{ratio} index tends to minimize at small values of K, Turi proposed a modification to the above ratio of *intra-clustering* distance to *inter-clustering* distance by multiplying the ratio with a Gaussian function of the number of clusters [19]. The modified index is calculated as [19]

$$Q_{RT} = Q_{ratio} \times [c \times g(\mu, \sigma) + 1] \tag{9}$$

where g is a Gaussian function with mean, μ, standard deviation, σ, and c is some constant. Function g is defined as

$$g(\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left[-\frac{(K-\mu)^2}{2\sigma^2}\right]} \tag{10}$$

where K is the number of clusters. The Gaussian function penalizes the ratio for small values of K in favour of larger values of K.

The next section discusses some of the existing data clustering methods to dynamically determine the number of clusters in a data set.

3. Dynamic Data Clustering Methods

Dynamically determining the optimal number of clusters in a data set is a challenging task, since *a priori* knowledge of the data is required and not always available. As discussed in the previous section, cluster validity indices can be used with a multiple execution of the clustering algorithm to dynamically determine the number of clusters. A disadvantage of the multiple execution approach is that the technique is computationally expensive and time consuming. Other techniques and clustering models have also been proposed in the literature and are discussed next.

Ball and Hall [20] proposed the Iterative Self-Organizing Data Analysis Technique (ISODATA) to dynamically determine the number of clusters in a data set. As with K-means clustering, ISODATA iteratively assigns patterns to the closest centroids. Different to K-means clustering, ISODATA utilises two user-specified thresholds to respectively merge two clusters (if the distance between their centroids is below the first threshold) and also split a cluster into two clusters (based on the second threshold). Even though ISODATA has an advantage above K-means clustering to dynamically determine the number of clusters in the data set, ISODATA has two additional user parameters (merging and splitting thresholds) which have an effect on the number of clusters determined. A similar model to ISODATA is the Dynamic Optimal Cluster-seek (DYNOC) which was proposed by Tou [21]. DYNOC also follows an iterative approach with splitting and merging of clusters but at the same time maximises the ratio of the minimum inter-clustering to the maximum intra-clustering distance. DYNOC also requires a user specified parameter which determines the splitting of a cluster. SYNERACT was proposed by Huang [22]

as an alternative to ISODATA. SYNERACT uses a hyperplane to split a cluster into smaller clusters for which the centroids need to be calculated. Similar to ISODATA and DYNOC, an iterative approach is followed to assign patterns to available clusters. Even though SYNERACT is faster than ISODATA and does not require the initial location of centroids or the number of clusters to be specified, SYNERACT does require values for two parameters which have an effect on the splitting of a cluster.

Veenman proposed a partitional clustering model which minimises a cluster validity index in order to dynamically determine the number of clusters in a data set [23]. The initial number of clusters is equal to the number of patterns in the data set. An iterative approach is followed to determine the splitting and merging of clusters. In each iteration, tests which are based on the minimisation of the cluster validity index determine the splitting or merging of clusters. The proposed algorithm has similar drawbacks as the multiple execution approaches, namely that the model is computationally expensive and has user parameters for the cluster validity index which influences the clustering results.

Another K-means based model was proposed by Pelleg and Moore [24] and uses model selection. The model is called X-means and initially start with a single cluster, $K = 1$ (which is the minimum number of clusters in any data set). The first step is then to apply K-means clustering on the K clusters which are then split in a second step according to a Bayesian Information Criterion (BIC) [25]. If the BIC is improved with the splitting of the clusters, the newly formed clusters are accepted, otherwise it is rejected. These steps are repeated until a user specified upper bound on K is reached. X-means clustering dynamically determines the number of clusters in the data set as the value of K which has the best BIC value. X-means also has a drawback of a user specified parameter for the upper bound on K. Hamerly and Elkan proposed a similar model as X-means clustering, called G-means clustering [26]. G-means also starts with a small value of K but only splits clusters which data do not have a Gaussian distribution. This is also a drawback of G-means clustering, since it is assumed that the data has spherical and/or elliptical clusters [26].

There are also other models proposed in the literature which is either based on K-means clustering or utilises K-means with similar approaches of splitting and merging clusters. These models are *Snob* [27] and Modified Linde-Buzo-Gray (MLBG) [28]. All of the discussed models suffer from either user parameters which influence the clustering results or can only cluster data sets with specific characteristics.

The following sections discuss the local network neighbourhood artificial immune system (LNNAIS) and propose two techniques which can be used with LNNAIS to dynamically determine the number of clusters in a data set.

4. The Local Network Neighbourhood Artificial Immune System

The main difference between LNNAIS and existing network based AIS models is the network topology of the ALCs and an index-based neighbourhood technique. Neighbours of an ALC in LNNAIS are determined by the indices of the ALCs in the population. An ALC is only allowed to link to its immediate neighbours to form an ALC network. The remainder of this section gives an overview of the LNNAIS algorithm. Since the purpose of this paper is to propose

a technique to dynamically determine the number of ALC networks in LNNAIS, more emphasis will be placed on the index-based neighbourhood technique in LNNAIS.

The LNNAIS algorithm is given in pseudo code in Algorithm 2. The *stopping criterion* for LNNAIS in this paper is based on a specified number of iterations, t_{max}. Figure 1 shows a flow chart for the steps in the LNNAIS algorithm. The following sections discuss each of these steps in more detail.

Algorithm 2 Local Network Neighbourhood AIS Algorithm

Set the maximum size of the ALC population as $\mathcal{B}_{max}$
Initialise an empty set of ALCs as population $\mathcal{B}$
while some stopping condition(s) not true **do**
  for each antigen, $\mathbf{a}_j \in \mathcal{A}$, at index position j in $\mathcal{A}$ **do**
    if $|\mathcal{B}| \leq 0$ **then**
      Initialise a new ALC, $\mathbf{b}$, with the same structure as pattern $\mathbf{a}_j$
      $\mathcal{B} = \mathcal{B} \cup \mathbf{b}$
    end if
    Calculate the antigen affinity between $\mathbf{a}_j$ and each $\mathbf{b}_i \in \mathcal{B}$ using equation (1)
    Select $\mathbf{b}_h \in \mathcal{B}$, at index h, as the ALC with highest calculated antigen affinity
    Proliferate $\mathbf{b}_h$ as discussed in section 4.2
    if $\mathbf{b}_h$ is activated ($|\mathcal{C}_h| > \epsilon_{clone}$) **then**
      Generate a mutated clone, $\mathbf{b}_h^{'}$, using equation (11)
      Secrete an antibody, $\mathbf{b}^*$, as discussed in section 4.3
      Determine the local network neighbourhood of $\mathbf{b}_h$ using equation (15)
      Co-stimulate the local network neighbourhood of $\mathbf{b}_h$ with $\mathbf{b}^*$, as discussed in section 4.4
    end if
  end for
  Apply the SDOT or IPT technique on $\mathcal{B}$ to determine the number of ALC networks (clusters)
end while

4.1. Initializing an ALC, an antigen mutated clone and the ALC population

The ALC population, $\mathcal{B}$, in LNNAIS is initialised as an empty set. The ALC population expands to a maximum size, $\mathcal{B}_{max}$, over time. The patterns in data set, $\mathcal{A}$, that needs to be partitioned are seen as antigen patterns and are randomly presented to the ALC population. The ALCs and antigen mutated clones in LNNAIS are encoded with the same structure as the antigen patterns in $\mathcal{A}$. If patterns in the data set are real-valued (or binary) vectors then the ALCs and antigen mutated clones are also real-valued (or binary) vectors. ALCs with antigen mutated clones are used in LNNAIS to adapt to the antigen patterns to form network structures and eventually cluster the data set. The initialisation of antigen mutated clones and the insertion of initialised ALCs into $\mathcal{B}$ are discussed next.

4.2. Proliferating the Clonal Selected ALC

Each ALC, $\mathbf{b}_i$, at index position i in $\mathcal{B}$, contains a set of antigen mutated clones, $\mathcal{C}_i$. An ALC $\mathbf{b}_h$ at index h in population $\mathcal{B}$ is selected as the ALC with the highest binding affinity (lowest

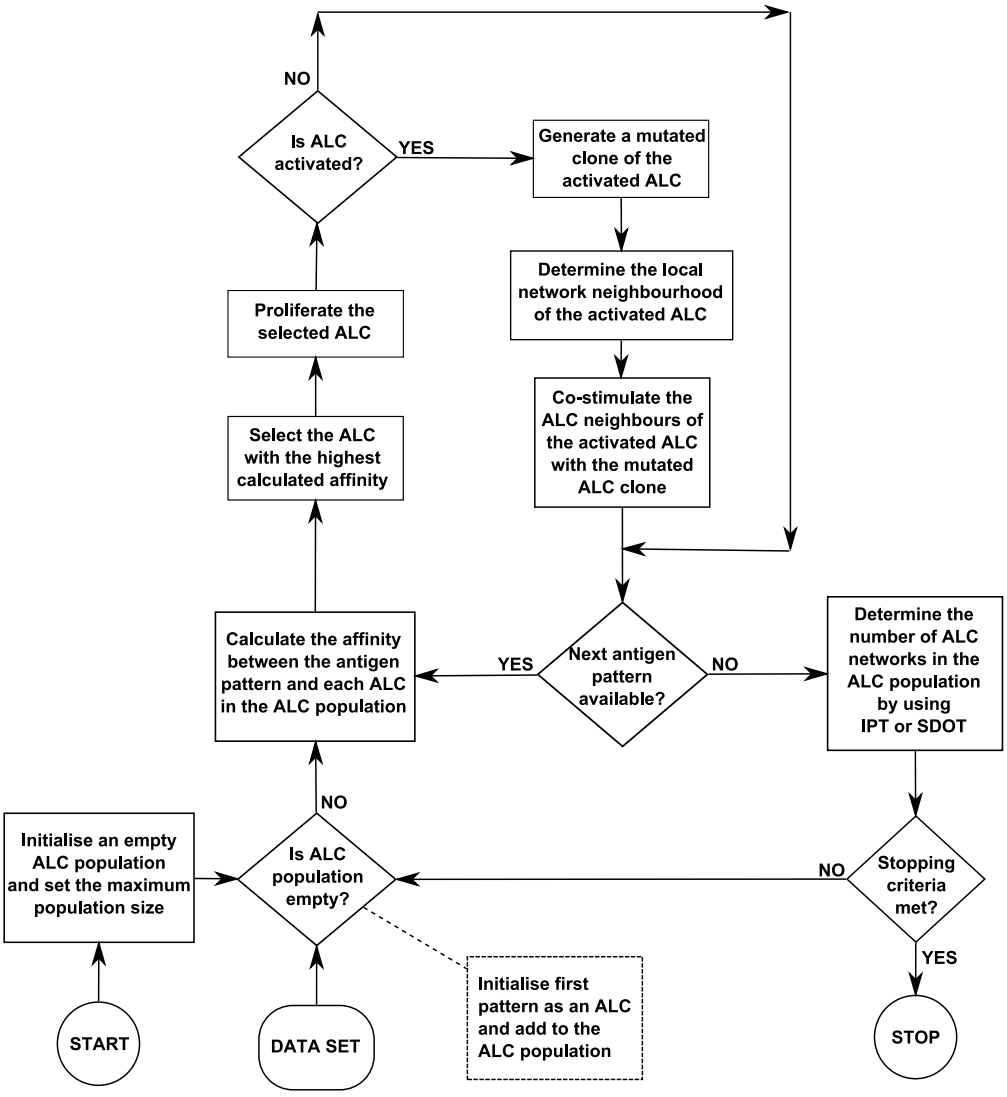

Figure 1: Flow chart of LNNAIS algorithm

Euclidean distance) with an antigen $\mathbf{a}$. The antigen pattern $\mathbf{a}$ is then initialised as an antigen mutated clone, $\mathbf{a}^{'}$, by inserting the antigen pattern at the first index position of $\mathcal{C}_h$, which is the set of antigen mutated clones for $\mathbf{b}_h$. This increases the clonal level of $\mathbf{b}_h$. The ALC activates when the clonal level, $|\mathcal{C}|$, exceeds the clonal level threshold, ϵ_{clone}, and generates a mutated ALC clone. The next section discusses the generation of a mutated ALC clone.

4.3. Generating a Mutated Clone of an Activated ALC

An activated ALC, $\mathbf{b}_h$, generates a mutated clone, $\mathbf{b}_h^{'}$, by using

$$\mathbf{b}_h^{'} = \mathbf{b}_h + \frac{\sum_{c=1}^{|\mathcal{C}_h|} \delta^* \left(\mathbf{b}_h, \mathbf{a}_c^{'}, \mathcal{C}_h\right) \left(\mathbf{a}_c^{'} - \mathbf{b}_h\right)}{\sum_{c=1}^{|\mathcal{C}_h|} \delta^* \left(\mathbf{b}_h, \mathbf{a}_c^{'}, \mathcal{C}_h\right)} \tag{11}$$

where

$$\delta^* \left(\mathbf{b}_h, \mathbf{a}^{'}, \mathcal{C}_h\right) = 1.0 - \frac{\delta \left(\mathbf{b}_h, \mathbf{a}^{'}\right)}{\delta_{max} + 1.0} \tag{12}$$

$$\delta_{max} = max_{c=1,\ldots|\mathcal{C}_h|} \left\{\delta \left(\mathbf{b}_h, \mathbf{a}_c^{'}\right)\right\} \tag{13}$$

$$\mathbf{a}_c^{'} \in \mathcal{C}_h \tag{14}$$

In the above definition, δ^* calculates the normalised affinity between an antigen mutated clone, $\mathbf{a}_c^{'} \in \mathcal{C}_h$, and an ALC, $\mathbf{b}_h$, with respect to the lowest affinity (highest Euclidean distance) in the set of antigen mutated clones, $\mathcal{C}_h$. The set of antigen mutated clones, $\mathcal{C}_h$, which is contained by an ALC $\mathbf{b}_h$, determines the mutated clone which will be generated when an ALC is activated. Antigen mutated clones in $\mathcal{C}_h$ with a higher binding affinity with ALC $\mathbf{b}_h$, have a higher influence on the mutation of the clone, which results in an ALC clone that is mutated more towards higher affinity antigen mutated clones in $\mathcal{C}_h$.

The antigen mutated clones in $\mathcal{C}_h$ with which $\mathbf{b}_h^{'}$ has a higher affinity than the parent ALC $\mathbf{b}_h$, is added to the clonal set of $\mathbf{b}_h^{'}$ (bind to $\mathbf{b}_h^{'}$). If more than half of the number of antigen mutated clones in $\mathcal{C}_h$ bind to $\mathbf{b}_h^{'}$, the parent ALC $\mathbf{b}_h$ is added as an antigen mutated clone to the clonal set of $\mathbf{b}_h^{'}$. The parent ALC is then replaced by $\mathbf{b}_h^{'}$ in $\mathcal{B}$ and secreted as a co-stimulating antibody, $\mathbf{b}^*$, to neighbouring ALCs. If less than half of the number of antigen mutated clones in $\mathcal{C}_h$ bind to $\mathbf{b}_h^{'}$, the parent ALC $\mathbf{b}_h$ is suppressed by removing all of the antigen mutated clones in $\mathcal{C}_h$. This prevents frequently activated ALCs from dominating the population. The mutated ALC clone, $\mathbf{b}_h^{'}$, is then inserted into $\mathcal{C}_h$; not only to co-stimulate the parent ALC, but also to preserve the memory of the antigen structure. The mutated ALC clone is secreted as a co-stimulating antibody, $\mathbf{b}^*$, to neighbouring ALCs. The following section discusses the co-stimulation of neighbouring ALCs within a local network neighbourhood.

4.4. Determining and Co-stimulating the Local Network Neighbourhood of an Activated ALC

The neighbourhood, $\mathcal{N}_{h,\rho}$, of an ALC, $\mathbf{b}_h \in \mathcal{B}$, is defined as

$$\mathcal{N}_{h,\rho} = \left\{\forall \mathbf{b}_j \in \mathcal{B} : \min_{j=h-(\rho-1),\ldots,h} \left\{\nu \left(h, \rho - 1\right)\right\}\right\} \tag{15}$$

where

$$\rho \leq |\mathcal{B}| \tag{16}$$

$$\mathcal{N}_{h,\rho} \subseteq \mathcal{B} \tag{17}$$

$$\mathbf{b}_h \in \mathcal{N}_{h,\rho} \tag{18}$$

and ν calculates the average network affinity between ALCs in the population from index position h to $h + (\rho - 1)$ and is defined as

$$\nu(x, y) = \frac{\sum_{i=x}^{y-1} \delta(\mathbf{b}_i, \mathbf{b}_{i+1})}{y - x} \tag{19}$$

The neighbourhood of an ALC, $\mathbf{b}_h$, is therefore determined by a network window of size ρ which starts at position $h - (\rho - 1)$, sliding over the ALC population in search of the highest average network affinity (minimum average Euclidean distance).

An activated ALC, $\mathbf{b}_h$, secretes an antibody, $\mathbf{b}^*$ (as discussed in section 4.3). The secreted antibody, $\mathbf{b}^*$, then co-stimulates the neighbouring ALCs in $\mathcal{N}_{h,\rho}$. The immediate neighbours of $\mathbf{b}_h$ at indices $h - 1$ and $h + 1$, react to the secreted antibody by adding the clonal set of the antibody to $\mathcal{C}_{h-1}$ and $\mathcal{C}_{h+1}$, respectively. The neighbouring ALCs at indices $h - 1$ and $h + 1$ can then also be activated and secrete antibodies (as explained in section 4.3). The secreted antibodies of the activated neighbouring ALCs at indices $h - 1$ and $h + 1$ will co-stimulate their immediate ALC neighbours at indices $h-2$ and $h+2$, respectively. If a neighbouring ALC is not activated by the co-stimulation of a predecessor's antibody, the antibody is inserted into the local network at the index of the neighbouring ALC, increasing the population size. The neighbouring ALCs with the highest network affinity in the population, which are not within the local network neighbourhood, are then merged to stabilise the population size. The process of co-stimulation continues until the ALCs on the boundary of the local network neighbourhood are co-stimulated or until a neighbouring ALC is not activated by the co-stimulation of a predecessor's antibody.

4.5. Determining the Number of ALC Networks in LNNAIS

An advantage of an index-based neighbourhood is that there is no need of a network affinity threshold with a proximity matrix of network affinities to determine the number of ALC networks in LNNAIS. It is also not necessary to follow a hybrid-approach of clustering the ALC population. An index-based neighbourhood results in the formation of a ring-like network topology as illustrated in figure 2.

The required number of ALC networks (or clusters in the data), K, can be determined by sorting the network affinities in descending order and selecting the first K network affinities in the sorted set. The K selected network affinities determine the boundaries of the ALC networks. Figure 2 illustrates this technique where $K = 3$. The edges which are selected as boundaries are pruned to form separate ALC networks (illustrated as dotted lines in figure 2). The centroid of each of the formed ALC networks (illustrated as clouds) is calculated using equation (4). A drawback to this approach in LNNAIS is the user specified parameter K.

Instead of specifying K, the above pruning technique is done with an iterative value of K. First K is set to 2 where only the top two boundaries are selected for pruning (top two network affinities in the sorted set of network affinities). The quality of the clusters is then measured with a

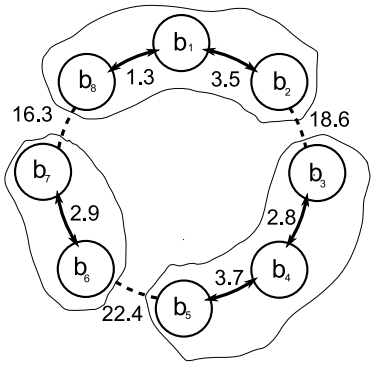

Figure 2: Determining the Number of Clusters in LNNAIS

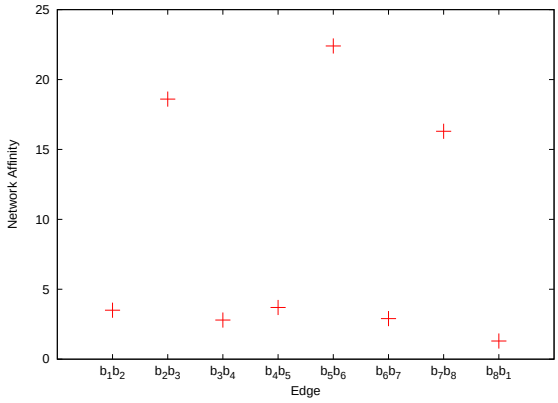

Figure 3: Plotting the Network Affinities

cluster validity index of choice. The same procedure is followed for $K = \{3, 4, 5, \ldots, \mathcal{B}_{max}\}$, measuring the quality with a cluster validity index for each value of K. The value of K with the highest (or lowest) cluster validity index is then selected as the optimal number of clusters. It is also possible to set a minimum and maximum for K, but this can also be seen as a drawback since two parameters need to be specified. If no minimum/maximum is specified it could also be a time consuming task (to a lesser extent when compared to the multiple execution technique) to iterate through all values of K, especially with large values of $\mathcal{B}_{max}$. Whether K is bounded by a minimum/maximum or not, an advantage of the Iterative Pruning Technique (IPT) to dynamically determine the number of clusters is that the LNNAIS model needs not to be executed for each value of K as in the case of the multiple execution technique. Therefore IPT is computationally less expensive.

Instead of sorting the network affinities in descending order, the network affinities can be plotted against the numbered edges (links) between the ALCs on a graph (as illustrated in figure 3). The K edges in the graph with the lowest plotted network affinity (highest Euclidean distance) are then selected as the boundaries of the ALC networks. Note that the network affinities of the selected edges in figure 3 are *outliers* compared to the network affinities of the remaining edges.

Therefore, to dynamically determine the number of boundaries (clusters) in an ALC population, the *outlier* network affinities needs to be identified. The next section discusses and explains a technique to identify *outliers* in a set and the application thereof in LNNAIS to dynamically determine the number of ALC networks.

5. Sequential Deviation Based Outlier Detection

Referring to the definition of data clustering in section 2, each cluster (or centroid) represents a *concept* or *trend* in the data set. Based on a *similarity* measure, an *outlier* feature vector is either not grouped with any cluster or has a major deviation from the centroid of a cluster with which the *outlier* is associated. Therefore an *outlier* is also known as an *exception* and is defined as a vector which is not *similar* to any of the centroids. *Outliers* are grossly different from and/or inconsistent with feature vectors of the same data set [29], which can be a result of inherent data variability [29].

Outlier detection and analysis is referred to as *outlier mining* and is described as follows [30]: In a data set of I feature vectors, the expected number of outlier vectors, o, are those feature vectors which are the most *dissimilar*, *exceptional* and/or *inconsistent* compared to the remainder of the data set. Outlier detection can be categorized into three approaches, namely the statistical approach, distance-based approach and deviation-based approach [29, 30]. Focusing on the deviation-based approach, there are two techniques in deviation based outlier detection [29], namely sequential exception and the on-line analytical processing (OLAP) data cube technique. The first of these two techniques is discussed next and the interested reader is referred to [29] for more information on the OLAP technique.

The sequential exception technique is based on a process followed by humans to detect an outlier after being represented with a series of *similar* feature vectors [31]. An outlier is defined as a feature vector that deviates from the series.

A sequence of subsets, $\{S_1, S_2, \ldots, S_o\}$ is built from a data set, P, consisting of I feature vectors, i.e. $2 \leq o \leq I$. Thus, $S_{o-1} \subset S_o : S_o \subseteq S$. A function of *dissimilarity* (not necessarily distance based) is calculated between each subset. The *dissimilarity* function is defined as any function that returns a low value to indicate more *similar* feature vectors and a high value to indicate less *similar* feature vectors [29, 30].

A *smoothing factor* function is calculated for each subset, S_o in the sequence. The subset, S_o, with the highest *smoothing factor* becomes the set of outliers, S_e [31, 29]. The *cardinality* of each subset is used to scale the *smoothing factor*. The *cardinality* of a set is defined as the number of feature vectors in the set [31, 29]. The *smoothing factor* is calculated as [31]

$$sf\left(S_o\right) = \left|S_o - S_{o-1}\right| \times \left(D\left(S_o\right) - D\left(S_{o-1}\right)\right) \tag{20}$$

where $|\bullet|$ is the cardinality of a set and D is the function of dissimilarity. Thus, the *smoothing factor* (sf), calculates the reduction in *dissimilarity* when removing a subset S_o of feature vectors from set S. The exception set S_e has the highest sf value and is defined as [31]

$$sf\left(S_e\right) \geq sf\left(S_o\right) \quad \forall S_o \subset S \tag{21}$$

If all feature vectors in S are *similar*, the *smoothing factor* equals zero [31].

In the context of dynamically determining the boundaries between the ALCs in LNNAIS, the sequential deviation technique can be applied to a sorted set (descending) of network affinities between the ALCs in LNNAIS. The set of network affinities is sorted to guarantee that the lowest network affinities (potential outliers with the highest Euclidean distance) forms part of the first sequential subsets. The first subset, S_1, will then contain the lowest network affinity, followed by S_2 which consists of S_1 and the second lowest network affinity and so forth. The function of dissimilarity $D(S_o)$ in equation (20) is calculated as the variance between the network affinities in subset S_o. Therefore the exception set S_e contains the lowest network affinities between the ALCs in LNNAIS and eventually determines the boundaries between the ALCs.

An added advantage of the Sequential Deviation Outlier Technique (SDOT) is that not only is the technique less computationally expensive but it also has no need for any boundary constraints on K. K is solely determined by the size of S_e. Furthermore, SDOT is a non-parametric technique. The following section discusses the time complexity of SDOT and IPT.

6. Time Complexity of SDOT and IPT

The time complexity of both SDOT and IPT are based on the complexity of sorting the network affinities between the ALCs in the ALC population and determining the number of boundaries between the ALCs in the ALC population of size $\mathcal{B}_{max}$. The maximum number of boundaries in an ALC population of size $\mathcal{B}_{max}$ is $\mathcal{B}_{max}$. The time complexity of sorting the $\mathcal{B}_{max}$ network affinities depends on the sorting algorithm used. Assume the time complexity of the sorting algorithm is some constant, χ_1, and that the time complexity of the selected validity index is χ_2. The worst case of time complexity for IPT is when the clustering quality of all possible boundaries needs to be calculated, giving a time complexity of $O(\chi_2 \mathcal{B}_{max} |\mathcal{A}| n)$ where $|\mathcal{A}|$ is the size of the data set that needs to be partitioned and n is the dimension of data set $\mathcal{A}$. The $\mathcal{B}_{max}$ and χ_2 parameters are fixed in advance and usually $\mathcal{B}_{max} << |\mathcal{A}|$. If $\mathcal{B}_{max} << |\mathcal{A}|$ then the time complexity of IPT is $O(|\mathcal{A}|)$ and if $\mathcal{B}_{max} \approx |\mathcal{A}|$ then the time complexity of IPT is $O\left(|\mathcal{A}|^2\right)$. Focusing on SDOT, the maximum number of smoothing factor function evaluations is equal to the size of the ALC population, which is $\mathcal{B}_{max}$. Assume the time complexity of the smoothing function is χ_3. The worst case of time complexity for SDOT is when the smoothing factor of $\mathcal{B}_{max}$ subsets need to be calculated to determine the exception set S_e (as discussed in section 5). This gives a time complexity of $O(\chi_3 \mathcal{B}_{max})$ for SDOT. Compared to the time complexity of IPT, the time complexity of SDOT is not influenced by the size of data set $\mathcal{A}$ and also not by the number of dimensions, n.

The following section discusses and compares the results obtained from K-means clustering using the multiple execution technique to determine the number of clusters in a data set and the results obtained from LNNAIS using SDOT and IPT to determine the number of clusters in a data set.

7. Experimental Results

This section compares and discusses the clustering results obtained by K-means clustering, LNNAIS using IPT, and LNNAIS using SDOT to dynamically determine the number of clusters

in a data set. K-means utilises the multiple execution technique with the Q_{DB} (as defined in equation (7)) and Q_{RT} (as defined in equation (9)) validity indices, referred to as KM$_{DB}$ and KM$_{RT}$, respectively. Two of the LNNAIS models utilises the iterative pruning technique with the same Q_{DB} and Q_{RT} validity indices as K-means, referred to as LNN$_{DB}$ and LNN$_{RT}$, respectively. For the Q_{RT} validity index, parameter c was set to 10 in all the experiments. The value of c was found empirically and values of $c > 10$ have no effect on Q_{RT} for all the data sets. LNN$_{SDOT}$ utilises the sequential deviation outlier technique and thus need no validity index.

All experimental results reported in this section are averages taken over 50 runs, where each run consisted of 1000 iterations of a data set. The parameter values for each data set were found empirically to deliver the best performance for each of the algorithms. The value of K was iterated from $K = 2$ to $K = 12$ for all data sets. Table 1 summarises the parameter values used by the respective algorithms for each data set. The selection of data sets used to benchmark the clustering performance and quality of the models represents a good distribution of data clustering problems with the number of patterns in the range $[150, 4601]$ and the number of features in the range $[2, 57]$. All the data sets have overlapping patterns except the hepta data set. The target data set also contains outlier patterns. The twospiral, hepta, engytime, chainlink and target data sets are part of a fundamental clustering problems suite [32]. The other data sets were collected from the UCI Machine Learning repository [33]. The clustering quality of the algorithms (based on the number of clusters determined by each of the algorithms), is determined by the Q_{ratio} index, J_{intra} and J_{inter} performance measures (as defined in equations (5),(3) and (2), respectively). The following hypothesis is defined to determine whether there is a difference between the clustering quality of two algorithms for a specific data set or not:

- *Null* hypothesis, H_0: There is no difference in the clustering quality, Q_{ratio}.
- *Alternative* hypothesis, H_1: There is a difference in the clustering quality, Q_{ratio}.

A non-parametric Mann-Whitney U test with a 0.95 confidence interval ($\alpha = 0.05$) was used to test the above hypothesis. The result is statistically significant if the calculated probability (p-value is the probability of H_0 being true) is less than α. In cases where there is a statistical significant difference between the clustering quality of two algorithms, the algorithm with the lowest critical value, z, tends to find clusters in the data set with a higher quality. The results for each of the data sets used are discussed next.

7.1. Iris data set

Figure 4 illustrates the Q_{RT} values where $c = 10$ for KM$_{RT}$ and LNN$_{RT}$ on the $y1$-axis at different values of K. The Q_{DB} values for KM$_{DB}$ and LNN$_{DB}$ is illustrated on the $y2$-axis of figure 4. Figure 4 highlights that the optimal number of clusters in the iris data set is obtained by KM$_{RT}$ and LNN$_{RT}$ at $K = 4$ and by KM$_{DB}$ and LNN$_{DB}$ at $K = 2$. Therefore, the optimal range of K is $K = 2$ to $K = 4$ for the iris data set. The average number of clusters determined by LNN$_{SDOT}$ is $K = 2.64$ which falls within the optimal range of K as determined above. Figure 6 illustrates for the iris data set the number of clusters respectively determined by the SDOT and IPT techniques over time. The value of K for IPT rapidly increases to 4 in the first few iterations and remains at 4 for the most of the remaining iterations. The value of K for SDOT increases to 2.7 and oscillates between 2.4 and 3.3 around an average K of 2.64 for the remaining iterations. Since LNNAIS is a stochastic algorithm which utilises a dynamic population of ALCs, the affinities between neighbouring ALCs change over time. Thus, it is expected that the network

Table 1: LNNAIS Parameter Values

Data set	$\mathcal{B}_{max}$	ρ	ϵ_{clone}
iris	25	3	5
twospiral	20	3	5
hepta	40	3	5
engytime	20	3	10
chainlink	40	3	5
target	30	3	5
ionosphere	20	3	20
glass	20	3	5
image segmentation	30	3	20
spambase	10	5	20

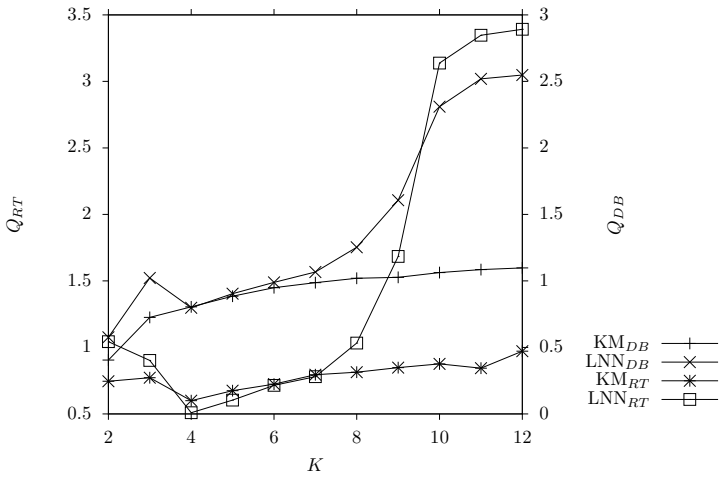

Figure 4: Optimal number of clusters obtained by K-means and LNNAIS for the iris data set

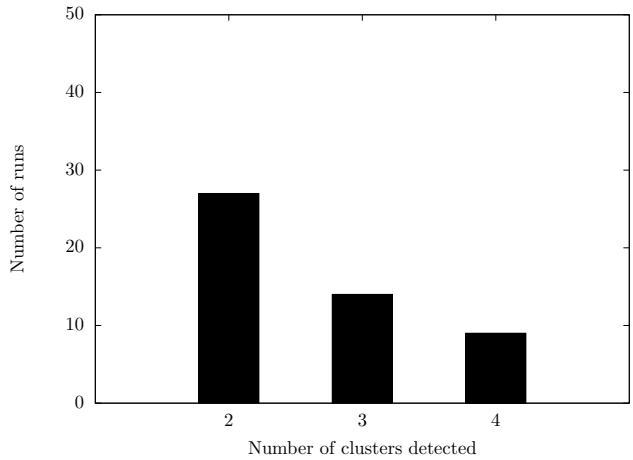

Figure 5: Histogram of the number of clusters detected in the iris data set by LNN$_{SDOT}$

boundaries detected by SDOT to determine the value of K will also differ over time and oscillate around an average K. Figure 5 illustrates a histogram of the frequency distribution of the number of clusters determined by LNN$_{SDOT}$ for the iris data set. The figure illustrates that LNN$_{SDOT}$ has high frequencies at $K = 2$ and $K = 3$. The figure also illustrates that for some of the runs LNN$_{SDOT}$ obtained $K = 4$ which is still within the optimal range of K for the iris data set.

Table 2 shows the results obtained by the different models to determine the optimal number of clusters in the iris data set. Referring to table 12, the Mann-Whitney U statistical hypothesis test rejects H_0 that the Q_{ratio} means are the same at a 0.05 level of significance between KM$_{RT}$ and LNN$_{SDOT}$ ($z = 7.58, p < 0.001$) and between LNN$_{RT}$ and LNN$_{SDOT}$ ($z = 6.69, p < 0.001$). Thus, there is a statistical significant difference in the clustering quality, Q_{ratio}, of the iris data set between KM$_{RT}$ and LNN$_{SDOT}$ and between LNN$_{RT}$ and LNN$_{SDOT}$. LNN$_{SDOT}$ tends to find clusters in the iris data set with a higher quality.

7.2. Twospiral data set

The optimal range of K as determined by the different models for the twospiral data set is $[3, 12]$ (as illustrated in figure 7). Furthermore, figure 7 shows that although the optimal number of clusters in the twospiral data set is obtained by KM$_{DB}$ at $K = 12$, the majority of the models obtain the optimal number of clusters in the twospiral data set at $K = 4$. The average number of clusters determined by LNN$_{SDOT}$ is $K = 4.06$ which is similar to the optimal number of clusters obtained by the majority of the models. Figure 8 illustrates a histogram of the frequency distribution of the number of clusters determined by LNN$_{SDOT}$ for the twospiral data set. The figure illustrates that LNN$_{SDOT}$ has high frequencies for $2 \leq K \leq 5$. Figure 9 illustrates that for the two spiral data set the IPT technique converges to $K = 4$ and SDOT oscillates between $K = 3.5$ and $K = 5$ around an average $K = 4.2$ which is near the value of K as determined by IPT. The statistical hypothesis test rejects H_0 that the Q_{ratio} means are the same between KM$_{RT}$ and LNN$_{SDOT}$ ($z = 8.328$, $p < 0.001$). There is thus a statistical significant difference between the clustering quality of KM$_{RT}$ and LNN$_{SDOT}$. KM$_{RT}$ tends to

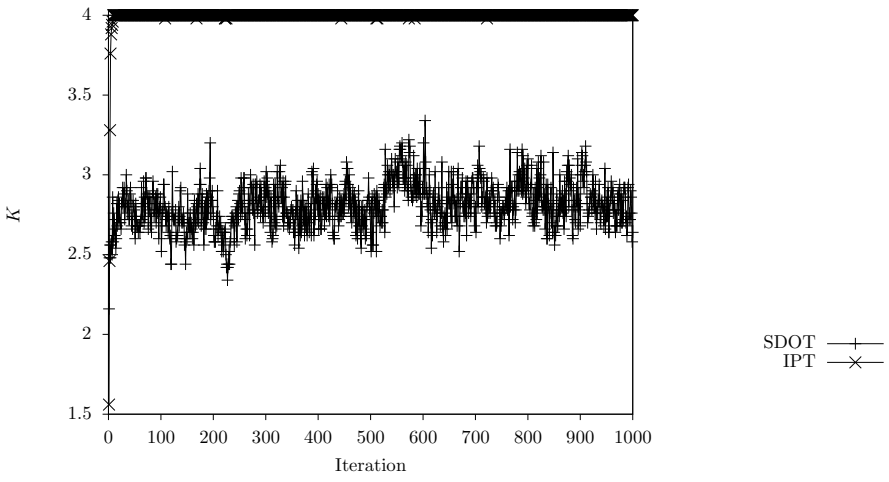

Figure 6: Convergence of LNNAIS using SDOT and IPT to optimal K for iris data set

Table 2: Descriptive Statistics: Iris

Algorithm	K	J_{intra}	J_{inter}	Q_{ratio}	Q_{DB}
KM_{DB}	2.00	0.856	3.927	0.218	0.405
	($\pm$ 0.00)	($\pm$ 0.000)	($\pm$ 0.000)	($\pm$ 0.000)	($\pm$ 0.000)
KM_{RT}	4.00	0.581	3.048	0.575	0.805
	($\pm$ 0.00)	($\pm$ 0.021)	($\pm$ 0.153)	($\pm$ 0.165)	($\pm$ 0.045)
LNN_{DB}	2.00	0.923	3.994	0.233	0.432
	($\pm$ 0.00)	($\pm$ 0.097)	($\pm$ 0.352)	($\pm$ 0.035)	($\pm$ 0.072)
LNN_{RT}	4.00	0.618	3.126	0.488	0.798
	($\pm$ 0.00)	($\pm$ 0.036)	($\pm$ 0.221)	($\pm$ 0.154)	($\pm$ 0.154)
LNN_{SDOT}	2.64	0.788	3.738	0.364	0.643
	($\pm$ 0.77)	($\pm$ 0.109)	($\pm$ 0.466)	($\pm$ 0.552)	($\pm$ 0.858)

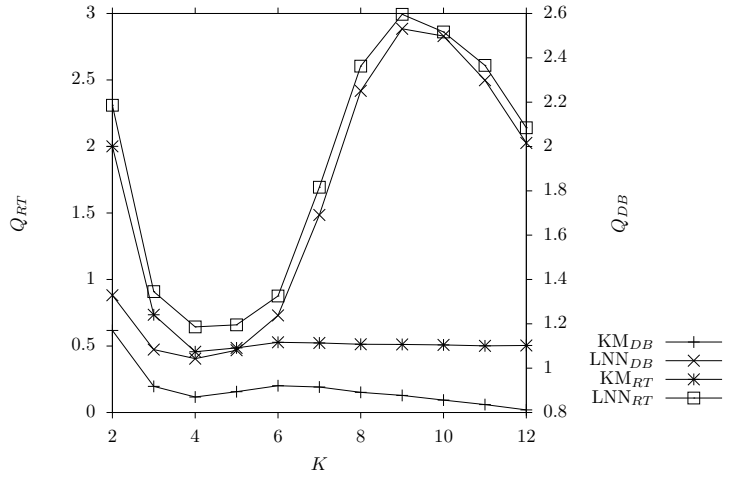

Figure 7: Optimal number of clusters obtained by K-means and LNNAIS for the two spiral data set

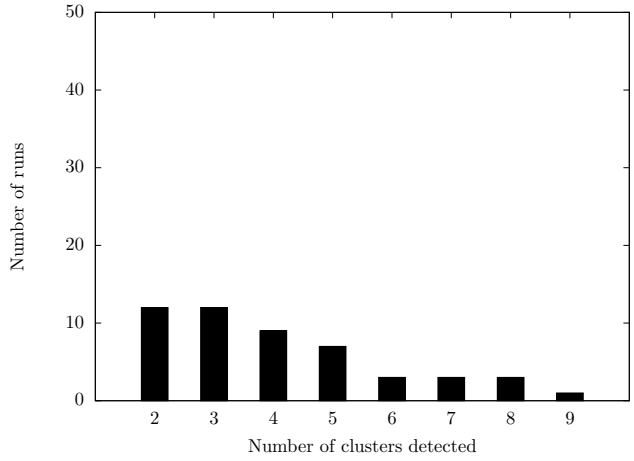

Figure 8: Histogram of the number of clusters detected in the two spiral data set by LNN_{SDOT}

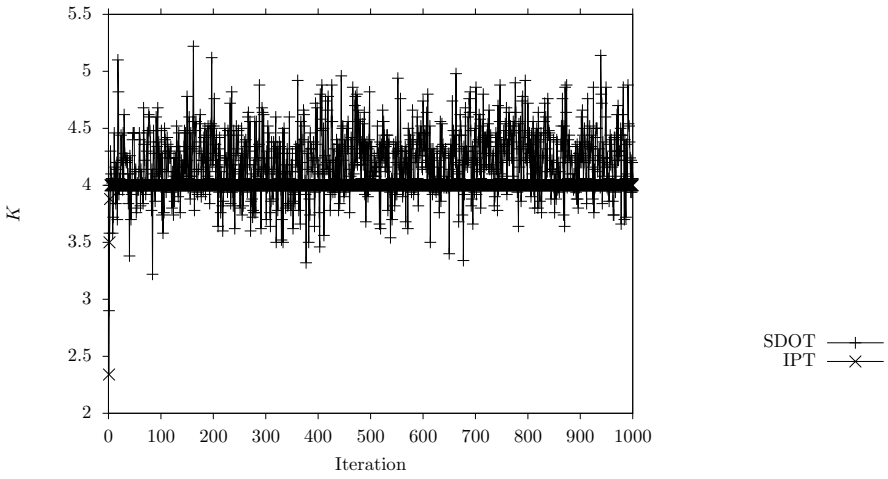

Figure 9: Convergence of LNNAIS using SDOT and IPT to optimal K for two spiral data set

find clusters in the twospiral data set with a higher quality than LNN$_{SDOT}$. There is however no statistical significant difference between the Q_{ratio} means of LNN$_{RT}$ and LNN$_{SDOT}$ (statistical hypothesis test accepts H_0, refer to table 12). Table 3 shows the results obtained by the different models to determine the optimal number of clusters in the twospiral data set.

7.3. Hepta data set

The average number of clusters determined by LNN$_{SDOT}$ for the hepta data set is $K = 6.64$ which is close to the true number of clusters in the hepta data set (hepta consists of seven clusters) and falls within the optimal range of K which is $[4, 7]$ (as illustrated in figure 10). Figure 11 illustrates a histogram of the frequency distribution of the number of clusters determined by LNN$_{SDOT}$ for the hepta data set. Figure 11 highlights that LNN$_{SDOT}$ has the highest frequency at seven clusters, which is the number of clusters in the hepta data set. Figure 12 illustrates for the hepta data set the number of clusters respectively determined by the SDOT and IPT techniques over time. The value of K for IPT converges to 6. The value of K for SDOT oscillates between $K = 6$ and $K = 7$ around an average K of 6.7 for the remaining iterations. Referring to table 12, there is a statistical significant difference between the clustering quality of KM$_{RT}$ and LNN$_{SDOT}$ and between LNN$_{RT}$ and LNN$_{SDOT}$. Although KM$_{RT}$ and LNN$_{RT}$ tend to find clusters in the hepta data set with a higher quality than LNN$_{SDOT}$ (refer to table 4), LNN$_{SDOT}$ was able to determine the number of clusters in the hepta data set more accurately.

7.4. Engytime data set

Table 5 shows the results obtained by the different models to determine the optimal number of clusters in the engytime data set. Figure 13 illustrates that the optimal range of K for the engytime data set is $2 \leq K \leq 7$ (also shown in table 5). LNN$_{SDOT}$ determined the number of clusters in the engytime data set as $K = 3.86$. The histogram of the frequency distribution of the number of clusters determined by LNN$_{SDOT}$ for the engytime data set illustrates that LNN$_{SDOT}$ has high frequencies for $2 \leq K \leq 4$ which is within the optimal range of K (refer to figure 14

Table 3: Descriptive Statistics: Twospiral

Algorithm	K	J_{intra}	J_{inter}	Q_{ratio}	Q_{DB}
KM_{DB}	12.00	0.212	1.018	0.504	0.812
	($\pm$ 0.00)	($\pm$ 0.004)	($\pm$ 0.024)	($\pm$ 0.084)	($\pm$ 0.034)
KM_{RT}	4.00	0.369	0.993	0.437	0.870
	($\pm$ 0.00)	($\pm$ 0.003)	($\pm$ 0.011)	($\pm$ 0.016)	($\pm$ 0.031)
LNN_{DB}	3.00	0.477	1.115	0.544	0.992
	($\pm$ 0.00)	($\pm$ 0.023)	($\pm$ 0.146)	($\pm$ 0.122)	($\pm$ 0.191)
LNN_{RT}	4.00	0.405	1.021	0.616	1.043
	($\pm$ 0.00)	($\pm$ 0.019)	($\pm$ 0.099)	($\pm$ 0.149)	($\pm$ 0.168)
LNN_{SDOT}	4.06	0.427	1.021	0.699	1.116
	($\pm$ 1.89)	($\pm$ 0.087)	($\pm$ 0.088)	($\pm$ 0.736)	($\pm$ 0.537)

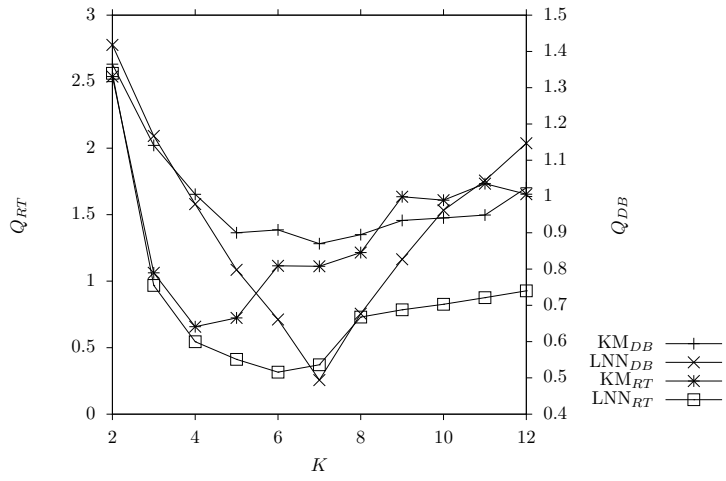

Figure 10: Optimal number of clusters obtained by K-means and LNNAIS for the hepta data set

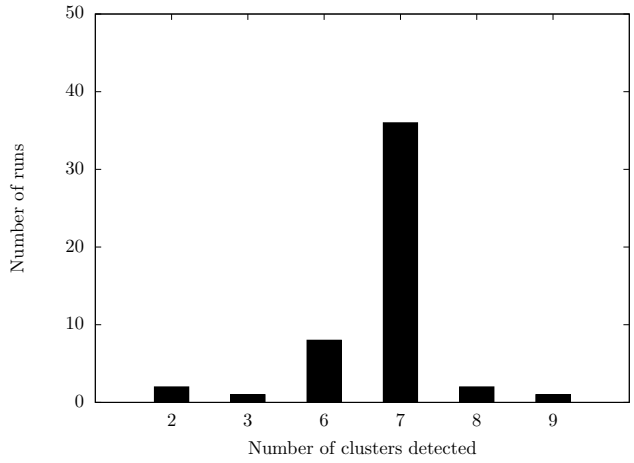

Figure 11: Histogram of the number of clusters detected in the hepta data set by LNN$_{SDOT}$

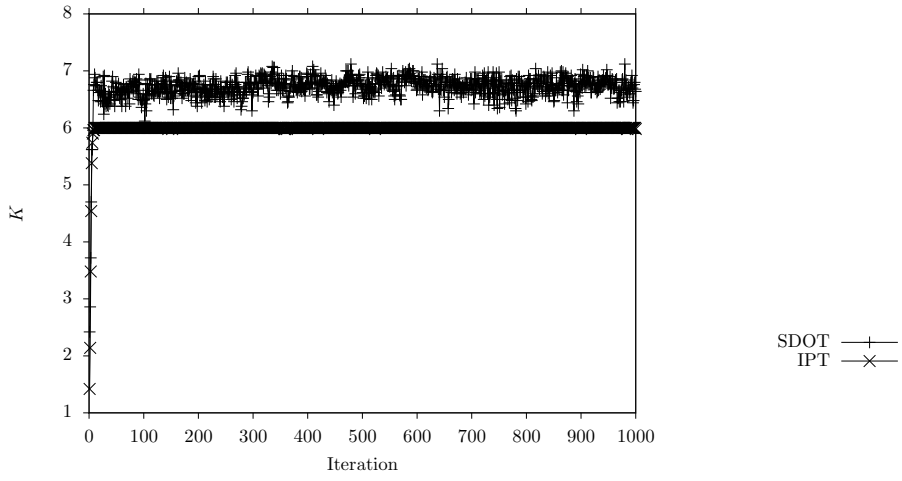

Figure 12: Convergence of LNNAIS using SDOT and IPT to optimal K for hepta data set

Table 4: Descriptive Statistics: Hepta

Algorithm	K	J_{intra}	J_{inter}	Q_{ratio}	Q_{DB}
KM_{DB}	7.00 (± 0.00)	0.993 (± 0.199)	4.041 (± 0.148)	1.112 (± 0.459)	0.870 (± 0.247)
KM_{RT}	4.00 (± 0.00)	1.680 (± 0.083)	3.902 (± 0.184)	0.630 (± 0.419)	1.006 (± 0.153)
LNN_{DB}	6.98 (± 0.14)	0.740 (± 0.122)	4.161 (± 0.097)	0.371 (± 0.259)	0.494 (± 0.219)
LNN_{RT}	5.98 (± 0.14)	1.019 (± 0.052)	4.307 (± 0.146)	0.316 (± 0.059)	0.661 (± 0.049)
LNN_{SDOT}	6.64 (± 1.21)	0.830 (± 0.397)	4.120 (± 0.231)	1.015 (± 4.978)	0.541 (± 0.365)

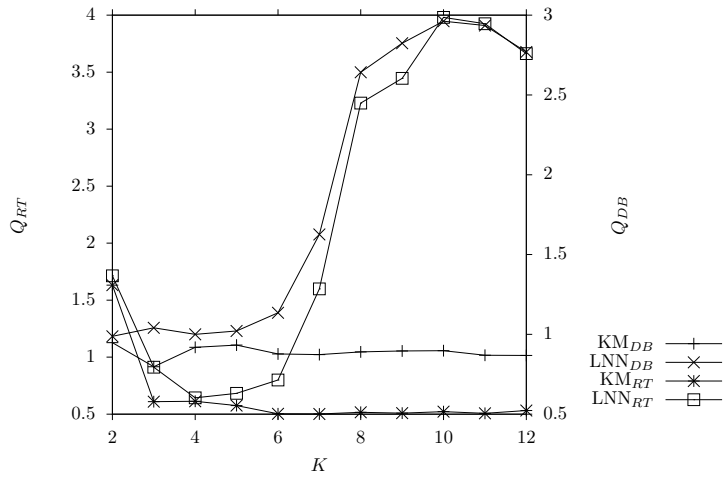

Figure 13: Optimal number of clusters obtained by K-means and LNNAIS for the engytime data set

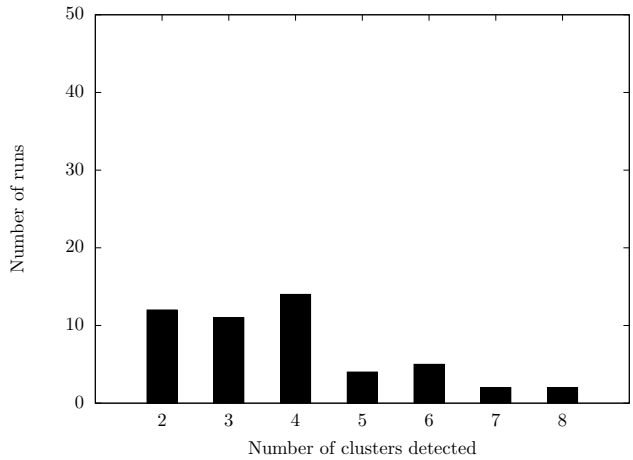

Figure 14: Histogram of the number of clusters detected in the engytime data set by LNN$_{SDOT}$

for frequency distribution). Figure 15 illustrates that IPT obtains $K = 4$ for all iterations and SDOT oscillates around an average K of 4.4 over time for the engytime data set. There is no statistically significant difference between the clustering quality of any of the models (refer to table 12). Therefore, all models tend to deliver clusters with similar quality. LNN$_{SDOT}$ has the advantage of dynamically determining the number of clusters in the engytime data set with similar clustering quality as the other models.

7.5. Chainlink data set

The optimal range of K for the chainlink data set is $[8, 12]$ (as illustrated in figure 16). Figure 17 illustrates that LNN$_{SDOT}$ has high frequencies for $K = 2$ and $4 \leq K \leq 7$ which are not within the optimal range of K. However, the figure also shows that there are cases where LNN$_{SDOT}$ determined the number of clusters within the optimal range of K at lower frequencies. Note that the similarity between the range of determined clusters in figure 17 and the range of K for the iterative and multiple execution approaches in figure 16 is a coincidence. Figure 18 illustrates that IPT obtains $K = 8$ for all iterations and SDOT oscillates around an average K of 6.5 between $K = 5.5$ and $K = 8$ over time for the chainlink data set. The average number of clusters determined by LNN$_{SDOT}$ for the chainlink data set is $K = 5.76$ (refer to table 6). Table 6 shows the results obtained by the different models to determine the optimal number of clusters in the chainlink data set.

Referring to table 12, the statistical hypothesis test rejects H_0 that the Q_{ratio} means are the same between KM$_{RT}$ and LNN$_{SDOT}$ ($z = 8.483$, $p < 0.001$). There is thus a statistical significant difference between the clustering quality of KM$_{RT}$ and LNN$_{SDOT}$. KM$_{RT}$ tends to find clusters in the chainlink data set with a higher quality than LNN$_{SDOT}$. There is also a statistical significant difference between the Q_{ratio} means of LNN$_{RT}$ and LNN$_{SDOT}$ ($z = 2.547$, $p = 0.011$). LNN$_{RT}$ tends to find clusters in the chainlink data set with a higher quality than LNN$_{SDOT}$.

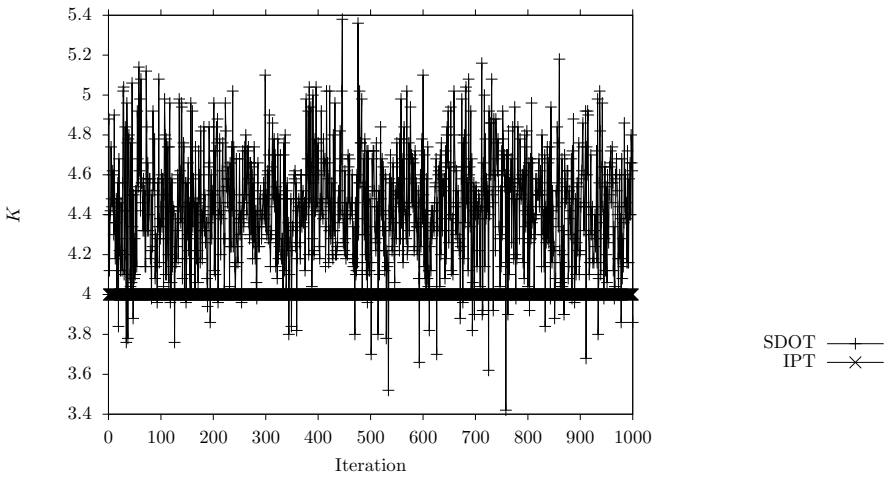

Figure 15: Convergence of LNNAIS using SDOT and IPT to optimal K for engytime data set

Table 5: Descriptive Statistics: Engytime

Algorithm	K	J_{intra}	J_{inter}	Q_{ratio}	Q_{DB}
KM$_{DB}$	3.00	1.165	3.184	0.396	0.797
	($\pm$ 0.00)	($\pm$ 0.000)	($\pm$ 0.000)	($\pm$ 0.000)	($\pm$ 0.000)
KM$_{RT}$	7.00	0.805	3.188	0.502	0.873
	($\pm$ 0.00)	($\pm$ 0.004)	($\pm$ 0.109)	($\pm$ 0.021)	($\pm$ 0.017)
LNN$_{DB}$	2.00	1.833	4.133	0.465	0.910
	($\pm$ 0.00)	($\pm$ 0.213)	($\pm$ 1.032)	($\pm$ 0.107)	($\pm$ 0.194)
LNN$_{RT}$	4.00	1.284	4.020	0.616	1.000
	($\pm$ 0.00)	($\pm$ 0.113)	($\pm$ 0.712)	($\pm$ 0.226)	($\pm$ 0.258)
LNN$_{SDOT}$	3.86	1.381	3.978	0.582	0.992
	($\pm$ 1.62)	($\pm$ 0.304)	($\pm$ 0.808)	($\pm$ 0.217)	($\pm$ 0.287)

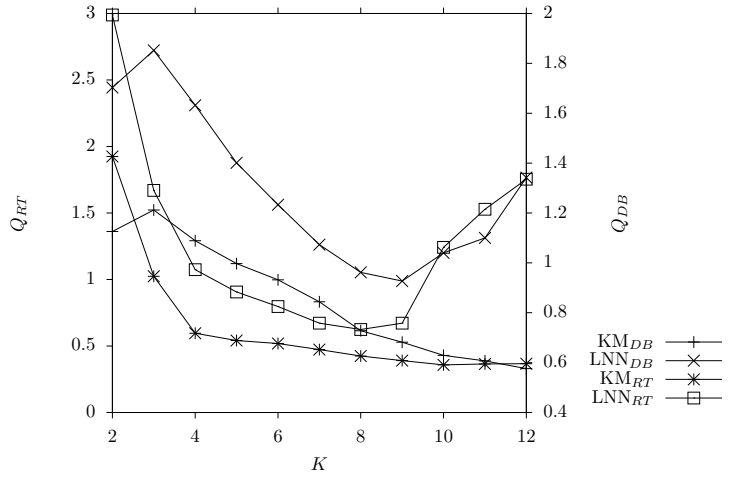

Figure 16: Optimal number of clusters obtained by K-means and LNNAIS for the chainlink data set

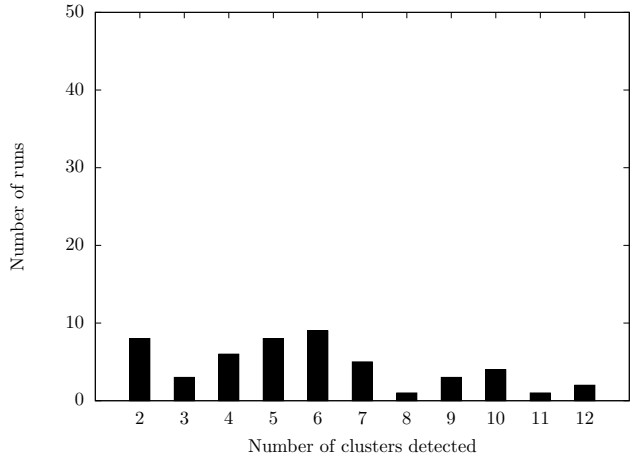

Figure 17: Histogram of the number of clusters detected in the chainlink data set by LNN_{SDOT}

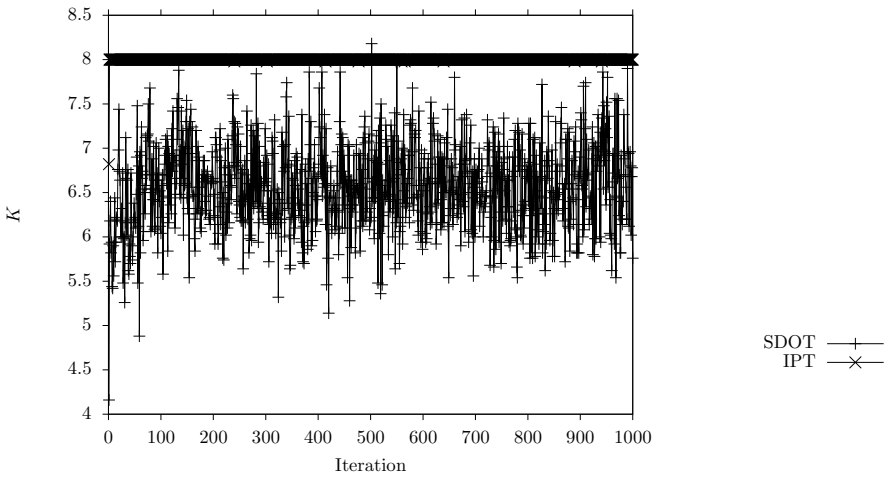

Figure 18: Convergence of LNNAIS using SDOT and IPT to optimal K for chainlink data set

Table 6: Descriptive Statistics: Chainlink

Algorithm	K	J_{intra}	J_{inter}	Q_{ratio}	Q_{DB}
KM$_{DB}$	12.00	0.262	1.500	0.367	0.576
	($\pm$ 0.00)	($\pm$ 0.009)	($\pm$ 0.025)	($\pm$ 0.063)	($\pm$ 0.017)
KM$_{RT}$	10.00	0.308	1.509	0.358	0.629
	($\pm$ 0.00)	($\pm$ 0.007)	($\pm$ 0.031)	($\pm$ 0.028)	($\pm$ 0.030)
LNN$_{DB}$	9.00	0.384	1.475	0.629	0.906
	($\pm$ 0.00)	($\pm$ 0.018)	($\pm$ 0.068)	($\pm$ 0.210)	($\pm$ 0.144)
LNN$_{RT}$	8.00	0.427	1.464	0.624	0.962
	($\pm$ 0.00)	($\pm$ 0.021)	($\pm$ 0.057)	($\pm$ 0.302)	($\pm$ 0.190)
LNN$_{SDOT}$	5.76	0.588	1.402	0.770	1.283
	($\pm$ 2.76)	($\pm$ 0.184)	($\pm$ 0.235)	($\pm$ 0.400)	($\pm$ 0.666)

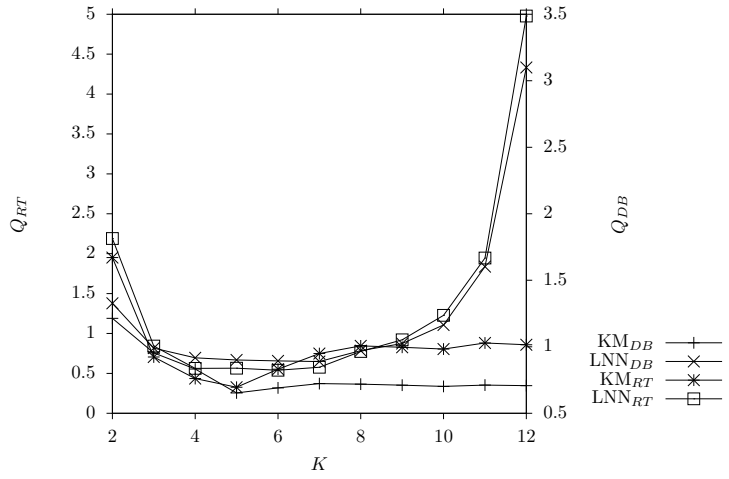

Figure 19: Optimal number of clusters obtained by K-means and LNNAIS for the target data set

7.6. Target data set

The average number of clusters determined by LNN_{SDOT} for the target data set is $K = 4.04$ which is close to the optimal range of K (as illustrated in figure 19, $5 \leq K \leq 8$). The frequency distribution of the number of clusters determined by LNN_{SDOT} for the target data set is illustrated in figure 20. LNN_{SDOT} has high frequencies for $K \leq 5$. Figure 21 illustrates for the target data set the number of clusters respectively determined by the SDOT and IPT techniques over time. IPT obtains $K = 6$ for the majority of the iterations. The value of K for SDOT oscillates between $K = 3$ and $K = 5.5$ around an average K of 4.2 for the remaining iterations. Table 7 shows the results obtained by the different models to determine the optimal number of clusters in the target data set.

The statistical hypothesis test rejects H_0 that the Q_{ratio} means are the same between KM_{RT} and LNN_{SDOT} ($z = 7.835, p < 0.001$). There is thus a statistical significant difference between the clustering quality of KM_{RT} and LNN_{SDOT} and KM_{RT} tends to find clusters in the target data set with a higher quality than LNN_{SDOT}. There is however no statistical significant difference between the Q_{ratio} means of LNN_{RT} and LNN_{SDOT} (statistical hypothesis test accepts H_0, refer to table 12).

7.7. Ionosphere data set

Table 8 shows the results obtained by the different models to determine the optimal number of clusters in the ionosphere data set. Figure 22 illustrates that the optimal range of K for the ionosphere data set is $2 \leq K \leq 5$ (also shown in table 8).

LNN_{SDOT} determined the average number of clusters in the ionosphere data set as $K = 8.28$. The frequency distribution of the number of clusters determined by LNN_{SDOT} for the ionosphere data set illustrates that LNN_{SDOT} has high frequencies for $8 \leq K \leq 11$ which is not within the optimal range of K (refer to figure 23 for frequency distribution). Figure 24 illustrates for

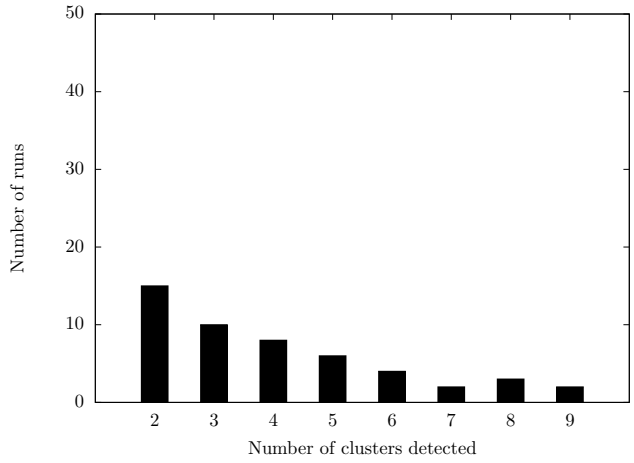

Figure 20: Histogram of the number of clusters detected in the target data set by LNN$_{SDOT}$

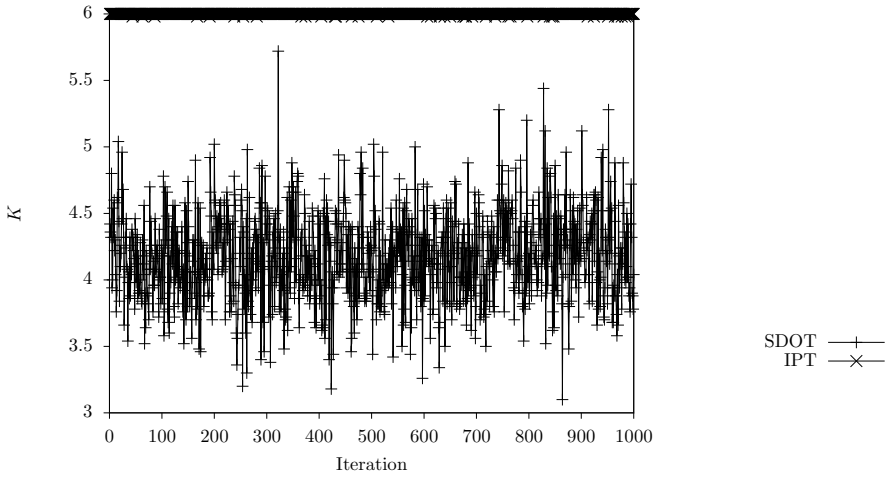

Figure 21: Convergence of LNNAIS using SDOT and IPT to optimal K for target data set

Table 7: Descriptive Statistics: Target

Algorithm	K	J_{intra}	J_{inter}	Q_{ratio}	Q_{DB}
KM_{DB}	5.00	0.533	2.313	0.326	0.653
	($\pm$ 0.00)	($\pm$ 0.012)	($\pm$ 0.102)	($\pm$ 0.013)	($\pm$ 0.014)
KM_{RT}	5.00	0.533	2.313	0.326	0.653
	($\pm$ 0.00)	($\pm$ 0.012)	($\pm$ 0.102)	($\pm$ 0.013)	($\pm$ 0.014)
LNN_{DB}	7.98	0.538	3.076	0.569	0.836
	($\pm$ 0.14)	($\pm$ 0.075)	($\pm$ 0.343)	($\pm$ 0.477)	($\pm$ 0.284)
LNN_{RT}	6.00	0.661	2.806	0.539	0.894
	($\pm$ 0.00)	($\pm$ 0.117)	($\pm$ 0.417)	($\pm$ 0.178)	($\pm$ 0.225)
LNN_{SDOT}	4.04	0.878	2.841	0.577	1.024
	($\pm$ 2.04)	($\pm$ 0.208)	($\pm$ 0.751)	($\pm$ 0.438)	($\pm$ 0.860)

Table 8: Descriptive Statistics: Ionosphere

Algorithm	K	J_{intra}	J_{inter}	Q_{ratio}	Q_{DB}
KM_{DB}	2.00	2.289	3.156	0.730	1.484
	($\pm$ 0.00)	($\pm$ 0.098)	($\pm$ 0.413)	($\pm$ 0.039)	($\pm$ 0.153)
KM_{RT}	4.00	2.085	3.438	0.877	1.776
	($\pm$ 0.00)	($\pm$ 0.065)	($\pm$ 0.481)	($\pm$ 0.164)	($\pm$ 0.283)
LNN_{DB}	2.00	2.888	4.083	0.720	1.437
	($\pm$ 0.00)	($\pm$ 0.278)	($\pm$ 0.642)	($\pm$ 0.100)	($\pm$ 0.257)
LNN_{RT}	5.00	2.473	4.277	0.911	1.755
	($\pm$ 0.00)	($\pm$ 0.272)	($\pm$ 0.517)	($\pm$ 0.180)	($\pm$ 0.258)
LNN_{SDOT}	8.28	2.251	5.012	2.791	1.956
	($\pm$ 2.12)	($\pm$ 0.322)	($\pm$ 0.424)	($\pm$ 6.519)	($\pm$ 1.737)

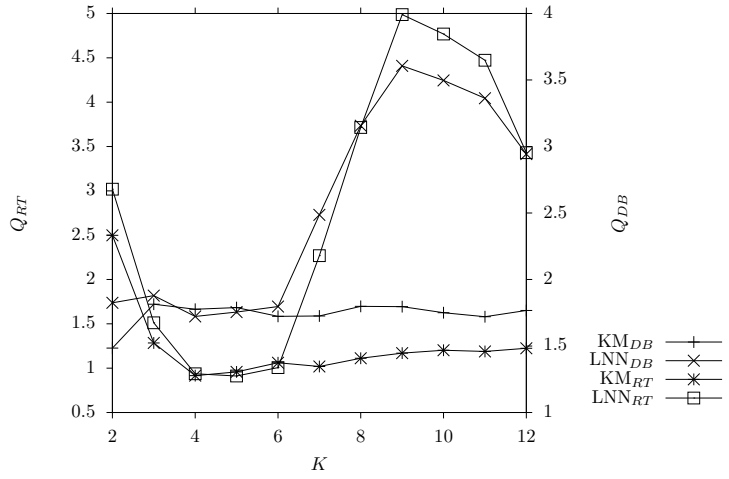

Figure 22: Optimal number of clusters obtained by K-means and LNNAIS for the ionosphere data set

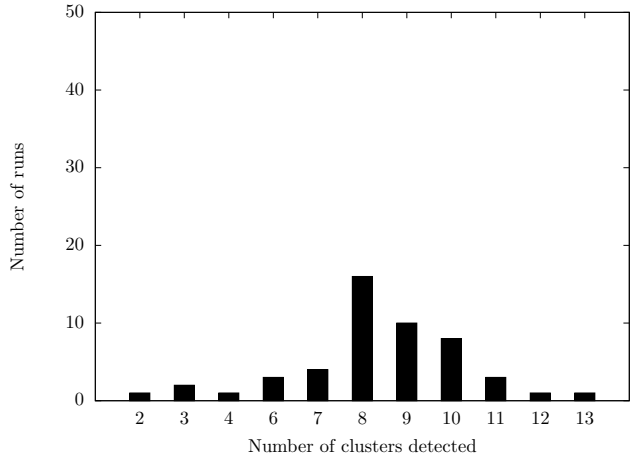

Figure 23: Histogram of the number of clusters detected in the ionosphere data set by LNN_{SDOT}

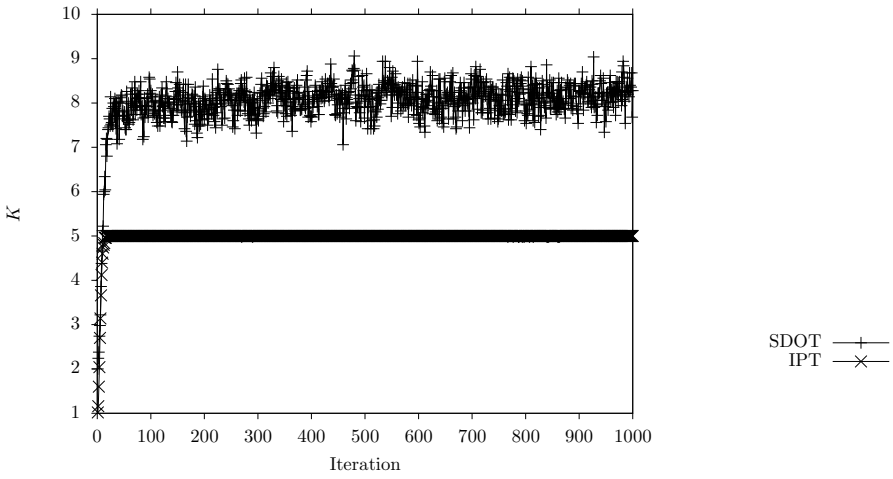

Figure 24: Convergence of LNNAIS using SDOT and IPT to optimal K for ionosphere data set

the ionosphere data set the number of clusters respectively determined by the SDOT and IPT techniques over time. The value of K for IPT rapidly increases to 5 in the first few iterations and remains at 5 for the majority of the remaining iterations. The value of K for SDOT rapidly increases to 8 and oscillates between $K = 7$ and $K = 9$ around an average K of 8 for the remaining iterations. Even though there is a difference in the optimal range of K between the models, there is no statistically significant difference between the clustering qualities of any of the models (refer to table 12). Therefore, all models tend to deliver clusters with similar quality at different optimal number of clusters. LNN$_{SDOT}$ has the advantage of dynamically determining the number of clusters in the ionosphere data set with similar clustering quality as the other models.

7.8. Glass data set

Figure 25 shows that the optimal number of clusters in the glass data set is obtained by KM$_{DB}$ and LNN$_{DB}$ at $K = 2$ and by KM$_{RT}$ and LNN$_{RT}$ at $K = 4$. Therefore the optimal range of K as determined by the different models for the glass data set is $[2, 4]$. Figure 27 illustrates that the value of K for IPT rapidly increases to $K = 4$ and SDOT oscillates around an average K of 3.6 in range $[3; 4.5]$ over time for the glass data set. Table 9 shows the results obtained by the different models to determine the number of clusters in the glass data set. The average number of clusters determined by LNN$_{SDOT}$ is $K = 3.34$ which falls within the optimal range of K.

A histogram of the frequency distribution of the number of clusters determined by LNN$_{SDOT}$ for the glass data set is illustrated in figure 26. LNN$_{SDOT}$ has high frequencies for $K \leq 5$. Referring to table 12, the Mann-Whitney U statistical hypothesis test rejects H_0 that the Q_{ratio} means are the same between KM$_{RT}$ and LNN$_{SDOT}$ ($z = 3.364$, $p < 0.001$) and between LNN$_{RT}$ and LNN$_{SDOT}$ ($z = 1.996$, $p = 0.046$). LNN$_{SDOT}$ tends to find clusters in the glass data set with a higher quality than KM$_{RT}$ and LNN$_{RT}$.

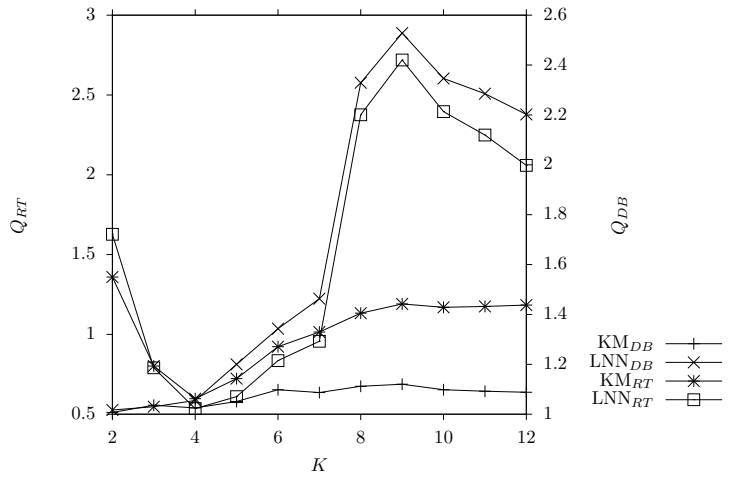

Figure 25: Optimal number of clusters obtained by K-means and LNNAIS for the glass data set

Table 9: Descriptive Statistics: Glass

Algorithm	K	J_{intra}	J_{inter}	Q_{ratio}	Q_{DB}
KM_{DB}	2.00	1.531	3.879	0.397	1.007
	($\pm$ 0.00)	($\pm$ 0.100)	($\pm$ 0.546)	($\pm$ 0.019)	($\pm$ 0.116)
KM_{RT}	4.00	1.212	4.263	0.572	1.025
	($\pm$ 0.00)	($\pm$ 0.056)	($\pm$ 0.627)	($\pm$ 0.152)	($\pm$ 0.149)
LNN_{DB}	2.00	2.354	5.792	0.427	0.892
	($\pm$ 0.00)	($\pm$ 0.484)	($\pm$ 1.379)	($\pm$ 0.121)	($\pm$ 0.236)
LNN_{RT}	4.00	1.575	5.197	0.512	1.055
	($\pm$ 0.00)	($\pm$ 0.208)	($\pm$ 0.769)	($\pm$ 0.161)	($\pm$ 0.266)
LNN_{SDOT}	3.34	2.003	5.998	0.493	0.875
	($\pm$ 1.56)	($\pm$ 0.518)	($\pm$ 0.929)	($\pm$ 0.310)	($\pm$ 0.291)

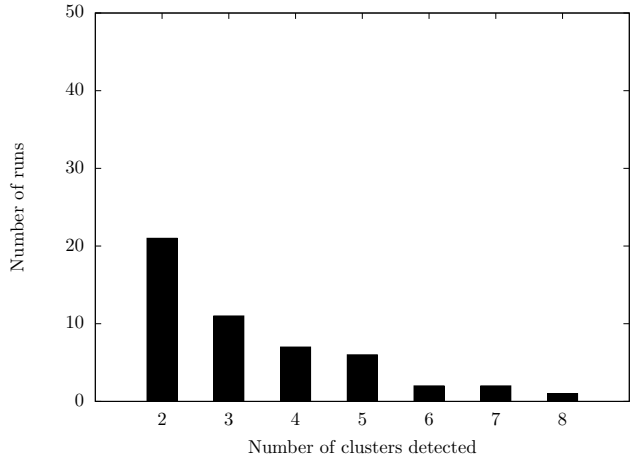

Figure 26: Histogram of the number of clusters detected in the glass data set by LNN$_{SDOT}$

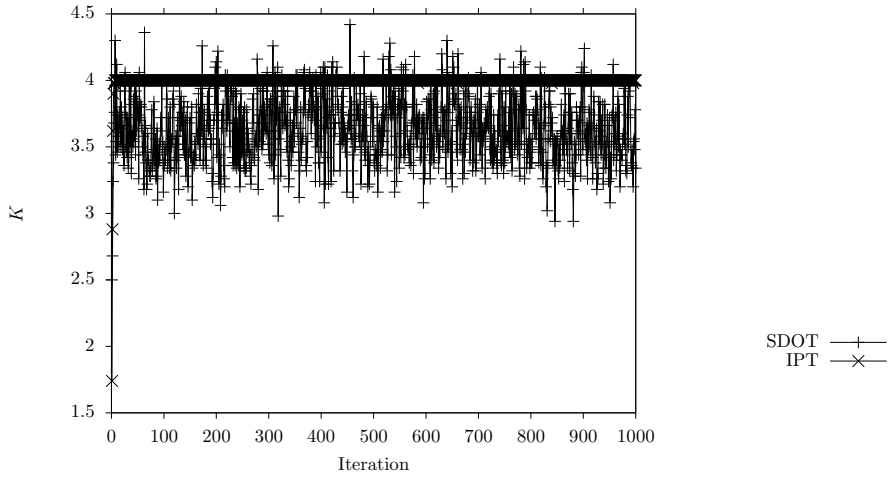

Figure 27: Convergence of LNNAIS using SDOT and IPT to optimal K for glass data set

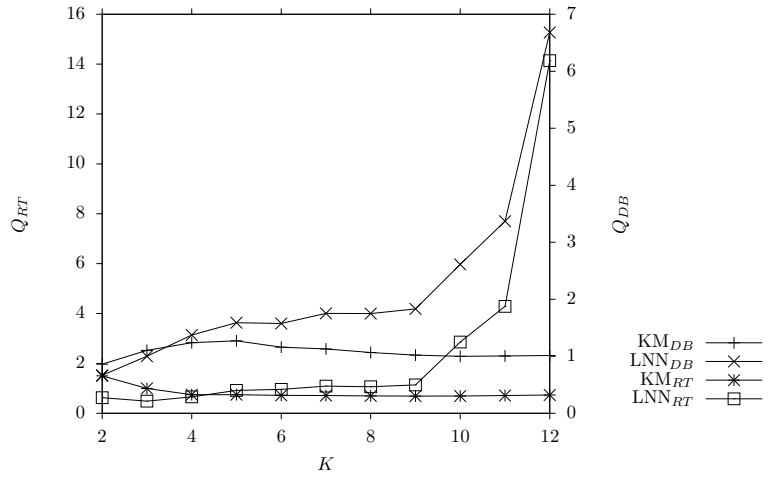

Figure 28: Optimal number of clusters obtained by K-means and LNNAIS for the image segmentation data set

7.9. Image Segmentation data set

Table 10 shows the results obtained by the different models to determine the optimal number of clusters in the image segmentation data set. Figure 28 shows that the optimal number of clusters in the image data set is obtained by KM_{DB} and LNN_{DB} at $K = 2$, by KM_{RT} at $K = 9$ and LNN_{RT} at $K = 3$. The average number of clusters determined by LNN_{SDOT} is $K = 3.28$ which falls within the optimal range of K. Figure 30 illustrates that IPT obtains $K = 3$ for all iterations and SDOT oscillates around an average K of 3.2 in range $[2.6; 3.7]$ over time for the image data set. The frequency distribution of the number of clusters determined by LNN_{SDOT} for the image segmentation data set is illustrated in figure 29. LNN_{SDOT} has high frequencies for $K \leq 5$. Referring to table 12, the Mann-Whitney U statistical hypothesis test rejects H_0 that the Q_{ratio} means are the same between KM_{RT} and LNN_{SDOT} ($z = 6.89, p < 0.001$) and between LNN_{RT} and LNN_{SDOT} ($z = 2.337, p = 0.019$). LNN_{SDOT} tends to find clusters in the image segmentation data set with a higher quality than KM_{RT} and LNN_{RT}.

7.10. Spambase data set

The average number of clusters determined by LNN_{SDOT} for the spambase data set is $K = 2.4$ which is close to the optimal range of K (as illustrated in figure 31, $2 \leq K \leq 4$). In figure 31, note that $Q_{RT} < 0$ for LNN_{RT} where $K \geq 10$. Q_{RT} values less than zero indicates that LNN_{RT} was unable to cluster the data set into the corresponding K clusters. Since $\mathcal{B}_{max} = 10$ for data set spambase (refer to table 1), the number of clusters $K \geq 10$ is more than the number of available ALCs in the population. The frequency distribution of the number of clusters determined by LNN_{SDOT} for the spambase data set is illustrated in figure 32. LNN_{SDOT} has high frequencies for $K \leq 3$. Figure 33 illustrates that IPT obtains $K = 2$ for all iterations and SDOT oscillates around an average K of 2.45 in range $[2.2; 2.7]$ over time for the spam base data set. Table 11 shows the results obtained by the different models to determine the optimal number of clusters in the spambase data set. The statistical hypothesis test rejects H_0 that the Q_{ratio} means are the same between KM_{RT} and LNN_{SDOT} ($z = 8.269, p < 0.001$). There is

Table 10: Descriptive Statistics: Image Segmentation

Algorithm	K	J_{intra}	J_{inter}	Q_{ratio}	Q_{DB}
KM_{DB}	2.00 (± 0.00)	101.487 (± 3.313)	238.922 (± 83.995)	0.439 (± 0.041)	0.861 (± 0.024)
KM_{RT}	9.00 (± 0.00)	58.442 (± 0.675)	322.656 (± 10.779)	0.688 (± 0.083)	1.021 (± 0.035)
LNN_{DB}	2.00 (± 0.00)	168.497 (± 33.481)	1148.026 (± 253.897)	0.155 (± 0.047)	0.551 (± 0.199)
LNN_{RT}	3.00 (± 0.00)	137.827 (± 15.718)	881.290 (± 141.104)	0.316 (± 0.253)	1.000 (± 0.602)
LNN_{SDOT}	3.28 (± 1.27)	142.847 (± 21.735)	975.017 (± 215.461)	43.919 (± 291.181)	88.577 (± 613.169)

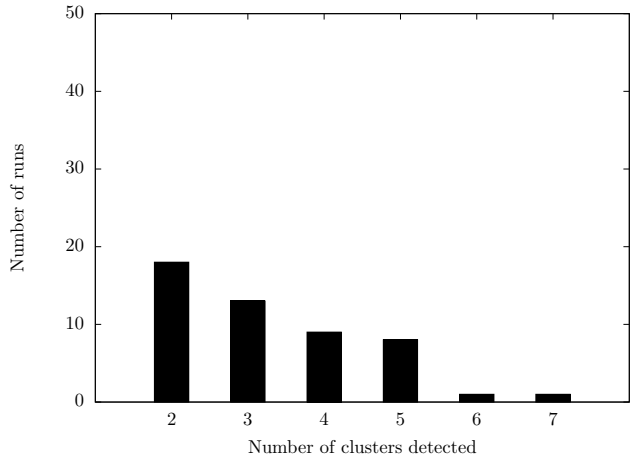

Figure 29: Histogram of the number of clusters detected in the image segmentation data set by LNN_{SDOT}

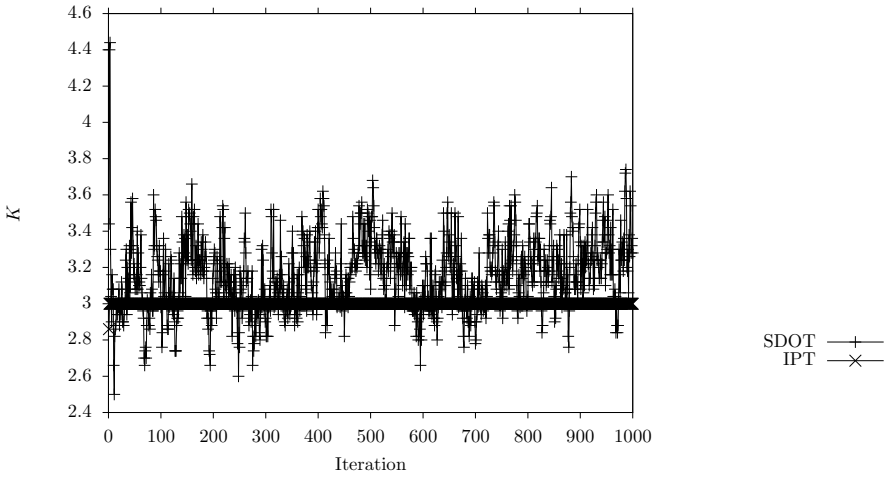

Figure 30: Convergence of LNNAIS using SDOT and IPT to optimal K for image data set

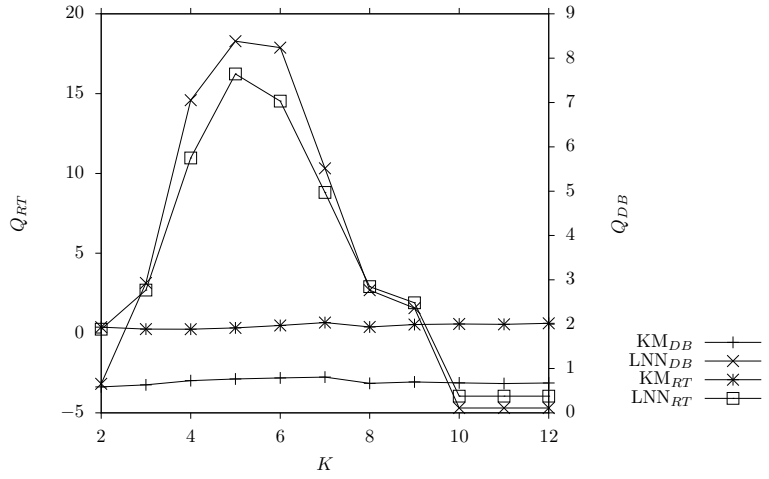

Figure 31: Optimal number of clusters obtained by K-means and LNNAIS for the spam base data set

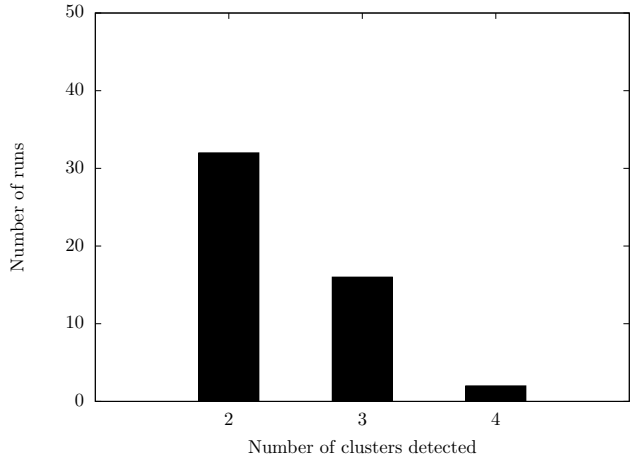

Figure 32: Histogram of the number of clusters detected in the spam base data set by LNN_{SDOT}

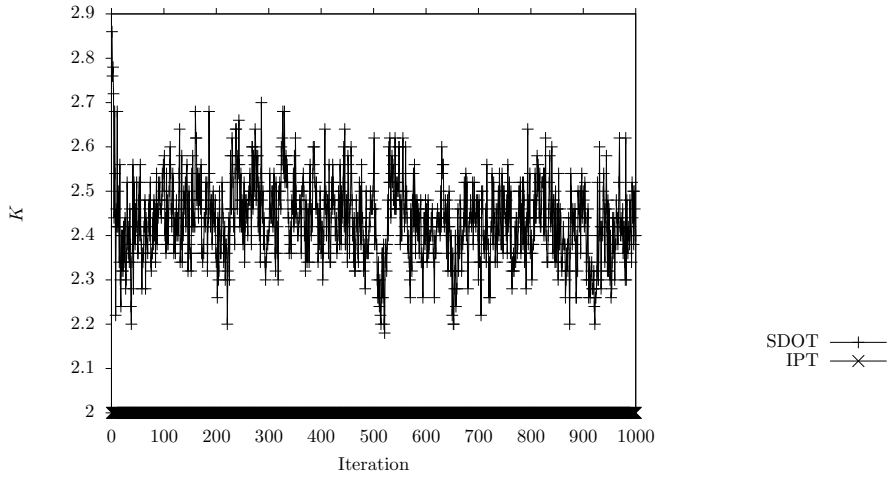

Figure 33: Convergence of LNNAIS using SDOT and IPT to optimal K for spam base data set

Table 11: Descriptive Statistics: Spambase

Algorithm	K	J_{intra}	J_{inter}	Q_{ratio}	Q_{DB}
KM$_{DB}$	2.00	216.058	2003.263	0.108	0.586
	($\pm$ 0.00)	($\pm$ 0.000)	($\pm$ 0.000)	($\pm$ 0.000)	($\pm$ 0.000)
KM$_{RT}$	4.00	129.353	2165.832	0.229	0.727
	($\pm$ 0.00)	($\pm$ 0.000)	($\pm$ 0.000)	($\pm$ 0.000)	($\pm$ 0.000)
LNN$_{DB}$	2.00	771.637	8288.589	0.095	0.546
	($\pm$ 0.00)	($\pm$ 317.716)	($\pm$ 2462.940)	($\pm$ 0.031)	($\pm$ 0.077)
LNN$_{RT}$	2.00	475.834	7639.878	0.071	0.655
	($\pm$ 0.00)	($\pm$ 282.100)	($\pm$ 2648.505)	($\pm$ 0.053)	($\pm$ 0.171)
LNN$_{SDOT}$	2.40	651.896	10416.929	0.076	0.548
	($\pm$ 0.57)	($\pm$ 382.136)	($\pm$ 2798.913)	($\pm$ 0.042)	($\pm$ 0.222)

thus a statistical significant difference between the clustering quality of KM$_{RT}$ and LNN$_{SDOT}$ and LNN$_{SDOT}$ tends to find clusters in the spambase data set with a higher quality than KM$_{RT}$. There is however no statistical significant difference between the Q_{ratio} means of LNN$_{RT}$ and LNN$_{SDOT}$ (statistical hypothesis test accepts H_0, refer to table 12).

For completeness table 12 also shows whether there is a statistical significant difference between the clustering quality of KM$_{RT}$ and LNN$_{RT}$ for all the data sets. Referring to table 12, for two of the data sets (engytime and ionosphere) LNN$_{SDOT}$ and LNN$_{RT}$ tend to deliver clusters with a similar quality as KM$_{RT}$. Out of the remaining eight data sets, both LNN$_{SDOT}$ and LNN$_{RT}$ deliver clusters of a higher quality than KM$_{RT}$ for five of the data sets. Comparing LNN$_{SDOT}$ with LNN$_{RT}$, for five of the data sets (twospiral, engytime, target, ionosphere and spambase) LNN$_{SDOT}$ tends to deliver clusters with a similar quality as LNN$_{RT}$. Out of the remaining five data sets, LNN$_{SDOT}$ delivers clusters of a higher quality than LNN$_{RT}$ for four of the data sets. In general LNN$_{SDOT}$ tends to deliver clusters of similar or higher quality for all data sets, followed by LNN$_{RT}$ and KM$_{RT}$.

Table 12: Statistical Hypothesis Testing between All Models for all data sets based on Q_{ratio} as performance criteria ($\alpha = 0.05$; with continuity correction; unpaired; non-directional)

Data set		z of A	z of B	p	Outcome	Lowest z
Model A	Model B					
iris						
LNN$_{SDOT}$	KM$_{RT}$	-7.58	7.58	< 0.001	Reject H_0	LNN$_{SDOT}$
LNN$_{RT}$	KM$_{RT}$	-3.209	3.209	0.001	Reject H_0	LNN$_{RT}$
LNN$_{SDOT}$	LNN$_{RT}$	-6.69	6.69	< 0.001	Reject H_0	LNN$_{SDOT}$
twospiral						
LNN$_{SDOT}$	KM$_{RT}$	8.328	-8.328	< 0.001	Reject H_0	KM$_{RT}$
LNN$_{RT}$	KM$_{RT}$	7.704	-7.704	< 0.001	Reject H_0	KM$_{RT}$
LNN$_{SDOT}$	LNN$_{RT}$	-0.5	0.5	0.617	Accept H_0	LNN$_{SDOT}$

Continued on next page

Model A	Model B	z of A	z of B	p	Outcome	Lowest z
hepta						
LNN_{SDOT}	KM_{RT}	-6.787	6.787	< 0.001	Reject H_0	LNN_{SDOT}
LNN_{RT}	KM_{RT}	-8.145	8.145	< 0.001	Reject H_0	LNN_{RT}
LNN_{SDOT}	LNN_{RT}	-4.391	4.391	< 0.001	Reject H_0	LNN_{SDOT}
engytime						
LNN_{SDOT}	KM_{RT}	1.017	-1.017	0.309	Accept H_0	KM_{RT}
LNN_{RT}	KM_{RT}	1.551	-1.551	0.121	Accept H_0	KM_{RT}
LNN_{SDOT}	LNN_{RT}	-0.855	0.855	0.393	Accept H_0	LNN_{SDOT}
chainlink						
LNN_{SDOT}	KM_{RT}	8.483	-8.483	< 0.001	Reject H_0	KM_{RT}
LNN_{RT}	KM_{RT}	8.566	-8.566	< 0.001	Reject H_0	KM_{RT}
LNN_{SDOT}	LNN_{RT}	2.547	-2.547	0.011	Reject H_0	LNN_{RT}
target						
LNN_{SDOT}	KM_{RT}	7.835	-7.835	< 0.001	Reject H_0	KM_{RT}
LNN_{RT}	KM_{RT}	8.145	-8.145	< 0.001	Reject H_0	KM_{RT}
LNN_{SDOT}	LNN_{RT}	-0.221	0.221	0.825	Accept H_0	LNN_{SDOT}
ionosphere						
LNN_{SDOT}	KM_{RT}	0.955	-0.955	0.340	Accept H_0	KM_{RT}
LNN_{RT}	KM_{RT}	1.169	-1.169	0.243	Accept H_0	KM_{RT}
LNN_{SDOT}	LNN_{RT}	0.283	-0.283	0.777	Accept H_0	LNN_{RT}
glass						
LNN_{SDOT}	KM_{RT}	-3.364	3.364	< 0.001	Reject H_0	LNN_{SDOT}
LNN_{RT}	KM_{RT}	-1.965	1.965	0.049	Reject H_0	LNN_{RT}
LNN_{SDOT}	LNN_{RT}	-1.996	1.996	0.046	Reject H_0	LNN_{SDOT}
image segmentation						
LNN_{SDOT}	KM_{RT}	-6.89	6.89	< 0.001	Reject H_0	LNN_{SDOT}
LNN_{RT}	KM_{RT}	-7.18	7.18	< 0.001	Reject H_0	LNN_{RT}
LNN_{SDOT}	LNN_{RT}	-2.337	2.337	0.019	Reject H_0	LNN_{SDOT}
spambase						
LNN_{SDOT}	KM_{RT}	-8.269	8.269	< 0.001	Reject H_0	LNN_{SDOT}
LNN_{RT}	KM_{RT}	-8.269	8.269	< 0.001	Reject H_0	LNN_{RT}
LNN_{SDOT}	LNN_{RT}	1.275	-1.275	0.202	Accept H_0	LNN_{RT}

8. Conclusion and Future Work

This paper presented two techniques which can be used with LNNAIS to dynamically determine the number of clusters in a data set. These techniques are the iterative pruning technique (IPT) and the sequential deviation outlier technique (SDOT). Although both of these techniques are computationally less expensive than the multiple execution approaches, the IPT technique either needs a specified range for K or needs to iterate through all possible edges (to a maximum of $\mathcal{B}_{max}$) which makes the technique parameter dependent in the former case and slightly more computationally expensive than SDOT in the latter. An advantage of IPT is that the technique

can use any cluster validity index to determine the number of clusters. The SDOT technique neither uses a cluster validity index nor does it require any boundary constraints on K. SDOT is a non-parametric technique. This is an advantage, since it is not always feasible to visually inspect to formed clusters and a specified range for K might not contain the optimum number of clusters.

LNN$_{RT}$, LNN$_{DB}$ (both using IPT with Q_{RT} and Q_{DB}, respectively) and LNN$_{SDOT}$ (using SDOT) were applied on different data sets to determine the optimal number of clusters. These results were compared to the results obtained from K-means clustering which used the multiple execution approach to determine the optimal number of clusters in each data set. Based on the Q_{ratio} index, in general LNN$_{SDOT}$ tends to deliver clusters of similar or higher quality for all data sets, followed by LNN$_{RT}$ and KM$_{RT}$.

Since the LNN$_{SDOT}$ model is computationally less expensive and is able to dynamically determine the number of clusters in a data set, the model can be seen as an enhancement to the LNNAIS model. Future work on the LNN$_{SDOT}$ model includes the application of the model in clustering non-stationary environments. Due to the possibility of the LNN$_{SDOT}$ model to dynamically determine the number of clusters, the model might indicate the division or merging of clusters in a non-stationary environment. The definition of a non-stationary environment and the stability of the LNN$_{SDOT}$ model in such an environment need to be investigated.

References

[1] M. Halkidi, Y. Batistakis, M. Vazirgiannis, Cluster validity methods: part i, ACM SIGMOD Record 31 (2002) 40–45.

[2] F. Kovács, C. Legány, A. Babos, Cluster validity measurement techniques, in: 6th International Symposium of Hungarian Researchers on Computational Intelligence, Budapest.

[3] S. Ray, R. H. Turi, Determination of number of clusters in k-means clustering and application in colour image segmentation, The 4th International Conference on Advances in Pattern Recognition and Digital Techniques, Calcuta (1999).

[4] M. D. G. Toledo, A Comparison in Cluster Validation Techniques, Master thesis, University of Puerto Rico, 2005.

[5] J. Timmis, M. Neal, A resource limited artificial immune system for data analysis, in: Research and Development in Intelligent Systems XVII, volume 14, Springer, Cambridge, UK., 2000, pp. 19–32.

[6] M. Neal, Meta-stable memory in an artificial immune network, In Artificial Immune Systems: Proceedings of ICARIS (2003) 168–180.

[7] O. Nasraoui, C. Cardona, C. Rojas, F. Gonzalez, Mining evolving user profiles in noisy web clickstream data with a scalable immune system clustering algorithm, in: Workshop Notes of WEBKDD 2003: Web Mining as Premise to Effective and Intelligent, pp. 71–81.

[8] L. N. de Castro, F. J. V. Zuben, An evolutionary immune network for data clustering, in: IEEE Brazilian Symposium on Artificial Neural Networks, Rio de Janeiro, pp. 84–89.

[9] L. de Castro, F. V. Zuben, AiNet: An Artificial Immune Network for Data Analysis, Idea Group Publishing, USA, pp. 231–259.

[10] N. K. Jerne, Towards a network theory of the immune system, Annals of Immunology (Inst. Pasteur) 125C (1974) 373–89. PMID: 4142565.

[11] A. J. Graaff, A. P. Engelbrecht, A local network neighbourhood artificial immune system for data clustering, in: D. Srinivasan, L. Wang (Eds.), 2007 IEEE Congress on Evolutionary Computation, IEEE Press, Singapore, 2007, pp. 260–267.

[12] A. J. Graaff, A. P. Engelbrecht, Towards a self regulating local network neighbourhood artificial immune system for data clustering, in: IEEE Congress on Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence), pp. 633–640.

[13] C. Y. Lee, E. K. Antonsson, Dynamic partitional clustering using evolutionary strategies, Proc. of the 3rd Asia?Pacific Conference on Simulated Evolution and Learning (2000).

[14] A. Jain, M. Murty, P. Flynn, Data clustering: a review, ACM Comput. Surv. 31 (1999) 264–323.

[15] E. Forgy, Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications, Biometrics 21 (1965) 768.
[16] D. L. Davies, D. W. Bouldin, A cluster separation measure, IEEE Transactions on Pattern Analysis and Machine Intelligence 1 (1979) 224–227.
[17] M. Halkidi, Y. Batistakis, M. Vazirgiannis, Clustering validity checking methods: part ii, ACM SIGMOD Record 31 (2002) 19–27.
[18] S. Jonnalagadda, R. Srinivasan, An information theory approach for validating clusters in microarray data., in: Joint Conference Intelligent Systems for Molecular Biology (ISMB) and European Conference on Computational Biology (ECCB), Glasgow, UK.
[19] R. H. Turi, Clustering-Based Colour Image Segmentation, PhD Thesis, Ph.D. thesis, Monash University, Australia, 2001.
[20] G. H. Ball, D. J. Hall, A clustering technique for summarizing multivariate data, Behavioral Science 12 (1967) 153–155.
[21] J. T. Tou, DYNOC—A dynamic optimal cluster-seeking technique, International Journal of Parallel Programming 8 (1979) 541–547.
[22] K. Y. Huang, A synergistic automatic clustering technique (SYNERACT) for multispectral image analysis, Photogrammetric engineering and remote sensing 68 (2002) 33–40.
[23] C. J. Veenman, M. J. Reinders, E. Backer, A maximum variance cluster algorithm, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (2002) 1273–1280.
[24] A. W. Moore, D. Pelleg, X-means: Extending k-means with efficient estimation of the number of clusters, in: Proceedings of the Seventeenth International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA, 2000, pp. 727–734.
[25] R. E. Kass, L. Wasserman, A reference bayesian test for nested hypotheses and its relationship to the schwarz criterion, Journal of the American Statistical Association 90 (1995) 928–934.
[26] G. Hamerly, C. Elkan, Learning the k in K-Means, The Seventh Annual Conference on Neural Information Processing Systems 17 (2003).
[27] C. S. Wallace, D. L. Dowe, Intrinsic classification by MML-the snob program, in: Proceedings of the 7th Australian Joint Conference on Artificial Intelligence, Armidale, NSW, Australia, pp. 37–44.
[28] C. Rosenberger, K. Chehdi, Unsupervised clustering method with optimal estimation of the number of clusters: Application to image segmentation, in: International Conference on Pattern Recognition, volume 1, pp. 1656–1659.
[29] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, 2001.
[30] Xie, Chen, Yu, Sequence Outlier Detection Based on Chaos Theory and Its Application on Stock Market, pp. 1221–1228.
[31] A. Arning, R. Agrawal, P. Raghavan, A linear method for deviation detection in large databases, Proc. KDD (1996) 164–169.
[32] A. K. Jain, R. C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Englewood Cliffs, New Jersey, 1988.
[33] A. Asuncion, D. Newman, UCI machine learning repository, 2007.