A NON-PARAMETERISED HIERARCHICAL POLE BASED CLUSTERING ALGORITHM (HPOBC)

Amparo Albalate, Steffen Rhinow
Institute of Information Technology, University of Ulm, Ulm, Germany
amparo.albalate@uni-ulm.de, steffen.rhinow@uni-ulm.de

David Suendermann
SpeechCycle Labs, NY, USA
david@speechcycle.com

Keywords: Divisive clustering, PoBOC, Silhouette width.

Abstract: In this paper we propose a hierarchical, divisive, clustering algorithm, called Hierarchical Pole Based Clustering (HPoBC), which is able to find the clusters in a data set without any user input parameter such as the number of clusters k. The algorithm is based on the Pole Based Overlapping Clustering (PoBOC) (Cleuziou et al., 2004). Initially, the top hierarchy level is composed by the set of clusters discovered by the PoBOC algorithm on the dataset. Then, each single cluster is again analysed using a combination of PoBOC and cluster validity methods (silhouettes) in order to search for new possible subclusters. This process is recursively repeated on each newly retrieved cluster until the silhouette score suggests to stop any further partitioning of the cluster. The HPoBC algorithm has been compared to the original PoBOC as well as other classical hierarchical approaches on five two-dimensional, synthetic data sets, using three cluster evaluation metrics.

1 INTRODUCTION

Cluster analysis refers to techniques used to discover the group structure in a certain data set. These algorithms have multiple applications, such as image segmentation, text mining, or the analysis of genomic and sensorial data, among others.

A large variety of clustering techniques have been proposed over the past decades. Because no prior knowledge is required about the object's group labels, clustering algorithms are unsupervised learning models. However, most algorithms in the clustering literature are parameterised approaches, i.e, the cluster solutions depend on some user input parameters descriptive for the dataset. Typical input parameters are the target number of clusters, or density indicators in density models.

The Pole Based Overlapping Clustering algorithm (PoBOC), proposed in (Cleuziou et al., 2004), is an overlapping, graph-based clustering approach which does not require any input information from the user. The algorithm iteratively identifies a set of initial cluster prototypes, and builds the clusters around these objects based on an object's neighbourhood notion.

One limitation of the PoBOC algorithm is related to the neighbourhood formulation applied to extract the final clusters. The neighboorhood of one object is defined in terms of the object's average distance to all other objects in the data set (see Section 2.1). This global definition can be suitable for discovering uniformly spread clusters on the data space. However, the algorithm may fail to identify all existing clusters if the input data are organised in a hierarchy of classes, in such a way that two or more subclasses are closer to each other than the average class distance.

In order to overcome this limitation, we propose a new hierarchical algorithm based on PoBOC, called "Hierarchical Pole-Based Clustering" (HPoBC). The hierarchy of clusters and subclusters is detected though a recursive approach. First, the PoBOC algorithm is used to identify the clusters in the data set, also referred to as "poles". Next, under the hypothesis that more subclusters may exist inside any pole, PoBOC is again locally applied to each initial pole. A cluster validity based on silhouette widths is then used in order to validate or reject the subcluster hypothesis. If the subcluster hypothesis is rejected by the silhouette score, we discard the candidate subclusters and

select the initial pole as part of the output clusters. Otherwise (the hypothesis is validated) we store the new identified poles (subclusters) and continue with a similar analysis inside each new pole. This procedure is applied recursively until the silhouette rejects every further hypothesis.

The paper is organised as follows: In Section 2 an overview of the PoBOC algorithm is presented. In Section 3, we introduce some of the classical hierarchical clustering approaches. In Section 4, the new HPoBC algorithm is described in detail. Finally, in Sections 5 and 6, we present evaluation results and draw conclusions, respectively.

2 THE POLE BASED OVERLAPPING CLUSTERING (PoBOC)

The Pole Based Overlapping Clustering is an overlapping, graph-based clustering technique proposed by (Cleuziou et al., 2004). The algorithm takes the matrix of object dissimilarities as single input and builts the output clusters in four main steps: (i) Definition of dissimilarity graph, (ii) construction of poles, (iii) pole restriction and (iv) affectation of objects to poles.

2.1 Definition of a dissimilarity graph

Let $\mathcal{X}$ denote the set of n data points (objects) in the data set, and D the dissimilarity matrix, computed over $\mathcal{X}$.

The dissimilarity graph $G(\mathcal{X}, V, D)$ is then specified by: (i) the dissimilarity matrix D, (ii) the data points or vertices, $\mathcal{X}$, and a set of edges V between all pairs of vertices (x_i, x_j) corresponding to mutual neighbour points.

Definition 1: (Neighbourhood of a point x): The neighbourhood of a point x, denoted by $N(x)$ is composed of all points of $\mathcal{X}$ whose dissimilarity to the point is smaller than the mean distance of the object x to all other objects in $\mathcal{X}$ ($d_{mean}(x;\mathcal{X})$):
$N(x) = \{x_j \in \mathcal{X} | D_{x_j,x} < d_{mean}(x;\mathcal{X})\ \}$

Definition 2: Two points (x_i, x_j) are mutual neighbours, and thus connected by an edge in the Graph G if each one belongs to the neighbourhood of the other:
$(x_i, x_j) \in V \leftrightarrow x_i \in N(x_j); x_j \in N(x_i)$

2.2 Pole Construction

This procedure builds incrementally a set of poles $\mathcal{P} = \{P_1, P_2 \ldots, P_k\}$. Let $\{\mathcal{O}\}$ denote the cumulated set of objects that belong to any of the extracted poles up to the current state (initially the empty set).

The poles are grown from initial points $\hat{x_i}$, which are the points with maximum mean distance to the cumulated set of poles $\mathcal{O}$. Initially, the object $\hat{x_0}$ with maximum distance to $\mathcal{X}$ is selected:

$$\hat{x_0} = \underset{x \in \mathcal{X}}{\operatorname{argmax}}\, d_{mean}(x, \mathcal{X}) \quad (1)$$

$$\hat{x_i} = \underset{x \in \mathcal{X}/\mathcal{O}}{\operatorname{argmax}}\, d_{mean}(x, \mathcal{O}) \quad (2)$$

Each P_i pole is then grown from the corresponding initial object $\hat{x_i}$, in such a way that all the pole members are enclosed in their respective neighbourhoods. This is implemented in the pole-construction procedure:

Algorithm 1 pole-construction($\hat{x}$, $G(\mathcal{X}, V, D)$)

Initialise $P = \hat{x}$
Obtain neighbourhood of P:
$N(P) = \{x \in \mathcal{X} | \forall x_i \in P, (x, x_i) \in V\}$
while $N(P) \neq \emptyset$ **do**
  attach the object x to P such that:
  $x \in N(P)$ and $x = \text{argmax}_{x_i \in N(P)}\, d_{mean}(x_i, P)$
  Update $N(P)$
  Return the resulting pole P
end while

The selection of the initial object $\hat{x_i}$ and the construction of the corresponding pole P_i is iteratively repeated until all objects in the data set are contained in any of the poles, $\{\mathcal{O}\} = \{\mathcal{X}\}$, or no initial object can be found which is sufficiently distant from the set of poles.

2.3 Pole Restriction

After the pole construction, overlapping objects may be obtained, which simultaneously belong to the neighbourhood of two or more poles. These objects compose the residual set $\{\mathcal{R}\}$. The pole restriction procedure consists of removing residual objects from the original poles, resulting in a new set of reduced, non-overlapping poles: $\tilde{\mathcal{P}}$.

2.4 Affectation Stage

The residual objects $\mathcal{R}$ obtained at the pole restriction stage require some post-processing strategy, in order to be reallocated into one or more of the restricted poles. This reallocation of objects in PoBOC is called affectation. First, the membership of each object x to each $\tilde{P_i}$ restricted pole, $u(x, \tilde{P_i})$ is computed as:

$$u(x, \tilde{P_i}) = 1 - \frac{d_{mean}(x, \tilde{P_i})}{D_{max}} \quad (3)$$

Then, the objects are affected to one or more poles. In a single-affectation approach, each object x is assigned to the pole maximising the membership $u(x, P_j)$. In a multi-affectation approach, the object is affected to the poles whose memberships are greater than some reference values given by a linear approximation on the set of object memberships, sorted in decreasing order.

3 HIERARCHICAL CLUSTERING

Classical approaches for hierarchical clustering obtain the cluster solution by iterative mergings or divisions of clusters (Everitt, 1974; Kaufmann and Rousseeuw, 1990). Two major hierarchical approaches can be distinguished: agglomerative and divisive.

Hierarchical agglomerative approaches Agglomerative algorithms are the so-called bottom-up approaches, starting with all points as individual clusters and successively merging the closest pair of clusters until all patterns are enclosed in a single cluster. The algorithms can be visualised using a graphical tree structure called dendogram where the pair of clusters that are merged at each iteration can be observed. The final cluster solution is selected by the user, by specifying a level to cut the dendogram or, equivalently, a desired number k of clusters. Different agglomerative approaches can be distinguished, depending on the proximity criterion to merge the next pair of clusters. For example, while the *single linkage* algorithm selects the pair of clusters with the minimum distance between their closest elements, the *complete linkage* algorithm selects the clusters with minimum distance between the farthest objects. In a similar way, the *average linkage* and *centroid* algorithms choose the clusters with the minimum average inter cluster distance and the minimum distance between their centroid objects, respectively.

Hierarchical divisive approaches As opposed to agglomerative algorithms, a divisive approach, such as the divisive analysis (DiANA) algorithm, starts at the top dendogram level where all objects compound a unique cluster and iteratively splits the biggest cluster until each object is in its own cluster. The reader is referred to (Everitt, 1974) for more details about the Divisive Analysis algorithm.

4 NEW HIERARCHICAL POLE-BASED APPROACH

The new clustering method is combination of the PoBOC algorithm and hierarchical divisive clustering strategies. In a divisive manner, the proposed hierarchical approach is initialised with the set of poles identified by the $PoBOC$ algorithm, and recursively applied to each obtained pole, searching for possible subclusters.

4.1 Pole-Based Clustering basis module

In order to detect the set of poles in the new hierarchical approach, we preserve the graph construction, pole construction and pole restriction stages of POBOC, but the affectation step has been replaced by a new procedure called *pole regrowth*:

Algorithm 2 pole regrowth($\{\tilde{\mathcal{P}}\}, \{\mathcal{R}\}$)

Input: set of poles and residual from the pole-reduction step: $\{\tilde{\mathcal{P}}\}, \{\mathcal{R}\}$)
Output: set of regrown poles $\{\hat{\mathcal{P}}\}$
Initialisation $\{\hat{\mathcal{P}}\} = \{\tilde{\mathcal{P}}\}$
while $R <> \emptyset$ **do**
  Find the pair (point $\in \{\mathcal{R}\}$, pole $\in \{\hat{\mathcal{P}}\}$) with minimum distance:
  $(x_i, \hat{P}_j) = \text{argmin}_{x \in R}(\text{argmin}_{\hat{P}}(D_{min}(x, \hat{P}_k)))$,
  $D_{min}(x_i, \hat{P}_j) = min(D_{x_i, x_j \in \tilde{P}_j})$
  Attach the point x_i to its closest pole and remove it from the residual set:
  Update $\hat{P}_j \leftarrow \hat{P}_j, x_i$
  Update $\{\mathcal{R}\} \leftarrow \{\mathcal{R}\} - x_i$
end while
Return $\{\hat{\mathcal{P}}\}$

The pole-regrowth procedure is an alternative to the PoBOC single affectation for reallocating overlapping objects into one of the restricted poles. As it can be observed in Figure 1(a), not only a pole but also an overlapping region may contain potential subclusters. If each overlapping object x_i is individually assigned to the pole maximising the membership $u(x_i; P)$, the objects inside a single cluster might be assigned to different poles[1]. The pole regrowth procedure is intended to avoid any undesired partitioning of clusters existing in overlapping areas while reallocating residual objects.

An example of the pole regrowth method is shown in Figure 1. Figure 1(a) shows two restricted poles in red and green colours, respectively. All points between these restricted poles are overlapping points. It

[1]Recall that the hierarchical approach is independently applied to the grown poles

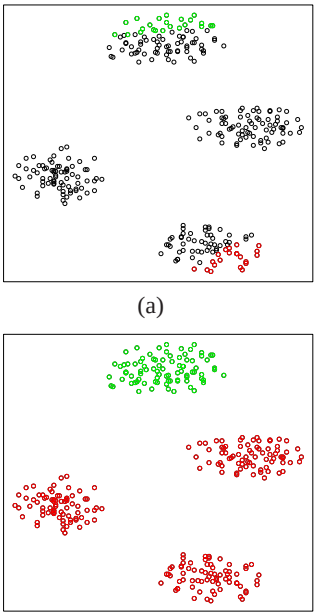

Figure 1: Example of the pole growth.1(a) Two restricted poles and their overlapping objects. 1(b) New poles obtained after the reallocation of overlapping objects by the pole regrowth method.

can be observed that many of these overlapping points build another two clusters, which PoBOC fails to detect. The reallocation of overlapping points by the pole regrowth procedure is illustrated in Figure 1(b). A single affectation would have splitted each overlapping cluster in two halfs (upper and bottom). Using the pole regrowth, all objects inside each overlapping cluster have been jointly assigned into a single pole. This fact allows to detect the overlapping clusters in further recursive steps.

We refer to the modified PoBOC algorithm as Pole-Based clustering module, which is the basis for the hierarchical approach described in the following paragraphs:

Algorithm 3 Pole-Based clustering module($C, \mathcal{X}$)

Input:
C: Matrix of coordinates (attributes) of all points in the data set,
$\mathcal{X}$: Indexes of the points (rows of C) to be clustered.
Output: Regrown poles $\{\hat{\mathcal{P}}\}$.
Compute dissimilarity matrix of C over $\mathcal{X}$: D_X
Compute dissimilarity graph over $\mathcal{X}$, $G_X(\mathcal{X}, V_X, D_X)$
$\{\mathcal{P}\} \leftarrow$ Pole Construction $(G_X(\mathcal{X}, V_X, D_X))$
$\{\tilde{\mathcal{P}}\}, \{R\} \leftarrow$ Pole Restriction $(\{\mathcal{P}\})$
$\{\hat{\mathcal{P}}\} \leftarrow$ Pole Regrow $(\{\tilde{\mathcal{P}}\}, \{R\}, D_X)$
Return $\{\hat{\mathcal{P}}\}$.

4.2 Hierarchical Pole-Based Clustering (HPoBC)

The proposed algorithm is called *Hierarchical Pole Based Clustering (HPoBC)*.

First, the Pole Based Clustering module is applied to the entire dataset to obtain an initial set of poles. Then, a recursive function, the *Pole Based Subcluster Analysis* is triggered on each pole with more than one object. If an individual pole is found, the corresponding object is attached to the set of final clusters as an individual cluster. This recursive function is continuously called with the objects of each obtained pole, internally denoted $P^{\hat{top}}$, because it refers to an upper level in the hierarchy. Analogously, the new set of poles found on $P^{\hat{top}}$ is denoted $P^{\hat{sub}}$, indicating a lower hierarchy level. These poles represent candidate subclusters. In order to decide wether $P^{\hat{sub}}$ compounds "true" subclusters or not, a criterion typically used for cluster validity is applied, namely the *silhouette width* (Rousseeuw, 1987). The silhouette width of a cluster partition $\mathcal{C}$ returns a quality score in the range [-1,1] where 1 corresponds to a perfect clustering. According to (Treeck, 2005), a silhouette score smaller or equal than 0.25 is an indicator for wrong cluster solutions. However, from our experiments, a more rigorous threshold $sil > 0.5$ has proven adequate for validating the candidate subclusters. The problem of deciding whether a data set contains a cluster structure or not is commonly referred to as cluster tendency in the cluster literature (Jolion and Rosenfeld, 1989). If the quality criterion is not fulfilled ($sil < 0.5$) the subcluster hypothesis is rejected, and the top cluster $P^{\hat{top}}$ is attached to the final clusters. Otherwise, we continue exploring each subcluster in order to search for more possible sublevels.

Algorithm 4 Hierarchical Pole-Based Clustering - HPoBC($\mathcal{X}$)

Initialisation
Points indexes $C \leftarrow 1, \ldots, length(\mathcal{X})$
Set of Cluster Solution: Clusters : $\{\emptyset\}$
Obtain set of grown poles on all $\mathcal{X}$ objects:
$\hat{\mathcal{P}} \leftarrow$ Pole-Based Clustering Module($\mathcal{X}, C$)
for all $\{\hat{P}_i\} \in \{\hat{\mathcal{P}}\}$ **do**
  if $|\{\hat{P}_i\}| > 1$ **then**
    Trigger recursive search for subclusters:
    Pole-Based Subcluster Analysis ($\mathcal{X}$, $\{\hat{P}_i\}$, Clusters)
  else
    Add $\{\hat{P}_i\}$ to Clusters
  end if
end for
Return Clusters

Algorithm 5 Pole-Based Subcluster Analysis ($\mathcal{X}$, $\{\hat{P}^{top}\}$,Clusters)

$\{\hat{P}^{sub}\} \leftarrow$ Pole-Based Clustering Module($\mathcal{X}$, $\{\hat{P}^{top}\}$)
stop $\leftarrow$ (silhouette-width($\{\hat{\mathcal{P}^{sub}}\}$) ≤ 0.5)
if stop=true **then**
  Add $\{\hat{P}^{top}\}$ to Clusters:
  Return
else
  for all $\{\hat{P}_i^{sub}\} \in \{\hat{\mathcal{P}^{sub}}\}$ **do**
    if $|\{\hat{P}_i^{sub}\}| > 1$ **then**
      Pole-Based Subcluster Analysis ($\mathcal{X}$, $\{\hat{P}_i^{sub}\}$, Clusters)
    else
      Add $\{\hat{P}_i^{sub}\}$ to Clusters
      Return
    end if
  end for
end if

5 EVALUATION METHODS

The PoBOC algorithm as well as the hierarchical pole based clustering (HPoBC) have been compared to other hierarchical approaches: the single, complete, centroid and average linkage and the divisive analysis (DiANA) algorithm. These classical hierarchical algorithms are examples of clustering approaches that require the target number of clusters (k) in order to find the cluster solutions. In order to allow for a comparison of PoBOC and HPoBOC to the hierarchical agglomerative approaches, these algorithms have been called with different values of the k parameter, and the silhouette index has been applied to validate each solution and predict the optimum number of clusters, k_{opt}. Note that, while the Silhouette index is used in agglomerative algorithms and DiANA as a cluster validity strategy to select the optimum k among a set of K possible cluster solutions, in the hierarchical Pole Based algorithm, the Silhouette scores are applied in a recursive and "local" manner, in order to evaluate the cluster tendency inside each obtained pole.

Data sets: The described approaches have been applied to the synthetic data sets of Figure 2: The first dataset (100p5c) comprises 100 objects in 5 spatial clusters (Figure 2 (a)), the second dataset (6Gauss) is a mixture of six Gaussians (1500 points) in two dimensions (Figure 2(e)). The third data set is a mixture of three Gaussians (3Gauss) in which the distance of the biggest class to the other two is larger than the distance among the two smaller Gaussians (Figure 2(i)). This data set illustrates a typical example in which using cluster validity based on Silhouettes may fail to predict the number of classes due to the different interclass distances. The fourth and fifth data (560p8c and 1000p9c) contain 560 and 1000 points in two dimensions, with 8 and 9 spatial clusters, respectively (Figures 2(m) and (q)).

The cluster solutions provided by PoBOC, HPoBC and an example hierarchical agglomerative approach (average linkage) are shown in the plots of Figure 2 (different colours are used to indicate different clusters).

5.1 Cluster evaluation metrics

For a comprehensive evaluation of the discussed algorithms, their cluster solutions have been also compared with the reference category labels, available for evaluation purposes, using three typical "external" cluster validation methods: Entropy, Purity, and Normalised Mutual Information.

Entropy The cluster entropy (Boley et al., 1999) reflects the degree to which the clusters are composed of heterogeneous patterns, ie, patterns that belong to different categories. According to the Entropy criterion, a good cluster should be mostly aligned to a single class, which means that a large number of the cluster objects belong to the same category. This quality condition corresponds to low entropy values. The entropy of a cluster i is defined as:

$$E_i = -\sum_{j=1}^{L} p_{ij} \log_2(p_{ij}) \tag{4}$$

where L denotes the number of reference categories, and p_{ij}, the probability that an element of category j is found in cluster i. This probability can be formulated as $p_{ij} = \frac{m_j}{mi}$, denoting m_j the number of elements of class j in the cluster i, and m_i, the total number of elements in the cluster i.

The total entropy of the cluster solution C is obtained by averaging the cluster entropies according to Equation 5 (m denotes the total number of elements in the data set):

$$E(\mathcal{C}) = \sum_{i=1}^{k} \frac{m_i}{m} E_i \tag{5}$$

As discussed above, "good" cluster solutions yield small entropy values.

Purity Like entropy, purity (Boley et al., 1999; Wu et al., 2009) is a metric to measure the extent to which a cluster contains elements of a single category.

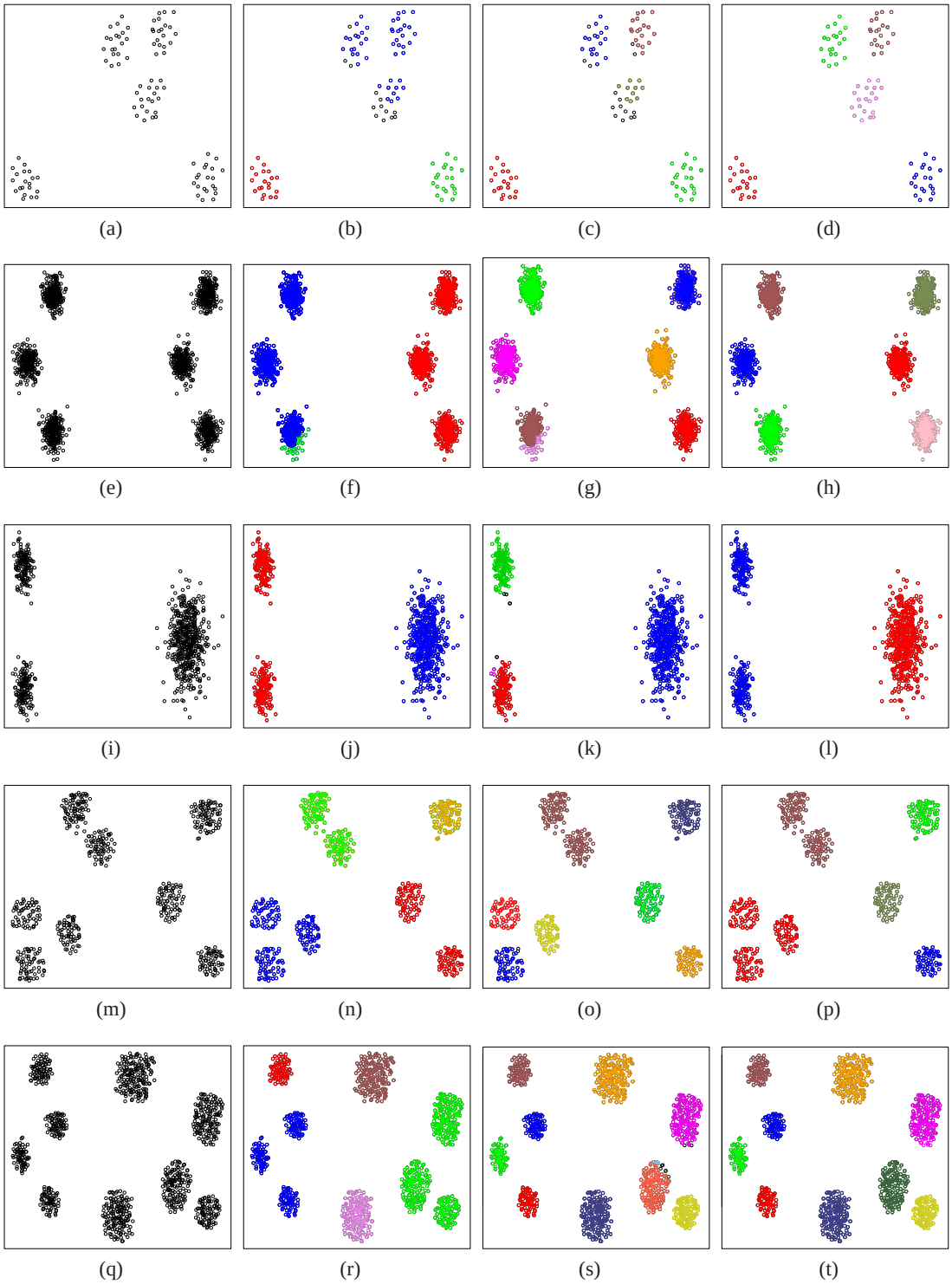

Figure 2: Spatial databases and the extracted clusters using PoBOC and HPoBC and the average linkage clustering algorithms. 2(a): Data set with 100 points in 5 spatial clusters (100p5c), 2(e): mixture of 6 gaussians, 1500 points (6Gauss), 2(i): mixture of three gaussians (3Gauss), 2(m): 560 points, 8 clusters (560p8c) and 2(q):1000 points, 9 clusters (1000p9c). 2(b),2(f), 2(j), 2(n) and 2(r): poles detected by PoBOC in the data sets. 2(c), 2(g), 2(k), 2(o)and 2(t): clusters detected by the new HPoBC algorithm (black circles indicate patterns detected as outliers by the algorithms). 2(d),2(h), 2(l), 2(p) and 2(s): clusters detected by the average linkage algorithm.

The purity of a cluster i is defined in terms of the maximum class probability, $P_i = \max_j (p_{ij})$

The overall purity of a cluster solution is calculated by averaging the cluster purities:

$$P(\mathcal{C}) = \sum_{i=1}^{k} \frac{m_i}{m} P_i \tag{6}$$

Higher purity values indicate a better quality of the clustering solution, up to a purity value equal to one, which is attained when the cluster partition is perfectly aligned to the reference classes.

Normalised Mutual Information (NMI) The Normalised Mutual information (NMI) measures the agreement between two partitions of the data, $\lambda^{(a)}$ and $\lambda^{(n)}$, (Equation 7).

$$NMI(\lambda^{(a)}, \lambda^{(b)}) = \tag{7}$$

$$= \frac{\sum_{h=1}^{k(a)} \sum_{l=1}^{k(b)} n_{h,l} \log\left(\frac{n \cdot n_{h,l}}{n_h^{(a)} n_l^{(b)}}\right)}{\sqrt{\left(\sum_{h=1}^{k(a)} n_h^{(a)} \log\left(\frac{n_h^{(a)}}{n}\right)\right)\left(\sum_{l=1}^{k(b)} n_l^{(b)} \log\left(\frac{n_l^{(b)}}{n}\right)\right)}}$$

Denoting n, the number of observations in the dataset, $k(a)$ and $k(b)$, the number of clusters in the partitions $\lambda^{(a)}$ and $\lambda^{(b)}$; $n_h^{(a)}$ and $n_l^{(b)}$, the number of elements in the clusters C_h and C_l of the partitions $\lambda^{(a)}$ and $\lambda^{(b)}$ respectively, and $n_{h,l}$, the number of overlapping elements between the clusters C_h and C_l. The normalised mutual information can be used as a quality metric of a cluster partition by comparing the cluster solution $\mathcal{C}$ with the reference class labels $\mathcal{L}$, $NMI(C, \mathcal{L})$.

5.2 Results

As can be seen in Tables 1 to 5, the performance of the HPoBC algorithm is consistently superior to the original PoBOC algorithm on all datasets and metrics. The classical (divisive and agglomerative) approaches with the help of Silhouettes to determine the optimum k are also able to detect the class structure in three data sets (100p5c, 1000p9c and 6Gauss). The performance of classical approaches is thus comparable to the HPoBC algorithm on the mentioned data sets. Note that, in some cases, the NMI score achieved by HPoBC is marginally inferior ($\leq 1.8\%$) to other hierarchical approaches, due to the false discovery by HPoBC of tiny clusters in the boundaries of a larger cluster. In contrast to the previous data sets, the DiANA and agglomerative hierarchical approaches fail to capture accurately the existing classes

Table 1: 560p8c Data

Clustering	# Clusters	NMI	Purity	Entropy
DiANA	5	0.850	0.660	0.840
single	5	0.850	0.660	0.840
complete	5	0.850	0.660	0.840
average	5	0.850	0.660	0.840
centroid	5	0.850	0.660	0.840
PoBOC	4	0.801	0.548	1.048
HPoBC	7	0.944	0.867	0.287

Table 2: 100p5c Data

Clustering	# Clusters	NMI	Purity	Entropy
(DiANA)	5	1	1	0
single	5	1	1	0
complete	5	1	1	0
average	5	1	1	0
centroid	5	1	1	0
PoBOC	3	0.801	0.693	0.817
HPoBC	5	1	1	0

on the datasets 560p8c and 6Gauss. The problem lies in the Silhouette scores, which fail to place the maximum (k_{opt}) at the correct number of clusters. This happens because the intra-class separation differs significantly among the clusters. However, this problem is not observed in the HPoBC algorithm, since Silhouette scores are used to evaluate the local cluster tendency. This implies a more "relaxed" condition in comparison to the use of Silhouettes for validating global clustering solutions. Thus, in these cases, the HPoBC algorithm is advantageous with respect to the classical hierarchical approaches, as evidenced by NMI improvements around 10%.

5.3 Complexity considerations for large databases

If denoting n, the total number of objects in the data set, the complexity of the PoBOC algorithm is estimated in the order of $O(n^2)$, similar to the classical hierarchical schemes. The complexity of the Pole Based Hierarchical Clustering depends on factors such as the number and size of poles retrieved at each step and the maximum number of recursive steps necessary to obtain the final cluster solution. The worst case in terms of the algorithm efficiency would happen if a pole with $n-1$ elements were continuously found until all elements composed individual clusters. In this case, the algorithm would reach

Table 3: mixture of six Gaussians

Clustering	# Clusters	NMI	Purity	Entropy
DiANA	6	0.980	0.992	0.049
single	6	1	1	0
complete	6	1	1	0
average	6	1	1	0
centroid	6	1	1	0
PoBOC	3	0.606	0.693	0.817
HPoBC	7	0.982	1	0

a cubic complexity $O(n(n+1)(2n+1))$. In general terms, if k is the number of recursive steps (levels descended in the hierarchy) necessary to reach the solution, the maximum complexity of the algorithm can be approximated as $O(k \cdot n^2)$. As for the analysed datasets, the algorithm needed 3 recursive steps at most to achieve the presented results. It leads to a quadratic complexity, comparable to the PoBOC algorithm and the rest of hierarchical approaches.

Table 4: 1000p9c

Clustering	# Clusters	NMI	Purity	Entropy
DiANA	9	1	1	0
single	9	1	1	0
complete	9	1	1	0
average	9	1	1	0
centroid	9	1	1	0
PoBOC	5	0.837	0.634	0.637
HPoBC	11	0.993	1	0

Table 5: Mixture of 3 Gaussians

Clustering	# Clusters	NMI	Purity	Entropy
DiANA	2	0.847	0.812	0.375
single	2	0.847	0.812	0.375
complete	2	0.847	0.812	0.375
average	2	0.847	0.812	0.375
centroid	2	0.847	0.812	0.375
PoBOC	2	0.847	0.812	0.375
HPoBC	4	0.990	1	0

6 Conclusions

In this paper we present a hierarchical clustering approach based on the Pole Based Clustering algorithm (PoBOC), which only needs the objects in a dataset as input, in contrast to other approaches that require the number of clusters as input parameter. The use of global object distances by PoBOC does not allow to differentiate between subclusters, specially if the data is organised in a hierarchy. We therefore propose a hierarchical version of PoBOC, called HPoBC, that recursively applies into each obtained cluster in order to adapt the object distances to local regions and accurately retrieve clusters as well as subclusters. Results obtained on five spatial databases have proven the better performance of the new hierarchical approach with respect to the baseline PoBOC, also comparable or superior with respect to other traditional hierarchical approaches. However, we need to emphasize the fact that the presented results have been obtained on synthetic data sets with noticeable differences between intercluster distances. In future work we further expect to validate the performance of the HPoBC algorithm on real databases.

REFERENCES

Boley, D., Gini, M., Gross, R., Han, E.-H., Karypis, G., Kumar, V., Mobasher, B., Moore, J., and Hastings, K. (1999). Partitioning-based clustering for web document categorization. *Decis. Support Syst.*, 27(3):329–341.

Cleuziou, G., Martin, L., Clavier, L., and Vrain, C. (2004). Poboc: An overlapping clustering algorithm, application to rule-based classification and textual data. In *Proceedings of the 16th European Conference on Artificial Intelligence ECAI*.

Everitt, B. (1974). *Cluster Analysis*. Heinemann Educ., London.

Jolion, J.-M. and Rosenfeld, A. (1989). Cluster detection in background noise. *Pattern Recogn.*, 22(5):603–607.

Kaufmann, L. and Rousseeuw, P. J. (1990). *Finding Groups in Data. An Introduction to Cluster Analysis*. Wiley, New York.

Rousseeuw, P. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Jornal Comp. Appl. Math.*, 20:53–65.

Treeck, B. (2005). *Entwicklung und Evaluierung einer Java-Schnittstelle zur Clusteranalyse von Peer-to-Peer Netzwerken. Bachelorarbeit*. Heinrich-Heine-Universität Düsseldorf.

Wu, J., Chen, J., Xiong, H., and Xie, M. (2009). External validation measures for k-means clustering: A data distribution perspective. *Expert Syst. Appl.*, 36(3):6050–6061.