

PLinkage: Density Ascending Clustering via Pairwise Linkage

Xiaohu Lu, Jian Yao*, Li Li, Yahui Liu

School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, Hubei, P.R. China

Wei Zhang

School of Control Science and Engineering, Shandong University, Jinan, Shandong, P.R. China

Abstract

In this paper, we present a density ascending clustering algorithm named Pairwise Linkage (PLinkage) for the clustering of Gaussian distributed data. The proposed method is based on the observation that: a data point should be in the same cluster with its closest ascending neighbouring point (CANP), and we call the relationship from this data point to its CANP as a “pairwise linkage”. It is reasonable that a data point should be classified into the same cluster with its closest neighbouring point, while the additional constraint ‘ascending’ makes sure that the pairwise linkages are unidirectionally built from the lower density data points to the higher density ones and finally terminate at the cluster centers. There are three steps for finishing data clustering by the proposed PLinkage method. First, the densities of each data point are calculated. Then, for each data point, its CANP is searched in the neighbourhood. If this data point is the one with the highest density in its neighbourhood we mark it as a cluster center, otherwise, a pairwise linkage is created from this data point to its CANP. Finally, the clusters can be quickly discovered by searching along the linkages starting from these cluster centers in a depth-first or breadth-first way. The proposed PLinkage clustering is non-iterative and can be easily adapted to other classification and segmentation issues. To sufficiently verify the efficiency of our proposed PLinkage clustering algorithm in specific applications, we propose a PLinkage based algorithm for image segmentation. Experimental results on various public

*Corresponding author.

Email address: jian.yao@whu.edu.cn (Jian Yao)

URL: <http://cvrs.whu.edu.cn/> (Jian Yao)

datasets with different dimensional data and the Caltech-256 object category dataset [1] with the CNN (Convolution Neural Network) based features sufficiently demonstrate the efficiency and robustness of the proposed PLinkage clustering algorithm, and also the image segmentation results on the BSDS dataset [2] are presented and analyzed.

Keywords: Pairwise Linkage, Density Ascending Clustering, Image Segmentation.

1. Introduction

Cluster analysis, which is a statistical technique commonly used in determining a profile, has been applied in many kinds of fields, such as data mining especially on the high dimensional spatial data sets [3], machine learning [4], astronomy objects grouping [5], pattern recognition [6], image analysis [7, 8], point cloud segmentation [9], and so on.

There exist possibly over a hundred of clustering algorithms published in the last several decades, not all provide models for their clusters and can thus not easily be categorized. Surveys and classifications of these algorithms can be found in the works of [10, 3]. Different notions giving their definitions and diverse principles describing the relationships between them result in various algorithms. Popular notions of clusters include groups with small distances among the cluster members (K -means [11]), dense areas of the data space (DBSCAN [12, 13]), and intervals or particular statistical distributions (AutoClass [14]). As to the principles, these algorithms can be divided roughly into two categories: partitioning methods and hierarchical ones. The partitioning clustering algorithms usually classify each data point to different clusters via a certain similarity measurement, for example, K -means [11] and CLARANS [15]. The hierarchical methods organize data into a hierarchical structure according to the proximity matrix and get the clustering result via the bottom-up agglomerative clustering [16] or the top-down divisive clustering [17]. There are also many other methods which can not be classified into these two categories, for example, the density based algorithm [18], the globally optimal search-based clustering algorithm [19], the graph-based clustering algorithm [20], and so on. What's noteworthy is that our proposed PLinkage clustering method, which searches the CANP based on the densities of the data points, belongs to the category of the density based algorithms.

25 Different to the various clustering algorithms, the cluster analysis itself is not one spe-
26 cific algorithm, but a general task to be solved. As a general task, there are two concepts
27 in the cluster analysis: (1) *elements*, which include the data points and the cluster cen-
28 ters, in some cases there may exists no cluster centers; (2) *principles*, which gather those
29 elements together. Unlike the traditional classification methods which focus mainly on
30 the categories of *principles*, we give a new view on the classification of clustering meth-
31 ods based on the *elements*, which divides the clustering algorithms into three categories:
32 center based, point-center based and point-point based.

33 The center based methods focus on the properties of the center of each cluster. There
34 is very few work in this field, until recently Rodriguez and Laio [21] proposed a clustering
35 algorithm focusing on two characteristics of the cluster centers: (1) the cluster centers are
36 surrounded by neighbours with lower densities; (2) the cluster centers are at relatively
37 large distances from any point with higher densities. Based on these two characteristics,
38 their method finds out the density peaks (DP) in the following four steps. Firstly, the
39 density ρ of each data point is calculated via a cutoff distance d_c . Then, the minimum
40 distance ϵ between each point and any other point with a higher density is obtained.
41 Thirdly, each data point is plotted out with ρ and ϵ being the X and Y axes, respectively,
42 in which the density peaks are prominent enough to be distinguished with other non-
43 centroid points. After the cluster centers have been found, finally each non-centroid
44 point is assigned to the same cluster as its nearest neighbour with a higher density. The
45 key problem of this DP algorithm lies on the selection of the density peaks. The minimum
46 distance ϵ in the DP method is actually a nonlinear factor which makes the density peaks
47 more prominent, however it is still an unsolved problem to choose the correct density
48 peaks in practical applications.

49 The point-center based methods focus on the relationship between the cluster cen-
50 ter and the corresponding data points. In general, there are two kinds of relationships
51 between the data points and the corresponding cluster centers, one is “assign” the data
52 point to the closest cluster center, the other is “shift” the data point to approach its cor-
53 responding center. Most of the partition clustering methods belong to the first category,

54 including the traditional algorithms, K -means, the K -medoids [22] and the CLARA al-
55 gorithm [17], which assign the data points to the K cluster centers updated in each
56 iteration. As to the “shift” based methods, the mean shift algorithm [23, 24], which
57 iteratively shifts a data point to the nearest stationary point of the underlying density
58 function, is a representative of these methods. The theory of the mean shift is simple,
59 all the data points near a cluster center will converge to it via iterative ascending shift.
60 A variation of the mean shift is the medoid shift method [25], which is similar to that
61 between K -means and K -medoids on the aspect of replacing the mean value with the
62 median value of the neighbouring data points.

63 The point-point based methods focus on the relationship between the data points.
64 In general, there are two relationships between data points: adjacent neighbouring point
65 (ANP) and closest neighbouring point (CNP). The DBSCAN method [12, 13], which
66 requires only one input parameter to discover clusters of arbitrary shapes and distinguish
67 noise even for large spatial databases, is a representative of the ANP based methods. This
68 method is very simple, for each unprocessed data point if it is judged as a core, which
69 means its neighbourhood is denser than a certain threshold, then a new cluster is created
70 and all of its adjacent neighbouring points are added into this cluster, the same occurs
71 on these ANPs that are also judged as cores iteratively until there exist no more cores
72 that can be added. As to the CNP based methods, most of the bottom-up agglomerative
73 hierarchical clustering algorithms belong to this category. The single-linkage method [26],
74 which iteratively combines two clusters that contain the closest pair of elements not yet
75 belonging to the same cluster until all the data points are merged into a single cluster,
76 and its variants complete-linkage and average-linkage are all the CNP based methods.

77 In this paper, we focus on the point-point relationship in the cluster analysis, and
78 give a new point-point relationship definition named “closest ascending neighbouring
79 point (CANP)": a data point \mathbf{p}_i should be in the same cluster with its closest ascending
80 neighbour \mathbf{p}_j . We call the relationship from \mathbf{p}_i to \mathbf{p}_j as a pairwise linkage. The conception
81 of PLinkage is derived from the idea of non-maximum suppression (NMS) [27], in which
82 each data point only needs to be compared with its neighbours and will be suppressed

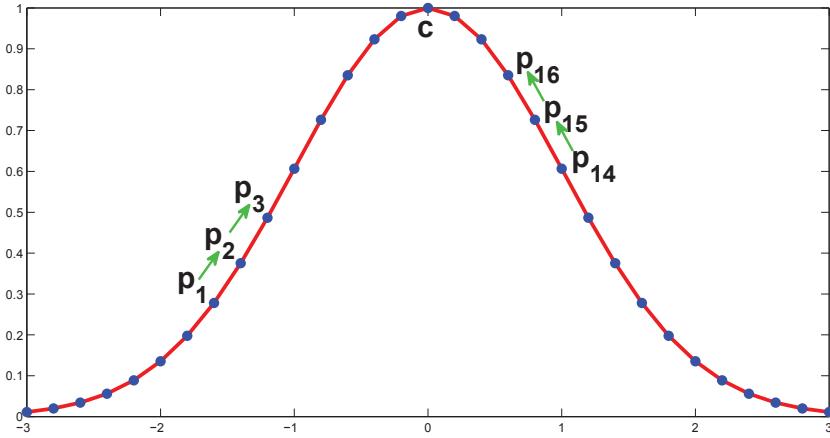


Figure 1: An illustration of the pairwise linkage on a 2D Gaussian curve. Derived from the non-maximum suppression, the PLinkage compares each data point to its neighbours and forms the linkages from $\mathbf{p}_1 \rightarrow \mathbf{p}_2 \rightarrow \mathbf{p}_3 \rightarrow \dots \rightarrow \mathbf{c}$.

if it is not local-maximum. Figure 1 shows an illustration of the NMS, from which we can observe that \mathbf{p}_1 is suppressed by \mathbf{p}_2 and the same suppression occurs on \mathbf{p}_2 when it is compared to \mathbf{p}_3 , which result in a linked list $\mathbf{p}_1 \rightarrow \mathbf{p}_2 \rightarrow \mathbf{p}_3$. In this way, all the data points on the curve are finally linked to the cluster center \mathbf{c} , which is the local-maximum one, just via comparing to their neighbouring points.

Comparing the PLinkage to the DP method [21] and the mean shift algorithm [23], we can find that the proposed PLinkage method shares some similarities with these two algorithms, for example, the conception of “closest neighbouring point” is applied in the DP method after the cluster centers have been found and the “ascending” turns out to be similar with the mean shift algorithm when the data point is on a mountain slope, but as a whole they are not comparable. Above all, the basic idea of PLinkage, “closest ascending neighbouring point”, can be considered as a combination of the “closest neighbouring point” conception from the DP method and the “ascending” from the mean shift algorithm. Besides this, as we have mentioned above, the PLinkage is a point-point based density ascending method which focuses on the linkages between data points, while the DP method is a center based density descending method which focuses on two features ρ and ϵ of the density peaks. The difference between PLinkage and mean shift is that the PLinkage method is non-iterative while the mean shift method is point-center based and

Table 1: Differences among PLinkage, DP, and mean shift.

Algorithms	PLinkage	DP	mean shift
Complexity	$\mathcal{O}(N^2) \rightarrow \mathcal{O}(NM)$	$\mathcal{O}(N^2)$	$\mathcal{O}(TN^2)$
Iterative?	NO	NO	YES
Type	point-point	center	point-center
Need K ?	NO	YES	NO

101 has to iteratively shift each data point several times to find the peak. Table 1 summarizes
102 the differences among the PLinkage, DP and mean shift in details.

103 The remainder of this paper is organized as follows. The proposed PLinkage clustering
104 algorithm is detailedly described in Section 2. An application of the PLinkage clustering
105 algorithm on image segmentation is presented in Section 3. Experimental results of the
106 PLinkage clustering algorithm and the PLinkage based image segmentation method are
107 presented in Section 4 followed by the conclusions drawn in Section 5.

108 2. Pairwise Linkage

109 In some cases, it may be neither difficult to distinguish the cluster centers from other
110 data points based on some properties of the cluster centers like the DP method [21],
111 nor to find out the principles between the cluster centers and the data points like the
112 K -means algorithm [11] and its variants [28], but it can be easy to discover the clusters
113 in the view of the relationships between data points. The proposed PLinkage method is
114 the one which solves the clustering issue of Gaussian distributed dataset via point-point
115 pairwise linkage relationship.

116 It is a general problem for the traditional density based clustering methods, like the
117 DBSCAN [12], to distinguish two adjacent clusters with heavy overlap as Figure 2(d)
118 shows. However, the strategy of linking a data point to its closest ascending neighbouring
119 point (CANP) can solve this problem perfectly. Figure 2 demonstrates the pairwise
120 linkage chains belonging to two overlapped clusters under the conditions of light overlap,
121 heavy one and complex one. We can see that the pairwise linkage from a data point to

its CANP makes it a simple task to depart the data points of different clusters in the overlapping area despite of the variation of overlapping conditions. No specific parameters are needed to tune, all we have to do is to find the CANP of each data point and then build a pairwise linkage from this data point to its CANP. Based on this observation we propose our density ascending clustering algorithm named Pairwise Linkage (PLinkage) which can discover the Gaussian distributed clusters in the following three steps:

(1) Firstly, the density of each data point is calculated by applying a radial basis function kernel, a Gaussian kernel for example, on its neighbourhood.

(2) Then, a pairwise linkage procedure is applied to link each data point to its closest ascending neighbouring point in density. If the density of a data point is local-maximum, it is recorded as a cluster center.

(3) Finally, all the clusters can be discovered by searching along the pairwise linkages starting from each cluster center.

2.1. Density Calculation

As a density ascending clustering method, the proposed PLinkage method requires continuous values to distinguish the density peaks from its surroundings that may share the same number of neighbouring data points. The kernel density estimation is applied because it is the most popular density estimation method which can give continuous density values. Given a data point \mathbf{x} in the d -dimensional space \mathbb{R}^d , the density function of \mathbf{x} can be formulated as following:

$$f(\mathbf{x}) = \sum_{i=1}^n K(\mathbf{x} - \mathbf{x}_i), \quad (1)$$

where \mathbf{x}_i is the neighbouring point of \mathbf{x} . According to Eq. (1), we can observe that, to calculate the density of \mathbf{x} in a certain application, two issues should be solved: the definition of the neighbourhood for \mathbf{x} and the choice of the kernel K .

Neighbourhood: We use a cutoff distance d_c , as shown in Figure 3, as a global parameter to demarcate the neighbourhood of a data point \mathbf{p}_i from other data points. Only the data points whose distances to \mathbf{p}_i are smaller than d_c are considered as the

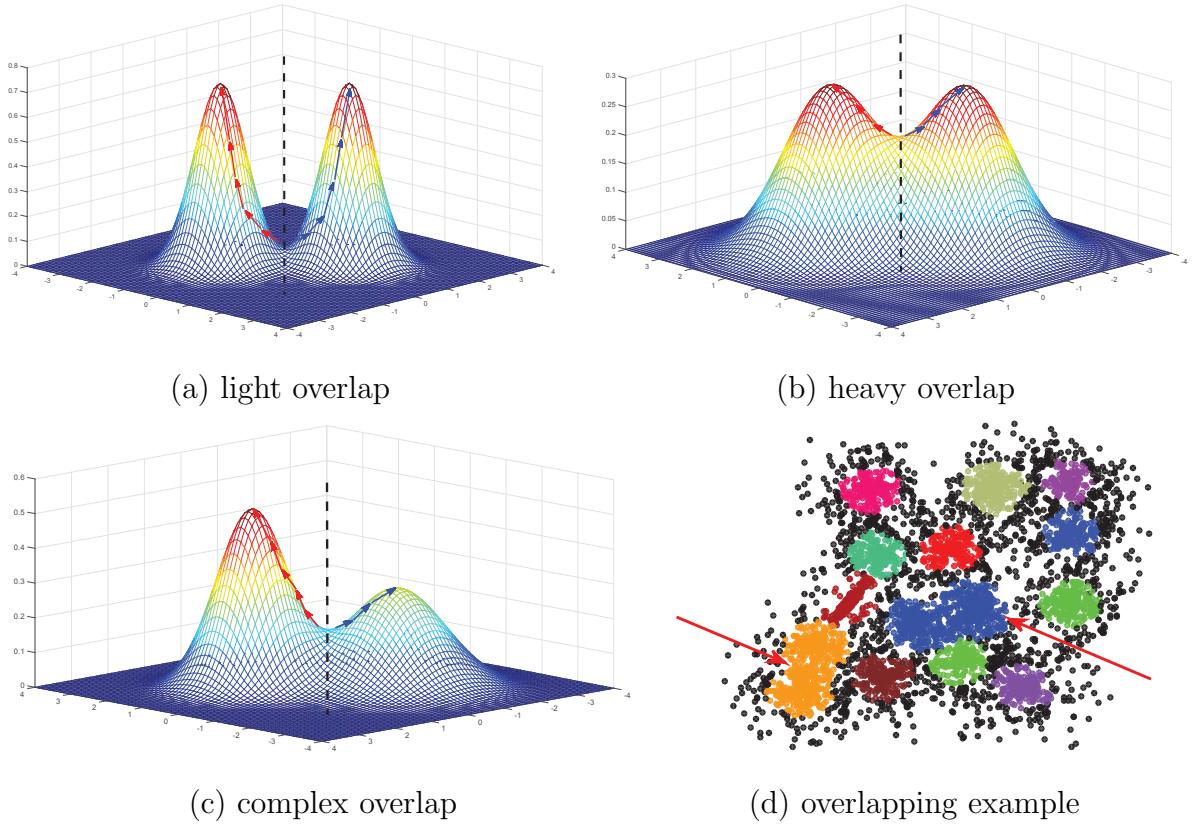


Figure 2: Demonstrations of the pairwise linkage chains of two overlapped clusters in different overlapping conditions. A clustering result of the DBSCAN method [12] on a typical heavy overlapped dataset is shown in (d).

¹⁴⁸ neighbourhood set of \mathbf{p}_i , which is denoted as \mathcal{I}_i , as the big green circle shown in Figure 3.
¹⁴⁹ The problem now is how to determine the value of d_c .

150 Setting d_c as a certain percentage of the sorted set of all the distances between any
 151 two data points, like the work of the DP method [21] does, is not an appropriate way
 152 because in this case the number of data points will count a lot on the determination of d_c .
 153 However, since d_c is an indicator of the distribution of the neighbouring points, it should
 154 be derived from the local neighbourhood instead of all the data points. Thus, in this
 155 paper, we propose a simple method to determine the value of d_c , which is described as
 156 follows: we uniformly choose 100 data points, for each data point \mathbf{p}_i of them, the distance
 157 between \mathbf{p}_i and its closest neighbour is recorded in \mathcal{D}_{cn} , and then we sort \mathcal{D}_{cn} and give

158 the definition of d_c as follows:

$$d_c = k \times \text{median}(\mathcal{D}_{cn}), \quad (2)$$

159 where k is a customized parameter which means the cutoff distance d_c is k times as big
160 as the median value of the set \mathcal{D}_{cn} . The effect of k is similar to the ‘percentage’ applied
161 in the DP method, which is suggested to be set in the interval of [2, 10] for general cases.

162 **Kernel:** It is well known that the density of a data point is correlated with the
163 distance between this point to its adjacent points. However, in practice the values of
164 distances varies from 0 to dozens of d_c , thus the radial basis function kernels like the
165 Gaussian kernel, the exponential kernel and the Laplacian kernel, which possess the non-
166 linear power, are all suitable for the density calculation. In this work, the Gaussian
167 kernel is applied because of its robustness to the noises. As to the bandwidth of the
168 Gaussian kernel, we set the bandwidth as d_c for all the data points. The reason is that d_c
169 demarcates the neighbourhood of a data point, and setting d_c as the bandwidth means
170 that the Gaussian kernel functions mainly on the neighbourhood of a data point.

171 Thus, given a data point \mathbf{p}_i , its density ρ_i can be calculated as follows:

$$\rho_i = \sum_{j=1}^N \exp\left(-\frac{\|\mathbf{p}_i - \mathbf{p}_j\|^2}{d_c^2}\right), \quad (3)$$

172 where N denotes the number of all the data points. Actually, the data points whose
173 distances to the current data point \mathbf{p}_i are larger than $3.0 \times d_c$ have little influence on
174 the value of ρ_i , thus the density can also be calculated via the neighbouring points $\{\mathbf{p}_j\}$
175 satisfying $d_{ij} < 3.0 \times d_c$ where d_{ij} denotes the distance between \mathbf{p}_i and \mathbf{p}_j , i.e., $d_{ij} = \|\mathbf{p}_i - \mathbf{p}_j\|$,
176 which can be achieved by building grids for all the data points resulting
177 in reducing the computational complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(NM)$, in which M is the
178 number of data points in $\{\mathbf{p}_j\}$.

179 *2.2. Pairwise Linkage*

180 With the densities of all the data points, the pairwise linkages can be built in a non-
181 iterative way, which is performed as follows. For a data point \mathbf{p}_i whose neighbourhood
182 set is \mathcal{I}_i , we traverse each point in \mathcal{I}_i to find the closest data point \mathbf{p}_j whose density is

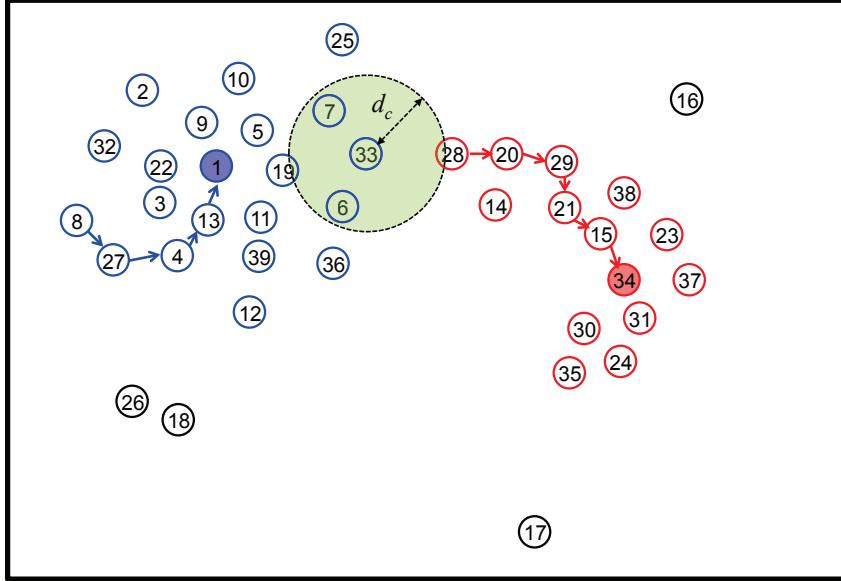


Figure 3: An illustration of the density ascending clustering procedure. The data points \mathbf{p}_1 and \mathbf{p}_{34} are the cluster centers, the symbol “ \rightarrow ” indicates the pairwise linkage, and the big circle in green denotes the neighbourhood set of a data point \mathbf{p}_{33} with a cutoff distance d_c .

greater than that of \mathbf{p}_i . The distance between two data point \mathbf{p}_i and \mathbf{p}_j is calculated as $d_{ij} = \|\mathbf{p}_i - \mathbf{p}_j\|$. If the CANP of \mathbf{p}_i is found, a pairwise linkage from \mathbf{p}_i to its CANP is built and recorded in the linkage list \mathcal{L}_l . Otherwise, if the density of \mathbf{p}_i is local-maximal, which means that there exists no data point in \mathcal{I}_i whose density is greater than that of \mathbf{p}_i , we consider \mathbf{p}_i as a cluster center and record it in the center list \mathcal{L}_c . The result of the pairwise linkage procedure is comprised of a linkage list \mathcal{L}_l recording all the pairwise linkages and a center list \mathcal{L}_c recording all the cluster centers.

2.3. Clustering

The proposed PLinkage performs top-down clustering, which is similar to the divisive clustering algorithm. For each cluster center \mathbf{c}_k in \mathcal{L}_c , we start searching the linkage list \mathcal{L}_l from \mathbf{c}_k in a depth-first or breadth-first way to gather all the data points that are directly or indirectly connected with \mathbf{c}_k , which generates a cluster whose center is \mathbf{c}_k . The whole clustering procedure finds out all the clusters, the set of which are denoted as \mathcal{C} .

Figure 3 shows an illustration of the clustering procedure. From Figure 3, we can

198 observe that \mathbf{p}_1 is the cluster center due to its local maximal density and there are four
 199 pairwise linkages between $(\mathbf{p}_1, \mathbf{p}_{13})$, $(\mathbf{p}_{13}, \mathbf{p}_4)$, $(\mathbf{p}_4, \mathbf{p}_{27})$, and $(\mathbf{p}_{27}, \mathbf{p}_8)$. Thus the clustering,
 200 starting from the cluster center \mathbf{p}_1 , is performed as $\mathbf{p}_1 \rightarrow \mathbf{p}_{13} \rightarrow \mathbf{p}_4 \rightarrow \mathbf{p}_{27} \rightarrow \mathbf{p}_8$. By this
 201 way, the clustering information is propagated from the dense data points to the sparse
 202 ones step by step.

203 *2.4. Adapting to Smoothly Distributed Dataset*

204 When the dataset is in the Gaussian distribution, as shown in Figure 1, the density
 205 ascending clustering via pairwise linkages can recover the clusters quite well, but may
 206 fail into fragmented clustering results when there exist multiple local maximums in the
 207 smoothly distributed dataset. If the dataset is in the smooth distribution, an additional
 208 cluster merging procedure is proposed with the following three steps. Firstly, for each
 209 cluster \mathcal{C}_p , the average density μ_p and the standard deviation σ_p of densities of all the
 210 data points in \mathcal{C}_p are calculated. Then, we search for the adjacent clusters for each one.
 211 Two clusters \mathcal{C}_p and \mathcal{C}_q are considered adjacently if the following condition is satisfied:

$$\begin{aligned} & \exists \quad \mathbf{p}_i \in \mathcal{C}_p \quad \text{and} \quad \mathbf{p}_j \in \mathcal{C}_q, \\ & \text{where } \mathbf{p}_i \in \mathcal{I}_j \quad \text{and} \quad \mathbf{p}_j \in \mathcal{I}_i, \end{aligned} \tag{4}$$

212 where \mathcal{I} denotes the neighbourhood of a data point. Finally, for each adjacent cluster pair
 213 \mathcal{C}_p and \mathcal{C}_q , the average densities of the adjacent points of \mathcal{C}_p and \mathcal{C}_q are denoted as $\bar{\rho}_p$ and
 214 $\bar{\rho}_q$, respectively. These two adjacent clusters will be merged if the following conditions
 215 are satisfied:

$$\bar{\rho}_p > \mu_q - 3 \times \sigma_q \quad \text{and} \quad \bar{\rho}_q > \mu_p - 3 \times \sigma_p. \tag{5}$$

216 The cluster merging is conducted iteratively, which means that all the clusters that are
 217 directly or indirectly adjacent to the starting cluster will be merged if the conditions in
 218 Eq. (5) are satisfied.

219 *2.5. Noises and Complexity*

220 **Noises:** In our work, we consider the noise on the cluster-level instead of the point-
 221 level like the DBSCAN method [12]. If a data point \mathbf{p}_i whose density is local-maximal

222 but smaller than the median density, $\text{median}(\rho)$, of all the data points, all the data points
223 in the same cluster with \mathbf{p}_i are considered as noises.

224 **Complexity:** The computational complexity of the procedure to calculate d_c is
225 $\mathcal{O}(NM)$, which depends on the sample points we choose. For density calculation it
226 is $\mathcal{O}(N^2)$, and if the k -d tree or the grid is built, the complexity can be greatly reduced to
227 $\mathcal{O}(NM)$ where M is the number of neighbouring points for each data point. For linkage
228 building and clustering, the computational complexities are $\mathcal{O}(NM)$ and $\mathcal{O}(N)$, respec-
229 tively. So the computational complexity of the PLinkage is $\mathcal{O}(N^2)$ without optimization,
230 which is faster than that of the mean shift method whose computational complexity
231 is $\mathcal{O}(TN^2)$ where T is the number of iterations of the algorithm and the medoid shift
232 method which is $\mathcal{O}(N^{2.38})$ [25]. As we can see, most of the computation of PLinkage is
233 on neighbouring data points searching and the density calculation, and fortunately it is
234 a parallel procedure and can be further accelerated by building grids for the data points
235 or other efficient search strategies, e.g., k -d tree. What should be noticed is that, the
236 proposed PLinkage suits best for the well structured dataset like the image, in Section 3
237 we will show the application of the PLinkage method on the issue of image segmentation.

238 Figure 3 shows an illustration of some basic ideas of the proposed PLinkage clustering
239 method, from which we can observe that there are two clusters in blue and red, respec-
240 tively. For each data point, the pairwise linkage is formed by searching for the closest
241 neighbouring point whose density is greater than its own. For example, \mathbf{p}_8 is first linked
242 to \mathbf{p}_{27} , \mathbf{p}_{27} is then linked to \mathbf{p}_4 , \mathbf{p}_4 is further linked to \mathbf{p}_{13} , and \mathbf{p}_{13} is finally linked to
243 \mathbf{p}_1 with the greatest density in its neighbourhood. In this way, the complete linkage is
244 found as $\mathbf{p}_8 \rightarrow \mathbf{p}_{27} \rightarrow \mathbf{p}_4 \rightarrow \mathbf{p}_{13} \rightarrow \mathbf{p}_1$, and thus all of these five data points are classified
245 into the same cluster whose cluster center is \mathbf{p}_1 . The same procedure occurs for all the
246 data points in blue as a separated cluster. Similarly the red cluster can be formed by this
247 way. As to the data points on the boundary between two clusters, the pairwise linkage
248 can still be applied. Taking the data point \mathbf{p}_{33} for example, \mathbf{p}_{19} , \mathbf{p}_7 , \mathbf{p}_6 , and \mathbf{p}_{28} are its
249 neighbouring points, \mathbf{p}_6 is its closest ascending neighbouring point (CANP), and thus
250 \mathbf{p}_{33} is classified into the blue cluster, which is quite reasonable. The four data points

251 in black, \mathbf{p}_{26} , \mathbf{p}_{18} , \mathbf{p}_{17} , and \mathbf{p}_{16} , are classified as outliers because there exists no CANP
252 in their neighbourhoods, nor the densities of their corresponding cluster centers are not
253 high enough.

254 As a summary, the proposed clustering method can discover the clusters and cluster
255 centers in a quite simple and efficient way. For each data point \mathbf{p}_i with a density ρ_i , we
256 find its closest neighbouring point $\text{CANP}(\mathbf{p}_i)$ whose density is greater than that of \mathbf{p}_i , and
257 classify the point \mathbf{p}_i to the same cluster with $\text{CANP}(\mathbf{p}_i)$. If the density ρ_i of the data point
258 \mathbf{p}_i is local-maximal and greater than the average density $\bar{\rho}$, we consider \mathbf{p}_i as a cluster
259 center. Then all the clusters can be discovered by searching along the pairwise linkages
260 starting from each cluster center. Algorithm 1 summaries the complete procedure in
261 details of the proposed PLinkage clustering method.

262 3. Application on Image Segmentation

263 The proposed PLinkage is a non-iterative algorithm and its computational complexity
264 is low, just $\mathcal{O}(NM)$ on the application when the data points are well structured, which
265 means it can be applied on some specific applications efficiently, e.g., image segmentation.
266 To validate the efficiency and robustness of our proposed PLinkage clustering algorithm
267 in some practical applications, we apply it on image segmentation as its application. The
268 related background and related works of image segmentation is introduced in Section 3.1.
269 The details how to apply the PLinkage clustering on image segmentation are presented
270 in Section 3.2.

271 3.1. Introduction to Image Segmentation

272 Image segmentation is a field which applies the clustering analysis a lot. Achanta et
273 al. [8] proposed to use the famous K -means method to solve the problem of superpixels
274 segmentation and excellent results were achieved, and the mean shift method [23] also
275 works well on image segmentation. Image segmentation aims at partitioning an image
276 into meaningful regions of coherent properties as a means for some specific objectives,
277 e.g., separating objects from their backgrounds. All pixels in a region are similar with
278 respect to some characteristic or computed property, such as color, intensity, or texture.

Algorithm 1 Pairwise Linkage

Input: The density of each data point and the cutoff distance d_c .

Output: The clusters \mathcal{C} and the cluster centers list \mathcal{L}_c .

```
1:  $\rho_i$  : the density of a data point  $\mathbf{p}_i$ 
2:  $\bar{\rho}$  : the average density of all the data points
3:  $\mathcal{I}_i$  : the neighbourhood set of a data point  $\mathbf{p}_i$ 
4:  $\mathcal{L}_l$  : the linkage list recording all the pairwise linkages
5: for each data point  $\mathbf{p}_i$  do
6:   Set  $LocalMaximum(\mathbf{p}_i) \leftarrow \text{TRUE}$ 
7:   Set  $d_{\min} \leftarrow \infty$  and  $CANP(\mathbf{p}_i) \leftarrow \emptyset$ 
8:   for each neighbouring point  $\mathbf{p}_j$  in  $\mathcal{I}_i$  do
9:     Set  $d_{ij} \leftarrow$  the distance between  $\mathbf{p}_i$  and  $\mathbf{p}_j$ 
10:    if  $\rho_j > \rho_i$  and  $d_{ij} < d_{\min}$  then
11:      Set  $LocalMaximum(\mathbf{p}_i) \leftarrow \text{FALSE}$ 
12:      Set  $CANP(\mathbf{p}_i) \leftarrow \mathbf{p}_j$  and  $d_{\min} \leftarrow d_{ij}$ 
13:    end if
14:   end for
15:   if NOT  $LocalMaximum(\mathbf{p}_i)$  then
16:     Record the linkage between  $\mathbf{p}_i$  and  $CANP(\mathbf{p}_i)$  into  $\mathcal{L}_l$ 
17:   else if  $LocalMaximum(\mathbf{p}_i)$  and  $\rho_i > \bar{\rho}$  then
18:     Insert the data point  $\mathbf{p}_i$  into  $\mathcal{L}_c$ 
19:   end if
20: end for
21: Collect the clusters  $\mathcal{C}$  by searching data points in the linkage list  $\mathcal{L}_c$  from each data
point in the center list  $\mathcal{L}_c$ 
```

²⁷⁹ Image segmentation has been approached from a wide variety of perspectives. Several
²⁸⁰ general-purpose algorithms have been developed for image segmentation, including his-
²⁸¹ togram thresholding, edge (boundary) based segmentation, clustering and region growing
²⁸² approaches.

283 The histogram thresholding is the simplest method of image segmentation [29], the
 284 key of which is to select the threshold value (or values when multiple levels are used)
 285 to satisfy some segmentation principles, for example the maximum entropy in the Otsu's
 286 method [30]. The edge (boundary) based segmentation methods try to detect the edges or
 287 boundaries of objects and then segment the image based on them, however it is not easy
 288 to detect object boundaries in the natural images [31]. Recently with the development of
 289 machine learning, a lot of boundary detection methods [32, 33] have been proposed and
 290 some good results of image segmentation have been achieved [34]. The algorithm proposed
 291 in the work of [35] incorporates the correntropy-based K -means method with a region-
 292 based level set segmentation framework to segment images. The common procedure of
 293 the region growing methods [36] is to compare one pixel with its neighbours, if a similarity
 294 criterion is satisfied, these pixels can be set to belong to the same cluster. The mean shift
 295 method [23] can be also applied on the image segmentation quite well, which filters the
 296 image via mean shift clustering and then segments it via region growing.

297 3.2. PLinkage on Image Segmentation

298 Let \mathbf{I}_{RGB} be an input RGB image, \mathbf{I}_{LAB} be the LAB image converted from the original
 299 image \mathbf{I}_{RGB} , \mathbf{p}_i be a 2D pixel in \mathbf{I}_{LAB} , $\mathbf{I}_{\text{LAB}}(\mathbf{p}_i)$ be the LAB color vector of the pixel \mathbf{p}_i in
 300 \mathbf{I}_{LAB} , and \mathcal{I}_i be the set of neighbouring pixels of \mathbf{p}_i which is comprised of the set of pixels
 301 in a window centered at \mathbf{p}_i with a size of $(2h_s + 1) \times (2h_s + 1)$. Then the PLinkage-based
 302 image segmentation can be conducted in the following three steps:

303 (1) **Density Calculating:** Given a pixel $\mathbf{p}_i \in \mathbf{I}_{\text{LAB}}$, its density is defined as:

$$\rho_i = \sum_{\mathbf{p}_j \in \mathcal{I}_i} k \left(\left\| \frac{d^s(\mathbf{p}_i, \mathbf{p}_j)}{h_s} \right\|^2 \right) k \left(\left\| \frac{d^r(\mathbf{p}_i, \mathbf{p}_j)}{h_r} \right\|^2 \right), \quad (6)$$

304 where $d^s(\mathbf{p}_i, \mathbf{p}_j)$ and $d^r(\mathbf{p}_i, \mathbf{p}_j)$ stand for the Euclidean distances between \mathbf{p}_i and \mathbf{p}_j in
 305 the image space and in the LAB color space, respectively, i.e., $d^s(\mathbf{p}_i, \mathbf{p}_j) = \|\mathbf{p}_i - \mathbf{p}_j\|$ and
 306 $d^r(\mathbf{p}_i, \mathbf{p}_j) = \|\mathbf{I}_{\text{LAB}}(\mathbf{p}_i) - \mathbf{I}_{\text{LAB}}(\mathbf{p}_j)\|$, and h_s and h_r are the bandwidths of the Gaussian
 307 kernels in the image space and in the LAB color space, respectively. The computational
 308 complexity of the PLinkage based image segmentation algorithm is $\mathcal{O}(NM)$ on the struc-
 309 tured 2D image, where $\mathcal{O}(M)$ denotes the complexity of density computation for each

pixel. In the practical application, applying the Gaussian kernel on the whole image is time consuming, which may lead to a great value of M . The methods to reduce the time consuming of kernel calculation are widely researched in the field of bilateral filtering [37, 38], which are very similar to our problem because they both need to calculate two kernels. However, in Eq. (6) we don't need to calculate the normalization term which is a key issue on reducing the computational complexity of bilateral filtering. For the first kernel, we have $d^s(\mathbf{p}_i, \mathbf{p}_j) \leq h_s$, for each integral value $\in [0, h_s]$, its corresponding kernel value can be pre-calculated and recorded in the list \mathcal{L}_s . Thus for each value of $d^s(\mathbf{p}_i, \mathbf{p}_j)$, its corresponding value on the first kernel can be simply set as the closest one in \mathcal{L}_s . For the second kernel, the same method can also be applied. To achieve this goal, one more restriction should be added on Eq. (6): $\mathbf{p}_j \in \mathcal{I}_i \quad \& \quad d^r(\mathbf{p}_i, \mathbf{p}_j) \leq h_r$, which means that only the neighbouring pixels whose LAB values are close enough to that of the current point \mathbf{p}_i can contribute to its density ρ_i . This restriction may be not proper in bilateral filtering which needs to preserve the edges, but it is quite reasonable in the density calculation which focuses on the similarity between each pixel and its neighbourhood. Thus, for each integral value in $[0, h_r^2]$, the value of the second kernel is calculated and recorded in the list \mathcal{L}_r . Then on the density calculation for each pixel, the value of the second kernel for each neighbour \mathbf{p}_j is obtained just by searching the list \mathcal{L}_r . In this way, the computational complexity of Eq. (6) can be greatly reduced, which is much more practical.

(2) ***Linkage Building***: Then for each pixel \mathbf{p}_i , the PLinkage clustering is applied to link it to the closest neighbouring point who has a higher density than that of \mathbf{p}_i . The distance between \mathbf{p}_i and its neighbour $\mathbf{p}_j \in \mathcal{I}_i$ is calculated as follows:

$$d(\mathbf{p}_i, \mathbf{p}_j) = d^r(\mathbf{p}_i, \mathbf{p}_j) + \alpha \times \frac{h_r \times d^s(\mathbf{p}_i, \mathbf{p}_j)}{h_s}, \quad (7)$$

where $d^r(\mathbf{p}_i, \mathbf{p}_j)$ and $d^s(\mathbf{p}_i, \mathbf{p}_j)$ have the same meanings as in Eq. (6), and $h_r \times d^s(\mathbf{p}_i, \mathbf{p}_j)/h_s$ is applied to convert the $d^s(\mathbf{p}_i, \mathbf{p}_j)$ into the LAB color space. The parameter α is applied to balance the effects of $d^r(\mathbf{p}_i, \mathbf{p}_j)$ and $d^s(\mathbf{p}_i, \mathbf{p}_j)$ on the value of $d(\mathbf{p}_i, \mathbf{p}_j)$. For example, $\alpha = 0.1$ means the weight of $d^r(\mathbf{p}_i, \mathbf{p}_j)$ is 10 times as big as that of $d^s(\mathbf{p}_i, \mathbf{p}_j)$, in this condition, $d^r(\mathbf{p}_i, \mathbf{p}_j)$ has more influence on $d(\mathbf{p}_i, \mathbf{p}_j)$. When all the neighbours of \mathbf{p}_i share

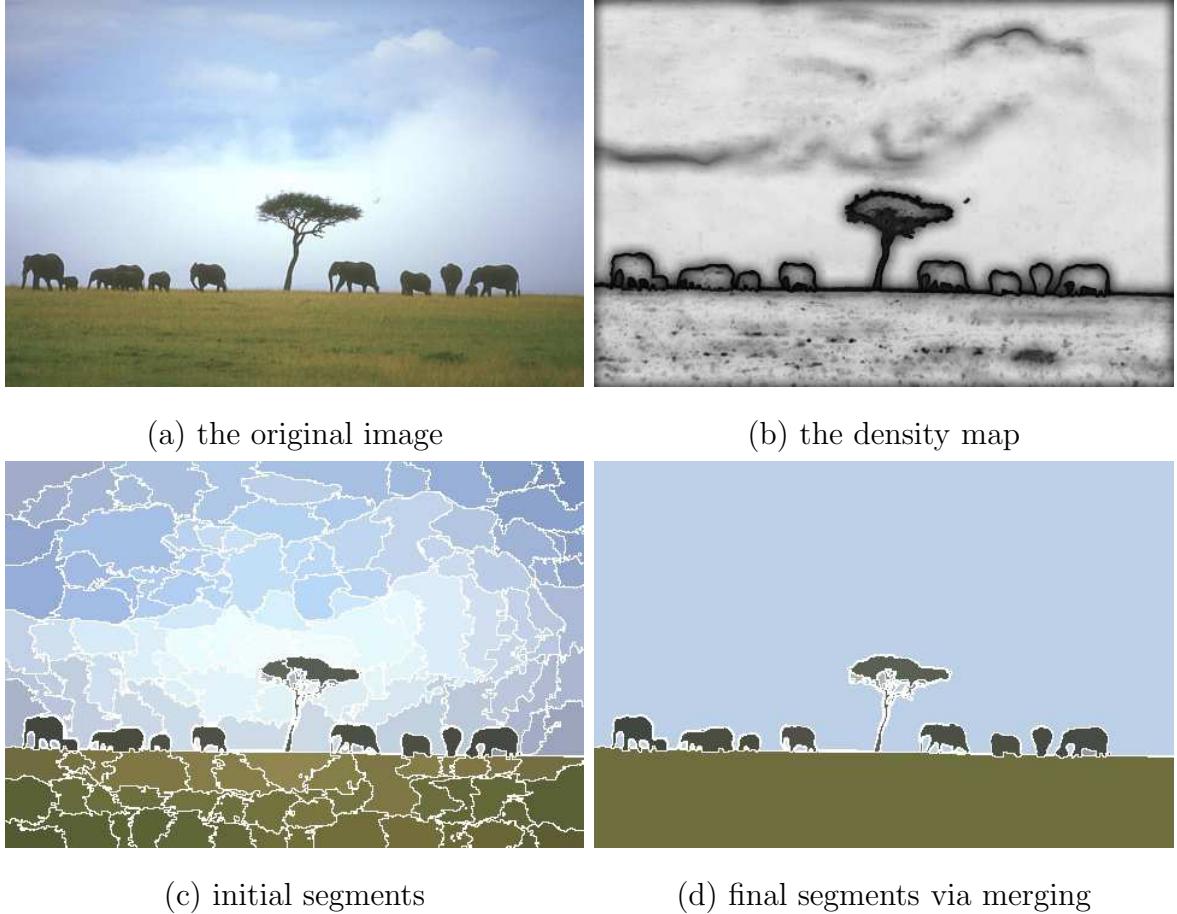


Figure 4: An example of applying the PLinkage based image segmentation method on an outdoor image.

338 very close LAB values with its own, the first item of Eq. (7) is very small, in this way
 339 the second item plays the crucial role, so the neighbour who is close to \mathbf{p}_i on the image
 340 is chosen as the $\text{CANP}(\mathbf{p}_i)$. While the neighbours differ much to \mathbf{p}_i in the LAB color
 341 space, the value of the first item is much larger than that of the second one, thus the
 342 neighbour whose LAB value is close to that of \mathbf{p}_i in the LAB color space is picked out as
 343 the $\text{CANP}(\mathbf{p}_i)$. A pixel \mathbf{p}_i is regarded as a cluster center if one of the following conditions
 344 are satisfied:

$$\begin{cases} d^r(\mathbf{p}_i, \text{CANP}(\mathbf{p}_i)) > h_r, \\ \text{LocalMaximum}(\mathbf{p}_i) = \text{TRUE}. \end{cases} \quad (8)$$

345 Initial segments can then be achieved by searching along the linkages starting from each
 346 cluster center.

347 (3) ***Segments Merging***: For each initial segment, its adjacent ones are searched and

348 merged iteratively if the distance between the average LAB values of two neighbouring
349 segments is smaller than h_r .

350 Figure 4 shows an example of applying the PLinkage based image segmentation
351 method with $(h_s, h_r, \alpha) = (10, 6.5, 0.5)$, from which we can observe that the initial seg-
352 ments can separate different objects well on the boundary, even though our method may
353 segment a whole object with lowly varied densities into multiple fragments, which can be
354 made up by the merging procedure.

355 4. Experimental Results

356 In this section, we first tested our proposed PLinkage clustering algorithm on various
357 public data sets with different dimensional synthetic data and the Caltech-256 object
358 category dataset [1] with CNN (Convolution Neural Network) based features to compare
359 it with four other clustering algorithms, and then we tested the proposed PLinkage based
360 image segmentation method on the BSDS dataset [2] by comparing with the mean shift
361 based image segmentation method. All our proposed algorithms in this paper were im-
362 plemented in C++, whose source codes and more experimental results are available at
363 <http://cvrs.whu.edu.cn/projects/PLinkage/>.

364 4.1. Evaluation on PLinkage Clustering

365 The only customized parameter needed for the proposed PLinkage clustering algo-
366 rithm is the k in Eq. (2) used to determine the value of d_c . We recommend $k \in [2, 10]$,
367 and set $k = 5$ for general cases, which will be proved in the following experiments to
368 be a good choice. Four other clustering methods were also tested to be compared with
369 the proposed PLinkage clustering algorithm, including: the DP (density peaks) [21], the
370 DBSCAN [12], the K -means [11] and the mean shift [23]. For these clustering methods,
371 we set their parameters as follows:

372 (1) We set the cutoff distance of the DP method as d_c . As to the problem of choosing
373 proper number of density peaks in the DP method, we sorted $v_i = \rho_i \times \epsilon_i$ in descending
374 order, and chose the front K data points as the density peaks where K was predefined
375 like K -means.

376 (2) We set $\text{eps} = d_c$ for the DBSCAN method because both of eps and d_c define
377 the neighbourhood of a data point, and MinPts was set as the median value of the
378 neighbouring size of 500 data point sampled uniformly.

379 (3) We applied the Gaussian kernel in mean shift and set the kernel bandwidth as d_c ,
380 the threshold ϵ used to stop the shifting was set as 1% of the kernel bandwidth.

381 **Measurements:** Let L_{GT} be the provided ground truth of the label of each data
382 point and L_{cluster} be that of the clustering result after applying a clustering method.
383 Two accuracy measurements, the correct cluster number (CCN) and the correct data
384 point number (CDN), of the clustering result can be achieved as follows: for each ground
385 truth cluster \mathcal{C}_{GT} with the label L_{GT} , we search in the data points whose labels are
386 L_{cluster} to find out the clusters $\mathcal{C}_{\text{cluster}}$ which have N overlapping data points with \mathcal{C}_{GT} ,
387 if $N > \text{size}(\mathcal{C}_{\text{GT}})/2$ and $N > \text{size}(\mathcal{C}_{\text{cluster}})/2$ where $\text{size}(\bullet)$ denotes the size of a set, we
388 consider that $\mathcal{C}_{\text{cluster}}$ is a correct cluster corresponding to \mathcal{C}_{GT} , thus add 1 to CCN and
389 add N to CDN. CDN is converted into ratio at last.

390 **2D Gaussian Distributed Dataset:** The S -sets [39]¹, which is comprised of 4
391 synthetic 2-D subsets with $N = 5000$ vectors and $M = 15$ Gaussian clusters with different
392 degrees of cluster overlapping, was applied in our work. The ground truth of the label of
393 each data point is also provided. In all the 4 subsets, we set $k = 8.0$ and $K = 15$, and the
394 clusters with sizes smaller than 10 were discarded as noises. Table 2 shows the clustering
395 results of the PLinkage, DP, DBSCAN, K -means and mean shift on the S -sets, in which
396 the symbol t denotes the total number of clusters after applying a clustering algorithm,
397 and c means the correct cluster number (CCN), and p is the ratio of the correct data point
398 (CDN) over the number of all data points in the tested set, i.e., classification accuracy.

399 From Table 2, we can observe that generally the CDN ratios (i.e., classification ac-
400 curacies) of the proposed PLinkage are little lower than those of the DP and the mean
401 shift due to that the PLinkage discovered more outliers than the other two methods, but
402 these noises were all discovered correctly as the black data points in the rectangles shown
403 in Figure 5(a). But the PLinkage can get less over-segmented clusters than the mean

¹Available at <http://cs.uef.fi/sipu/datasets/>

Table 2: Comparison between PLinkage, DP, DBSCAN, K -means and mean shift on the S -sets with 15 clusters.

Algorithms	PLinkage			DP			DBSACAN			K -means			mean shift		
Measurements	t	c	p	t	c	p	t	c	p	t	c	p	t	c	p
S_1	15	15	0.9680	15	15	0.9948	15	15	0.7350	15	15	0.9934	16	15	0.9894
S_2	15	15	0.9456	15	13	0.9682	15	15	0.7162	15	15	0.9698	16	15	0.9630
S_3	16	15	0.8346	15	3	0.8522	13	9	0.3912	15	15	0.8550	18	15	0.8472
S_4	15	15	0.7836	15	4	0.7984	12	8	0.3770	15	15	0.7966	18	15	0.7856

shift method, for example, in S_1 , S_3 and S_4 the total numbers of clusters obtained by the PLinkage are exactly the same as ground truth, while those of clusters obtained by the mean shift method are 16, 16, and 18, respectively. The DP method performed well on all the four subsets, however these good performance is based on the key problem, the number of density peaks, was predefined as $K = 15$. The DBSCAN method can distinguish all the 15 clusters well in S_1 and S_2 , but mixed up with the increasing of overlapping between clusters, as shown in Figure 5(c), which illustrates the shortage of the region growing strategy of DBSCAN. The K -means method performed quite well on all the 4 subsets mainly for the reason that the distances between clusters are generally the same despite the variation of overlapping. As a summary, we can see that the proposed PLinkage method performed as well as the DP, K -means and mean shift methods, however the PLinkage does not need the predefined number K of clusters, and is not as iterative as the mean shift.

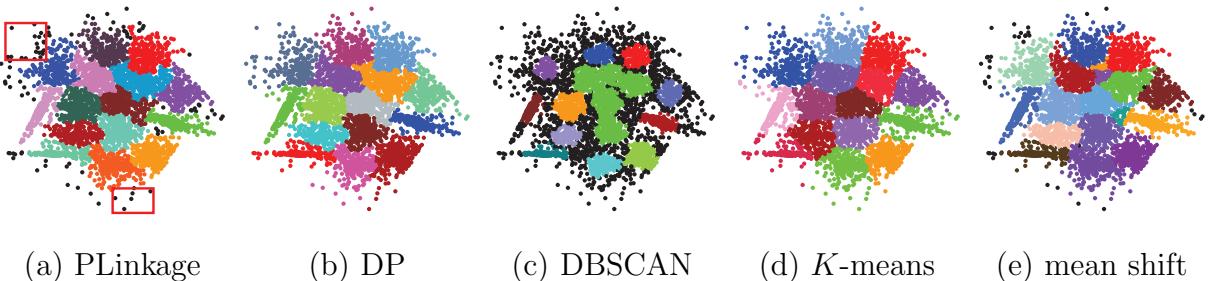


Figure 5: The clustering results on the subset S_4 in the S -sets by five different clustering methods. Different clusters are in different colors, and the noises are marked in black.

417 **2D Smoothly Distributed Dataset:** The Chameleon-sets [40] which contains 4
 418 synthetic 2-D subsets in the smooth distribution ², was applied to test the performances
 419 of these five methods. Since that there is no ground truth attached, we only evaluate the
 420 performances by vision. In all the 4 subsets, we set $k = 5.0$ and the predefined numbers
 421 of clusters are 6, 6, 9 and 8, respectively. The clusters with sizes smaller than 10 were
 422 discarded as noises. Figure 6 shows the clustering results of these five methods on two
 423 subsets of the Chameleon-sets. We can see that the adapting of the PLinkage method
 424 from the Gaussian distribution to the smooth distribution performed quite well on the
 425 Chameleon-sets, which is much better than the resting four methods. The DP method
 426 tends to mis-classify the data points into noises when the densities of the border points
 427 are close to these of the central points of the cluster due to its improper noise strategy, as
 428 can be seen in the second image on the second row in Figure 6 . The DBSCAN method
 429 is also inclined to mix the clusters when there exist large overlapping between clusters.
 430 The K -means method is improper for these dataset with arbitrary cluster shapes. The
 431 mean shift method is suitable for the Gaussian distributed dataset, but in these smoothly
 432 distributed ones it is prone to get more fragmented clustering results.

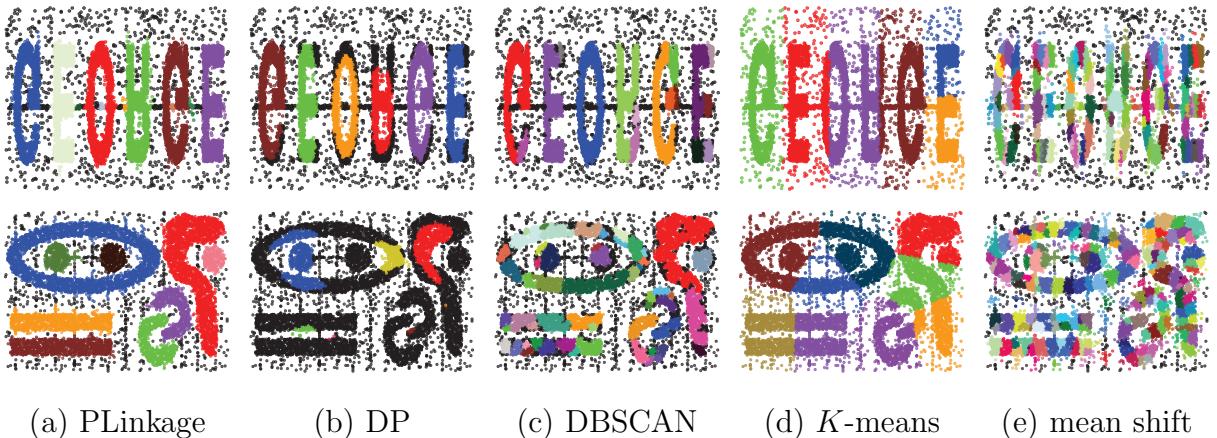


Figure 6: The clustering results on two subsets in the Chameleon-sets by five different clustering methods. Different clusters are in different colors, and the noises are marked in black.

433 **2D Arbitrarily Distributed Dataset:** We chose five of eight subsets of the Shape-

²Available at <http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download>

434 sets³ to test the performance of these five methods, because the data point of them are in
 435 the approximate Gaussian distribution. These five subsets include the Aggregation [41],
 436 the Spiral [42], the D31 [43], the R15 [43] and the Flame [44]. In this experiment, we
 437 set $k = 5.0$ for all these subsets. Table 3 shows the clustering results of the PLinkage,
 438 DP, DBSCAN, K -means and mean shift on the Shape-sets, in which the symbol ‘ M ’
 439 stands for the ground truth number of clusters. From Table 3, we can see that in these
 440 five subsets, the proposed PLinkage method can discover exactly the number of ground
 441 truth clusters, especially on the Aggregation, the Flame and the Spiral cases, as Figure 7
 442 shows.

Table 3: Comparison between PLinkage, DP, DBSCAN, K -means and mean shift on the Shape-sets.

Algorithms	PLinkage			DP			DBSACAN			K -means			mean shift		
Measurements	t	c	p	t	c	p	t	c	p	t	c	p	t	c	p
Aggregation $M = 7$	7	7	1.0000	7	7	1.0000	7	7	0.8896	7	4	0.5609	7	7	0.9987
Spiral $M = 3$	3	3	1.0000	3	3	1.0000	3	0	0.0000	3	0	0.0000	3	0	0.0000
D31 $M = 31$	31	31	0.9548	31	1	0.9684	31	31	0.8135	31	30	0.9313	31	31	0.9735
R15 $M = 15$	15	15	0.9783	15	15	0.9967	15	15	0.8733	15	15	0.9967	15	15	0.9967
Flame $M = 2$	2	2	0.9792	2	2	0.9792	1	1	0.4875	2	2	0.8458	2	2	0.9833

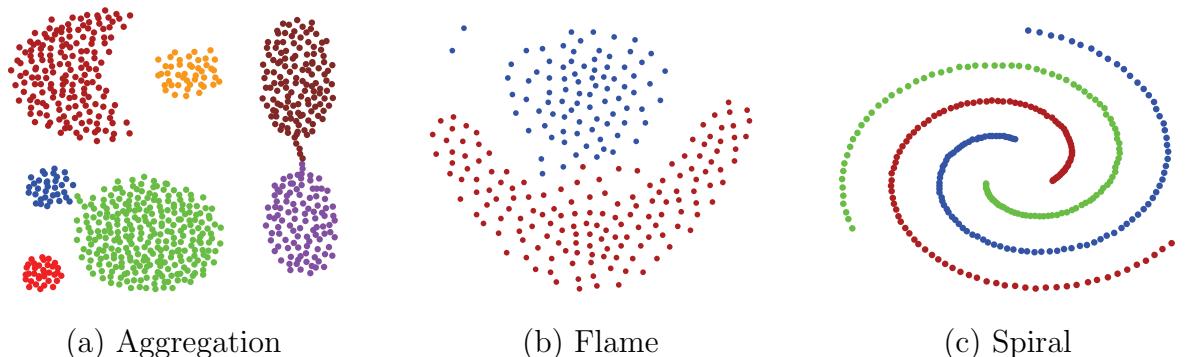


Figure 7: Clustering results of the proposed PLinkage on the Aggregation, Flame and Spiral subsets in the Shape-sets.

443 **3D Gaussian Distributed Dataset:** We used a simulated 3D dataset to test the

³Available at <http://cs.uef.fi/sipu/datasets/>

robustness of the proposed PLinkage. The 3D dataset is composed of three subsets in the Gaussian distribution, each contains 200 data points, as Figure 8(a) shows. In this experiment, we also set $k = 5.0$. Figure 8(b) shows the clustering result, and the numbers of correct classified data points for three subsets are 175 (87.5%), 195 (97.5%) and 191 (95.5%), respectively. In all the three 2D views in Figure 8, we can see that all the three subsets were divided well.

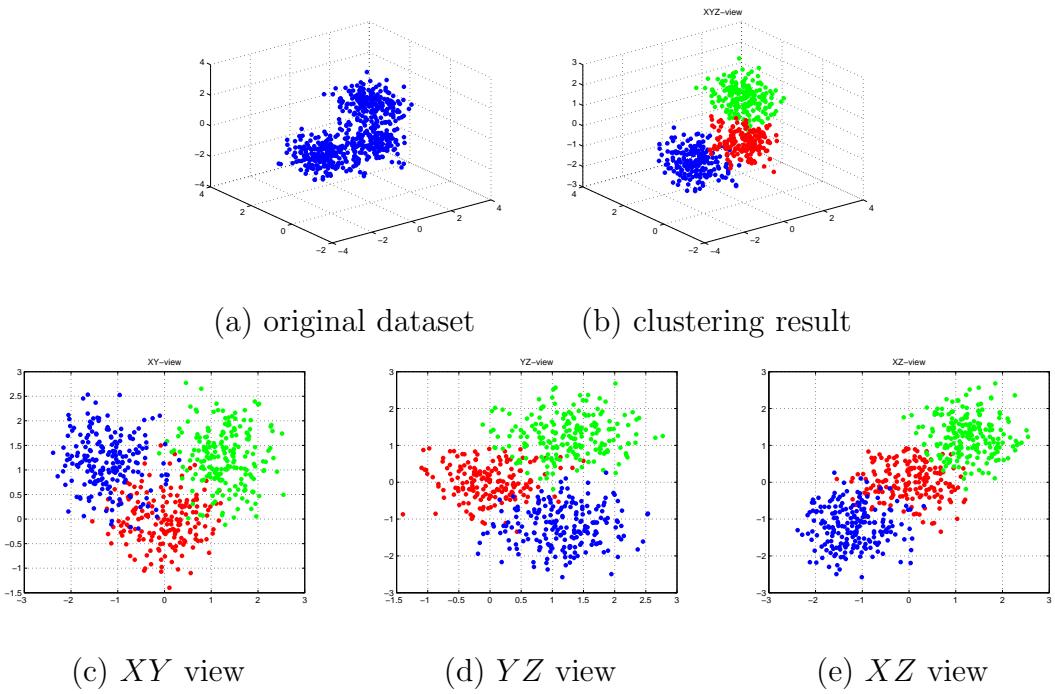


Figure 8: Clustering results of PLinkage on the 3D (XYZ) simulated test data: the original data points and the clustering result projected onto the XY plane, the YZ plane, and the XZ plane from left to right.

ND Gaussian Distributed Dataset: To test the ability of the proposed PLinkage on handling the high-dimensional data sets, we applied it on the DIM-sets [45] which contains 6 high-dimensional data sets with $N = 1024$ vectors and $M = 16$ clusters in the Gaussian distribution⁴. The dimensions of these six subsets are 32, 64, 128, 256, 512 and 1024, respectively. In this experiment, we also set $k = 5.0$ for all the cases. Table 4 shows the clustering results, from which we can see that the proposed PLinkage

⁴Available at <http://cs.uef.fi/sipu/datasets/>

Table 4: Clustering results of the proposed PLinkage method on the DIM-sets.

Subsets	32D	64D	128D	256D	512D	1024D
Measurements	t c p					
PLinkage	16 16 0.9980	16 16 0.9951	16 16 0.9824	16 16 1.0000	16 16 1.0000	16 16 1.0000

method can discover all the 16 clusters quite well on all the 6 subsets, which means the PLinkage can be applied for clustering high dimensional data sets, especially in the Gaussian distribution.

Caltech-256 CNN Feature Dataset: We also tested the PLinkage on the a subset of the Caltech-256 dataset [1] ⁵ to see the ability of the PLinkage on high level issues such as image classification. Five categories, each contains around 100 sample images, were chosen to form a subset. Then we applied the MatConvNet [46] ⁶ as the CNN framework and used the pre-trained model of the VGG [47] on the ImageNet [48] to get the 4096-d feature vector of each image. Then the PCA method was performed on each 4096-d vector to get it down to 128-d, because it would save some space and (ideally) not hurt accuracy. In this way, for each image in the subset of Caltech-256, a 128-d feature vector was obtained. Then we input these feature vectors into PLinkage, and a satisfactory clustering result can be obtained by applying the following two adjustments: (1) the distance measurement of two vectors v_1 and v_2 was now set as $1 - \cos(v_1 \cdot v_2^\top)$, and (2) $k = 1.5$. The reason for these two modifications is that the cosine value of high dimensional vectors can reflect their similarity better than the distance, and a small k can give more satisfactory clustering result. These five categories include the American-flag, the Baseball-bat, the Beer-mug, the Billiards and the Calculator, with 97, 127, 94, 100 and 100 sample images, respectively. Table 5 shows the clustering results of each category in detail. We can see that in clustering results of the American-flag and Baseball-bat categories there exist nearly 1/5 images that were discarded as noise due to the complex background of these images. However, the inter-category classification

⁵ Available at http://www.vision.caltech.edu/Image_Datasets/Caltech256/

⁶ Available at <http://www.vlfeat.org/matconvnet/>



(a) correct clustering result

American-flag



American-flag
↓
Billiards



Baseball-bat



Baseball-bat
↓
Billiards



Beer-mug



Billiards
↓
Baseball-bat



Billiards



Billiards
↓
Calculator



Calculator



Calculator
↓
Baseball-bat

(b) noises examples

(c) misclassification examples

Figure 9: An illustration of the proposed PLinkage on the Caltech-256 dataset. “↓” on (c) indicates “misclassified to”, for example, on the first row the American-flag was “misclassified to” the Billiards.

⁴⁷⁸ error is quite low, which means that the PLinkage can distinguish these five categories
⁴⁷⁹ from each other very well, also the correct classification ratios of each category are 77%,

Table 5: Clustering results of the proposed PLinkage method on the Caltech256 dataset with CNN features.

	American-flag	Baseball-bat	Beer-mug	Billiards	Calculator	Noise	Correct Ratio
American-flag	75	1	0	1	0	20	0.77
Baseball-bat	0	89	1	6	0	31	0.70
Beer-mug	0	0	88	0	0	5	0.94
Billiards	0	3	0	86	1	10	0.86
Calculator	0	1	0	0	95	4	0.95

480 70%, 94%, 86%, and 95%, respectively. Figure 9 is a demonstration of the clustering
481 result on these five categories, Figure 9(a) shows 10 correct clustering result of each
482 category. Figure 9(b) shows the noise images of each category, from which we can see
483 that either the background of them are complex or the perspectives are oblique, thus the
484 images overwhelm the target objects and were discarded as noises. The misclassification
485 examples shown in Figure 9(c) are mostly reasonable, for example, on the first row the
486 American-flag was misclassified into the billiard because the flag is actually on the desk,
487 which is very similar to the billiard. Also on the last row, the calculator was regarded as
488 a baseball-bat by mistake for the reason of its long and thin shape, which is exactly the
489 characteristic of the baseball-bat. As a conclusion, we can see that, considering the fact
490 that it is only a simply density based clustering algorithm, the proposed PLinkage can
491 be applied on some high level issues, like image classification, to a certain degree.

492 4.2. Evaluation on PLinkage-Based Image Segmentation

493 **Influence of α :** α is used to balance the effects of $d^r(\mathbf{p}_i, \mathbf{p}_j)$ and $d^s(\mathbf{p}_i, \mathbf{p}_j)$ on the
494 value of $d(\mathbf{p}_i, \mathbf{p}_j)$ in Eq. (7), which means that the greater α is the more likely a data
495 point may be linked to its neighbour in the image space instead of in the LAB color
496 space. Figure 10 shows the image segmentation results on an outdoor image with $\alpha = 0.0$
497 and $\alpha = 0.5$, respectively. By setting $\alpha = 0.0$, it means that only the LAB information
498 is considered in the searching for CANP. We can see that, in Figure 10(b), more details
499 were preserved, for example, the rail of the build and the drape of the dress marked in

the red rectangles, but there are also many white dots beside the boundaries. While in Figure 10(c), there are less details but no white dots. This is for the reason that, when $\alpha = 0.0$ each pixel was linked to its CANP in the LAB color space regarding of the distance between them in the image space, the CANP can distinguish the small difference of the image on each side of the weak boundary like the drape of the dress very well. However, the pixels on one side of a boundary may also be linked to a CANP on the other side because there is no distance restriction in the image space, which leads to the white dots.

When $\alpha = 0.5$, both the distances in the LAB color space and the image space were taken into consideration, the pixel collection is more like region growing in this way, so the sky was segmented into a whole and some details were faded out, for example, for the clouds in the upper right, there are more details in Figure 10(b) than Figure 10(c). However, the segmentation result with $\alpha = 0.5$ is much neater, so we recommend to set $\alpha = 0.5$ for general cases. Of course, we can set a more suitable value for α for some specific applications.



Figure 10: An example of applying the PLinkage-based image segmentation on an outdoor image with different values of α .

Influence of h_s and h_r : The meanings of h_s and h_r are as the same as those of the mean shift based image segmentation method. Generally, we recommend to set $h_s = 10$ for constant, because $h_s = 10$ means that the neighbourhood of each pixel is a 21×21 window which is sufficient for general applications. Also, a larger value of h_s means more time consuming in density calculation. As to h_r , the greater it is the less segments there are. Figure 11 shows the segmentation results with different values of h_r , from which we can see that with the increasing of h_r , the neighbouring segments with



(a) $h_r = 4.0$

(b) $h_r = 6.5$

(c) $h_r = 10.0$

Figure 11: An example of applying the PLinkage-based image segmentation on a prairie image with different values of h_r .

close colors were merged step by step, while the segments with distinguishing colors were kept individually. According to a large number of experimental tests, we recommend to set $(h_s, h_r) = (10, 6.5)$ for general cases.

Robustness: To test the robustness of the proposed PLinkage-based image segmentation algorithm, we applied it on the BSDS500 dataset ⁷, and compared it with the mean shift based image segmentation algorithm ⁸. We set $h_s = 10$ and $h_r = 6.5$ for both the PLinkage and mean shift methods. Figure 12 shows the segmentations results of these two methods, from which we can see that the segmentation results of the PLinkage-based method are neater and more complete than those of the mean shift based method in general, for example, in the second column in Figure 12, the hair (marked in rectangular) and hand of the women were segmented quite well by the PLinkage-based method but separated into a lot of fragments in the mean shift based method. On the other hand, the mean shift based method can often achieve more details than the PLinkage-based method, for example, in the image regions covered by the surface of the building in the third column and the trees in the fourth column.

5. Conclusion

In this paper, we present a density ascending clustering algorithm named Pairwise Linkage (PLinkage) for the clustering of the Gaussian distributed data. Derived from

⁷Available at <https://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/>

⁸Available at <http://coewww.rutgers.edu/riul/research/code.html>



Figure 12: Image segmentation results on six representative images by applying the PLinkage based method and the mean shift based one, respectively: the original images, the results of PLinkage and the results of mean shift from top to down.

539 the idea of non-maximum suppression (NMS), the proposed method is based on the
 540 assumption that: a data point should be in the same cluster with its closest ascending
 541 neighbouring point (CANP). The relationship from this data point to its CANP is called
 542 “pairwise linkage”, and the clusters can be discovered by applying a depth-first or breadth-
 543 first searching from the cluster centers. The proposed PLinkage is non-iterative and can
 544 be easily adapted to the clustering of the smooth distributed dataset and the image
 545 segmentation easily. The PLinkage based algorithm for image segmentation was also
 546 proposed in this paper. Experiments on various public data sets with different dimensional
 547 data were conducted to illustrate the robustness of the proposed PLinkage, and also the
 548 image segmentation results on the BSDS dataset [2] sufficiently validated the practical

549 application of the PLinkage clustering algorithm on image segmentation.

550 **Acknowledgment**

551 This work was partially supported by the National Natural Science Foundation of
552 China (Project No. 41571436), the Hubei Province Science and Technology Support
553 Program, China (Project No. 2015BAA027), the National Natural Science Foundation of
554 China under Grant 91438203, and the South Wisdom Valley Innovative Research Team
555 Program.

556 **References**

- 557 [1] G. Griffin, A. Holub, P. Perona, Caltech-256 object category dataset,
558 http://www.vision.caltech.edu/Image_Datasets/Caltech256/ (2007).
- 559 [2] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images
560 and its application to evaluating segmentation algorithms and measuring ecological statis-
561 tics, in: IEEE International Conference on Computer Vision (ICCV), Vol. 2, 2001, pp.
562 416–423.
- 563 [3] P. Berkhin, A survey of clustering data mining techniques, in: Grouping multidimensional
564 data, Springer, 2006, pp. 25–71.
- 565 [4] Y. Anzai, Pattern Recognition & Machine Learning, Elsevier, 2012.
- 566 [5] K. L. Wagstaff, V. G. Laidler, Making the most of missing values: Object clustering with
567 partial data in astronomy, in: Astronomical Data Analysis Software and Systems XIV,
568 2005.
- 569 [6] M. R. N. Kalhori, M. F. Zarandi, Interval type-2 credibilistic clustering for pattern recog-
570 nition, Pattern Recognition 48 (11) (2015) 3652–3672.
- 571 [7] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern
572 Analysis and Machine Intelligence 22 (8) (2000) 888–905.

- 573 [8] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Susstrunk, SLIC superpixels com-
574 pared to state-of-the-art superpixel methods, *IEEE Transactions on Pattern Analysis and*
575 *Machine Intelligence* 34 (11) (2012) 2274–2282.
- 576 [9] X. Zhang, G. Li, Y. Xiong, F. He, 3D mesh segmentation using mean-shifted curvature, in:
577 *Advances in Geometric Modeling and Processing*, Vol. 4975 of *Lecture Notes in Computer*
578 *Science*, Springer, 2008, pp. 465–474.
- 579 [10] R. Xu, D. Wunsch, Survey of clustering algorithms, *IEEE Transactions on Neural Networks*
580 16 (3) (2005) 645–678.
- 581 [11] J. MacQueen, et al., Some methods for classification and analysis of multivariate obser-
582 vations, in: *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and*
583 *Probability*, 1967, pp. 281–297.
- 584 [12] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters
585 in large spatial databases with noise, in: *Proceedings of the 2nd International Conference*
586 *on Knowledge Discovery and Data Mining*, 1996, pp. 226–231.
- 587 [13] K. M. Kumar, A. R. M. Reddy, A fast DBSCAN clustering algorithm by accelerating
588 neighbor searching using groups method, *Pattern Recognition*.
- 589 [14] P. Cheeseman, J. Kelly, M. Self, J. Stutz, W. Taylor, D. Freeman, AutoClass: A Bayesian
590 classification system, in: *Readings in Knowledge Acquisition and Learning*, Morgan Kauf-
591 mann Publishers Inc., 1993, pp. 431–441.
- 592 [15] R. T. Ng, J. Han, Clarans: A method for clustering objects for spatial data mining, *IEEE*
593 *Transactions on Knowledge and Data Engineering* 14 (5) (2002) 1003–1016.
- 594 [16] W. Zhang, D. Zhao, X. Wang, Agglomerative clustering via maximum incremental path
595 integral, *Pattern Recognition* 46 (11) (2013) 3056–3065.
- 596 [17] L. Kaufman, P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*,
597 Vol. 344, John Wiley & Sons, 2009.
- 598 [18] A. Rinaldo, L. Wasserman, Generalized density clustering, *The Annals of Statistics* 38 (5)
599 (2010) 2678–2722.

- 600 [19] M. K. Ng, J. C. Wong, Clustering categorical data sets using tabu search techniques,
601 Pattern Recognition 35 (12) (2002) 2783–2790.
- 602 [20] P. Novák, P. Neumann, J. Macas, Graph-based clustering and characterization of repetitive
603 sequences in next-generation sequencing data, BMC bioinformatics 11 (1) (2010) 1.
- 604 [21] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science
605 344 (6191) (2014) 1492–1496.
- 606 [22] H.-S. Park, C.-H. Jun, A simple and fast algorithm for K-medoids clustering, Expert Sys-
607 tems with Applications 36 (2) (2009) 3336–3341.
- 608 [23] D. Comaniciu, P. Meer, Mean shift: A robust approach toward feature space analysis,
609 IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (5) (2002) 603–619.
- 610 [24] K.-L. Wu, M.-S. Yang, Mean shift-based clustering, Pattern Recognition 40 (11) (2007)
611 3035–3052.
- 612 [25] Y. A. Sheikh, E. A. Khan, T. Kanade, Mode-seeking by medoidshifts, in: IEEE Interna-
613 tional Conference on Computer Vision, 2007, pp. 1–8.
- 614 [26] R. Sibson, SLINK: an optimally efficient algorithm for the single-link cluster method, The
615 Computer Journal 16 (1) (1973) 30–34.
- 616 [27] A. Neubeck, L. Van Gool, Efficient non-maximum suppression, in: International Conference
617 on Pattern Recognition (ICPR), 2006, pp. 850–855.
- 618 [28] H.-S. Park, C.-H. Jun, A simple and fast algorithm for k-medoids clustering, Expert Sys-
619 tems with Applications 36 (2) (2009) 3336–3341.
- 620 [29] M. Sezgin, B. Sankur, Survey over image thresholding techniques and quantitative perfor-
621 mance evaluation, Journal of Electronic Imaging 13 (1) (2004) 146–165.
- 622 [30] N. Otsu, A threshold selection method from gray-level histograms, Automatica 11 (285-296)
623 (1975) 23–27.

- 624 [31] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images
625 and its application to evaluating segmentation algorithms and measuring ecological statistics,
626 in: IEEE International Conference on Computer Vision (ICCV), Vol. 2, 2001, pp.
627 416–423.
- 628 [32] D. R. Martin, C. C. Fowlkes, J. Malik, Learning to detect natural image boundaries us-
629 ing local brightness, color, and texture cues, IEEE Transactions on Pattern Analysis and
630 Machine Intelligence 26 (5) (2004) 530–549.
- 631 [33] P. Dollár, C. Zitnick, Structured forests for fast edge detection, in: Proceedings of the
632 IEEE International Conference on Computer Vision, 2013, pp. 1841–1848.
- 633 [34] P. Arbelaez, M. Maire, C. Fowlkes, J. Malik, Contour detection and hierarchical image
634 segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (5)
635 (2011) 898–916.
- 636 [35] L. Wang, C. Pan, Robust level set image segmentation via a local correntropy-based K-
637 means clustering, Pattern Recognition 47 (5) (2014) 1917–1925.
- 638 [36] R. Nock, F. Nielsen, Statistical region merging, IEEE Transactions on Pattern Analysis
639 and Machine Intelligence 26 (11) (2004) 1452–1458.
- 640 [37] F. Porikli, Constant time O(1) bilateral filtering, in: IEEE Conference on Computer Vision
641 and Pattern Recognition, 2008, pp. 1–8.
- 642 [38] Q. Yang, K.-H. Tan, N. Ahuja, Real-time O(1) bilateral filtering, in: IEEE Conference on
643 Computer Vision and Pattern Recognition, 2009, pp. 557–564.
- 644 [39] P. Fränti, O. Virmajoki, Iterative shrinking method for clustering problems, Pattern Recog-
645 nition 39 (5) (2006) 761–775.
- 646 [40] G. Karypis, E.-H. Han, V. Kumar, Chameleon: Hierarchical clustering using dynamic
647 modeling, Computer 32 (8) (1999) 68–75.
- 648 [41] A. Gionis, H. Mannila, P. Tsaparas, Clustering aggregation, ACM Transactions on Knowl-
649 edge Discovery from Data (TKDD) 1 (1) (2007) 4.

- 650 [42] H. Chang, D.-Y. Yeung, Robust path-based spectral clustering, *Pattern Recognition* 41 (1)
651 (2008) 191–203.
- 652 [43] C. J. Veenman, M. J. Reinders, E. Backer, A maximum variance cluster algorithm, *IEEE*
653 *Transactions on Pattern Analysis and Machine Intelligence* 24 (9) (2002) 1273–1280.
- 654 [44] L. Fu, E. Medico, FLAME, a novel fuzzy clustering method for the analysis of DNA mi-
655 croarray data, *BMC Bioinformatics* 8 (3) (2007) 1–15.
- 656 [45] P. Franti, O. Virmajoki, V. Hautamaki, Fast agglomerative clustering using a k-nearest
657 neighbor graph, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28 (11)
658 (2006) 1875–1881.
- 659 [46] A. Vedaldi, K. Lenc, Matconvnet – convolutional neural networks for matlab, in: Proceed-
660 ing of the ACM International Conference on Multimedia, 2015.
- 661 [47] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recog-
662 nition, *CoRR* abs/1409.1556.
- 663 [48] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical
664 image database, in: *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE,
665 2009, pp. 248–255.