Scale-invariant density-based clustering initialization algorithm and its application

Chunsheng Hua, Ryusuke Sagawa, Yasushi Yagi
Department of Intelligent Media, ISIR of Osaka University,
8-1 Mihogaoka, Ibaraki, Osaka, 567-0047, Japan
huachunsheng@gmail.com, sagawa,yagi@am.sanken-u.ac.jp

Abstract

In this paper, we bring out a new density-based clustering initialization algorithm which is invariant to the scale factor. Instead of using the scale factor while the cluster initialization, in this research, we determine the number and position of clusters according to the changes of cluster density with the division and agglomeration processes. During the division process, the initial cluster seeds are produced by a self-propagate method according to the density changes. The number of clusters is determined by agglomerating pair of RNN (reciprocal nearest neighbor) cluster seeds, when the density of newly merged cluster is increased. When no more cluster seeds can be merged any more, the remained number of cluster seeds is regarded as the real cluster number. Through various experiments, the effectiveness of the proposed algorithm has been proved.

1. Introduction

Clustering algorithm is an elemental part of pattern recognition. Its kernel is to classify the unlabeled nature data into meaningful groups with little prior information. However, most of the clustering algorithms require the user to provide the initial number and position of clusters, when the number of clusters is huge, this work is tedious and difficult to be manually performed.

To solve this problem, the clustering initialization algorithms can be divided into two types: (**I**) assuming the number of clusters is given, searching for the best initial position; (**II**) both the position and number of clusters are assumed to be unknown.

SMEM (Split-Merge EM) algorithm [6] (type **I**) estimates the best initial position of clusters by embedding the split-merge criterion into EM algorithm. The cluster that maximizes the split criterion (a ratio between the local data density and the density of current parameter) will be split. When the posterior of each data in two neighbor clusters is almost equal to each other, such two clusters are merged together. The SMEM will be stopped when the Q function value is maximized. Improved step-wise SMEM algorithm (SSMEM)[9] allows both the initial position and number of clusters to be unknown. However, its success depends on the manually selected threshold of split and merge criterion. The success of LBG [4] also relies on the prior known number of clusters and the carefully selected threshold.

The initialization methods of type **II** can be further categorized into: *divisive* and *agglomerative* types. Simple Cluster Seeking (SCS) [2] method (divisive type) searches for the new cluster seeds whose distance to the previous seed is longer than the fixed threshold, where the initial seed is randomly assigned.

Random sampling [1] is widely applied for the agglomerative cluster initialization. In [1], the uniformly sampled cluster seeds that can not attract data will be ignored and the remained seeds is used as the initial clusters. The problem is that its result is not reliable if the number of seeds is smaller than that of real clusters.

Compared with the distance-based divisive initialization algorithms, the density-based methods [5, 7, 12] can be regarded as the non-parameter method which uses the gradient to describe the cluster feature. Mode seeking method [5] can detect the clusters by merging two randomly sampled cluster seeds together when the distance between them is smaller than the selected scale factor. Therefore, the performance of this algorithm heavily depends on the selection of scale factor. Further studies on the auto selection of scale factor (e.g. influence zones [7], critical scale [12]) have been reported.

To solve the problem of selecting scale factor, we bring out a scale-invariant density-based cluster initialization algorithm. The key contribution of this work is to use the density changes to detect the cluster boundary. Since the gap between clusters can be considered as the cluster boundary whose density is zero, merging two isolated clusters will decrease the density of merged cluster. Therefore, after producing the initial cluster seeds with a self-propagate method, the final number of clusters can be determined by iteratively merging two RNN clusters if the density of merged cluster is increased until no more cluster seeds can be merged.

2. Scale-invariant density-based clustering initialization

2.1 Division Criterion

Although random sampling can be applied to produce the initial cluster seeds, the assumption of its success is that the sampled seeds are more than the number of real clusters.

Therefore, large number of initial seeds is always required in random sampling, which will lead to much useless computation on the dead seeds. Especially, when the dataset is unknown, it is almost impossible to know how many initial seeds are enough.

To solve this problem, in this work, an iterative division method is used to produce the cluster seeds according to the natural density distribution of clusters. The dataset is firstly divided into m regions as r_i, and the density of r_i is calculated as:

$$d_i = \frac{N_i}{V_i} \quad (i = 1 \sim m), \tag{1}$$

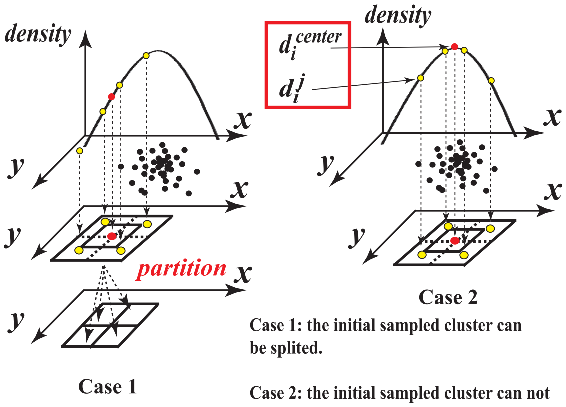
where N_i, V_i represent the number of data and volume of r_i, respectively. To avoid the unnecessary computation on the dead cluster seeds, dead seed is defined as:

$$d_i \leq k * d_{mean}, \quad d_{mean} = \frac{1}{K}\sum_{i=1}^{K} d_i, \quad (d_i \neq 0), \tag{2}$$

where, k is a constant and K is the number of non-dead seeds. Such dead seeds will be ignored and the remained seeds will be 4-equal divided into $R_i^j, (j = 1 \sim 4)$ as shown in Fig.1. The density of R_i^j is calculated as d_i^j. As shown in Fig.1, a sub region R_i^{center} that has the same size as R_i^j and centered at r_i is extracted and its density is defined as d_i^{center}. r_i will be partitioned if it satisfies:

$$d_i^{center} \leq d_i^j, \tag{3}$$

the partitioned cluster seeds will be iteratively split until it reaches the fixed iteration. Then, K-means clustering will be used to refine the shape and density of each cluster seed R_i. Experimental results also show that this algorithm is not limited to the normal distribution, it can be applied to any compact data. Fig.2 shows the result of division process on the simulated data. where 8 real clusters exist and 32 cluster seeds are produced after the division process.



Case 1: the initial sampled cluster can be splited.

Case 2: the initial sampled cluster can not be splited

Figure 1. Illustration of division.

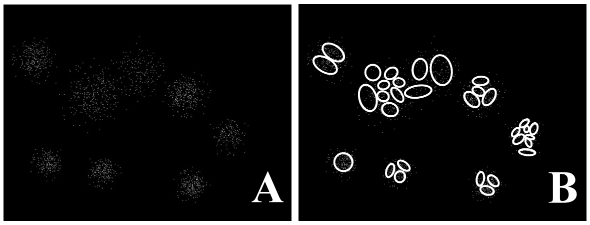


Figure 2. Division of the simulated data. A: input data; B: division result

2.2 Agglomeration Criterion

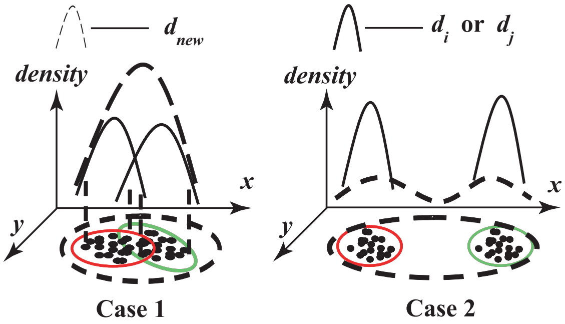
After the division process, the kernel of this work is to detect the number and position of clusters by merging the cluster seeds according to the density distribution. The conventional density-based methods [5, 7, 9] use the scale factor to find out the number of clusters. Therefore, their performance heavily depends on the manual selection of proper scale factor.

Since a valley must exist between two neighbor peaks in the density histogram, we consider the gap (whose density is about zero) between clusters as the cluster boundary. Therefore, the cluster detection can be converted into detecting the gaps among clusters. On this consideration, merging two isolated clusters will decrease the density of merged cluster. Therefore, the cluster detection in this work can be performed by checking the density changes before and after merging.

After finding out the reciprocal nearest neighbor R_j to each cluster seed R_i and its density d_j, the density of cluster R_{new} will be defined as d_{new}, where $R_{new} = R_i \cup R_j$. Like Case 1 in Fig.3, the two clusters will be merged together, if they satisfy:

$$d_{new} \geq argmin(d_i, d_j), \tag{4}$$

When all the cluster seeds have been checked, K-means clustering is used to refine the cluster density as well as get the new cluster center. This merging process will be repeated until no more cluster seeds can be merged. The initial position of clusters is obtained from the cluster refinement by K-means clustering.



Case 1: two clusters should be merged
Case 2: two clusters should not be merged

Figure 3. Illustration of agglomeration.

Fig.4 shows the performance of this agglomeration process on the result of Fig.2. This process is stopped at **B** when no clusters can be merged any more.

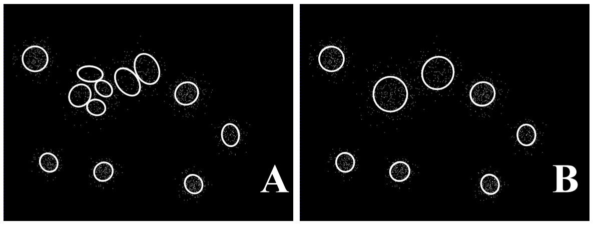


Figure 4. Merge the divided clusters. A: merge once; B: merge twice

3. Experiment

To confirm the effectiveness of the proposed algorithm, we applied it to both the simulated data [1] and the Berkeley dataset [2] for image segmentation.

R 1-2 of Fig.5 show that the proposed algorithm could correctly find the initial number and position of clusters even some of them intersect with each other.

R,M 3 $\sim$ 5 show the comparative results of the proposed algorithm and mean shift on image segmentation. In M 3 $\sim$ 5, mean shift algorithm [3] ran at the default scale factor (h_s, h_r, M)=(7,6.5,20), and the corresponding detected cluster numbers are 362, 1291 and 1135. In contrast, the proposed algorithm could provide similar image segmentation result to mean shift, but its cluster detection results are 29, 32, and 14 clusters, which is more reasonable. That is because, in mean shift algorithm, the pair of neighbor clusters are merged by checking the distance between them with the selected scale factor, when the distance between clusters is larger than scale factor, such clusters will be merged, otherwise will be kept isolated. Therefore, it is very possible that the single cluster with large variance may be forcibly split by the small scale factor, and two isolated clusters may be wrongly merged by a large scale factor.

Because the proposed algorithm used the density changes before and after merging to determine if the pair of neighbor clusters should be really merged or not, it became invariant to the scale factor. When the parameter of mean shift is changed to get similar cluster detection result to our algorithm (shown in ML3$\sim$5 of Fig.5), the detected numbers of clusters are 30, 37 and 44 clusters, but the image segmentation result were degraded because many isolated clusters were forcibly merged by the large scale factors. The proposed algorithm was directly applied to a 5D color-position feature space which contained 320 × 480 × 256^3 elements. The processing speed of our algorithm is 13 seconds/frame with a desktop of Intel C2D 2.66Ghz and 4GB memory.

4. Conclusion

In this paper, we brought out a scale-invariant density-based clustering initialization algorithm. According to the density changes, a self propagate method is brought out in this algorithm to produce the cluster seeds at all the possible position. By defining the gap between clusters as the cluster boundary whose density is zero, the cluster detection can be achieved by checking the density changes before and after merging two RNN cluster seeds. Through experiments, this algorithm can be used as the initialization method for clustering, image segmentation, object tracking (like background subtraction), etc.

Acknowledge

[1] some data comes from the SPAETH dataset

[2] http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping /segbench/

[3] the EDISON software

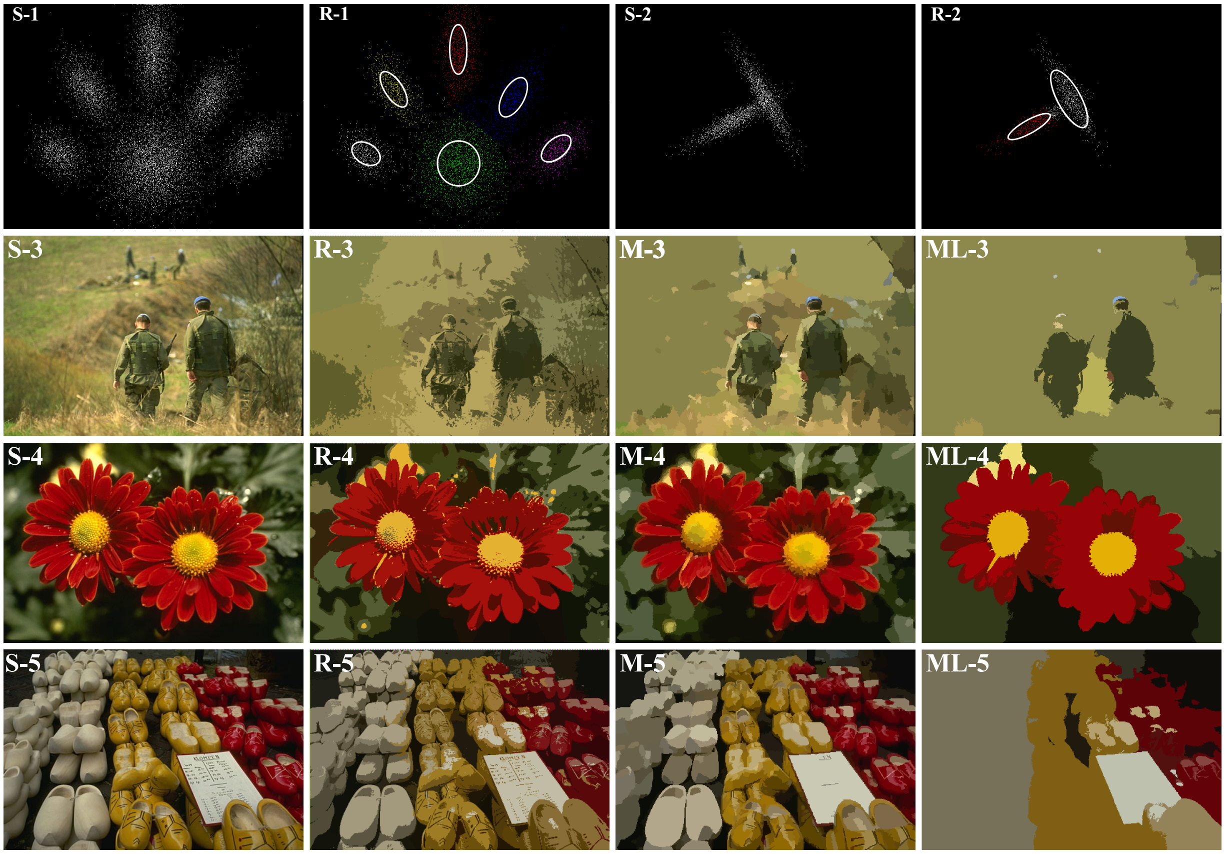


Figure 5. Result of the simulated data and real image. S: source; R: result of our algorithm; M: result of mean shift; ML: mean shift with large scale. In M $3 \sim 5$, (h_s, h_r, M)= (7,6.5,20). Cluster detection result: R-3: 29 clusters; R-4: 32 clusters; R-5: 14 clusters; M-3:362 clusters; M-4:1291 clusters; M-5: 1135 clusters. ML-3:(h_s, h_r, M)=(19,15,20), 30 clusters; ML-4: (h_s, h_r, M)=(31,26,20), 37 clusters; ML-5: (h_s, h_r, M)=(39,32,20), 44 clusters.

This work is supported by the Special Coordination Funds for Promoting Science and Technology of Ministry of Education, Culture, Sports, Science and Technology.

References

[1] E.Forgy: Cluster analysis of multivariate data: efficiency vs interpretability of classifications. *WNAR*, 1965

[2] J.T.Tou, R.C.Conzalez: Pattern Recognition Principles Addison Wesley, Massachusetts, 1974

[3] Dempster, A.P, Laird, N.M, Rubin, D.B.: Maximum Likehood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society B*, 39, pp.1-38, 1977

[4] Y.Linde, A.Buzo, R.Gray: An Algorithm for Vector Quantizer Design *IEEE Trans. Communications*, Vol.com-28, No.1, pp.84-95, 1980

[5] D.Comaniciu, P.Meer: Distribution Free Decomposition of Multivariate Data *SPR*, 1998

[6] N.Ueda, etc: SMEM Algorithm for Mixture Models. *Neural Computation* 12, pp.2109-2128, 2000

[7] M.Herbin, N.Bonnet, etc: Estimation of the number of clusters and influence zones *PRL*, Vol.22, pp.1557-1568, 2001

[8] J.He, M.Lan, etc: Initialization of Cluster Refinement Algorithms: A Review and Comparative Study. *IJCNN*, Vol.1 pp.297-302, 2004

[9] H.X.Wang, B.Luo, etc: Estimation for the number of components in a mixture model using stepwise split-merge EM algorithm. *PRL*,Vol.25, pp.1799-1809, 2004

[10] B.Leibe, K.Mikolajczyk, B.Schiele: Efficient clustering and matching for object class recognition, *BMVC*,2006

[11] C.Zhang, X.Zhang, etc: Neighbour number, valley seeking and clustering. *PRL*, Vol.28, pp.173-180, 2007

[12] T.Sakai, T,Komazaki, etc: Critical Scale for Unsupervised Cluster Discovery *MIRU*,pp.1165-1170, 2007