A Stage by Stage Pruning Algorithm for Detecting the Number of Clusters in a Dataset

Yanqiao Zhu and Jinwen Ma*

Department of Information Science, School of Mathematical Sciences & LMAM
Peking University, Beijing, 100871, P.R. China
jwma@math.pku.edu.cn

Abstract. Determining the number of clusters in a dataset has been one of the most challenging problems in clustering analysis. In this paper, we propose a stage by stage pruning algorithm to detect the cluster number for a dataset. The main idea is that from the dataset we can search for the representative points of clusters with the highest accumulation density and delete the other points from their neighborhoods stage by stage. As the radius of the neighborhood increases, the number of searched representative points decreases. However, the structure of actual clusters of the dataset makes the representative point number be stable at the true cluster number in a relative large interval of the radius, which helps us to detect the cluster number. It is demonstrated by the simulation and practical experiments that the proposed algorithm can lead to an effective estimate of the cluster number for a general dataset.

Keywords: Clustering analysis, Cluster number detection, Accumulation density, Representative point, Stage by stage pruning.

1 Introduction

As a powerful data analysis tool, clustering analysis has been widely used in information processing and pattern recognition. Actually, there have been a variety of clustering approaches such as the k-means algorithm [1] as well as the fuzzy c-means algorithm [2], the frequency sensitive competitive learning (FSCL) algorithm [3], the mixture models with the EM algorithm [4] and the spectral clustering. However, most of these approaches take the number k of clusters as a pre-known information, i.e., the cluster number should be given in advance. Due to the great diversity of data structure, the determination of the cluster number for a general dataset has been still a rather challenging problem in clustering analysis. In fact, many attempts have been made to detect or estimate the number of clusters in a dataset.

The traditional approach is to choose an optimal number k^* of clusters in the dataset via one of information, coding and statistical selection criteria such as Akaike's Information Criterion (AIC) [5], Bayesian Inference Criterion (BIC) [6], Minimum Message Length (MML) [7] and GAP statistic [8]. But the validating

* Corresponding author.

D.-S. Huang et al. (Eds.): ICIC 2010, LNCS 6215, pp. 222–229, 2010.

process is computationally consumptive because we need to repeat the entire parameter learning process (such as the implementation of the EM algorithm on all the dataset) at a large number of possible values of k. Moreover, all the existing selection criteria have their limitations and often lead to a wrong result.

Alternatively, there are split-and-merge or split clustering algorithms which can also lead to the true number of clusters in the dataset from any initial setting of k. In fact, the ISODATA algorithm [9], the general competitive clustering [10], and the BYY split-and-merge EM algorithm [11] all have no requirement on the initial setting of k, but it is certain that such a clustering algorithm will converge more quickly and correctly if the initial setting of k is close to the correct value, i.e., the true number of clusters in the dataset. In certain cases, the proper setting of k is even necessary. So, it is very valuable to detect or even estimate the number of clusters in the dataset before the implementation of clustering.

Practically, we can design a search algorithm or procedure on the data to detect or estimate the number of clusters in the dataset. In fact, there have been already such search approaches. The visual method and the subtractive clustering method are two typical examples. The visual method firstly constructs a reordered dissimilarity image where the dark blocks along the diagonal represent the clusters and can be detected by certain image processing techniques [12,13]. In the subtractive clustering method, the accumulation density of each data point is computed such that each peak of the density distribution corresponds to a cluster center. Thus, the number of peaks of the density distribution can give an estimate of the cluster number [14]. Moreover, Yu and Cheng tried to discover the searching scope of the optimal cluster number for the FCM algorithm from the perspective of information theory [15].

In the current paper, we try to propose a stage by stage pruning method to detect the cluster number in a dataset. We begin to define the accumulation density for each data point. We then search for the representative points of clusters with the highest accumulation density and delete the other points from their neighborhoods stage by stage. As the radius of the neighborhood increases, the number of searched representative points decreases. But the dividable cluster structure of the dataset enables the representative point number to be stable at the true cluster number in a relative large interval of the radius. This flat level gives a good estimate of the cluster number. Actually, the simulation experiments demonstrate that the proposed algorithm can lead to an effective estimate of the cluster number for a general dataset.

The rest of this paper is organized as follows. Section 2 presents the stage by stage pruning algorithm. The simulation and practical experiments are conducted in Section 3. Section 4 contains a brief conclusion.

2 The Stage By Stage Pruning Algorithm

2.1 The Basic Idea

We begin with a description of the basic idea for our stage by stage pruning algorithm. Actually, it comes from the intuition that if we can prune the data such that there is only one representative point left in each cluster, we can easily get the number of clusters just as the number of representative points left.

In order to implement this idea, we can utilize the accumulation density distribution of the data point which can be defined and computed on the dataset. A representative point for a cluster is supposed to own the highest density around its neighborhood, serving as the cluster center. So we first search for the data point with the highest accumulation density and let it be the representative point and prune the other points in its neighborhood with a given and fixed radius. In the remaining data points except the representative one, we can again search another representative point and prune the other points in the neighborhood of this representative point. In such a way, we can obtain a set of representative points stage by stage.

Clearly, the number of representative points can equal to the number of clusters when the radius of the neighborhood is properly set. But it is rather difficult to set the radius of the neighborhood properly. However, we can increase the radius step by step and get the representative point number for each value of the radius. As the radius of the neighborhood increases, the number of searched representative points decreases. But the dividable cluster structure of the dataset enables the representative point number to be stable at the true cluster number in a relative large interval of the radius. This flat level actually gives a good estimate of the cluster number.

So far, we need only to define the accumulation density of a data point. The most convenient way is to count up the number of its neighbors (whose distances to this data point are less than the given threshold r which will be consistent with the radius of the neighborhood used each search step). In fact, with such a density of the data point, the above stage by stage pruning procedure can be conducted on the three-cluster dataset given in the left part of Fig. 1, in which the radius of the neighborhood increases from one to 50 times of the initial radius[1] (in 50 steps). It can be clearly observed from the right part of Fig. 1 that the attenuating curve of the number of representative points has a clear flat level 3 corresponding to the true number of clusters in the dataset. This simulation result demonstrates that the idea of the stage by stage pruning procedure with the change of the radius of the neighborhood is effective and can be applied to detect the number of clusters in a dataset.

2.2 The Presentation of the Algorithm

We further present the stage by stage pruning algorithm according to the above basic idea. For clarity, we refer to the number of representative points left at the end of a pruning step as the state of the algorithm. Suppose that the dataset consists of N data points being denoted by $x_i \in R^n, i = 1, 2, \ldots, N$. $maxDist = max\{\|x_i - x_j\| \mid 1 \le i < j \le N\}$ denotes the maximum distance between any two data points, which is also referred to as the diameter of the dataset. We adopt

[1] The initial radius is set to be 1/50 of the maximum distance between any two data points in the dataset.

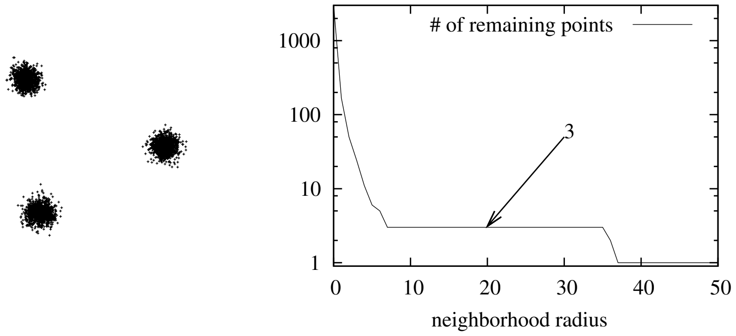

Fig. 1. The demonstration of the stage by stage pruning procedure

the following neighborhood increasing scheme and algorithm stop criterion. We equally divide the interval $[0, maxDist]$ into M parts and use $\{i \times maxDist/M : i = 1, 2, \ldots, M/2\}$ as the increasing radius sequence[2], i.e., $r = i \times maxDist/M$ is used to define the size of the neighborhood for the i-th step. As for the stop criterion, we introduce another parameter K. If one state sustains more than K steps (in other words, it appears more than $K + 1$ times consecutively), we then stop the algorithm and use the number of remaining representative data points in the state as an estimate of the cluster number. Details of our proposed stage by stage pruning algorithm are summarized in Algorithm 1.

3 Experimental Results

In this section, various simulation experiments are carried out to demonstrate the performance of the stage by stage pruning algorithm for detecting the number of clusters in a dataset. Moreover, the stage by stage pruning algorithm is implemented to detect the number of clusters in certain real-world datasets such as the Iris and wine datasets.

3.1 On Five Synthetic Datasets

We begin with a brief description of five synthetic datasets (shown in Fig. 2) used in our simulation experiments:

1. The clusters in $\mathcal{D}_1$ and $\mathcal{D}_2$ have equal number of samples, but those in the other three datasets have different numbers of samples.
2. The clusters in $\mathcal{D}_1, \mathcal{D}_3$ are well separated, but those in each of the other three datasets are overlapped at certain degree.
3. The clusters in $\mathcal{D}_1$ and $\mathcal{D}_2$ are spherical in shape, but those in the other three datasets are elliptic in shape.

[2] Usually we can take an even number for M, when $i = M/2$, $r = 1/2 \times maxDist$ is just the radius of the whole data.

```
Input: attribute vectors of data
Output: estimate of the cluster number
begin
    choose values for parameters $M$ and $K$;
    compute the distance between data points (Euclidean distance between
    corresponding attribute vectors);
    compute neighbors of each data point for each radius
    in $\{i/M \times maxDist: i = 1, 2, \ldots, M/2\}$;
end
set $count = 0$, $preSet = \emptyset$, $curSet = \{1, 2, \ldots, N\}$, $r = 1/M \times maxDist$;
while $count \leq K$ and $r \leq 1/2 \times maxDist$ do
    $preSet = curSet$;
    $curSet = \emptyset$;
    repeat
        begin
            find the data point in the remaining data which owns the most
            neighbors for current $r$ (randomly choose one if there are a few);
            add the corresponding index to $curSet$;
            delete all of its neighboring data points and set its neighbors to 0.;
        end
    until all of the data has been processed, either deleted or reserved ;
    if $curSet == preSet$ then
        $count = count + 1$
    else
        $count = 0$
    end
    $r = r + 1/M \times maxDist$;
end
output the number of remaining data points in current state;
```

Algorithm 1. The stage by stage pruning algorithm

We now implement the stage by stage pruning algorithm on the five synthetic datasets with $M = 50$ and $K = 2$. Since there is some randomness when more than one data points own the largest number of the neighbors, different simulations may lead to different results, especially for datasets with complicated structure. So, we run 10 simulations for each dataset, and take the mode of the corresponding results as the final output. The results are summarized in Tab. 1.

For $\mathcal{D}_1$ and $\mathcal{D}_3$, nine out of the ten simulations give the correct cluster number, and so does the final output. For $\mathcal{D}_2$, $\mathcal{D}_4$, the advantage of the correct cluster number is not so obvious. As for $\mathcal{D}_5$, the correct cluster number 3 and the wrong cluster number 5 both win 3 times. So we take the average value 4 as the final output, which is 1 larger than the true cluster number.

By studying the simulation results on the five synthetic datasets, we can find that as the level of overlap among clusters increases, the ratio of correct cluster number in the simulation results tends to decrease. And the overall performance of the algorithm on $\mathcal{D}_1$ and $\mathcal{D}_2$ is better than that on each of the other three

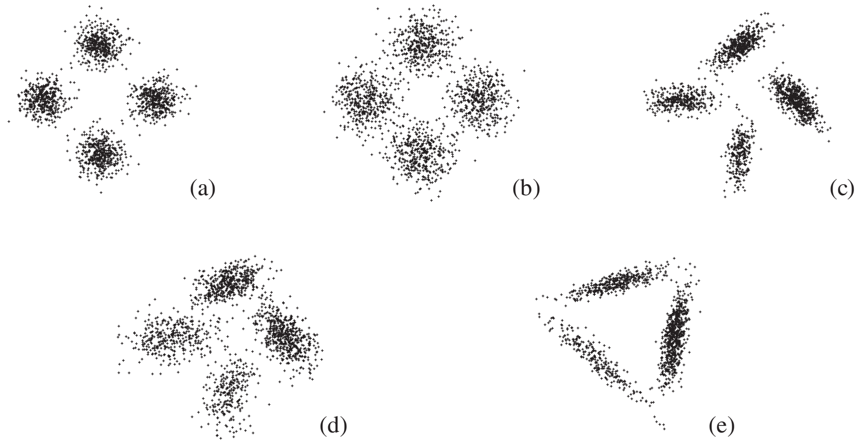

Fig. 2. Five synthetic datasets used in the simulation experiments. (a) dataset $\mathcal{D}_1$; (b) dataset $\mathcal{D}_2$; (c) dataset $\mathcal{D}_3$; (d) dataset $\mathcal{D}_4$; (e) dataset $\mathcal{D}_5$.

Table 1. Experimental results on five synthetic datasets (frequency of each output in 10 simulations)

Output Dataset	2	3	4	5	6	others	final output
$\mathcal{D}_1$	0	0.1	0.9(True)	0	0	0	4
$\mathcal{D}_2$	0.1	0.1	0.6(True)	0.1	0.1	0	4
$\mathcal{D}_3$	0	0.1	0.9(True)	0	0	0	4
$\mathcal{D}_4$	0.1	0.1	0.5(True)	0.3	0	0	4
$\mathcal{D}_5$	0	0.3(True)	0	0.3	0.2	0.2	4

datasets. Two reasons may account for this. First, in the algorithm, we will randomly choose one if a few data points all own the highest accumulation density at the same time. As the level of overlap among clusters increases, the randomness plays a more active role. Second, the algorithm is designed based on spherical neighborhood, which is not so appropriate for clusters in elliptic shape. As for $\mathcal{D}_5$, since clusters in the dataset are far from spherical-shaped, our algorithm cannot find an accurate result, but we can still get an effective estimate.

The algorithm is not sensitive to the differences among sample sizes of clusters, which can be supported by simulation results on $\mathcal{D}_3$. However, due to the neighborhood increasing scheme, the increment of neighborhood radius for all the clusters is the same, so we can expect the algorithm may not work well on datasets with clusters of dramatically different geographical sizes, say, one with diameter of 1000, while the others with diameter of 1.

3.2 On Real-World Datasets

On the Iris Data. The Iris dataset consists of 150 samples of three classes: Iris Versicolor, Iris Virginical and Iris Setosa, with each class containing 50 samples. Each datum consists of four attributes which represents measures of the plants morphology. Parameters are the same as previous settings. Six out of the ten simulations give the correct cluster number, so does the final output.

On the Wine Data. The wine data are typical high-dimensional real-world data. It contains 178 samples of three types of wine with 13-dimensional attributes. Ten experiments are conducted under $M = 50$ and $K = 2$. The results are better than that of Iris data with 80% of the simulations providing the correct cluster number.

3.3 Further Discussions

There are two free parameters in our proposed algorithm, M and K. Actually, M controls the increment of neighborhood radius, while K represents the stabling scale to output the result. Larger M means more computational cost since the radius increases slower, but the corresponding results are supposed to be more accurate. Meanwhile, with M being fixed, larger K means that we expect the clusters are farther away from each other, so clusters neighboring each other may be mistaken as one. Moreover, larger M usually is supposed be accompanied with larger K. The ideal case is that we can choose appropriate parameters based on the data to be processed, which is what we will look into in the future.

We can also consider the introduction of certain pre-process techniques before performing the algorithm, which aims at removing possible noisy data and making the boundaries more clear. For example, removing those data with few neighbors may improve the search result. Here we use Euclidean distance to define the neighborhood in this paper. But there are other choices such as the Mahalanobis distance or weighted Euclidean distance which treats each attribute differently based on their properties, which may be more proper for the datasets with more complicated structure.

4 Conclusions

We have established a stage by stage pruning algorithm for detecting the number of clusters in a dataset. The proposed algorithm searches for the representative points of clusters owning the highest accumulation densities with their neighborhoods with the neighborhood radius increasing stage by stage and takes the first stable representative point number as the estimate of the cluster number. It is demonstrated well by the experiments on both the synthetic and real-world datasets that the stage by stage pruning algorithm can provide an effective estimate of the cluster number.

Acknowledgements

This work was supported by the Natural Science Foundation of China for grant 60771061.

References

1. MacQueen, J.B.: Some Methods for classification and Analysis of Multivariate Observations. In: Proceedings of 5-*th* Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297. University of California Press, Berkeley (1967)
2. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
3. Ahalt, S.C., Krishnamurty, A.K., Chen, P., Melton, D.E.: Competitive learning algorithms for vector quantization. Neural Networks 3, 277–291 (1990)
4. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, B 39, 1–38 (1977)
5. Akaike, H.: A new look at the statistical model identification. IEEE Transactions on Automatic Control AC-19, 716–723 (1974)
6. Schwarz, G.: Estimating the dimension of a model. The Annals of Statistics 6, 461–464 (1978)
7. Wallace, C., Dowe, D.: Minimum Message Length and Kolmogorov Complexity. Computer Journal 42, 270–283 (1999)
8. Tibshirani, R., Walther, G., Hastie, T.: Estimating the number of clusters in a data set via the gap statistic. Journal of Royal Statistical Society, Series B (Statistical Methodology) 63, 411–423 (2001)
9. Ball, G.H., Hall, D.J.: ISODATA: a novel method of data analysis and pattern classification. Technique Report No. 699616, Stanford Research International (1965)
10. Boujemaa, N.: Generalized competitive clustering for image segmentation. In: Proc. of 19th International Conference of the North American Fuzzy Information Processing Society, pp. 133–137 (2000)
11. Li, L., Ma, J.: A BYY Split-and-Merge EM Algorithm for Gaussian Mixture Learning. In: Sun, F., Zhang, J., Tan, Y., Cao, J., Yu, W. (eds.) ISNN 2008, Part I. LNCS, vol. 5263, pp. 600–609. Springer, Heidelberg (2008)
12. Wang, L., Leckie, C., Ramamohanarao, K., Bezdek, J.: Automatically determining the number of clusters in unlabeled data sets. IEEE Transactions on Knowledge and Data Engineering 21, 335–350 (2009)
13. Sledge, I., Huband, J., Bezdek, J.: (Automatic) Cluster cluster count extraction from unlabeled datasets. In: Joint Proc. Fourth Int'l.Conf. Natural Computation and Fifth Int'l. Conf. Fuzzy Systems and Knowledge Discovery (2008)
14. Yager, R.R., Filev, D.P.: Approximate Clustering via the Mountain Method. IEEE Transactions Systems, Man, and Cybernetics 24, 1279–1284 (1994)
15. Yu, J., Cheng, Q.: The optimal range of number of clusters for the Fuzzy clustering methods. Science in China 32, 274–280 (2002)