Estimating the Number of Clusters as a Pre-processing Step to Unsupervised Learning

Paulo Rogerio Nietto[1](✉) and Maria do Carmo Nicoletti[1,2]

[1] Faculdade Campo Limpo Paulista (FACCAMP), Campo Limpo Paulista, SP, Brazil
{pnietto,carmo}@cc.faccamp.br
[2] Universidade Federal de São Carlos (UFSCar), São Carlos, SP, Brazil

Abstract. A great challenge in machine learning, as far as unsupervised algorithms are concerned, is to devise methods for pre-estimating the number of clusters associated to a given set of patterns to be clustered. By doing so and by using the number of clusters as input to clustering algorithms that require the information, the chances of getting better results increase substantially. The work described in this paper investigates the performance of an algorithm, based on the sequential clustering BSAS (*Basic Sequential Algorithmic Scheme*), to produce an ordered list (by frequency of occurrences), containing good estimates for the number of clusters in a given set of patterns. The BSAS is a convenient choice since the order in which patterns are presented to the algorithm can impact the induced clustering. The results of the experiments in eight sets of patterns can be considered empirical evidence that the procedure can be a practical and reliable option, as a pre-processing step, to using clustering algorithms that require the number of clusters.

Keywords: Clustering · Estimation of the number of clusters · Sequential clustering algorithms

1 Introduction

Machine learning (ML) algorithms that deal with data that do not have an associated class are known as unsupervised learning algorithms. Clustering algorithms constitute a group of unsupervised algorithms which, currently, can be considered the most popular among the unsupervised learning algorithms. In a simplistic way it can be said that the main goal of a clustering algorithm is to partition a set of patterns into groups (clusters) of patterns, so those that share the same cluster are similar to each other and those belonging to different clusters are not that similar. It can be found in the literature several clustering algorithms based on a variety of mathematical and statistical formalisms, such as in [1–4]; several taxonomies [5–8], associated to these algorithms, can also be found. Given a set of patterns, a reasonable number of clustering algorithms require, also, as an input parameter, the number (K) of clusters the induced clustering should have. This, somehow, makes the quality of the resulting clustering be heavily dependent on the value of K. In some applications the value of K can be estimated by human experts who have

A.M. Madureira et al. (eds.), *Intelligent Systems Design and Applications*,
Advances in Intelligent Systems and Computing 557,
DOI 10.1007/978-3-319-53480-0_3

experience and deep knowledge about the domain of the data; however, in most cases, a suitable value for K is unknown and must be estimated based only on the patterns themselves. A short description suggesting a procedure to estimate the number of clusters (K) in a set of patterns, based on the patterns themselves, was proposed in [6]. The procedure involves the use of clustering algorithms which do not require the information about the number of clusters. One of such algorithms is the *Basic Sequential Algorithmic Scheme* (BSAS) [6], which was adopted in the work described in this paper, to support the proposal of an algorithm, the *ClusterEstimate*, for determining the number K of clusters prior to a clustering task, aiming at directing it. The paper is organized as follows. Section 2 presents the *ClusterEstimate* algorithm and its main procedures. Section 3 gives some insights of a customized version of the algorithm BSAS, to provide technical material for the understanding of its use, by the *ClusterEstimate* algorithm. The eight sets of patterns used in the experiments aimed at evaluating the proposal are presented in Sect. 4 and the experiments using the *ClusterEstimate* and results are presented and discussed. Section 6 resumes the work done.

2 Estimating the Number of Clusters of a Clustering by Using a Sequential Clustering Algorithm

This section focuses on a brief textual description, suggested in [6], for determining the number of clusters associated with a given set of patterns, which has been translated as the *ClusterEstimate* algorithm (pseudocode in Fig. 1).

```
procedure ClusterEstimate(X,S,P)
Input: X = {$P_1$, $P_2$,...,$P_N$} % dataset with N patterns
       S % number of times to run the BSAS to a specific $\Theta$
       P % iterative process step
begin
  min ← minimum d($P_i$,$P_j$)
  max ← maximum d($P_i$,$P_j$)
  NC ← {}   % vector of number of clusters
  for $\Theta$ = min to max step P
    begin
      NS ← {}  % number of clusters for the current $\Theta$
      for run =1 to S do
        begin
          NS ← NS ∪{BSAS_NC(X,$\Theta$,CQ)}
          Shake(X,X)
        end
      NC ←  NC ∪ mostFrequentValue(NS)
    end
  plot(NC)
end
```

Fig. 1. High level pseudocode of the *ClusterEstimate* algorithm.

The *ClusterEstimate* algorithm is based on the BSAS algorithm and employs the Euclidean distance as dissimilarity measure. Notice that the goal of the algorithm is to estimate the number of clusters that a clustering of a given set of patterns should have. With respect to the *ClusterEstimate* procedure, and considering X to be the set of patterns to be clustered, (1) variables *min* and *max* contain the lowest and highest dissimilarity values, respectively; (2) the value of the iterative process step should be informed by the user; (3) the value of the *S* parameter relates to the precision of the desired results; the greater the *S* value is, the greater is the precision of results; (4) procedure *BSAS_NC*(X, Θ, CQ) returns the number of clusters generated by the *BSAS_NC* algorithm, described in Fig. 2, using a dissimilarity threshold value of Θ; (5) the *Shake*(X,X) procedure randomly changes the order of patterns in the set X; (6) *mostFrequentValue*(NS) returns the numbers of clusters that are the most frequent, considering the *S* clusterings generated by *BSAS_NC,* with a determined threshold and (7) *plot*(NC) simply creates a plotting graph of the number of clusters *versus* threshold values – usually the plotting has a certain number of wide flat regions.

```
procedure BSAS_NC(X,Θ,CQ)
Input: X = {$P_1$, $P_2$,...,$P_N$} % dataset with N patterns
       Θ                 % dissimilarity threshold
Output: CQ  % Number of clusters
begin
CQ ← 1
$C_{CQ}$ ← {$P_1$}
for i=2 to N do
 find $G_k$: d($P_i$,$G_k$) = $\min_{1\le j\le CQ}$ d($P_i$,$G_j$)
 if d($P_i$,$G_k$) > Θ then
    CQ ← CQ +1
    $G_{CQ}$ ← {$P_i$}
  else
     $G_k$ ← $G_k$ ∪ {$P_i$}
    updateCentroid(Gk) %update the centroid of $G_k$
end
return CQ
end
```

Fig. 2. High level pseudocode of the *BSAS_NC* algorithm.

3 The BSAS Algorithm

BSAS is a clustering algorithm characterized as sequential, where data patterns are presented to algorithm only once or twice; the algorithm does not require a value for K. The BSAS has been employed in a few works such as [9, 10] and its customized version, *BSAS_NC* (*Basic Sequential Algorithmic Scheme for Number of Clusters)* is shown in Fig. 2.

Each new pattern from a given set of patterns is processed by *BSAS_NC* either by assigning it to an existing cluster or, then, by defining a new cluster containing it (a singleton so far), depending on the dissimilarity measure between the pattern and the centroids of the clusters already constructed. The order in which patterns are presented has an important role when the algorithm is constructing the clustering. Different presentation orders may result in totally different clusterings, in terms of the number of clusters as well as in the patterns that belong to each of them. The original *BSAS* has two parameters: (1) the dissimilarity threshold (Θ) and (2) the maximum number of clusters allowed (q). As *ClusterEstimate* uses BSAS to help determining the number of clusters in a given set of patterns, the original algorithm was modified and turned into the *BSAS_NC*, by removing the need for the q parameter. Unlike proper clustering algorithms, the main role of the *BSAS_NC* version is only to output the number of clusters induced by the algorithm. An important issue that affects the results of the algorithm is the choice of the dissimilarity threshold value (Θ). If the value is too low, unnecessary clusters will be created and, if the value is too high, only a small number of suitable clusters will be created.

4 Data Domains for the Experiments

To investigate the performance of *ClusterEstimate*, a collection of sets of two-dimensional synthetic patterns, with different shapes, densities and quantity of patterns was used. All the eight sets of patterns presented in Fig. 3 were generated based on some typical clustering situations. Particularly, the sets of patterns in Fig. 3(a)–(d) were created based on those used in the research described in [11], that had its focus on the so called *gestalt* clusters. The clustering detection approaches described in the research were motivated by the human perception of two-dimensional sets of patterns as separate groupings or *gestalts*, where the principle of grouping is proximity, as described in [12]. Based on the human perception, each set of patterns exhibited in Fig. 3(a) and (b), can be approached as two distinct clusters of points. Although the set of patterns in Fig. 3(a) and (b) show similarities, the two clusters in each of them have completely different shapes. The set of points in Fig. 3(c) involves

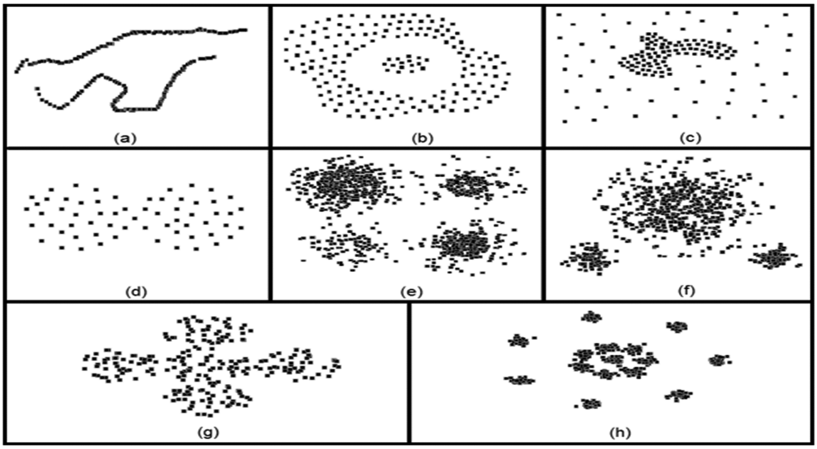

Fig. 3. Collection of eight sets of patterns used in the experiments, having as number of patterns: (a) 200, (b) 174; (c) 142; (d) 63; (e) 1,000; (f) 700; (g) 250 and (h) 600.

sharp gradient detection. Figure 3(d) shows one cluster with a small narrow section (*neck*), whose removal divides it into two distinct clusters. The set of patterns shown in Fig. 3(e), which consists of four clusters with different densities and slightly separated, was created inspired by the set "*Differentdensity*", used in the research work described in [13]. The set "*Skewdistribuition*", also described in [13], inspired the creation of the set of patterns in Fig. 3(f), which can be visually described as consisting of three clusters of patterns, where two of them are small and the third is large.

The set of patterns in Fig. 3(g) was created based on the "*AD_5_2*", used in the clustering experiments described in [14]; sets of patterns similar to (g) are typically used to evaluate algorithms with respect to the overlap areas that clusters may have. The patterns in the central area of (g) can be approached under two different perspectives (1) as a partial continuation of each of the well distinctive four clusters (each petal) or, then (2) as a cluster on its own. If (1) is assumed, the set of patterns has four clusters and if (2) is assumed, the set of patterns has 5 clusters. The set of patterns in Fig. 3(h) was created based on pattern set "*R15*", used in the research work described in [15]; its clusters are spherical and positioned so to compose two rings of clusters, that share a central cluster. The purpose of such disposition of patterns is to verify if clustering algorithms can identify the two clusters, each formed by a ring of small groups of patterns and, also, check how they behave towards the central group of patterns, central to both rings.

5 Experiments, Results and Analysis

The experiments presented in this section were conducted by visual inspection on the plotting graph which is one of the outputs of *ClusterEstimate*. Through visual inspection, the wider flat lines (parallel to the x-axis) in the plotting are compared; each flat line is associated with the number of runs (of *BSAS_NC*) where the same number of clusters has been maintained. The iterative step value (P) used in the experiments is calculated so to correspond to 100 threshold values; each threshold value is used 10 times (*i.e.*, S = 10) which, on average, corresponds to 1,000 executions of the *BSAS_NC* procedure. The methodology for conducting each experiment was defined by the sequence of four steps, namely: (1) establishing the iterative process step parameter using Eq. (1); (2) execute the *ClusterEstimate* with the two parameters (2.1) the iterative process step as defined by Eq. (1) and (2.2) ten executions for each threshold value; (3) visually select the five widest flat lines of the resulting plotting created by *ClusterEstimate*, ignoring those associated with one cluster and (4) visually checking if any of the five widest flat lines of the resulting plotting shows the number of clusters clearly identifiable. Consider X be a set of patterns to be clustered. Let MaxDiss and MinDiss be the largest and the smallest dissimilarity value between any two patterns of set X, respectively. Equation (1) gives the value of the iterative process step (P).

$$\mathrm{P} = (\mathrm{MaxDiss} + \mathrm{MinDiss})/100 \tag{1}$$

ClusterEstimate has the bias of returning, as result, a single cluster (considering that is the most frequent). This is because (1) the higher the threshold value, the lower will be the number of clusters that will be created and (2) the order in which the patterns are presented to *BSAS_NC* influences the clustering obtained by the algorithm. When the first pattern presented to *BSAS_NC* has dissimilarity values in relation to other patterns lower than the threshold value, only one cluster is returned. Due to that, this work ignores, in the plotting created by *ClusterEstimate*, those flat lines associated with a single cluster. For some sets of patterns, the *ClusterEstimate* output may result in several flat lines and, some of them may deserve to be inspected further, as other possibilities for alternative numbers of clusters. Therefore, as an empirical methodological decision, the five widest flat lines have been considered throughout all the experiments. In Fig. 4(a) the complete plotting graph of the overall results of *ClusterEstimate,* in the set of patterns of Fig. 3(a), using iterative process step = 0.022, is shown. Figure 4(b) shows an enlargement of the plotting shown in (a), focusing on the wider linear segments parallel to the x-axis.

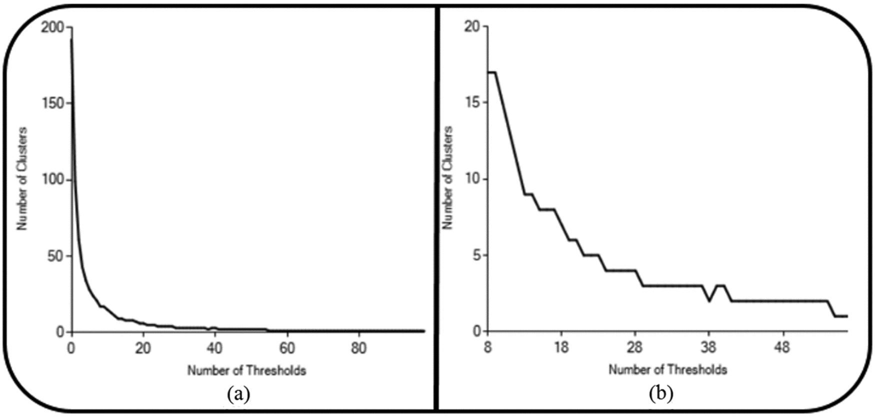

Fig. 4. Plotting graph generated by *ClusterEstimate*, having as input the set of patterns of Fig. 3(a). (a) the complete plotting. (b) enlargement of a sector of (a), focusing on its wider flat lines.

ClusterEstimate returned eight possible numbers of clusters, associated to the set of patterns in Fig. 3(a), as can be identified in Fig. 4(b), by the small linear segments parallel to the x-axis. The five first numbers of clusters, ordered by decreasing frequency are: (1) 2 clusters, (2) 3 clusters, (3) 4 clusters, (4) 5 clusters and (5) 8 clusters. *ClusterEstimate* found, as first option for the number of clusters, 2, which coincides with the number of clusters visually perceived in Fig. 3(a). Due to space restriction, all figures that follow will only show the enlarged plotting graph of the region of interest (wider linear segments parallel to the x-axis) extracted from the full plotting graph result produced by *ClusterEstimate*. Figure 5(a) shows only the enlarged plotting graph of the region of interest when using the set of patterns given in Fig. 3(b), having the iterative process step = 0.127. As can be checked in Fig. 5(a), *ClusterEstimate* found six possible numbers of clusters for the set of patterns in Fig. 3(b). The five first numbers of clusters, (concerning the length of their associated segments), ordered by decreasing frequency are: (1) 2 clusters, (2) 4 clusters, (3) 5 clusters, (4) 3 clusters and (5) 12 clusters. As can be visually verified in Fig. 3(b), *ClusterEstimate* found, as first element of its output list (*i.e.*, 2), the right number of clusters. Figure 5(b) shows only the enlarged plotting graph of the region of interest when using the set of patterns given in Fig. 3(c), considering the iterative process step = 0.212. As can be checked in Fig. 5(b), the *ClusterEstimate* algorithm also found six possible numbers of clusters for the set of patterns in Fig. 3(c). The five first numbers of clusters (concerning the length of their associated segments), ordered by decreasing frequency are: (1) 5 clusters, (2) 3 clusters, (3) 2 clusters, (4) 9 clusters and (5) 4 clusters. Although the number of clusters that can be visually detected in the given set of patterns is 3, this number appears in the third position in the list of the 5 most suitable numbers of clusters found by *ClusterEstimate*.

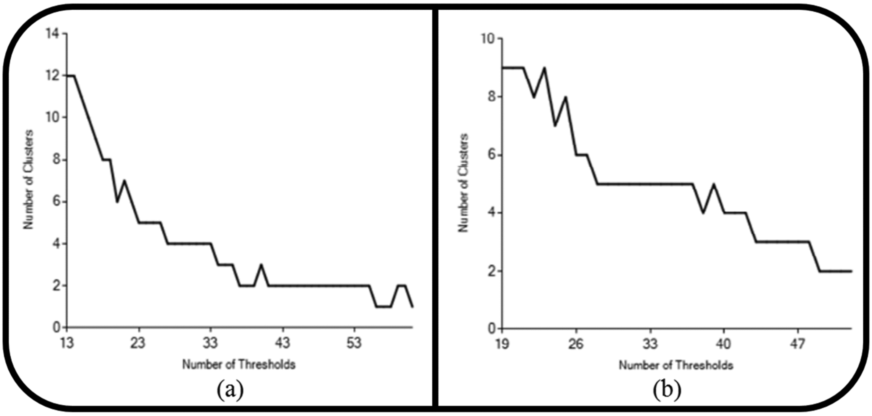

Fig. 5. (a) enlargement of the region of interest of the plotting graph obtained using Fig. 3(b) and (b) enlargement of the region of interest of the plotting graph obtained using Fig. 3(c).

Figure 6(a) shows only the enlarged plotting graph of the region of interest when using the set of patterns given in Fig. 3(d) and the iterative process step = 0.047. *ClusterEstimate* found six possible numbers of clusters associated to the set of patterns of Fig. 3(d), as can be identified in Fig. 6(a). The five first numbers of clusters ordered by decreasing frequency are: (1) 2 clusters, (2) 4 clusters, (3) 5 clusters, (4) 3 clusters and (5) 8 clusters. The first option in the list of numbers of clusters produced by *ClusterEstimate* is 2, which coincides with the number of clusters that can be visually identified in Fig. 3(d). Figure 6(b) shows only the enlarged plotting graph of the region of interest, when using the set of patterns given in Fig. 3(e) and iterative process step = 0.24. The *ClusterEstimate* procedure identified four possible numbers of clusters for the set of patterns shown in Fig. 3(e), which ordered by decreasing frequency are: (1) 4 clusters, (2) 2 clusters, (3) 3 clusters and (4) 5 clusters. As can be visually seen in Fig. 3(e), the *ClusterEstimate* procedure has delivered, as first option, the right number of clusters, i.e., 4.

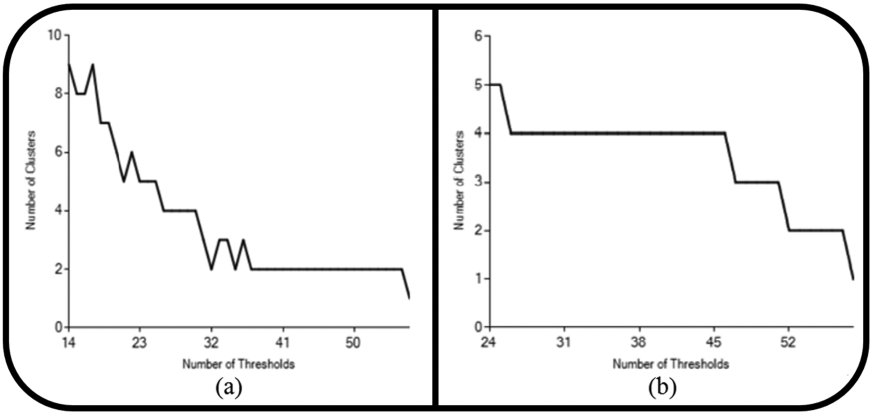

Fig. 6. (a) enlargement of the region of interest of the plotting graph obtained using Fig. 3(d) and (b) enlargement of the region of interest of the plotting graph obtained using Fig. 3(e).

Figure 7(a) shows only the plotting graph of the region of interest when using the set of patterns given in Fig. 3(f) and iterative process step = 0.08. *ClusterEstimate* found eleven possible numbers of clusters for the set of patterns in Fig. 3(f) and the five first numbers of clusters, ordered by decreasing frequency are: (1) 3 clusters, (2) 2 clusters, (3) 5 clusters, (4) 6 clusters and (5) 7 clusters. As can be confirmed via visual inspection of the set of patterns in Fig. 3(f), *ClusterEstimate* found, as its first option, the right number of clusters. Figure 7(b) shows only the enlarged plotting graph of the region of interest when using the set of patterns given in Fig. 3(g), having the iterative process step = 0.12. *ClusterEstimate* found eighteen possible numbers of clusters associated to the set of patterns in Fig. 3(g); the five first numbers of clusters ordered by decreasing frequency are: (1) 4 clusters, (2) 3 clusters, (3) 5 clusters, (4) 2 clusters and (5) 6 clusters. As discussed before, in Sect. 4, the set of patterns in Fig. 3(g) can be visually clustered into 4 or 5 clusters, depending on the approach used concerning the set of patterns 'shared' by the well-defined 4 clusters. Both options, though, are part of the 5-best options identified by *ClusterEstimate,* placed in the first and third positions, respectively, in such list. Figure 8 shows the enlarged plotting graph of the region of interest when using the set of patterns given in Fig. 3(h) and iterative process step = 0.14. *ClusterEstimate* found ten possible numbers of clusters associated with the set of patterns in Fig. 3(h); the five first numbers of clusters ordered by decreasing frequency are: (1) 8 clusters, (2) 3 clusters, (3) 2 clusters, (4) 10 clusters and (5) 15 clusters. This experiment is the one (out of 8) where the *ClusterEstimate* was not that entirely successful, because it placed the right number of clusters (15) in the fifth position (last) of its list. In five out of the eight experiments conducted *ClusterEstimate* obtained, as the first option in the list of possible numbers of clusters, the right number of visually identifiable clusters. *ClusterEstimate* has not returned, though, as the first option in its output list, the right number of visually identifiable clusters in the sets of patterns in Fig. 3(c), (g) and (h). All the three sets of patterns have clusters that "meet" each other, suggesting that the *ClusterEstimate* has its performance decreased when that happens. However, for the three sets of patterns, *ClusterEstimate* managed to deliver the right number of clusters among the first five of its list. As pointed out before, the results obtained with the set of patterns shown in Fig. 3(g), in particular, can be argued to have either four or five clusters, depending on the criteria used for visually identifying them.

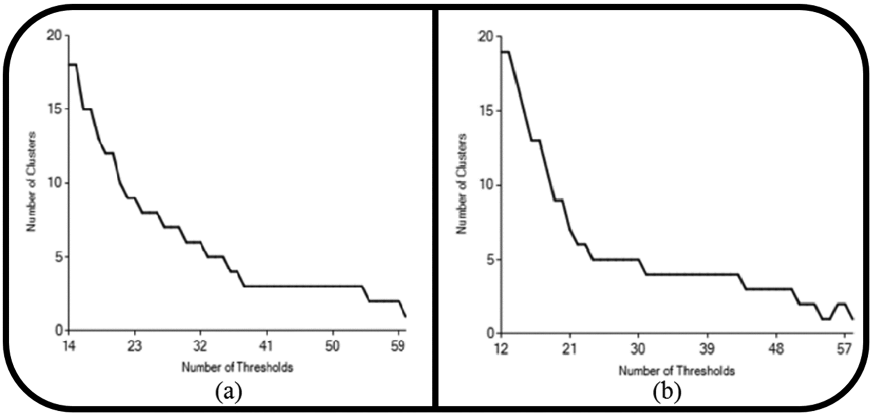

Fig. 7. (a) enlargement of the region of interest of the plotting graph obtained using Fig. 3(f) and (b) enlargement of the region of interest of the plotting graph obtained using Fig. 3(g).

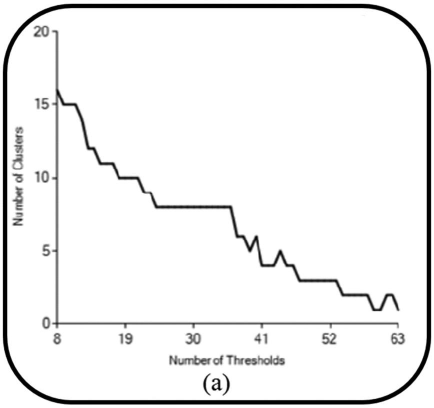

Fig. 8. Enlargement of the region of interest of the plotting graph obtained using Fig. 3(h).

6 Conclusions

This paper discusses and empirically evaluates the performance of an algorithm for estimating the number of clusters in a given set of patterns. The algorithm employs a variation of a sequential clustering algorithm, known as BSAS, which is highly dependent of the order the patterns are processed; in a way, this dependency was explored for detecting a possible number of clusters when, after various runs, the algorithm always produces a clustering with the same number of clusters. Experiments were conducted using eight synthetics set of patterns, several of them inspired by others available in the literature. The *ClusterEstimate* procedure has detected, as its first option, the same number of clusters as a human inspection would, in five out the eight sets. In three sets of patterns, however, although not listed as first option, the procedure still had the right number among its five-best results.

Acknowledgments. Authors thank CAPES, CNPq and FACCAMP.

References

1. Asano, T., Bhattacharya, B., Keil, M., Yao, F.: Clustering algorithms based on minimum and maximum spanning trees. In: Proceedings. of the Fourth Annual Symposium on Computational Geometry (SCG 1988), pp. 252–257 (1988)
2. Hartuv, E., Shamir, R.: A clustering algorithm based on graph connectivity. Inf. Process. Lett. **76**, 175–181 (2000)
3. Päivinen, N.: Clustering with minimum spanning tree of scale-free structure. Pattern Recogn. **26**, 921–930 (2005)
4. Luxburg, U.: A tutorial on spectral clustering. J. Stat. Comput. **17**, 395–416 (2007)
5. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Comput. Surv. **31**(3), 264–323 (1999)
6. Theodorides, S., Kotroumbas, K.: Pattern Recognition, 4th edn. Elsevier, USA (2009)
7. Berkhin, P.: A survey of clustering data mining techniques. In: Kogan, J., Nicholas, C., Teboulle, M. (eds.) Grouping Multidimensional Data, pp. 25–71. Springer, Heidleberg (2006)
8. Xu, R., Wunsch, D.: Survey of clustering algorithms. IEEE Trans. Neural Netw. **16**, 645–678 (2005)
9. Nicoletti, M.C., Real E.M., Oliveira, O.L.: The impact of refinement strategies on sequential clustering algorithms. In: Proceedings of the 13th International Conference on Intelligent Systems Design and Applications (ISDA 2013), pp. 47–52 (2013)
10. Real, E.M., Nicoletti, M.C., Oliveira, O.L.: A closer look into sequential clustering algorithms and associated post-processing refinement strategies. Int. J. Innov. Comput. Appl. **6**, 1–12 (2014)
11. Zahn, C.T.: Graph-theoretical methods for detecting and describing gestalt clusters. IEEE Trans. Comput. **C-20**, 68–86 (1971)
12. Wertheimer, M.: Principles of perceptual organization. In: Beardsley, D., Wertheimer, M. (eds.) Readings in Perception. Van Nostrand, Princeton (1958)
13. Liu, Y., Li, Z., Xiong, H., Gao X., Wu, J.: Understanding of internal clustering validation measures. In: Proceedings of the 10th International IEEE Conference on Data Mining (ICMD), pp. 911–916 (2010)
14. Bandyopadhyay, S., Saha, S.: Unsupervised Classification. Springer, Heidelberg (2013)
15. Veenman, C.J., Reinders, M.J.T., Backer, E.: A maximum variance cluster algorithm. IEEE Trans. Pattern Anal. Mach. Learn. **24**, 1273–1280 (2002)