Identifying Single Good Clusters in Data Sets

Frank Klawonn

Department of Computer Science
University of Applied Sciences Braunschweig/Wolfenbuettel
Salzdahlumer Str. 46/48
D-38302 Wolfenbuettel, Germany
f.klawonn@fh-wolfenbuettel.de
http://public.rz.fh-wolfenbuettel.de/~klawonn/

Abstract. Local patterns in the form of single clusters are of interest in various areas of data mining. However, since the intention of cluster analysis is a global partition of a data set into clusters, it is not suitable to identify single clusters in a large data set where the majority of the data can not be assigned to meaningful clusters. This paper presents a new objective function-based approach to identify a single good cluster in a data set making use of techniques known from prototype-based, noise and fuzzy clustering. The proposed method can either be applied in order to identify single clusters or to carry out a standard cluster analysis by finding clusters step by step and determining the number of clusters automatically in this way.

Key words: Cluster analysis, local pattern discovery

1 Introduction

Cluster analysis aims at partitioning a data set into clusters. It is usually assumed that, except for some noise data, most of the data can be assigned to clusters. However, when we are interested in detecting local patterns, standard clustering techniques are not suited for this task.

In various applications, cluster analysis is applied, although the focus is on detecting single interesting patterns, instead of partitioning the data set. For instance, cluster analysis is very often applied in the context of gene expression data in order to find groups (clusters) of genes with a similar expression pattern. The approach described in this paper was also motivated by an analysis of gene expression data where we applied standard clustering in the first step [4], but the main intention of the biologists was to find local patterns instead of a global partition into clusters. However, there are many other areas like the analysis of customer profiles where local patterns are of high interest.

A number of different approaches for the detection of local patterns have already been proposed and studied in the literature. For categorical data, numerous variants of the a priori algorithm for finding frequent item sets and association rules are very popular [8]. Scan statistics [7, 3] can be used to search

for local peaks in continuous data sets. However, due to the high computational costs, they are not suited for high-dimensional data and are very often applied in the context of geographical clusters, for instance places with an unusually high rate of a certain disease. In [9] a statistical approach is described that tries to circumvent the high computational costs of scan statistics by restricting the search space for the price of sub-optimal solutions. In this paper, we do not follow the more statistical idea of finding regions with high densities in the data space, but clusters that are more or less well separated from the rest of the data.

Fig. 1. An example data set.

Figure 1 shows an almost ideal example of a data set we consider here. It contains an almost well-separated cluster close to the top-left of the figure made of 200 data points, whereas the other 600 data points are scattered all over the data space and do not form meaningful clusters. Of course, figure 1 serves only illustration purposes, real data sets will have more than two dimensions.

The approach presented in this paper follows the concept of prototype-based cluster analysis, however, trying to find only one single cluster at a time. From the perspective of the single cluster, that we are trying to find in one step, data not belonging to this cluster is considered as noise. Therefore, we incorporate the

idea of noise clustering into our approach. Section 2 provides a brief overview on the necessary background of prototype and objective function-based clustering including noise clustering. In section 3 the new approach is introduced in detail. Short comments on application scenarios are provided in section 4, before we conclude the paper with a perspective on future work.

2 Prototype- and Objective Function-Based Clustering

In prototype-based clustering clusters are described by certain parameters that determine the prototype of the cluster. In the most simple case of c-means clustering, the prototype has the same form as a data object, assuming that clusters correspond more or less to (hyper-)spheres. Nevertheless, more flexible cluster shapes can also be covered by using more sophisticated prototypes. Cluster shapes might range from ellipsoidal shapes of varying size to non-solid clusters in the form of lines, hyperplanes or shells of circles and ellipses, the latter being more interesting in the area of image analysis. In this paper, we only mention c-means prototypes for our approach. However, our approach can be easily applied to any other cluster shape that is used in prototype-based clustering. For an overview on different cluster shapes and an introduction to objective function-based clustering we refer for instance to [5].

Once the form of the prototype is chosen, the idea of most prototype-based clustering techniques is to minimize the following objective function

$$f \;=\; \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{m} d_{ij} \tag{1}$$

under the constraints

$$\sum_{i=1}^{c} u_{ij} \;=\; 1 \qquad \text{for all } j = 1, \ldots, n. \tag{2}$$

It is assumed that the number of clusters c is fixed. We will not discuss the issue of determining the number of clusters here and refer for an overview to [1, 5]. The set of data to be clustered is $\{x_1, \ldots, x_n\} \subset \mathbb{R}^p$. d_{ij} is some distance measure specifying the distance between datum x_j and cluster i, for instance the (quadratic) Euclidean distance of x_j to the ith cluster centre in the case of c-means clustering. u_{ij} is the membership degree of datum x_j to the ith cluster. In the case of classical deterministic clustering, we require $u_{ij} \in \{0, 1\}$. However, here we will need the more general concept of fuzzy clustering and allow $u_{ij} \in [0, 1]$. The parameter $m > 1$, called fuzzifier, controls how much fuzzy clusters may overlap. The constraints (2) lead to the name probabilistic clustering, since in this case the membership degree u_{ij} can also be interpreted as the probability that x_j belongs to cluster i.

The parameters to be optimized are the membership degrees u_{ij} and the cluster parameters that are not given explicitly here. They are hidden in the

distances d_{ij}. Since this is a non-linear optimization problem, the most common approach to minimize the objective function (1) is to alternatingly optimize either the membership degrees or the cluster parameters while considering the other parameter set as fixed.

Davé [2] introduced the technique of noise clustering. Noise clustering uses the same objective function as (1). However, one of the clusters – the noise cluster – does not have any prototype parameters to be optimized. All data objects have a fixed (large) distance δ to this noise cluster. In this way, data objects that are far away from all other clusters are assigned to the noise cluster.

3 Identifying Single Clusters

As mentioned in the introduction, we are not interested in partitioning the data set, but in finding single clusters step by step. In order to find one single cluster, we adopt the idea of prototype-based clustering reviewed in the previous section.

We can simplify the notation, since we do not have to deal with c clusters, but only with two clusters: The proper cluster, we want to identify, and the noise cluster. We denote the membership degree of data object x_j to the cluster to be identified by u_j and its distance to this cluster by d_j. According to the constraint (2), the membership degree to the noise cluster is $1 - u_j$. The distance to the noise cluster is denoted by δ. We also choose $m = 2$ as the fuzzifier. This means that the objective function (1) including the constraints (2) simplifies to

$$f_1 = \sum_{j=1}^{n} u_j^2 d_j + (1 - u_j)^2 \delta^2. \tag{3}$$

The distance δ to the noise cluster influences the possible size of the single cluster, we want to identify. The larger the noise distance, the larger the single cluster can be. However, we are not able to specify δ a priori. In [6] an approach was proposed that also considers a single cluster together with a noise cluster. There, the noise distance δ is varied. Starting from a very large value δ is decreased in small steps until it reaches zero. While δ decreases, data objects are moved from the proper cluster to the noise cluster. The proper cluster is identified by analysing the fluctuation of the data from the proper cluster to the noise cluster. Although effective, this approach requires high computational costs for the repeated clustering while δ is decreasing. Also the analysis of the fluctuation of the data is not trivial.

In this paper, we try to adapt δ automatically during the clustering process. Therefore, we extend the objective function (3) by three further terms. We want our proper cluster to be well-separated from the remaining data, i.e. from the noise cluster. When the proper cluster is well separated from the noise cluster, membership degrees should tend to the values zero and one. There should be few data with intermediate values. Assuming $u_j \in [0, 1]$, the following term is maximal, if $u_j \in \{0, 1\}$ holds for all data objects j. It is minimal, if all u_j are

equal to 0.5.

$$f_2 = \sum_{j=1}^{n} u_j^2 + (1-u_j)^2 \tag{4}$$

It is also desirable, that our proper cluster is not empty and all data are assigned to the noise cluster. The term

$$f_3 = \sum_{j=1}^{n} u_j^2 \tag{5}$$

is maximised, when data objects are assigned to the proper cluster with high membership degrees.

Finally, we need an additional condition for the noise distance δ. Otherwise, if we could choose δ freely, minimizing (3) would automatically lead to $\delta = 0$. The fourth term

$$f_4 = \delta \tag{6}$$

should be maximised in order to favour larger values δ. A large δ also means that the proper cluster can be larger.

The objective function, we want to minimize for identifying the single cluster, is a linear combination of these four terms. Since only (3) should be minimized, whereas the other three should be maximised, we choose a negative coefficient for (4), (5) and (6). The overall objective function to be minimized is

$$f = \frac{a_1}{n} f_1 - \frac{a_2}{n} f_2 - \frac{a_3}{n} f_3 - a_4 f_4. \tag{7}$$

We have introduced the factor $\frac{1}{n}$ for the first three terms, in order to make the choice of the coefficients independent of the number of data. $\frac{1}{n} f_1$ is the weighted average distance, weighted by the membership degrees, of the data to the two clusters. $\frac{1}{n} f_2$ can be interpreted as an indicator of how well separated the proper cluster is from the remaining data. It can assume values between 0.5 and 1. $\frac{1}{n} f_3$ corresponds to the proportion of data in the proper cluster. The final term f_4 is already independent of the number of data.

The parameters in f to be optimized are

- the membership degrees $u_j \in [0,1]$ ($j \in \{1,\ldots,n\}$),
- the noise distance $\delta > 0$ and
- the cluster prototype parameters that are hidden in the distances d_j.

In order to apply the alternating optimization scheme, we have to find the optimal values for each set of parameters, while the other parameters are considered as fixed.

Taking the partial derivative of f with respect to u_j leads to

$$\frac{\partial f}{\partial u_j} = 2\frac{a_1}{n} u_j d_j^2 - 2\frac{a_1}{n}\delta^2 + 2\frac{a_1}{n} u_j \delta^2 - 2\frac{a_2}{n} u_j - 2\frac{a_2}{n} u_j + 2\frac{a_2}{n} - 2\frac{a_3}{n} u_j. \tag{8}$$

For a minimum, it is necessary that the partial derivative is zero. Setting (8) to zero, we obtain

$$u_j = \frac{a_1\delta^2 - a_2}{a_1 d_j^2 + a_1\delta^2 - 2a_2 - a_3}. \tag{9}$$

The partial derivative of f with respect to δ is

$$\frac{\partial f}{\partial \delta} = 2\frac{a_1}{n}\delta\sum_{j=1}^{n}(1-u_j)^2 - a_4,$$

leading to

$$\delta = \frac{a_4}{2a_1\frac{1}{n}\sum_{j=1}^{n}(1-u_j)^2}. \tag{10}$$

The cluster prototype parameters occur only in the distances d_j and therefore only in the term f_1 of the objective function. Therefore, the derivation for the cluster prototype parameters is the same as for standard fuzzy clustering. In the most simple case of a fuzzy c-means prototype, the prototype is a vector $v \in \mathbb{R}^p$ like the data objects. The corresponding equation for v is then

$$v = \frac{\sum_{j=1}^{n} u_j^2 x_j}{\sum_{j=1}^{n} u_j^2}. \tag{11}$$

The four coefficients $a_1, \ldots, a_4$ determine, how much influence the corresponding terms in the objective function have. Since only the proportions between these coefficients and not their absolute values play a role in the optimization, we can choose $a_1 = 1$ without loss of generality. Therefore, equations (9) and (10) can be simplified to

$$u_j = \frac{\delta^2 - a_2}{d_j^2 + \delta^2 - 2a_2 - a_3} \tag{12}$$

and

$$\delta = \frac{a_4}{2\frac{1}{n}\sum_{j=1}^{n}(1-u_j)^2}, \tag{13}$$

respectively.

The principal algorithm to find a single cluster is then as follows:

1. Choose $a_2, a_3, a_4 \geq 0$.
2. Choose $\varepsilon > 0$ for the stop criterion.
3. Initialise v and δ (randomly or as described in section 4).
4. Update the u_j's according to equation (12).
5. Update δ according to equation (13).

6. Update v according to equation (11) (or to the corresponding equation, if other than fuzzy c-mean prototypes are considered).

7. Repeat steps 4,5,6 until v is not changed significantly anymore, i.e. until $\| v^{\text{new}} - v^{\text{old}} \| < \varepsilon$.

In step 4 we have to make sure that $0 \leq u_j \leq 1$ holds. In order to satisfy this condition, we define a lower bound for the noise distance δ. When we want the denominator in (12) to be positive, even for small distances d_j or at least for distances about $d_j = a_3$, we have to require that $\delta^2 \geq 2a_2$, i.e. $\delta \geq \sqrt{2a_2}$ holds. Therefore, we define $\delta = \sqrt{2a_2}$ in case (13) yields a value smaller than $\sqrt{2a_2}$. For very small values of d_j this might still lead to a negative denominator in (12). It is obvious that we should choose $u_j = 1$ in these cases, i.e. assign the data object x_j fully to the very close proper cluster.

Recommendations for the choice of the parameters a_2, a_3, a_4 will be provided in the next section.

4 Application Scenarios

The main objective of identifying a single cluster is still to find the correct cluster prototype and to assign the corresponding data correctly to the cluster. Therefore, the most important term in the objective function (7) is f_1. Since we assume $a_1 = 1$, the other parameters should be chosen smaller than one. Our experiments with various data sets have shown that in most cases $a_4 \approx a_3 \approx 10a_2$ is a suitable relation between the coefficients. The crucial point is then the choice of the parameter a_2. Since this coefficient determines also the minimal noise distance, it should depend on the expected distances d_j in the data set. When we assume that the data set is normalised to the unit hyper-cube, the distance values d_j still depend on the dimension p of the data and $a_4 \approx (p \cdot 0.1)^2$ worked quite well.

In the example data set from figure 1 our algorithm is able to identify the cluster in the top left correctly, depending on the initialisation. As long as the initial cluster centre v is not too far away from the dense data cluster – the initial prototype does not have to be within the data cluster – the cluster will be identified correctly. However, when the initial prototype v is too far away from the cluster to be identified, the cluster might not be found. We cannot expect this, since our algorithm does not carry out any explicit scanning of the data set. Therefore, we recommend to carry out standard c-means clustering and use the resulting cluster centres as initialisations for our algorithm. The initial value for δ can then be based on the average distance of the data to the corresponding cluster. We have applied this technique to gene expression data and were able to identify clusters relevant from the biological point of view. Due to the limited space, we cannot discuss the details of this application here.

5 Conclusions

We have proposed an efficient approach to identify single clusters. Future work will focus on the influence of the choice of the coefficients a_2, a_3, a_4 to be chosen in our algorithm as well as on results using more complex cluster prototypes.

References

1. Bezdek, J.C., Keller, J., Krishnapuram, R., Pal, N.R.: Fuzzy Models and Algorithms for Pattern Recognition and Image Processing. Kluwer Academic Publishers, Boston, (1999)
2. Davé, R.N.: Characterization and Detection of Noise in Clustering. Pattern Recognition Letters **12** (1991) 657–664
3. Duczmal, L., Assunção, R.: A Simulated Annealing Strategy for the Detection of Arbitrarily Shaped Spatial Clusters. Computational Statistics & Data Analysis **45** (2004) 269–286
4. Georgieva, O., Klawonn, F., Härtig, E.: Fuzzy Clustering of Macroarray Data. In: Reusch, B. (ed.): Computational Intelligence, Theory and Applications. Springer-Verlag, Berlin (2005) 83–94
5. Höppner, F., Klawonn, F., Kruse, R., Runkler, T.: Fuzzy Cluster Analysis. Wiley, Chichester (1999)
6. Klawonn, F., Georgieva, O.: Identifying Single Clusters in Large Data Sets. In: Wang, J. (ed.): Encyclopedia of Data Warehousing and Mining. Idea Group, Hershey (2006) 582–585
7. Kulldorff, M.: A Spatial Scan Statistic. Communications in Statistics **26** (1997) 1481–1496
8. Zhang, C., Zhang, S.: Association Rule Mining. Springer-Verlag, Berlin (2002)
9. Zhang, Z., Hand, D.J.: Detecting Groups of Anomalously Similar Objects in Large Data Sets. In: Famili, A.F., Kook, J.N., Peña, J.M., Siebes, A. (eds.): Advances in Intelligent Data Analysis VI. Springer-Verlag, Berlin (2005) 509–519