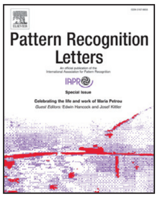
Pattern Recognition Letters 53 (2015) 31–37

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec



Is there any need for rough clustering?[☆]

Georg Peters[a,b,*]

[a] Munich University of Applied Sciences, Lothstr. 34, Munich 80335, Germany
[b] Australian Catholic University, Sydney, Australia

ARTICLE INFO

Article history:
Received 14 April 2014
Available online 22 November 2014

Keywords:
Rough clustering
Soft clustering
k-means

ABSTRACT

Clustering plays an important role in data mining. Some of the most famous clustering methods belong to the family of *k*-means algorithms. A decade ago, Lingras and West enriched the field of soft clustering by introducing rough *k*-means. Although rough clustering has been a very active field of research a pointed evaluation if it is really needed is still missing. Thus, the objective of the paper is to compare rough *k*-means and *k*-means. In *k*-means the number of correctly clustered objects is to be maximized which corresponds to minimizing the number of incorrectly clustered objects. In contrast to *k*-means, in rough clustering the numbers of correctly and incorrectly clustered objects are not complements anymore. Hence, in rough clustering the number of incorrectly clustered objects can be explicitly minimized. This is of striking relevance for many real life applications where minimizing the number of incorrectly clustered objects is more important than maximizing the number of correctly clustered objects. Therefore, we argue that rough *k*-means is often a strong alternative to *k*-means.

1. Introduction

Clustering is one of the most popular methods in data mining with countless real-life applications. Its objective is to group similar objects into the same cluster while dissimilar objects should belong to different clusters. A three-digit number of clustering approaches has probably been proposed to date. They include extensions and derivatives of existing ones but also approaches that established new branches of clustering algorithms. Categorizing such a huge number of approaches is a challenge in itself (see, e.g., Ref. [5, p. 56]) for possible classificatory schemes). Frequently used categorizations include hierarchical in contrast to partitive algorithms or hard as opposed to soft clustering. Within the group of partitive algorithms *k*-means [13] is probably the most famous and widely used. In our paper, we also refer to it as hard *k*-means. In the soft clustering domain fuzzy [1] and possibilistic [6] *c*-means play prominent roles.

Rough sets theory [19] has become an important part of soft computing. A decade ago, Lingras and West [12] proposed rough *k*-means. In the past ten years, rough clustering has been a very active field of research with many methodical enhancements and applications in a diverse range of domains (see Section 2 for a brief review). But with respect to a virtually unmanageable number of existing clustering algorithms, including ‘blockbusters’ like *k*-means, any new approach should be critically challenged: Does it provide anything novel that goes beyond pure academic interest?

Therefore, our objective is to investigate if there is any need for rough clustering. To obtain a manageable field of research we limit ourselves to the partitive clustering bioscope and spend special interest to hard and rough *k*-means. We show that the definition of a cluster by two approximations in rough clustering provides distinct advantages over hard *k*-means especially when penalties are imposed on incorrectly clustered objects.

The remainder of the paper is organized as follows. In the next section, we give a brief introduction to rough clustering. In Section 3, we discuss validation methods for hard and rough clustering. In Section 4, we perform comparative experiments showing the enriching interpretational possibilities of rough clustering in comparison to hard clustering.

2. Principles of rough clustering

2.1. Fundamental idea of rough clustering

Rough *k*-means was introduced by Lingras and West [12]. In contrast to original rough set theory that deals with categorical data rough *k*-means is derived from the interval interpretation of rough sets [12,30]. Like hard *k*-means it uses the distances between objects and means to determine the clusters.

Similar to original rough sets theory, a cluster C is defined by a lower approximation $\underline{C}$ and an upper approximation $\overline{C}$. The lower

[☆] This paper has been recommended for acceptance by S. Sarkar.

[*] Corresponding author at: Munich University of Applied Sciences, Lothstr. 34, 80335 Munich, Germany. Tel.: +49 89 1265 3709; fax: +49 89 1265 3780.

E-mail address: georg.peters@cs.hm.edu

http://dx.doi.org/10.1016/j.patrec.2014.11.003
0167-8655/© 2014 Elsevier B.V. All rights reserved.

approximation is a subset of the upper approximation: $\underline{C} \subseteq \overline{C}$. The region of the upper approximation that is not covered by the lower approximation is called boundary: $\widehat{C} = \overline{C} \setminus \underline{C}$. Objects assigned to the lower approximation surely belong to the cluster. Boundary objects also belong to one and only one cluster. However, due to missing or contradicting information, their membership cannot be decided. Obviously, a boundary object has to belong to two or more boundaries indicating its unclear membership status. As we will discuss further down boundaries function as 'buffer zones' for these objects; this often constitutes a crucial advantage of rough over hard clustering with respect to the number of incorrectly clustered objects.

In the mean time, many extensions of rough k-means have been proposed. They include some refinements [21], evolutionary versions (e.g., Ref. [15]), rough medoids [25] or dynamic approaches [27]. Hybrid clustering merging fuzzy, possibilistic and rough approaches include algorithms of Maji and Pal [14] or Mitra and Barmann [17]. Recently, Peters [22] proposed π rough k-means that uses the Principle of Indifference [7]. Areas of applications include bioinformatics [15,16], traffic control [8] and business [9]. For more on rough clustering see, e.g., Refs. [10,11]; for a survey on soft clustering, including an analysis of the relationship between hard, fuzzy and rough clustering, the reader is referred to Ref. [24].

2.2. Algorithmic structure

Algorithmically, rough k-means is closely related to hard k-means. The only difference is that rough clusters are defined by two approximations instead of one and only one crisp border as in hard k-means. This has the following two implications for rough clustering: (1) The means are derived from weighted sums of the objects depending on their memberships to the approximations. (2) The decision if an object is assigned to a lower approximation of a cluster or its boundary is based on a user-defined threshold parameter ζ.

In the further course of our paper, we apply π rough k-means [22]. The main advantage of π rough k-means in comparison to previous rough k-means algorithms is that it does not require any user-defined weights. The weights are derived from Laplace's Principle of Indifference [7]. For example, if an object is member of three boundaries its weight to each of the corresponding clusters equals the reciprocal of its number of memberships (in our example, its weight is $\frac{1}{3}$). Accordingly, an object in a lower approximation is weighted by $\frac{1}{1}$ since it belongs to one and only one cluster.

By applying Laplace's Principle of Indifference, the weights can be interpreted as probabilities in π rough k-means. This implies that all objects x_n can be treated identically, independently whether they are members of a lower approximation or a boundary. They only differ by their numbers of memberships $|T_n|$: $|\underline{T}_n| = 1$ for an object x_n in a lower approximation and $2 \leq |\widehat{T}_n| \leq K$ (K the number of clusters) for boundary objects. The range of the membership probabilities (weights) is: $\underline{w} = \frac{1}{1} = 1$ and $\frac{1}{K} \leq \widehat{w} \leq \frac{1}{2}$ (with $\underline{w} = \frac{1}{|\underline{T}_n|}$ and $\widehat{w} = \frac{1}{|\widehat{T}_n|}$).

Hence, we no longer need to differentiate between objects in lower approximations and boundaries. It is sufficient to consider all objects of a cluster (i.e., the objects in its upper approximation) and distinguish them by their membership probabilities $\frac{1}{|T_n|}$ only.[1]

For a set of N objects x_n $(n = 1, \ldots, N)$ π rough k-means proceeds as follows:[2]

Initialization

- Set the number of clusters $1 < K < N (k = 1, \ldots, K)$ and the threshold parameter $\zeta \geqslant 1$.
- Determine the initial means (e.g., randomly or maximum distance between means). Assign each object x_n to the corresponding upper approximation of its nearest mean.[3]

Iteration

- Compute the new means:

$$m_k = \frac{\sum_{x_n \in \overline{C_k}} \frac{x_n}{|T_n|}}{\sum_{x_n \in \overline{C_k}} \frac{1}{|T_n|}} \tag{1}$$

  with $|T_n|$: number of upper approximations x_n belongs to.
- Assign the objects to the approximations:
  - Determine the nearest mean of object x_n:

$$d_h^{\min} = d(x_n, m_h) = \min_{k=1,\ldots,K} d(x_n, m_k) \tag{2}$$

  - Determine similarly near means to x_n. Including the nearest mean to x_n we obtain:

$$T_n = \left\{ t : \frac{d(x_n, m_t)}{d_h^{\min}} \leq \zeta \right\} \tag{3}$$

  - Assign object x_n to the upper approximations:

$$x_n \in \overline{C_t}, \forall t \in T_n \tag{4}$$

- IF [current upper approximations unchanged to previous $\vee$
  maximum number of iterations reached]
  THEN [stop] ELSE [repeat iteration].

2.3. Discussion of selected properties

Upper approximations only. We would like to emphasize again that the algorithm requires only upper approximations. They fully define rough clusters. The corresponding lower approximations and boundaries can be easily derived out of them: $|T_n| = 1 \Leftrightarrow x_n \in \underline{C_t}$ and $2 \leq |T_n| \leq K \Leftrightarrow x_n \in \widehat{C_t}, \forall t \in T_n$.

Initial settings. In general, the initial parameters in rough k-means are: (1) the weights, $\underline{w}$ and $\widehat{w} = 1 - \underline{w}$, (2) the number of clusters K and (3) the threshold parameter ζ.

1. *Weights.* In π rough k-means the weights are determined by applying Laplace's Principle of Indifference. Hence, in contrast to previous rough k-means algorithms, they are not user-defined. So, we do not have to set them initially.
2. *Number of clusters.* Like in most k-means cluster algorithms (hard k-means, fuzzy c-means etc.) the setting of the number of clusters is of crucial importance and (unfortunately) still a big challenge. Due to its more recent introduction in comparison to k-means, fuzzy c-means and others, research in this field is relatively new in rough clustering. Applying and adapting some of the well-established methods of hard k-means and fuzzy c-means are reasonable ways to determine the number of clusters K.
3. *Threshold parameter.* The parameter ζ determines the size of the boundaries. For $\zeta \to 1$ rough k-means convergence towards hard-means, i.e., the boundaries become empty. For increasing ζ the number of boundary objects also increases. In practice, threshold parameters in the range of approximately $1.2 \leq \zeta \leq 1.8$ often deliver promising results. In some applications it has been observed

[1] In π rough k-means, like in fuzzy c-means, the weights specify the degree of membership of an object to a cluster: in π rough k-means they are based on probabilities and in fuzzy c-means they are derived from similarities. In contrast to this, weights imposed on the variables (derived from their importance) have also been proposed in clustering, e.g., by Huang et al. [4].

[2] For sake of simplicity, we present π rough k-means without the optional step that prevents (highly unlikely) divisions by 0 when calculating the means.

An implementation of π rough k-means in R [28] is available at *CRAIN* (package: *SoftClustering*).

[3] This step is virtually identical to hard k-means with the only difference that we need to define the approximation an object is assigned to (→upper approximation) instead of the assignment to 'just' a cluster as in hard k-means.