A unified approach for cluster-wise and general noise rejection approaches for k-means clustering

Seiki Ubukata

Graduate School of Engineering, Osaka Prefecture University, Sakai, Osaka, Japan

ABSTRACT

Hard C-means (HCM; k-means) is one of the most widely used partitive clustering techniques. However, HCM is strongly affected by noise objects and cannot represent cluster overlap. To reduce the influence of noise objects, objects distant from cluster centers are rejected in some noise rejection approaches including general noise rejection (GNR) and cluster-wise noise rejection (CNR). Generalized rough C-means (GRCM) can deal with positive, negative, and boundary belonging of object to clusters by reference to rough set theory. GRCM realizes cluster overlap by the linear function threshold-based object-cluster assignment. In this study, as a unified approach for GNR and CNR in HCM, we propose linear function threshold-based C-means (LiFTCM) by relaxing GRCM. We show that the linear function threshold-based assignment in LiFTCM includes GNR, CNR, and their combinations as well as rough assignment of GRCM. The classification boundary is visualized so that the characteristics of LiFTCM in various parameter settings are clarified. Numerical experiments demonstrate that the combinations of rough clustering or the combinations of GNR and CNR realized by LiFTCM yield satisfactory results.

INTRODUCTION

Clustering, which is an important task in data mining/machine learning, is a technique for automatically extracting group (cluster) structures from data without supervision. It is useful for analyzing large-scale unlabeled data. Hard C-means (HCM; k-means) (*MacQueen, 1967*) is one of the most widely used partitive clustering techniques. Real-world datasets often contain noise objects (outliers) with irregular features that may distort cluster shapes and deteriorate clustering performance. Since C-means-type methods are formulated based on the minimization of the total within-cluster sum-of-squared-error, they are strongly affected by noise objects, which are distant from cluster centers. We focus on two types of noise rejection, namely, general noise rejection (GNR) and cluster-wise noise rejection (CNR). In GNR approaches, whether each object is noise or not is defined in the whole cluster structure. Objects distant from any cluster center are rejected as noise. On the other hand, in CNR approaches, whether each object is noise or not is defined for each cluster. For each cluster, objects distant from its center are rejected as noise. Both

GNR and CNR perform noise rejection while GNR performs exclusive cluster assignment whereas CNR allows cluster overlap.

HCM assigns each object to one and only one cluster with membership in the Boolean (hard; crisp) domain $\{0, 1\}$, and thus it cannot represent belonging to multiple clusters or non-belonging to any cluster. However, in real-world datasets, belonging of object to clusters is often unclear. Soft computing approaches are useful to represent belonging to multiple clusters or non-belonging to any cluster. Clustering based on rough set theory (*Pawlak, 1982*; *Pawlak, 1991*) considers positive, negative, and boundary belonging of object to clusters. Lingras and West proposed rough C-means (LRCM) (*Lingras & West, 2004*) as a rough-set-based C-means clustering, and Peters proposed a refined version of RCM (PRCM) (*Peters, 2006*). Ubukata et al. proposed the generalized RCM (GRCM) (*Ubukata, Notsu & Honda, 2017*) by integrating LRCM and PRCM. GRCM realizes cluster overlap by a linear function threshold with respect to the distance to the nearest cluster and detects the upper area composed of objects that possibly belong to the cluster. Specifically, the threshold based on the distance to the nearest cluster center is lifted by the linear function to allow the cluster to be assigned to relatively near clusters as well as the nearest cluster.

In this study, we investigate the characteristics of the linear function threshold-based object-cluster assignment in GRCM. We show that the linear function threshold-based assignment in relaxed GRCM can realize GNR, CNR, and their combinations as well as rough assignments. One important point is that the linear function threshold-based assignment essentially includes GNR and CNR in compliance with RCM standards without any extra formulation. As a unified approach for GNR and CNR in HCM, we propose linear function threshold-based C-means (LiFTCM) by relaxing GRCM. The classification boundary is visualized so that the characteristics of LiFTCM in various parameter settings are clarified. Numerical experiments demonstrate that the combinations of rough clustering or the combinations of GNR and CNR realized by LiFTCM yield satisfactory results.

The remainder of the paper is organized as follows. In "Related Work," related works are discussed. "Preliminaries" presents the preliminaries for clustering methods. In "A unified approach for cluster-wise and general noise rejection approaches," we show that the linear function threshold-based assignment in relaxed GRCM can realize GNR, CNR, and their combinations as well as rough assignments. In "Proposed Method," we propose LiFTCM as one of the relaxed GRCM. In "Visualization of Classification Boundaries," the classification boundaries of LiFTCM with various parameter settings are considered. In "Numerical Experiments," the clustering performance of LiFTCM with various parameter settings is discussed. In "Discussion," the calculation of the cluster center in the proposed method are discussed. Finally, the conclusions are presented in "Conclusions."

RELATED WORK

Noise rejection in regression analysis and *C*-means-type clustering

Many machine learning tasks such as regression analysis are formulated in a framework of least mean squares (LMS) proposed by Legendre or Gauss (*Legendre, 1805*; *Gauss, 1809*),

which minimizes the sum of the squared residuals to fit models to a dataset. However, since the LMS criterion is strongly affected by noise objects and has the lack of robustness, various robust estimation methods have been proposed to reduce the influence of noise objects. Least absolute values (LAV) (*Edgeworth, 1887*) is a criterion that minimizes the sum of the absolute values of the residuals to reduce the influence of large residuals. M-estimator (*Huber, 1964*; *Huber, 1981*) is one of the most widely used robust estimators, which replaces the square function in LMS by a symmetric function with a unique minimum at zero that reduces the influence of large residuals. Least median of squares (LMedS) (*Hampel, 1975*; *Rousseeuw, 1984*) minimizes the median of the squared residuals. Least trimmed squares (LTS) (*Rousseeuw & Leroy, 1987*) minimizes the sum of the squared residuals up to h-th objects in ascending list of residuals.

Since C-means-type clustering methods are generally formulated based on the minimization of the within-cluster sum-of-squared-error, the above-mentioned robust estimation methods are promising approaches to noise in the cluster structure (*Kaufmann & Rousseeuw, 1987*; *Dubes & Jain, 1988*). In C-means-type clustering, the distance between object and its nearest cluster center is identified as the residual. Thus, in GNR, objects distant from any cluster center are rejected as noise. For instance, trimmed C-means (TCM; trimmed k-means, TKM) (*Cuesta-Albertos, Gordaliza & Matrán, 1997*; *Garcia-Escudero & Gordaliza, 1999*) introduces LTS criterion to HCM. TCM calculates the new cluster center by using objects up to h-th in ascending list of the distances to their nearest cluster centers. As a result, objects more than a certain distance away are rejected as noise. Noise rejection in C-means-type clustering is also well discussed in the context of fuzzy C-means (FCM) (*Dunn, 1973*; *Bezdek, 1981*). In noise fuzzy C-means (NFCM) (*Davé, 1991*; *Davé & Krishnapuram, 1997*), a single noise cluster is introduced in addition to the intended regular clusters and objects distant from any cluster center are absorbed in the noise cluster. Another approach to noise is CNR. For instance, possibilistic C-means (PCM) (*Krishnapuram & Keller, 1993*; *Krishnapuram & Keller, 1996*) considers cluster-wise noise rejection, in which each cluster is extracted independently while rejecting objects distant from its center. The membership values are interpreted as degrees of possibility of the object belonging to clusters. PCM represents typicality as absolute membership to clusters rather than relative membership by eliminating the sum-to-one constraint. Fuzzy possibilistic C-means (FPCM) (*Pal, Pal & Bezdek, 1997*) uses both relative typicalities (memberships) and absolute typicalities. Possibilistic fuzzy C-means (PFCM) (*Pal et al., 2005*) is a hybridization of FCM and PCM using both probabilistic memberships of FCM and possibilistic memberships of PCM.

In this study, we show that GNR, CNR, and their combinations are realized by the linear function threshold-based object-cluster assignment in the proposed LiFTCM. The above-mentioned approaches introduce various mechanisms to realize GNR and CNR. In contrast, the linear function threshold-based assignment essentially includes GNR, CNR, and their combinations in compliance with RCM standards without any extra formulation.

Generalized approaches to hard, fuzzy, noise, possibilistic, and rough clustering

Maji & Pal (2007a) proposed rough-fuzzy C-means (RFCM) as a hybrid algorithm of FCM and RCM. RFCM is formulated so that objects in the lower areas have crisp memberships and objects in the boundary areas have FCM-based fuzzy memberships. Furthermore, *Maji & Pal (2007b)* proposed rough-fuzzy possibilistic C-means (RFPCM) based on possibilistic fuzzy C-means (PFCM) (*Pal et al., 2005*). Masson and Denœux proposed evidential C-means (ECM) (*Masson & Denoeux, 2008*) as one of the evidential clustering (EVCLUS) (*Denoeux & Masson, 2003*; *Denoeux & Masson, 2004*) methods based on the Dempster-Shafer theory of belief functions (evidence theory). Evidential clustering considers the basic belief assignment, which indicates the membership (mass of belief) of each object to each subset of clusters with the probabilistic constraints that derive *credal* partition. Credal partition can represent hard and fuzzy partitions with a noise cluster considering assignments to a singleton and the empty set. Possibilistic and rough partitions are represented by using the plausibility function and the belief function (*Denoeux & Kanjanatarakul, 2016*).

Although RFCM and RFPCM provide interesting perspectives on the handling of the uncertainty in the boundary area, the object-cluster assignment is different from that of RCM and transform into different types of approach. Although the credal partition in ECM has high expressiveness including hard, noise, possibilistic, and rough clustering, the object-cluster assignment and cluster center calculation of ECM do not boil down to those of RCM. In contrast to the above-mentioned approaches, the formulation of the proposed LiFTCM is fully compliant with RCM standards. This study reveals that RCM itself inherently includes GNR, CNR, and their combinations as well as rough clustering aspects without any extra formulation.

PRELIMINARIES

Hard C-means and noise rejection

Let $U = \{x_1, \ldots, x_i, \ldots, x_n\}$ be a set of n objects, where each object $x_i = (x_{i1}, \ldots, x_{ij}, \ldots, x_{ip})^\top$ is a p-dimensional real feature vector. In C-means-type methods, C ($2 \leq C < n$) represents the number of clusters, and C clusters composed of similar objects are extracted from U. Each cluster has a prototypical point (cluster center), which is a p-dimensional vector $\boldsymbol{b}_c = (b_{c1}, \ldots, b_{cj}, \ldots, b_{cp})^\top$. Let u_{ci} be the degree of belonging of object i to cluster c. Let $d_{ci} = ||x_i - \boldsymbol{b}_c||$ be the distance between the cluster center $\boldsymbol{b}_c$ and the object i.

The optimization problem of HCM (*MacQueen, 1967*) is given by

$$\text{min.} \quad J_{HCM} = \sum_{c=1}^{C} \sum_{i=1}^{n} u_{ci} d_{ci}^2, \tag{1}$$

$$\text{s.t.} \quad u_{ci} \in \{0, 1\}, \forall c, i, \tag{2}$$

$$\sum_{c=1}^{C} u_{ci} = 1, \forall i. \tag{3}$$

HCM minimizes the total within-cluster sum-of-squared-error (Eq. (1)) under the Boolean domain constraints (Eq. (2)) and the sum-to-one constraints across clusters (Eq. (3)).

HCM first initializes cluster centers and then alternately updates u_{ci} and $\boldsymbol{b}_c$ until convergence by using the following update rules:

$$u_{ci} = \begin{cases} 1 & \left(c = \underset{1 \leq l \leq C}{\operatorname{argmin}} d_{li}\right), \\ 0 & (\text{otherwise}), \end{cases} \tag{4}$$

$$\boldsymbol{b}_c = \frac{\sum_{i=1}^{n} u_{ci}\boldsymbol{x}_i}{\sum_{i=1}^{n} u_{ci}}. \tag{5}$$

There are various strategies for initializing cluster centers. A naive strategy is to choose C objects as initial cluster centers from U by simple random sampling without replacement. Alternatively, there are strategies that set the initial cluster centers away from each other to reduce initial value dependencies and improve clustering performance, such as KKZ (*Katsavounidis, Kuo & Zhang, 1994*) and k-means++ (*Arthur & Vassilvitskii, 2007*).

General noise rejection (GNR)
Since HCM is formulated based on the LMS criterion, it is strongly affected by noise objects. Like TCM, which introduces the LTS criterion, the influence of noise objects can be reduced by rejecting objects distant from any cluster. In this type of GNR, each object is assigned to the nearest cluster under the condition that the distance d_{ci} is less than or equal to a threshold (noise distance) $\delta(\delta > 0)$:

$$u_{ci} = \begin{cases} 1 & \left(c = \underset{1 \leq l \leq C}{\operatorname{argmin}} d_{li} \wedge d_{ci} \leq \delta\right), \\ 0 & (\text{otherwise}). \end{cases} \tag{6}$$

The smaller δ is, the more objects are rejected as noise. The noise distance δ can depend on how many (what percentage of) objects to reject as noise.

### Cluster-wise noise rejection (CNR)
GNR is based on HCM-based exclusive assignment and cannot represent cluster overlap. By performing noise rejection independently for each cluster, possibilistic aspects that present non-belonging to any cluster and belonging to multiple clusters are achieved. In this type of CNR, noise rejection is performed for each cluster by rejecting objects over δ_c distant from its center:

$$u_{ci} = \begin{cases} 1 & (d_{ci} \leq \delta_c), \\ 0 & (\text{otherwise}). \end{cases} \tag{7}$$

The smaller δ_c is, the more objects are rejected as noise for each cluster c. The cluster-wise noise distance δ_c can depend on how many (what percentage of) objects to reject as noise for each cluster.

Generalized rough C-means
In RCM-type methods, which are rough set clustering schemes, membership in the lower, upper, and boundary areas of each cluster represents positive, possible, and uncertain
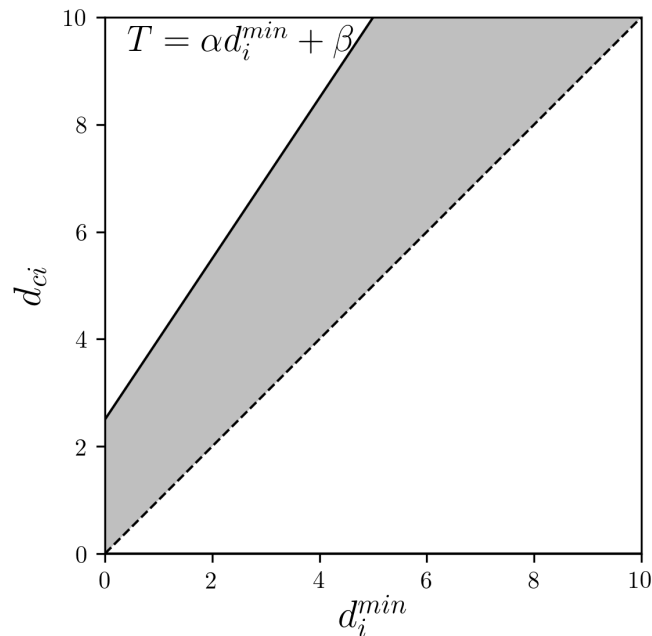
Figure 1 GRCM: the linear function threshold T and the allowable range of d_{ci} (gray area).

Full-size ⬛ DOI: 10.7717/peerjcs.238/fig-1

belonging to the cluster, respectively (*Lingras & West, 2004*; *Peters, 2006*; *Peters et al., 2013*; *Ubukata, Notsu & Honda, 2017*). GRCM is constructed based on a heuristic scheme, not an objective function.

In every iteration, the membership $\overline{u}_{ci}$ of object i to the upper area of cluster c is first calculated as follows:

$$d_i^{\min} = \min_{1 \leq l \leq C} d_{li}, \tag{8}$$

$$\overline{u}_{ci} = \begin{cases} 1 & (d_{ci} \leq \alpha d_i^{\min} + \beta), \\ 0 & (\text{otherwise}), \end{cases} \tag{9}$$

where α ($\alpha \geq 1$) and β ($\beta \geq 0$) are user-defined parameters that adjust the volume of the upper areas. GRCM assigns each object to the upper area of not only its nearest cluster but also of other relatively nearby clusters using a linear function of the distance to its nearest cluster as a threshold. Larger α and β imply larger clustering roughness and larger overlap of the upper areas of the clusters. Figure 1 shows the linear function threshold T and the allowable range of d_{ci} (gray area) in GRCM.

The membership $\underline{u}_{ci}$ and $\hat{u}_{ci}$ of object i to the lower and boundary areas, respectively, is calculated using $\overline{u}_{ci}$ as follows:

$$\underline{u}_{ci} = \begin{cases} 1 & \left(\overline{u}_{ci} = 1 \wedge \sum_{l=1}^{C} \overline{u}_{li} = 1 \right), \\ 0 & (\text{otherwise}), \end{cases} \tag{10}$$

Ubukata (2019), *PeerJ Comput. Sci.*, DOI 10.7717/peerj-cs.238

6/20

$$\hat{u}_{ci} = \begin{cases} 1 & \left(\overline{u}_{ci} = 1 \wedge \sum_{l=1}^{C} \overline{u}_{li} \neq 1 \right), \\ 0 & \text{(otherwise)} \end{cases} \tag{11}$$

$$= \overline{u}_{ci} - \underline{u}_{ci}. \tag{12}$$

GRCM represents each cluster by the three regions. Therefore, the new cluster center is determined by the aggregation of the centers of these regions. The cluster center $\boldsymbol{b}_c$ is calculated by the convex combination of the centers of the lower, upper, and boundary areas of the cluster c:

$$\boldsymbol{b}_c = \begin{cases} \dfrac{\sum_{i=1}^{n} \overline{u}_{ci} \boldsymbol{x}_i}{\sum_{i=1}^{n} \overline{u}_{ci}} & \left(\sum_{i=1}^{n} \underline{u}_{ci} = 0 \vee \sum_{i=1}^{n} \hat{u}_{ci} = 0 \right), \\ \underline{w} \dfrac{\sum_{i=1}^{n} \underline{u}_{ci} \boldsymbol{x}_i}{\sum_{i=1}^{n} \underline{u}_{ci}} + \overline{w} \dfrac{\sum_{i=1}^{n} \overline{u}_{ci} \boldsymbol{x}_i}{\sum_{i=1}^{n} \overline{u}_{ci}} + \hat{w} \dfrac{\sum_{i=1}^{n} \hat{u}_{ci} \boldsymbol{x}_i}{\sum_{i=1}^{n} \hat{u}_{ci}} & \text{(otherwise),} \end{cases} \tag{13}$$

$$\underline{w}, \overline{w}, \hat{w} \geq 0, \tag{14}$$

$$\underline{w} + \overline{w} + \hat{w} = 1, \tag{15}$$

where $\underline{w}$, $\overline{w}$, and $\hat{w}$ are user-defined parameters that represent the impact of the centers of the lower, upper, and boundary areas, respectively. *Ubukata, Notsu & Honda (2017)* suggest $\hat{w} = 0$ because the centers of the boundary areas tend to cause instability in the calculations and poor classification performance.

A UNIFIED APPROACH FOR CLUSTER-WISE AND GENERAL NOISE REJECTION APPROACHES

In this section, we show that GNR, CNR, and their combinations are realized by the linear function threshold in relaxed GRCM. Here, we consider relaxing the condition $\alpha \geq 1$ to $\alpha \geq 0$ in Eq. (9).

HCM

In HCM, each object is assigned to the cluster whose center is nearest to the object. This assignment can be interpreted as assigning object i to cluster c if d_{ci} is equal to (or less than) $d_i^{\min}$, that is,

$$u_{ci} = \begin{cases} 1 & (d_{ci} \leq d_i^{\min}), \\ 0 & \text{(otherwise)}. \end{cases} \tag{16}$$

This is the case $\alpha = 1$ and $\beta = 0$ in the linear function threshold $\alpha d_i^{\min} + \beta$ for the assignment of upper area in GRCM (Eq. (9)). Figure 2A shows the linear function threshold T and the allowable range of d_{ci} in HCM. The allowable range is limited to the case $d_{ci} = d_i^{\min}$. We note that if there are multiple nearest cluster centers for an object, HCM requires certain tie-breaking rules for satisfying the sum-to-one constraints, such as exclusive assignment based on cluster priority and uniform assignment by distributing the membership, depending on the implementation. However, in the present linear function threshold-based assignment, an object has membership 1 with respect to all its nearest clusters.

The calculation of u_{ci} in HCM can be represented by that of $\overline{u}_{ci}$ in GRCM. The lower and boundary areas are not used in HCM. Thus, the cluster center calculation of HCM
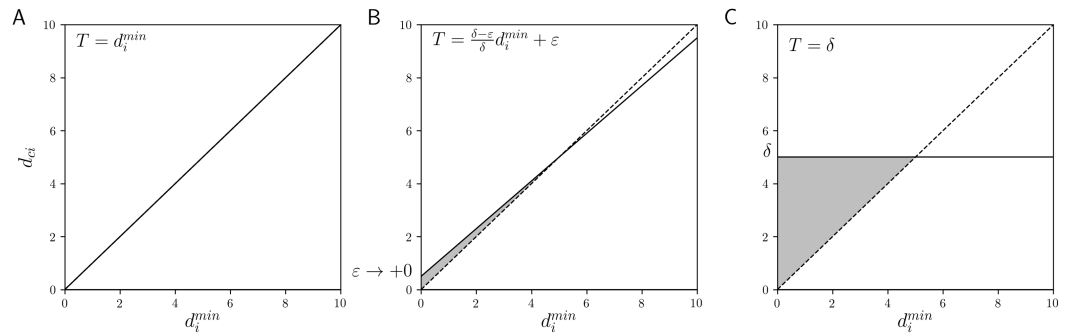
Figure 2 The linear function threshold T and the allowable range of d_{ci} (gray area): (A) HCM, (B) GNR, and (C) CNR.

is consistent with that of GRCM only using the upper areas, that is, $\overline{w} = 1$ in Eq. (13). Therefore, GRCM($\alpha = 1, \beta = 0, \overline{w} = 1$) represents HCM.

GNR

In GNR, a condition that the distance is less than δ is imposed in addition to the threshold-based HCM assignment (Eq. (16)) to reject noise objects over δ distant from any clusters. For each object i to be assigned to the cluster c, d_{ci} must be equal to (or less than) $d_i^{\min}$, and equal to or less than the noise distance δ, that is,

$$u_{ci} = \begin{cases} 1 & (d_{ci} \leq d_i^{\min} \wedge d_{ci} \leq \delta), \\ 0 & (\text{otherwise}). \end{cases} \tag{17}$$

This assignment can also be approximated using the linear function threshold by setting $\alpha = \frac{\delta - \varepsilon}{\delta}$ and $\beta = \varepsilon$, where $\varepsilon \to +0$, that is,

$$u_{ci} = \begin{cases} 1 & \left(d_{ci} \leq \frac{\delta - \varepsilon}{\delta} d_i^{\min} + \varepsilon\right), \\ 0 & (\text{otherwise}). \end{cases} \tag{18}$$

Equation (18) implies that $u_{ci} = 1$ if $d_{ci} \leq d_i^{\min}$ and $d_{ci} \leq \delta$. Thus, Eq. (18) approaches the update rule Eq. (17). In order to show that Eqs. (17) and (18) are equivalent, we show that the condition $d_{ci} \leq d_i^{\min} \wedge d_{ci} \leq \delta$ and the condition $d_{ci} \leq \frac{\delta - \varepsilon}{\delta} d_i^{\min} + \varepsilon$ are equivalent, under the condition $\delta > 0$ and $\varepsilon \to +0$.

Proposition 1 *If* $d_{ci} \leq d_i^{\min} \wedge d_{ci} \leq \delta$, *then* $d_{ci} \leq \frac{\delta - \varepsilon}{\delta} d_i^{\min} + \varepsilon$.

proof.
(1) $d_{ci} \leq d_i^{\min} \wedge d_{ci} \leq \delta$ (Assumption)
(2) $d_{ci} \leq d_i^{\min}$ (Conjunction elimination: (1))
(3) $d_{ci} \leq \delta$ (Conjunction elimination: (1))
(4) $d_i^{\min} \leq d_{ci}$ (Definition: (Eq. (8)))
(5) $d_i^{\min} \leq \delta$ (Transitivity: (3), (4))
(6) $\frac{\varepsilon}{\delta} d_i^{\min} + \frac{\delta - \varepsilon}{\delta} d_i^{\min} \leq \frac{\varepsilon}{\delta} \delta + \frac{\delta - \varepsilon}{\delta} d_i^{\min}$ (Multiply by $\frac{\varepsilon}{\delta}$ and add $\frac{\delta - \varepsilon}{\delta} d_i^{\min}$ in both sides of (5))

(7) $d_i^{\min} \leq \frac{\delta-\varepsilon}{\delta} d_i^{\min} + \varepsilon$ (Deformation: (6))

(8) $d_{ci} \leq \frac{\delta-\varepsilon}{\delta} d_i^{\min} + \varepsilon$ (Transitivity: (2), (7)) □

Proposition 2 *If* $d_{ci} \leq \frac{\delta-\varepsilon}{\delta} d_i^{\min} + \varepsilon$, *then* $d_{ci} \leq d_i^{\min} \wedge d_{ci} \leq \delta$, *under the condition that* ε *is sufficiently small.*

proof.

(1) $d_{ci} \leq \frac{\delta-\varepsilon}{\delta} d_i^{\min} + \varepsilon$ (Assumption)

(2) $d_{ci} \leq d_i^{\min}$ (From (1) and $\varepsilon \to +0$)

(3) $d_i^{\min} \leq d_{ci}$ (Definition: (Eq. (8)))

(4) $d_{ci} \leq \frac{\delta-\varepsilon}{\delta} d_{ci} + \varepsilon$ (From (1), (3))

(5) $\delta d_{ci} \leq \delta d_{ci} - \varepsilon d_{ci} + \delta \varepsilon$ (Multiply by δ in both sides of (4))

(6) $d_{ci} \leq \delta$ (Deformation: (5))

(7) $d_{ci} \leq d_i^{\min} \wedge d_{ci} \leq \delta$ (Conjunction introduction: (2), (6)) □

Hence, (Eq. (17)) induces (Eq. (18)), and vice versa.

Figure 2B shows the linear function threshold T and the allowable range of d_{ci} (gray area) in GNR. Since the intersection of the two lines $y = \frac{\delta-\varepsilon}{\delta} d_i^{\min} + \varepsilon$ and $y = d_i^{\min}$ is (δ, δ), if $d_{ci} > \delta$, object i is never assigned to cluster c. If $d_i^{\min} \leq \delta$, the threshold approaches the HCM-based nearest assignment. These characteristics are consistent with those of GNR.

Similar to HCM, in GNR, the cluster centers are calculated only using the upper areas. Therefore, GRCM$(\alpha = \frac{\delta-\varepsilon}{\delta}, \beta = \varepsilon, \overline{w} = 1)$ represents GNR.

CNR

The object-cluster assignment of CNR is determined only by the magnitude relation between d_{ci} and δ_c without considering $d_i^{\min}$. We note that the case $\alpha = 0$ and $\beta = \delta_c$ in Eq. (9) corresponds to the update rule Eq. (7) of CNR. Figure 2C shows the linear function threshold T and the allowable range of d_{ci} (gray area) in CNR. Independent of $d_i^{\min}$, if $d_{ci} \leq \delta$, object i is assigned to cluster c.

Similar to HCM and GNR, in CNR, the cluster centers are calculated only using the upper areas. Therefore, GRCM$(\alpha = 0, \beta = \delta_c, \overline{w} = 1)$ represents CNR.

Smooth transition between GNR and CNR by tuning linear function threshold

In reference to the threshold-based assignment of GNR, i.e., Eq. (18), we construct the following rule using a parameter $t \in [0, \delta_c]$:

$$u_{ci} = \begin{cases} 1 & (d_{ci} \leq \frac{\delta_c - t}{\delta_c} d_i^{\min} + t), \\ 0 & (\text{otherwise}). \end{cases} \tag{19}$$

If $t = 0$, then Eq. (19) reduces to Eq. (16) of HCM. If $t = \varepsilon$, where $\varepsilon \to +0$, then Eq. (19) changes to Eq. (18) of GNR. If $t = \delta_c$, then Eq. (19) reduces to Eq. (7) of CNR. If $t \in (0, \delta_c)$, then Eq. (19) represents the combinations of GNR and CNR. Thereby, smooth transition between HCM, GNR, and CNR is realized.

Ubukata (2019), *PeerJ Comput. Sci.*, DOI 10.7717/peerj-cs.238

9/20
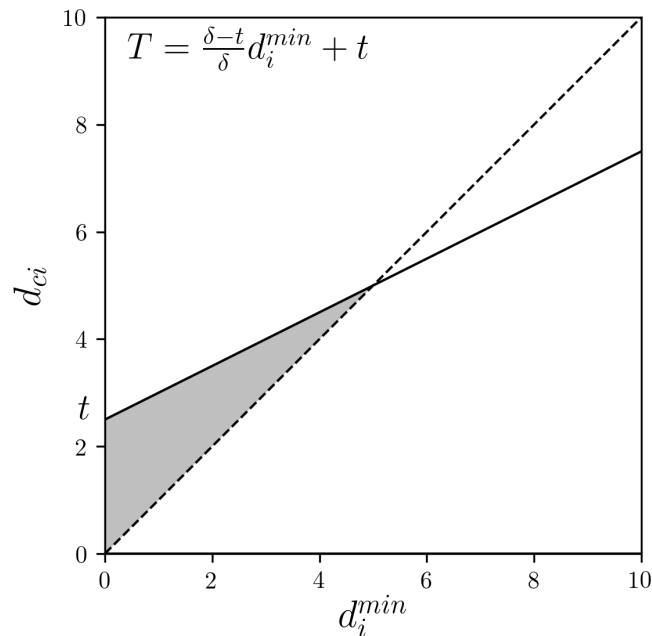
Figure 3 Combination of GNR and CNR: the linear function threshold T and the allowable range of d_{ci} (gray area).

Figure 3 shows the linear function threshold T and the allowable range of d_{ci} (gray area) in the combinations of GNR and CNR. It can be seen that this linear function can transition between the states shown in Fig. 2 by t.

For practical use, we consider the normalized parameter $z \in [0,1]$. We let $z = \frac{t}{\delta_c} \in [0,1]$ and replace t in Eq. (19) with $z\delta_c$:

$$u_{ci} = \begin{cases} 1 & (d_{ci} \leq (1-z)d_i^{\min} + z\delta_c), \\ 0 & (\text{otherwise}). \end{cases} \quad (20)$$

Then, $z = 0$ represents HCM, $z \to +0$ represents GNR, $z \in (0,1)$ represents the combinations of GNR and CNR, and $z = 1$ represents CNR. By Eq. (20), the threshold value is represented by the convex combination of $d_i^{\min}$ and δ_c. That is, HCM, GNR, and CNR can be characterized depending on which of $d_i^{\min}$ and δ_c is emphasized as the threshold value.

PROPOSED METHOD

In this study, we propose LiFTCM as one of the relaxed GRCM. "LiFT" is an acronym that stands for "linear function threshold" and suggests that the threshold is lifted by the linear function.

A sample procedure of LiFTCM is described in algorithm 1.

Although this algorithm just corresponds to the case where the condition $\alpha \geq 1$ in GRCM is relaxed to $\alpha \geq 0$, LiFTCM can represent GNR, CNR, and their combinations in addition to GRCM. If $0 \leq \alpha \leq 1$, LiFTCM includes HCM, GNR, CNR, and their

---

Algorithm 1 LiFTCM

Step 1. Determine α $(\alpha \geq 0)$, β $(\beta \geq 0)$, and $\underline{w}, \overline{w}, \hat{w} \geq 0$ such that $\underline{w} + \overline{w} + \hat{w} = 1$.

Step 2. Initialize $\boldsymbol{b}_c$.

Step 3. Calculate $\overline{u}_{ci}$ using Eqs. (8) and (9).

Step 4. Calculate $\underline{u}_{ci}$ and $\hat{u}_{ci}$ using Eqs. (10) and (11).

Step 5. Calculate $\boldsymbol{b}_c$ using Eq. (13).

Step 6. Repeat **Steps 3-5** until $\overline{u}_{ci}$ do not change.

Table 1 Relationship between HCM, GNR, CNR, and rough clustering, and their combinations in terms of the linear function threshold in LiFTCM.

Linear function Threshold: $\alpha d_i^{\min} + \beta$	$\beta = 0$	$\beta \to +0$	$0 < \beta$
$\alpha = 0$	–	–	CNR
$0 < \alpha < 1$	–	GNR	Combinations of GNR and CNR
$\alpha = 1$	HCM	HCM	LRCM
$1 < \alpha$	PRCM	PRCM	Combinations of LRCM and PRCM (GRCM)

combinations. If $\alpha \geq 1$, LiFTCM includes HCM, LRCM, PRCM, and their combinations. Table 1 summarizes the relationships between HCM, GNR, CNR, and rough clustering, and their combinations depending on the values of the parameters α and β in LiFTCM.

As it is difficult to adjust noise sensitivity by directly changing α and β when noise rejection is considered in LiFTCM, it is convenient to fix the cluster-wise noise distance δ_c and adjust the combination of HCM, GNR, and CNR by the parameter $z \in [0, 1]$ with $\alpha = (1 - z)$ and $\beta = z\delta_c$.

The representations of the conventional methods by setting the parameters of LiFTCM are summarized as follows:

1. HCM: LiFTCM$(\alpha = 1, \beta = 0, \overline{w} = 1)$.
2. LRCM: LiFTCM$(\alpha = 1, \beta \geq 0, \overline{w} = 0)$.
3. PRCM: LiFTCM$(\alpha \geq 1, \beta = 0, \hat{w} = 0)$.
4. GRCM: LiFTCM$(\alpha \geq 1, \beta \geq 0)$.
5. GNR: LiFTCM$(\alpha = 1 - z, \beta = z\delta_c, z \to +0, \overline{w} = 1)$.
6. CNR: LiFTCM$(\alpha = 0, \beta = \delta_c, \overline{w} = 1)$.
7. Combinations of GNR and CNR: LiFTCM$(\alpha = 1 - z, \beta = z\delta_c, z \in [0, 1], \overline{w} = 1)$.

VISUALIZATION OF CLASSIFICATION BOUNDARIES

In this section, we visualize the classification boundaries of the proposed LiFTCM. LiFTCM was applied to a grid point dataset, in which $n = 100 \times 100$ objects are uniformly arranged in the unit square $[0, 1] \times [0, 1]$. $C = 3$ clusters $(c = 1, 2, 3)$, which correspond to the primary colors $(Red, Green, Blue)$, respectively, are extracted by LiFTCM. The RGB-color of object i is determined by $(R, G, B)_i = (255 \times \overline{u}_{1i}, 255 \times \overline{u}_{2i}, 255 \times \overline{u}_{3i})$. Objects belonging to a single cluster are represented by primary colors, objects belonging to multiple clusters are represented by additive colors, and objects not belonging to any cluster are represented

Ubukata (2019), *PeerJ Comput. Sci.*, DOI 10.7717/peerj-cs.238

11/20
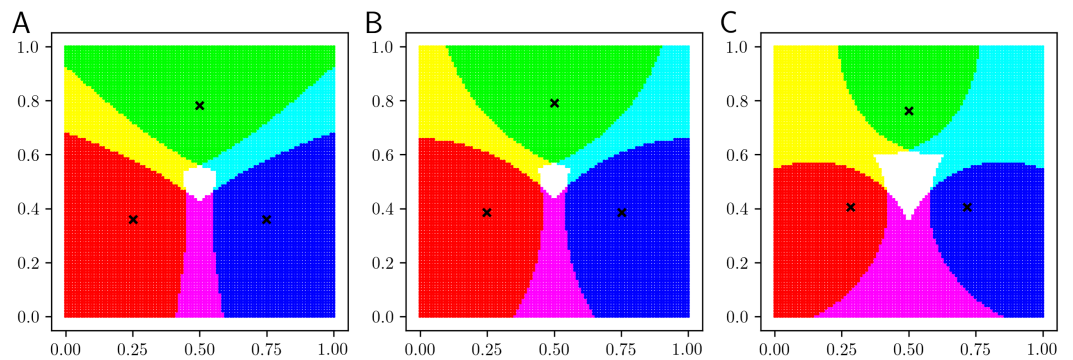
Figure 4 Classification boundaries of LiFTCM($\alpha \geq 1$, $\beta \geq 0$, $\overline{w} = 1$) representing LRCM, PRCM, and GRCM assignments: (A) LiFTCM($\alpha = 1$, $\beta = 0.1$, $\overline{w} = 1$) (LRCM assignment), (B) LiFTCM($\alpha = 1.4$, $\beta = 0$, $\overline{w} = 1$) (PRCM assignment), and (C) LiFTCM($\alpha = 1.4$, $\beta = 0.1$, $\overline{w} = 1$) (GRCM assignment).

Full-size 🖼 DOI: 10.7717/peerjcs.238/fig-4

by black color. The cluster centers are indicated by cross marks. Initial cluster centers were determined by $\boldsymbol{b}_1 = (0, 0)^\top$, $\boldsymbol{b}_2 = (0.5, 1)^\top$, and $\boldsymbol{b}_3 = (1, 0)^\top$.
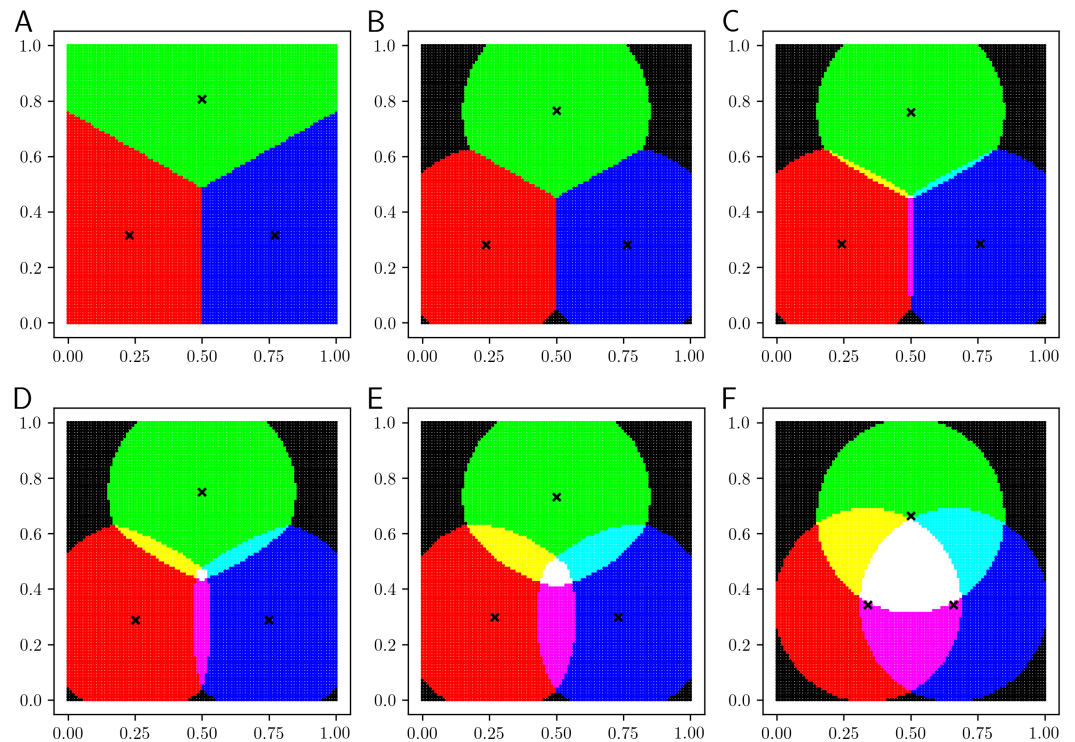
Figure 4 shows the results of LiFTCM($\alpha \geq 1$, $\beta \geq 0$, $\overline{w} = 1$), which corresponds to GRCM($\overline{w} = 1$). Figure 4A shows the result of LiFTCM($\alpha = 1$, $\beta = 0.1$, $\overline{w} = 1$), which is interpreted as the LRCM assignment. Figure 4B shows the result of LiFTCM($\alpha = 1.4$, $\beta = 0$, $\overline{w} = 1$), which is interpreted as the PRCM assignment. Figure 4C shows the result of LiFTCM($\alpha = 1.4$, $\beta = 0.1$, $\overline{w} = 1$), which is interpreted as the GRCM assignment. Thereby, cluster overlap is realized by lifting the threshold by a linear function.

Figure 5 shows the results of LiFTCM($\alpha = 1 - z$, $\beta = z\delta_c$, $z \in [0, 1]$, $\overline{w} = 1$) in which noise rejection is intended. The noise distance was set to $\delta_c = 0.35$ and the parameter z was set to $\{0, 0.001, 0.25, 0.5, 0.75, 1\}$. Figure 5A shows the result for $z = 0$. A hard partition with a Voronoi boundary is generated in the same manner as in HCM. Figure 5B shows the result for $z = 0.001$. Such a small value of z realize general noise rejection, that is, objects over δ_c distant from any cluster are rejected. The boundary between clusters is the Voronoi boundary, and objects whose distance to any cluster is greater than the noise distance δ_c are shown in black and rejected as noise. As z approaches 1 in the order of Figs. 5C–5E, the overlap between clusters increases. Figure 5F shows the result for $z = 1$. In this case, cluster-wise noise rejection is performed and each cluster is composed of a circle with radius δ_c centered at the cluster center. By adjusting the threshold relative to δ_c, cluster overlap and noise rejection are realized simultaneously.

Thereby, LiFTCM can realize HCM, GRCM, GNR, CNR, and their combinations by lifting the threshold by a linear function.

Schematic diagram

Figure 6 is a schematic diagram of the proposal of this study. Representations of HCM, LRCM, PRCM, GRCM, GNR, CNR, and their combinations by the linear function threshold in LiFTCM with the parameters (α, β), and their relationships are shown. (α, β) = (1, 0) is the default state and represents HCM assignment. Increasing α from 1 and β from 0 increases cluster overlap. Simultaneously increasing α and β increases clustering

Ubukata (2019), *PeerJ Comput. Sci.*, DOI 10.7717/peerj-cs.238

12/20

Figure 5 **Classification boundaries of LiFTCM($\alpha = 1 - z$, $\beta = z\delta_c$, $\overline{w} = 1$) representing HCM, GNR, CNR, and their combinations: (A) $z = 0$ (HCM), (B) $z = 0.001$ (GNR), (C) $z = 0.25$ (combination), (D) $z = 0.5$ (combination), (E) $z = 0.75$ (combination), and (F) $z = 1$ (CNR).**

Full-size 🖼 DOI: 10.7717/peerjcs.238/fig-5

roughness. This shows combinations of LRCM and PRCM, namely, GRCM. LiFTCM gives an interpretation in $0 \leq \alpha \leq 1$ in addition to GRCM. As proposed in the smooth transition, when the parameter z is increased from 0 to 1, (α, β) transits from $(1, 0)$ to $(0, \delta_c)$, namely, from HCM to CNR via GNR. The parameter z has the effect of changing clustering more possibilistic. Cluster overlap in CNR is attributed to the increase in β in LRCM. The direction in which the destination δ_c is lowered is the direction in which noise objects are more rejected.

NUMERICAL EXPERIMENTS

This section presents the results of numerical experiments for evaluating the clustering performance of the proposed LiFTCM with various parameter settings in four real-world datasets downloaded from UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/) and summarized in Table 2. Performance was evaluated by the accuracy of class center estimation. The datasets are labeled and include the feature vector and the correct class label of each object. Each dataset was partitioned into disjoint classes according to the class labels, and the center of each class (class center) was calculated. LiFTCM was applied to the generated unlabeled datasets. The number C of clusters was set to the number of classes. To avoid initial value dependence, the initial cluster centers were set to the cluster centers generated by KKZ-based HCM. Considering the correspondence of the clusters and the
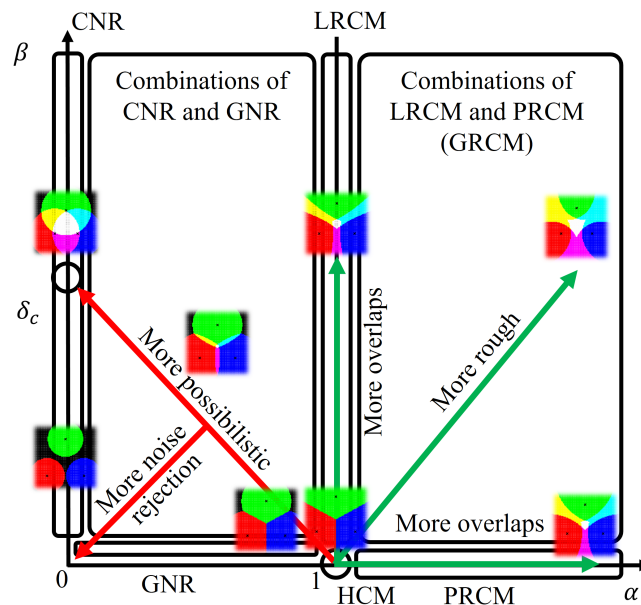
Figure 6 Schematic diagram: representations of HCM, LRCM, PRCM, GRCM, GNR, CNR, and their combinations by a linear function threshold in LiFTCM with the parameters (α, β), and their relationships.

classes, the minimum total error of the cluster centers and the class centers, which is called center-error, was taken as the measurement value. Let $\hat{b}_c$ be the class center of the class corresponding to cluster c. Center-error is calculated by

$$center\_error = \sum_{c=1}^{C} ||b_c - \hat{b}_c||. \tag{21}$$

If the center-error is small, the accuracy of class center estimation is high, and clustering performance is assumed to be high.

Figure 7 shows the center-error measurements as α and β take 100 equally distributed values using contour lines. Colors closer to purple imply smaller center-error and hence better clustering performance. Figure 7A shows the results for the Iris dataset. Performance is improved at approximately $\alpha = 1$ and $\beta = 0.35$, and when α is increased, performance is maintained by decreasing β. This implies that moderate roughness improves performance. Figure 7B shows the results for the Wine dataset. Performance is improved at approximately $\alpha = 1.9$ and $\beta = 100$. When α and β exceed certain values, performance deteriorates rapidly. This implies that moderate roughness is acceptable, but excessive roughness degrades performance. Figure 7C shows the results for the Glass dataset. Performance is improved at approximately $\alpha = 1.3$ and $\beta = 0.7$. As with the Iris dataset, performance is improved with moderate roughness. Figure 7D shows the results for the Breast Cancer Wisconsin dataset. Performance is improved at approximately $\alpha = 1$ and $\beta = 2$, and it is clear that performance is improved with moderate roughness, as is the case with the Iris and the Glass datasets. Therefore, it is suggested that performance is improved when α and

Table 2 Characteristics of the datasets and the range of parameters α, β, δ_c, and z, which tune the linear function threshold in LiFTCM.

Dataset	#classes	#features	#objects (#objects in classes)	Settings of parameters
Iris	3	4	150 (50, 50, 50)	$\alpha \in [1, 1.2]$, $\beta \in [0, 0.6]$, $\delta_c \in [0.85, 1.5]$, $z \in [0, 1]$
Wine	3	13	178 (59, 71, 48)	$\alpha \in [1, 2.4]$, $\beta \in [0, 250]$, $\delta_c \in [150, 1000]$, $z \in [0, 1]$
Glass	6	9	214 (70, 76, 17, 13, 9, 29)	$\alpha \in [1, 1.6]$, $\beta \in [0, 1.5]$, $\delta_c \in [10, 30]$, $z \in [0, 1]$
Breast Cancer Wisconsin	2	9	683 (444, 239)	$\alpha \in [1, 1.5]$, $\beta \in [0, 4]$, $\delta_c \in [5, 70]$, $z \in [0, 1]$

β are increased to obtain moderate roughness. Thus, the representation of combinations of LRCM and PRCM by LiFTCM performs well.
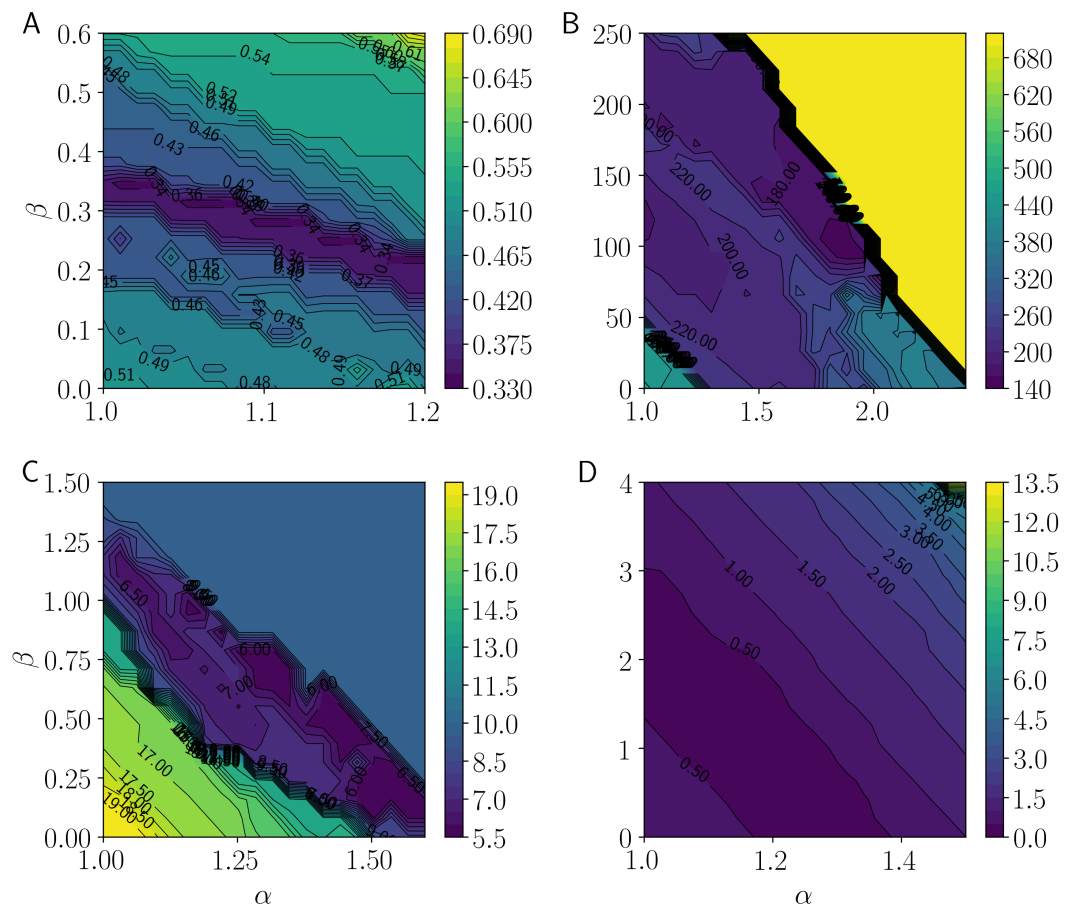
Figure 8 shows the center-error measurements as δ_c and z take 100 equally distributed values using contour lines. Figure 8A shows the results for the Iris dataset. Performance is improved at approximately $\delta_c = 1.1$ and $z = 0.3$, or at approximately $\delta_c = 1.3$ and $z = 0.3$. This implies that setting an appropriate noise distance and combinations of noise and possibilistic clustering yield satisfactory results. Figure 8B shows the results for the Wine dataset. Performance is improved at approximately $\delta_c = 300$ and $z = 0.5$. When δ_c is increased, performance is maintained by decreasing z. This implies that general noise rejection performs better than cluster-wise noise rejection when the noise distance is large. Figure 8C shows the results for the Glass dataset. Performance is improved at approximately $\delta_c = 25$ and $z = 0.05$. Among combinations, those closer to general noise rejection perform well. Figure 8D shows the results for the Breast Cancer Wisconsin dataset. Performance is improved at approximately $\delta_c = 20$ and $z = 0.2$. As with the other datasets, combinations perform well. As in the case of the Wine dataset, states close to general noise rejection perform well when δ_c is large. Therefore, the representation of combinations of GNR and CNR by LiFTCM is satisfactory. When the noise distance is large, states close to GNR tend to yield satisfactory results.

DISCUSSION

Cluster center calculation utilizing probabilistic memberships

RCM-type methods have the problem that even if the number of objects in the boundary area is small, they have unnaturally large impacts on the new cluster center compared to the objects in the lower area, because the cluster center is calculated by the convex combination of these areas. To cope with the problem, Peters proposed πPRCM by introducing the

PeerJ Computer Science _____

Ubukata (2019), *PeerJ Comput. Sci.*, DOI 10.7717/peerj-cs.238 15/20

Figure 7 **Minimum total errors between cluster centers and class centers by LiFTCM($\alpha \geq 1, \beta \geq 0, \overline{w} = 1$) representing GRCM($\overline{w} = 1$): (A) Iris, (B) Wine, (C) Glass, and (D) Breast Cancer Wisconsin.**
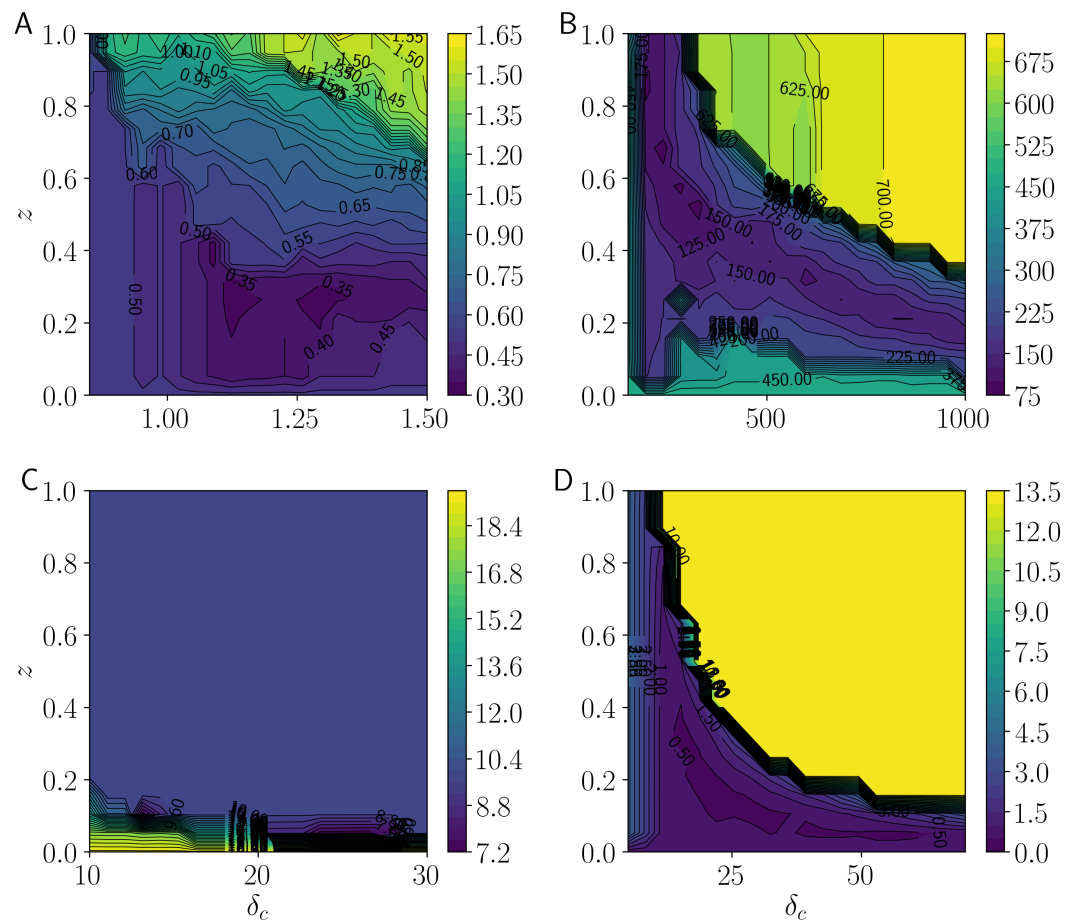**Full-size** 🔳 **DOI: 10.7717/peerjcs.238/fig-7**

cluster center calculation based on the normalized membership of the membership to the upper area, which satisfies the probabilistic constraint (*Peters, 2014*; *Peters, 2015*). "π" is an acronym that stands for "Principle of Indifference," in which the probability is assigned equally by dividing the number of possible clusters. Ubukata et al. proposed π GRCM (*Ubukata et al., 2018*) based on GRCM. The proposed LiFTCM has almost the same formulation as GRCM except that the condition $\alpha \geq 1$ is relaxed to $\alpha \geq 0$. Thus, π LiFTCM can be formulated in a similar manner to π GRCM by introducing the following normalized membership $\tilde{u}_{ci}$ of the membership to the upper area and the cluster center calculation based on $\tilde{u}_{ci}$:

$$\tilde{u}_{ci} = \frac{\overline{u}_{ci}}{\sum_{l=1}^{C} \overline{u}_{li}}, \tag{22}$$

$$\boldsymbol{b}_c = \frac{\sum_{i=1}^{n} \tilde{u}_{ci} \boldsymbol{x}_i}{\sum_{i=1}^{n} \tilde{u}_{ci}}. \tag{23}$$

Here, attention should be paid to the following cases. In the case of $\alpha < 1$, that is, in the case of GNR and CNR, since non-belonging of the object to any cluster is handled and

Figure 8 Minimum total errors between cluster centers and class centers by LiFTCM($\alpha = 1 - z$, $\beta = z\delta_c$, $z \in [0, 1]$, $\overline{w} = 1$) representing HCM, GNR, CNR, and their combinations: (A) Iris, (B) Wine, (C) Glass, and (D) Breast Cancer Wisconsin.

Full-size ⊡ DOI: 10.7717/peerjcs.238/fig-8

thus the denominator $\sum_{l=1}^{C} \overline{u}_{li}$ can become zero, it is necessary to set $\tilde{u}_{ci} = 0$ for all clusters in such cases.

CONCLUSIONS

In this study, as a unified approach for general noise rejection (GNR) and cluster-wise noise rejection (CNR) in hard C-means (HCM), we proposed linear function threshold-based C-means (LiFTCM) by relaxing generalized rough C-means (GRCM) clustering. We showed that the linear function threshold-based assignment in LiFTCM can represent GNR, CNR, and their combinations as well as GRCM. By the visualization of the classification boundaries, transitions among conventional methods based on LiFTCM and their characteristics were clarified. In the numerical experiments, the clustering performance by LiFTCM with various parameter settings was evaluated. It was demonstrated that combinations of LRCM and PRCM, or combinations of GNR and CNR by LiFTCM performed well.

We plan to investigate the relationship between the proposed method and fuzzy clustering with noise rejection. Automatic determination of parameters will also be considered.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

Grant Disclosures

Competing Interests

The authors declare there are no competing interests.

Author Contributions

- Seiki Ubukata conceived and designed the experiments, performed the experiments, analyzed the data, contributed reagents/materials/analysis tools, prepared figures and/or tables, performed the computation work, authored or reviewed drafts of the paper, approved the final draft.

Data Availability

The following information was supplied regarding data availability:

The raw datasets are available in the Supplementary File.

Supplemental Information

Supplemental information for this article can be found online at http://dx.doi.org/10.7717/peerj-cs.238#supplemental-information.

REFERENCES

Arthur D, Vassilvitskii S. 2007. k-means++: the advantages of careful seeding. In: *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms.* Society for Industrial and Applied Mathematics, 1027–1035.

Bezdek JC. 1981. *Pattern recognition with fuzzy objective function algorithms.* New York: Plenum Press.

Cuesta-Albertos JA, Gordaliza A, Matrán C. 1997. Trimmed k-means: an attempt to robustify quantizers. *The Annals of Statistics* **25(2)**:553–576 DOI 10.1214/aos/1031833664.

Davé RN. 1991. Characterization and detection of noise in clustering. *Pattern Recognition Letters* **12(11)**:657–664 DOI 10.1016/0167-8655(91)90002-4.

Davé RN, Krishnapuram R. 1997. Robust clustering methods: a unified view. *IEEE Transactions on Fuzzy Systems* **5(2)**:270–293 DOI 10.1109/91.580801.

Denœux T, Kanjanatarakul O. 2016. Evidential clustering: a review. In: *International symposium on integrated uncertainty in knowledge modelling and decision making.* Cham: Springer, 24–35.

Denœux T, Masson M. 2003. Clustering of proximity data using belief functions. In: *Intelligent systems for information processing.* Amsterdam: Elsevier, 291–302.

Denœux T, Masson M-H. 2004. EVCLUS: evidential clustering of proximity data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **34(1)**:95–109 DOI 10.1109/TSMCB.2002.806496.

Dubes RC, Jain AK. 1988. *Algorithms for clustering data.* New Jersey: Prentice hall Englewood Cliffs.

Dunn JC. 1973. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics* **3(3)**:32–57 DOI 10.1080/01969727308546046.

Edgeworth FY. 1887. On observations relating to several quantities. *Hermathena* **6(13)**:279–285.

García-Escudero LÁ, Gordaliza A. 1999. Robustness properties of k means and trimmed k means. *Journal of the American Statistical Association* **94(447)**:956–969 DOI 10.1080/01621459.1999.10474200.

Gauss CF. 1809. *Theoria Motus Corporum Coelestium in Sectionibus Conicis Solem Ambientium.* Hamburg: Friedrich Perthes und I. H. Besser.

Hampel FR. 1975. Beyond location parameters: Robust concepts and methods. *Bulletin of the International Statistical Institute* **46(1)**:375–382.

Huber PJ. 1964. Robust estimation of a location parameter. *The Annals of Mathematical Statistics* **35(1)**:73–101.

Huber PJ. 1981. *Robust statistics.* New York: John Wiley & Sons.

Katsavounidis I, Kuo C-CJ, Zhang Z. 1994. A new initialization technique for generalized Lloyd iteration. *IEEE Signal Processing Letters* **1(10)**:144–146 DOI 10.1109/97.329844.

Kaufmann L, Rousseeuw PJ. 1987. Clustering by means of medoids. In: Dodge Y, ed. *Statistical data analysis based on the L1—norm and related methods.* Amsterdam: Elsevier, 405–416.

Krishnapuram R, Keller JM. 1993. A possibilistic approach to clustering. *IEEE Transactions on Fuzzy Systems* **1(2)**:98–110 DOI 10.1109/91.227387.

Krishnapuram R, Keller JM. 1996. The possibilistic c-means algorithm: insights and recommendations. *IEEE Transactions on Fuzzy Systems* **4(3)**:385–393 DOI 10.1109/91.531779.

Legendre A-M. 1805. *Nouvelles méthodes pour la détermination des orbites des cométes.* Paris: F. Didot.

Lingras P, West C. 2004. Interval set clustering of web users with rough k-means. *Journal of Intelligent Information Systems* **23(1)**:5–16 DOI 10.1023/B:JIIS.0000029668.88665.1a.

Ubukata (2019), *PeerJ Comput. Sci.*, DOI 10.7717/peerj-cs.238

19/20

MacQueen J. 1967. Some methods for classification and analysis of multivariate observations. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, vol. 1*. Oakland, 281–297.

Maji P, Pal SK. 2007a. RFCM: a hybrid clustering algorithm using rough and fuzzy sets. *Fundamenta Informaticae* **80(4)**:475–496.

Maji P, Pal SK. 2007b. Rough set based generalized fuzzy *C*-means algorithm and quantitative indices. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **37(6)**:1529–1540 DOI 10.1109/TSMCB.2007.906578.

Masson M-H, Denœux T. 2008. ECM: an evidential version of the fuzzy c-means algorithm. *Pattern Recognition* **41(4)**:1384–1397 DOI 10.1016/j.patcog.2007.08.014.

Pal NR, Pal K, Bezdek JC. 1997. A mixed c-means clustering model. In: *Proceedings of 6th international fuzzy systems conference, vol. 1*. IEEE, 11–21.

Pal NR, Pal K, Keller J. M, Bezdek JC. 2005. A possibilistic fuzzy c-means clustering algorithm. *IEEE Transactions on Fuzzy Systems* **13(4)**:517–530 DOI 10.1109/TFUZZ.2004.840099.

Pawlak Z. 1982. Rough sets. *International Journal of Computer & Information Sciences* **11(5)**:341–356 DOI 10.1007/BF01001956.

Pawlak Z. 1991. *Rough sets: theoretical aspects of reasoning about data*. Vol. 9. Dordrecht: Kluwer Academic Publishers.

Peters G. 2006. Some refinements of rough k-means clustering. *Pattern Recognition* **39(8)**:1481–1491 DOI 10.1016/j.patcog.2006.02.002.

Peters G. 2014. Rough clustering utilizing the principle of indifference. *Information Sciences* **277**:358–374 DOI 10.1016/j.ins.2014.02.073.

Peters G. 2015. Is there any need for rough clustering? *Pattern Recognition Letters* **53**:31–37 DOI 10.1016/j.patrec.2014.11.003.

Peters G, Crespo F, Lingras P, Weber R. 2013. Soft clustering–fuzzy and rough approaches and their extensions and derivatives. *International Journal of Approximate Reasoning* **54(2)**:307–322 DOI 10.1016/j.ijar.2012.10.003.

Rousseeuw PJ. 1984. Least median of squares regression. *Journal of the American Statistical Association* **79(388)**:871–880 DOI 10.1080/01621459.1984.10477105.

Rousseeuw PJ, Leroy A. 1987. *Robust regression and outlier detection*. New York: John Wiley & Sons, Inc.

Ubukata S, Kato H, Notsu A, Honda K. 2018. Rough set-based clustering utilizing probabilistic memberships. *Journal of Advanced Computational Intelligence and Intelligent Informatics* **22(6)**:956–964 DOI 10.20965/jaciii.2018.p0956.

Ubukata S, Notsu A, Honda K. 2017. General formulation of rough c-means clustering. *International Journal of Computer Science and Network Security* **17(9)**:29–38.