Algorithms for generating clusters with nonlinear boundaries

Sadaaki Miyamoto
Department of Risk Engineering
School of Systems and Information Engineering
University of Tsukuba
Ibaraki 305-8573, Japan
E-mail: miyamoto@risk.tsukuba.ac.jp

Abstract—Many data clustering algorithms have been proposed among which the c-means techniques are most well-known and widely used. The c-means algorithms in a broad sense are classified into four major categories of the crisp c-means, the fuzzy c-means, and a family of competitive learning algorithms. Since basic algorithms produce piecewise linear boundary of a cluster, clusters with nonlinear boundaries cannot be handled by these algorithms. In this paper we overview two approaches of dealing with nonlinearities. One is to use additional variables whereby boundaries of quadratic curves are produced and classification errors can be reduced. Second method can be handled by these algorithms. In this paper we overview two boundary of a cluster, clusters with nonlinear boundaries cannot be continuously transformed into linear boundaries and in other cases strong nonlinearities with open surfaces can be continuously transformed into linear boundaries, while data in real worlds have more or less nonlinear clusters; in some cases mild nonlinearities with boundaries specified by open surfaces can be continuously transformed into linear boundaries and in other cases strong nonlinearities cannot be continuously transformed into linear boundaries.

In this study we overview current methods of generating clusters with these mild and strong nonlinearities. We consider algorithms of the crisp c-means [3], fuzzy c-means [1], [2], [6], and the competitive learning [9], [3].

These methods include new algorithms of the author’s. Mild nonlinearities are handled by introducing additional variables for clustering [7], while strong nonlinearities are dealt with by employing the kernel-trick in support vector machines [16].

A typical illustrative data set is used to see the capability of the kernel-based methods of clustering.

II. PRELIMINARIES

A. Objective Functions of Fuzzy c-Means

Let the set of objects for clustering be \( X = \{x_1, \ldots, x_n\} \); they are points in the \( p \)-dimensional Euclidean space: \( x_k = (x_{k1}, \ldots, x_{kp})^T \). On the other hand, a cluster \( i \) is represented by the center \( v_i = (v_{i1}, \ldots, v_{ip}) \).

The membership matrix is \( U = (u_{ik}) \), where \( u_{ik} \) is the degree of membership of \( x_k \) to cluster \( i \); the sequence of the cluster centers is \( V = (v_1, \ldots, v_c) \).

The basic alternate optimization algorithm of fuzzy c-means is the iteration of FC in the following [2].

FC: Basic Fuzzy c-Means Algorithm.

FC0. Set the initial value of \( \bar{V} \).

FC1. Solve \( \min_{V \in M} J(U, \bar{V}) \) and let \( \bar{U} \) be the optimal solution.

FC2. Solve \( \min_{V} J(\bar{U}, V) \) and let \( \bar{V} \) be the optimal solution.

FC3. If the solution \((\bar{U}, \bar{V})\) is convergent, stop; else go to FC1.

End of FC.

The ordinary constraint for \( U \) is

\[
M = \{ (u_{ik}) : u_{ik} \in [0, 1], \sum_{i=1}^{c} u_{ik} = 1, \forall k \}.
\]

As the objective function \( J(U, V) \) the following has mainly been considered.

\[
J_0(U, V) = \sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m \|x_k - v_i\|^2 = \sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m d_{ik},
\]

where we put

\[
d_{ik} = \|x_k - v_i\|^2.
\]
The optimal solutions of $u_{ik}$ and $v_i$ are respectively given by

$$u_{ik} = \left[ \sum_{j=1}^{c} \left( \frac{||x_k - v_j||^2}{||x_k - v_j||^2} \right)^{-\frac{1}{2m}} \right]^{-1} \left( \frac{d_{ik}}{d_{jk}} \right)^{-\frac{1}{2m}}$$

and

$$v_i = \frac{\sum_{k=1}^{n} u_{ik} x_k}{\sum_{k=1}^{n} u_{ik}}.$$  

The following entropy-based objective function [10] is more suited for the present discussion.

$$J_1(U, V) = \sum_{i=1}^{c} \sum_{k=1}^{n} \{ u_{ik} d_{ik} + \lambda u_{ik} \log u_{ik} \}. \quad (1)$$

The solutions are

$$u_{ik} = \frac{\exp(-\lambda^{-1}d_{ik})}{\sum_{j=1}^{c} \exp(-\lambda^{-1}d_{jk})} \quad (2)$$

and

$$v_i = \frac{\sum_{k=1}^{n} u_{ik} x_k}{\sum_{k=1}^{n} u_{ik}}. \quad (3)$$

### B. Crisp c-means

The crisp $c$-means algorithm can be obtained as the special case of the objective function $J_1(U, V)$ with $\lambda = 0$ for which the algorithm FC is used. The constraint should also be changed to

$$M = \left\{ (u_{ik}) : u_{ik} \in \{0, 1\}, \sum_{i=1}^{c} u_{ik} = 1, \forall k \right\}.$$  

The solution $U$ is accordingly given by the nearest center rule:

$$u_{ik} = \begin{cases} 1 & (i = \arg \min_{1 \leq j \leq c} ||x_k - v_j||^2), \\ 0 & \text{(otherwise),} \end{cases}$$

while the center is given by (3).

### C. Competitive learning algorithm for clustering

Competitive learning algorithms [3] can also be employed for clustering data. We discuss clustering using the idea of the learning vector quantization [9].

The algorithm is very simple: it is sufficient to repeat learning cluster centers and the allocation of an object to the nearest cluster. In the following algorithm LVQC, $m_i(t)$ is a cluster center and $x(t)$ is randomly selected from the object set.

**Algorithm LVQC:**

**LVQC1.** Set initial value of $m_i, i = 1, \ldots, c.$

**LVQC2.**

$$m_i(t) = \arg \min_{1 \leq j \leq c} ||x(t) - m_j(t)||$$

**LVQC3.** Update $m_1(t), \ldots, m_c(t)$:

$$m_i(t + 1) = m_i(t) + \alpha [x(t) - m_i(t)], \quad m_i(t + 1) = m_i(t), \ i \neq l$$

Allocate the object $x(t)$ to cluster $G_l$.

### End of LVQC.

#### III. VARIABLE FOR CLUSTER VOLUME SIZE

Let us consider an extension of the entropy-based method. Although this extension has been discussed by Ichihashi et al. [7] in a more general form including the covariance matrix, we do not discuss the covariance matrix here for simplicity, but the use of covariance variable is not difficult [5], [6], [7].

That is, a generalized objective function where an additional variable $\alpha = (\alpha_1, \ldots, \alpha_c)$ for controlling cluster volume sizes is used:

$$J'_1(U, V, \alpha) = \sum_{b=1}^{c} \sum_{i=1}^{n} u_{ib} ||x_b - v_i||^2 + \lambda^{-1} \sum_{b=1}^{c} \sum_{i=1}^{n} u_{ib} \log \alpha_i^{-1} u_{ib} \quad (4)$$

The constraint for $\alpha$ is

$$A = \{ \alpha : \sum_{i=1}^{c} \alpha_i = 1, \alpha_i \geq 0, i = 1, \ldots, c \}.$$  

Then the alternate optimization is as follows.

**FC': An Extended Algorithm of Fuzzy c-Means.**

**FC'0.** Set initial value of $V, \bar{V}, \bar{\alpha}.$

**FC'1.** Solve $\min_{V \in M} J(U, V, \bar{\alpha})$ and let the optimal solution be $\bar{U}.$

**FC'2.** Solve $\min_{V \in V} J(\bar{U}, V, \bar{\alpha})$ and let the optimal solution be $\bar{\bar{V}}.$

**FC'3.** Solve $\min_{\alpha \in A} J(\bar{U}, \bar{\bar{V}}, \alpha)$ and let the optimal solution be $\bar{\alpha}.$

**FC'4.** If the solution $(\bar{U}, \bar{\bar{V}}, \bar{\alpha})$ is convergent, stop; else go to **FC'1**.

**End of FC'**.

The optimal solutions are

$$u_{ib} = \frac{\alpha_i \exp(-\lambda^{-1}d_{ib})}{\sum_{j=1}^{c} \alpha_j \exp(-\lambda^{-1}d_{jb})} \quad (5)$$

$$v_i = \frac{\sum_{b=1}^{n} u_{ib} x_b}{\sum_{b=1}^{n} u_{ib}} \quad (6)$$

and

$$\alpha_i = \frac{\sum_{b=1}^{n} u_{ib}}{n} \quad (7)$$

### IV. NONLINEAR CLUSTER BOUNDARIES

Recently support vector machines have been studied by many researchers [16]. Nonlinear classification technique therein uses the kernel trick, that is, a mapping into a high-dimensional feature space of which the functional form is unknown but their inner product has an explicit representation using a kernel function.

We discuss the application of the kernel functions to crisp/fuzzy $c$-means [12] and clustering by competitive learning.
Let a mapping defined on the data space into a high-dimensional feature space be $\Phi(x): \mathbb{R}^p \rightarrow H$; $H$ is in general a Hilbert space of which the inner product and the norm are respectively denoted by $\langle \cdot, \cdot \rangle$ and $\| \cdot \|_H$. The explicit form of $\Phi(x)$ is unknown but the product is given by a kernel function:

$$K(x, y) = \langle \Phi(x), \Phi(y) \rangle.$$

Most important kernel function is the Gaussian kernel $K(x, y) = \exp[-c\|x - y\|^2]$ which is used in the numerical example below.

### A. Fuzzy and crisp $c$-means using kernels

Let us first consider fuzzy $c$-means. The following objective function is used instead of $J_1$.

$$J_1^K(U, V) = \sum_{i=1}^c \sum_{k=1}^n \left\{ u_{ik} \left\| \Phi(x_k) - v_i \right\|^2_H + \lambda u_{ik} \log u_{ik} \right\}.$$ 

For $J_0$, the corresponding objective function are analogously defined but we omit the detail.

We proceed to consider the solution in the alternate optimization. The solution for $U$ is given by the same formula of (5) but the center is

$$v_i = \frac{\sum_{k=1}^n u_{ik} \Phi(x_k)}{\sum_{k=1}^n u_{ik}},$$

which we cannot calculate, since the explicit form of $\Phi(x_k)$ is unknown.

We hence substitute (8) into

$$d_{ik} = \left\| \Phi(x_k) - v_i \right\|^2_H.$$ 

After some manipulation, we have

$$d_{ik} = K_{kk} - 2 U_i \sum_{j=1}^n u_{ij} K_{jk}$$

$$+ \frac{1}{U_i} \sum_{j=1}^n \sum_{l=1}^n u_{ij} u_{il} K_{jl},$$

where

$$U_i = \sum_{k=1}^n u_{ik},$$

$$K_{jk} = \langle \Phi(x_j), \Phi(x_k) \rangle.$$ 

Therefore the alternate optimization of FC is reduced to the iteration of calculating $U$ by (8) and $d_{ik}$ by (10) until convergence of $U$.

The iterative formulas for $J_0$ and $J_1^K$ are derived likewise. We omit the detail.

A crisp $c$-means algorithm using the kernel trick is easy to derive. Basically the idea is the same: we repeat

$$u_{ik} = \begin{cases} 1 & (i = \arg \min_{1 \leq j \leq c} d_{jk}), \\ 0 & (\text{otherwise}). \end{cases}$$

and (10) until convergence.

Variations of the crisp $c$-means algorithm have been proposed by Girolami [4] and by Miyamoto et al. [11].

### B. LVQ clustering with kernels

The clustering by competitive learning algorithm with kernels can also be derived. For this purpose the LVQC algorithm should be rewritten in terms of $d_{ik}(t)$, $t = 1, 2, \ldots$. We have

$$d_{ik}(t + 1) = (1 - \alpha) d_{ik}(t) + \alpha (1 - \alpha) d_{ik}(t)$$

$$= K_{kk} + \alpha^2 K_{kl} - (1 - \alpha) K_{kk} + \alpha (1 - \alpha) K_{kl} - 2 \alpha K_{kl}$$

$$+ \left\{ (1 - \alpha)^2 - (1 - \alpha) + (1 - \alpha) \right\} ||m_i(t)||^2$$

$$= \alpha (K_{kk} - 2 K_{kl} + K_{kl}).$$

Hence we are led to the following algorithm.

**Algorithm K-LVQC:**

1. **K-LVQC1.** Set initial value of $d_{ik}$, $i = 1, \ldots, c$, $k = 1, \ldots, n$.

2. **K-LVQC2.** Calculate

$$d_{ik}(t) = \arg \min_{1 \leq j \leq c} d_{jk}(t).$$

3. **K-LVQC3.** Update $d_{ik}$ by (11).

### V. AN ILLUSTRATIVE EXAMPLE

We discuss a simple and typical illustrative example of a nonlinear cluster boundary. Figure 1 shows a data set classified by the ordinary fuzzy $c$-means using $J_0$ with $m = 2$. The two clusters are shown by $\square$ and $\times$. Two small circles $\bigcirc$ show cluster centers. Apparently, the two circular groups recognized by sight cannot be separated by the ordinary fuzzy $c$-means, since one group is inside the other and hence the cluster boundary should be circular, whereas the crisp and fuzzy $c$-means produces the Voronoi regions [9] with piecewise linear boundaries in general. It is also clear that such circular boundary cannot be obtained by using extensions such as $J_1$.

Figure 2 has been obtained from the method of the kernel trick using the Gaussian kernel with $c\|x - y\| = 0.1$. The crisp and fuzzy $c$-means methods [12], [11] produced the same clusters.

The K-LVQC algorithm produces a similar result [8] of two clusters of the outer circle and the inner ball; we omit the details. See [8].

### VI. CONCLUSION

We have overviewed several objective functions for crisp and fuzzy $c$-means which are employed for generating clusters with nonlinear boundaries. An additional variable for controlling cluster volume sizes has been introduced. Such additional variables enable to generate cluster boundaries with quadratic curves, but stronger nonlinearities cannot be handled by such ordinary methods. Hence the use of the kernel trick has been considered and a new iterative formula has been derived. Moreover a competitive learning algorithm for clustering with
Fig. 1. Data set of ‘a ball and circle’ classified by the ordinary fuzzy \( c \)-means

Fig. 2. Data set of ‘a ball and circle’ classified by the kernel-based fuzzy \( c \)-means (\( \text{GaussianConst}=0.1 \))

kernels has been studied. A typical example of a nonlinear cluster boundary has been shown.

The entropy-based fuzzy \( c \)-means [10], [13] has close relationships with statistical models [14], [15], but relationships among them when kernels are used are not yet uncovered, which is a subject for future study.

Although recent studies on fuzzy clustering is more focused on applications than theories, there are many rooms for further theoretical development and advanced algorithms. Relationships with neural network techniques should furthermore be investigated.

Acknowledgment

This study has partly been supported by the Grant-in-Aid for Scientific Research (B), Japan Society for the Promotion of Science, No.16300065.

REFERENCES