2009 International Symposium on Nonlinear Theory and its Applications
NOLTA'09, Sapporo, Japan, October 18-21, 2009

A Semi-Supervised Entropy Regularized Fuzzy c-Means

Yuchi Kanzawa†, Yasunori Endo‡ and Sadaaki Miyamoto‡

†Shibaura Institute of Technology, Japan
Email: kanzawa@sic.shibaura-it.ac.jp
‡University of Tsukuba, Japan

Abstract—In this paper, two semi-supervised clustering methods are proposed, which are based on entropy regularized fuzzy c-means algorithm. First, two fuzzy c-means algorithms are introduced. The one is the standard one and the other is the entropy regularized one. Second, two semi-supervised standard fuzzy c-means algorithms are introduced, which are derived from adding loss function of memberships to the original optimization problem. Third, two new optimization problems are proposed, in which one is derived from adding new loss function of memberships to the original optimization problem and the other is derived from adding the loss function used in the latter semi-supervised standard fuzzy c-means algorithm. Last, two iterative algorithms are proposed by solving the optimization problems.

1. Introduction

Fuzzy c-means (FCM) [1] is one of the well-known fuzzy clusterings and many FCM variants have been proposed after FCM. In these variants, FCM algorithm based on the concept of regularization by entropy has been proposed by one of the authors [2]. This algorithm is called regularized entropy FCM (eFCM) and is discussed not only for its usefulness but also for its mathematical relations with other techniques.

In addition to the dissimilarity information used by FCM, in many cases a small amount of knowledge is available concerning class labels for some items. Instead of simply using this knowledge for the external validation of the results of clustering, one can imagine letting it guide the clustering process. Semi-supervised c-means clustering models attempt to this problem. Pedrycz [3] proposed a method in which users may have a small set of labeled data that can be used to supervise clustering of the remaining data. This algorithms that use a finite design set of labeled data to help clustering algorithms partition a finite set of unlabeled data. Yamazaki et al. [4] and Yamashiro et al. [5] also proposed another similar method.

In this paper, we propose two types of semi-supervised fuzzy c-means algorithm. One feature is that the proposed method is based on eFCM, where Pedrycz's one is based on sFCM. Another one is that the loss function of the proposed method is KL-divergence between the membership and the teacher value, where Pedrycz's one is the power of the membership and the teacher value.

2. Preliminaries

2.1. Notations

The data set $x = \{x_i \mid x_i \in \mathbb{R}^p, i \in \{1, \ldots, N\}\}$ is given. The membership by which x_i belongs to the j-th cluster is denoted by $u_{i,j}$ ($i \in \{1, \cdots, N\}, j \in \{1, \cdots, C\}$) and the set of $u_{i,j}$ is denoted by $u \in \mathbb{R}^{N \times C}$ called the partition matrix. The constraint for u is

$$\sum_{j=1}^{C} u_{i,j} = 1 \quad (0 \leq u_{i,j} \leq 1). \tag{1}$$

The cluster center set is denoted by $v = \{v_j \mid v_j \in \mathbb{R}^p, j \in \{1, \ldots, C\}\}$.

2.2. FCM

sFCM is the algorithm obtained by solving the following optimization problem:

$$\underset{u,v}{\text{minimize}}\, J_{\text{sfcm}}(u, v) \text{ subject to } \sum_{j=1}^{C} u_{i,j} = 1. \tag{2}$$

where

$$J_{\text{sfcm}}(u, v) = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{i,j}^m \|x_i - v_j\|^2. \tag{3}$$

The parameter m is the fuzzifier satisfying $m > 1$. In this paper, $\|\cdot\|^2$ stands for square of Euclidean norm:

$$\|x_i - v_j\|^2 = \sum_{k=1}^{p} (x_i - v_j)^2. \tag{4}$$

The algorithm obtaining the optimal solutions u and v is omitted by sake of papers.

eFCM is the algorithm obtained by solving the following optimization problem:

$$\underset{u,v}{\text{minimize}}\, J_{\text{efcm}}(u, v) \text{ subject to } \sum_{j=1}^{C} u_{i,j} = 1 \tag{5}$$

where

$$J_{\text{efcm}}(u, v) = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{i,j} \|x_i - v_j\|^2 + \lambda^{-1} \sum_{i=1}^{N} \sum_{j=1}^{C} u_{i,j} \log(u_{i,j}). \tag{6}$$

The second term of the right-hand side in Eq. (6) is for regularization by entropy. The parameter λ is the fuzzifier satisfying $\lambda > 0$. The algorithm obtaining the optimal solutions u and v is omitted by sake of papers.

2.3. Semi-Supervised sFCM

The semi-supervised sFCM by Pedrycz (S-sFCMp) [3] is the algorithm obtained by solving the following opti-

mization problem:

$$\underset{u,v}{\text{minimize}}\, J_{\text{s-sfcmp}}(u,v) \text{ subject to } \sum_{j=1}^{C} u_{i,j} = 1. \quad (7)$$

where

$$J_{\text{s-sfcmp}}(u,v) = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{i,j}^{m}\|x_i - v_j\|^2 + \alpha \sum_{i=1}^{N}\sum_{j=1}^{C}(u_{i,j} - b_i\tilde{u}_{i,j})^m\|x_i - v_j\|^2, \quad (8)$$

where $\alpha > 0$, the pointer $b_i = 1$ if x_i is labeled, and $b_i = 0$ otherwise, $\tilde{u}_{i,j}$ is the label for memberships $u_{i,j}$. The algorithm obtaining the optimal solutions u and v is omitted by sake of papers.

The semi-supervised sFCM by Pedrycz (S-sFCMm) [4] is the algorithm obtained by solving the following optimization problem:

$$\underset{u,v}{\text{minimize}}\, J_{\text{s-sfcmm}}(u,v) \text{ subject to } \sum_{j=1}^{C} u_{i,j} = 1. \quad (9)$$

where

$$J_{\text{s-sfcmm}}(u,v) = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{i,j}^{m}\|x_i - v_j\|^2 + \alpha \sum_{i=1}^{N}\sum_{j=1}^{C}|u_{i,j} - \tilde{u}_{i,j}|, \quad (10)$$

where $\alpha > 0$ and $\tilde{u}_{i,j}$ is the label for memberships $u_{i,j}$. The algorithm obtaining the optimal solutions u and v is omitted by sake of papers.

3. Semi-Supervised eFCM

3.1. Semi-Supervised eFCM by KL-divergence

In this section, we proposed a new semi-supervised entropy regularized fuzzy c-means algorithms. This can be obtained by solving a new optimization problem, in which KL divergence between a membership and the corresponding teacher value is added to the original optimization problem of eFCM.

First, we consider the following optimization problem:

$$\underset{u,v}{\text{minimize}}\, J_{\text{efcm}}(u,v) + \sum_{i=1}^{N}\alpha_i\sum_{j=1}^{C} u_{i,j}\log\left(\frac{u_{i,j}}{\tilde{u}_{i,j}}\right) \quad (11)$$

$$\text{subject to } \sum_{j=1}^{C} u_{i,j} = 1. \quad (12)$$

This corresponding Lagrange function is described as

$$L_{\text{s-efcmk}}(u,v) = J_{\text{efcm}}(u,v) + \sum_{i=1}^{N}\alpha_i\sum_{j=1}^{C} u_{i,j}\log\left(\frac{u_{i,j}}{\tilde{u}_{i,j}}\right) + \sum_{i=1}^{N}\gamma_i\left(\sum_{j=1}^{C} u_{i,j} - 1\right), \quad (13)$$

where α_i is a weight parameter for KL divergence between a membership and the corresponding teacher value and $\gamma = (\gamma_1, \cdots, \gamma_N)^{\mathrm{T}}$ is Karush-Kuhn-Tucker vector. Karush-Kuhn-Tucker conditions are described as below:

$$\frac{\partial L_{\text{s-efcmk}}}{\partial u_{i,j}} = 0, \quad (14)$$

$$\frac{\partial L_{\text{s-efcmk}}}{\partial \gamma_i} = 0, \quad (15)$$

$$\frac{\partial L_{\text{s-efcmk}}}{\partial v_j} = 0. \quad (16)$$

KKT condition (14) implies that

$$d_{i,j} + \lambda^{-1}(\log(u_{i,j}) + 1) + \alpha_i\log\left(\frac{u_{i,j}}{\tilde{u}_{i,j}}\right) + \gamma_i = 0, \quad (17)$$

from which we have

$$u_{i,j} = \exp\left(\frac{-d_{i,j} + \alpha_i\log\left(\tilde{u}_{i,j}\right)}{\lambda^{-1} + \alpha_i}\right)\exp\left(\frac{-\lambda^{-1} - \gamma_i}{\lambda^{-1} + \alpha_i}\right). \quad (18)$$

From this form and KKT condition (15), we have

$$\exp\left(\frac{-\lambda^{-1} - \gamma_i}{\lambda^{-1} + \alpha_i}\right) = \frac{1}{\sum_{k=1}^{C}\exp\left(\frac{-d_{i,j} + \alpha_i\log(\tilde{u}_{i,j}}{\lambda^{-1} + \alpha_i}\right)}. \quad (19)$$

Therefore we have the the optimal solution of $u_{i,j}$ as

$$u_{i,j} = \frac{\exp\left(\frac{-\lambda d_{i,j} + \alpha_i\lambda\log(\tilde{u}_{i,j})}{1 + \lambda\alpha_i}\right)}{\sum_{k=1}^{C}\exp\left(\frac{-\lambda d_{i,k} + \alpha_i\lambda\log(\tilde{u}_{i,k})}{1 + \lambda\alpha_i}\right)}. \quad (20)$$

If the weight parameter of KL-divergence α_i is equal to zero, the optimal membership value coincides with the one for eFCM.

From KKT condition (16), we have the optimal solution of cluster centers v_j as

$$v_j = \left(\sum_{i=1}^{N} u_{i,j}\right)^{-1}\sum_{i=1}^{N} u_{i,j}x_i. \quad (21)$$

From the above discussion, we propose a new semi-supervised entropy regularized fuzzy c-means algorithms by KL-divergence as below:

Algorithm 1

Step 1 *Give the number of cluster C, the teacher values of membership $\tilde{u}_{i,j}$, the weight parameter of KL-divergence between the unknown membership $u_{i,j}$ and the teacher value of membership $\tilde{u}_{i,j}$ and the fuzzifier parameter λ. Set the initial cluster centers v.*

Step 2 *Calculate u by Eq. (20).*

Step 3 *Calculate v by Eq. (21).*

Step 4 *Check the stopping criterion for (u, v). If the criterion is not satisfied, go back to* Step 2.

3.2. Semi-Supervised eFCM by Manhattan Loss Function

In this section, we propose another new semi-supervised entropy regularized fuzzy c-means algorithms. This can be obtained by solving a new optimization problem, in which a loss function of Manhattan distance between a membership and the corresponding teacher value is added to the original optimization problem of eFCM.

First, we consider the following optimization problem:

$$\underset{u,v}{\text{minimize}}\, J_{\text{efcm}}(u,v) + \sum_{i=1}^{N}\beta_i \sum_{j=1}^{C}|u_{i,j}-\tilde{u}_{i,j}| \tag{22}$$

$$\text{subject to } \sum_{j=1}^{C} u_{i,j} = 1. \tag{23}$$

This corresponding Lagrange function is described as

$$\begin{aligned} L_{\text{s-efcmm}}(u,v) =& J_{\text{efcm}}(u,v) + \sum_{i=1}^{N}\beta_i \sum_{j=1}^{C}|u_{i,j}-\tilde{u}_{i,j}| \\ &+ \sum_{i=1}^{N}\gamma_i\left(\sum_{j=1}^{C}u_{i,j}-1\right), \end{aligned} \tag{24}$$

where β_i is a weight parameter for the Manhattan distance between a membership and the corresponding teacher value, and $\gamma = (\gamma_1, \cdots, \gamma_N)^{\mathrm{T}}$ is Karush-Kuhn-Tucker vector. Karush-Kuhn-Tucker conditions are described as below:

$$\frac{\partial L_{\text{s-efcmm}}}{\partial u_{i,j}} = 0, \quad (u_{i,j} \neq \tilde{u}_{i,j}), \tag{25}$$

$$\begin{aligned} u_{i,j} =& \tilde{u}_{i,j}, \quad \lim_{u_{i,j}-\tilde{u}_{i,j}\to -0}\frac{\partial L_{\text{s-efcmm}}(u)}{\partial u_{i,j}} \leq 0 \text{ and} \\ & \lim_{u_{i,j}-\tilde{u}_{i,j}\to +0}\frac{\partial L_{\text{s-efcmm}}(u)}{\partial u_{i,j}} \geq 0, \end{aligned} \tag{26}$$

$$\frac{\partial L_{\text{s-efcmm}}}{\partial \gamma_i} = 0, \tag{27}$$

$$\frac{\partial L_{\text{s-efcmm}}}{\partial v_j} = 0. \tag{28}$$

From KKT condition (28), we have the optimal solution of cluster centers v_j as Eq. (21). From KKT condition (25) and (26), we have

$$u_{i,j} = \exp\Big(-\lambda(d_{i,j}+\gamma_i+\beta_i)-1\Big), \quad (u_{i,j} > \tilde{u}_{i,j}) \tag{29}$$

$$u_{i,j} = \tilde{u}_{i,j}, \quad \left(\begin{array}{c} d_{i,j}+\lambda^{-1}(1+\log(u_{i,j}))+\beta_i+\gamma_i \geq 0 \\ \text{and} \\ d_{i,j}+\lambda^{-1}(1+\log(u_{i,j}))-\beta_i+\gamma_i \leq 0 \end{array}\right), \tag{30}$$

$$u_{i,j} = \exp\Big(-\lambda(d_{i,j}+\gamma_i-\beta_i)-1\Big), \quad (0 < u_{i,j} < \tilde{u}_{i,j}), \tag{31}$$

which we reformulate by cases of γ_i into

$$\begin{aligned} u_{i,j} =& \exp(-\lambda(d_{i,j}+\gamma_i+\beta_i)-1), \\ & (\gamma_i < -d_{i,j}-\lambda^{-1}(1+\log(\tilde{u}_{i,j}))-\beta_i) \end{aligned} \tag{32}$$

$$\begin{aligned} u_{i,j} =& \tilde{u}_{i,j}, \\ & (-d_{i,j}-\lambda^{-1}(1+\log(\tilde{u}_{i,j}))-\beta_i \leq \\ & \gamma_i \leq -d_{i,j}-\lambda^{-1}(1+\log(\tilde{u}_{i,j}))+\beta_i), \end{aligned} \tag{33}$$

$$\begin{aligned} u_{i,j} =& \exp(-\lambda(d_{i,j}+\gamma_i-\beta_i)-1), \\ & (-d_{i,j}-\lambda^{-1}(1+\log(\tilde{u}_{i,j}))+\beta_i < \gamma_i). \end{aligned} \tag{34}$$

Since $u_{i,j}$ is decreasing for γ_i and satisfies

$$\lim_{\gamma_i\to\infty} u_{i,j}(\gamma_i) = 0, \tag{35}$$

$$\lim_{\gamma_i\to-\infty} u_{i,j}(\gamma_i) \to \infty, \tag{36}$$

there exists the unique γ_i such that the condition (27) satisfies, which implies that we have the unique optimal solution $u_{i,j}$ of the optimization problem (22) and (23). The following algorithm obtains such $u_{i,j}$.

Algorithm 2

Step 1. *Let G be the set of $(-d_{i,j}-\lambda^{-1}(1+\log(\tilde{u}_{i,j}))+\beta_i, +, j)$ and $(-d_{i,j}-\lambda^{-1}(1+\log(\tilde{u}_{i,j}))-\beta_i, -, j)$, where the first component is the values of γ_i at which $u_{i,j}(\gamma_i)$ is not differentiable, the second one is the signs before the corresponding β_i, and the last one is the cluster index j of $u_{i,j}$. Sort the element of G with the increasing order of the first component. Set $t = 1$. Let $K_1 = \{1, \cdots, C\}$, $K_2 =$ and $K_3 =$ be the set of cluster indices.*

Step 2. *Calculate*

$$\begin{aligned} &\exp(-\lambda(\hat{\gamma}_i-\beta_i)-1)\sum_{k\in K_1}\exp(-\lambda d_{i,k}) + \\ &\sum_{k\in K_2}\tilde{u}_{i,k} + \exp(-\lambda(\hat{\gamma}_i+\beta_i)-1)\sum_{k\in K_3}\exp(-\lambda d_{i,k}), \end{aligned} \tag{37}$$

where $\hat{\gamma}_i$ is the first component of G_t. If Eq. (37) is equal to 1, end this algorithm with $u_{i,j}$ as below:

$$u_{i,j} = \begin{cases} \exp(-\lambda(d_{i,j}+\hat{\gamma}_i-\beta_i)-1) & (j \in K_1), \\ \tilde{u}_{i,j} & (j \in K_2), \\ \exp(-\lambda(d_{i,j}+\hat{\gamma}_i+\beta_i)-1) & (j \in K_3). \end{cases} \tag{38}$$

If Eq. (37) is greater than 1, end this algorithm with $u_{i,j}$ as

$$u_{i,j} = \begin{cases} U_i\exp(-\lambda(d_{i,j}-\beta_i)) & (j \in K_1), \\ \tilde{u}_{i,j} \quad (j \in K_2), & \\ U_i\exp(-\lambda(d_{i,j}+\beta_i)) & (j \in K_3), \end{cases} \tag{39}$$

by solving the following equation for γ_i as

$$\begin{aligned} &\exp(-\lambda(\hat{\gamma}_i-\beta_i)-1)\sum_{k\in K_1}\exp(-\lambda d_{i,k}) + \\ &\sum_{k\in K_2}\tilde{u}_{i,k} + \exp(-\lambda(\hat{\gamma}_i+\beta_i)-1)\sum_{k\in K_3}\exp(-\lambda d_{i,k}) = 1, \end{aligned} \tag{40}$$

where

$$U_i = \frac{1 - \sum_{k\in K_2} \tilde{u}_{i,k}}{\sum_{k\in K_1} \exp(-\lambda(d_{i,k} - \beta_i)) + \sum_{k\in K_3} \exp(-\lambda(d_{i,k} + \beta_i))} \tag{41}$$

If Eq. (37) is less than 1 and the second component of G_t is +, move the element whose third component is j from K_1 to K_2. Otherwise, move the element whose third component is j from K_1 to K_2.

Step 3. *Set $t \leftarrow t + 1$ and go back to* Step 2..

From the above discussion, we propose a new semi-supervised entropy regularized fuzzy c-means algorithms by Manhattan loss function:

Algorithm 3

Step 1 *Give the number of cluster C, the teacher values of membership $\tilde{u}_{i,j}$, the weight parameter of KL-divergence between the unknown membership $u_{i,j}$ and the teacher value of membership $\tilde{u}_{i,j}$ and the fuzzifier parameter λ. Set the initial cluster centers v.*

Step 2 *Calculate u by Algorithm 2.*

Step 3 *Calculate v by Eq. (21).*

Step 4 *Check the stopping criterion for (u, v). If the criterion is not satisfied, go back to* Step 2.

4. Constructing Dissimilarity Matrix, Kernel Gram Matrix and Kernel Function for Must-link

In this section, we construct dissimilarity matrix $d^* \in \mathbb{R}^{n\times n}$ and kernel function $K^* : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ affected by *must-link* — two data have to be together in the same cluster — in order to apply to fuzzy relational clustering method and to kernel fuzzy clustering method, respectively.

The easiest idea of constructing dissimilarity matrix d^* affected by *must-link* that x_i and $x_{\tilde{i}}$ have to be together in the same cluster, is replacing the corresponding element $d_{i,\tilde{i}}$ by a certain small positive value. With such the dissimilarity matrix d^*, fuzzy relational clustering algorithm will produce a result.

If x_i and $x_{\tilde{i}}$ have to be together in the same cluster, other data close to x_i and $x_{\tilde{i}}$ (x_i and other data close to $x_{\tilde{i}}$) also should be together in the same cluster. Based on this idea, we construct dissimilarity matrix as below. First, make the graph whose i-th node is correspond to x_i and whose edge connecting i-th and $\tilde{i}$-th nodes has the value $d_{i,\tilde{i}}$. Second, replace the value of $(i, \tilde{i})$-edge by a certain small nonnegative value if $(i, \tilde{i})$ is an element of *must-link*. Third, replace the value of $(i, \tilde{i})$-edge by the minimum adding each edge in $(i, \tilde{i})$-path, which can be achieved by dynamic programming. Last, adopt the value of $(i, \tilde{i})$-edge as $d^*_{i,\tilde{i}}$. With such the dissimilarity matrix d^*, fuzzy relational clustering algorithm produce a result affected not only by *must-link* but also by other data close to the data in *must-link*. From the dissimilarity matrix d^*, we can construct kernel gram matrix and apply to kernel fuzzy clustering methods. Kernel gram matrix must be positive semi-definite. Another matrix d^{**} by diagonal shift or eigen-value shift of d^* can be kernel gram matrix.

In order to construct fuzzy classification function, we need not only kernel gram matrix but also kernel function affected by *must-link*. Such kernel function based on Gaussian kernel can be obtained as below. First, insert a simple Gaussian kernel $K(x, y) = \exp(-\sigma\|x-y\|^2)$ to a list of functions $\mathcal{L}$ and adopt the kernel function as the maximal of $\mathcal{L}$,

$$K(\bar{x}, \bar{y}) = \max_{\kappa(x,y)\in\mathcal{L}} \kappa(\bar{x}, \bar{y}), \tag{42}$$

which is corresponding to original dissimilarity matrix d. Second, insert two functions $\exp(-\sigma_{i,\tilde{i}}(\|x - x_i\|^2 + \|x_{\tilde{i}} - y\|^2 + d^*_{i,\tilde{i}}))$ and $\exp(-\sigma_{i,\tilde{i}}(\|x - x_{\tilde{i}}\|^2 + \|x_i - y\|^2 + d^*_{i,\tilde{i}}))$ to $\mathcal{L}$ if $(x_i, x_{\tilde{i}})$ is an element of *must-link*, and adopt the kernel function as the maximal of $\mathcal{L}$. This updated kernel function is corresponding to the second step in the previous paragraph and remark that $K(x_i, x_{\tilde{i}}) = d^*_{i,\tilde{i}}$. Third, with d^* obtained through the third step in the previous step, insert two functions $\exp(-\sigma_{i,\tilde{i}}(\|x - x_i\|^2 + \|x_{\tilde{i}} - y\|^2 + d^*_{i,\tilde{i}}))$ and $\exp(-\sigma_{i,\tilde{i}}(\|x - x_{\tilde{i}}\|^2 + \|x_i - y\|^2 + d^*_{i,\tilde{i}}))$ to $\mathcal{L}$ and adopt the kernel function as the maximal of $\mathcal{L}$. In order for such function to be kernel, the original function $\exp(-\sigma\|x-y\|^2)$ may be replaced by $a\exp(-\sigma\|x - y\|^2)$ with $a > 1$, where this replacement is corresponding to diagonal shift of $d^*_{i,\tilde{i}}$.

5. Conclusion

In this paper, we first proposed two new semi-supervised clustering algorithms, in which one is derived from the optimization problem by adding KL-divergence between the membership and the teacher value to the original one for eFCM and in which the other is derived from the optimization problem by adding Manhattan distance between the membership and the teacher value to the original one for eFCM. We also proposed how to construct dissimilarity matrix, kernel gram matrix and kernel function affected by *must-link*.

References

[1] Bezdek, J.P.: Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum, New York (1981).

[2] Miyamoto, S. and Umayahara, K.: "Methods in Hard and Fuzzy Clustering", in: Liu, Z.-Q. and Miyamoto, S. (eds), Soft computing and human-centered machines, Springer-Verlag Tokyo (2000).

[3] Pedrycz, W.: "Algorithms of Fuzzy Clustering with Partial Supervision", Pattern Recognition Letter, Vol.3, pp.13-20 (1985).

[4] Yamazaki, M., Miyamoto, S. and Lee, In-Jae: "Semi-supervised Clustering with Two Types of Additional Functions", Proc. 24th Fuzzy System Symposium, 2E2-01 (2009).

[5] Yamashiro, M., Endo, Y., Hamasuna, Y., Miyamoto, S.: "A Study on Semi-supervised Fuzzy c-Means", Proc. 24th Fuzzy System Symposium, 2E3-04 (2009).