Udk 519.7:004.8

Ye. Bodyanskiy, A. Shafronenko
Kharkiv National University of Radio Electronics

ADAPTIVE FUZZY CLUSTERING FOR DATA WITH MISSING VALUES BASED ON THE NEAREST PROTOTYPE - CENTROID STRATEGY

Розглянуто задачу кластеризації масивів векторних даних, що мають пропущені значення у деяких компонентах. Запропоновано адаптивний підхід до кластеризації таких даних за умов, коли класи перетинаються. В основі підходу є використання модифікованої мапи Кохонена із функцією сусідства спеціального вигляду.

Ключові слова: нечітка кластеризація, самоорганізовна мережа Кохонена, правило навчання, неповні дані з пропущенними значеннями.

The problem of clustering vector data sets with missing values in some components is considered. The adaptive approach to clustering of data in situation then classes overlap is proposed. The basis of the approach is the using of the modified Kohonen maps with the neighborhood function of special kind.

Key words: fuzzy clustering, Kohonen self-organizing network, learning rule, incomplete data with missing values.

Introduction

The task of clustering for data sets with missing values often occurs in applications and for its solutions have been successfully used artificial neural networks [1] and method soft computing [2]. The main assumption of this approach that the original array is set a priori, the number of missing values is known in advance, and processing is organized in a batch mode. In this paper we propose an adaptive fuzzy clustering procedure that is designed to deal with the data sequence containing an unknown number of missing values, solves the problem in the on-line mode, characterized by numerical simplicity using strategy of nearest prototype [2].

Problem statement

Baseline information for solving the task of clustering in a batch mode is the sample of observations, formed from N n-dimensional feature vectors $X = \{x, x, ..., x\} \subset R$, $x \in X$, $k = 1, 2, ..., N$. The result of clustering is the partition of original data set into m classes $(1 < m < N)$ with some level of membership $U_q(k)$ of k-th feature vector to the q-th cluster $(1 \le q \le m)$. Incoming data previously are centered and standardized by all features, so that all observations belong to the hypercube $[-1,1]^n$. In the presence of an unknown number of missing values in vector-images $\tilde{x}_k$, that form array $\tilde{X}$, let's introduce the sub-arrays:

$$X_F = \{\tilde{x}_k \in \tilde{X} \mid \tilde{x}_k - \textit{vector containing all components}\};$$

$$X_P = \{\tilde{x}_{ki}, 1 \le i \le n, 1 \le k \le N \mid \textit{all values } \tilde{x}_k, \textit{ available in } \tilde{X}\};$$

$$X_G = \{\tilde{x}_{ki} = ?, 1 \le i \le n, 1 \le k \le N \mid \textit{all values } \tilde{x}_k, \textit{absent in } \tilde{X}\}.$$

The strategy of the nearest centroid-prototype

The strategy of the nearest centroid-prototype can be considered as a hybrid strategy of the optimum completion and partial distances [2] and consists of a sequence of steps:

1. Setting the initial conditions for the algorithm: fuzzifier $\beta > 0$; m; desired accuracy $\varepsilon > 0$; prototypes (centroids) of clusters $w_q^{(0)}$; the number of processing epochs $\tau = 0,1,2,...,Q$; $X_G^{(0)} = \{-1 \le \hat{x}_{ki}^{(0)} \le 1\} - N_G$ arbitrary estimates of missing values $\tilde{x}_{ki} \in X_G$;

2. Calculation of membership levels by solving the optimization problem:

$$U_q^{(\tau+1)}(k) = (\sum_{l=1}^{m} (\left\| \hat{x}_k^{(\tau)} - w_l^{(\tau)} \right\|^2)^{\frac{1}{1-\beta}})^{-1} (\left\| \hat{x}_k^{(\tau)} - w_q^{(\tau)} \right\|^2)^{\frac{1}{1-\beta}},$$

(here vector $\hat{x}_k^{(\tau)}$ differs from $\tilde{x}_k$ by replacing missing values $\tilde{x}_{ki} \in X_G$ by estimates $\hat{x}_{ki}^{(\tau)}$ that are calculated for the τ-th epoch of data processing);

3. Calculation the prototypes (centroids) of clusters:

$$w_q^{(\tau+1)} = (\sum_{k=1}^{N} (U_q^{(\tau+1)}(k))^{\beta})^{-1} \sum_{k=1}^{N} (U_q^{(\tau+1)}(k))^{\beta} \hat{x}_k^{(\tau)};$$

4. Checking the stop conditions:

if $\left\| w_q^{(\tau+1)} - w_q^{(\tau)} \right\| < \varepsilon \ \forall \ 1 \le q \le m$ or $\tau = Q$, then the algorithm terminates, otherwise go to step 5;

5. Estimating of missing values by finding the prototype $w_q^{(\tau+1)}$ nearest to $\tilde{x}_k$ in the sense of the partial distances

$$D_P^2(\tilde{x}_k, w_q) = \frac{n}{\delta_{k\Sigma}} \sum_{i=1}^{n} (\tilde{x}_{ki} - w_{qi})^2 \delta_{ki}$$

where

$$\delta_{ki} = \begin{cases} 0 \,|\, \tilde{x}_{ki} \in X_G, \\ 1 \,|\, \tilde{x}_{ki} \in X_F, \end{cases} \qquad \delta_{k\Sigma} = \sum_{i=1}^{n} \delta_{ki},$$

finding

$$w_q^{(\tau+1)} = \arg\min_q \{D_P^2(\tilde{x}_k, w_1^{(\tau+1)}), ..., D_P^2(\tilde{x}_k, w_m^{(\tau+1)})\}$$

and replacing the missing observations $\tilde{x}_{ki}$ by estimates $\hat{x}_{ki}^{(\tau+1)} = w_{qi}^{(\tau+1)}$.

Next, go to step 2.

Easy to see that on the fifth step, if considered it from the position of the Kohonen's self-organizing maps [3], the winner neuron nearest to observation $\tilde{x}_k$ is founded in the terms of the partial distances D_P^2.

Then we can write the strategies in the form of the nearest prototype:

$$\begin{cases} U_q^{(\tau+1)}(k) = (\sum_{l=1}^{m} (\left\| \hat{x}_k^{(\tau)} - w_l(k) \right\|^2)^{\frac{1}{1-\beta}})^{-1} (\left\| \hat{x}_k^{(\tau)} - w_q(k) \right\|^2)^{\frac{1}{1-\beta}}, \\ \text{where } \hat{x}_{ki}^{(\tau)} = w_{qi}(k), \ \ w_q(k) = \arg\min_q \{D_P^2(\tilde{x}_k, w_1(k)), ..., D_P^2(\tilde{x}_k, w_m(k))\}, \\ w_q(k+1) = w_q(k) + \eta(k+1)(U_q^{(Q)}(k))^{\beta} (\hat{x}_k^{(Q)} - w_q(k)) \ \ \forall q = 1,2,...,m, \end{cases} \quad (1)$$

with memberships between instance of real time k and $k+1$ are tuned in an accelerated time $\tau = 0,1,2,...,Q$, and centroids in on-line mode $k = 0,1,2,...,N,...$ using the Kohonen's self-learning rule [3] "The Winner Takes More" with the neighborhood function $(U_q^{(Q)}(k))^{\beta}$.

Thus, the fuzzy clustering for data with missing values can be organized in sequential mode by conventional Kohonen's map with a special neighborhood function $(U_q^{(Q)}(k))^{\beta}$ that having the Cauchy distribution form.

Centroid-prototypes can also be recalculate in an accelerated time according to the last relation in (1), although this complicates the realization of computational clustering procedure, which in this case has the form

$$\begin{cases} U_q^{(\tau+1)}(k) = \dfrac{(\left\| \hat{x}_k^{(\tau)} - w_q^{(\tau)}(k) \right\|^2)^{\frac{1}{1-\beta}}}{\sum\limits_{l=1}^{m} (\left\| \hat{x}_k^{(\tau)} - w_l(k) \right\|^2)^{\frac{1}{1-\beta}}}, \\ \text{where } \hat{x}_{ki}^{(\tau)} = w_{qi}^{(\tau)}(k), \quad w_q^{(\tau)}(k) = \arg\min\{D_P^2(\tilde{x}_k, w_1^{(\tau)}(k)),...,D_P^2(\tilde{x}_k, w_m^{(\tau)}(k))\}, \\ w_q^{(Q)}(k) = w_q^{(Q)}(k+1), \\ w_q^{(\tau+1)}(k+1) = w_q^{(\tau)}(k+1) + \eta(k+1)(U_q^{(Q)}(k))^{\beta}(\hat{x}_k^{(\tau)} - w_q^{(\tau)}(k+1)) \, . \end{cases} \tag{2}$$

Possibilistic adaptive strategy of the nearest centroid

The main disadvantage of probabilistic fuzzy clustering algorithms (FCM and similar procedures) associated with hard condition on the sum of membership levels for each vector-image, which must be equal to unity, i.e. indirectly attached the sense of probabilities to the memberships that is not always correct from view of the problem.

For restrictions of these assumption possibilistic fuzzy clustering algorithms were introduced. The basic procedure (PCM) has the form [4]:

$$\begin{cases} U_q^{(\tau+1)}(k) = \dfrac{1}{1 + (\dfrac{\left\| \tilde{x}_k - w_q^{(\tau)} \right\|^2}{\mu_q^{(\tau)}})}, \\ w_q^{(\tau+1)} = \dfrac{\sum\limits_{k=1}^{N} (U_q^{(\tau+1)}(k))^{\beta}}{\sum\limits_{k=1}^{N} (U_q^{(\tau+1)}(k))^{\beta}}, \\ \mu_q^{(\tau+1)} = \dfrac{\sum\limits_{k=1}^{N} (U_q^{(\tau+1)}(k))^{\beta} \left\| \tilde{x}_k - w_q^{(\tau+1)} \right\|^2}{\sum\limits_{k=1}^{N} (U_q^{(\tau+1)}(k))^{\beta}} \end{cases}$$

where $\mu_q \geq 0$ determines the distance at which level of membership takes the value 0.5, i.e. if

$$\left\| \tilde{x}_k - w_q \right\|^2 = \mu_q \, ,$$

then $U_q(k) = 0,5$.

Possibilistic strategy of nearest prototype-centroid in this case can be written as the following sequence of steps:

1. Setting the initial conditions for the algorithm: fuzzifier $\beta > 0$; m; desired accuracy $\varepsilon > 0$; prototypes (centroids) of clusters $w_q^{(0)}$; the number of processing epochs $\tau = 0,1,2,...,Q$; $X_G^{(0)} = \{-1 \le \hat{x}_{ki}^{(0)} \le 1\} - N_G$ arbitrary estimates of missing values $\tilde{x}_{ki} \in X_G$;

2. Calculation of membership levels by solving the optimization problem:

$$U_q^{(\tau+1)}(k) = \frac{1}{1 + \left(\frac{\left\| \hat{x}_k^{(\tau)} - w_q^{(\tau)} \right\|^2}{\mu_q^{(\tau)}}\right)^{\frac{1}{\beta-1}}};$$

3. Calculation the prototypes (centroids) of clusters:

$$w_q^{(\tau+1)}(k) = \frac{\sum_{k=1}^{N} (U_q^{(\tau+1)}(k))^{\beta} \hat{x}_k^{(\tau)}}{\sum_{k=1}^{N} (U_q^{(\tau+1)}(k))^{\beta}};$$

4. Checking the stop conditions:

if $\left\| w_q^{(\tau+1)} - w_q^{(\tau)} \right\| < \varepsilon \ \forall \ 1 \le q \le m$ or $\tau = Q$, then the algorithm terminates, otherwise go to step 5;

5. Estimating of missing values by finding the prototype $w_q^{(\tau+1)}$ nearest to $\tilde{x}_k$ in the terms of the partial distances:

$$D_P^2(\tilde{x}_k, w_q) = \frac{n}{\delta_{k\Sigma}} \sum_{i=1}^{n} (\tilde{x}_{ki} - w_{qi})^2 \delta_{ki}$$

where

$$\delta_{ki} = \begin{cases} 0 \,|\, \tilde{x}_{ki} \in X_G, \\ 1 \,|\, \tilde{x}_{ki} \in X_F, \end{cases} \qquad \delta_{k\Sigma} = \sum_{i=1}^{n} \delta_{ki},$$

finding

$$w_q^{(\tau+1)} = \arg\min_q \{D_P^2(\tilde{x}_k, w_1^{(\tau+1)}), ..., D_P^2(\tilde{x}_k, w_m^{(\tau+1)})\}$$

and replacing the missing observations $\tilde{x}_{ki}$ by estimates $\hat{x}_{ki}^{(\tau+1)} = w_{qi}^{(\tau+1)}$.

6. Calculation the scalar distance parameter:

$$\mu_q^{(\tau+1)} = \frac{\sum_{k=1}^{N} (U_q^{(\tau+1)}(k))^{\beta} \left\| \hat{x}_k^{(\tau+1)} - w_q^{(\tau+1)} \right\|^2}{\sum_{k=1}^{N} (U_q^{(\tau+1)}(k))^{\beta}}.$$

Next, go to step 2.

Similarly to probabilistic adaptive clustering strategies based on the nearest centroid it is possible to organize the process of possibilistic clustering. In this case, an analogue of the algorithm (1) is the procedure:

$$
\begin{cases}
U_q^{(\tau+1)}(k) = \dfrac{1}{1+\left(\dfrac{\left\|\hat{x}_k^{(\tau)} - w_q(k)\right\|^2}{\mu_q^{(\tau)}}\right)^{\frac{1}{\beta-1}}}, \\
\text{where } \hat{x}_{ki}^{(\tau)} = w_{qi}(k), \;\; w_q(k) = \arg\min\limits_{q}\{D_P^2(\tilde{x}_k, w_1(k)), ..., D_P^2(\tilde{x}_k, w_m(k))\}, \\
w_q(k+1) = w_q(k) + \eta(k+1)(U_q^{(Q)}(k))^{\beta}(\hat{x}_k^{(Q)} - w_q(k)) \;\; \forall q = 1, 2, ..., m, \\
\mu_q^{(\tau+1)} = \dfrac{\sum\limits_{p=1}^{k}(U_q^{(\tau+1)}(p))^{\beta}\left\|\hat{x}_k^{(\tau)} - w_q(k)\right\|^2}{\sum\limits_{p=1}^{k}(U_q^{(\tau+1)}(p))^{\beta}}
\end{cases} \tag{3}
$$

and of the algorithm (2) –

$$
\begin{cases}
U_q^{(\tau+1)}(k) = \dfrac{1}{1+\left(\dfrac{\left\|\hat{x}_k^{(\tau)} - w_q(k)\right\|^2}{\mu_q^{(\tau)}}\right)^{\frac{1}{\beta-1}}}, \\
\text{where } \hat{x}_{ki}^{(\tau)} = w_{qi}^{(\tau)}(k), \;\; w_{qi}^{(\tau)}(k) = \arg\min\limits_{q}\{D_P^2(\tilde{x}_k, w_1^{(\tau)}(k)), ..., D_P^2(\tilde{x}_k, w_m^{(\tau)}(k))\}, \\
w_q^{(\tau)}(k+1) = w_q^{(\tau)}(k+1) + \eta(k+1)(U_q^{(Q)}(k))^{\beta}(\hat{x}_{k+1}^{(\tau)} - w_q^{(\tau)}(k+1)) \;\; \forall q = 1, 2, ..., m, \\
\mu_q^{(\tau+1)} = \dfrac{\sum\limits_{p=1}^{k}(U_q^{(\tau+1)}(p))^{\beta}\left\|\hat{x}_k^{(\tau+1)} - w_q^{(\tau+1)}(k)\right\|^2}{\sum\limits_{p=1}^{k}(U_q^{(\tau+1)}(p))^{\beta}}.
\end{cases} \tag{4}
$$

From a computational point of view the possibilistic procedures are more cumbersome, however, their advantages connects with the fact that on their basis it's easy to organize the process of new clusters finding.

Experiments

Experimental research conducted on two standard samples of data such as Wine and Iris of UCI repository [5].

To estimate the quality of the algorithm we have used quality partitioning criteria into clusters such as: Partition Coefficient (PC), Classification Entropy (CE), Partition Index (SC), Separation Index (S), Xie and Beni's Index (XB), Dunn's Index (DI).

We also compared the results of proposed algorithms with other well-known ones such as Fuzzy C-means (FCM) clustering algorithm and Gustafson-Kessel clustering algorithm.

As seen from the experimental results (Table 1, Table 2 and Table 3), the proposed algorithms have shown better results than the FCM and Gustafson-Kessel clustering algorithm.

Conclusions

The problem of adaptive probabilistic and possibilistic fuzzy clustering with missing data that are fed for processing in the on-line mode, based on the strategy of the nearest prototype-centroid is considered. The developed approach differs by conventional ones from computational simplicity, and the operation can be organized using Kohonen self-organizing maps.

Table 1

Results of experiments with 10 missing values

Algorithms	gaps	Iris UCI repository							Wine UCI repository					
		PC	**CE**	**SC**	**S**	**XB**	**DI**	**PC**	**CE**	**SC**	**S**	**XB**	**DI**	**PC**
Adaptive fuzzy possibilistic clustering data of the nearest centroid with missing values	10	9,1249e-07	**-4,6617e-04**	0,3733	48,7067	**48,8056**	**0,4010**	**1,1640e-13**	**-4,5675e-04**	**7,3872e+05**	2,7115e+08	**2,7180e+08**	**0,0218**	9,1249e-07
FCM		**0,7617**	0,4283	0,0143	**1,4946e-04**	3,8569	0,0275	0,7908	0,3806	7,3348e-04	**6,8417e-06**	5,7110	0,0117	**0,7617**
Gustafson-Kessel		0,9462	0,1145	**0,4789**	0,0032	3,4618	0,3398	0,5507	0,6393	8,5933	0,0483	1,0750	0,1015	0,9462

Table 2

Results of experiments with 50 missing values

Algorithms	gaps	Iris UCI repository							Wine UCI repository					
		PC	**CE**	**SC**	**S**	**XB**	**DI**	**PC**	**CE**	**SC**	**S**	**XB**	**DI**	**PC**
Adaptive fuzzy possibilistic clustering data of the nearest centroid with missing values	50	9,1249e-07	**-4,6617e-04**	0,3775	48,7067	**48,8301**	**0,3365**	**7,5181e-12**	**-5,4992e-04**	**1,1834e+04**	4,1981e+06	**4,2024e+06**	**0,0240**	9,1249e-07
FCM		**0,7399**	0,4632	0,0174	**1,8345e-04**	4,4887	0,0355	0,7892	0,3838	7,6110e-04	**7,1760e-06**	8,8618	0,0237	**0,7399**
Gustafson-Kessel		0,9422	0,1177	**0,5219**	0,0035	3,4413	0,3341	0,5824	0,6010	4,7678	0,0268	1,1703	0,1030	0,9422

Table 3

Results of experiments with 100 missing values

Algorithms	gaps	Iris UCI repository							Wine UCI repository					
		PC	**CE**	**SC**	**S**	**XB**	**DI**	**PC**	**CE**	**SC**	**S**	**XB**	**DI**	**PC**
Adaptive fuzzy possibilistic clustering data of the nearest centroid with missing values	100	**1,7429e-06**	**-4,6617e-04**	0,1089	25,5	**25,7608**	**0,2277**	**2,1059e-11**	**-5,7912e-04**	**4,3587e+03**	1,4987e+06	**1,5006e+06**	**0,0336**	**1,7429e-06**
FCM		0,8161	0,3536	0,0091	**9,2846e-05**	3,6533	0,0575	0,7798	0,3973	7,6412e-04	**7,3406e-06**	9,2453	0,0330	0,8161
Gustafson-Kessel		0,8454	0,2750	**0,8994**	0,0060	2,9395	0,1392	0,5959	0,5854	7,1943	0,0404	1,2627	0,0904	0,8454

1. Marwala T. Computational Intelligence for Missing Data Imputation, Estimation, and Management: Knowledge Optimization Techniques. Hershey-New York, Information Science Reference, 2009. 2. Hathaway R.J., Bezdek J.C. Fuzzy c-means clustering of incomplete data. IEEE Trans. on Systems, Man, and Cybernetics, №5, 31, 2001, P. 735-744. 3. Kohonen T. Self-Organizing Maps. Berlin: Springer-Verlag, 1995. 4. Krishnapuram R., Keller J.M. A possibilistic approach to clustering. Fuzzy Systems, 1993, 1, №2, P.98–110. 5. Center for Machine Learning and Intelligent Systems, http://cml.ics.uci.edu.