FUZZY CLUSTERING

RESUMINDO OS ARTIGOS LIDOS ATÉ AQUI (12/01)

FUZZY CLUSTERING WITH LEARNABLE CLUSTER DEPENDENT KERNELS

OUIEM BCHIR

HICHEM FRIGUI

***Abstract*—We propose a new relational clustering approach, called Fuzzy clustering with Learnable Cluster dependent Kernels (FLeCK), that learns multiple kernels while seeking compact clusters. A Gaussian kernel is learned with respect to each cluster. It reflects the relative density, size, and position of the cluster with respect to the other clusters. These kernels are learned by optimizing both the intra-cluster and the inter-cluster similarities. Moreover, FLeCK is formulated to work on relational data. This makes it applicable to data where objects cannot be represented by vectors or when clusters of similar objects cannot be represented efficiently by a single prototype. The experiments show that FLeCK outperforms several other algorithms. In particular, we show that when data include clusters with various inter and intra cluster distances, learning cluster dependent kernel is crucial in obtaining a good partition.**

fuzzy clustering techniques have been shown to be suitable to describe real situations with overlapping boundaries

While for object data representation, each object is represented by a feature vector, for the relational representation only information of how two objects are related is available. Relational data representation is more general in the sense that it can be applied when only the degree of (dis)similarity between objects is available or when groups of similar objects cannot be represented efficiently by a single prototype.

Although FCM based approaches are quite efficient, they are not suitable for all types of data, and cannot handle clusters of different sizes and densities effectively [8]. Another major drawback of FCM based approaches is that they cannot separate clusters that are non-linearly separable in the input space.

The Gaussian similarity function, used in kernel and spectral clustering, is defined as

$$\mathbf{S}_{jk} = exp\left(-\frac{dist\left(\mathbf{x}_j, \mathbf{x}_k\right)}{\sigma^2}\right) \tag{1}$$

where $dist$ measures the dissimilarity between patterns and σ controls the rate of decay of S. The scaling parameter σ is commonly set manually.

KERNEL PARAMETER OPTIMIZATION IN STRETCHED KERNEL-BASED FUZZY CLUSTERING

CHUNHONG LU
ZHAOMIN ZHU
XIAOFENG GU

Abstract. Although the kernel-based fuzzy c-means (KFCM) algorithm utilizing a kernel-based distance measure between patterns and cluster prototypes outperforms the standard fuzzy c-means clustering for some complex distributed data, it is quite sensitive to selected kernel parameters. In this paper, we propose the stretched kernel-based fuzzy clustering method with optimized kernel parameter. The kernel parameters are updated in accordance with the gradient method to further optimize the objective function during each iteration process. To solve the local minima problem of the objective function, a function stretching technique is applied to detect the global minimum. Experiments on both synthetic and real-world datasets show that the stretched KFCM algorithm with optimized kernel parameters has better performance than other algorithms.

applies Euclidean distance measure between objects and prototypes can obtain good clustering results for spherically-structured data, but cannot obtain effective clustering analysis for some complex distributed data such as the mixture structure with heterogeneous cluster prototypes and non-spherical geometry of data. The kernel-based fuzzy c-means (KFCM) [2] algorithm was then presented to overcome this drawback. A kernel function is defined to transform nonlinear distributed data to the higher dimensional feature space so that the naturally distributed data can be partitioned linearly. Obviously, the KFCM algorithm can improve the results of FCM algorithm by selecting appropriate kernel function and reasonable parameters. Due to its superiority to other kernel functions, the RBF kernel function has been employed in all four kernel clustering algorithms [3], which is thus also chosen in this work.

There are several examples of a kernel function. Using the RBF kernel as a kernel function in this paper, σ^2 as the variance parameter, we have

$$K(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}. \tag{2}$$

Stretched kernel-based fuzzy c-means algorithm with optimal kernel parameter (SKFCM-opt σ)

Step 1. Set learning rate θ, the maximum iteration number n, stopping criterion ε, initial iteration k=0;γ_1>0, γ_2>0 and τ to a very small positive number.

Step 2. Initialize fuzzifier $m > 1$, usually set to 2, fuzzy partition U, prototypes v_j, kernel parameter σ_0 using Eq. (15).

Step 3. Update the memberships, cluster center based on formula (7) and (8); update kernel parameter σ according to Eq. (12) and $\sigma > 0$ must be satisfied during the iteration process. It is assumed that the kernel width exceeds zero at each iterating. If it is close to zero or a negative number, just giving 2ε to it in order to avoid the risk of degeneracy.

Step 4. Calculate the value difference of the objective function between consecutive iterations. Once a local minimizer is found, update the objective function J using newly-obtained J according to Eqs. (13) and (14).

Step 5. Repeat the total iteration process until termination criteria satisfied or maximum iterations reached.

Step 6. Select one minimizer that yields the best optimal solution of formula (5) and the minimizer is regarded as the best optimal value.

GAUSSIAN KERNEL-BASED FUZZY CLUSTERING WITH AUTOMATIC BANDWITH COMPUTATION

FRANCISCO C. LUCAS S. MARCELO F.

Abstract. The conventional Gaussian kernel-based fuzzy c-means clustering algorithm has widely demonstrated its superiority to the conventional fuzzy c-means when the data sets are arbitrarily shaped, and not linearly separable. However, its performance is very dependent on the estimation of the bandwidth parameter of the Gaussian kernel function. Usually this parameter is estimated once and for all. This paper presents a Gaussian fuzzy c-means with kernelization of the metric which depends on a vector of bandwidth parameters, one for each variable, that are computed automatically. Experiments with data sets of the UCI machine learning repository corroborate the usefulness of the proposed algorithm.

The most popular kernel function in applications is the Gaussian kernel. In general, this kernel function provides effective results and requires the tuning of a single parameter, that is, the bandwidth parameter [4]. This parameter is tuned once and for all, and it is the same for all variables. Thus, implicitly the conventional Gaussian kernel fuzzy c-means assumes that the variables are equally rescaled and, therefore, they have the same importance to the clustering task. However, it is well known that some variables have different degrees of relevance while others are irrelevant to the clustering task [7,11,17,20].

The main contribution of this paper is to provide a Gaussian kernel c-means fuzzy clustering algorithms, with both kernelization of the metric and automated computation of the bandwidth parameters using an adaptive Gaussian kernel. In these kernel-based fuzzy clustering algorithm, the bandwidth parameters change at each algorithm iteration and differ from variable to variable. Thus, these algorithms are able to rescale the variables differently and thus select the relevant ones for the clustering task.

Algorithm 1. KCM-K and KCM-K-H algorithms

Iput
- $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ (the data set); c (the number of clusters); $\gamma > 0$ (a suitable parameter); T (maximum number of iterations); ϵ (threshold parameter);

Output
- KCM-K-GH and KCM-K-LH: the matrix of prototypes $\mathbf{G} = (\mathbf{g}_1, \ldots, \mathbf{g}_c)$;
- KCM-K-H: the vector of bandwidth parameters $\mathbf{s} = (s_1^2, \ldots, s_p^2)$;
- KCM-K-GH and KCM-K-LH: the matrix of membership degrees $\mathbf{U} = (u_{ki})$ $(1 \leq k \leq n; 1 \leq i \leq c)$.

Initialization
- $t = 0$;
- KCM-K and KCM-K-H: randomly select c distinct prototypes $\mathbf{g}_i^{(t)} \in \mathcal{D}$ $(1 \leq i \leq c)$;
- KCM-K-H: set $\frac{1}{(s_j^{(t)})^2} = (\gamma)^{\frac{1}{p}}$ $(1 \leq j \leq p)$;
- KCM-K: compute the components $u_{ki}^{(t)}$ $(1 \leq k \leq n; 1 \leq i \leq c)$ of the the matrix of membership degrees $\mathbf{U}^{(t)}$ according to Eq. (5);
- KCM-K-H: compute the components $u_{ki}^{(t)}$ $(1 \leq k \leq n; 1 \leq i \leq c)$ of the the matrix of membership degrees $\mathbf{U}^{(t)}$ according to Eq. (13).
- KCM-K: compute $J_{KFCM-K}(\mathbf{G}^{(t)}, \mathbf{U}^{(t)})$ according to Eq. (3);
- KCM-K-H: compute $J_{KFCM-K-H}(\mathbf{G}^{(t)}, \mathbf{s}^{(t)}, \mathbf{U}^{(t)})$ according to Eq. (8).

repeat
- $t = t + 1$;
- **Step 1: representation.**
  - KCM-K: compute the cluster representatives $\mathbf{g}_1^{(t)}, \ldots, \mathbf{g}_c^{(t)}$ using Eq. (4);
  - KCM-K-H: compute the cluster representatives $\mathbf{g}_1^{(t)}, \ldots, \mathbf{g}_c^{(t)}$ using Eq. (9).
- **Step 2: computation of the vector of bandwidth parameters**
  - KCM-K: skip this step;
  - KMC-K-H: compute the vector of bandwidth parameters $\mathbf{s}^{(t)}$ using Eq. (11);
- **Step 3: assignment**
  - KCM-K: compute the components $u_{ki}^{(t)}$ $(1 \leq k \leq n; 1 \leq i \leq c)$ of the the matrix of membership degrees $\mathbf{U}^{(t)}$ according to Eq. (5).
  - KCM-K-H: compute the components $u_{ki}^{(t)}$ $(1 \leq k \leq n; 1 \leq i \leq c)$ of the the matrix of membership degrees $\mathbf{U}^{(t)}$ according to Eq. (13).
- KCM-K: compute $J_{KFCM-K}(\mathbf{G}^{(t)}, \mathbf{U}^{(t)})$ according to Eq. (3).
- KCM-K-H: compute $J_{KFCM-K-H}(\mathbf{G}^{(t)}, \mathbf{s}^{(t)}, \mathbf{U}^{(t)})$ according to Eq. (8).

until
- KCM-K: $|J_{KFCM-K}(\mathbf{G}^{(t)}, \mathbf{U}^{(t)}) - J_{KFCM-K}(\mathbf{G}^{(t-1)}, \mathbf{U}^{(t-1)})| < \epsilon$ or $t > T$;
- KCM-K-H: $|J_{KFCM-K-H}(\mathbf{G}^{(t)}, \mathbf{s}^{(t)}, \mathbf{U}^{(t)}) - J_{KFCM-K-H}(\mathbf{G}^{(t-1)}, \mathbf{s}^{(t-1)}, \mathbf{U}^{(t-1)})| < \epsilon$ or $t > T$.