Neurocomputing 207 (2016) 264–275

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

How many clusters? A robust PSO-based local density model

Hui-Liang Ling [a], Jian-Sheng Wu [a,b,*], Yi Zhou [c,**], Wei-Shi Zheng [a]

[a] School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China
[b] Information Engineering School, Nanchang University, Nanchang 330038, China
[c] Department of Biomedical Engineering, Zhongshan School of Medicine, Sun Yat-sen University, Guangzhou 510080, Guangdong Province, China

ARTICLE INFO

Article history:
Received 18 December 2015
Received in revised form
17 March 2016
Accepted 31 March 2016
Communicated by Ran He
Available online 13 May 2016

Keywords:
Clustering
Estimation of number of clusters
Local density
PSO

ABSTRACT

While most clustering methods assume that the number of data clusters is known, automatically estimating the number of clusters by algorithm itself is still a challenging problem in the data clustering field. In this paper, we aim to develop a novel local and not differentiable clustering method based on Particle Swarm Optimization, which can estimate the number of clusters automatically. In particular, the proposed approach measures the local compactness of each cluster by local density function, pushes the PSO towards maximizing such a compactness, and penalizes the whole procedure to avoid estimating quite a lot of clusters during the evolution. The compactness modeling makes the clustering robust to outliers and noise. In addition, due to the merit of PSO, although kernel trick is used in our modeling, it does not consume too much memory when more and more data are processed. The evaluation on the synthetic dataset and the five publicly available datasets shows that our algorithm can estimate the appropriate number of clusters and outperforms six related state-of-the-art clustering methods that can also estimate the number of clusters.

1. Introduction

Clustering has been broadly used in various scientific and engineering disciplines, including data mining [1,2], document retrieval [3], image segmentation [4], pattern classification [5], biology [6], etc. To date, many algorithms have been developed by researchers [7–10], and most of these clustering methods work with the assumption that the number of clusters is known in advance. However, it is often impracticable for users to have sufficient prior knowledge on the number of clusters. It prevents applying these clustering methods in many scientific areas (e.g. biology [6]). Hence, developing a clustering approach that can estimate the number of clusters is challenging but very necessary.

By far, some clustering algorithms [11–19] have been proposed to address the problem of estimating the number of data clusters. Support Vector Clustering (SVC) [11,12] is inspired by supporting vector machine and has two main steps, including sphere construction and cluster labeling, which is costly for large scale data. Its time cost will be unacceptable even when the size of dataset is about only a thousand. Affinity Propagation (AP) [13] is another clustering algorithm that can estimate the number of clusters automatically. It treats all data points as possible exemplars and exchanges availability and self-responsibility messages between any two of them iteratively until a high-quality set of exemplars emerges [13]. However, it is very sensitive to parameter settings and the parameter 'preference' is hard to locate while oscillations cannot be eliminated automatically if occur [20]. In addition, it is also sensitive to outliers. Although the approach [17] is robust to the outliers and noise in data, it has to re-partition the dataset $k-1$ times using the commonly used clustering algorithm, such as K-means, with the cluster numbers increasing 1 each time, where k is the maximum number of clusters and usually is set to be a large number, and its performance is sensitive to the value k. As for SCAMS [15], it is computationally expensive, and how to apply DBSCAN [14] to high dimensional data is a problem left to solve.

Most of the above methods are induced from analytic mathematical models, in which the optimal solution can be computed analytically. However, sometimes analytic model is hard to model complex criterion that cannot be solved analytically. Evolutionary Computation [21] is a family of bio-inspired algorithms that can deal with the problem properly without complex computation operators. Hence, it is promising to consider the use of Evolutionary Computation for intelligent computing in clustering. Recently, some evolutionary computation methods like genetic algorithm (GA) and particle swarm optimization (PSO) have been applied in clustering [22–34]. Many of them cannot estimate the number of clusters automatically. Although the algorithms in [24,25,29,30] can find the number of clusters automatically, the

* Corresponding author at: Information Engineering School, Nanchang University, Nanchang 330038, China. Tel.: +86 20 84110175.
** Corresponding author.
E-mail addresses: linghl@mail2.sysu.edu.cn (H.-L. Ling), jiansheng4211@gmail.com (J.-S. Wu), zhouyi@mail.sysu.edu.cn (Y. Zhou), wszheng@ieee.org (W.-S. Zheng).

http://dx.doi.org/10.1016/j.neucom.2016.03.071
0925-2312/

estimation of the number of clusters in those algorithms is sensitive to the noise and outliers in dataset, which will be also shown on synthetical dataset in our experiments (see Section 4.2). ACDE [33] is computationally expensive and may not work well in practice. MDPSO [34] suffers from premature convergence due to lack of divergence.

In summary, the weaknesses of SVC, SCAMS and DBSCAN lie in the speed issue and for AP it is the parameter sensitivity. For the aforementioned PSO and GA-based clustering methods which can estimate the number of clusters automatically, they are sensitive to the noise and outliers in data.

This work proposes a *PSO-based Local Density Clustering method* (PLDC). Particle Swarm Optimization is an optimization method belonging to Evolutionary Computation, which employs ideas from biological evolution to solve computational problems in an intelligent way. It is a population based stochastic optimization technique for continuous nonlinear functions [35]. In the proposed PLDC, a set of cluster centers is identified by translating the clustering problem into a multi-modal optimization problem, that is to search a set of cluster center candidates which are characterized by a local energy fitness function. The distance-based locally informed particle swarm algorithm (LIPS) [36] is used for the multi-modal optimization. Then a cluster center selection mechanism is introduced to select cluster centers. These procedures are performed in feature space in the proposed algorithm. As the local density around the outliers is extremely small, PLDC can easily detect those outliers, thus getting a robust and precise estimation of the number of clusters. The experiments on our synthetic dataset and other publicly available datasets of images and texts show that the proposed PLDC outperforms the state-of-the-art clustering methods including the Genetic algorithm and Particle Swarm Optimization based approaches [30,24,31,32]. In addition, the proposed clustering is a nonlinear clustering method and it does not need to store any kernel matrix, thus avoiding the requirement of large scale memory.

Compared to existing methods, the contribution of this paper is to propose a robust PSO-based clustering method, which adopts local density of data to measure the compactness of clusters and can estimate the number of clusters automatically. It is robust to noise, therefore getting much better clustering of data, i.e., much higher clustering accuracy and more accurate estimation of the number of clusters, than many other state-of-the-art clustering methods. In addition, the time and storage cost are not high in our algorithm.

In the remainder of this paper, we first introduce Particle Swarm Optimization and algorithm in Section 2. Then we present the proposed algorithm in Section 3. We compare the proposed algorithm with related state-of-the-art clustering methods and analyze the experimental results in Section 4. Finally, we conclude the paper in Section 5.

2. Preliminary

Before introducing the methods proposed in our proposal, we will first introduce some notations used in this paper and the background. The frequently used notations are listed in Table 1 and the background are introduced below:

2.1. Particle Swarm Optimization

Particle Swarm Optimization (PSO) [37] is a population-based stochastic search process that is originally introduced by Kennedy and Eberhart. It has been successfully applied in machine learning tasks, such as clustering [25–28] and classification [38]. The task of PSO is to optimize a fitness function by searching a set of particles that represent potential solutions to the fitness function. By iteratively improving the solutions during evolution, PSO gradually approaches the optimum of the specified fitness function. In each generation, particles move around in the search space by utilizing the intelligence of the whole swarm according to some moving rules and gradually move towards the optimum.

Table 1
Notations.

Notation	Definition
N	Size of dataset
d	Dimension of data
N_P	Size of population
X	Data matrix consisting of N data points in d-dimensional space. Each row is a data sample
Y	$N_P \times d$ matrix consisting of N_P particles (solutions) in d-dimensional space
V	$N_P \times d$ velocity matrix correspondent to Y
pBest	$N_P \times d$ matrix consisting of the best solution each particle has achieved so far
gBest	The Best solution of *pBest* in each iteration

In detail, at the start of the algorithm, a population Y and its correspondent velocity V are initialized randomly in general. A PSO contains four steps which are conducted iteratively:

(1) For given matrices Y and V, let Y_i and V_i denote the i-th row of Y and V. Then we update Y as

$$Y_i \leftarrow Y_i + V_i. \tag{1}$$

If any solutions have invalid value to the specific problem, refine them.

(2) Evaluate population Y according to a specific fitness function.

(3) For a given matrix *pBest*, let $pBest_i$ denote the i-th row of *pBest*. Then *pBest* is updated as

$$pBest_i \leftarrow \text{argmax}\{fitness(Y_i), fitness(pBest_i)\}. \tag{2}$$

And *gBest* is updated to be the solution in *pBest* with the highest fitness value.

(4) After updating the population Y, V is updated as

$$V_i = \omega V_i + c_1 r_1 (pBest_i - Y_i) + c_2 r_2 (gBest_i - Y_i), \tag{3}$$

where c_1 and c_2 are the acceleration constants, ω is the inertia weight to balance the global and local search performance, and r_1 and r_2 are two random numbers lying in $[0, 1]$. Repeat the above four steps iteratively until the termination condition is met, e.g. the maximum generation is reached or the population converges. A brief flow chart of PSO is shown in Fig. 1.

2.2. Distance-based locally informed particle swarm algorithm

Multi-modal optimization, which is different from single-modal optimization, amounts to finding multiple global and local optima of a function. By far, niching methods are widely used for solving multi-modal optimization problems [39–41]. However, they need to specify certain niching parameters, which depend on optima distribution of the fitness function and are hard to know in advance, thus limiting their application to practice problems. Distance-based locally informed particle swarm algorithm (LIPS) [36] is another multi-modal optimization approach. It can eliminate the requirement of niching parameters and has better search ability. Compared to the classical PSO algorithms, every particle in LIPS adopts local information from its nearest neighborhood measured in terms of Euclidean distance and then updates the velocity in a different way by

$$V_i = \omega(V_i + \varphi(P_i - X_i)), \tag{4}$$