Pattern Recognition 44 (2011) 2197–2209

Contents lists available at ScienceDirect

Pattern Recognition

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

An efficient hyperellipsoidal clustering algorithm for resource-constrained environments

Masud Moshtaghi [a,*], Sutharshan Rajasegarar [b], Christopher Leckie [a], Shanika Karunasekera [a]

[a] NICTA Victoria Research Laboratories, Department of Computer Science and Software Engineering, University of Melbourne, Melbourne, Australia
[b] Department of Electrical and Electronic Engineering, University of Melbourne, Melbourne, Australia

ARTICLE INFO

Article history:
Received 23 September 2010
Received in revised form
4 February 2011
Accepted 7 March 2011
Available online 15 March 2011

Keywords:
HyCARCE
Data clustering
Hyperellipsoidal clustering
Wireless sensor networks
Low computational cost clustering algorithm

ABSTRACT

Clustering has been widely used as a fundamental data mining tool for the automated analysis of complex datasets. There has been a growing need for the use of clustering algorithms in embedded systems with restricted computational capabilities, such as wireless sensor nodes, in order to support automated knowledge extraction from such systems. Although there has been considerable research on clustering algorithms, many of the proposed methods are computationally expensive. We propose a robust clustering algorithm with low computational complexity, suitable for computationally constrained environments. Our evaluation using both synthetic and real-life datasets demonstrates lower computational complexity and comparable accuracy of our approach compared to a range of existing methods.

1. Introduction

Clustering algorithms have been extensively applied to a wide variety of domains, such as marketing, information retrieval and image segmentation. These application domains can differ in terms of characteristics that include the dimensionality of the input data, the volume of data to be clustered and any resource constraints for clustering. Consequently, various clustering algorithms have been proposed in the literature, each targeting specific types of applications [1,2]. The focus of this paper is on clustering algorithms that can be used in computationally constrained environments such as wireless sensor nodes.

A good clustering algorithm should have a small number of input parameters and it should be reasonably insensitive to changes in these parameters. Furthermore, the ability to select these parameters with a minimum of knowledge about the dataset, such as the approximate number of records or the approximate range of each field, is also important. Many clustering algorithms require knowledge of the number of the clusters in advance, e.g., K-means [1] and its derivatives. Generally, this knowledge is not available prior to clustering. Other types of clustering algorithms, like density-based and grid-based algorithms, tackle this problem of finding the number of clusters as a part of the clustering algorithm. However, they introduce additional parameters such as the initial grid size or effective density radius, and the results of clustering can be very sensitive to the choice of these parameters. There are other features that we might expect from a clustering algorithm, for example, resilience to noise and automatic detection of the cluster boundaries. These features are necessary to ensure the successful application of clustering in standalone systems.

Clustering as a knowledge discovery tool can provide a small device such as a PDA or a wireless sensor node with decision making capabilities. Clustering can also be used to characterize the distribution of data in an explicit form to reduce the amount of data that needs to be communicated between the nodes in a mobile Ad hoc network (MANET) or a wireless sensor network (WSN) where communication constraints exist [3]. The application of clustering algorithms is also a challenge in environments with limited computational capabilities. In this paper, we present a clustering algorithm with low computational complexity to suit computationally constrained agents, for example, a node in a wireless sensor network. Although multiple parameters control the behavior of the proposed algorithm, only one input parameter needs to be set by the user, i.e., the initial grid cell size. We propose an automated method of setting the other input parameters of the algorithm.

The main characteristics of the proposed clustering algorithm are (1) automatic selection of the number of clusters; (2) low computational cost ($O(N)$); (3) explicit cluster boundary detection; (4) and embedded outlier detection. Our results demonstrate that the proposed clustering algorithm has lower computational complexity and better or comparable accuracy to a range of well-known clustering algorithms. In the next section we summarize the related work in this field. In Section 3, we present our problem formulation. We then present our new clustering algorithm called hyperellipsoidal

* Corresponding author.
E-mail address: mosm@unimelb.edu.au (M. Moshtaghi).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.03.007

Table 1
Comparison of different clustering algorithms (N is the number of data points, K is the number of clusters, d is the dimensionality of the data, c is a constant).

Algorithm name	Methodologies used	Computational cost	Automatic boundary detection	Predefined number of clusters	Embedded outlier detection
K-means	Partitioning method/minimizing SSE	$O(NKd)$	No	Yes	No
Gustafson–Kessel	Fuzzy partitioning method (extended fuzzy c-means)	Considered $O(N^2)$	No	Yes	No
Subtractive clustering	Density based (extension of the mountain method)	$O(N^2)$	Yes	No	Yes
CLARANS	Graph theory/randomized search	Considered Quadratic	No	Yes	No
DBSCAN	Density-based	$O(N\log N)$	No	No	Yes
DENCLUE	Density-based/Kernel density estimate	$O(N\log N)$	No	No	Yes
MAFIA	Grid/density based	$O(Nd+c^d)$	Yes	No	No
K-windows	Grid/iterative optimization	$O(N\log^{d-1}N)$	Yes	No	No

clustering for resource-constrained environments (HyCARCE) in Section 4. Section 5 presents an empirical evaluation of our approach, followed by a discussion on how to choose the parameters for HyCARCE in Section 6 and a summary of our conclusions in Section 7.

2. Related work

Clustering algorithms can be divided into the following categories: (1) partitioning methods, (2) hierarchical methods, (3) graph-based methods, (4) density-based methods, and (5) grid-based methods. A more detailed classification of the algorithms can be found in [1,2]. In this paper, we focus on clustering algorithms that have low computational complexity, so that they can be used in constrained environments.

Hierarchical clustering algorithms including single-linkage, complete-linkage and average linkage, all have a complexity of $O(N^2)$ in both time and space, which usually is considered very high in constrained environments. In [4], the authors proposed a clustering algorithm based on the minimum volume ellipsoid (MVE). The MVE estimator is applied iteratively to find the hyperellipsoidal-shaped cluster that best matches a unimodal Gaussian distribution using the Kolmogorov–Smirnov test. This iterative algorithm has high computational complexity [5]. K-means, K-medoid and EM-based algorithms, like the competitive hyperellipsoidal clustering algorithm in [6], can be categorized as partitioning methods. Among these algorithms, only K-means has complexity close to $O(N)$ in both space and time. However, it is very sensitive to the initial number of clusters chosen and it yields spherical clusters which, in many cases, cannot flexibly represent the input data. The clustering algorithm proposed in [5] uses regularized Mahalanobis distance to find K clusters in the data by using a competitive neural network. Similar to K-means, this algorithm is also very sensitive to the initial seed and predefined number of clusters. Partitioning algorithms try to minimize a function of the sum of the errors, usually the sum of squared errors (SSE), which is usually defined by the inter cluster distance. Another group of clustering algorithms use graph theoretic methods to partition the input space to find the clusters. For example, CLARANS [7], which was originally developed for spatial databases, uses randomized search in a graph to find the clusters. The whole process of this algorithm still can be considered quadratic in time.

Another approach to clustering is density-based clustering. In this approach clusters are defined as dense regions in the input space and kernel density estimates are used to find the clusters. Two pioneers in this approach are the mountain method [8] and its extension, the subtractive approach proposed in [9]. The computational complexity of subtractive clustering is $O(N^2)$. Two well-known examples of density-based clustering algorithms with lower computational complexity are GDBSCAN [10] and DENCLUE [11,12]. One of the drawbacks of these algorithms is that they usually do not give good clustering results when the input space consists of a mixture of dense and sparse regions. DENCLUE can achieve a good computational complexity of $O(N\log N)$ in time. However, it requires two input parameters and the result of the algorithm is very sensitive to one of these parameters. This algorithm can detect clusters of arbitrary shape but does not give a simple well-defined boundary for the clusters.

The next group of algorithms try to reduce the computational cost by imposing a grid structure on the data. MAFIA [13] uses a grid structure in conjunction with a density-based approach to find the clusters. The computational cost of this algorithm is $O(Nd+c^d)$ where c is a constant and d is the number of dimensions in the data. This algorithm has low computational cost, especially in low dimensions, but has the same drawbacks as the density-based approaches. Another algorithm is K-windows and its extensions [14,15], which were initially proposed as a basis for K-means to estimate the number of clusters. This algorithm is $O(N\log^{d-1}N)$ in time and space, where d is the number of the dimensions. This algorithm can work well in low resource environments, especially when the data has low dimensionality. The main drawback of this algorithm is using the d-dimensional boxes, which can only represent data distributions of shapes like hyper-cubes or hyper-spheres. A summary overview of the main clustering algorithms discussed above is shown in Table 1.

3. Problem statement

Our aim is to find a set of clusters $C=\{c_j : j=1\dots K\}$ in a dataset of N records $S=\{s_k : k=1\dots N\}$, where each record $s_k \in \Re^d$ is a vector of d-dimensions. Each cluster c_j is represented by a hyperellipsoid e_j that marks the boundary of the cluster. A general form of a hyperellipsoidal boundary is a set of points X in $\Re^d$ that satisfy

$$(X-m)^T A(X-m)=1 \tag{1}$$

where m is the center of the hyperellipsoid and A is a $d\times d$ symmetric positive definite matrix called the characteristic matrix, whose eigenvectors specify the principal directions of the hyperellipsoid, and the inverse of the square root of its eigenvalues is the corresponding equatorial radii. The reason that we have chosen ellipsoids to represent our clusters is that they have the flexibility to