Pattern Recognition 48 (2015) 918–932

Contents lists available at ScienceDirect

Pattern Recognition

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

A simple statistics-based nearest neighbor cluster detection algorithm

Gerhard X. Ritter [a], José-A. Nieves-Vázquez [a], Gonzalo Urcid [b,*]

[a] CISE Department, University of Florida, Gainesville, FL 32611, USA
[b] Optics Department, INAOE, Tonantzintla, Pue 72000, Mexico

ARTICLE INFO

Article history:
Received 21 February 2014
Received in revised form
8 August 2014
Accepted 3 October 2014
Available online 29 October 2014

Keywords:
Clusters
Cluster detection
Clustering
Cluster analysis
Digital geometry
Nearest neighbors
Neighborhoods
Statistics
Pattern recognition

ABSTRACT

We propose a new method for autonomously finding clusters in spatial data. The proposed method belongs to the so called nearest neighbor approaches for finding clusters. It is a repetitive technique which produces changing averages and deviations of nearest neighbor distance parameters and results in a final set of clusters. The proposed technique is capable of eliminating background noise, outliers, and detection of clusters with different densities in a given data set. Using a wide variety of data sets, we demonstrate that the proposed cluster seeking algorithm performs at least as well as various other currently popular algorithms and in several cases surpasses them in performance.

1. Introduction

Cluster analysis divides data into groups that are useful for specific applications. These groups are called *clusters* and the data points in a given cluster are in some sense similar. Similarity of data objects may be defined in terms of color, statistics, spectral values, and a host of other features. Some excellent summaries can be found in [22,10,24,27,25], and [39,1] with the last reference devoting six chapters to cluster analysis. These sources demonstrate that there is no single optimal cluster detection algorithm but a plethora of methods for cluster detection. A scanning of current cluster seeking algorithms available in the open literature makes it clear that cluster detection is still an experiment oriented endeavor in the sense that the performance of a given algorithm is not only dependent on the type of data being analyzed, but is also strongly influenced by the chosen measure of pattern similarity as well as the method used for identifying clusters in the data. For example, suppose we have a set of objects $\mathcal{O}$ specified by a sequence $p_1, \ldots, p_n$ of properties or attributes such as a specific color, shape, and weight range with different objects lacking different properties. Such a set is often transformed into a set of binary vectors $X \subset \mathbb{R}^n$, where $\mathbf{x} = (x_1, \ldots, x_n) \in X$ is defined by $x_i = 1$, if and only if, p_i holds else $x_i = 0$. In this situation, two objects $\mathbf{x}, \mathbf{y} \in X$ are viewed as *similar* if they share a large majority of properties or attributes. Suppose X contains the elements $\mathbf{w}, \mathbf{x}, \mathbf{y}$, and $\mathbf{z}$ given by: $w_i = 1$ if $i = 1$ else $w_i = 0$, $x_i = 1$ if $i = n$ else $x_i = 0$, $y_i = 0$ if $i = 1$ else $y_i = 1$, and $z_i = 0$ if $i = n$ else $z_i = 1$. Employing the Euclidean metric, one obtains $d(\mathbf{w}, \mathbf{y}) = n$, which can be very large in some settings. This shows that $\mathbf{w}$ and $\mathbf{y}$ are spatially far apart when viewed as points in a n-dimensional Euclidean space. This can also be interpreted that the two vectors are very dissimilar as they have no common attributes. However, we also have $d(\mathbf{w}, \mathbf{x}) = \sqrt{2} = d(\mathbf{y}, \mathbf{z})$ even though $\mathbf{w}$ and $\mathbf{x}$ share no attributes and are, therefore, totally dissimilar while $\mathbf{y}$ and $\mathbf{z}$ are very similar. Thus, the Euclidean metric provides little information when used as a clustering tool for this type of data. Likewise, the L_∞ metric is of little use in cluster analysis of binary data since $d(\mathbf{x}, \mathbf{y}) = \bigvee_{i=1}^{n} |x_i - y_i| = 1$ for all distinct pairs $\mathbf{x}, \mathbf{y} \in X$, where X is an n-dimensional binary data set and $\bigvee$ denotes the maximum.

It is pertinent to note that some researchers test their clustering algorithms on well known data sets that are commonly used in machine learning and training of artificial neural networks for pattern classification which, although related, differs from the subject of cluster detection and cluster analysis. A typical example is the 4-dimensional Iris data set consisting of three classes, with each class corresponding to a distinct species of the genus Iris [17,10]. Four features, specific to a given species, are described in vector format. Two of the classes are geometrically closely intertwined in 4-space and can be successfully separated with neural network techniques when using the complete data set as training data but fails when

* Corresponding author. Tel.: +52 222 266 3100; fax: +52 222 247 2940.
E-mail addresses: ritter@cise.ufl.edu (G.X. Ritter), gurcid@inaoep.mx (G. Urcid).

http://dx.doi.org/10.1016/j.patcog.2014.10.009
0031-3203/

using 50% and even 60% of the data for training [34]. The problem is that the two intertwined sets do not form two well defined spatial clusters that can be determined using current clustering techniques. For this reason we do not consider many of the standard data sets that are commonly used in pattern classification tasks for evaluating performance of cluster seeking algorithms.

In our approach we view clusters in terms of their spatial arrangement and distribution by using the old adage that "birds of a feather will flock together." For instance, when observing migrating cranes one sees beautiful V shaped formations, while blackbirds will flock into cloud shaped 3-dimensional globular clusters. Several migrating species of birds form huge rotating 3D doughnut or spiral shaped clusters before assuming a single line or a V shaped formation. In his seminal paper entitled *Data Clustering*: *50 years beyond K-means*, A.K. Jain points out that there are no cluster algorithms available that are able to detect all seven clusters shown in Fig. 1 even though these clusters are readily apparent to a human data analyst [25]. The problems raised by Jain's example are manifold. First, there is the issue of the noisy background that is interspersed with the data clusters. Next, the two globular clusters on the left side of the figure have different densities. Finally, the well defined geometric pattern clusters on the right side have cluster center problems. The circular clusters share the same geometric cluster center, while the center for one of the two spiral cluster may be located inside or closer to the other spiral. These clusters are troublesome for various center based approaches to clustering.

The basic idea underlying center based approaches is to group a set $X \subset \mathbb{R}^n$ of feature vectors into K clusters using an appropriate similarity measure for comparison with the cluster's center. Generally, this measure is the distance between the feature vector and the cluster's center and assigns the feature vector $\mathbf{x}^j$ to cluster C_k whenever the distance from $\mathbf{x}^j$ to the cluster's center $\mathbf{c}^k$ is the minimum over all K clusters. The k-means (hard c-means) clustering algorithm, first developed by MacQueen [29], belongs to this group. The algorithm was later modified by Dunn and Bezdek [11,5,6] to include fuzzy c-means clustering and has become one of the most popular and widely used clustering method. Since the number of actual clusters in high dimensional data is generally not known, the initial input value K can critically affect the algorithm's output. Similarly, different initial centroid values usually result in different output and performance. Consequently, various modifications of c-means algorithms have been proposed in order to get around some of these problems [28,43,9,44,33,45,41,47]. The performance of these modifications always improved on the examples given by their authors but still failed when applied to Jain's example as well as other data sets some of which are given in subsequent sections.

Jain's example will yield very mixed results for many clustering algorithms in vogue today. However, it is our opinion that any automatic clustering algorithm worth its salt should be able to find the three clusters shown in Fig. 2.

Here the data set X consists of 192 points; i.e., $X = \{\mathbf{x}^1, \ldots, \mathbf{x}^{192}\} \subset \mathbb{R}^2$. The statistics associated with X are trivial. Every point $\mathbf{x}^j \in X$ has a neighboring point whose distance from $\mathbf{x}^j$ is of unit length. This is true for the Euclidean as well as the chessboard and the city-block distance metric. More specifically, for $j = 1, \ldots, 192$, the number $\tau_j = \bigwedge_{k=1, k\neq j}^{192} d(\mathbf{x}^j, \mathbf{x}^k) = 1$, where d denotes any of the three distances mentioned and $\bigwedge$ denotes the global minimum. Hence, the average nearest neighbor distance and the standard deviation of the nearest neighbor distances are given by $\mu = \sum_{j=1}^{192} \tau_j / 192 = 1$ and $\sigma^2 = \sum_{j=1}^{192} (\tau_j - \mu)^2 / 192 = 0$, respectively.

Nevertheless, when applying either the c-means or the fuzzy c-means algorithm in Matlab and specifying $K=3$, one may not obtain the 3 correct clusters as shown in Fig. 3 unless one provides the *actual* clusters. Even when using the correct number K and selecting randomly each starting point or *seed* in each of the actual clusters may still result in incorrect identification of the true clusters. This happens because cluster detection in data containing clusters of greatly varying sizes and densities remains problematic when applying the various modified c-means techniques cited earlier. The reason for this is that the fundamental building blocks of these modifications are all based on the classical c-means and fuzzy c-means methodology and as such inherit some of the undesirable properties of their predecessors.

For example, using the alternative c-means clustering algorithm proposed by Wu and Yang [43] produces the results shown in Fig. 3(b). Here the fuzzy membership threshold used was set at 0.4; i.e., data points below 0.4 where not assigned to any cluster. One can assign all points to clusters by simply assigning cluster membership to a cluster if the vector's fuzzy membership is less with respect to the other two clusters. In this case the result is shown in Fig. 3(c). The method of randomly assigning data points to three sets of equal size and defining the seeds to be the center points of these sets may result in the three clusters shown in Fig. 3(d). Given the inherent difficulties encountered by the c-means method listed here and elsewhere [4,32], we considered several cluster seeking methods that were not based on the c-means paradigm.

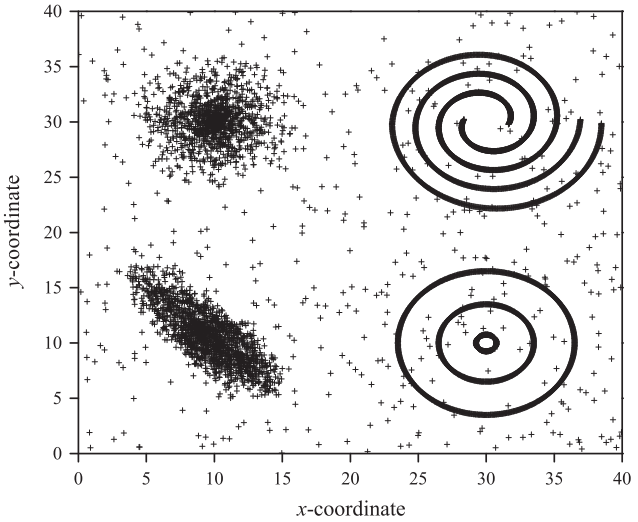

Fig. 1. Seven clusters that differ in shape, size, and density in a noisy background.

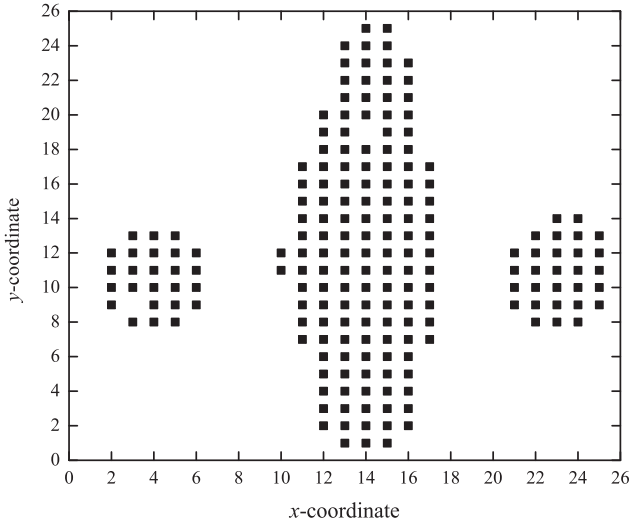

Fig. 2. Three separated globular clusters differing only in size.