Pattern Recognition Letters 52 (2015) 17–24

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

An adaptive neighbourhood construction algorithm based on density and connectivity ☆

Tülin İnkaya [a,*], Sinan Kayalıgil [b], Nur Evin Özdemirel [b]

[a] Industrial Engineering Department, Uludağ University, Görükle, Bursa 16059, Turkey
[b] Industrial Engineering Department, Middle East Technical University, Çankaya, Ankara 06800, Turkey

ARTICLE INFO

Article history:
Received 21 May 2014
Available online 28 September 2014

Keywords:
Data neighbourhood
Gabriel graph
Density
Connectivity
Clustering
Local outlier detection

ABSTRACT

A neighbourhood is a refined group of data points that are locally similar. It should be defined based on the local relations in a data set. However, selection of neighbourhood parameters is an unsolved problem for the traditional neighbourhood construction algorithms such as k-nearest neighbour and ε-neighbourhood. To address this issue, we introduce a novel neighbourhood construction algorithm. We assume that there is no a priori information about the data set. Different from the neighbourhood definitions in the literature, the proposed approach extracts the density, connectivity and proximity relations among the data points in an adaptive manner, i.e. considering the local characteristics of points in the data set. It is based on one of the proximity graphs, Gabriel graph. The output of the proposed approach is a unique set of neighbours for each data point. The proposed approach has the advantage of being parameter-free. The performance of the neighbourhood construction algorithm is tested on clustering and local outlier detection. The experimental results with various data sets show that, compared to the competing approaches, the proposed approach improves the average accuracy 3–66% in the neighbourhood construction, and 4–70% in the clustering. It can also detect outliers successfully.

1. Introduction

Extracting the neighbourhood structures in a data set provides valuable information about density, connectivity and proximity relations. Hence, neighbourhood construction is one of the fundamental issues in data mining for problems such as clustering, outlier detection, and classification [10,13,7]. It is also important in neighbourhood based search algorithms such as Simulated Annealing and Tabu Search [1], and in solving location and routing problems [11,5,16].

There are four main streams of neighbourhood definitions in the literature: k-nearest neighbourhood (KNN) and its variants, distance based approaches, graph based approaches, and artificial neural network based approaches. In KNN a given number (k) of points within the close vicinity of a point form the neighbourhood [27,22]. KNN variants, such as mutual KNN and reverse KNN [34], are introduced to incorporate mutuality relations, however there is still the issue of the proper choice of the parameter, k.

One of the simplest and well-known distance based approaches is the ε-neighbourhood. Points that are closer than a given distance, ε, form the neighbourhood. With inappropriate choice of ε, this definition is prone to generating empty neighbourhoods. Even though there are extensions that suggest the use of varying parameters for each neighbourhood [2], there is not a unified and generally accepted approach to determine these parameters. In order to improve the performance of the distance based neighbourhood, O'Callaghan [29] addresses the symmetric distribution of the neighbours around a point. For this purpose, in addition to the distance parameter, he uses a direction parameter to restrict the angle between the neighbours. However, the number of parameters increases to two in this case. In an alternative scheme, Chaudhuri [9] proposes nearest centroid neighbourhood which is also based on symmetric distribution of the neighbours. He combines the distance information and KNN connectivity. Since the neighbours are located symmetrically, these approaches lead to dissimilarities in the neighbourhoods with varying density.

Graph based approaches define the neighbourhood based on the proximity relations. Proximity graphs such as minimum spanning tree, Delaunay triangulation, Gabriel graph, and relative neighbourhood graph are used for this purpose [38,25,35,14,26]. These proximity graphs generate complete graphs, so additional steps are required to determine and eliminate the inconsistent edges within the neighbourhood. Although this scheme results in improvement for the regions with varying densities, the performance is sensitive to the proper parameter selection.

☆ This paper has been recommended for acceptance by M.A. Girolami.

[*] Corresponding author. Tel.: +90 224 294 2605; fax: +90 224 294 1903.
E-mail address: tinkaya@uludag.edu.tr, tinkaya@gmail.com (T. İnkaya).

http://dx.doi.org/10.1016/j.patrec.2014.09.007
0167-8655/© 2014 Elsevier B.V. All rights reserved.

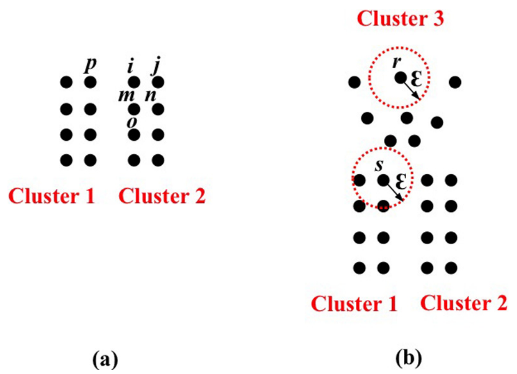

Fig. 1. Example neighbourhoods constructed for the clustering problem. (a) KNN neighbourhood, and (b) ε-neighbourhood.

Artificial neural network based approaches have been dominated by self-organizing map (SOM) [24] due to its visualization capability. SOM projects a high-dimensional data set on the prototype vectors of a low-dimensional grid structure. Each grid corresponds to a neuron. SOM network is trained to assign similar points to the same neuron. Two neighbouring neurons on the grid imply close data points in the original data set. Hence, SOM helps visualize and explore the neighbourhood relations in the data set [17,4,37].

We illustrate the sensitivity of the neighbourhood approaches to the parameters in a clustering problem in Fig. 1. Basically, clustering is forming groups of points based on a measure of similarity so that the similarities between the points inside the same group are high, while the similarities of the points from different groups are low [21]. Neighbourhood includes locally similar points, so a data point and its neighbours are expected to be in the same cluster. In Fig. 1 neighbourhoods constructed using KNN and ε-neighbourhood algorithms are presented on two data sets. We measure the similarity in terms of Euclidean distance. In Fig. 1(a) there are two clusters. The KNN neighbourhood of point i in Cluster 2 includes points from the same cluster, i.e. points j, m, and n, when $k \leq 3$. On the other hand, point p in Cluster 1 is included in the neighbourhood of point i, when $k \geq 4$. Hence, KNN fails to extract the indirect connection between points i and o over point m, and results in a mixed neighbourhood. The data set in Fig. 1(b) has three clusters, and one of them has intracluster density variation. For a fixed radius of ε, the neighbourhood of point s includes points from the same cluster, Cluster 1. However, the neighbourhood of point r is empty, even though it is not an outlier. When we increase the value of ε, point r is no longer an outlier. However, this also increases the neighbourhood of point s and gives rise to neighbourhood mixes from Clusters 2 and 3.

To address such limitations of the neighbourhood approaches in the literature, we propose a locally adaptive neighbourhood construction algorithm, namely NC. We assume that there is no a priori information about the data set. The main idea of NC is the determination of the density based connectivity and proximity relations among the points. These relations are extracted in an adaptive manner using Gabriel graph (GG) [15]. In order to demonstrate the performance of NC, we apply it to clustering and local outlier detection problems. The former problem is to form groups of points based on a measure of similarity, and we propose a procedure to form clusters from the connected neighbourhoods of NC. The latter problem is to determine the points that are outlying relative to their neighbourhoods [7], and we introduce a procedure for local outlier detection using the NC neighbourhoods. Our experimental results with various data sets show the success of NC in clustering and local outlier detection. The major advantages of NC are: (i) it is a parameter-free algorithm, (ii) it produces a neighbourhood unique to each data point, (iii) it avoids generalizations about the neighbourhoods of points, and (iv) it is successful in various data sets with varying densities and outliers.

The rest of the article is organized as follows. The NC algorithm is described in Section 2 in detail. In Section 3 we demonstrate the advantages of the NC algorithm on clustering and local outlier detection problems. We introduce two procedures for this purpose, and present an extensive experimental study about NC. Finally, we conclude in Section 4.

2. Neighbourhood construction (NC) algorithm

2.1. Notation and definitions

We use the notation given below throughout the paper.

D	set of data points
i, j, p	indices for data points
d_{ij}	Euclidean distance between points i and j
CC_i	set of core neighbours of point i
BC_i	set of density connected neighbours of point i
PC_i	set of extended neighbours of point i
CS_i	set of final neighbours of point i

Definition. $\mathrm{B}(i,j,d_{ij})$ is the set of points inside the ball passing through points i and j with diameter d_{ij}.

Definition. Points i and j are **directly connected** by an edge of the GG, if and only if $\mathrm{B}(i, j, d_{ij}) \cap \mathrm{D} = \emptyset$, or equivalently, $d_{ij} \leq \min_p \{\sqrt{d_{ip}^2 + d_{pj}^2} : p \in \mathrm{D}\}$.

Definition. Points i and j are **indirectly connected**, if $\mathrm{B}(i,j,d_{ij}) \cap \mathrm{D} \neq \emptyset$, i.e. there exists at least one point between i and j.

Definition. Density between points i and j, $density_{ij}$, is the number of points lying in $\mathrm{B}(i,j,d_{ij}) \cap \mathrm{D}$.

2.2. Steps of the NC algorithm

The NC algorithm is composed of four consecutive steps outlined in Fig. 2. These steps are explained below and their pseudocodes are given in the supplementary material.

Step 1. Extraction of core neighbours by direct connectivity.
Given a point i as the base point, first, we list all remaining points in D in non-decreasing order of their distance to point i, and we form the ordered set T_i. Let $\mathrm{T}_i[j]$ denote jth member of the ordered set T_i. Following the GG construction, we determine the nearest point with an indirect connection to point i, $\mathrm{T}_i[indirect_i]$. Points that are closer to point i than $\mathrm{T}_i[indirect_i]$ are directly connected to point i. We call such points with a density of zero *core neighbours* of point i, and include them in CC_i. Indirect connections to other points are established via core neighbours.

An example of step 1 is presented in Fig. 3(a)–(c). The ordered set for point 1 is T_1 = {2, 3, 4, 5, 6, 8, 9, 7, 10}. Points 2 and 3 are directly connected to point 1, and the nearest neighbour with indirect connection is point 4. Hence, points 2 and 3 form core neighbour set of point 1, CC_1.

Step 2. Extraction of density connected neighbours by density tracking.
Neighbours of point i can include density connected points. Hence, as one moves to the next member of T_i, density is expected to stay the same or to increase for “true” neighbours of point i. In this step, the nearest point in T_i at which the density starts to decrease is determined. We call this the *break point* of point i, $\mathrm{T}_i[break_i]$. This point may indicate beginning of a different density region. We form BC_i from the points that are closer to point i than the break point. BC_i is a superset of CC_i, and includes points with direct connections as well.

Fig. 3(d) shows an example for step 2. Density values of points in $\mathrm{T}_1 = \{2, 3, 4, 5, 6, 8, 9, 7, 10\}$ are 0, 0, 2, 1, 2, 0, 1, 2, 2,