Using Grid-clustering Methods in Data Classification

Peter Grabusts and Arkady Borisov
Riga Technical University
Decision Support Systems Group
Institute of Information Technology
1 Kalku Street, LV-1658 Riga, Latvia
E-mail:peter@ru.lv, phone: +371 7089530

Abstract

This paper examines grid-clustering method. Unlike the conventional methods, this method organizes the space surrounding the patterns. It uses a multidimensional grid data structure. The resulting block partitioning of the value space is clustered via a neighbor search. The mathematical description of the algorithms employed is given. Some case studies and ideas how to use the algorithms are described.

1. Introduction

In many situations it is possible to model the system behavior qualitatively using the expert's knowledge about the system according to input and output data of the system. The data could be obtained using perfect mathematical modeling or from expert's knowledge or from experimental analysis of the system. As these data can also contain uncertainty, a suitable approach is required so as to include any information about the system. In such situations, the system identification can be done by a clustering technique.

Clustering [1] is one of the most fundamental issues in pattern recognition. It plays a significant role in searching for structures in data. Given a finite set of data X, the problem of clustering in X is to find several cluster centers that can properly characterize relevant classes of X. In classic cluster analysis [2] these classes are required to form a partition of X such that the degree of association is strong for data within blocks of the partition and weak for data in different blocks. However, this requirement is too strong in many practical applications and it is thus desirable to replace it with a weaker requirement.

There are some basic methods of fuzzy clustering. One of them which is based on partitions is called K-means clustering method. There also exists another method called grid-clustering[3]. In this paper we describe main properties of grid-clustering methods and illustrate them by experiments.

2. Grid-clustering method

The conventional cluster algorithms calculate a distance based on a dissimilarity metric (Euclidean etc.) between cluster centers. The patterns are clustered accordingly to the resulting dissimilarity index. The grid-clustering algorithm [2] differs from the conventional cluster algorithms in that it organizes not the patterns but the value space, which surrounds the patterns. To organize the value space, a variation of the multidimensional data structure of the grid file is used, which is called Grid Structure.

The patterns are treated as points in d-dimensional value space and are randomly inserted into the grid structure. The points are stored according to their pattern values. The grid structure partitions the value space and administrates the points by a set of surrounding rectangular shaped blocks.

Let X=(x1, x2, ..., x_n) be a set of n patterns. x_i is the i-th pattern consisting of a tuple of describing features (a_{i1}, a_{i2}, ..., a_{id}), where d is the number of dimension. A block is a d-dimension rectangular shaped cube containing up to a maximum of bs patterns (bs-block size). The following properties are satisfied:

For all \( x_i \), \( x_i \in B_j \), \( B_i \cap B_k = 0 \), if \( j \neq k \)

\( B_j \neq 0 \cup B_j = X \)

In this case the patterns are disjointly partitioned among the blocks.

Grid-clustering method clusters blocks \( B_j \) (and so the patterns X) into a nested sequence of disjoint clusters, where \( (C_{u1}, C_{u2}, ..., C_{wu}) \) is the u-th clustering. The initial (0-th) clustering is that each block is a cluster, i.e. \( C_{0j}=B_j \), \( j=1,...,b \) and \( w=b \).

The blocks can be considered as a pre-clustering phase or an initialization of cluster centers. The cardinality of these centers is dependent on the block size and is defined by \( 1<p_B<bs \) where \( p_B \) is the number of patterns contained in block B. The grid-clustering algorithm uses this block information via the index structure of the grid file and clusters the patterns according to their surrounding blocks. For
example [3], Fig. 1 shows the value space partition with 3 clusters:

3. Grid-clustering algorithm

The algorithm calculates the density of each block using the numbers of patterns and the spatial volume of the block. Spatial volume $V_B$ of a block $B$ is the Cartesian product of the extents $e_i$ of block $B$ in each dimension:

$$V_B = \prod_{i=1}^{d} e_{B_i}$$

Density $D_B$ of block $B$ is the ratio of the number of patterns $p_B$ contained in block $B$ and the spatial volume $V_B$ of $B$:

$$D_B = \frac{p_B}{V_B}.$$ 

The blocks are sorted according to their density. The result is a sequence $B_1, B_2, \ldots, B_i$, $i'$ denotes a permutation of the index $i$ reflecting the sorted order. The blocks with the highest density (with strongest pattern correlation) build the clustering centers. The remaining blocks are clustered in iteration in the sequence to their density, building new cluster centers or merging with existing clusters. Only blocks, which adjoin a cluster, can be merged. Adjacent blocks are called neighbor blocks.

A neighbor search is done starting at the cluster center, inspecting adjacent blocks, finding a neighbor and recursively proceeding with this block. This search is similar to the traversal of a graph finding tree. The blocks represent the nodes and an edge between two nodes exists if the respective blocks adjoin.

In general [2], the execution of the grid-clustering algorithm includes 5 steps:

- Sorting of the blocks;
- Identifying cluster centers;
- Traversal of neighbor blocks.

4. Experiments and conclusions

Currently the author is working on the series of experiments that give evidence for the advantages of grid-clustering and compare it with conventional clusterization schemes.

The plan of experiments included the comparison of fuzzy C-means clustering algorithm and the grid-clustering algorithm aimed to determine the most optimal algorithm and use it later in the RBF networks.

For example, we have the feature space with two clusters (see Fig.2):