{ester | kriegel | sander | xwxu}@informatik.uni-muenchen.de

Abstract

Clustering algorithms are attractive for the task of class identification in spatial databases. However, the application to large spatial databases rises the following requirements for clustering algorithms: minimal requirements of domain knowledge to determine the input parameters, discovery of clusters with arbitrary shape and good efficiency on large databases. The well-known clustering algorithms offer no solution to the combination of these requirements. In this paper, we present the new clustering algorithm DBSCAN relying on a density-based notion of clusters which is designed to discover clusters of arbitrary shape. DBSCAN requires only one input parameter and supports the user in determining an appropriate value for it. We performed an experimental evaluation of the effectiveness and efficiency of DBSCAN using synthetic data and real data of the SEQUOIA 2000 benchmark. The results of our experiments demonstrate that (1) DBSCAN is significantly more effective in discovering clusters of arbitrary shape than the well-known algorithm CLARANS, and that (2) DBSCAN outperforms CLARANS by a factor of more than 100 in terms of efficiency.

Keywords: Clustering Algorithms, Arbitrary Shape of Clusters, Efficiency on Large Spatial Databases, Handling Nlj4-275oise.

1. Introduction

Numerous applications require the management of *spatial* data, i.e. data related to space. *Spatial Database Systems (SDBS)* (Gueting 1994) are database systems for the management of spatial data. Increasingly large amounts of data are obtained from satellite images, X-ray crystallography or other automatic equipment. Therefore, automated knowledge discovery becomes more and more important in spatial databases.

Several tasks of *knowledge discovery in databases* (KDD) have been defined in the literature (Matheus, Chan & Piatetsky-Shapiro 1993). The task considered in this paper is *class identification*, i.e. the grouping of the objects of a database into meaningful subclasses. In an earth observation database, e.g., we might want to discover classes of houses along some river.

Clustering algorithms are attractive for the task of class identification. However, the application to large spatial databases rises the following requirements for clustering algorithms:

(1) Minimal requirements of domain knowledge to determine the input parameters, because appropriate values are often not known in advance when dealing with large databases.

(2) Discovery of clusters with arbitrary shape, because the shape of clusters in spatial databases may be spherical, drawn-out, linear, elongated etc.

(3) Good efficiency on large databases, i.e. on databases of significantly more than just a few thousand objects.

The well-known clustering algorithms offer no solution to the combination of these requirements. In this paper, we present the new clustering algorithm DBSCAN. It requires only one input parameter and supports the user in determining an appropriate value for it. It discovers clusters of arbitrary shape. Finally, DBSCAN is efficient even for large spatial databases. The rest of the paper is organized as follows. We discuss clustering algorithms in section 2 evaluating them according to the above requirements. In section 3, we present our notion of clusters which is based on the concept of density in the database. Section 4 introduces the algorithm DBSCAN which discovers such clusters in a spatial database. In section 5, we performed an experimental evaluation of the effectiveness and efficiency of DBSCAN using synthetic data and data of the SEQUOIA 2000 benchmark. Section 6 concludes with a summary and some directions for future research.

2. Clustering Algorithms

There are two basic types of clustering algorithms (Kaufman & Rousseeuw 1990): partitioning and hierarchical algorithms. *Partitioning algorithms* construct a partition of a database D of n objects into a set of k clusters. k is an input parameter for these algorithms, i.e some domain knowledge is required which unfortunately is not available for many applications. The partitioning algorithm typically starts with an initial partition of D and then uses an iterative control strategy to optimize an objective function. Each cluster is represented by the gravity center of the cluster *(k-means algorithms)* or by one of the objects of the cluster located near its center *(k-medoid algorithms)*. Consequently, partitioning algorithms use a two-step procedure. First, determine k representatives minimizing the objective function. Second, assign each object to the cluster with its representative "closest" to the considered object. The second step implies that a partition is equivalent to a voronoi diagram and each cluster is contained in one of the voronoi cells. Thus, the shape of all