Data Mining and Knowledge Discovery, 3, 263–290 (1999)

A Fast Parallel Clustering Algorithm for Large Spatial Databases

XIAOWEI XU* Xiaowei.Xu@mchp.siemens.de
Corporate Technology, Siemens AG, Otto-Hahn-Ring 6, D-81730 München, Germany

JOCHEN JÄGER jaeger@informatik.uni-muenchen.de
HANS-PETER KRIEGEL kriegel@informatik.uni-muenchen.de
Institute for Computer Science, University of Munich, Oettingenstr. 67, D-80538 München, Germany

Editors: Yike Guo and Robert Grossman

Abstract. The clustering algorithm DBSCAN relies on a density-based notion of clusters and is designed to discover clusters of arbitrary shape as well as to distinguish noise. In this paper, we present PDBSCAN, a parallel version of this algorithm. We use the 'shared-nothing' architecture with multiple computers interconnected through a network. A fundamental component of a shared-nothing system is its distributed data structure. We introduce the dR*-tree, a distributed spatial index structure in which the data is spread among multiple computers and the indexes of the data are replicated on every computer. We implemented our method using a number of workstations connected via Ethernet (10 Mbit). A performance evaluation shows that PDBSCAN offers nearly linear speedup and has excellent scaleup and sizeup behavior.

Keywords: clustering algorithms, parallel algorithms, distributed algorithms, scalable data mining, distributed index structures, spatial databases

1. Introduction

Spatial Database Systems (SDBS) (Gueting, 1994) are database systems for the management of spatial data, i.e. point objects or spatially extended objects in a 2D or 3D space or in some high-dimensional feature space. Knowledge discovery becomes more and more important in spatial databases since increasingly large amounts of data obtained from satellite images, X-ray crystal-lography or other automatic equipment are stored in spatial databases.

Data mining is a step in the KDD process consisting of the application of data analysis and discovery algorithms that, under acceptable computational efficiency limitations, produce a particular enumeration of patterns over the data (Fayyad et al., 1996). Clustering, i.e. grouping the objects of a database into meaningful subclasses, is one of the major data mining methods (Matheus et al., 1993). There has been a lot of research on clustering algorithms for decades but the application to large spatial databases introduces the following new conditions:

*This work was performed while the author was still working at the Institute for Computer Science, University of Munich.

(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 non-convex, spherical, drawn-out, linear, elongated, etc.
(3) Good efficiency on very large databases, i.e. on databases of significantly more than just a few thousand objects.

Ester et al. (1996) present the density-based clustering algorithm DBSCAN. For each point of a cluster, its *Eps*-neighborhood (for some given $Eps > 0$) has to contain at least a minimum number of points ($MinPts > 0$). DBSCAN meets the above requirements in the following sense: first, DBSCAN requires only two input parameters (*Eps, MinPts*) and supports the user in determining an appropriate value for it. Second, it discovers clusters of arbitrary shape and can distinguish noise. Third, using spatial access methods, DBSCAN is efficient even for very large spatial databases. In addition, a generalized version of DBSCAN can cluster point objects as well as spatially extended objects (Sander et al., 1998).

In this paper, we present a parallel clustering algorithm PDBSCAN which is based on DBSCAN for knowledge discovery in very large spatial databases. We use the 'shared-nothing' architecture which has the main advantage that it can be scaled up to hundreds and probably thousands of computers. As a data structure, we introduce the dR*-tree, a distributed spatial index structure. The main program of PDBSCAN, the master, starts a clustering slave on each available computer in the network and distributes the whole data set onto the slaves. Every slave clusters only its local data. The replicated index provides an efficient access of data, and the interference between computers is also minimized through the local access of the data. The slave-to-slave and master-to-slaves communication is implemented by message passing. The master manages the task of dynamic load balancing and merges the results produced by the slaves.

We implemented our method on a number of workstations connected via Ethernet (10 Mbit). A performance evaluation shows that PDBSCAN scales up very well and has excellent speedup and sizeup behavior. The results from this study, besides being of interest in themselves, provide a guidance for the design of parallel algorithms for other spatial data mining tasks, e.g. classification and trend detection.

This paper is organized as follows. Section 2 surveys previous efforts to parallelize other clustering algorithms. Section 3 briefly describes the algorithm DBSCAN and Section 4 presents our parallel clustering algorithm PDBSCAN. Section 5 shows experimental results and evaluates our parallel algorithm with respect to speedup, scalability, and sizeup. Section 6 lists the conclusions and highlights directions for future work.

2. Related work on parallel clustering algorithms

Several authors have previously proposed some parallel clustering algorithms. Rasmussen and Willett (1989) discuss parallel implementations of the single link clustering method on an SIMD array processor. Their parallel implementation of the SLINK algorithm does not decrease the $O(n^2)$ time required by the serial implementation, but a significant constant speedup factor is obtained. Li and Fang (1989) describe parallel partitioning clustering (the