An Incremental Hierarchical Clustering

Arnaud Ribert, Abdel Ennaji, Yves Lecourtier

P.S.I. Faculté des Sciences, Université de Rouen
76 821 Mont Saint Aignan Cédex, France
Arnaud.Ribert@univ-rouen.fr

Abstract

This article describes a new algorithm to treat time incremental data by a hierarchical clustering. Although hierarchical clustering techniques enable one to automatically determine the number of clusters in a data set, they are rarely used in industrial applications, because a large amount of memory is required when treating more than 10,000 elements. To solve this problem, the proposed method proceeds by updating the hierarchical representation of the data instead of re-computing the whole tree when new patterns have to be taken into account. Memory gains, evaluated for a real problem (handwritten digit recognition) allow to treat databases containing 7 times more data than the classical algorithm.

1. Introduction

Numerous techniques of pattern recognition and image segmentation require an efficient clustering method to understand, interpret and simplify large amounts of multi-dimensional data [7][10][9][1]. Most of the time, k-means algorithm and other partitional methods are used in industrial applications [5][11][7][16]. However, they present the major drawback of requiring close-to-the-final-solution initial conditions, specially concerning the number of clusters. This a priori knowledge is rarely at the user's disposal because, if they use a clustering technique, this is precisely because they ignore the structure of their data. So, this constraint is intractable in the general case.

In fact, several authors recommend to use a hierarchical clustering before starting the k-means algorithm, since it does not require any a priori knowledge and provides a good representation of the data. But hierarchical clustering generally requires a great amount of memory, since it increases with the square of the number of elements in the database. So, treating 10,000 elements requires 200 Mo of RAM, while 800 Mo are needed to deal with 20,000 elements. Such a memory cost may explain why hierarchical clustering is rarely used in industrial applications.

On the other hand, clustering techniques consider that the given database is completely representative of the problem, whereas in fact, this is rarely the case when dealing with complex problems. In other words, from a practical point of view, a complex problem is often an incremental one. Consequently, it would be very useful to be able to enrich a database to take into account patterns which were not available at the time of the constitution of the database. The problem is then to update one's clustering.

Another advantage of incremental approaches is to consider the original database as a small growing one. In this scope, it might be possible to reduce the memory requirements of the hierarchical clustering by building a first numerical taxonomy (i.e. the tree resulting of a hierarchical clustering) using the available memory and updating it when new elements are taken into account. The success of this strategy depends on the capability to update a numerical taxonomy while keeping the largest possible part of it. This is the objective of the algorithm that is described in this paper.

In the next section, the main characteristics of the hierarchical clustering are recalled. Sections 3 and 4 describe respectively the way to determine the anchoring point of a new element in a numerical taxonomy and how to update this one, starting from the anchoring point. Eventually, section 5 presents experimental results over real databases.

2. A brief recall on the hierarchical clustering

A hierarchical clustering method is a procedure to represent data as a nested sequence of partitions. An