Discrete Applied Mathematics 169 (2014) 43–67

Contents lists available at ScienceDirect

Discrete Applied Mathematics

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

Balanced compact clustering for efficient range queries in metric spaces

Alberto Ceselli [a], Fabio Colombo [b], Roberto Cordone [b,*]

[a] Dipartimento di Informatica, Università degli Studi di Milano, Via Bramante 65, 26013 - Crema, Italy
[b] Dipartimento di Informatica, Università degli Studi di Milano, Via Comelico 39, 20135 - Milano, Italy

ARTICLE INFO

Article history:
Received 17 July 2013
Received in revised form 2 December 2013
Accepted 23 December 2013
Available online 10 January 2014

Keywords:
Similarity search
Clustering
Information retrieval
Integer programming
Tabu search

ABSTRACT

Given a set of points in a metric space, an additional *query point* and a positive threshold, a *range query* determines the subset of points whose distance from the query point does not exceed the given threshold. This paper tackles the problem of clustering the set of points so as to minimize the number of distance evaluations required by a range query. This problem models the efficient extraction of information from a database when the user is not interested in an exact match retrieval, but in the search for similar items. Since this need has become widespread in the management of text, image, audio and video databases, several data structures have been proposed to support such queries. Their optimization, however, is still left to extremely simple heuristic rules, if not to random choices. We propose the *Balanced Compact Clustering Problem* (*BCCP*) as a combinatorial model of this problem. We discuss its approximation properties and the complexity of special cases. Then, we present two Integer Programming formulations, prove their equivalence and introduce valid inequalities and variable fixing procedures. We discuss the application of a general-purpose solver on the more efficient formulation. Finally, we describe a Tabu Search algorithm and discuss its application to randomly generated and to real-world benchmark instances up to one hundred thousands points.

1. Introduction

The *Balanced Compact Clustering Problem* (*BCCP*) concerns a metric space (D, d), where the *domain* D is a set of *points* and $d : D \times D \to \mathbb{R}^+$ is a *pseudometric*, enjoying the following properties: (1) $d_{ij} \geq 0$ for all $i, j \in D$; (2) $d_{ii} = 0$ for all $i \in D$; (3) $d_{ij} = d_{ji}$ for all $i, j \in D$ (*symmetry*); (4) $d_{ij} \leq d_{ik} + d_{kj}$ for all $i, j, k \in D$ (*triangle inequality*). By assuming d to be a pseudometric, we admit the possibility that $d_{ij} = 0$ even if $i \neq j$. In particular, if $d_{ij} = 0 \Rightarrow i = j$, d is a *metric*. We denote a finite subset N of the domain (with $|N| = n$) as the *database* and a positive real value $\delta > 0$ as the *query radius*.

The *BCCP* looks for a collection $\mathcal{S} \subset 2^N$ of compact and δ-separable subsets of N whose union should cover N ($\bigcup_{S \in \mathcal{S}} S = N$). Since the concepts of compactness and separability and the motivation of the objective function are not trivial, for the sake of clarity we here informally introduce the context of the problem, while the complete formal definition is provided in Section 2. Informally speaking, a subset is compact when it includes all points closer than a given threshold to a given representative point, and it is separable if the other points are sufficiently far away. The objective of the *BCCP* is to minimize the number of subsets, $s = |\mathcal{S}|$, plus the cardinality of the largest one, $\eta = \max_{S \in \mathcal{S}} |S|$.

The *BCCP* is an abstract model for a database subject to a *range query*, which is the search for the points in the database which are "similar", in terms of the pseudometric d, to a *query point* k newly extracted from the domain. A range query

* Corresponding author. Tel.: +39 0250316235; fax: +39 0250316373.
E-mail addresses: alberto.ceselli@unimi.it (A. Ceselli), fabio.colombo2@unimi.it (F. Colombo), roberto.cordone@unimi.it (R. Cordone).

0166-218X/$ – see front matter
http://dx.doi.org/10.1016/j.dam.2013.12.019

returns the set $Q_{k\delta}$ of all points closer to k than the query radius δ.

$$Q_{k\delta} = \{i \in N : d_{ki} \leq \delta\} \quad k \in D,\ \delta > 0. \tag{1}$$

Similarity retrieval is used as an internal component in a wide spectrum of applications, such as searching on huge collections of text documents, DNA or protein sequences, images, audio and video streams, but also performing nearest neighbour classification, compressing video streams (to reuse image patches) or solving multiobjective optimization problems (to avoid near-duplication of solutions) [10].

A straightforward implementation of the range query would compare d_{ki} to δ for each database point $i \in N$. However, several data structures have been proposed in the literature to speed up the query. Some examples are the VP-tree [14], the M-tree [4], the List of Clusters [3] and the GNA-tree [1]; a taxonomy and a more detailed discussion on their features can be found in [10]. These structures exploit the properties of metric spaces in order to rule out most of the database points from an explicit consideration. The time required to build these structures is amortized over a long series of queries, and usually ignored when considering the query cost. Therefore, the main goal which is pursued in their design is to optimize the efficiency of a single range query.

Most of these data structures induce a partition or covering of the database points into regions associated to representative points and exploit the distance of the query point with respect to a representative in order to avoid considering the other points of the region. In other words, all points within a region are handled collectively. The computational effort required by a range query is therefore the sum of

1. an *internal complexity*, including the computation of the distance between the query point and the representative points;
2. an *external complexity*, including the computation of the distance between the query point and the database points which could not be ruled out.

As discussed in the following, the total computational effort is the sum of the two terms, where the former is related to the number of regions, and the latter to their cardinality. Since the former term increases and the latter decreases with the number of regions, there is an intermediate optimal structure that depends on the distribution of the points and on the query radius δ. Presently, such an optimization is performed with extremely simple heuristic rules and even with random choices (see [2] for a comparison between random and heuristically chosen representatives, and a discussion on criteria to achieve good choices). Furthermore, some of these structures repeat recursively the process when large amounts of data are involved. In this case, each subset generated is handled as a full database and subdivided into smaller subsets. The process terminates when the cardinality of the subsets obtained is small enough to allow in reasonable time the exhaustive computation of their distances from the query point. Such a recursive process yields a tree or forest structure and the execution of a range query requires in general to compute the distance between the query point and representative points selected at different levels of the structure, as described in the next section. Though at first we will focus on the basic decomposition, we will later extend the discussion to a multi-level decomposition.

To the best of our knowledge, this paper is the first attempt to provide a formal framework and nontrivial optimization algorithms for the problem of minimizing the query time on these data structures. Rather than focusing on a specific data structure, we model the basic core shared by most of them, in particular the use of subsets defined with respect to the pseudometric d and the search for an effective balance between the internal and the external complexity. As discussed in Section 2.1, a number of additional requirements can be imposed by restricting the collection of available subsets. As well, tuning the two terms of the objective function allows to relax the separability constraint or to minimize the average query time rather than the maximum one.

Section 2 introduces the formal framework of the *BCCP*. Section 3 proves some theoretical properties, namely its $\mathcal{NP}$-completeness, its approximability and the existence of polynomial algorithms for special cases. Section 4 presents two Integer Programming (*IP*) formulations, proves their equivalence, and introduces valid inequalities and variable fixing procedures to strengthen them. Since the latter take advantage from the availability of good heuristic solutions, Section 5 describes a Tabu Search algorithm to obtain them. In the end, Section 6 discusses the computational results both for the heuristic and for a commercial *IP* solver, focusing on the benefits gained from the valid inequalities and the fixing procedures. The results concern random instances of the basic problem with up to $n = 250$ points. Applying the multi-level decomposition approach, the discussion is extended to real-world benchmark instances up to $n \approx 100\,000$ points, drawn from the *metric spaces library* [6].

2. The balanced compact clustering problem

In this section we provide some theoretical background drawn from the literature, in order to explain the principles of compactness and separability, which allow to perform range queries more efficiently than by a straightforward sequence of distance comparisons. Then, we introduce a formal definition of the problem to trade the construction time of an auxiliary data structure for an optimized worst-case query time. Section 2.1 briefly describes how to extend the model when considering the average query time, or relaxing the separability property or introducing some additional requirements. Section 2.2 generalizes the model to multi-level data structures.

Definition 1. Given a database N, the *compact subset* $S_i^j = \left\{l \in N : d_{il} \leq d_{ij}\right\}$ is the set including all points of N not farther than j from i; point i is denoted as the *centre* of the subset.