Graphical Models 76 (2014) 609–619

Contents lists available at ScienceDirect

Graphical Models

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

Improved pruning of large data sets for the minimum enclosing ball problem ☆

Linus Källberg *, Thomas Larsson

School of Innovation, Design and Engineering, Mälardalen University, Sweden

ARTICLE INFO

Article history:
Received 30 October 2013
Received in revised form 18 June 2014
Accepted 22 June 2014
Available online 3 July 2014

Keywords:
Minimum enclosing balls
Bounding spheres
Acceleration techniques
Pruning
Culling

ABSTRACT

Minimum enclosing ball algorithms are studied extensively as a tool in approximation and classification of multidimensional data. We present pruning techniques that can accelerate several existing algorithms by continuously removing interior points from the input. By recognizing a key property shared by these algorithms, we derive tighter bounds than have previously been presented, resulting in twice the effect on performance. Furthermore, only minor modifications are required to incorporate the pruning procedure. The presented bounds are independent of the dimension, and empirical evidence shows that the pruning procedure remains effective in dimensions up to at least 200. In some cases, performance improvements of two orders of magnitude are observed for large data sets.

1. Introduction

Given a center point $c \in \mathbb{R}^d$ and a radius $r \in \mathbb{R}$, let $B(c, r)$ denote the ball $\{x \in \mathbb{R}^d : \|x - c\| \leqslant r\}$, that is, the subset of $\mathbb{R}^d$ within Euclidean distance r from c. Then some given finite set of points $P = \{p_1, \ldots, p_n\} \subset \mathbb{R}^d$ is enclosed by $B(c, r)$ if $\|p_j - c\| \leqslant r$ holds for $j = 1, \ldots, n$. The minimum enclosing ball (MEB) problem is to find the unique ball with minimum radius that encloses P. Henceforth, we denote the center point and radius of the MEB by c^* and r^*, respectively, and we write $B(c^*, r^*)$ compactly as B^*. The MEB problem, which is also known as the 1-center problem or the minimax location problem, has been studied for more than a century, and still receives much attention today due to its relevance in important application areas such as rendering, animation, collision detection, robotics, and machine learning.

It is well-known that B^* is determined by at most $d + 1$ boundary points from P. Thus, the fundamental task of an exact MEB solver is to locate these *support points* [15,8,7]. In many types of input, however, a majority of the points are strictly inside B^*. Being able to identify and eliminate, or *prune*, such points early on during MEB computations is therefore likely to speed up the subsequent processing.

In situations where the exact solution is not required, a $(1 + \epsilon)$-approximation of B^*, i.e., an enclosing ball $B(c, r)$ such that $r \leqslant (1 + \epsilon)r^*$, can be computed efficiently by collecting a small *core-set* of representative input points [4,3]. This subset has the property that its MEB enlarged by a factor of at most $(1 + \epsilon)$ encloses also P. A number of algorithms have been presented that compute a $(1 + \epsilon)$-approximate solution by finding a core-set of size $O(1/\epsilon)$, which is independent of both n and d [2,10,14,3,16,12]. Eliminating interior points has the potential to accelerate also these algorithms.

Clearly, the support points belong to the convex hull of the input set, and it would be possible to initially eliminate all points not on the hull. However, the time complexity of computing the convex hull would exceed that of most MEB algorithms used in practice. Furthermore, when all input

☆ This paper has been recommended for acceptance by Frederic Chazal and Peter Lindstrom.

* Corresponding author.
E-mail addresses: linus.kallberg@mdh.se (L. Källberg), thomas.larsson@mdh.se (T. Larsson).

http://dx.doi.org/10.1016/j.gmod.2014.06.003
1524-0703/

points are on the hull, nothing would be pruned. On the other hand, the algorithm by Megiddo [13] uses a sophisticated prune-and-search method that always manages to reduce the input points by a constant factor in each algorithm pass. Thus, termination is guaranteed in linear time in fixed dimension. A realization of Megiddo's approach, however, would be intractable due to its exponential dependency on the dimension.

A simpler and more practical pruning approach is described by Ahipaşaoğlu and Yıldırım [1]. By interleaving pruning passes with the main iterations of the two algorithms by Yıldırım [16], they obtain significant performance improvements in dimensions $d \leqslant 100$. Both of these algorithms run in $O(dn/\epsilon)$ time, and arrive at a $(1+\epsilon)$-approximation of B^* through a sequence of intermediate approximations with monotonically increasing radii that converge to r^* from below. In each iteration, a scan of the entire input set is performed to find the point farthest from the current center point. These repeated farthest-point queries, each taking $O(dn)$ time, dominate the execution time. Thus, reducing the size of the input gives immediate speed benefits in subsequent iterations.

Given such an intermediate approximation $B(c,r)$, where $r \leqslant r^*$, they derive the following upper bound $\bar{\Delta}$ on the distance $\|c-c^*\|$:

$$\bar{\Delta} = \sqrt{R^2 - r^2}, \tag{1}$$

where $R = \max_{p_j \in P}\|p_j - c\|$. Using this bound, they also derive the following conservative condition to determine if a point p is enclosed in the interior of B^*:

$$\|p - c\| < r - \bar{\Delta}. \tag{2}$$

Thus, any point satisfying this condition can safely be removed from the input.

In each pruning pass, Eq. (1) is first applied to the most recent intermediate ball, and then the pruning condition in Eq. (2) is evaluated for each point in P. In order to avoid the overhead of invoking the pruning procedure in situations where little or nothing is pruned, the procedure is skipped every time the right-hand side of Eq. (2) evaluates to a value smaller than or equal to $0.55r$, where the factor 0.55 was determined empirically.

Nielsen and Nock [14] propose a distance-filtering technique to speed up repeated searches for the farthest point. Using the Cauchy–Schwarz inequality, they are able to compute an upper bound on each squared distance $\|p_j - c\|^2$ in constant time during the search. In this way, the $O(d)$ cost of computing the exact squared distance can be avoided whenever the upper bound does not exceed the largest squared distance encountered so far. An advantage of this approach is that no additional parameters, such as r and R above, are required, which might make it more generally applicable. A disadvantage, however, is that the effectiveness of this method is sensitive to how the input points are distributed in relation to the origin. Alternative distance filtering strategies that do not suffer from this problem are proposed in a recent publication [9]. A drawback shared by all these filtering approaches, however, is that every input point must be at least touched in every iteration, which is in stark contrast to the pruning procedures discussed herein, which continuously eliminate points entirely from further processing.

In this article, we improve the approach by Ahipaşaoğlu and Yıldırım so that more points are eliminated in each pass. We develop our methods from the concept of *viability* of the intermediate balls used to derive the pruning bounds. Specifically, our main contributions are (i) the key insight that viability is in fact satisfied in many existing algorithms for the MEB problem, which makes our pruning methods widely applicable, (ii) a tighter bound $\bar{\Delta}$, which in itself immediately leads to more effective point reductions, (iii) an improved alternative to the pruning condition in Eq. (2) and (iv) a thorough empirical evaluation where the presented techniques are applied to several state-of-the-art algorithms. In addition to enabling improved pruning, we believe that our theoretical results bring additional understanding of the MEB problem and may be useful in a wider perspective.

2. Theoretical results

This section presents the theoretical results of the paper. In the first part, we discuss the general assumptions underlying our approach. Then in the second part, we present bounds that enable improved pruning.

2.1. Viable balls

We will require the following property on any approximate solution used to derive our bounds.

Definition 1. A ball $B(c,r)$ is *viable* if it satisfies

$$r^2 + \|c - c^*\|^2 \leqslant r^{*2}. \tag{3}$$

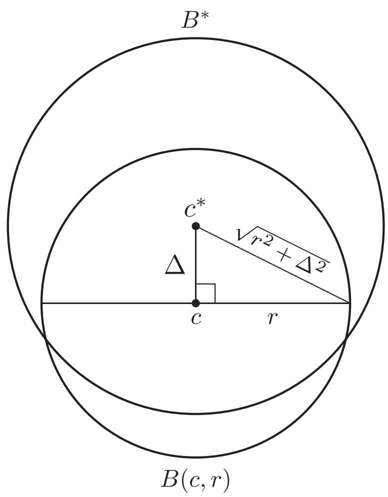

Fig. 1. Geometrical interpretation of viability. When the condition is satisfied by a ball $B(c,r)$, at least any diameter that is perpendicular to $c^* - c$ is enclosed in B^*, since the distance from c^* to the endpoints of such a diameter is $\sqrt{r^2 + \Delta^2} \leqslant r^*$, where $\Delta = \|c - c^*\|$.