The MINIBALL Algorithm

Sushrut Jog

Introduction. I studied this algorithm as a part of my 2024 Summer internship on Topological Data Analysis under Dr Kashyap Rajeevsarathy.

The main idea behind topological data analysis is getting the shape out of the data by using various techniques. Much like the game in which you connect the appropriate to reveal the picture, we wish to meaningfully connect points in the dataset to reveal the underlying shape. However, we need some rules to appropriately connect the points. For instance, points which are closer to each other should get connected before the points which are farther. Also, we should take into consideration all possible connections – meaning, by the very end of the process, all dots should be connected to each other.

In the first section, we will take a look at some types of filtrations, which are essentially the intermediate structures that we get as we start connecting the dots. In the second section, we will look at an efficient algorithm which computes if the given simplex σ belongs in the filtration or not, which tells us which points to connect in that particular filtration.

1. Filtrations and Data

Here, data is synonymous to a finite set of points in some well-defined metric space. The most common example for this is a finite set of points in $\mathbb{R}^n$ with the standard Euclidean metric. Even though it's not necessary that any finite dataset will have an embedding in $\mathbb{R}^n$, we will consider as our finite set to be $M \subset \mathbb{R}^n$ with the Euclidean metric.

We have a well-defined notion of *thickening* of a point by any scale ε where $\varepsilon > 0$ is a real number. We define $M^{+\varepsilon}$ as the union of all ε-balls in $\mathbb{R}^n$ around all points of M. In simple words, you consider elements of M to be *point clouds* rather than points and take their union. Because of the metric on $\mathbb{R}^n$, this makes sense for any $\varepsilon > 0$. This allows us to get the geometric shape of M across various scales.

Definition 1.1. A `filtration` of a simplicial complex K of length n is a nested sequence of strictly increasing subcomplexes

$$F_1K \subset F_2K \subset \cdots \subset F_nK = K.$$

For real values of $\varepsilon, \delta > 0$, the filtrations are defined in a continuous setting as a sequence of subcomplexes as for all $\varepsilon \leq \delta$

$$F_\varepsilon K \subseteq F_\delta K.$$

Definition 1.2. Let (M, d) be a finite metric space. The `Vietoris-Rips filtration` of M is the filtration $\{VR_\varepsilon(M) \mid \varepsilon > 0\}$ defined by the rule "$\{x_0, x_1, \ldots, x_k\} \subseteq M$ is a simplex in $VR_\varepsilon(M)$ *iff* $d(x_i, x_j) \leq \varepsilon$, for all i, j in $\{0, 1, \ldots, k\}$."

Definition 1.3. Let M be a finite subspace of a metric space (Z, d). The `Čech filtration` $\{C_\varepsilon(M) \mid \varepsilon > 0\}$ is given by the rule "$\{x_0, x_1, \ldots, x_k\} \subseteq M$ is a simplex in $C_\varepsilon(M)$ *iff* there exists $z \in Z$ such that $d(x_i, z) \leq \varepsilon$, for all $i \in \{0, 1, \ldots, k\}$."

2. The Smallest Enclosing Ball

Let $\sigma \subseteq S$ be a subset of the given points. Then, it's clear that deciding whether or not σ belongs to $C_r(S)$ is equivalent to deciding whether or not σ fits inside a ball of radius r.

Let the miniball of σ be the smallest closed ball that contains σ. Note that the miniball of a set has to be unique, or else there will be at least one point of σ which will be contained in one miniball and not in the other, which is contradictory. Now, the radius of the miniball is less than or equal to r *iff* $\sigma \in C_r(S)$, and thus, it suffices to find r to solve the problem.

Let $|\sigma| = d + 1$. The only important points of finding the miniball of a point set are the points which lie on the boundary of the set. Therefore, the miniball is determined by some subset $\beta \subseteq \sigma$ of $k + 1 \leq d + 1$ points, which all lie on its boundary. Once we know this subset of $k + 1$ points, we can verify that we have indeed found the miniball by testing that it contains all other points. When we have many more points than the dimensions, the chance that the randomly chosen point belongs in the set v is very small, and discarding it as an interior point is easy.

The MiniBall(τ, β) algorithm uses this exact technique. It partitions σ in two disjoint sets τ and β and returns the miniball that contains all points of τ in its interior and all points of β on its boundary. To get the miniball of the entire set σ, we call MiniBall$(\sigma, \emptyset)$.

The Algorithm

Algorithm 1: MiniBall(τ, β)

input : (τ, β) such that $\sigma = \tau \sqcup \beta$
output: The miniball B of σ with points of τ in its interior and points of β on the boundary

if $\tau = \emptyset$ **then**
  compute the miniball B of β directly
else
  choose a random point $u \in \tau$;
  $B =$ MiniBall$(\tau \setminus \{u\}, \beta)$;
  if $u \notin B$ **then**
    $B =$ MiniBall$(\tau \setminus \{u\}, \beta \cup \{u\})$
return B

It is clear that the algorithm is randomized and recursive. The base case of the recursion is when we have two points, both lying on the boundary of the ball. This also means that we are checking for the condition $|\beta| \geq 2$ during every iteration.

The algorithm works as follows. We first check if the interior set is empty or not. If it is empty, all points must lie on the boundary of the miniball, and we can get the miniball directly in constant time using coordinate geometry. When the interior is nonempty, however, the general strategy is as follows: we choose one random point $u \in \tau$ and construct the miniball B recursively with set β on the boundary and the set $\tau \setminus \{u\}$ as the interior. If $u \in B$, we have found the required miniball and we are done. But if $u \notin B$, then it means that u lies outside the miniball of $\sigma \setminus \{u\}$, and thus, u should be a part of the boundary of the miniball of σ. Therefore, when $u \notin B$, we find the miniball with $\beta \cup \{u\}$ as the boundary and $\tau \setminus \{u\}$ as the interior, and we are done.

Analysis. When all points are on the boundary, assuming that the dimension we're working in – d – is fixed, we can get the miniball in constant time. Thus, the runtime depends on how many times we are executing the test '$u \notin B$.' Let $t_i(n)$ be the number of such expected tests which call MiniBall with n points in τ and possibly $i = d + 1 - |\beta|$ voids on the boundary.

Clearly, $t_i(0) = 0$, as we are looking at 0 points in the interior and i possible voids on the boundary, and as stated earlier, we can do this in constant time. In an amortised sense, this constant amount of price that is needed to calculate the ball for at most $d+1$ points on the boundary is paid by the **if** test that initiated the call. Now let $n > 0$. There is one call to the function with $n-1$ points in the interior and i possible points on the boundary, one test '$u \notin B$' and one call with the parameters $n-1$ and $i-1$. The probability of execution of the second call actually happens is at most $^i/_n$. Thus, we get the bound

$$t_i(n) \leq t_i(n-1) + 1 + \frac{i}{n} \cdot t_{i-1}(n-1).$$

Now, putting $i = 0$, we get $t_0(n) \leq t_0(n-1) + 1$, which implies that $t_0(n) \leq n$. Also, $t_1(n) \leq t_1(n-1) + 2 \leq 2n$, and so on. In general, we obtain the bound

$$t_i(n) \leq (i-1)! \cdot n.$$

This bound is constant times n as i is always bounded above by the number of dimensions we are working in – $d+1$ – which is a constant.

Therefore, the expected running time of the algorithm is $\mathcal{O}(n)$, or equivalently, it takes constant time per point.

References

[1] Edelsbrunner H, Harer J. *Computational Topology: An Introduction.* American Mathematical Society, 2010.
[2] Emo Welzl. *Smallest Enclosing Disks (Balls and Ellipsoids).* Lecture Notes in Computer Science **555**, 1991.

---

21 Aug 2024