Computational Geometry Learning
Exam

MPRI

Duration 2h

You are allowed access to the official course notes, the slides, and any personal notes.

1 Separation of a point set

Let $p_1, \ldots, p_n$ be n points of $\mathbb{R}^d$. We want to compute two points p_i and p_j, $i, j \in \{1, \ldots, n\}$, such that

$$||p_i - p_j|| = \min_{1 \leq k < l \leq n} ||p_k - p_l||.$$

The segment joining the points p_i and p_j is called a *minimal segment* and $||p_i - p_j||$ is called the *separation* of the point set.

A first algorithm

1. Show that a minimal segment is an edge of the Delaunay triangulation of $p_1, \ldots, p_n$. Deduce an algorithm to calculate a minimal segment. Analyze its complexity.

In the following we propose a randomized incremental algorithm to compute a minimal segment.

The planar case : $d = 2$

The algorithm considers the points one by one and maintains, at each step, a minimal segment.

For convenience, the index of a point is the step at which it has been inserted. For $i = 2, \ldots, n$, let $S_i = \{p_1, \ldots, p_i\}$, i.e. the set of the i first points and let δ_i be the minimal distance between two points of S_i.

Let G_i denote the grid of size δ_i, i.e. each cell in the grid is a square with edges of length δ_i and the vertices of the grid are the points $(\lambda\delta_i, \mu\delta_i)$, where λ and μ are integers. Each point of S_i is stored in the cell of the grid it belongs to. For simplicity, we assume that any point of S_i belongs to exactly one cell of the grid. Let B be the cell of the grid that contains the new point p_{i+1}

2a. Show that each cell contains at most four points of S_i.

2b. Show that if the shortest distance between two points of S_{i+1} is realized between p_{i+1} and a point p_j $(j \leq i)$, then $p_j \in B$ or to one of the eight cells incident to B.

Deduce that, if one knows B, the shortest distance, noted δ_{i+1}, between p_{i+1} and a point of S_i can be computed in time $O(1)$.

If $\delta_{i+1} \geq \delta_i$, we set $\delta_{i+1} := \delta_i$, $G_{i+1} := G_i$, and we store p_{i+1} in G_{i+1}.

Otherwise, we build a new grid G_{i+1} of size δ_{i+1}, we locate and we store the points of S_{i+1} in G_{i+1}.

The algorithm then considers the next point.

3. Detail the algorithm. In particular, show that, given S_i and δ_i, one can represent G_i by a data structure of size $O(i)$ that allows 1. to insert a new point in time $O(\log i)$ and 2. to report the points that belong to a given cell in time $O(\log i)$. Deduce the cost of one step of the algorithm when $\delta_{i+1} = \delta_i$ and when $\delta_{i+1} \neq \delta_i$.

4. Show that if the points are inserted in random order, the probability that $\delta_{i+1} \neq \delta_i$ is $\leq \frac{2}{i+1}$ (consider the case where δ_i is realized by a unique pair of points and then the case where δ_i is realized for several pairs of points). Deduce that the expected cost of the algorithm is $O(n \log n)$.

Extension to higher dimensions

5. Show that the previous results can be extended in d-dimensional space. How does the constant in the $O()$ depend on d?

2 Voronoi

Let $P = \{p_1, \ldots, p_n\}$ be a set of points of $\mathbb{R}^d$ in general position (no 2 points are equal, no 3 points are aligned ($d >= 2$), no 4 points are coplanar ($d >= 3$) or cocircular ($d = 2$), etc). To each p_i, we associate its Voronoi cell $V(p_i) = \{x \in \mathbb{R}^d : \forall p_j \in P, ||x - p_i|| \leq ||x - p_j||\}$. Similarly, we associate to p_i its Far cell $F(p_i) = \{x \in \mathbb{R}^d : \forall p_j \in P, ||x - p_i|| \geq ||x - p_j||\}$. The Voronoi cells and their faces / intersections form a cell-complex called the Voronoi diagram of P. The Far cells and their faces / intersections form a cell-complex called the Far diagram of P.

Give a short proof (you can use existing theorems on polytopes) to justify your answer to each question.

1. When is $V(p_i)$ empty ?
2. When is $F(p_i)$ empty ?
3. When is $V(p_i)$ non-convex ?
4. When is $F(p_i)$ non-convex ?
5. When is $V(p_i)$ unbounded ?
6. When is $F(p_i)$ unbounded ?
7. What is the maximal combinatorial complexity of the Voronoi diagram ?
8. What is the maximal combinatorial complexity of the Far diagram ?

(bonus) Suggest an algorithm to build the Far diagram in $\mathbb{R}^2$ and analyze its complexity.

3 Homology and persistence

Note : when asked for a persistence diagram, you may instead draw a persistence barcode. In both cases, you need to write the coordinates of every relevant point on the picture.

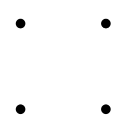

FIGURE 1 – corners of a square of side length 2

For the points in Figure 1 (in $\mathbb{R}^2$ with the Euclidean distance), and without justification :

— Draw the persistence diagram of the Čech filtration.

— Recall that the Rips complex $R^\rho(P)$ of parameter $\rho \geq 0$ on points P is the maximal complex, w.r.t inclusion, with vertex set P, and containing all edges connecting points at distance at most ρ. Draw the persistence diagram of the Rips filtration.

— Draw the persistence diagram of the sublevelset filtration of the distance function to the point set.

What are the Betti numbers (for homology with $\mathbb{Z}/2\mathbb{Z}$-coefficients) of the 1-dimensional complex in Figure 2 ? Give an explicit computation.

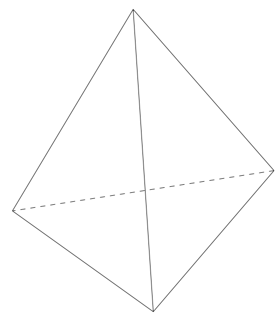

FIGURE 2 – 1-skeleton of a tetrahedron

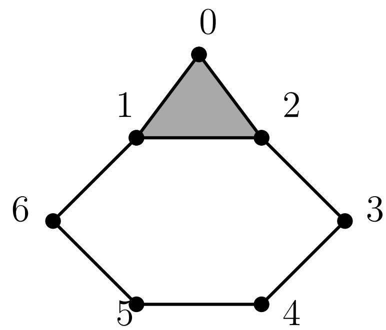

FIGURE 3

4 Homology and Morse theory

- Draw the Hasse diagram of the complex in Figure 3 and give a Morse matching with exactly two critical cells.
- Give the boundary maps for the corresponding Morse complex (with $\mathbb{Z}/2\mathbb{Z}$-coefficients).
- Give an expression (with generators) for the kernels and images of the boundary maps of the Morse complex, and deduce the Betti numbers (for homology with $\mathbb{Z}/2\mathbb{Z}$-coefficients) of the complex in Figure 3.