CS 164 & CS 266: Computational Geometry

Lecture 12

Nearest neighbors and Voronoi diagrams

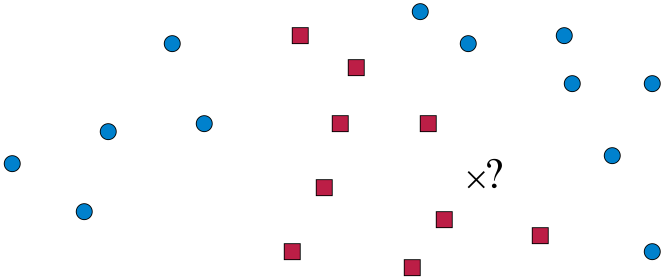
David Eppstein
University of California, Irvine

Fall Quarter, 2023

Supervised learning

Suppose you have seen a training set of sites (data points), classified in some way

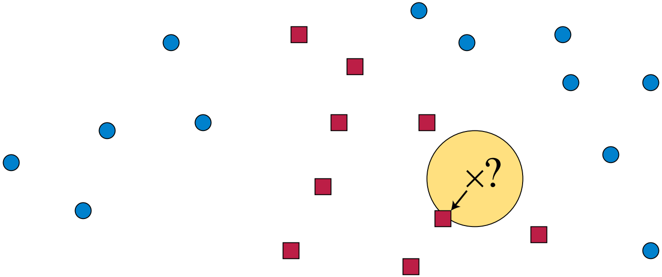
Now you get another point with unknown classification



What do you think its classification might be?

Nearest-neighbor classification

Find the nearest training site to the unknown point

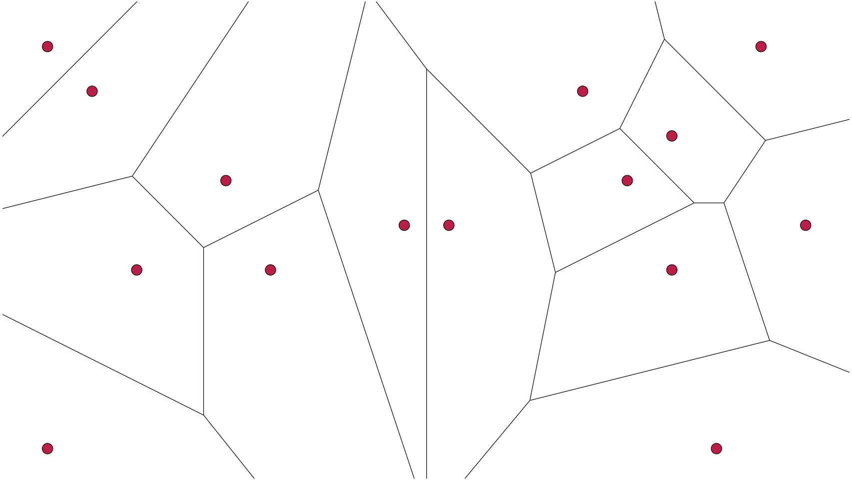


Guess that it has the same classification

Locus method for nearest neighbor queries

Construct regions of points all having the same nearest site

Then use point location (last time) on this subdivision



It is called a Voronoi diagram

Why the name?

Studied by Ukrainian mathematician Georgy Feodosevich Voronoy (1868–1908)

Also called Dirichlet tessellations after German mathematician Johann Peter Gustav Lejeune Dirichlet (1805–1959)

Their regions are called Voronoi cells or Thiessen polygons after American meteorologist Alfred H. Thiessen (1872–1956)

Georgy Voronoy

Why a meteorologist?

Suppose you want to know how much rain fell in a given area from a given storm

But you only have measurements at a finite set of rain gauges (measuring cups that fill with rainwater) giving the number of inches of rain at those locations

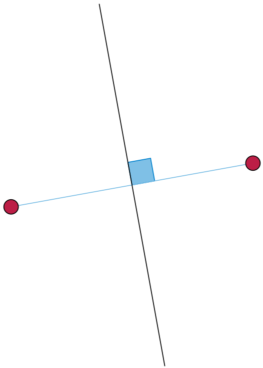
Nearest neighbor interpolation: Guess that the amount of rain at every other location equals the amount at the nearest gauge

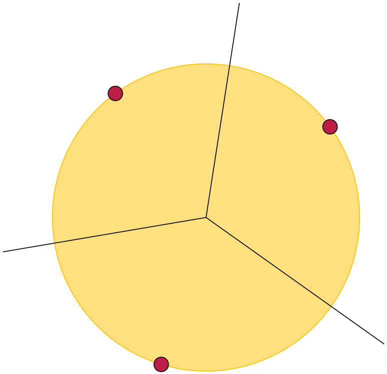
Estimated total rainfall = sum of areas of Voronoi cells multiplied by level of rain at each gauge

[Thiessen 1911]

Properties of Voronoi diagrams

Boundary between any two cells lies on a line, the perpendicular bisector of two sites





Vertices (where ≥ 3 cells meet) are equally far from their sites

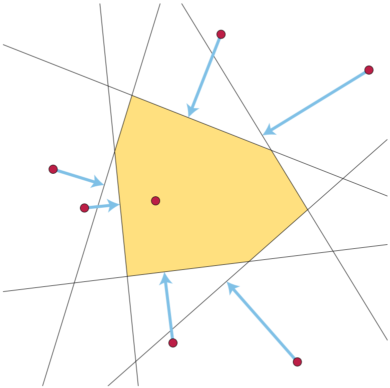
When points are in general position (no four on a circle),
exactly three cells meet at each vertex

More properties

Each site (= input point)
has a Voronoi cell

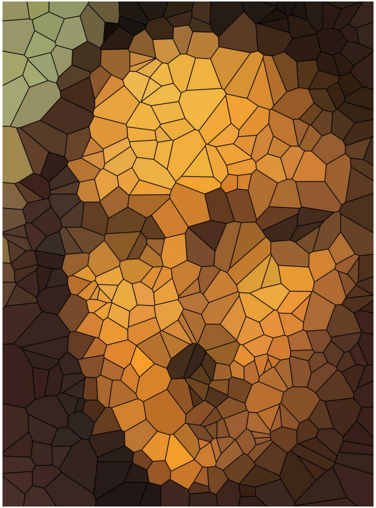
It is a convex polygon
(possibly infinite)
the intersection of halfplanes
bounded by perpendicular
bisectors to other sites

edges $\leq 2n - 3$
vertices $\leq 3n - 8$
(Euler formula)



Applications

- Nearest neighbor queries ("the post office problem") and nearest neighbor classification
- Nearest neighbor interpolation; weather estimation
- Image analysis; finding boundaries between image components
- Modeling shapes of cells in 2d biological surfaces (leaves, skin), crystal grains in polycrystalline materials (especially metals), and soap bubble foams
- Stained glass effect in graphics



Fortune's algorithm

Main ideas

Plane sweep

Sweep a vertical line left-to-right across sites

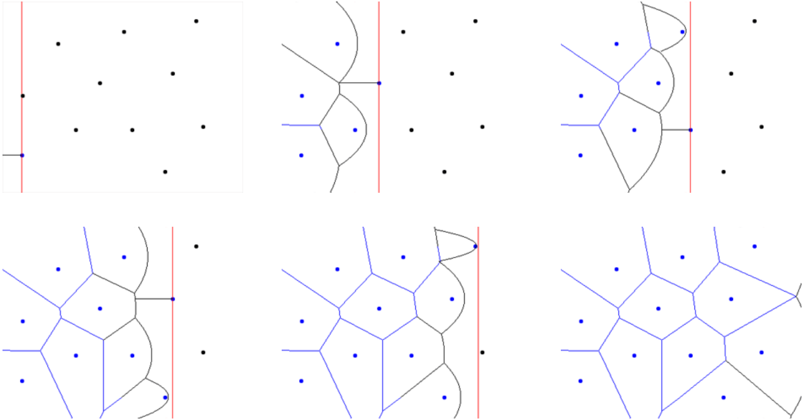
Sweep another curve “beach line” behind the sweep line

Behind beach line:
Construct Voronoi diagram of all sites crossed by sweep line

Between beach line and sweep line:
diagram may depend on points we haven’t yet swept

[Fortune 1987]

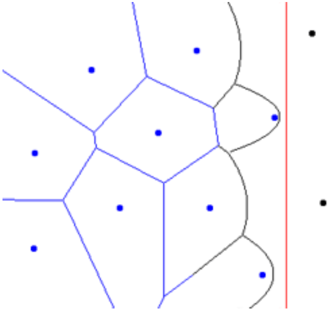
Some frames from a Wikipedia animation



Unfortunately, the animation does not include anything in between these frames

For another example with smoother animation, see
`https://www.youtube.com/watch?v=Y5X1TvN9TpM`

What is the beach line?



Left of the beach line: Points nearer to some swept-over site than to the sweep line

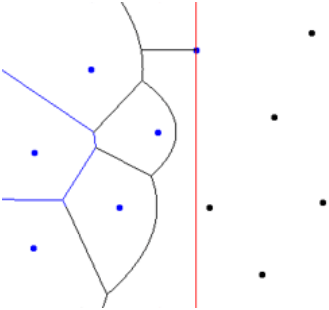
Right of the beach line: Points nearer to the sweep line than to any swept-over site

Each site is separated from the beach line by a parabola

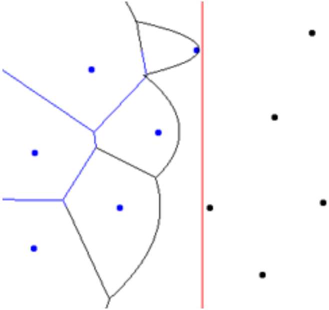
Left of beach line = union of insides of parabolas

Beach line and Voronoi edges

Pairs of beach line parabolas meet along Voronoi edges

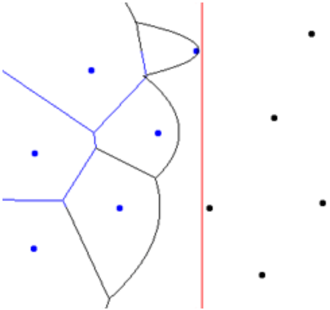


When a new parabola starts, it meets an existing parabola at same y-coordinate as new site



The edge between these two sites grows upward and downward from that point

How to represent the beach line

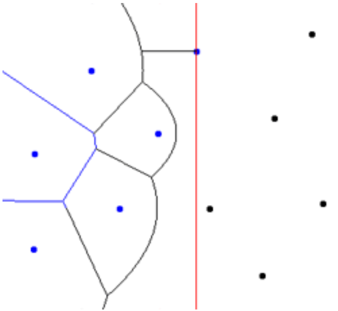


Store sequence of points where two parabolas meet, sorted vertically (by y-coordinate), in a binary search tree

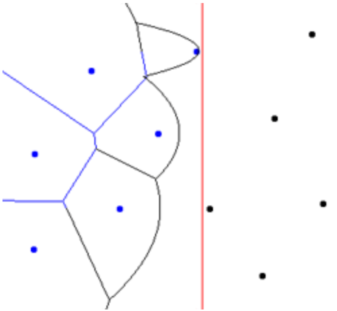
Store which Voronoi edge each point lives on and whether it is upper or lower point on its edge

Can compute its coordinates from x-coordinate of sweep line, as the point equidistant from 2 sites + sweep line

Two kinds of events



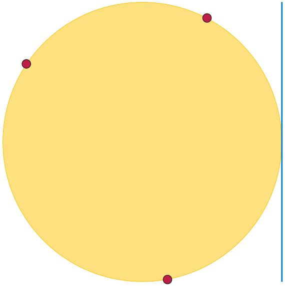
Sweep line crosses site, add new parabola to beach line



Beach line crosses Voronoi vertex, terminate Voronoi edges that meet there, start new edge

Priority for event points

Sites are crossed by sweep line in order by their x-coordinate



Voronoi vertex for three sites is crossed by beach line when the sweep line crosses the rightmost point of circle through three sites, x-coordinate = circle center + radius

The algorithm

Initialize an empty binary search tree for the beach line.

Initialize DCEL representation of empty Voronoi diagram.

Initialize priority queue of event points, prioritized by x-coordinate, initially all sites. Later, keep a Voronoi vertex event point in the queue for each consecutive pair of Voronoi edges touched by the beach line, and update these when we change the beach line.

While event queue is non-empty:

- Find and remove the event point with minimum x-coordinate
- If it's a site, binary search for its y-coordinate in the beach line search tree, and add a new thin parabola to the beach line at that coordinate
- Otherwise, it's a Voronoi vertex. Terminate the existing Voronoi edges that meet at that vertex, remove the beach line curves that were between those edges, and start a new Voronoi edge between the beach line curves that are now consecutive.

Analysis

Each time we process an event, it creates or terminates a Voronoi edge $\Rightarrow O(n)$ events

Each event involves $O(1)$ operations in the priority queue, DCEL, and search tree data structures

Data structure operations take time $O(\log n)$ (faster for DCEL)

Total time $O(n \log n)$

References and image credits

Tim Chartier. Stained glass Mona Lisa, July 9 2022. URL `https://twitter.com/timchartier/status/1545776675270672384`.

G. Lejeune Dirichlet. Über die Reduktion der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen. *Journal für die Reine und Angewandte Mathematik*, 1850 (40):209–227, 1850. doi: 10.1515/crll.1850.40.209.

Steven Fortune. A sweepline Algorithm for Voronoi diagrams. *Algorithmica*, 2:153–174, 1987. doi: 10.1007/BF01840357.

Friman. Standard Garden rain gauge. Public domain (CC0) image, May 17 2010. URL `https://commons.wikimedia.org/wiki/File:Regnmaaler1.JPG`.

Alfred H. Thiessen. Precipitation averages for large areas. *Monthly Weather Review*, 39(7): 1082–1089, July 1911. doi: 10.1175/1520-0493(1911)39⟨1082b:pafla⟩2.0.co;2.

Georges Voronoï. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Premier mémoire. Sur quelques propriétés des formes quadratiques positives parfaites. *Journal für die Reine und Angewandte Mathematik*, 1908(133):97–178, 1908a. doi: 10.1515/crll.1908.133.97.

Georges Voronoï. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième mémoire. Recherches sur les parallélloèdres primitifs. *Journal für die Reine und Angewandte Mathematik*, 1908(134):198–287, 1908b. doi: 10.1515/crll.1908.134.198.