Voronoi Diagram in Hyperbolic Plane
Algorithms & Application

MATH430 Final Project

Cheng-Yuan Lee

Chapter 1

Introduction

Contents

1.1 Purpose of the Project

As a competitive programmer, I have always been fascinated by computational geometry algorithms and their applications in solving a wide range of problems. Among these algorithms, the Voronoi diagram has particularly captured my interest due to its effectiveness in addressing complex challenges encountered in programming competitions. Additionally, the class have introduced me to various types of geometry, with hyperbolic geometry standing out as especially intriguing. This curiosity has led me to explore Voronoi diagrams within the hyperbolic plane. I am interested in understanding whether these diagrams can be computed as efficiently as their Euclidean counterparts and what unique problems they might help solve in hyperbolic spaces. This project aims to investigate these questions, delving into the properties and applications of Voronoi diagrams in hyperbolic geometry.

1.2 Overview of Voronoi Diagrams

Given a set S of n points in the two-dimensional plane, a **Voronoi diagram** partitions the plane into distinct regions based on proximity to the points in S. Specifically, each region associated with a point $p \in S$ consists of all points q in the plane that are closer to p than to any other point in S. In other words, a point q belongs to the region of its nearest neighbor in S.

The Voronoi diagram can be visualized as a planar graph formed by the boundaries between these regions. In this graph:

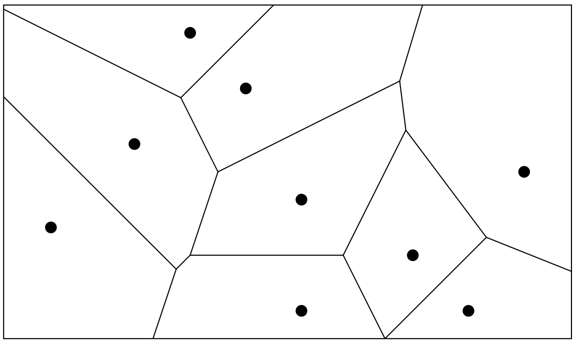

Figure 1.1: Example of a Voronoi Diagram in the Euclidean Plane

- **Vertices** occur where three or more boundary segments intersect or extend to infinity. These vertices represent points that are equidistant to three or more points in S and correspond to the circumcenters of the triangles formed by these points.
- **Edges** are the boundary segments that connect two vertices. Each edge consists of points that are equidistant to exactly two points in S and lies on the perpendicular bisector of the line segment joining these two points.

This structure not only provides a clear geometric representation of proximity relationships within the set S but also serves as a foundational tool in various applications, ranging from computer graphics and spatial analysis to problems in competitive programming.

1.3 Motivation for Hyperbolic Geometry

In recent years, there has been a significant shift in machine learning towards leveraging non-Euclidean geometries. Traditional Euclidean geometry often falls short when representing complex datasets that exhibit hierarchical or tree-like structures, which are inherently non-Euclidean. Hyperbolic geometry, with its unique properties, provides a natural framework for modeling such data more effectively.
A notable advancement in this area is the development of Hyperbolic Neural Networks by Ganea et al. in 2018. These networks demonstrated improved performance on tasks involving hierarchical data, highlighting the potential of hyperbolic spaces in machine learning applications. This trend underscores the growing importance of understanding non-Euclidean geometries and their algorithms.
Given this momentum, it is crucial to explore how fundamental algorithms, such as Voronoi diagrams, operate within hyperbolic spaces. Voronoi diagrams play a key role in various applications, including nearest neighbor search, clustering, and spatial partitioning. Investigating their behavior in the hyperbolic plane not only enhances our theoretical understanding of hyperbolic geometry but also contributes to the development of more efficient and effective algorithms tailored for complex, real-world datasets.

Chapter 2

Preliminaries

Contents

2.1 Basics of Hyperbolic Geometry

In this paper, we will be discussing the construction of the Voronoi diagram in Poincare's upper half-plane model.

2.2 Voronoi Diagrams

2.3 Fortune's Algorithm

Chapter 3

Voronoi Diagrams in the Hyperbolic Plane

Contents

3.1 Construction

3.2 Examples

Chapter 4

Applications and Relevance

Contents

4.1 Theoretical Importance

4.2 Practical Applications