Computing the Gromov hyperbolicity constant of a discrete metric space

Thesis by

Anas Nabil Ismail

Submitted in Partial Fulfillment of the Requirements for the degree of

Masters of Science

King Abdullah University of Science and Technology

Computer Science

Visual Computing

Thuwal, Makkah Province, Kingdom of Saudi Arabia

July, 2012

The undersigned approve the thesis of Anas Nabil Ismail

Dr. Committee Member 1 Name Committee Member	Signature	Date
Dr. Committee Member 2 Name Committee Member	Signature	Date
Dr. Advisor's Name Committee Chair(Thesis Supervisor)	Signature	Date

King Abdullah University of Science and Technology

2012

ABSTRACT

Computing the Gromov hyperbolicity constant of a discrete metric space

Anas Nabil Ismail

Although it was invented by Mikhail Gromov, in 1987, to describe some family of groups[1], the notion of Gromov hyperbolicity has many applications and interpretations in differnet fields. It has applications in Biology, Networking, Graph Theory, and many other areas of research. The Gromov hyperbolicity constant of several families of graphs and geometric spaces has been determined. However, so far, the only known algorithm for calculating the Gromov hyperbolicity constant δ of a discrete metric space is the brute force algorithm with running time $O(n^4)$ using the four-point condition. In this thesis, we first introduce an approximation algorithm which calculates a $O(\log n)$-approximation of the hyperbolicity constant δ, based on a layering approach, in time $O(n^2)$, where n is the number of points in the metric space. We also calculate the fixed base point hyperbolicity constant δ_r for a fixed point r using a $(\max, \min)-$matrix multiplication algorithm by Duan in time $O(n^{2.688})$ [2]. We use this result to present a 2-approximation algorithm for calculating the hyperbolicity constant in time $O(n^{2.688})$. We also provide an exact algorithm to compute the hyperbolicity constant δ in time $O(n^{3.688})$ for a discrete metric space. We then

present some partial results we obtained for designing some approximation algorithms to compute the hyperbolicity constant δ.

Acknowledgments

I would like to, sincerely, thank my supervisor Prof. Antoine Vigneron for his support, guidance, and for his encouragement throughout the course of the thesis. His valuable feedback and enthusiasm, for research, was very helpful. I would also like to thank him for providing me with an excellent research environment.

I would like also to thank Dr. Hervé Fournier for his collaboration and help. I learned from him during the two-week visit he paid to Kaust.

TABLE OF CONTENTS

List of Illustrations

List of Algorithms

Chapter 1

Introduction

In 1987, Gromov gave a new definition [1] of hyperbolic metric spaces based on a new inner product he introduced. A simpler definition is a generalization of tree metrics. This definition is called the four point condition definition. Tree metrics are very important and have many applications. For example, they define Phylogenetic Trees, which are of big importance in Biology to study DNA sequences of different species and to try to form a ”tree of life” that relates many species together [3]. Such trees encode some hierarchy and are defined based on some distance measures between different data points (DNA sequences). Due to some errors in measuring the data, Phylogenetic trees could be approximated by a Gromov hyperbolic space.

So the four point condition definition of Gromov hyperbolicity constant δ is as follows: A metric space S is said to be $\delta-$hyperbolic if $\forall a, b, c$, and $t \in S$, the two largest numbers in

$$\left\{d\left(a,b\right)+d\left(c,t\right),d\left(a,c\right)+d\left(b,t\right),d\left(a,t\right)+d\left(b,c\right)\right\}$$

differ by at most 2δ. This is called the four point condition. A space is Gromov hyperbolic if it is $\delta-$hyperbolic for some $\delta \geq 0$. The Gromov hyperbolicity constant of a metric space S is the smallest δ such that S is $\delta-$hyperbolic. Tree metrics correspond to $0-$hyperbolic spaces. That is, the two largest numbers in

$$\{d(a,b)+d(c,t)\,,d(a,c)+d(b,t)\,,d(a,t)+d(b,c)\}$$

are equal.

Given a discrete metric space S, one important question is: What is the Gromov hyperbolicity constant δ for S?

The brute force algorithm to calculate the Gromov hyperbolicity constant is simple. It loops over all four points $a,b,c,$ and t, and calculates the difference between the two largest numbers in

$$\{d(a,b)+d(c,t)\,,d(a,c)+d(b,t)\,,d(a,t)+d(b,c)\}$$

and returns half this difference as shown in the algorithm below.

Algorithm 1 Brute force algorithm

```
procedure BruteForce(M)                                    ▷ M is the metric
    two_delta← 0
    for i = 1 → n do
        for j = 1 → n do
            for k = 1 → n do
                for l = 1 → n do
                    A ← [M]_ij + [M]_kl
                    B ← [M]_ik + [M]_jl
                    C ← [M]_il + [M]_kj
                    Max ← max (A, B, C)
                    Mid ← second max (A, B, C)   ▷ the second max of A,B, and C
                    Diff ← Max−Mid
                    if diff > two_delta then
                        two_delta← diff
                    end if
                end for
            end for
        end for
    end for
    return two_delta/2.0
end procedure
```

The running time of this algorithm is clearly $\theta(n^4)$. Our goal is to provide faster exact and approximation algorithms to calculate the Gromov hyperbolicity constant δ for a discrete metric space S.

1.1 Motivation

The hyperbolicity of a metric space, especially a graph, has been considered a lot in many research papers in Computer Science and in different fields. This is due to its various applications. It also has some theoretical value.

1.1.1 Practical applications

One application of studying graph hyperbolicity is in networking, especially in routing. When a packet needs to be transmitted from one node to another in a network, the routing problem is the problem of specifying which node the packet should go to next, so that it can reach its target. It is preferable and, in some cases, essential to do this fast. One way to solve this problem is to save the topology of the whole network at each node in routing tables. If the network is huge, such as the internet for example, this solution doesn't seem feasible. One technique to efficiently solve this problem is to use greedy routing. That is, a packet is forwarded to the node that is the closest of the current node's neighbors to the destination node. One way to do this is to embed the graph of the network in some metric space. For example, in the case of geographical networks, the Euclidean Plane could be used [4]. It turns out to be the case that hyperbolic graphs having small hyperbolicity constants could be embeded efficiently in hyperbolic spaces and then some greedy routing algorithm could be used [4].

Also, the hyperbolicity of a graph could be used in computer security research. For example, it is used in research related to determining the speed at which a virus spreads in a network [5]. It could also be used to measure how fast rumors could spread in a social network.

1.1.2 Theoretical motivation

Many papers discuss Gromov hyperbolic spaces and some even build algorithms for those spaces. Many of those algorithms depend on the value of the hyperbolicity constant δ of the space. However, the fundamental problem of computing the hyperbolicity constant δ is not addressed, except using the naive brute force algorithm. In this thesis, we present faster algorithms to compute and approximate the value of δ.

1.2 Problem Statement and Objectives

Our problem can be stated as follows:

Given a metric space (S, M), can we compute the hyperbolicity constant δ in time $o(n^4)$? How fast can we approximate the value of δ?

We present a fast exact algorithm to compute the hyperbolicity constant δ with running time $O(n^{3.688})$, as well as two fast approximation algorithms.

1.3 Organization

This thesis is divided into four chapters. The first chapter is an introductory chapter. It presents a general overview of the problem, the objectives of this document, as well as its organization.

In chapter two, we present the necessary background, so that the reader would be familiar with the topic and is able to continue. We will introduce some general definitions, as well as some definitions of Gromov hyperbolicity and some equivalences between them. We will also talk about some applications of Gromov hyperbolicity in Biology and Networking. We also discuss some related work. We will mention some work related to some algorithms done for Gromov hyperbolic spaces, as well as using the concept of Gromov hyperbolicity to study some network properties.

In chapter three, we present our results. We present several algorithms. The first algorithm is an approximation algorithm. It calculates a $O(\log n)$-approximation of δ, based on a layering approach, in time $O(n^2)$, where n is the number of points in the metric space. The second algorithm is for computing a fixed base hyperbolicity constant δ_r in time $O(n^{2.688})$. Then we use it to design an 2-approximation algorithm for calculating the hyperbolicity constant δ. We also present an exact algorithm for calculating the hyperbolicity constant δ in time $O(n^{3.688})$. We then present some partial results we have, for designing some approximation algorithms to compute the hyperbolicity constant δ.

In chapter four, we conclude our work and we discuss some future work.

Chapter 2

Background Information and related works

In this chapter we introduce some required background information to help the reader. In the first section, we introduce some general preliminary information that will be used later. Then we discuss Gromov hyperbolic spaces. We present some definitions of Gromov hyperbolicity. In the last section, we discuss some of the applications of the concept of Gromov hyperbolicity in Biology and Networking.

2.1 Preliminary

In this section, we present some definitions and algorithms that will be useful later.

2.1.1 Metric Spaces

Definition.

Let S be an arbitrary set. A function $d : S \times S \rightarrow \mathbb{R} \cup \{\infty\}$ is a metric on S iff $\forall x, y$, and $z \in S$, the following [6] is satisfied:

1. Positiveness: $d\left(x, y\right) > 0$ if $x \neq y$, and $d\left(x, x\right) = 0$.

2. Symmetry: $d(x, y) = d(y, x)$.

3. Triangle inequality: $d(x, y) \leq d(x, z) + d(z, y)$.

The pair consisting of the set and the metric on it is called a metric space. So (S, d) is a metric space, where d is the metric on S. Usually elements of S are called points of that metric space, while $d(x, y)$ is called the distance between the point x and the point y. It is conventional, sometimes, to write "the metric space S" instead of "the metric space (S, d)" if there is no confusion about which metric is used. Since we are discussing discrete metrics, we will use (S, M) to represent a metric space, S is the set of points, and M is an $n \times n$ matrix, with $n = |S|$, such that $[M]_{ij} = d(i, j)$.

Examples.

1. The $n-$dimensional Euclidean space is the pair $(\mathbb{R}^n, d)$, where d is the Euclidean metric defined as follows:

$$d(X, Y) = \sqrt{\sum_{i=1}^{n} (X_i - Y_i)^2}$$

   where X and $Y \in \mathbb{R}^n$.

2. A metric can be defined over an undirected graph with positive weights as the shortest path distance. That is, $d(u, v)$ = the weight of the shortest path between vertex u and vertex $v \in V(G)$.

   Given a graph G, the metric M over this graph could be computed using the FloydWarshall algorithm in $O(n^3)$.

2.1.2 Geodesic Spaces

One of the main definitions of Gromov hyperbolicity depends on embedding a graph in a geodesic metric space. In this section, we will present some definitions that will be useful later.

Isometry. Let (X, d_X) and (Y, d_Y) be two metric spaces. An isometry is a map $\varphi : X \to Y$ such that for all a and $b \in X$, $d_X(a, b) = d_Y(\varphi(a), \varphi(b))$.

Geodesic. A geodesic between two points a and b in a metric space X is a locally distance minimizing curve. It is a map $\Upsilon : [0, L] \to X$ such that $\Upsilon(0) = a$, $\Upsilon(L) = b$, and $\forall t_1 < t_2 \in [0, L], d(\Upsilon(t_2), \Upsilon(t_1)) = t_2 - t_1$, where $d(x, y)$ is the distance between x and y. The map Υ is an isometry and L is the length of the geodesic.

Geodesic space. A metric space X is called a geodesic metric space iff all pairs a, $b \in X$ can be connected by a geodesic.

2.1.3 Hyperbolic Groups

The idea of Gromov hyperbolicity started when Gromov was working on hyperbolic groups[1].

Groups.

A group [7] is a mathematical structure consisting of a set G of elements as well as a binary operation $*$, called multiplication. This binary operation takes two elements of the group as input and produces an element of the group as output. The operation should have some properties called the group axioms:

1. Closure: $\forall g_1,\ g_2 \in G, g_1 * g_2 \in G$.
2. Associativity: $\forall g_1, g_2,\ g_3 \in G, g_1 * (g_2 * g_3) = (g_1 * g_2) * g_3$.
3. Existence of identity: $\exists$ an element $e \in G$, called the identity element, such that $\forall g \in G, e * g = g * e = g$.

4. Existence of inverse: $\forall g \in G, \exists g'$, called the inverse of g, such that $g * g' = g' * g = e$.

Example. $(\mathbb{Z}, +)$ is a group. In this example, $\mathbb{Z}$ is the set of integers and the binary operation is the addition. We say that $\mathbb{Z}$ forms a group under the operation of addition.

1. Closure: It is clear that if we add two integers, we get a new integer.
2. Associativity: Integers are associative under addition.
3. Existence of Identity: Zero is the identity for $\mathbb{Z}$ under addition.
4. Existence of inverse: The inverse of and integer a is $-a$.

Subgroups.

Given a group $(G, *)$, a subset T of G is called a subgroup, if and only if T forms a group under the same operation $*$.

Example. A subgroup of $(\mathbb{Z}, +)$ is $(2\mathbb{Z}, +)$, that is the set of even integers under the operation $+$.

Generating set.

A generating set of a group is a set of elements of G, such that all elements of G can be expressed as a multiplication of a finite number of elements from this set, and their inverses.

The generating set of $(\mathbb{Z}, +)$ is $\{1\}$ and we write $\mathbb{Z} =< 1 >$.

Cayley graphs.

A Cayley graph is a graph which encodes the mathematical structure of some group. It is also called a Cayley color graph.

Given a group G and its generating set S, the Cayley graph $C(G,S)$ is a coloured directed graph that is defined in the following way:

1. There is a vertex for Each element $g \in G$.
2. Each generator $s \in S$ has a color $c(s)$.
3. $\forall g \in G$ and $\forall s \in S$, we make a directed edge of color $c(s)$ connecting the vertex corresponding to element g and the vertex corresponding to the element $g * s$.

Below is figure 2.1 showing the Cayley graph of $(\mathbb{Z},+)$.

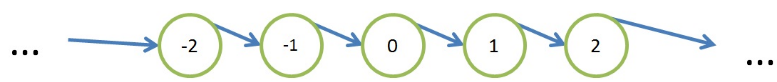

Figure 2.1: The Cayley graph of $(\mathbb{Z},+)$.

Word metric.

A word metric on a Group G is a metric defined over G that represents a measure of distance between any two elements g and $g' \in G$ [1]. Let G be a group and S is a generating set of G. Let $S^{-1} = \{s^{-1} : s \in S\}$. A word over $S \cup S^{-1}$ is defined to be a sequence, of finite length, $w = s_1, ..., s_n$, where n represents the length of the word w. We will assume that, in this sequence, there is no i, such that ${s_i}^{-1} = s_{i+1}$. This means that there are no two consecutive elements canceling each other. When the elements of a word w are multiplied, the result is an element $g \in G$, and this process is called evaluation of w. The empty word evaluates to the identity element of the group. Given $g \in G$, the word norm $|g|$ is the length of the shortest word w composed from $S \cup S^{-1}$, such that w evaluates to g. The distance between two elements g_1 and $g_2 \in G$, $d(g_1, g_2)$ is $|{g_1}^{-1} g_2|$. This is the length of the shortest word w such that the evaluation of $g_1 w = g_2$. The word metric of a group G is related to its Cayley graph. $d(g_1, g_2)$ is also defined to be the length of the shortest path between nodes corresponding to g_1 and g_2 in the Cayley graph.

Hyperbolic Groups.

A group G is hyperbolic iff its word metric is hyperbolic. Equivalently, a group is hyperbolic iff its Cayley graph is hyperbolic.

2.1.4 Tree Approximation

Let $G = (V, E)$ be a graph and $d_G(u, v)$ be the shortest path distance between vertex u and vertex $v \in V$. A tree $T = (V, E')$ is a distance $\epsilon-$approximation of G if $\forall u$ and $v \in V$, $|d_G(u, v) - d_T(u, v)| \leq \epsilon$, where $d_T(u, v)$ is the distance between vertex u and vertex v in T. Chepoi and Dragan [8, 9] presented an algorithm that constructs an $O(\delta \log n)-$approximation tree in time $O(|E|)$. This algorithm makes a leveling of the graph G and constructs a tree to approximate the distance. A leveling of a graph G with respect to u, a base point, is to partition V into spheres $S_t(u)$, $t = 0, 1, 2, ...$, where $S_t(u)$ is a sphere of radius t centered at u. $S_t(u) = \{v \in V : d_v(G, u) \leq t\}$. Figure 2.2 shows the basic of idea of how this algorithm works.

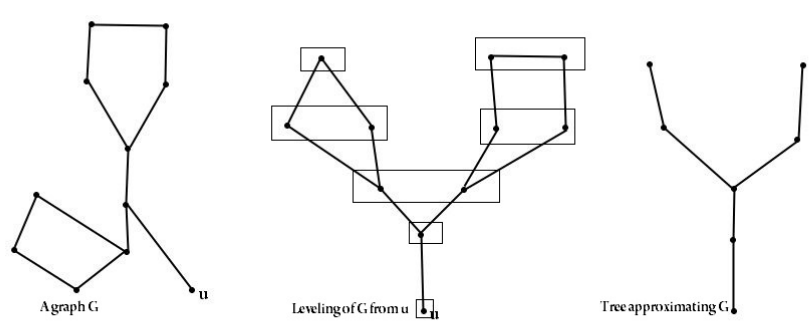

Figure 2.2: Constructing an approximation tree of a graph G.

We will use this algorithm to provide an $O(\log n)$ −approximation algorithm to approximate the value of, δ, the Gromov hyperbolicity constant of the graph G.

2.1.5 $(\max - \min)$ −matrix multiplication algorithm

In their paper, titled "Fast Algorithms for (max, min)−Matrix Multiplication and Bottleneck Shortest Paths", Duan and Pettie introduced a fast algorithm to do (max, min)−Matrix Multiplication [2]. The running time of this algorithm is $O(n^{2.688})$. We will use this matrix multiplication algorithm as a subroutine in our algorithms.

Definition. Given two real matrices A and B, the $(\max - \min)$ product of A and B is defined as follows:

$$[A \otimes B]_{ij} = \max_k \min\{[A]_{ik}, [B]_{kj}\}$$

as shown in figure 2.3.

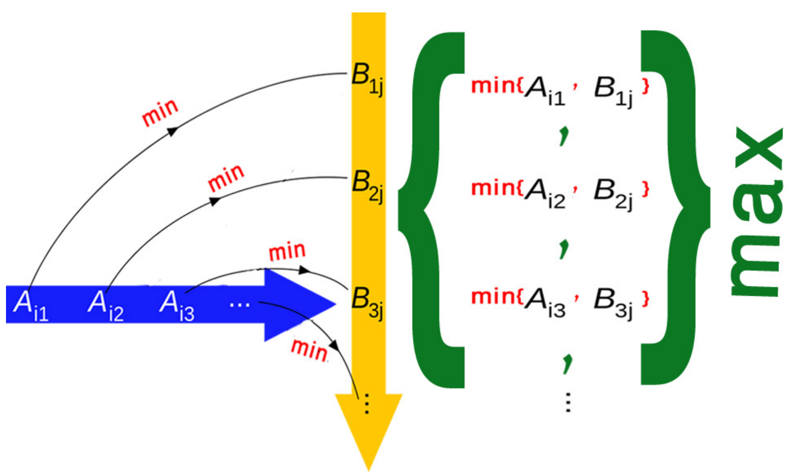

Figure 2.3: $(\max - \min)$ −matrix multiplication.

We will use this fast algorithm as a subroutine in our algorithms later.

2.1.6 Chinese remainder theorem

The Chinese remainder theorem [7] is a famous theorem in number theory that was first presented in a book by the Chinese mathematician Sun Tzu around $A.D.100$. It determines an integer x which, when divided by the given divisors, leaves out the given remainders.

Definitions.

1. Two integers a and b are relatively prime iff their greatest common divisor $\gcd(a, b) = 1$.

2. $x \equiv a(\mod n)$ means that $x - a$ is divisible by n.

3. The integers $a_1, a_2, ..., a_n$ are pairwise relatively prime if every two of them are relatively prime. That is, $\forall i \neq j \ \gcd(a_i, a_j) = 1$.

The Chinese remainder theorem [7] states that: If $n = n_1 \times n_2 \times ... \times n_k$, where $n_1, n_2, ..., n_k$ are pairwise relatively prime, then for any integers $a_1, a_2, ..., a_k$, $x \equiv a_i(\mod n_i)$, for $i \in \{1, 2, ..., k\}$, has a unique solution modulo n, for x.

The following theorem [7] will be used in some proof later.

Theorem 1 (Chinese remainder theorem)**.** *If $n = n_1 \times n_2 \times ... \times n_k$, where $n_1, n_2, ..., n_k$ are pairwise relatively prime, then for all x and a, $x \equiv a(\mod n_i)$, for $i = 1, 2, ..., k$ $\iff$ $x \equiv a(\mod n)$.*

2.2 Gromov Hyperbolic Spaces

There are many definitions for Gromov hyperbolicity of a metric space. Here we will only discuss three of them. They are:

1. The Gromov inner product definition.

2. Slim triangle definition.

3. The four point condition definition.

Different definitions could be useful in different situations.

2.2.1 Gromov inner product definition

In this section, I will summarize the Gromov inner product definition presented by Gromov in his paper [1].

Gromov inner product.

Given a metric space X, the distance between two points x and $y \in X$, could be denoted by $d(x, y)$ or $|x - y|$. If we choose some point $r \in X$ to be a reference point, then the distance between $x \in X$ and r could be denoted by $d(x, r)$, $|x - r|$, $|x|$, or $|x|_r$. Gromov defined an inner product on a metric space X with a base point r. The inner product of two points x and $y \in X$ is defined as follows:

$$(x|y)_r = \frac{1}{2}\left(d(r, x) + d(r, y) - d(x, y)\right).$$

Hyperbolicity.

A metric space X, with a base point r, is hyperbolic, if it satisfies the following inequality:

$$(x|y)_r \geq \min((x|z)_r, (z|y)_r) - \delta$$

for a fixed $\delta \geq 0$ and all x, y, and $z \in X$.

We say that X is δ_r-hyperbolic with respect to a base point r. A metric space is $\delta-$hyperbolic if $\forall r \in X$, $\delta_r < \delta$. We call X hyperbolic if it is $\delta-$hyperbolic for some $\delta \geq 0$. The Gromov hyperbolicity constant of a metric space X is the smallest δ such that X is $\delta-$hyperbolic. Our goal is to provide algorithms to compute exactly and approximately the value of the Gromov hyperbolicity constant of a given metric space.

Examples of Gromov hyperbolic spaces.

1. A tree metric is hyperbolic with Gromov hyperbolicity constant 0. The inner product of two points x and y, $(x|y)_r = \frac{1}{2}(d(r,x) + d(r,y) - d(x,y))$, represents the distance between the reference point r and the least common ancestor of x and y as shown in figure 2.4.

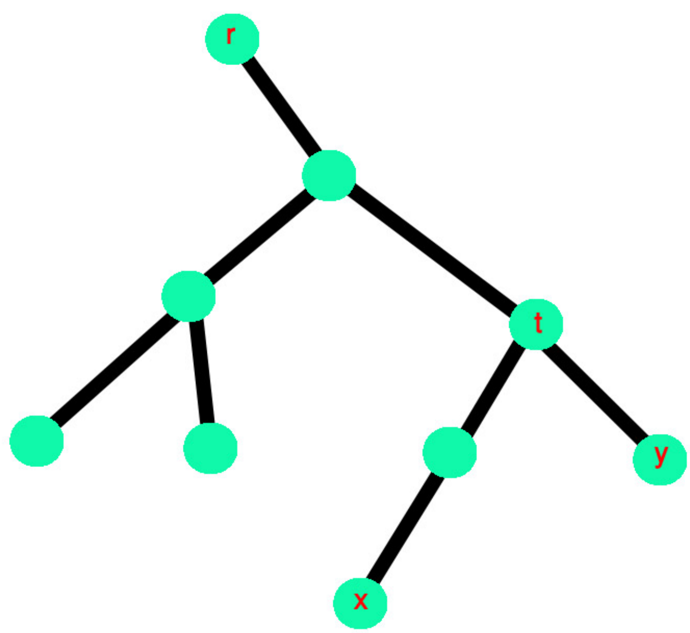

Figure 2.4: Visualization of the meaning of the Gromov inner product on a tree. It represents the distance between the reference point r and the least common ancestor t, of x and y.

2. The Euclidean Plane R^2 is not hyperbolic while the real line R^1 is $0-$hyperbolic.

It might not be very obvious to see why this is the case, but it will be more obvious using another definition.

The inner product definition of Gromov hyperbolicity of a metric space implies that, for a fixed base point r, $\forall x, y, z \in X$, the two minimums of $\{(x|y)_r, (z|y)_r, (x|z)_r\}$ differ by at most δ_r. The Gromov hyperbolicity constant of a metric space X is $\max_r \delta_r$.

Properties of Gromov inner products.

1. $(x|y)_r = (y|x)_r$.
2. $(x|r)_r = (r|x)_x = 0$.
3. $\forall x, y$, and r, $(x|y)_r \geq 0$.

Proof. The proof of 1 and 2 follows directly from direct substitution. The proof of 3 follows from the triangle inequality.

2.2.2 Slim triangle definition

Gromov's definition of a hyperbolic space generalizes the idea of standard hyperbolic spaces with negative curvature. It doesn't only include Riemannian manifolds, but discrete spaces as well. In the hyperbolic plane H^2 , the sum of the angles of a triangle is less than π by a positive amount called the defect. The area of a hyperbolic triangle is proportional to this defect. Area=defect$\times\frac{1}{-k}$, where k is the curvature of the space, and it is negative for hyperbolic spaces. So, a triangle in a hyperbolic space looks slimmer than a triangle in the Euclidean space, for example, as shown in figure 2.5.

The area of a triangle is a measure of how slim it is. If a triangle is Gromov hyperbolic, then we can talk about how slim it is using the basic concept of the

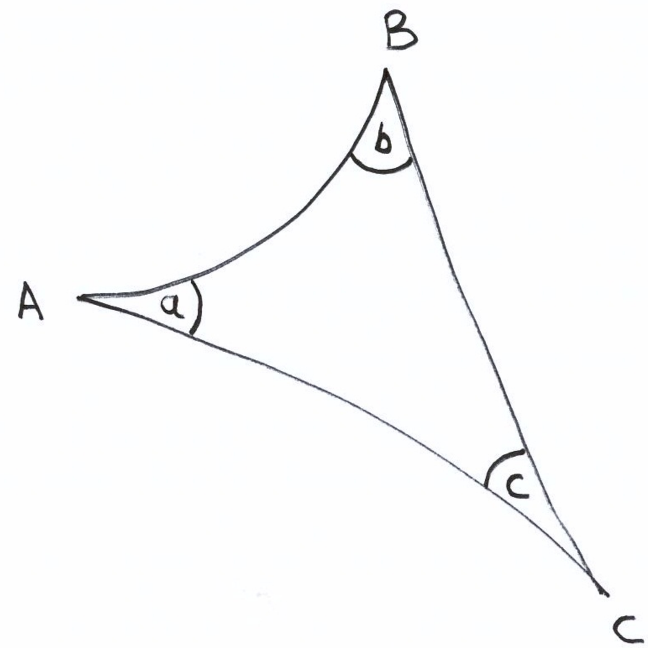

Figure 2.5: Hyperbolic triangle.

distance, instead of the area.

Slim triangle.

This definition is attributed to Rips. Given any three points x, y, and z in a geodesic space X, the geodesic triangle connecting these points is $\delta-$slim iff each side of the triangle is contained in a $\delta-$neighborhood of the union of the other two sides. Figure 2.6 shows a slim triangle.

Triangles in a geodesic space are slim if there is $\delta \geq 0$ such that all geodesic triangles in X are $\delta-$slim.

Connection to Gromov hyperbolicity.

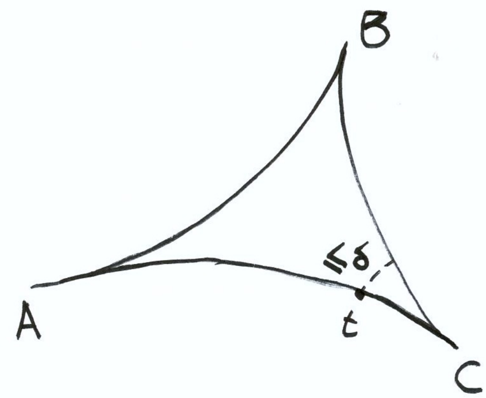

Figure 2.6: Slim triangle.

Both definitions are equivalent up to a small constant. The proof of this equivalence is presented in [10]. Using the definition for Gromov hyperbolicity it is easy to see why the Euclidean Plane R^2 is not hyperbolic while the real line R^1 is $0-$hyperbolic.

For a triangle $\triangle xyr$, in a geodesic space, the inner product $(x|y)_r$ measures the extent to which the triangle inequality of $\triangle xyr$ is far from being an equality [6].

2.2.3 The four point condition definition

Another definition of Gromov hyperbolicity, that doesn't require the space to be a geodesic space, is the four point condition definition. It removes the need for geodesics, but it requires four points instead of three.

Definition.

let a, b, c, and d be four points in a metric space X. Let L, M, and S be the largest, middle, and smallest numbers in $\{(d\,(a,b) + d\,(c,d)), (d\,(a,c) + d\,(b,d)), (d\,(a,d) + d\,(c,b))\}$. We denote the value of $\frac{L-M}{2}$, by $\delta'(a,b,c,d)$.

A metric space X is $\delta-$hyperbolic if $\forall a, b, c,$ and $d \in X$, $\delta'(a, b, c, d) \leq \delta$. The Gromov hyperbolicity constant of a metric space X, is the smallest δ such that X is $\delta-$hyperbolic. This is called the four point condition definition of the Gromov hyperbolicity.

While the slim triangle definition makes it easier to visualize the intuitive meaning of Gromov hyperbolicity, the four point condition definition is computationally easier to work with than the slim triangle definition.

Connection to the Gromov inner product definition.

The two definitions, the Gromov inner product and the four point condition, are exactly the same. That is, a metric space is $\delta-$hyperbolic using the Gromov inner product definition iff it is $\delta-$hyperbolic using the four point condition definition.

Proof. Proof that the inner product definition implies the 4-point condition:

The inner product definition of Gromov hyperbolicity states that $\forall x, y, z, r$, we have:

$(x|y)_r \geq \min\limits_{x,y,z}\{(y|z)_r\,, (x|z)_r\} - \delta_r.$

$\delta_r \geq \min\limits_{x,y,z}\{(y|z)_r\,, (x|z)_r\} - (x|y)_r$

but we know that : $(y|z)_x = \frac{1}{2}\left(d\left(x, y\right) + d\left(x, z\right) - d\left(y, z\right)\right)$

so, $\delta_r \geq \frac{1}{2}\min\limits_{x,y,z}\{\left(d\left(z, r\right) + d\left(x, y\right) - d\left(x, z\right) - d\left(y, r\right)\right), \left(d\left(z, r\right) + d\left(x, y\right) - d\left(y, z\right) - d\left(x, r\right)\right)\}$

$\delta_r \geq \frac{1}{2}\left(d\left(z, r\right) + d\left(x, y\right)\right) + \frac{1}{2}\min\limits_{x,y,z}\{\left(-d\left(x, z\right) - d\left(y, r\right)\right), \left(-d\left(y, z\right) - d\left(x, r\right)\right)\}$

$\delta_r \geq \frac{1}{2}\left(d\left(z, r\right) + d\left(x, y\right) - \max\limits_{x,y,z}\{\left(d\left(x, z\right) d\left(y, r\right)\right), \left(d\left(y, z\right) d\left(x, r\right)\right)\}\right)$

$\forall x, y, z, r$

which implies the four point definition. To prove the other direction, we follow the same proof from bottom to up.

2.3 Applications

2.3.1 Biology

Phylogenetic trees.

A metric space is a tree metric if its points could be leaves of some tree that preserves the distances [11]. A Phylogenetic tree is a tree that shows some evolutionary relationships between different species based on some similarities between them [3]. Those similarities can be based on their DNA sequences. Those trees can be defined by a tree metric and the distance between two nodes on the tree, is a measure of the dissimilarity between the DNA sequences of the corresponding species. Figure 2.7 gives a simplified idea about such a tree.

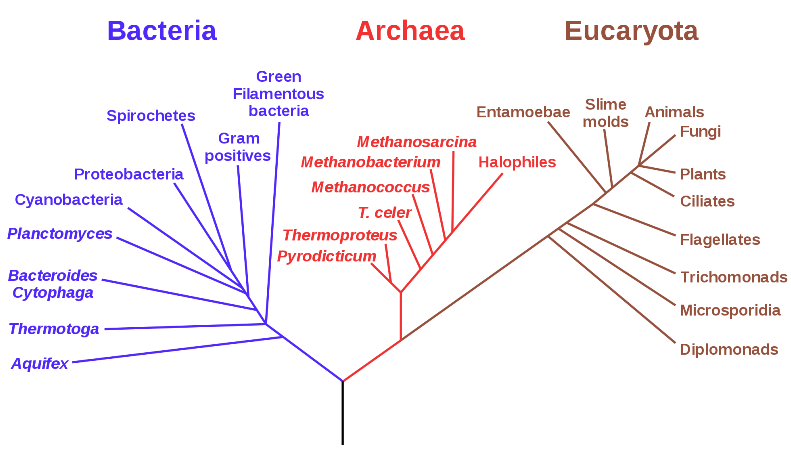

Figure 2.7: Phylogenetic tree.

Tree metrics are a special kind of Gromov hyperbolic spaces. They correspond to $0-$hyperbolic spaces. In fact, the Gromov hyperbolicity constant of a metric space is a measure of how far it is from a tree.

The smaller the Gromov hyperbolicity constant of a metric space, the closer it is to a tree metric, as shown in figure 2.8.

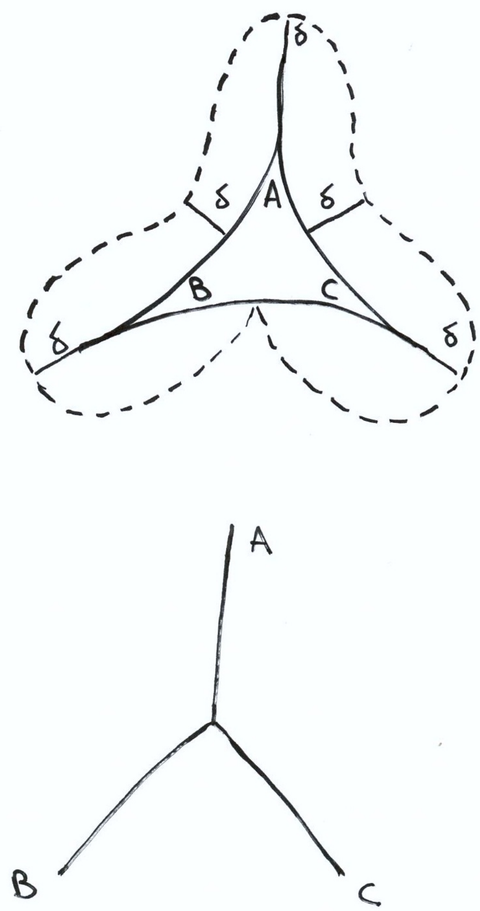

Figure 2.8: The smaller the value of δ of a metric space, the closer it is to a tree metric. Idea from [12].

Due to some errors in measurements, some trees could be embedded in some hyperbolic spaces. This embedding should be with a small distortion. So, more understanding of Gromov hyperbolicity will help build more accurate Phylogenetic trees.

2.3.2 Networking

The concept of Gromov hyperbolicity has many applications in networking. Some properties of a network depend on the hyperbolicity of the graph of this network. It has many applications in routing, navigation, and congestion on a network.

In [13], Jonckheere, is studying which families of networks exhibit some properties that are expected from Riemannian manifolds with negative curvatures. They investigate for which values of δ, the network exhibits some useful properties.

In [14], they study the hyperbolicity of some classes of networks and its effect on the navigability of the network. They also study how adding random connections in the network affects the navigability and how this is related to the value of the Gromov hyperbolicity constant of the network.

In [15], Chepoi and Dragan investigate some properties of networks that depend on the value of the Gromov hyperbolicity constant of the network. They show that some families of networks have some parameters that depend on the value of δ. These parameters include, the number of bits in routing label schemes, routing time, and some expected errors.

2.3.3 Algorithms for Gromov hyperbolic spaces

In [8], Chepoi and Dragan introduce some algorithms and calculate values of some quantities related to a metric space. They assume that the given metric space is geodesic. In fact any Gromov hyperbolic space could be embedded in a geodesic space with some distortion.

In this paper they discuss and investigate some important quantities that could be calculated for a hyperbolic space. They investigate and find some fast approximations for the diameter, radius, and the center of a given hyperbolic space. They also introduce the tree approximation algorithm we mentioned. They also investigate the hyperbolicty of some family of graphs.

In [16], Krauthgamer and Lee discuss some approximate algorithms on Gromov hyperbolic spaces. They discuss embeddings of Gromov hyperbolic spaces. They also provide fast approximation algorithms for nearest neighbor search and for the traveling salesman problem on hyperbolic spaces.

Most algorithms, in the literature, for Gromov hyperbolic spaces, assume that the space is geodesic. And so far, there was no algorithm, other than the naive brute force algorithm, presented to calculate the Gromov hyperbolicity constant of a discrete metric space, either exactly or approximately.

Chapter 3

Results

3.1 Main Results

In this section, we present results we have for calculating the Gromov hyperbolicity constant. First we present an $O(\log n)$-approximation of the hyperbolicity constant δ, based on the layering approach presented in Section 2.1.4.

We also present an algorithm to calculate a fixed point hyperbolicity constant δ_r in $O(n^{2.688})$, based on the $(\max, \min)-$matrix multiplication algorithm presented in Section 2.1.5. We use this algorithm to introduce a 2-approximation algorithm for calculating δ in $O(n^{2.688})$. We also use it to construct an exact algorithm for calculating the hyperbolicity constant δ in $O(n^{3.688})$.

3.1.1 Algorithm: $O(\log n)$-approximation

Given a metric M of a graph G, this algorithm computes an $O(\log n)$-approximation of the value of δ. That is, our algorithm returns t such that $\delta \leq t \leq (c\delta \log n)$, where $c > 0$ and δ is the optimal value for the hyperbolicity constant for this metric space.

Algorithm 2 $O(\log n)$-approximation

```
procedure LognApproximation(M)
    m ← 0
    T ← ApproximationTree(G)                 ▷ using the algorithm mentioned before
    A ← All_pairs_shortest_path (T)
    for i = 1 → n do
        for j = 1 → n do
            d ← |[A]_ij − [M]_ij|
            if m < d then
                m ← d
            end if
        end for
    end for
    return t = 4m
end procedure
```

How it works.

Let $d_G(a,b)$ be the shortest distance between a and b in the graph G and $d_T(a,b)$ be the distance between a and b in the approximation tree of G. We know that for all a and $b \in S$, $|d_G(a,b) - d_T(a,b)| = O(\delta \log n)$.

Let $m = \max_{u,v} |d_G(a,b) - d_T(a,b)|$. So $m = O(\delta \log n)$.

let $\delta_G(a,b,c,d)$ =the difference between the maximum and the middle values in $\{d_G(a,b)+d_G(c,d), d_G(a,c)+d_G(b,d), d_G(a,d)+d_G(b,c)\}$ and $\delta_T(a,b,c,d)$ =the difference between the maximum and the middle values in $\{d_T(a,b)+d_T(c,d), d_T(a,c)+d_T(b,d), d_T(a,d)+d_T(b,c)\}$.

For all $a,b,c,d \in V$, we have $\delta_T(a,b,c,d)+4m \geq \delta_G(a,b,c,d)$. Then $\delta_T(a,b,c,d)+4m \geq \delta$. But $\delta_T(a,b,c,d) = 0$. So, $4m \geq \delta$. So G is $4m-$hyperbolic. So, we have $\delta \leq t = 4m \leq (c\delta \log n)$, where $c > 0$. Therefore our algorithm provides a $O(\log n)-$approximation of the value of the Gromov hyperbolicity constant of the graph G

Analysis.

Step 3: $O\left(n^2\right)$

Step 4: $O\left(n^2\right)$, by running n breadth-first searches.

Steps 5 and 6: $O\left(n^2\right)$

Total : $O\left(n^2\right)$

3.1.2 Algorithm: Calculating a Fixed Point Hyperbolicity Constant δ_r

This algorithm calculates the Gromov hyperbolicity constant δ_r with respect to a fixed base point r. From Gromov's inner product definition,

$$\delta_r = \max_{i,k,j}\{\min\{(i|k)_r, (k|j)_r\} - (i|j)_r\}$$

Algorithm description.

This algorithm works as follows: the input is given as a point r in the metric space S, an $n \times n$ matrix M containing distances between points from S; $[M]_{ij}$ contains the distance between point i and point $j \in S$. We then calculate a new $n \times n$ matrix A_r, called the matrix of Gromov inner products with respect to the base point r. This matrix A_r is defined as follows: $[A_r]_{ij} = (i|j)_r = \frac{1}{2}\left(d\left(r,i\right) + d\left(r,j\right) - d\left(i,j\right)\right)$. Then we calculate ${A_r}^2 = A_r \otimes A_r$, where $\otimes$ means $(\max, \min)-$matrix multiplication. Define $B = {A_r}^2 - A_r$. Then δ_r is the *maximum* coefficient of B.

Algorithm.

Algorithm 3 Fixed base-point algorithm

1: **procedure** FixedBasePoint(M, r) ▷ M is the metric and r is the base point
2: $[A_r]_{ij} \leftarrow (i|j)_r = \frac{1}{2}\left([M]_{ri} + [M]_{rj} - [M]_{ij}\right)$
3: ${A_r}^2 \leftarrow A_r \otimes A_r$ ▷ $(\max, \min)-$Matrix Multiplication
4: $B \leftarrow {A_r}^2 - A_r$
5: $c \leftarrow \max\limits_{i,j} B$
6: **return** c
7: **end procedure**

Analysis.

Step 2: $O\left(n^2\right)$

Step 3: $O\left(n^{2.688}\right)$

Step 4: $O\left(n^2\right)$

Step 5: $O\left(n^2\right)$

Total : $O\left(n^{2.688}\right)$

Proof of correctness.

In step 5 of the algorithm, we return c, which we claim to be the fixed base point Gromov hyperbolicity constant δ_r. The proof goes as follows:

$c = \max\limits_{i,j} B$

$c = \max\limits_{i,j}\{A_r \otimes A_r - A_r\}$

$A_r \otimes A_r = \max\limits_{k}\{\min\{[A_r]_{ik}, [A_r]_{jk}\}\}$

$c = \max\limits_{i,j}\{\max\limits_{k}\{\min\{[A_r]_{ik}, [A_r]_{jk}\}\} - [A_r]_{ij}\}$

$c = \max\limits_{i,j}\{\max\limits_{k}\{\min\{[A_r]_{ik}, [A_r]_{jk}\} - [A_r]_{ij}\}\}$

$c = \max\limits_{i,j} \max\limits_{k}\{\min\{[A_r]_{ik}, [A_r]_{jk}\} - [A_r]_{ij}\}$

$c = \max\limits_{i,j,k}\{\min\{[A_r]_{ik}, [A_r]_{jk}\} - [A_r]_{ij}\}$

$c = \max_{i,j,k}\{\min\{(i|k)_r, (j|k)_r\} - (i|j)_r\}$. This is the same as Gromov inner product definition of the Gromov hyperbolicity constant, which proves the correctness of our algorithm.

3.1.3 Algorithm: A 2-approximation algorithm for calculating the hyperbolicity constant δ

We present a $2-$approximation algorithm to approximate the value of the hyperbolicity constant for a general metric space S. That is, our algorithm returns δ such that $\delta \leq \delta^* \leq 2\delta$, where δ^* is the optimal value for the hyperbolicity constant for this metric space.

If we calculate the hyperbolicity constant δ_r for a fixed base point r, as in the previous algorithm, It turns out that δ_r is a 2-approximation for the optimal value of the hyperbolicity constant δ^* for S.

Proof.

In his paper [1], Gromov stated, in corollary $1.1.B$, that a metric space S is δ_r-hyperbolic with respect to a fixed base point $r \in S$, then S is $2\delta_r-$hyperbolic with respect to any other point $s \in S$. Gromov presented a lemma($1.1.A$), without a proof, then he used this lemma to prove the main result. Another proof for using the lemma to prove the result could be found in the proof of proposition 2.2 in [10]. Here, we will provide a proof of the lemma.

Lemma. If S is $\delta-$hyperbolic then

$$(x|y)_r + (t|z)_r \geq \min((x|z)_r + (t|y)_r, (t|x)_r + (y|z)_r) - 2\delta$$

for all $x, y, z, t \in S$

Proof. $(x|y)_r + (t|z)_r \geq \min((x|t)_r\,, (t|y)_r) + (t|z)_r - \delta$

$(x|y)_r + (t|z)_r \geq \min((x|t)_r + (t|z)_r\,, (t|y)_r + (t|z)_r) - \delta$

$(x|y)_r + (t|z)_r \geq \min((x|t)_r + \min((z|y)_r\,, (y|t)_r), (t|y)_r + \min((z|x)_r\,, (x|t)_r)) - 2\delta$

Now we have four cases:

1. $(z|y)_r \leq (y|t)_r$ and $(z|x)_r \leq (x|t)_r$:

   $(x|y)_r + (t|z)_r \geq \min((t|x)_r + (y|z)_r\,, (x|z)_r + (t|y)_r) - 2\delta$, which is exactly what we want.

2. $(z|y)_r \leq (y|t)_r$ and $(x|t)_r \leq (z|x)_r$:

   $(x|y)_r + (t|z)_r \geq \min((t|x)_r + (y|z)_r\,, (x|t)_r + (t|y)_r) - 2\delta$. If $A \geq \min(B, C)$, and $B \leq C$, then $A \geq \min(B, D)$, where $D \geq C$. So we have the following $(x|y)_r + (t|z)_r \geq \min((t|x)_r + (y|z)_r\,, (x|z)_r + (t|y)_r) - 2\delta$, which is what we want.

3. $(y|t)_r \leq (z|y)_r$ and $(z|x)_r \leq (x|t)_r$:

   $(x|y)_r + (t|z)_r \geq \min((y|t)_r + (x|t)_r\,, (z|x)_r + (y|t)_r) - 2\delta$. Using a similar argument as in the above case, we can reach $(x|y)_r + (t|z)_r \geq \min((t|x)_r + (y|z)_r\,, (x|z)_r + (t|y)_r) - 2\delta$

4. $(y|t)_r \leq (z|y)_r$ and $(x|t)_r \leq (z|x)_r$:

   In this case we find out that $(x|y)_r + (t|z)_r \geq (y|t)_r + (x|t)_r - 2\delta$. This doesn't lead anywhere. So we will use another derivation to show that if $(y|t)_r \leq (z|y)_r$ and $(x|t)_r \leq (z|x)_r$, then we achieve our target.

   $(x|y)_r + (t|z)_r \geq \min((x|z)_r\,, (z|y)_r) + (t|z)_r - \delta$

   $(x|y)_r + (t|z)_r \geq \min((x|z)_r + (t|z)_r\,, (z|y)_r + (t|z)_r) - \delta$

   $(x|y)_r + (t|z)_r \geq \min((x|z)_r + \min((z|y)_r\,, (y|t)_r), (z|y)_r + \min((z|x)_r\,, (x|t)_r)) - 2\delta$

   Now if $(y|t)_r \leq (z|y)_r$ and $(x|t)_r \leq (z|x)_r$, then it follows directly that $(x|y)_r + (t|z)_r \geq \min((x|z)_r + (t|y)_r\,, (t|x)_r + (y|z)_r) - 2\delta$ $\square$

Analysis.

$O\left(n^{2.688}\right)$

3.1.4 Algorithm: An exact algorithm for calculating the hyperbolicity constant δ

This algorithm calculates the exact value of the Gromov hyperbolicity constant δ in $O\left(n^{3.688}\right)$.

Algorithm Description.

This algorithm is a simple extension to the fixed point algorithm. The input is given as an $n \times n$ matrix M containing the distances between points from S; $[M]_{ij}$ contains the distance between point i and point $j \in S$. $\forall r$in the metric space S, calculate δ_r. The optimal value of the hyperbolicity constant δ^* is the max over all δ_r.

Algorithm.

Algorithm 4 Exact algorithm to hyperbolicity

```
1: procedure ComputeHyperbolicity(S, M)
2:     for all r ∈ S do
3:         D[r] ← FixedBasePoint(M, r)
4:     end for
5:     return max D
6: end procedure
```

Analysis.

Step 2: $O\left(n\right)$

Step 3: $O\left(n^{2.688}\right)$

Step 5: $O\left(n\right)$

Total : $O\left(n^{3.688}\right)$

3.2 Partial Results

In this section we present some partial results we have. Those partial results were produced in the process of searching for an approximation algorithm to approximate the hyperbolicity constant δ of a metric space S.

We would like to approximate the value of δ by using some rounding scheme.

We will represent the Gromov inner products, for a fixed base r, by colors in a matrix and we will try to detect some triangles with some properties.

We would like to have triangles with three colors such that the smallest two colors are equal after the rounding.

We want to avoid two cases:

- Triangles with colors $i < j < k$
- Triangles with colors $i < j = k$

So far we have an algorithm to detect the first case only. A triangle with 3 different colors is called a rainbow triangle.

3.2.1 Rainbow Triangle Detection in 3-colored graphs

The Rainbow triangle detection problem is as follows:
Given a graph G with edges colored one of three colors, lets say $\{Red, Green, Blue\}$, decide whether there is a triangle with edges having different colors[17].

Proposed Algorithm.

The input is an adjacency matrix A, such that $[A]_{ij} = r$ iff the edge connecting nodes i and j is red, $[A]_{ij} = g$ iff the edge connecting nodes i and j is green, and $[A]_{ij} = b$ iff the edge connecting nodes i and j is blue. The idea of the algorithm is simple. It checks if there exists i, j such that, there a path of length one with color blue, and a path of length two, with one green edge and one red edge, between nodes i and j. We use boolean matrix multiplication for this.

Algorithm.

Algorithm 5 Rainbow Triangle Dection

1: **procedure** RAINBOW?(A)
2: $[R]_{ij} \leftarrow 1 \iff [A]_{ij} = r, 0$ otherwise
3: $[G]_{ij} \leftarrow 1 \iff [A]_{ij} = g, 0$ otherwise
4: $[B]_{ij} \leftarrow 1 \iff [A]_{ij} = b, 0$ otherwise
5: $D = R \odot G$ ▷ boolean multiplication
6: $F \leftarrow D \& B$ ▷ boolean and operation
7: $c \leftarrow \sum_{ij} F_{ij}$ ▷ add all elements in D
8: **if** $c > 0$ **then**
9: **return** true
10: **else**
11: **return** false
12: **end if**
13: **end procedure**

At step 5, $[D]_{ij}$ =1 iff there is a path of length two between node i and node j, and this path has a red edge and a green edge. At step 6, $[F]_{ij}$ =1 iff there is a blue edge between node i and node j and there is a path of length two, between the two nodes, that has a red edge and a green edge.

Analysis.

Step 2: $O(n^2)$

Step 3: $O(n^2)$

Step 4: $O(n^2)$

Step 5: $O(n^{2.3727})$ [18]

Step 6: $O(n^2)$

Step 7: $O(n^2)$

Total : $O(n^{2.7327})$

3.2.2 Rainbow Triangle Detection in $k-$colored graphs

We are trying to reduce the problem of calculating the Gromov hyperbolicity constant to finding a colored triangle, of some property, in a given matrix. Here, the matrix will contain the Gromov inner products based on a fixed base point r. That is $[A_r]_{ij} = (i|j)_r$.

$k-$colored triangles.

In this problem, we try to detect a rainbow triangle in a matrix A of k colors. The naive approach is to, for all triples of distinct colors a, b, and c, apply the Rainbow triangle detection algorithm explained in Algorithm 5. There are $O(k^3)$ such triples. Each will need $O(n^{2.3727})$ time, resulting in total running time of $O(k^3 n^{2.3727})$. This is not feasible because in our problem k might be as big as n^2. To solve this problem, we will try to make k smaller to have a more efficient algorithm.

Algorithm.

We will reduce the number of different colors by computing $A_t = A$ modulo p_t, where p_t is some prime number, such that $[A_t]_{ij} = [A]_{ij} \bmod p_t$. We will do this for different prime numbers p_t. We will choose the first m prime numbers $(p_1, p_2, ..., p_m) = (2, 3, 5, ...)$. We choose m to be the smallest integer such that the product $p_1 \times p_2 \times$

... $\times p_m > n^6$. Since p_t is at least 2, then $m = O(\log n)$. It is also known that the m^{th} prime number $p_m = \theta(m \log m)$, so $p_m = \log n \log\log n$. Now the algorithm goes as follows: for all $t \in \{1, 2, ..., m\}$, calculate A_t, such that $[A_t]_{ij} = [A]_{ij} \bmod p_t$ and apply the previous algorithm on A_t and return true if it returns true. If for all $t \in \{1, 2, ..., m\}$, Rainbow?(A_t) returns false, return false.

How it works.

If for some $t \in \{1, 2, ..., m\}$, we find a rainbow triangle in A_t, this means that there is a rainbow triangle in A. Conversely, assume that there is a rainbow triangle in A with colors $a < b < c$ and it is not detected after applying Rainbow?(A_t) for all t, we will show that this leads to a contradiction. Now we have $0 < (b-a)(c-a)(c-b) < n^6 < p_1 \times p_2 \times ... \times p_m$. By the Chinese remainder theorem, if for all t, $(b-a)(c-a)(c-b) \bmod p_t$ is zero, then we would have $(b-a)(c-a)(c-b) = 0 \bmod p_1 \times p_2 \times ... \times p_m$, which is a contradiction.

Analysis.

With this improved algorithm, we have $k = O(\log n \log\log n)$ colors and the naive algorithm is repeated $m = O(\log n)$ times with a running time of $O(n^{2.3727})$. This gives total running time of $O\left(n^{2.3727} \log^4 n (\log\log n)^3\right)$.

Chapter 4

Concluding Remarks

4.1 Summary

In this thesis we presented some algorithms to compute exact and approximate values of the Gromov hyperbolicity constant of discrete metric spaces. We presented some overview of the definitions of Gromov hyperbolic spaces. Then we presented some of the applications of the concept of Gromov hyperbolicicty in Networking and in Biology. We also presented a survey of related works for either computing the value of the hyperbolicity constant and some algorithms for Gromov hyperbolic spaces.

We also provided some fast algorithms to compute an exact and approximate value for Gromov hyperbolicity constant δ.

4.2 Future Work

As a future work, we could continue developing on some of the partial results we have. We could investigate if there is some fast algorithms for detecting $2-$colored triangles. This may help produce better approximations algorithms. Another point to investigate is the relation between Gromov hyperbolic spaces and tree metrics, which could lead to some faster algorithms. From a theoretical point of view, it would be interesting to investigate what the lower bound of computing the hyperbolicity constant δ is, or show equivalence between this problem and some other algorithmic problem.

Bibliography/References

[1] S. Gersten, *Essays in group theory*, ser. Mathematical Sciences Research Institute publications. Springer-Verlag, 1987. [Online]. Available: http://books.google.com.sa/books?id=uAOFAAAAIAAJ

[2] R. Duan and S. Pettie, "Fast algorithms for (max,min)-matrix multiplication and bottleneck shortest paths," in *In Proc. 19th SODA*. ACM Press, 2009, pp. 384–391.

[3] F. E. Su, "Topics in geometric combinatorics," 2004.

[4] G. Xing, C. Lu, R. Pless, and Q. Huang, "On greedy geographic routing algorithms in sensing-covered networks," in *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, ser. MobiHoc '04. New York, NY, USA: ACM, 2004, pp. 31–42. [Online]. Available: http://doi.acm.org/10.1145/989459.989465

[5] E. Jonckheere and P. Lohsoonthorn, "Geometry of network security," in *American Control Conference, 2004. Proceedings of the 2004*, vol. 2, 30 2004-july 2 2004, pp. 976 –981 vol.2.

[6] D. Burago, Y. Burago, and S. Ivanov, *A course in metric geometry*. American Mathematical Society, 2001.

[7] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Third Edition*, 3rd ed. The MIT Press, 2009.

[8] V. Chepoi, F. F. Dragan, B. Estellon, M. Habib, and Y. Vaxs, "Diameters, centers, and approximating trees of -hyperbolic geodesic spaces and graphs (extended abstract)," 2008.

[9] V. Chepoi and F. F. Dragan, "A note on distance approximating trees in graphs," *Eur. J. Comb.*, vol. 21, no. 6, pp. 761–766, 2000.

[10] D. C. V. F. J. M. Alonso, T. Brady, "Notes on word hyperbolic groups," 1990.

[11] H. Bandelt, "Recognition of tree metrics," *SIAM Journal on Discrete Mathematics*, vol. 3, no. 1, pp. 1–6, 1990. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/0403001

[12] [Online]. Available: http://www.caida.org/funding/netse/figures/gromov.png

[13] E. A. Jonckheere, P. Lohsoonthorn, and F. Ariaei, "Scaled gromov four-point condition for network graph curvature computation." *Internet Mathematics*, vol. 7, no. 3, pp. 137–177, 2011. [Online]. Available: http://dblp.uni-trier.de/db/journals/im/im7.html#JonckheereLA11

[14] W. Chen, W. Fang, G. Hu, and M. W. Mahoney, "On the hyperbolicity of small-world networks and tree-like graphs," *CoRR*, vol. abs/1201.1717, 2012. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1201.html#abs-1201-1717

[15] V. Chepoi, F. F. Dragan, B. Estellon, M. Habib, Y. Vaxs, and Y. Xiang, "Additive spanners and distance and routing labeling schemes for hyperbolic graphs."

[16] R. Krauthgamer and J. R. Lee, "Algorithms on negatively curved spaces," in *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer*

Science, ser. FOCS '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 119–132. [Online]. Available: http://dx.doi.org/10.1109/FOCS.2006.9

[17] A. Gyárfás and G. Simony, "Edge colorings of complete graphs without tricolored triangles," *J. Graph Theory*, vol. 46, no. 3, pp. 211–216, Jul. 2004. [Online]. Available: http://dx.doi.org/10.1002/jgt.v46:3

[18] Breaking the coppersmith-winograd barrier. [Online]. Available: http://www.citeulike.org/user/caf/author/{VassilevskaWilliams}:V