Information Processing Letters 115 (2015) 576–579

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Computing the Gromov hyperbolicity of a discrete metric space

Hervé Fournier [a], Anas Ismail [b], Antoine Vigneron [b,*]

[a] Univ Paris Diderot, Sorbonne Paris Cité, Institut de Mathématiques de Jussieu, UMR 7586 CNRS, 75205 Paris, France
[b] King Abdullah University of Science and Technology (KAUST), Visual Computing Center, Thuwal 23955-6900, Saudi Arabia

ARTICLE INFO

Article history:
Received 4 September 2013
Received in revised form 12 May 2014
Accepted 8 February 2015
Available online 12 February 2015
Communicated by R. Uehara

Keywords:
Algorithms design and analysis
Approximation algorithms
Discrete metric space
Hyperbolic space
(max,min) matrix product

ABSTRACT

We give exact and approximation algorithms for computing the Gromov hyperbolicity of an n-point discrete metric space. We observe that computing the Gromov hyperbolicity from a fixed base-point reduces to a (max,min) matrix product. Hence, using the (max,min) matrix product algorithm by Duan and Pettie, the fixed base-point hyperbolicity can be determined in $O(n^{2.69})$ time. It follows that the Gromov hyperbolicity can be computed in $O(n^{3.69})$ time, and a 2-approximation can be found in $O(n^{2.69})$ time. We also give a $(2\log_2 n)$-approximation algorithm that runs in $O(n^2)$ time, based on a tree-metric embedding by Gromov. We also show that hyperbolicity at a fixed base-point cannot be computed in $O(n^{2.05})$ time, unless there exists a faster algorithm for (max,min) matrix multiplication than currently known.

1. Introduction

Gromov introduced a notion of metric-space hyperbolicity [2,9] using a simple four point condition. (See Section 1.1.) This definition is very attractive from a computer scientist point of view as the hyperbolicity of a finite metric space can be easily computed by brute force, by simply checking the four point condition at each quadruple of points. However, this approach takes $\Theta(n^4)$ time for an n-point metric space, which makes it impractical for some applications to networking [6]. Knowing the hyperbolicity is important, as the running time and space requirements of previous algorithms designed for Gromov hyperbolic spaces are often analyzed in terms of their Gromov hyperbolicity [4,5,10]. So far, it seems that no better algorithm than brute force was known for computing the Gromov hyperbolicity [3]. In this note, we give faster exact and approximation algorithms based on previous work on (max-min) matrix products by Duan and Pettie [7], and the tree-metric embedding by Gromov [9].

The exponent of matrix multiplication μ is the infimum of the real numbers $\omega > 0$ such that two $n \times n$ real matrices can be multiplied in $O(n^{\omega})$ time, exact arithmetic operations being performed in one step [11]. Currently, μ is known to be less than 2.373 [12]. In the following, ω is a real number such that we can multiply two $n \times n$ real matrices in $O(n^{\omega})$ time.

Our algorithm for computing the Gromov hyperbolicity runs in $O(n^{(5+\omega)/2})$ time, which is $O(n^{3.69})$. (See Section 2.1.) For a fixed base-point, this improves to $O(n^{(3+\omega)/2})$, which also yields a 2-factor approximation for the general case within the same time bound. (See Section 2.2.) We also give a quadratic-time $(2\log_2 n)$-approximation algorithm. (See Section 2.3.) Finally, we show that hyperbolicity at a fixed base-point cannot be computed in time $O(n^{3(\omega-1)/2}) = O(n^{2.05})$, unless (max,min) matrix product can be computed in time $O(n^{\tau})$ for $\tau < (3+\omega)/2$. (See Section 3.) The currently best known algorithm runs in $O(n^{(3+\omega)/2})$ time [7].

* Corresponding author.
E-mail addresses: fournier@math.univ-paris-diderot.fr (H. Fournier), anas.ismail@kaust.edu.sa (A. Ismail), antoine.vigneron@kaust.edu.sa (A. Vigneron).

http://dx.doi.org/10.1016/j.ipl.2015.02.002
0020-0190/

1.1. Gromov hyperbolic spaces

An introduction to Gromov hyperbolic spaces can be found in the article by Bonk and Schramm [2], and in the book by Ghys and de la Harpe [8]. Here we briefly present some definitions and facts that will be needed in this note.

A metric space (M, d) is said to be δ-hyperbolic for some $\delta \geqslant 0$ if it obeys the so-called *four point condition*: For any $x, y, z, t \in M$, the largest two distance sums among $d(x, y) + d(z, t)$, $d(x, z) + d(y, t)$, and $d(x, t) + d(y, z)$, differ by at most 2δ. The *Gromov hyperbolicity* δ^* of (M, d) is the smallest $\delta^* \geqslant 0$ such that (M, d) is δ^*-hyperbolic.

For any $x, y, r \in M$, the *Gromov product* of x, y at r is defined as

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

The point r is called the *base point*. Gromov hyperbolicity can also be defined in terms of the Gromov product, instead of the four point condition above. The two definitions are equivalent, with the same values of δ and δ^*. So a metric space (M, d) is δ-hyperbolic if and only if, for any $x, y, z, r \in M$

$$(x|z)_r \geqslant \min\{(x|y)_r, (y|z)_r\} - \delta.$$

The Gromov hyperbolicity δ^* is the smallest value of δ that satisfies the above property. In other words,

$$\delta^* = \max_{x,y,z,r} \{\min\{(x|y)_r, (y|z)_r\} - (x|z)_r\}.$$

The hyperbolicity δ_r at base point r is defined as

$$\delta_r = \max_{x,y,z} \{\min\{(x|y)_r, (y|z)_r\} - (x|z)_r\}. \tag{1}$$

Hence, we have

$$\delta^* = \max_r \delta_r. \tag{2}$$

2. Algorithms

In this section, we consider a discrete metric space (M, d) with n elements, that we denote $x_1, \ldots, x_n$. Our goal is to compute exactly, or approximately, its hyperbolicity δ^*, or its hyperbolicity δ_r at a base point r.

2.1. Exact algorithms

The (max,min)-product $A \otimes B$ of two real matrices A, B is defined as follows:

$$(A \otimes B)_{ij} = \max_k \min\{A_{ik}, B_{kj}\}.$$

Duan and Pettie [7] gave an $O(n^{(3+\omega)/2})$-time algorithm for computing the (max,min)-product of two $n \times n$ matrices.

Let r be a fixed base-point. By Eq. (1), if A is the matrix defined by $A_{ij} = (x_i|x_j)_r$ for any i, j, then δ_r is simply the largest coefficient in $(A \otimes A) - A$. So we can compute δ_r in $O(n^{(3+\omega)/2})$ time. Maximizing over all values of r, we can compute the hyperbolicity δ^* in $O(n^{(5+\omega)/2})$ time, by Eq. (2).

2.2. Factor-2 approximation

The hyperbolicity δ_r with respect to any base-point is known to be a 2-approximation of the hyperbolicity δ^* [2]. More precisely, we have $\delta_r \leqslant \delta^* \leqslant 2\delta_r$. So, using the algorithm of Section 2.1, we can pick an arbitrary base-point r and compute δ_r in $O(n^{(3+\omega)/2})$ time, which gives us a 2-approximation of δ^*.

2.3. Logarithmic factor approximation

Gromov [9] (see also the article by Chepoi et al. [4, Theorem 1] and the book by Ghys and de la Harpe [8, Chapter 2]) showed that any δ-hyperbolic metric space (M, d) can be embedded into a weighted tree T with an additive error $2\delta \log_2 n$, and this tree can be constructed in time $O(n^2)$. In particular, if we denote by d_T the metric corresponding to such a tree T, then

$$d(a, b) - 2\delta^* \log_2 n \leqslant d_T(a, b) \leqslant d(a, b) \text{ for any } a, b \in M. \tag{3}$$

This construction can be performed without prior knowledge of δ^*.

We compute $D = \max_{a,b \in M} d(a, b) - d_T(a, b)$ in time $O(n^2)$. We claim that:

$$\delta^* \leqslant D \leqslant 2\delta^* \log_2 n. \tag{4}$$

So we obtain a $(2\log_2 n)$-approximation D of δ^* in time $O(n^2)$.

We still need to prove the double inequality (4). It follows from Eq. (3) that $d(a, b) - d_T(a, b) \leqslant 2\delta^* \log_2 n$ for any a, b, and thus $D \leqslant 2\delta^* \log_2 n$. In the following, we prove the other inequality.

For any x, y, z, t, we denote by $\delta(x, y, z, t)$ the difference between the two largest distance sums among $d(x, y) + d(z, t)$, $d(x, z) + d(y, t)$, and $d(x, t) + d(y, z)$. Thus, if for instance $d(x, y) + d(z, t) \geqslant d(x, z) + d(y, t) \geqslant d(x, t) + d(y, z)$, we have $\delta(x, y, z, t) = d(x, y) + d(z, t) - d(x, z) - d(y, t)$. We also need to introduce the difference $\delta_T(x, y, z, t)$ between the two largest sums among $d_T(x, y) + d_T(z, t)$, $d_T(x, z) + d_T(y, t)$, and $d_T(x, t) + d_T(y, z)$.

For any $a, b \in M$, we have $d(a, b) - D \leqslant d_T(a, b) \leqslant d(a, b)$, so $\delta(x, y, z, t) - \delta_T(x, y, z, t) \leqslant 2D$, because in the worst case, the largest sum with respect to d is the same as the largest sum with respect to d_T, and the second largest sum with respect to d_T is equal to the second largest sum with respect to d minus $2D$. But by construction, d_T is a tree metric [4], so $\delta_T(x, y, z, t) = 0$ for any x, y, z, t. Therefore $\delta(x, y, z, t) \leqslant 2D$ for any x, y, z, t, which means that $\delta^* \leqslant D$.

3. Conditional lower bounds

We show that computing hyperbolicity at a fixed base-point is intimately connected with (max,min)-product. From the previous section, any improvement on the complexity of (max,min)-product yields an improvement on our algorithm to compute hyperbolicity. We show that a partial converse holds: Any improvement on the complexity of computing hyperbolicity at a fixed base-point below