Global curvature of data

1 Status and ToDo

ToDo:

- Compare different curvatures on synthetic and real graphs (Liuda and Egor)
- Summarize which theory we have and which we can obtain (Liuda)
- Start writing the paper in ICLR format (Liuda)

Ideas for research:

- Gromov's hyperbolicity cannot be easily converted to curvature. We can think about some similarly motivated measure which can be. Using either the distance from one edge of a triangle to the other edges or using the perimeter of the orthic triangle.
- Think about optimal curvature of ER graphs. We have the result on Ollivier curvature, but it is supposed to be different (for example, Ollivier is bounded between -2 and 1)

2 Selected relevant papers

Learning Mixed-Curvature Representations in Product Spaces [7] Embed graphs to a product manifold combining multiple copies of spherical, hyperbolic, and Euclidean spaces. The code is available at `https://github.com/HazyResearch/hyperbolics`. This is the main paper for the current project: we use their code for embedding, we also use two their approaches as our baselines: gradient descent and parallelogram.

Representation Tradeoffs for Hyperbolic Embeddings [19] Interesting observation that not only dimensionality, but also precision is important. Propose a new algorithm for graph embedding: first, embed a graph into a weighted tree, and then embed that tree into the hyperbolic disk. The code is available at `https://github.com/HazyResearch/hyperbolics`. One of their observations, which is quite natural, is that the dependence of the required precision on deg_{max} is logarithmic, but on the maximum path length is linear.

When they embed a tree to a hyperbolic disk (ball) they use Sarkar's algorithm for a 2-dimensional disk [21]. This algorithm is simple: take the next node, consider its children, distribute them uniformly on a circle around the node, as far as possible from the parent of this node. The authors also extend Sarkar's algorithm to higher dimensions, where they use spherical coding to space out points on a hypersphere.

An important ingredient of Sarkar's algorithm is scaling τ: we assume that the length of an edge is not 1, but τ. Larger τ allows to reduce distortion (increasing τ basically means increasing absolute value of curvature), but larger τ requires larger precision also, so there is a trade-off.

A theoretical result about embedding trees on hyperbolic disk is the following: if $\tau = \frac{1+\varepsilon}{\varepsilon} \cdot 2 \log \frac{deg_{max}}{\pi/2}$, then the worst-case distortion is no more than $1+\varepsilon$.

Their embedding of graphs into weighted trees is based on the paper [1] which I discuss next.

Reconstructing Approximate Tree Metrics [1] Note: this paper discusses distortion, which is actually worst-case distortion D_{WC}.

In another paper it is shown that a metric space is a tree metric if and only if the sub-metric induced by any four points is a tree metric, which is equivalent to the following Four-Points Condition (4PC): for any four points, out of the three possible matchings, the two matchings that have the maximum weight have equal weight.

Also, they refer to Gromov's δ-hyperbolicity, which is the following. A metric space δ-hyperbolic if for every four points w, x, y, z that are ordered such that $d(w,x)+d(y,z) \leq d(w,y)+d(x,z) \leq d(w,z)+d(x,y)$, we have $d(w,z)+d(x,y) \leq d(w,y)+d(x,z)+\delta$. The δ-hyperbolicity condition is equivalent to assuming that every 4 points embed into a tree metric with additive distortion $O(\delta)$.

In the current paper, the authors propose $\varepsilon - 4PC$, where $d(w,z)+d(x,y) \leq d(w,y)+d(x,z)+2\varepsilon \cdot \min d(w,x), d(y,z)$. $\varepsilon = 0$ gives 4PC. For $\varepsilon = 1$ the property is always satisfied due to the triangle inequality. The authors argue that from the practical point of view this property is more useful.

They show that of ε-4PC, then there is a constant c_1 such that we can embed into a tree metric with a maximum distortion of $(1+\varepsilon)^{c_1 \cdot \log n}$. And also there is a lower bound (for some metric space) with another constant c_2.

They give an algorithm for embedding. The algorithm is based on adding Steiner nodes recursively. Note that the distances among every set of three points x, y, z of a metric space can be exactly represented by a tree by adding one additional Steiner node. The Gromov product $(x|y)_z = \frac{1}{2} \cdot (d(x,z)+d(y,z)-d(x,y))$ is the distance from such Steiner node to z.

Poincaré GloVe: Hyperbolic Word Embeddings [23] This paper discusses and computes average δ-hyperbolicity, which is discussed in Appendix B. Using the notation of the paragraph above, they write that for four point $\delta = (d(w,z)+d(x,y)-d(w,y)-d(x,z))/2$. Then, the supremum of these numbers over all 4-tuples is δ_{worst}. They (reasonably) argue that the average δ_{avg} is more meaningful than δ_{worst}. If we can define triangles, then an equivalent definition is: δ_{worst} is the smallest $\delta > 0$ such that for any triangle there exists a point at distance at most δ from each side of the triangle. Note that Euclidean space is ∞-hyperbolic. They also report the normalized value $2\delta_{avg}/d_{avg}$ which is invariant to metric scaling. Note that low values of both δ_{avg} and d_{avg} are indicators of some hyperbolicity.

Ricci Curvature of the Internet Topology [14] Empirical paper: analyze Ricci curvature of some graphs. Also, analyze computation of Ricci curvature. Show that in networks, the distribution of Ricci curvature is highly non-homogeneous. And there are many negatively curved edges, but also communities of positive curved edges. The curvature distribution for different type of Internet topology can be very different. As expected, removing edges of negative curvatures disconnect network fast.

The correlation between Ricci curvature and other properties is shown on Figures 9-13. They consider betweenness centrality, farness centrality, degree and clustering coefficient.

Also, this paper has a nice intuitive introduction into Gromov hyperbolicity and sectional and Ricci curvature. Speaking about Gromov, this paper has yet another definition of this curvature: metric has δ-hyperbolicity if all geodesic triangles are δ-slim, that is, for any three points a, b, c, the geodesic paths (shortest paths) between them satisfy the following property: any point on one path is within distance δ from the closest vertex on the other two paths.s Further, we can compare $\delta(\Delta abc)$ with the diameter of the triangle Δabc. The ratio is bounded above by $3/2$ in a Riemannian manifold of constant nonpositive curvature, thus, in other papers, it was suggested to use $3/2$ as the threshold to distinguish whether the graph metric is hyperbolic or not.

Upper bound on scaled Gromov-hyperbolic δ [8] This paper defines δ in yet another way and discusses the bound $2/3$ mentioned above. Namely, now $\delta(\Delta ABC)$ is the smallest value of $d(X,Y) + d(Y,Z) + d(Z,X)$, where X, Y, Z are some points on the geodesics forming this triangle. For a tree we have $\delta = 0$. In Eucledian space the points X, Y, Z form orthic triangle.

What relations are reliably embeddable in Euclidean space? [2] About embeddings of directed graphs to Euclidean space. Consider several types of embeddings. In translational embeddings, two nodes are connected by a directed edge if adding a given vector to one of them approximately gives the other. Prove both positive and negative results for this type: e.g., layered graph can be embedded, while cycles cannot. In distance embeddings, each node has both in- and out-embeddings, then threshold model is applied. Prove that any graph has distance embedding. Also, focus on precision of embeddings: define δ-robustness which is directly related to the precision. In similarity embeddings, the dot product has to be larger than a threshold.

Althouhg directed graphs are not so interesting for us now, the paper has a nice related work section with results on undirected graph and I will list some results below.

Let us consider threshold Euclidean embedding, where we connect all nodes at distance less than 1. [13] proves that for any graph $G(V, E)$ there exists such embedding with dimension $d \leq |V|$. The smallest such d is called sphericity and computing it is NP-hard. Also, [10] shows that embeddings achieving this minimum dimension sometimes require precision exponential in $|V|$.

Let us consider threshold similarity embedding, which is based on dot product rather than Euclidean distance. In this case, as shown in [18], threshold similarity embedding exists with $d \leq |V|$. And the minimal dimension d is at most sphericity of G. For example, a complete binary tree on n nodes has sphericity $\Omega(logn)$ but can be embedded in R^3 using a dot-product embedding.

3 How to estimate global curvature

3.1 Number of k-neightbors

Let us take a node v. Let $N_k(v)$ denote the number of nodes at distance exactly k from v. Assuming that nodes are roughly uniformly distributed, we can compare $N_k(v)$ with a surface area of a ball of radius k.

In euclidean space, surface area of a ball grows as k^{d-1} (up to some multiplicative function of d).

Let us consider spherical space. Here, if I am not mistaken, up to some multiplier depending on d, we have $(R \sin \theta)^{d-1}$. Curvature is $c = 1/R^2$, radius is $k = R \cdot \theta$, so surface area is $\left(\frac{1}{\sqrt{c}} \sin \sqrt{c}\, k\right)^{d-1}$.

Let us consider hyperbolic space. Here, assuming the the curvature is $-c$, we have $\left(\frac{1}{\sqrt{c}} \sinh \sqrt{c}\, k\right)^{d-1}$.

So, the algorithm can be the following (recall that we are given dimentsion d): consider $N_k^{\frac{1}{d-1}}$ for some small k, check if the obtained function is convex or concave or straight line. Then, fit the corresponding function (sin, sinh), obtain curvature.

We currently implement this idea as the easiest one.

3.2 Modifications of the above idea

Assumption about uniform distribution is rather strong. In particular, we could take into account the edges between the vertices of N_k. Here I see two options.

The first one is the following: let us take nodes at distance k and consider the maximum independent subset in this set (assume that its size is N'_k). Then, we know that for each vertex there are no other vertices in the ball of radius 0.5, which gives us the lower bound for the required volume: $N'_k/2$. So, we can use the algorithm discussed above, but with N'_k instead of N_k.

The second one is similar to the previous, but maybe considering the independent set is too restrictive. So, we may look at the number of non-edges between the vertices on distance k. If this number if L, then we can say that the volume grows as N''_k with $N''_k(N''_k - 1)/2 = L$ (no good theory behind that).

3.3 Intersecting balls

Consider our graph G. Let us take two balls (considering graph distance) at distance l and with radius k. I do not have formulas now, but in euclidean/hyperbolic/spherical spaces we can compute the volume of the intersection of such balls relative to the volume of their union. This will be the function of dimension and curvature, dimension is given, so we can use this to estimate the curvature.

I see two weaknesses of this approach: 1) formulas seems to be more complicated, even for euclidean space, 2) it is not clear which radius and which distance to consider, my first idea was to take distance 1 and radius 1 (i.e., analyze the fraction of common neighbors), but this fails on, e.g., integer lattice in euclidean space.

4 Measuring quality of embeddings

Mean Average Precision (MAP) Let us consider the node v. $N_i(v)$ be the number of nodes closest to i we have to consider to cover i neighbors of i. Note that in the ideal situation we have $N_i(v) = i$. Then MAP is defined as

$$\frac{1}{n}\sum_{v}\frac{1}{deg(v)}\sum_{i=1}^{deg(v)}\frac{i}{N_i(v)}.$$

Distortion This is a standard metric for graph embeddings.

$$D(f) = \frac{1}{\binom{n}{2}}\sum_{u\neq v}\frac{|d(f(u),f(v)) - d_G(u,v)|}{d_G(u,v)}.$$

There is also a variant called worst-case distortion:

$$D_{WC}(f) = \frac{\max_{u\neq v} d(f(u),f(v))/d_G(u,v)}{\min_{u\neq v} d(f(u),f(v))/d_G(u,v)}$$

Threshold-based loss Note that MAP is scale-invariant, while distortion is not scale-invariant. Here is a more reasonable non-scale-invariant measure. Let G be the original graph and let G' be the graph recovered from embedding with threshold 1. Let n_{11} the number of pairs of vertices, which are connected in both G and G', n_{00} — which are not connected in both cases, n_{01} and n_{10} — connected in one graph and not connected in another. Then there are several ways to measure similarity using these numbers (Rand, Jaccard, etc.), among them unbiased ones are correlation coefficient and Sokal and Sneath index. Let's use correlation coefficient since it is more well-known:

$$\frac{n_{00}n_{11} - n_{01}n_{10}}{\sqrt{(n_{00} + n_{01})(n_{10} + n_{11})(n_{00} + n_{10})(n_{01} + n_{11})}}$$

.

5 Curvature in graphs

There are many different notions of graphs curvature. Here we describe a few of them, starting with the most prominent one Ollivier-Ricci curvature. We will use these curvatures as our baselines.

5.1 Ollivier-Ricci curvature

To define this curvature we first need to introduce the Wasserstein metric.

Let $(\mathcal{X}, d)$ be a metric space (which technically should be Radon or Polish) and consider two probability measures μ_1 and μ_2 on this space. Recall that a coupling between μ_1 and μ_2 is a joint probability measure μ whose marginals are μ_1 and μ_2. Let $\Gamma(\mu_1, \mu_2)$ denote the set of all such couplings. The Wasserstein metric (Kantorovich-Rubenstein distance of order one) between μ_1 and μ_2 is then given by

$$W_1(\mu_1, \mu_2) = \inf_{\mu \in \Gamma(\mu_1, \mu_2)} \int_{\mathcal{X} \times \mathcal{X}} d(x, y)\, d\mu(x, y) \tag{1}$$

Due to a duality theorem by Kantorovich and Rubenstein we have the following equivalent definition of $W_1(\mu_1, \mu_2)$

$$W_1(\mu_1, \mu_2) = \sup_{f \in L_1} \int_{\mathcal{X}} f(x)\, d\mu_1(x) - \int_{\mathcal{X}} f(y)\, d\mu_2(y), \tag{2}$$

where the supremum is taken over all Lipschitz continuous function on (X, d) with Lipschitz constant 1, i.e.

$$|f(x) - f(y)| \leq d(x, y).$$

For a graph G we consider the shortest path metric on G, denoted by d_G, and let W_1^G denote the Wasserstein metric with respect to the metric space (G, d_G). Furthermore, for each node v let m_v denote the uniform probability measure on the neighbors of v, i.e.

$$m_v(u) = \frac{1_{u \sim v}}{\deg(v)},$$

where $\deg(v)$ denotes the degree of v. Then the classic definition of Ollivier-Ricci curvature between two neighboring nodes $v \sim u$ in G is defined as

$$\kappa_G(u, v) = 1 - W_1^G(m_v, m_u). \tag{3}$$

It is important to note that Ollivier-Ricci curvature is defined in much more generality in terms of metrics and random walks, see [17]. Thus different version on graphs can be considered. Definition (3) corresponds to the classical choices of graph distance and random walk on the graph.

The strength of Ollivier-Ricci curvature lies in the fact that when we consider its continuous version on Riemannian manifolds than this converges to the Ricci curvature, see [17, Example 7]. It turns out that this also holds when we consider dense random geometric graphs on Riemannian manifolds (upcoming paper). However, for this one needs to deviate from the classical version and consider random-walks on larger neighborhoods. Still these results highlights that Ollivier-Ricci curvature can properly encode the curvature of the underlying manifold.

Below we list some simple examples of Ollivier-Ricci curvature in graphs.

Circle Consider a circle on n nodes and let $v \sim u$ be two neighbors. Then $W_1^G(m_u, m_v) = 1$ and hence $\kappa_G(u,v) = 0$.

Trees Consider any tree graph T, let $v \sim u$ be two neighbors. Then Proposition 2 in [9] states that

$$\kappa_G(m_x, m_y) = -2\left(1 - \frac{1}{\deg(v)} - \frac{1}{\deg(u)}\right)_+,$$

where $(t)_+ = \max\{0, t\}$. In particular, if either $\deg(v) = 1$ or $\deg(u) = 1$ then it follows that $\kappa_G(u,v) = 0$. However, if $\deg(v) \geq 3$ for all nodes v then $\kappa_G(v,u) < 0$. Thus trees are intrinsic examples of negatively curved graphs, according to Ollivier-Ricci curvature.

Lattice Consider the 2-dimensional lattice $\mathbb{Z}^2$ and let $v \sim u$ be two neighbors. Then the optimal transportation from m_u to m_v is given by a translation by the vector $y - x$ and hence $W_1^G(m_v, m_u) = d_G(u,v) = 1$, see also [17, Example 5], and thus $\kappa_G(u,v) = 0$.

Complete graph Consider a complete graph on n nodes. Then, for any two nodes u and v it follows from Example 1 in [9] that $\kappa_G(m_v, m_u) = \frac{n-2}{n-1} \to 1$ as $n \to \infty$.

Erdős-Renyi graph The following results from [12] characterize Ollivier-Ricci curvature in $G(n,p)$:

1. If $p \geq \log(n)^{1/3} n^{-1/3}$ then almost surely

$$\kappa_G(u,v) = p + O\left(\sqrt{\frac{\log(n)}{np}}\right).$$

2. If $2\log(n)^{1/2}n^{-1/2} \leq p < \log(n)^{1/3}n^{-1/3}$ then almost surely

$$\kappa_G(u,v) = O\left(\frac{\log(n)}{np^2}\right).$$

3. If $\log(n)^{1/2}n^{-2/3} \ll p \ll \log(n)^{-1/2}$ then almost surely

$$\kappa_G(u,v) = -1 + O\left(np^2\right) + O\left(\frac{\log(n)}{n^2p^3}\right).$$

4. If $\log(n)n^{-1} \ll p \ll n^{-2/3}$ then almost surely

$$\kappa_G(u,v) = -2 + O\left(n^2p^3\right) + O\left(\sqrt{\frac{\log(n)}{np}}\right)$$

5.2 Forman-Ricci curvature

Another notion of graph curvature called Forman-Ricci curvature [22] is based on the discretization of Ricci curvature proposed by Forman [4]. It is defined for general weighted graph G, with both vertex weights $w(v)$ and edge weights $w(u,v)$, as follows

$$F(u,v) = w(u,v)\left(\frac{w(u)+w(v)}{w(u,v)} - \sum_{u' \in N(u)\setminus v} \frac{w(u)}{w(u,u')} - \sum_{v' \in N(v)\setminus u} \frac{w(v)}{w(v,v')}\right), \quad (4)$$

where $N(v)$ denotes the neighborhood of a node v. A definition of Forman-Ricci curvature for vertices is then obtained by averaging over all neighbors

$$F(v) = \frac{1}{\deg(v)} \sum_{u \in N(v)} F(u,v). \quad (5)$$

When the graph G is not weighted, i.e. when $w(v) = 1 = w(v,u)$ for all nodes v and edges (u,v), we get

$$\begin{aligned} F(u,v) &= (2 - (\deg(v) - 1) - (\deg(u) - 1)) \\ &= 4 - (\deg(v) + \deg(u)). \end{aligned}$$

In particular, Forman-Ricci curvature of a graph is zero if and only if the graph is a circle.

The general formula by Forman included the faces of any dimension. This can of course be done for graphs as well, see [24]. In (4) only 1-dimensional faces (edges) are included. However, one can also include 2-dimensional faces. This was considered in [20], where only the 2-dimensional faces on three nodes (triangles) were considered. In the case of an unweighted graph we then obtain

$$F_2(u,v) = F(u,v) + 3\Delta(u,v), \quad (6)$$

where $\Delta(u,v)$ is the number of triangles that contain the edge (u,v). This is the same as the number of common neighbors of u and v.

5.3 Approximation of Ollivier curvature

An explicit formula for Ollivier-Ricci curvature in graphs is obtained in [11], under somewhat restrictive assumptions. In general it could be used as an approximation.

For an edge (u, v), let $\triangle_{uv}$, $\square_{uv}$ and $\pentagon_{uv}$ denote, respectively, the number of triangles, squares and pentagons in the graph that include the edge (u, v). Moreover let $a \wedge b = \min\{a, b\}$, $a \vee b = \max\{a, b\}$ and $[a]_+ = a \vee 0$. Then we define:

$$\begin{aligned} \kappa^*_G(u, v) = {} & \frac{\triangle_{uv}}{\deg(u) \wedge \deg(v)} - \left[1 - \frac{1 + \triangle_{uv} + \square_{uv}}{\deg(u) \vee \deg(v)} - \frac{1}{\deg(u) \wedge \deg(v)}\right]_+ \\ & - \left(\frac{\triangle_{uv}}{\deg(u) \vee \deg(v)} - \frac{\triangle_{uv}}{\deg(u) \wedge \deg(v)}\right) \\ & \vee \left(1 - \frac{1 + \triangle_{uv} + \square_{uv} + \pentagon_{uv}}{\deg(u) \vee \deg(v)} - \frac{1}{\deg(u) \wedge \deg(v)}\right). \end{aligned} \tag{7}$$

In [11] it was shown that $\kappa^*_G(u, v) = \kappa_G(u, v)$ if and only if short cycles of length 3,4 or 5 that include the edge (u, v) share no other edges. For example, if w is a neighbor of both u and v (so w is part of a 3-cycle) then w cannot be connected to any other neighbor of either u or v, since that would mean that w is part of a 4-cycle that includes the edge (u, v). Although this requirement is quite restrictive, one could still use (7) as an approximation (or even a definition) of curvature in graphs.

[By adjusting the definitions used in [11] it might be possible to derive an upper bound for Ollivier curvature in terms of disjoint triangles, squares and pentagrams. It will look very similar to (7) **and will be exact when the independent short cycle assumption holds.]**

5.4 Heuristic sectional curvature from [7]

In [7] a different notion of curvature is used, based on the parallelogram law. This is one of our main baselines, so we should compare with this theoretically and empirically. For a parallelogram with nodes a, b, c, d and sides ab, ac, bd, cd, the parallelogram law states that:

$$2d(c, d)^2 + 2d(b, d)^2 = d(b, c)^2 + d(a, d)^2.$$

Now imagine a point m in the center of the line connecting a and d, i.e. $d(a, m) = d(a, d)/2 = d(m, d)$. Then we have that

$$\begin{aligned} 4d(a, m)^2 = d(a, d)^2 &= 2d(c, d)^2 + 2d(b, d)^2 - d(b, c)^2 \\ &= 2d(a, b)^2 + 2d(a, c)^2 - d(b, c)^2, \end{aligned}$$

where the last line is because $d(a, c) = d(b, d)$ and $d(a, b) = d(c, d)$. This then implies that

$$d(a, m)^2 + \frac{d(b, c)^2}{4} - \frac{d(a, b)^2 + d(a, c)^2}{2} = 0. \tag{8}$$

The key intuition behind the definition in [7] is that this holds in flat Euclidean space while in a positively or negative curved space the left hand side in (8) should be positive or negative, respectively.

For graphs, Let v be a node in G, u, w neighbors of v and z any other node. Then we define

$$\xi_G(v, u, w, z) = d(v, z)^2 + \frac{d(u, w)^2}{4} - \frac{d(v, u)^2 + d(v, w)^2}{2}. \tag{9}$$

Note that this resembles the left hand side in (8) with $a = v$, $b = u$, $c = w$ and $m = z$. Again, based on the parallelogram law, the intuition is that if these four vertices are supposed to be embedded in negative, flat or positive curved space then $\xi_G(v, u, w, z)$ should be negative, zero or positive, respectively. Now consider any node v and two of its neighbors u and w. Then we define the graph sectional curvature of a node v and is neighbors u, w as the average of $\xi_G(v, u, w, z)$,

$$\xi_G(v; u, w) = \frac{1}{|G| - 1} \sum_{z \in G \setminus v} \frac{\xi_G(v, u, w, z)}{2d_G(v, z)}.$$

Here the normalization constant $2d_G(v, z)$ is include to yield the right scalings for trees and cycles.

The curvature of the graph is then defined as the average of $\xi_G(v; u, w)$. This is computed as $\xi_G = \frac{\xi_G(v^*, u^*, w^*, z^*)}{2d_G(v^*, z^*)}$, where v^*, u^*, w^* and z^* are selected uniformly at random from G, $N(v^*)$, $N(v^*) \setminus u^*$ and $G \setminus v^*$, respectively. Here $N(v)$ denotes the neighborhood of v.

6 Global graph curvature

6.1 Possible definition

Assume that we are given a loss function $L(e(G), G)$ which compares graph G and its embedding. For example, L can be distortion.

Assume that we have some ideal network embedder $e_{d,C}(G)$ which embeds G into a d-dimensional space of curvature C and minimizes L. Then we can define a d-dimensional curvature of G in the following way:

$$C_d(G) = \text{argmin}_C L(e_{d,C}(G), G)\,.$$

Note that $C_d(G)$ can be a set of points. Below we analyze some some simple graphs and consider L to be either distortion or threshold-based loss.

Important note: for now we ignore precision problem. It is a problem, but otherwise it will be much harder to get theoretical results.

6.2 Star on n vertices

Ollivier-Ricci curvature $\kappa_G = 0$

Forman-Ricci curvature $F = 3 - n = F_2$

Ollivier approximation $\kappa^*_G = 0$

Distortion-based curvature For any d, we have $C_d(G) = -\infty$.

Proof. Let us first analyze lower bound for distortion for given curvature.

Distortion of a graph is average distortion over all pairs of nodes. Let v be the central node and v_1, v_2, v_3 be any other three nodes. Then, it is sufficient to find the lower bound for the following function:

$$D(G) \geq \frac{\binom{n}{3}}{\binom{n}{2}} \left(\sum_{1\leq i<j\leq 3} \frac{|d(f(v_i), f(v_j)) - 2|}{2(n-3)} + \sum_{1\leq i\leq 3} \frac{|d(f(v_i), f(v)) - 1|}{\binom{n-2}{2}} \right).$$

So, basically, we need to esitmate a "weighted" distortion of such four nodes. One idea can be the following: if we consider three angles at the node v, then at least one of them is at most $2\pi/3$, so we can get a lower bound only for this triangle. So, we need a lower about for

$$D' = \frac{|d(f(v_1), f(v_2)) - 2|}{2(n-3)} + \frac{|d(f(v_1), f(v)) - 1| + |d(f(v_2), f(v)) - 1|}{\binom{n-2}{2}}.$$

[**This seems to be a doable technical task: we want to embed a triangle with one angle $2\pi/3$ into a space of a fixed curvature to minimize such weighted distortion. I will return to this later.**]

Let us analyze upper bound for distortion now. To do this, we want to find some "good" embedding. Let us consider hyperbolic space with curvature c and with $d = 2$. Let v be the central node, then we spread all other points on a circle of some radius $1 + \varepsilon$. The angle between them is $2\pi/(n-1)$. So, our loss consists of two terms, the first one is

$$\frac{(n-1)\varepsilon}{\binom{n}{2}} = \frac{2\varepsilon}{n}.$$

Let $c = -\frac{1}{k^2}$. The second term is (assuming that n is even, but odd is similar)

$$\frac{(n-1)}{\binom{n}{2}} \sum_{i=1}^{(n-2)/2} \left(2 - k \cdot \operatorname{arccosh}\left(\sinh^2\left(\frac{1+\varepsilon}{k} \left(1 - \cos\frac{2\pi l}{(n-1)} \right) \right) + 1 \right) \right).$$

This can be further optimized, but let us find some rough bound on that. Take $\varepsilon = 0$, then ther first term is zero. Also all angles replace by the smallest ones (the assumption on odd or even is not longer needed). Then the second term is:

$$\frac{\binom{n-1}{2}}{\binom{n}{2}} \left(2 - k \cdot \operatorname{arccosh}\left(\sinh^2\left(\frac{1}{k} \left(1 - \cos\frac{2\pi}{(n-1)} \right) \right) + 1 \right) \right).$$

Now note that

$$\cdot \operatorname{arccosh}\left(\left(1-\cos\frac{2\pi}{(n-1)}\right)\sinh^2\left(\frac{1}{k}\right)+1\right).$$

This term tends to zero (for fixed n) as $k \to 0$ ($c \to \infty$).

[Optional ToDo here: 1) show the better bound 2) relation between n and c (which c we need to get distortion less than ε)]

□

Threshold-based curvature We have $C_d(G) = (-\infty, C)$, where we can compute C exactly for $d = 2$ and approximately for larger d.

Note: for now I assume b to be large enough to be in hyperbolic space only. But it is easy to see that the general statement about $(-\infty, C)$ is true always.

Proof. If $d = 2$, we want to have $n-1$ points to be at distance at most 1 from v, but at distance > 1 from each other. It is easy to show that it is optimal to place the leaf nodes at distance exactly 1. Now C is just the largest curvature which allows to have distance exactly 1 between closest leaves. Let us find C. We use hyperbolic low of cosines again and let $c = -\frac{1}{k^2}$, $\alpha = \frac{2\pi}{n-1}$:

$$\cosh\frac{1}{2k} = \frac{1}{2\sin\frac{\alpha}{2}}$$

$$k = \frac{1}{2\operatorname{arccosh}\frac{1}{2\sin\frac{\alpha}{2}}}$$

$$C = -\left(2\operatorname{arccosh}\frac{1}{2\sin\frac{\alpha}{2}}\right)^2$$

Note that for $n = 6$ we have $C = 0$.

Also note that if n is large, then $\sin\frac{\alpha}{2} = \sin\frac{\pi}{n-1} \sim \frac{\pi}{n-1}$. Then, $\operatorname{arccosh}\frac{1}{2\sin\frac{\alpha}{2}} \sim \operatorname{arccosh}\frac{n-1}{2\pi} \sim \log n$, so we get $C \sim -4\log^2 n$.

For $d > 2$ we can get lower and upper bounds for C. One bound we can obtain by covering the surface of a unit ball by the balls of radius 1/2. The other bound can be obtained by using spherical packing. **[To be finished formally.]**

□

6.3 Cycle of size n

Ollivier-Ricci curvature $\kappa_G(u, v) = 0$ for all edges, so by averaging we obtain 0.

Forman-Ricci curvature $F(u, v) = 0 = F_2(u, v)$.

Ollivier approximation $\kappa^*_G = 0$.

Distortion-based curvature For any dimension d and for $n \geq 4$, there is a unique curvature which gives zero distortion:

$$C_d(G) = \left(\frac{2\pi}{n}\right)^2.$$

This is easy to explain: if we consider three consequent vertices, then the middle one should lie on the geodetic between the other two. So, they all lie on a great circle (of length n).

Threshold-based curvature Here, again, we have $C_d(G) = (-\infty, C)$ with some $C > 0$. It is easy to prove that any curvature $c < \left(\frac{4\pi}{n}\right)^2$ is ok, but I have to think whether we can get some larger bound.

6.4 Complete graph on n vertices

Ollivier-Ricci curvature $\kappa_G(u,v) = \frac{n-2}{n-1}$, so average is the same.

Forman-Ricci curvature $F(u,v) = 6 - 2n$, $F_2(u,v) = n$.

Ollivier approximation $\kappa_G^* = ...$ **[Easy, but did not check yet. We can see the difference with Ollivier with this example.]**

Distortion-based curvature Assume that $d = n-2$, then it is easy to find $C_d(G)$. Note that we want to get $n-1$-simplex, which can be embedded into a $n-2$-dimensional spherical space, the radius is $R = a\sqrt{\frac{n-1}{2n}}$, and we want

$$2R \arcsin \frac{a}{2R} = 1.$$

$$2a\sqrt{\frac{n-1}{2n}} \arcsin \sqrt{\frac{n}{2(n-1)}} = 1.$$

$$a = \sqrt{\frac{2n}{(n-1)}} \frac{1}{2 \arcsin \sqrt{\frac{n}{2(n-1)}}}.$$

$$R = \frac{1}{2 \arcsin \sqrt{\frac{n}{2(n-1)}}}$$

so we have

$$C_{n-2}(G) = 4\left(\arcsin \sqrt{\frac{n}{2(n-1)}}\right)^2.$$

Also, it seems that for all n and d curvature $-\infty$ gives zero distortion since we can make a star from a clique using one Steiner node.

Threshold-based curvature We can embed cliques to any space (all nodes to one point).

6.5 Tree with branching factor b

Distortion-based curvature It is $-\infty$, similarly to stars.

Threshold-based curvature This can be analyzed for $d = 2$ and we can get lower and upper bounds. One bound follows from the corresponding result for star on $b+2$ vertices (we need enough space to embed the children of one node). For the other bound, I have the precise combinatorial construction which gives $z \log b$ with some constant z for which I have some analytical expression. I can write more details about this later, but the basic idea is that each level of a tree lies on a circle of some radius. This idea is well-known but I was not able to find rigorous result for our loss function.

7 Practical tasks

7.1 Network reconstruction and link prediction

[6] uses some standard complex networks: Blogcatalog, YouTube, HepTh, AstroPh, protein-protein interactions. As metrics they use Precision at k and Mean Average Precision (MAP).

[15] uses the following networks: AstroPh, CondMat, GrQc, HepPh. Each dataset is split into train, validation, and test sets. Parameters are tuned on validation, MAP is measured on test.

[15] also uses transitive closure of the WordNet noun hierarchy. Tasks: reconstruction and link prediction. Measure mean rank and (MAP). The same dataset is used in [19], where MAP is measured. Similarly, [5] uses this, treats link prediction as classifiaction task and measures precision, recall, F1.

[19] also uses fully-balanced trees along with phylogenetic trees expressing genetic heritage (of mosses growing in urban environments), a graph of Ph.D. advisor-advisee relationships, biological sets involving disease relationships, protein interactions in yeast bacteria, collaboration network Gr-QC. Measure MAP and distortion.

[16] uses the following taxonomies: WordNet (noun and verb hierarchy), EuroVoc, ACM, MeSH. Measures MR and MAP. In addition, they also evaluate how well the norm of the embeddings correlates with the ground-truth ranks in the embedded taxonomy: they measure the Spearman rank-order correlation of the normalized rank with the norm of the embedding.

[16] also embeds Enron Email Corpus and measures Spearman correlation of the norms of the embedding with the organizational rank.

7.2 Lexical Entailment

[15] uses HyperLex datasets. Uses embedding of WordNet. Records Spearman's rank correlation with the ground-truth ranking.

7.3 Node classification

[6] uses Blogcatalog and protein-protein interactions. Use embeddings as features (as input to a one-vs-rest logistic regression).

References

[1] Ittai Abraham, Mahesh Balakrishnan, Fabian Kuhn, Dahlia Malkhi, Venugopalan Ramasubramanian, and Kunal Talwar. Reconstructing approximate tree metrics. In *Proceedings of the twenty-sixth annual ACM symposium on Principles of distributed computing*, pages 43–52. ACM, 2007.

[2] Robi Bhattacharjee and Sanjoy Dasgupta. What relations are reliably embeddable in euclidean space? *arXiv preprint arXiv:1903.05347*, 2019.

[3] Josep Díaz, Dieter Mitsche, Guillem Perarnau, and Xavier Pérez-Giménez. On the relation between graph distance and euclidean distance in random geometric graphs. *Advances in Applied Probability*, 48(3):848–864, 2016.

[4] Robin Forman. Bochner's method for cell complexes and combinatorial ricci curvature. *Discrete and Computational Geometry*, 29(3):323–374, 2003.

[5] Octavian-Eugen Ganea, Gary Becigneul, and Thomas Hofmann. Hyperbolic entailment cones for learning hierarchical embeddings. In *International Conference on Machine Learning*, pages 1632–1641, 2018.

[6] Palash Goyal and Emilio Ferrara. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94, 2018.

[7] Albert Gu, Frederic Sala, Beliz Gunel, and Christopher Ré. Learning mixed-curvature representations in product spaces. *ICLR*, 2019.

[8] Edmond A Jonckheere, Poonsuk Lohsoonthorn, and Fariba Ariaei. Upper bound on scaled gromov-hyperbolic δ. *Applied Mathematics and Computation*, 192(1):191–204, 2007.

[9] Jürgen Jost and Shiping Liu. Ollivier's ricci curvature, local clustering and curvature-dimension inequalities on graphs. *Discrete & Computational Geometry*, 51(2):300–322, 2014.

[10] Ross J Kang and Tobias Müller. Sphere and dot product representations of graphs. *Discrete & Computational Geometry*, 47(3):548–568, 2012.

[11] Christy Kelly, Carlo A Trugenberger, and Fabio Biancalana. Self-assembly of geometric space from random graphs. *Classical and Quantum Gravity*, 36(12):125012, 2019.

[12] Yong Lin, Linyuan Lu, and Shing-Tung Yau. Ricci curvature of graphs. *Tohoku Mathematical Journal, Second Series*, 63(4):605–627, 2011.

[13] Hiroshi Maehara. Space graphs and sphericity. *Discrete Applied Mathematics*, 7(1):55–64, 1984.

[14] Chien-Chun Ni, Yu-Yao Lin, Jie Gao, Xianfeng David Gu, and Emil Saucan. Ricci curvature of the internet topology. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 2758–2766. IEEE, 2015.

[15] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in neural information processing systems*, pages 6338–6347, 2017.

[16] Maximillian Nickel and Douwe Kiela. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *International Conference on Machine Learning*, pages 3776–3785, 2018.

[17] Yann Ollivier. Ricci curvature of markov chains on metric spaces. *Journal of Functional Analysis*, 256(3):810–864, feb 2009.

[18] Jan Reiterman, Vojtech Rödl, and E Šinajová. Geometrical embeddings of graphs. *Discrete Mathematics*, 74(3):291–319, 1989.

[19] Frederic Sala, Chris De Sa, Albert Gu, and Christopher Re. Representation tradeoffs for hyperbolic embeddings. In *International Conference on Machine Learning*, pages 4457–4466, 2018.

[20] Areejit Samal, RP Sreejith, Jiao Gu, Shiping Liu, Emil Saucan, and Jürgen Jost. Comparative analysis of two discretizations of ricci curvature for complex networks. *Scientific reports*, 8(1):8650, 2018.

[21] Rik Sarkar. Low distortion delaunay embedding of trees in hyperbolic plane. In *International Symposium on Graph Drawing*, pages 355–366. Springer, 2011.

[22] RP Sreejith, Karthikeyan Mohanraj, Jürgen Jost, Emil Saucan, and Areejit Samal. Forman curvature for complex networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2016(6):063206, 2016.

[23] Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. Poincaré glove: Hyperbolic word embeddings. *arXiv preprint arXiv:1810.06546*, 2018.

[24] Melanie Weber, Emil Saucan, and Jürgen Jost. Coarse geometry of evolving networks. *Journal of Complex Networks*, 6(5):706–732, 11 2017.