NAZARBAYEV UNIVERSITY

SCHOOL OF SCIENCE AND HUMANITIES

Sultan Nurmukhamedov

Constructing Word Embeddings from the Random Hyperbolic Graph.

Mathematics major

Capstone project

Supervisor:

Zh. Assylbekov, PhD

Second reader:

T. Mach, PhD

Nur-Sultan, 2020

Contents

Abstract

Recent studies have shown that the appropriate space for word embeddings is not the Euclidean space, but negatively curved, hyperbolic space. We randomly throw points in the hyperbolic disc and claim that these points are already word representations. However, it is yet to be uncovered which point corresponds to which word of the human language of interest. This correspondence can be approximately established using a pointwise mutual information between words and graph matching techniques. The embeddings were evaluated at WS353 task, and then separately on its similarity and relatedness parts.

Chapter 1

Introduction

Word embeddings are vector representations of human words. What is the easiest way to get a vector of real numbers from a word? It seems that it will be natural to take the n-dimensional vector, where n is the vocabulary size, with only one non-zero element equal to 1 in the position corresponding to the word index in the dictionary. This approach is called one-hot encoding. Suppose we have 5 words in our dictionary “I”, “love”, “dogs”,“and”, “cats”. One-hot encoding of our words would be “I” =$[1, 0, 0, 0, 0]^{\mathrm{T}}$, “love” =$[0, 1, 0, 0, 0]^{\mathrm{T}}$, “dogs” =$[0, 0, 1, 0, 0]^{\mathrm{T}}$, and so on. The idea of one-hot encoding has a significant drawback: vector representation of a word has nothing to do with its meaning. In our case words ”dogs” and ”cats” are as similar as words “love” and “and”, which is not a useful approach. The goal is to construct such embeddings that will reflect similarities and dissimilarities between words. The idea is based on the distributional hypothesis, which states that words in similar contexts have similar meanings.

One of the widely used word embeddings models is the Skip-gram with negative sampling (SGNS) of Mikolov et al. (2013). SGNS is the machine learning model that is trained to find the most related words for a given word. However it was shown by Levy and Goldberg (2014) that SGNS is implicitly factorizing a matrix, whose entries are the pointwise mutual information (PMI) between words. Moreover, Assylbekov and Jangeldin (2020) have shown that vectors of comparable quality can also be obtained from factorizing a binarized PMI (BPMI) matrix, where $\mathrm{BPMI}_{i,j} = 1,$ if $\mathrm{PMI}_{i,j} > 0,$ and $\mathrm{BPMI}_{i,j} = 0$, otherwise. The obtained binarized PMI matrix can be interpreted as an adjacency matrix of a particular graph. Also, they have shown that the graph obtained from BPMI matrix possesses properties of a complex network, namely, scale-free degree distribution and a strong clustering coefficient.

On the other side,Krioukov et al. (2010), have shown that hyperbolic geometry underlies complex networks. Which means that if we construct Random Hyperbolic Graph (RHG) it will have scale-free degree distribution and a strong clustering. Therefore, it seems reasonable to analyze RHG as a model for word embeddings. This capstone is the first step in constructing word vectors from randomly thrown points on the hyperbolic disk.

Chapter 2

Theory

2.1 Hyperbolic geometry

First, let us briefly review the basic facts about hyperbolic geometry. There are three types of isotropic spaces: Euclidian (curvature is 0), Spherical (curvature is positive) and hyperbolic (curvature is negative). Hyperbolic geometry emerges from relaxing Euclid's fifth axiom, which says that for any straight line and a point not on it, there "exists one and only one straight line which passes" through that point and never intersects the first line. In hyperbolic plane there is an infinite number of parallel lines that pass through a single point. (Figure 1)

There are four models commonly used for hyperbolic geometry: the Klein model, the Poincare disk, the Poincare half-plane model and hyperboloid model. Each model represents different aspects of hyperbolic geometry, but no model represents all of its properties. I will use the Poincare disk model, in which the hyperbolic plane $\mathbb{H}^2$ is represented on a disk of radius 1 and lines are arcs of circles that are orthogonal to the boundary of the disk, plus all diameters of the disk. This model is conformal, which means that Euclidean angles between lines in the model are equal to the hyperbolic angle. However, it is not true for areas and distances. Euclidean and hyperbolic distances from the center of the disk are related by: $r_e \ = \tanh(\frac{r_h}{2})$, where r_e is euclidean distance and r_h is hyperbolic distance.

One of the defining characteristics of hyperbolic space is that it expands faster (exponentially) than Euclidian space (polynomially). For example formulas for the disk area and length of a circle are:

$$\text{Disk area} = \ 2\pi\frac{\cosh{(\zeta r - 1)}}{\zeta^2} = A(r),$$

$$\text{Circle length} = \ 2\pi\frac{\sinh{(\zeta r)}}{\zeta} = L(r).$$

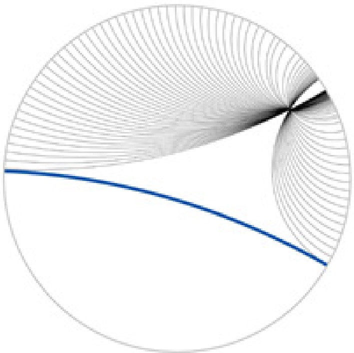


Figure 1: Poincaré disk with hyperbolic parallel lines. https://commons.wikimedia.org

Where r is the radius of the disk and $\zeta = \sqrt{-\kappa}$, where κ is the curvature of the space

Distance between two points is:

$$\mathrm{x} = \frac{\cosh^{-1}(\cosh(\zeta r_1)\cosh(\zeta r_2) \ - \sinh(\zeta r_1)\sinh(\zeta r_2)\cos(\varDelta\theta)\ )}{\zeta},$$

where $\varDelta\theta = \pi - |\pi - |\theta_1 - \theta_2||$ is the angle between the points. (r_1, θ_1) and (r_2, θ_2) are polar coordinates of the points.

2.2 Random Hyperbolic Graph

Using the facts above, we can construct Random Hyperbolic Graph as in the work of Krioukov et al. (2010). Firstly, we throw randomly N nodes on a hyperbolic disk of certain radius R. We assign angular coordinates via uniform distribution $\theta \sim$ Unif $[0, 2\pi]$ and radial coordinates from the exponential probability distribution function:

$$p(r) = \frac{\alpha \sinh(\alpha r)}{\cosh(\alpha R)\ - 1} \approx \alpha e^{\alpha(r-R)},$$

where $\alpha > 0$ is parameter, which allows us to control the distribution of nodes. For example, for $\alpha = \zeta$ the node density is uniform, since:

$$p(r) = \frac{\alpha \sinh(\alpha r)}{\cosh(\alpha R)\ - 1} = \frac{\zeta \sinh(\zeta r)}{\cosh(\zeta R)\ - 1} = \frac{L(r)}{A(r)}.$$

A larger α means that more nodes are close to the boundary of the disk, small α means that more nodes are close to the origin of the disk.

To form a graph we need to connect pair of nodes with some probability function. That probability function must have only one parameter,which is hyperbolic distance between the nodes. The simplest one is the Heaviside step function, i.e. two nodes are neighbors if and only if the hyperbolic distance between them is less than or equal to R.

2.3 Skip-Gram with Negative Sampling (SGNS)

In this section, I briefly describe one of the widely used word embeddings models Skip-gram with negative sampling (SGNS) of Mikolov et al. (2013). SGNS is the machine learning model that is trained to find the most related words for a given word. Most word embeddings algorithms construct two sort of vectors: for words and for contexts.

Notation:

1. Let's $\mathcal{W} := \{1, \ldots, n\}$ be our vocabulary.
2. The vector representation of word $i \in \mathcal{W}$ is $\mathrm{w}_i \in \mathbb{R}^d$, context $j \in \mathcal{W}$ is $\mathbf{c}_j \in \mathbb{R}^d$, where d is an embedding dimension.
3. D is the collection of observed words and context pairs. We say that word j is in the context of word i if j is in the window of i. For example, a ± 2 window means 2 words to the left and 2 words to the right of the target word.
4. $\#(i, j)$ is the number of times pair (i, j) is in D.
5. $\#\,(i)$ is the number of times word w occurs in D, i.e. $\#\,(i) = \sum_{j' \in \mathcal{W}}\ \#(i, j')$

6. $\# (j)$ is the number of times word c occurs in D, i.e. $\# (j) = \sum_{i' \in \mathcal{W}} \#(i', j)$

7. $\boldsymbol{W} \in \mathbb{R}^{|\mathcal{W}| \times d}$ is matrix with the word vectors as rows.

8. $\boldsymbol{C} \in \mathbb{R}^{|\mathcal{W}| \times d}$ is matrix with the context vectors as rows.

9. $\Pr(D = 1 |\ (i, j)) = log\sigma\left(\ \mathbf{w}_i \cdot\ \mathbf{c}_j\right) = \frac{1}{1 + e^{-\mathbf{w}_i \cdot\ \mathbf{c}_j}}$ is probability that pair (i, j) is from D.

10. $\Pr(D = 0 |\ (i, j)) = 1 - \Pr(D = 1 |\ (i, j)) = log\sigma\left(-\ \mathbf{w}_i \cdot\ \mathbf{c}_j\right)$ is probability that pair (i, j) is not from D.

Our goal is to maximize $\Pr(D = 1 |\ (i, j))$ for $(i, j) \in D$ and maximize $\Pr(D = 0 |\ (i, j))$ for randomly sampled pairs ("negative sampling"). Let k be the number of negative samples and j' is a sampled context from the empirical distribution $P_D(j') = \frac{\#(j')}{|D|}$.
For a single pair (i, j) the objective function is:

$$\Pr(D = 1 |\ (i, j)) + k\mathbb{E}_{j' \sim P_D}\left[\Pr(D = 0 |\ (i, j'))\right] = log\sigma\langle \mathbf{w}_i, \mathbf{c}_j \rangle + k\mathbb{E}_{j' \sim P_D}\left[log\sigma\langle -\mathbf{w}_i, \mathbf{c}_{j'} \rangle\right].$$

SGNS's objective function is:

$$\sum_{i \in \mathcal{W}} \sum_{j \in \mathcal{W}} \#(i, j)[log\sigma\langle \mathbf{w}_i, \mathbf{c}_j \rangle + k\mathbb{E}_{j' \sim P_D}\langle -\mathbf{w}_i, \mathbf{c}_{j'} \rangle].$$

Without going into technical details, maximizing this function using stochastic gradient descent we get word and context vectors, where words with similar meaning have similar embedding.

2.4 PMI and Alternative word representation

Pointwise mutual information (PMI) is a measure between a pair of words i and j, defined as the probability of their co-occurrence divided by the probabilities of them appearing individually:

$$\mathbf{PMI}(i, j) = \log \frac{p(i, j)}{p(i)\, p(j)}.$$

Empirically we can calculate PMI from the real text as follows:

$$p(i, j) = \frac{\#(i, j)}{|D|},\ \ p(i) = \frac{\#(i)}{|D|},\ \ p(j) = \frac{\#(j)}{|D|}.$$

Pairs with number of co-occurrence slightly lower than number of occurrences of each word individually have high PMI. Similarly, pairs whose number of co-occurrence is much less than occurrences of each word have small PMI. The table below shows counts of pairs of words getting the most and the least PMI scores filtering by 1,000 or more co-occurrences and $|D| = 50000952$.

Word i	Word j	$\# (i)$	$\# (j)$	$\#(i, j)$	$PMI\,(i, j)$
puerto	rico	1938	1311	1159	10.03
hong	kong	2438	2694	2205	9.73
to	in	1025659	1187652	1066	-3.13
of	and	1761436	1375396	1190	-3.70

Table 1 The most and the least PMI scores filtered by 1000 or more co-occurrence

PMI matrix is a symmetric matrix with entries $\mathrm{M}_{i,j}^{\mathrm{PMI}} = \mathbf{PMI}(i, j)$.

Levy and Goldberg (2014) have shown that SGNS is implicitly factorizing a matrix, whose entries are the pointwise mutual information (PMI) between words:

$$\langle \mathbf{w}_i, \mathbf{c}_j \rangle = \mathbf{PMI}(i, j) - \log k \to \mathbf{W} \cdot \mathbf{C}^T = \mathbf{M}^{PMI} - \log k,$$

where k corresponds to the number of ”negative” samples in SGNS model.

The main problem with PMI is that most of the pairs in our matrix were never observed in the text, for these pairs $\mathbf{PMI} = \log 0 = -\infty$. One way to solve this problem is to use Positive PMI (PPMI) where all negative values become 0: $\mathbf{PPMI}\left(i, j\right) = \max\left\{\mathbf{PMI}\left(i, j\right), 0\right\}$

These facts suggest the use of shifted PPMI (SPPMI):

$$\mathbf{SPPMI}_k\left(i, j\right) = \max\left\{\mathbf{PMI}\left(i, j\right) - \log k, 0\right\}.$$

Levy and Goldberg (2014) also show empirically that the low-rank Singular Value Decomposition (SVD) of the SPPMI matrix produces word vectors which are comparable in quality to those of the SGNS.

Singular Value Decomposition factorizes $\mathrm{M} \in \mathbb{R}^{n \times n}$ into the product of three matrices:

$$\mathbf{M} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}}.$$

where $\mathbf{U}, \mathbf{V}$ are orthonormal matrices and S is a diagonal matrix of singular values.

The rank d approximation of M is $\mathrm{M}_d = \mathrm{U}_d \mathrm{S}_d \mathrm{V}_d^{\mathrm{T}}$, where S_d is the diagonal matrix formed from the top d singular values. $\mathbf{U}_d$ and $\mathbf{V}_d$ are matrices produced by selecting corresponding columns from $\mathbf{U}$ and $\mathbf{V}$.
$\mathbf{W} \approx \mathbf{U}_d\sqrt{\mathbf{S}_d}$ and $\mathbf{C} \approx \mathbf{V}_d\sqrt{\mathbf{S}_d}$
Word vectors produced that way are comparable in quality to those of the SGNS.

2.5 BPMI

Assylbekov and Jangeldin (2020) introduced stronger roughening of the original PMI matrix – Binarized PMI (BPMI).

$$\mathbf{A}_{ij} := H\left(\log \tfrac{p(i,j)}{p(i)p(j)}\right), \tag{1}$$

where $H(x) = 1$ if $x > 0$, and $H(x) = 0$ otherwise.
Vectors of comparable quality can also be obtained from a low-rank approximation of a binarized PMI matrix. Thus, a binarized PMI matrix is also an option when it comes to word vectors. BPMI matrix can be interpreted as an adjacency matrix for some graph. The graph obtained from BPMI matrix possess properties of a complex network, namely, scale-free degree distribution and strong clustering coefficient, and according to Krioukov et al. (2010), such graph possesses an effective hyperbolic geometry underneath. The following chain summarizes this argument:

$$\boxed{\text{Word Embeddings}} \longrightarrow \boxed{\text{BPMI}} \longrightarrow \boxed{\text{Complex Network}} \longrightarrow \boxed{\text{Hyperbolic Space}}$$

In this work, we go from the final point (hyperbolic space) to the starting one (word embeddings), and the next section provides the details of our method.

Chapter 3

Constructing Word Embeddings from the random hyperbolic graph.

3.1 Distances in RHG

Proposition 3.1.1. *Let X be a distance between two points from the Random Hyperbolic Graph with parameters α, ζ and radius R. The probability distribution function of X is given by*

$$f_X(x) = \int_0^R \int_0^R \frac{\zeta \sinh(\zeta x)}{\pi\sqrt{1 - A(r_1, r_2, x)} \sinh(\zeta r_1)\sinh(\zeta r_2)} \rho(r_1)\rho(r_2) dr_1 dr_2, \tag{2}$$

for $A(r_1, r_2, x) \in (-1, 1)$ where $A(r_1, r_2, x) = \frac{\cosh(\zeta r_1)\cosh(\zeta r_2) - \cosh(\zeta x)}{\sinh(\zeta r_1)\sinh(\zeta r_2)}$, and $\rho(r) = \frac{\alpha \sinh \alpha r}{\cosh \alpha R - 1}$.

Proof. Let us throw randomly two points (r_1, θ_1) and (r_2, θ_2) into the hyperbolic disk of radius R, i.e. $r_1, r_2 \overset{\text{i.i.d.}}{\sim} \rho(r), \theta_1, \theta_2 \overset{\text{i.i.d.}}{\sim}$ Uniform$[0, 2\pi)$. Let X be the distance between these points (X is a random variable). Let γ be the angle between these points, then $\gamma := \pi - |\pi - |\theta_1 - \theta_2|| \sim$ Uniform$[0, \pi)$ and thus

$$f_{\cos\gamma}(t) = \frac{1}{\pi\sqrt{1 - t^2}}, \quad t \in (-1, 1).$$

Since the distance in our model of hyperbolic plane is given by

$$X = \frac{\cosh^{-1}[\cosh \zeta r_1 \cosh \zeta r_2 - \sinh \zeta r_1 \sinh \zeta r_2 \cos\gamma]}{\zeta},$$

we have

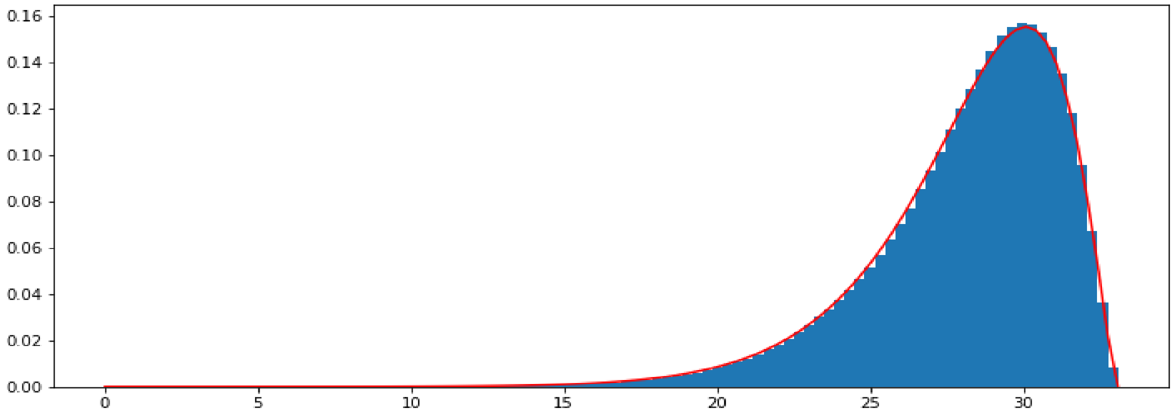
$$\Pr(X \leq x) = \Pr\left(\cos\gamma \geq \underbrace{\frac{\cosh \zeta r_1 \cosh \zeta r_2 - \cosh \zeta x}{\sinh \zeta r_1 \sinh \zeta r_2}}_{A(r_1, r_2, x)}\right)$$
$$= \Pr(\cos\gamma \geq A(r_1, r_2, x)) = \int_{A(r_1, r_2, x)}^{+\infty} \frac{dt}{\pi\sqrt{1 - t^2}} = \frac{1}{2} - \frac{\sin^{-1} A(r_1, r_2, x)}{\pi},$$

and therefore

$$f_{X|r_1, r_2}(x) = \frac{d}{dx}\left[\frac{1}{2} - \frac{\sin^{-1} A(r_1, r_2, x)}{\pi}\right]$$
$$= \frac{\zeta \sinh \zeta x}{\pi\sqrt{1 - A(r_1, r_2, x)} \sinh(\zeta r_1) \sinh \zeta r_2} \quad \text{for A(r}_1, r_2, x) \in (-1, 1).$$

Integrating $f_{X|r_1, r_2}(x)\rho(r_1)\rho(r_2)$ with respect to r_1 and r_2 we get (2). □

You can see that (2) perfectly matches simulation in the picture below



Normalized histogram of distances and theoretical p.d.f of distances in the RHG

Notice, that the adjacency matrix of our graph is $\mathbf{B}_{ij} := H(R - x_{ij})$, and comparing this to (1), we see that if $\mathbf{A}$ and $\mathbf{B}$ induce structurally similar graphs then the distribution of the PMI values $\log \frac{p(i,j)}{p(i)p(j)}$ should be similar to the distribution of $R - x_{ij}$ values. To test this empirically, we compute a PMI matrix of a well-known corpus, `text8`,[1] and compare the distribution of the PMI values with the p.d.f. of $R - X$, where X is a distance between two random points of a hyperbolic disk (the exact form of this p.d.f. is given in Proposition 3.1.1). The results are shown in Figure 2. As we can see, the two distributions are indeed similar and the main difference is in the shift—distribution of $R - X$ is shifted to the left compared to the distribution of the PMI values. This possibly explains why the PMI entries are shifted to the *left* in the matrix factorization approach of Levy and Goldberg (2014). We hypothesize that the nodes of the RHG treated as points of the hyperbolic space are already reasonable word embeddings for the words.

3.2 Average degree distribution

To compute the degeee distribution P(k) we have to calculate the average degree $\overline{k}(r)$ of nodes located at distance r from the origin.

Proposition 3.2.1. *Let $\overline{k}(r_1)$ be the average degree of a node located at distance r_1 from the origin of Random Hyperbolic Graph with parameters α, ζ radius R and N number of vertices, then*

$$\overline{k}(r_1) = N[\int_0^{r^\star} p(r_2)dr_2 + \int_{r^\star}^{R} [\frac{\rho(r_2)}{2} - \frac{sin^{-1}A(r_1, r_2, R)\rho(r_2)}{\pi}]dr_2. \tag{3}$$

$A(r_1, r_2, x) = \frac{\cosh(\zeta r_1)\cosh(\zeta r_2) - \cosh(\zeta x)}{\sinh(\zeta r_1)\sinh(\zeta r_2)}$, $\rho(r) = \frac{\alpha \sinh \alpha r}{\cosh \alpha R - 1}$ *and $r^\star$ is such that $A(r_1, r^\star, R) = -1$.*

Proof. Suppose the given point has polar coordinate (r_1, θ_1). Let us throw randomly a point (r_2, θ_2) into the hyperbolic disk of radius R, i.e. $r_2 \overset{\text{i.i.d.}}{\sim} \rho(r)$, $\theta_2 \overset{\text{i.i.d.}}{\sim} \text{Unif}[0, 2\pi)$. Let X be the distance between these points. Using the same technique as in the proof of Proposition 3.1.1 we get:

[1] `http://mattmahoney.net/dc/textdata.html`

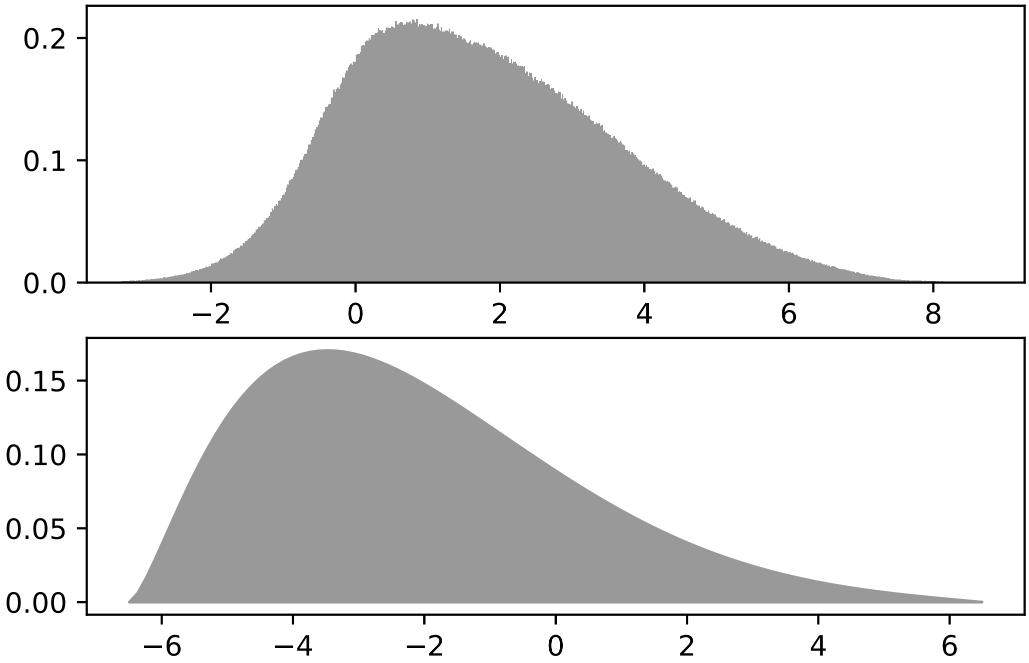


Figure 2: Distribution of PMI values (top) and of $R - X$.

$$\Pr(X \leq R) = \begin{cases} 0, & \text{if } A(r_1, r_2, R) \geq 1 \\ \frac{1}{2} - \frac{\sin^{-1} A(r_1,r_2,R)}{\pi}, & \text{if } A(r_1, r_2, R) \in (-1, 1) \\ 1, & \text{if } A(r_1, r_2, R) \leq -1 \end{cases}$$

Integrating $\Pr(X \leq R)\rho(r_2)$ with respect to r_2 we get (3). □

However, it is difficult to work with these integrals. Let us show that this exact expression (3) can be approximated to more elegant equation for $\overline{k}(r)$ that was used in the work of (Krioukov et al., 2010).

3.3 Average degree distribution approximation

Proposition 3.3.1. *Let $\overline{k}(r_1)$ be the average degree of a node located at distance r_1 from the origin of Random Hyperbolic Graph with parameters α, ζ radius R and N number of vertices, then*

$$\overline{k}(r_1) \approx N\left[\frac{2}{\pi}\xi e^{-\zeta r_1/2} - \left(\frac{2}{\pi}\xi - 1\right)e^{-\alpha r_1}\right], \tag{4}$$

where $\xi = \frac{\alpha/\zeta}{\alpha/\zeta - 1/2}$

Proof. We must show that (3) $\approx$ (4)

Since in the RHG vast majority of nodes are close to the boundary of the disk for large R $\coth(\zeta r_1) = 1$.

Let us solve $A(r_1, r^\star, R) = -1$ for $r^\star$

$$A(r_1, r^\star, R) = \frac{\cosh(\zeta r_1)\cosh(\zeta r^\star)}{\sinh(\zeta r_1)\sinh(\zeta r^\star)} - \frac{\cosh(\zeta R)}{\sinh(\zeta r_1)\sinh(\zeta r^\star)} = -1$$

$$\Longleftrightarrow \frac{\cosh(\zeta r^\star)}{\sinh(\zeta r^\star)} - \frac{\cosh(\zeta R)}{\sinh(\zeta r_1)\sinh(\zeta r^\star)} = -1$$

$$\Longleftrightarrow \cosh(\zeta r^\star) - \frac{\cosh(\zeta R}{\sinh(\zeta r_1)} = -\sinh(\zeta r^\star)$$

$$\Longleftrightarrow \cosh(\zeta r^\star) + \sinh(\zeta r^\star) = \frac{\cosh(\zeta R}{\sinh(\zeta r_1)}$$

$$\Longleftrightarrow r^\star = \frac{1}{\zeta}\ln\left[\frac{\cosh(\zeta R)}{\sinh(\zeta r_1)}\right] \approx R - r_1.$$

Now we can rewrite equation (3):

$$\overline{k}(r_1) = N\bigg[\int_0^{R-r_1} p(r_2)dr_2 + \int_{R-r_1}^{R}\bigg(\frac{\rho(r_2)}{2} - \frac{\sin^{-1}A(r_1,r_2,R)\rho(r_2)}{\pi}\bigg)dr_2\bigg].$$

Let us calculate the first integral:

$$\int_0^{r^\star} p(r_2)dr_2 \approx \int_0^{R-r_1} \alpha e^{\alpha(r_2-R)}dr_2 = e^{-\alpha r_1}.$$

For the second integral the Taylor series of $\sin^{-1}$ around 1 with two first terms was used.

$$\sin^{-1}A(r_1,r_2,R) \approx \pi/2 - \sqrt{2}(1-A(r_1,r_2,R))^{1/2}$$

$$\int_{R-r_1}^{R}\bigg(\frac{\rho(r_2)}{2} - \frac{\sin^{-1}A(r_1,r_2,R)\rho(r_2)}{\pi}\bigg)dr_2 \approx \frac{\sqrt{2}}{\pi}\int_{R-r_1}^{R}(1-A(r_1,r_2,R))^{1/2}\rho(r_2)dr_2$$

$$= \frac{\sqrt{2}}{\pi}\int_{R-r_1}^{R}\bigg(1-\bigg(1-\frac{\cosh(\zeta R)}{\sinh(\zeta r_1)\sinh(\zeta r_2)}\bigg)\bigg)^{1/2}\rho(r_2)dr_2$$

$$= \frac{\sqrt{2}}{\pi}\int_{R-r_1}^{R}\rho(r_2)\bigg(\frac{e^{\zeta R}+e^{-\zeta R}}{(e^{\zeta r_1}-e^{-\zeta r_1})\sinh(\zeta r_2)}\bigg)^{1/2}dr_2 \approx \frac{\sqrt{2}}{\pi}\int_{R-r_1}^{R}\rho(r_2)\bigg(\frac{e^{\zeta(R-r_1)}}{\sinh(\zeta r_2)}\bigg)^{1/2}dr_2$$

$$\approx \frac{\sqrt{2}}{\pi}\int_{R-r_1}^{R}\alpha e^{\alpha(r_2-R)}\bigg(\frac{2e^{\zeta(R-r_1)}}{e^{\zeta r_2}-e^{-\zeta r_2}}\bigg)^{1/2}dr_2 \approx \frac{\sqrt{2}}{\pi}\int_{R-r_1}^{R}\alpha e^{\alpha(r_2-R)}\bigg(\frac{2e^{\zeta(R-r_1)}}{e^{\zeta r_2}}\bigg)^{1/2}dr_2$$

$$= \frac{\sqrt{2}}{\pi}\alpha\sqrt{2e^{\zeta(R-r_1)}}\int_{R-r_1}^{R}e^{\alpha r_2-\alpha R-\zeta r_2/2}dr_2 = \frac{\sqrt{2}}{\pi}\frac{\alpha\sqrt{2e^{\zeta(R-r_1)}}}{\alpha-\zeta/2}(e^{-\zeta R/2}-e^{\zeta r_1/2-\zeta R/2-\alpha r_1})$$

$$= \frac{2}{\pi}(\xi e^{-\zeta r_1/2} - \xi e^{-\alpha r_1}).$$

Summing the both integrals and multiplying by N we got:

$$\overline{k}(r_1) \approx N\bigg[\frac{2}{\pi}\xi e^{-\zeta r_1/2} - (\frac{2}{\pi}\xi - 1)e^{-\alpha r_1}\bigg].$$

□

The resulting formula is the same as in the work of (Krioukov et al., 2010). You can see that obtained approximation perfectly fits the simulation in the Figures 3 and 4. Using this approximation we can calculate average degree for our graph:

$$\overline{k} = \int_0^R \rho(r)\overline{k}(r)dr \approx \frac{2}{\pi}\xi^2 N e^{-\zeta R/2}. \tag{5}$$

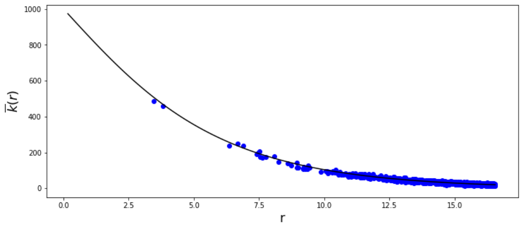


Figure 3: Theoretical(black curve) and experimental (blue dots) average degree distribution in RHG with parameters $\alpha = 0.5$, $\zeta = 0.5 \;\; R = 16.55$

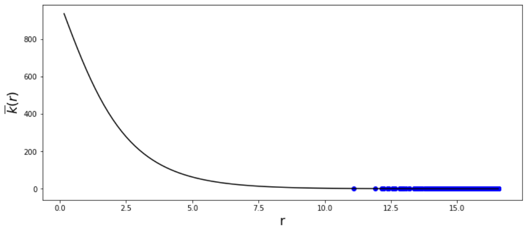


Figure 4: Theoretical (black curve) and experimental (blue dots) average degree distribution in RHG with parameters $\alpha = 1.2$, $\zeta = 1.2 \;\; R = 16.55$

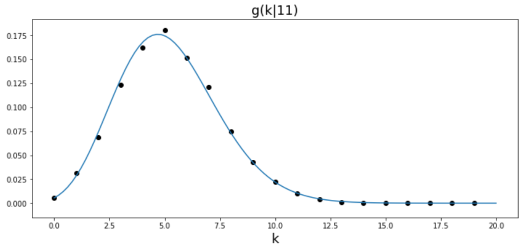


Figure 5: Theoretical and experimental distribution of degree for a node at distance 11 from the origin in RHG with parameters $\alpha = 1, \;\zeta = 1 \;\; R = 16.55$

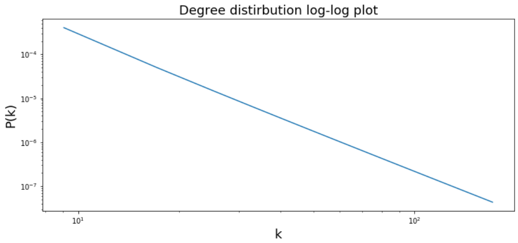


Figure 6: P(k) of RHG in log-log scale with parameters $\alpha = 1, \;\zeta = 1 \;\; R = 16.55$

3.4 Calculating P(k)

Now we can compute degree distribution P(k) using by this formula:

$$P(k) = \int_0^R g(k|r)p(r)dr, \tag{6}$$

where $g(k|r)$ is the conditional probability that a node with radial coordinate r has degree k. It was shown in the work of (Boguna, 2003) that for complex networks $g(k|r)$ is Poisson distribution with parameter $\overline{k}(r)$. You can see that Poisson distribution perfectly matches simulation for a node at distance 11 from the origin in Figure 5

Now we can rewrite equation 6 as:

$$P(k) = \int_0^R \frac{e^{-\overline{k}(r)}[\overline{k}(r)]^k}{k!} \alpha e^{\alpha(r-R)} dr.$$

This integral can be easily calculated by substitution $t = \overline{k}(r)$:

$$P(k) = \frac{\overline{k}^2}{2} \int_{\overline{k}/2}^{\overline{k}(0)} \frac{e^{-t} t^{k-2}}{k!} dt \approx \frac{\overline{k}^2}{2} \int_{\overline{k}/2}^{\infty} \frac{e^{-t} t^{k-2}}{k!} dt = \frac{\overline{k}^2}{2} \frac{\Gamma(k-2, \overline{k}/2)}{k!}, \tag{7}$$

where $\Gamma(s,x) = \int_x^\infty t^{s-1} e^{-t} dt$ is incomplete gamma function.
It can be shown that equation (7) behaves as a power law of the form $k^{-(2\alpha/\zeta+1)}$. The degree distribution of the RHG follows a power law. In log-log scale the power law function is a straight line. In figure 6 you can see the equation (7) in log-log scale.

3.5 Finding permutation

We can treat nodes of RHG as an embedding for words of our vocabulary. However, we do not know which point corresponds to which word of the human language. Our goal is to find one-to-one correspondence between human words and nodes on the hyperbolic disk.
Let $\mathbf{A}, \mathbf{B}$ are adjacency matrices of BPMI graph and RHG. The problem of finding the correspondence between words and nodes can be interpreted as the following optimization problem:

$$\left\|\mathbf{A} - \mathbf{P}\mathbf{B}\mathbf{P}^{\mathrm{T}}\right\|_{\mathrm{F}}^{2} \to \min\nolimits_{\mathbf{P}\in\mathcal{P}_n},$$

where $\mathbf{P}$ is a permutation matrix, $\mathcal{P}_n$ – set of all permutation matrices of size $n \times n$ and $\|\circ\|_{\mathrm{F}}$ is a Frobenius norm.

$$\begin{aligned}\left\|\mathbf{A} - \mathbf{P}\mathbf{B}\mathbf{P}^{\mathrm{T}}\right\|_{\mathrm{F}}^{2} &= \|\mathbf{A}\|_{\mathrm{F}}^{2} - 2\left\langle \mathbf{A}, \mathbf{P}\mathbf{B}\mathbf{P}^{\mathrm{T}}\right\rangle_{\mathrm{F}} + \left\|\mathbf{P}\mathbf{B}\mathbf{P}^{\mathrm{T}}\right\|_{\mathrm{F}}^{2} \\ &= \|\mathbf{A}\|_{\mathrm{F}}^{2} - 2\left\langle \mathbf{A}, \mathbf{P}\mathbf{B}\mathbf{P}^{\mathrm{T}}\right\rangle_{\mathrm{F}} + \|\mathbf{B}\|_{\mathrm{F}}^{2} \\ &= \|\mathbf{A}\|_{\mathrm{F}}^{2} - 2\mathrm{tr}(\mathbf{A}^{\mathrm{T}}\mathbf{P}\mathbf{B}\mathbf{P}^{\mathrm{T}}) + \|\mathbf{B}\|_{\mathrm{F}}^{2}.\end{aligned}$$

As $\|\mathbf{A}\|_{\mathrm{F}}^{2}, \|\mathbf{B}\|_{\mathrm{F}}^{2}$ are constants, the problem is to maximize $\mathrm{tr}(\mathbf{A}^{\mathrm{T}}\mathbf{P}\mathbf{B}\mathbf{P}^{\mathrm{T}})$.
However, this problem also known as the quadratic assignment problem (QAP) is one of the fundamental combinatorial optimization problems. The problem is NP-hard, so there is no known algorithm for exact solution of this problem in polynomial time. The approximate solution can be found in the work of Umeyama (1988):
First, take SVD of $\mathbf{A}$ and $\mathbf{B}$:

$$\begin{aligned}\mathbf{A} &= \mathbf{U_A}\mathbf{S_A}\mathbf{V}_{\mathbf{A}}^{\mathrm{T}}, \\ \mathbf{B} &= \mathbf{U_B}\mathbf{S_B}\mathbf{V}_{\mathbf{B}}^{\mathrm{T}}.\end{aligned}$$

Let $\tilde{\mathbf{P}} = \overline{\mathbf{U_A}}\ \overline{\mathbf{U_B}}^{\mathrm{T}}$ where $\overline{\mathbf{U_A}}$ and $\overline{\mathbf{U_B}}^{\mathrm{T}}$ are matrices which have as each element the absolute value of each element of $\mathbf{U_A}$, and $\mathbf{U_B}$, respectively.
The next step is to find the 'nearest' permutation matrix $\mathbf{P}$ to $\tilde{\mathbf{P}}$, i.e. we have to solve:

$$\left\|\mathbf{P} - \tilde{\mathbf{P}}\right\|_{\mathrm{F}}^{2} \to \min\nolimits_{\mathbf{P}\in\mathbf{P}_n}.$$

This is typical linear assignment problem and can be solved using Auction algorithm of Bertsekas (1979).
Finally, we can obtain word vectors from $\mathbf{P}\mathbf{B}\mathbf{P}^{\mathrm{T}}$ via low-rank approximation as was described in section 2.4

Chapter 4

Evaluation

In this section we evaluate the quality of word vectors resulting from a RHG against those from the SGNS, PMI, and BPMI. We use the `text8` corpus mentioned in the previous section. We were ignoring words that appeared less than 500 times (resulting in a vocabulary of 3,446 tokens). We set window size to 2, and dimensionality of word vectors to 200. The embeddings were evaluated on WS353 task (Finkelstein et al., 2002), and then separately on its similarity and relatedness parts.

	Overall	Similarity	Relatedness
SGNS	.669	.767	.661
PMI + SVD	.432	.498	.433
BPMI + SVD	.362	.432	.322
RHG + Permute + SVD	.263	.254	.246

Table 2: Evaluation of word embeddings on the WS353 task. Evaluation metric is the Spearman's correlation with the human ratings.

The results of evaluation are provided in Table 2. As we can see, vector representations of words generated by a random hyperbolic graph lag behind in quality from word vectors obtained by other standard methods. From this perspective, the results can be considered negative. However, as mentioned in the introduction, the main goal is to find a simple mathematical structure for word vectors. Moreover, this paper is the first step in finding such a structure.

Chapter 5

Conclusion

Since hyperbolic space underlies complex networks, it is logical to exploit its properties in constructing word vectors. It was shown that by throwing points randomly on the hyperbolic plane, we get word representations such that each point corresponds to a certain word of the human language, and this correspondence is determined by the relation (hyperbolic distance) to other words. Hyperbolic space seems to be the right direction of the search. Perhaps, the mediocre quality of the resulting vectors is due to the scarcity of the two-dimensional hyperbolic plane $\mathbb{H}^2$. This conclusion is fully consistent with the principle of semiotic arbitrariness of De Saussure (2011)—the relationship between a word (sign) and the real-world thing it denotes is an arbitrary one.

References

Zhenisbek Assylbekov and Alibi Jangeldin. 2020. Binarized pmi matrix: bridging word embeddings and hyperbolic spaces. *arXiv preprint arXiv:2002.12005*.

Pastor-Satorras Boguna. 2003. Class of correlated random networks with hidden variables.

Ferdinand De Saussure. 2011. *Course in general linguistics*. Columbia University Press.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: The concept revisited. *ACM Transactions on information systems*, 20(1):116–131.

Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguná. 2010. Hyperbolic geometry of complex networks. *Physical Review E*, 82(3):036106.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Proceedings of NeurIPS*, pages 2177–2185.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Shinji Umeyama. 1988. An eigendecomposition approach to weighted graph matching problems. *IEEE transactions on pattern analysis and machine intelligence*, 10(5):695–703.