From Word Embeddings to the Hyperbolic Space and Back

Zh. Assylbekov[1] A. Jangeldin[1] S. Nurmukhamedov[1]
A. Sheverdin[1] T. Mach

[1]School of Sciences and Humanities
Nazarbayev University

HSE Seminar, 2 Apr 2020

Static vs Contextualized Word Vectors

Vocabulary $\mathcal{W} := \{1, \ldots, n\}$.

Static vs Contextualized Word Vectors

Vocabulary $\mathcal{W} := \{1, \ldots, n\}$.

- *Static* embedding of a word $i \in \mathcal{W}$:

$$\mathbf{w}_i = f(i), \quad \mathbf{w}_i \in \mathbb{R}^d.$$

Static vs Contextualized Word Vectors

Vocabulary $\mathcal{W} := \{1, \ldots, n\}$.

- *Static* embedding of a word $i \in \mathcal{W}$:

$$\mathbf{w}_i = f(i), \quad \mathbf{w}_i \in \mathbb{R}^d.$$

  - WORD2VEC [Mikolov et al., 2013a,b], GLOVE [Pennington et al., 2014].

Static vs Contextualized Word Vectors

Vocabulary $\mathcal{W} := \{1, \ldots, n\}$.

- *Static* embedding of a word $i \in \mathcal{W}$:

$$\mathbf{w}_i = f(i), \quad \mathbf{w}_i \in \mathbb{R}^d.$$

  - WORD2VEC [Mikolov et al., 2013a,b], GLOVE [Pennington et al., 2014].
  - Problems with polysemous words: $f(\text{bank})$.

Static vs Contextualized Word Vectors

Vocabulary $\mathcal{W} := \{1, \ldots, n\}$.

- *Static* embedding of a word $i \in \mathcal{W}$:

$$\mathbf{w}_i = f(i), \quad \mathbf{w}_i \in \mathbb{R}^d.$$

  - WORD2VEC [Mikolov et al., 2013a,b], GLOVE [Pennington et al., 2014].
  - Problems with polysemous words: $f(\text{bank})$.

- *Contextualized* embedding of a word $i \in \mathcal{W}$ in a sentence $j_1, \ldots, j_{l-1}, i, j_{l+1}, \ldots, j_k$:

$$\mathbf{w}_i = g(j_1, \ldots, j_{l-1}, j_{l+1}, \ldots, j_k), \quad \mathbf{w}_i \in \mathbb{R}^d.$$

Static vs Contextualized Word Vectors

Vocabulary $\mathcal{W} := \{1, \ldots, n\}$.

- *Static* embedding of a word $i \in \mathcal{W}$:

$$\mathbf{w}_i = f(i), \quad \mathbf{w}_i \in \mathbb{R}^d.$$

  - WORD2VEC [Mikolov et al., 2013a,b], GLOVE [Pennington et al., 2014].
  - Problems with polysemous words: $f(\text{bank})$.

- *Contextualized* embedding of a word $i \in \mathcal{W}$ in a sentence $j_1, \ldots, j_{l-1}, i, j_{l+1}, \ldots, j_k$:

$$\mathbf{w}_i = g(j_1, \ldots, j_{l-1}, j_{l+1}, \ldots, j_k), \quad \mathbf{w}_i \in \mathbb{R}^d.$$

  - ELMO [Peters et al., 2018], BERT [Devlin et al., 2019].

Static vs Contextualized Word Vectors

Vocabulary $\mathcal{W} := \{1, \ldots, n\}$.

- *Static* embedding of a word $i \in \mathcal{W}$:

$$\mathbf{w}_i = f(i), \quad \mathbf{w}_i \in \mathbb{R}^d.$$

  - WORD2VEC [Mikolov et al., 2013a,b], GLOVE [Pennington et al., 2014].
  - Problems with polysemous words: $f(\text{bank})$.

- *Contextualized* embedding of a word $i \in \mathcal{W}$ in a sentence $j_1, \ldots, j_{l-1}, i, j_{l+1}, \ldots, j_k$:

$$\mathbf{w}_i = g(j_1, \ldots, j_{l-1}, j_{l+1}, \ldots, j_k), \quad \mathbf{w}_i \in \mathbb{R}^d.$$

  - ELMO [Peters et al., 2018], BERT [Devlin et al., 2019].
  - $g(\text{financial}, \text{crisis}) \neq g(\text{river})$.

Advantages of Static Embeddings

- Trained much faster

Advantages of Static Embeddings

- Trained much faster: few hours vs few days

Advantages of Static Embeddings

- Trained much faster: few hours vs few days
- Require less computing resources

Advantages of Static Embeddings

- Trained much faster: few hours vs few days
- Require less computing resources: 1 GPU vs 8–16 GPUs

Advantages of Static Embeddings

- Trained much faster: few hours vs few days
- Require less computing resources: 1 GPU vs 8–16 GPUs
- Plenty of theoretical research: Levy and Goldberg [2014], Arora et al. [2016], Hashimoto et al. [2016], Gittens et al. [2017], Tian et al. [2017], Ethayarajh et al. [2019], Allen et al. [2019], Allen and Hospedales [2019], Assylbekov and Takhanov [2019], Zobnin and Elistratova [2019]

Advantages of Static Embeddings

- Trained much faster: few hours vs few days
- Require less computing resources: 1 GPU vs 8–16 GPUs
- Plenty of theoretical research: Levy and Goldberg [2014], Arora et al. [2016], Hashimoto et al. [2016], Gittens et al. [2017], Tian et al. [2017], Ethayarajh et al. [2019], Allen et al. [2019], Allen and Hospedales [2019], Assylbekov and Takhanov [2019], Zobnin and Elistratova [2019]
- Integral part of contextualized models

Research Question

- Arora et al. [2016], Assylbekov and Takhanov [2019] assume that
$$\mathbf{w}_i \overset{\text{i.i.d.}}{\sim} \text{Isotropic distribution, e.g. } \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

Research Question

- Arora et al. [2016], Assylbekov and Takhanov [2019] assume that
$$\mathbf{w}_i \overset{\text{i.i.d.}}{\sim} \text{Isotropic distribution, e.g. } \mathcal{N}(\mathbf{0}, \mathbf{I}).$$
- BUT! Word vectors are NOT independent:

$$\mathbf{w}_{\text{king}} - \mathbf{w}_{\text{man}} + \mathbf{w}_{\text{woman}} \approx \mathbf{w}_{\text{queen}}$$

Research Question

- Arora et al. [2016], Assylbekov and Takhanov [2019] assume that
$$\mathbf{w}_i \overset{\text{i.i.d.}}{\sim} \text{Isotropic distribution, e.g. } \mathcal{N}(\mathbf{0}, \mathbf{I}).$$
- BUT! Word vectors are NOT independent:
$$\mathbf{w}_{\text{king}} - \mathbf{w}_{\text{man}} + \mathbf{w}_{\text{woman}} \approx \mathbf{w}_{\text{queen}}$$
- Can we impose a more realistic mathematical structure on the set of word vectors?

Notation

$\mathbf{x}$	vector
x	scalar
$\langle \mathbf{x}, \mathbf{y} \rangle$	Euclidean inner product
$\mathbf{A} = (a_{ij})$	matrix with ij-th entry a_{ij}
i.i.d.	independent and identically distributed
$\propto$	proportional to
$\sim$	distributed as
$\mathbf{w}_i$	vector for a center word $i \in \mathcal{W}$
$\mathbf{c}_j$	vector for a context word $j \in \mathcal{W}$
$\{(i, j)\}$	dataset of co-occurence pairs (i, j)
$\#(i, j)$	number of times i and j co-occur
N	dataset size: $N = \sum_{(i,j) \in \mathcal{W}^2} \#(i, j)$

Notation

$\mathbf{x}$	vector
x	scalar
$\langle \mathbf{x}, \mathbf{y} \rangle$	Euclidean inner product
$\mathbf{A} = (a_{ij})$	matrix with ij-th entry a_{ij}
i.i.d.	independent and identically distributed
$\propto$	proportional to
$\sim$	distributed as
$\mathbf{w}_i$	vector for a center word $i \in \mathcal{W}$
$\mathbf{c}_j$	vector for a context word $j \in \mathcal{W}$
$\{(i,j)\}$	dataset of co-occurence pairs (i,j)
$\#(i,j)$	number of times i and j co-occur
N	dataset size: $N = \sum_{(i,j) \in \mathcal{W}^2} \#(i,j)$

the cat sat on the mat $\rightarrow$
(*the*, *cat*), (*cat*, *the*), (*cat*, *sat*), (*sat*, *cat*), (*sat*, *on*), (*on*, *sat*), (*on*, *the*), (*the*, *on*), (*the*, *mat*), (*mat*, *the*)

SGNS as Matrix Factorization

WORD2VEC SGNS [Mikolov et al., 2013a,b] solves

$$\sum_{i \in \mathcal{W}} \sum_{j \in \mathcal{W}} \#(i,j) \left(\log \sigma(\langle \mathbf{w}_i, \mathbf{c}_j \rangle) + k \cdot \mathbb{E}_{j' \sim p} [\log \sigma(-\langle \mathbf{w}_i, \mathbf{c}_{j'} \rangle)] \right) \to \max_{\{\mathbf{w}_i\}, \{\mathbf{c}_j\}} \tag{1}$$

SGNS as Matrix Factorization

WORD2VEC SGNS [Mikolov et al., 2013a,b] solves

$$\sum_{i \in \mathcal{W}} \sum_{j \in \mathcal{W}} \#(i,j) \left(\log \sigma(\langle \mathbf{w}_i, \mathbf{c}_j \rangle) + k \cdot \mathbb{E}_{j' \sim p}[\log \sigma(-\langle \mathbf{w}_i, \mathbf{c}_{j'} \rangle)]\right) \to \max_{\{\mathbf{w}_i\},\{\mathbf{c}_j\}} \tag{1}$$

Levy and Goldberg [2014]:

$$(1) \quad \Leftrightarrow \quad \underbrace{\log \frac{p(i,j)}{p(i)p(j)}}_{\text{PMI}_{ij}} - \log k \approx \langle \mathbf{w}_i, \mathbf{c}_j \rangle$$

Modified SGNS and BPMI factorization

Assylbekov and Jangeldin [2020]: Solving

$$\sum_{i\in\mathcal{W}}\sum_{j\in\mathcal{W}}\#(i,j)\left(\log\langle\mathbf{w}_i,\mathbf{c}_j\rangle+\mathbb{E}_{j'\sim p}[\log(1-\langle\mathbf{w}_i,\mathbf{c}_{j'}\rangle)]\right)\to\max_{\{\mathbf{w}_i\},\{\mathbf{c}_j\}} \tag{2}$$

gives word embeddings comparable to SGNS.[1]

[1] .649 vs .678 on WordSim353 task

Modified SGNS and BPMI factorization

Assylbekov and Jangeldin [2020]: Solving

$$\sum_{i\in\mathcal{W}}\sum_{j\in\mathcal{W}} \#(i,j)\left(\log\langle\mathbf{w}_i,\mathbf{c}_j\rangle + \mathbb{E}_{j'\sim p}[\log(1-\langle\mathbf{w}_i,\mathbf{c}_{j'}\rangle)]\right) \to \max_{\{\mathbf{w}_i\},\{\mathbf{c}_j\}} \tag{2}$$

gives word embeddings comparable to SGNS.[1] Also,

$$(1) \quad \Leftrightarrow \quad \langle\mathbf{w}_i,\mathbf{c}_j\rangle \approx H\left(\log\frac{p(i,j)}{p(i)p(j)}\right),$$

where $H(x) = \begin{cases} 1 \text{ for } x > 0 \\ 0 \text{ for } x \leq 0 \end{cases}$

[1] .649 vs .678 on WordSim353 task

BPMI & Hyperbolic Geometry

- BPMI is an *adjacency* matrix of some graph

BPMI & Hyperbolic Geometry

- BPMI is an *adjacency* matrix of some graph
- Such graph is a *complex network* (explained later)

BPMI & Hyperbolic Geometry

- BPMI is an *adjacency* matrix of some graph
- Such graph is a *complex network* (explained later)
- Krioukov et al. [2010]: Complex network possesses an effective *hyperbolic geometry* underneath.

BPMI & Hyperbolic Geometry

- BPMI is an *adjacency* matrix of some graph
- Such graph is a *complex network* (explained later)
- Krioukov et al. [2010]: Complex network possesses an effective *hyperbolic geometry* underneath.

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

BPMI & Hyperbolic Geometry

- BPMI is an *adjacency* matrix of some graph
- Such graph is a *complex network* (explained later)
- Krioukov et al. [2010]: Complex network possesses an effective *hyperbolic geometry* underneath.

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

Can we go from the final point to the starting one?

Hyperbolic geometry

- Curvature κ:

Hyperbolic geometry

- Curvature κ:
  - $\kappa = 0$ — Euclidean geometry $\mathbb{R}^2$
  - $\kappa > 0$ — Spherical geometry $\mathcal{S}^2$
  - $\kappa < 0$ — Hyperbolic geometry $\mathbb{H}^2$

Hyperbolic geometry

- Curvature κ:
  - $\kappa = 0$ — Euclidean geometry $\mathbb{R}^2$
  - $\kappa > 0$ — Spherical geometry $\mathcal{S}^2$
  - $\kappa < 0$ — Hyperbolic geometry $\mathbb{H}^2$
- $\mathbb{H}^2$ **cannot** be isometrically embedded into $\mathbb{R}^n$ ($\forall n$):

Hyperbolic geometry

- Curvature κ:
  - $\kappa = 0$ — Euclidean geometry $\mathbb{R}^2$
  - $\kappa > 0$ — Spherical geometry $\mathcal{S}^2$
  - $\kappa < 0$ — Hyperbolic geometry $\mathbb{H}^2$
- $\mathbb{H}^2$ **cannot** be isometrically embedded into $\mathbb{R}^n$ ($\forall n$):
  - we cannot map points of $\mathbb{H}^2$ into points of $\mathbb{R}^n$ in such way that the distances between points are preserved.

Hyperbolic geometry

- Curvature κ:
  - $\kappa = 0$ — Euclidean geometry $\mathbb{R}^2$
  - $\kappa > 0$ — Spherical geometry $\mathcal{S}^2$
  - $\kappa < 0$ — Hyperbolic geometry $\mathbb{H}^2$
- $\mathbb{H}^2$ **cannot** be isometrically embedded into $\mathbb{R}^n$ ($\forall n$):
  - we cannot map points of $\mathbb{H}^2$ into points of $\mathbb{R}^n$ in such way that the distances between points are preserved.
  - Many equivalent models of $\mathbb{H}^d$, e.g.:

Hyperbolic geometry

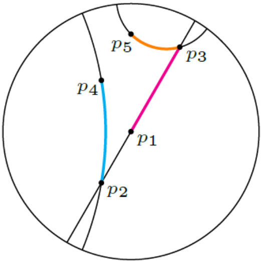
- Curvature κ:
  - $\kappa = 0$ — Euclidean geometry $\mathbb{R}^2$
  - $\kappa > 0$ — Spherical geometry $\mathcal{S}^2$
  - $\kappa < 0$ — Hyperbolic geometry $\mathbb{H}^2$
- $\mathbb{H}^2$ **cannot** be isometrically embedded into $\mathbb{R}^n$ $(\forall n)$:
  - we cannot map points of $\mathbb{H}^2$ into points of $\mathbb{R}^n$ in such way that the distances between points are preserved.
  - Many equivalent models of $\mathbb{H}^d$, e.g.:
    - Hyperboloid model
    - Poincaré model
    - Upper half-space model

Hyperbolic geometry

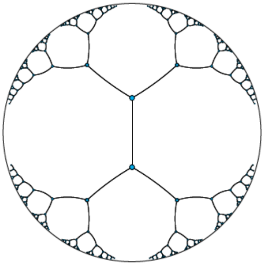
- Curvature κ:
  - $\kappa = 0$ — Euclidean geometry $\mathbb{R}^2$
  - $\kappa > 0$ — Spherical geometry $\mathcal{S}^2$
  - $\kappa < 0$ — Hyperbolic geometry $\mathbb{H}^2$
- $\mathbb{H}^2$ **cannot** be isometrically embedded into $\mathbb{R}^n$ $(\forall n)$:
  - we cannot map points of $\mathbb{H}^2$ into points of $\mathbb{R}^n$ in such way that the distances between points are preserved.
  - Many equivalent models of $\mathbb{H}^d$, e.g.:
    - Hyperboloid model
    - Poincaré model
    - Upper half-space model
- We'll use the so-called *native model* [Krioukov et al., 2010].

Native model of $\mathbb{H}^2$

Interior of the Euclidean disk of radius R:



(a) Geodesics of the Poincaré disk

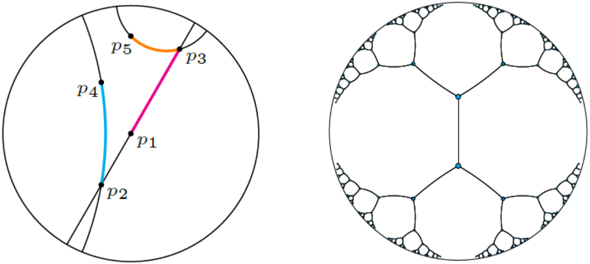


(b) Embedding of a tree in $\mathcal{B}^2$

2

Native model of $\mathbb{H}^2$

Interior of the Euclidean disk of radius R:



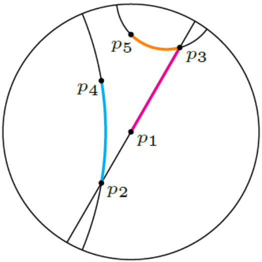
(a) Geodesics of the Poincaré disk (b) Embedding of a tree in $\mathcal{B}^2$

- if (r, θ) are polar coordinates of $p \in \mathbb{H}^2$, then $r =$ hyperbolic distance of p from the origin.

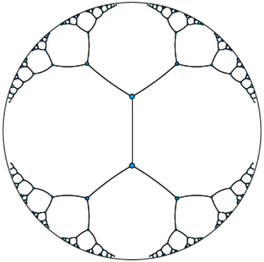
2

Native model of $\mathbb{H}^2$

Interior of the Euclidean disk of radius R:



(a) Geodesics of the Poincaré disk



(b) Embedding of a tree in $\mathcal{B}^2$

- if (r, θ) are polar coordinates of $p \in \mathbb{H}^2$, then r = hyperbolic distance of p from the origin.
- distance x between two points $p = (r, \theta)$ and $p' = (r', \theta')$ satisfies[2]

$$\cosh x = \cosh r \cosh r' - \sinh r \sinh r' \cos(\theta - \theta'). \qquad (3)$$

[2] for curvature $\kappa = -1$

Euclidean vs Hyperbolic geometries

Property	Euclidean	Hyperbolic[3]
Parallel lines	1	∞
Shape of triangles		
Sum of angles in triangles	π	$< \pi$
Circle length	$2\pi r$	$2\pi \sinh r = O(e^r)$
Disk area	πr^2	$2\pi(\cosh r - 1) = O(e^r)$

[3] $\kappa = -1$

Random Hyperbolic Graph (RHG)

Construction by Krioukov et al. [2010]:

- place randomly n points (nodes) into a hyperbolic disk of radius R

Random Hyperbolic Graph (RHG)

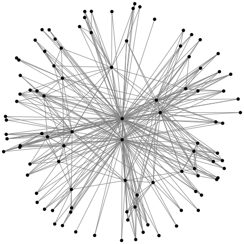
Construction by Krioukov et al. [2010]:

- place randomly n points (nodes) into a hyperbolic disk of radius R
- connect those pairs of points (i, j) for which $x_{ij} \leq R$.

Random Hyperbolic Graph (RHG)

Construction by Krioukov et al. [2010]:

- place randomly n points (nodes) into a hyperbolic disk of radius R
- connect those pairs of points (i, j) for which $x_{ij} \leq R$.



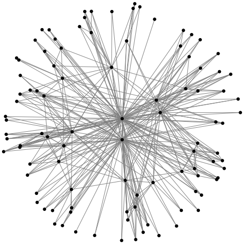
$$\theta \sim \mathcal{U}[0, 2\pi)$$

$$r \sim \rho(r) := \frac{\alpha \sinh \alpha r}{\cosh \alpha R - 1}, \ \alpha \in (0, 1)$$

Random Hyperbolic Graph (RHG)

Construction by Krioukov et al. [2010]:

- place randomly n points (nodes) into a hyperbolic disk of radius R
- connect those pairs of points (i, j) for which $x_{ij} \leq R$.



$$\theta \sim \mathcal{U}[0, 2\pi)$$

$$r \sim \rho(r) := \frac{\alpha \sinh \alpha r}{\cosh \alpha R - 1}, \ \alpha \in (0, 1)$$

- R and α are chosen to fit the RHG degree distribution to that of BPMI.

Graph Terminology

- Graph: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
  - $\mathcal{V}$ – set of vertices $\{i\}$.
  - $\mathcal{E}$ – set of pairs (i, j) with $i, j \in \mathcal{V}$

Graph Terminology

- Graph: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
  - $\mathcal{V}$ – set of vertices $\{i\}$.
  - $\mathcal{E}$ – set of pairs (i, j) with $i, j \in \mathcal{V}$
- Adjacency matrix (e_{ij}) with $e_{ij} = \begin{cases} 1 & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{if } (i,j) \notin \mathcal{E} \end{cases}$.

Graph Terminology

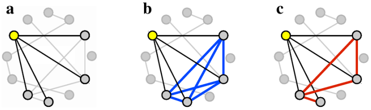
- Graph: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
  - $\mathcal{V}$ – set of vertices $\{i\}$.
  - $\mathcal{E}$ – set of pairs (i, j) with $i, j \in \mathcal{V}$
- Adjacency matrix (e_{ij}) with $e_{ij} = \begin{cases} 1 & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{if } (i,j) \notin \mathcal{E} \end{cases}$.
- *Degree* of a vertex: $\deg(i) = \sum_{j \in \mathcal{V}} e_{ij}$

Graph Terminology

- Graph: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
  - $\mathcal{V}$ – set of vertices $\{i\}$.
  - $\mathcal{E}$ – set of pairs (i, j) with $i, j \in \mathcal{V}$
- Adjacency matrix (e_{ij}) with $e_{ij} = \begin{cases} 1 & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{if } (i,j) \notin \mathcal{E} \end{cases}$.
- *Degree* of a vertex: $\deg(i) = \sum_{j \in \mathcal{V}} e_{ij}$
- Let $\mathcal{G}_i = \{j \in \mathcal{V} \mid e_{ij} = 1\}$ — the set of nearest neighbors of a vertex i, and $l_i = \sum_{j \in \mathcal{V}} e_{ij} \left[\sum_{k \in \mathcal{G}_i;\ j<k} e_{jk} \right]$

Graph Terminology

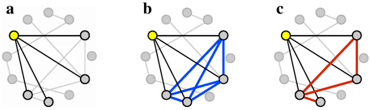
- Graph: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
  - $\mathcal{V}$ – set of vertices $\{i\}$.
  - $\mathcal{E}$ – set of pairs (i, j) with $i, j \in \mathcal{V}$
- Adjacency matrix (e_{ij}) with $e_{ij} = \begin{cases} 1 & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{if } (i,j) \notin \mathcal{E} \end{cases}$.
- *Degree* of a vertex: $\deg(i) = \sum_{j \in \mathcal{V}} e_{ij}$
- Let $\mathcal{G}_i = \{j \in \mathcal{V} \mid e_{ij} = 1\}$ — the set of nearest neighbors of a vertex i, and $l_i = \sum_{j \in \mathcal{V}} e_{ij} \left[\sum_{k \in \mathcal{G}_i;\ j<k} e_{jk} \right]$
- Local clustering coefficient $C(i) = \frac{l_i}{\binom{|\mathcal{G}_i|}{2}}$



Reference node has 4 neighbors — 6 possible links between neighbors — 4 existing links between neighbors

Graph Terminology

- Graph: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
  - $\mathcal{V}$ – set of vertices $\{i\}$.
  - $\mathcal{E}$ – set of pairs (i, j) with $i, j \in \mathcal{V}$
- Adjacency matrix (e_{ij}) with $e_{ij} = \begin{cases} 1 & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{if } (i,j) \notin \mathcal{E} \end{cases}$.
- *Degree* of a vertex: $\deg(i) = \sum_{j \in \mathcal{V}} e_{ij}$
- Let $\mathcal{G}_i = \{j \in \mathcal{V} \mid e_{ij} = 1\}$ — the set of nearest neighbors of a vertex i, and $l_i = \sum_{j \in \mathcal{V}} e_{ij} \left[\sum_{k \in \mathcal{G}_i;\ j<k} e_{jk} \right]$
- Local clustering coefficient $C(i) = \frac{l_i}{\binom{|\mathcal{G}_i|}{2}}$



Reference node has 4 neighbors | 6 possible links between neighbors | 4 existing links between neighbors

- *Clustering coefficient* $C = \frac{1}{n} \sum_{i \in \mathcal{V}} C(i)$.

Complex Networks

- $\bar{k} = \frac{1}{n} \sum_{j \in \mathcal{V}} e_{ij}$ – average degree per vertex.

Complex Networks

- $\bar{k} = \frac{1}{n} \sum_{j \in \mathcal{V}} e_{ij}$ – average degree per vertex.
- Random Graphs:

$$e_{ij} \overset{\text{iid}}{\sim} \mathrm{Bernoulli}(p)$$

Erdős and Rényi [1960] showed

$$C \approx \frac{\bar{k}}{n} \quad \text{and} \quad \deg(i) \sim \mathrm{Binomial}(n-1, p)$$

Complex Networks

- $\bar{k} = \frac{1}{n}\sum_{j \in \mathcal{V}} e_{ij}$ – average degree per vertex.
- Random Graphs:
$$e_{ij} \overset{\text{iid}}{\sim} \mathrm{Bernoulli}(p)$$
Erdős and Rényi [1960] showed
$$C \approx \frac{\bar{k}}{n} \quad \text{and} \quad \deg(i) \sim \mathrm{Binomial}(n-1, p)$$
- Complex Networks:
$$C \gg \frac{\bar{k}}{n} \quad \text{and} \quad p(\deg(i) = k) \propto \frac{1}{k^{\gamma}},$$
where γ is some constant.

Complex Networks

- $\bar{k} = \frac{1}{n} \sum_{j \in \mathcal{V}} e_{ij}$ – average degree per vertex.
- Random Graphs:

$$e_{ij} \overset{\text{iid}}{\sim} \mathrm{Bernoulli}(p)$$

  Erdős and Rényi [1960] showed

$$C \approx \frac{\bar{k}}{n} \quad \text{and} \quad \deg(i) \ \sim \ \mathrm{Binomial}(n-1, p)$$

- Complex Networks:

$$C \gg \frac{\bar{k}}{n} \quad \text{and} \quad p(\deg(i) = k) \propto \frac{1}{k^{\gamma}},$$

  where γ is some constant.
- RHG *is* a complex network.

RHG and BPMI

A — BPMI matrix:

$$\mathbf{A}_{ij} = H(\mathrm{PMI}_{ij})$$

B — adjacency matrix of the RHG:

$$\mathbf{B}_{ij} = H(R - x_{ij})$$

RHG and BPMI

A — BPMI matrix:

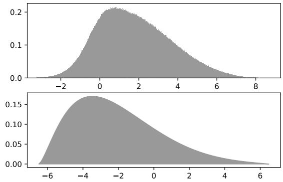
$$\mathbf{A}_{ij} = H(\mathrm{PMI}_{ij})$$

B — adjacency matrix of the RHG:

$$\mathbf{B}_{ij} = H(R - x_{ij})$$

If RHG and BPMI are structurally similar, then

$$R - x_{ij} \sim \mathrm{PMI}_{ij}$$



RHG and BPMI

A — BPMI matrix:

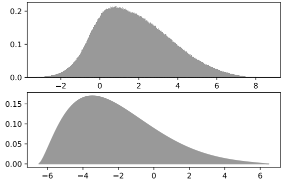
$$\mathbf{A}_{ij} = H(\mathrm{PMI}_{ij})$$

B — adjacency matrix of the RHG:

$$\mathbf{B}_{ij} = H(R - x_{ij})$$

If RHG and BPMI are structurally similar, then

$$R - x_{ij} \sim \mathrm{PMI}_{ij}$$



Can we (approximately) match RHG nodes to BPMI nodes? i.e. find a permutation matrix **P** that solves

$$\|\mathbf{A} - \mathbf{P}\mathbf{B}\mathbf{P}^\top\| \to \min_{\mathbf{P} \in \mathcal{P}_n}$$

Approximate Graph Matching

$$\|\mathbf{A} - \mathbf{P}\mathbf{B}\mathbf{P}^\top\| \to \min_{\mathbf{P} \in \mathcal{P}_n}$$

Approximate solution [Umeyama, 1988]:

1. Find eigendecompositions of **A** and **B**:

$$\mathbf{A} = \mathbf{U}_A \boldsymbol{\Lambda}_A \mathbf{U}_A^\top, \quad \mathbf{B} = \mathbf{U}_B \boldsymbol{\Lambda}_B \mathbf{U}_B^\top$$

Approximate Graph Matching

$$\|\mathbf{A} - \mathbf{P}\mathbf{B}\mathbf{P}^\top\| \to \min_{\mathbf{P} \in \mathcal{P}_n}$$

Approximate solution [Umeyama, 1988]:

1. Find eigendecompositions of $\mathbf{A}$ and $\mathbf{B}$:

$$\mathbf{A} = \mathbf{U}_A \boldsymbol{\Lambda}_A \mathbf{U}_A^\top, \quad \mathbf{B} = \mathbf{U}_B \boldsymbol{\Lambda}_B \mathbf{U}_B^\top$$

2. $\widetilde{\mathbf{P}} := |\mathbf{U}_A||\mathbf{U}_B|^\top$

Approximate Graph Matching

$$\|\mathbf{A} - \mathbf{P}\mathbf{B}\mathbf{P}^\top\| \to \min_{\mathbf{P} \in \mathcal{P}_n}$$

Approximate solution [Umeyama, 1988]:

1. Find eigendecompositions of **A** and **B**:

$$\mathbf{A} = \mathbf{U}_A \boldsymbol{\Lambda}_A \mathbf{U}_A^\top, \quad \mathbf{B} = \mathbf{U}_B \boldsymbol{\Lambda}_B \mathbf{U}_B^\top$$

2. $\widetilde{\mathbf{P}} := |\mathbf{U}_A||\mathbf{U}_B|^\top$
3. To obtain a permutation matrix $\mathbf{P}$ from $\widetilde{\mathbf{P}}$ we apply the Auction algorithm of Bertsekas [1979].

Word Embeddings from RHG

Word embedding matrix $\mathbf{W}$ can be obtained from $\mathbf{PBP}^\top$ by

1. SVD:

$$\mathbf{PBP}^\top = \mathbf{U\Sigma V}.$$

Word Embeddings from RHG

Word embedding matrix $\mathbf{W}$ can be obtained from $\mathbf{PBP}^\top$ by

1. SVD:
$$\mathbf{PBP}^\top = \mathbf{U\Sigma V}.$$
2. As in Levy and Goldberg [2014]:
$$\mathbf{W} := \mathbf{U}_{1:n,1:d}\mathbf{\Sigma}^{1/2}_{1:d,1:d}.$$

Evaluation

- Dataset: `text8`

Evaluation

- Dataset: `text8`
- Ignore words that appeared less than 500 times

Evaluation

- Dataset: `text8`
- Ignore words that appeared less than 500 times
- Vocabulary: 3,446 tokens

Evaluation

- Dataset: `text8`
- Ignore words that appeared less than 500 times
- Vocabulary: 3,446 tokens
- Evaluation: word similarity task WS353

Evaluation

- Dataset: `text8`
- Ignore words that appeared less than 500 times
- Vocabulary: 3,446 tokens
- Evaluation: word similarity task WS353

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

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

Evaluation

- Dataset: `text8`
- Ignore words that appeared less than 500 times
- Vocabulary: 3,446 tokens
- Evaluation: word similarity task WS353

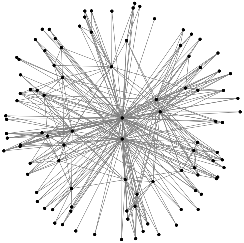
	Overall	Similarity	Relatedness
SGNS	.669	.767	.661
PMI + SVD	.432	.498	.433
BPMI + SVD	.362	.432	.322
RHG + Permute + SVD	.263	.254	.246

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

Bad quality of word embeddings from RHG.

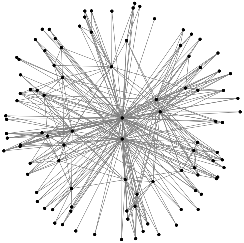
Conclusion

- Throwing points randomly in hyperbolic disk, we get word representations.



Conclusion

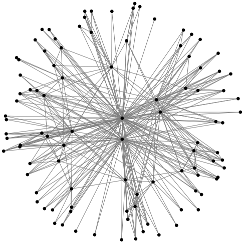
- Throwing points randomly in hyperbolic disk, we get word representations.



- Each point corresponds to a word of human language.

Conclusion

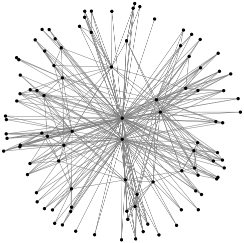
- Throwing points randomly in hyperbolic disk, we get word representations.



- Each point corresponds to a word of human language.
- Relation $\approx$ Hyperbolic distance.

Conclusion

- Throwing points randomly in hyperbolic disk, we get word representations.



- Each point corresponds to a word of human language.
- Relation $\approx$ Hyperbolic distance.
- Semiotic arbitrariness [De Saussure, 2011]:

```
What's in a name? That which we call a rose
By any other name would smell as sweet.
```

References I

Carl Allen and Timothy Hospedales. Analogies explained: Towards understanding word embeddings. In *International Conference on Machine Learning*, pages 223–231, 2019.

Carl Allen, Ivana Balazevic, and Timothy Hospedales. What the vec? towards probabilistically grounded embeddings. In *Advances in Neural Information Processing Systems*, pages 7465–7475, 2019.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. A latent variable model approach to pmi-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399, 2016.

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

References II

Zhenisbek Assylbekov and Rustem Takhanov. Context vectors are reflections of word vectors in half the dimensions. *Journal of Artificial Intelligence Research*, 66:225–242, 2019.

Dimitri P Bertsekas. A distributed algorithm for the assignment problem. *Lab. for Information and Decision Systems Working Paper, MIT*, 1979.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

Paul Erdős and Alfréd Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1):17–60, 1960.

References III

Kawin Ethayarajh, David Duvenaud, and Graeme Hirst. Towards understanding linear word analogies. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3253–3262, 2019.

Alex Gittens, Dimitris Achlioptas, and Michael W Mahoney. Skip-gram- zipf+ uniform= vector additivity. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 69–76, 2017.

Tatsunori B Hashimoto, David Alvarez-Melis, and Tommi S Jaakkola. Word embeddings as metric recovery in semantic spaces. *Transactions of the Association for Computational Linguistics*, 4:273–286, 2016.

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

References IV

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

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

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

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of EMNLP*, pages 1532–1543, 2014.

References V

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237, 2018.

Ran Tian, Naoaki Okazaki, and Kentaro Inui. The mechanism of additive composition. *Machine Learning*, 106(7):1083–1130, 2017.

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

Alexey Zobnin and Evgenia Elistratova. Learning word embeddings without context vectors. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 244–249, 2019.