Introduction to representation learning in hyperbolic space

Atsushi Suzuki

Lecturer, King's College London

Take home message

If your data have a hierarchical or tree-like structure, hyperbolic space might help you.

We anticipate that, at the end of this lecture, you will be able to

- Explain hyperbolic space's exponential space growth property,
- Judge whether hyperbolic space can contribute to applications that you are interested in, and
- Have clues to design a machine-learning model using hyperbolic space.

Outline

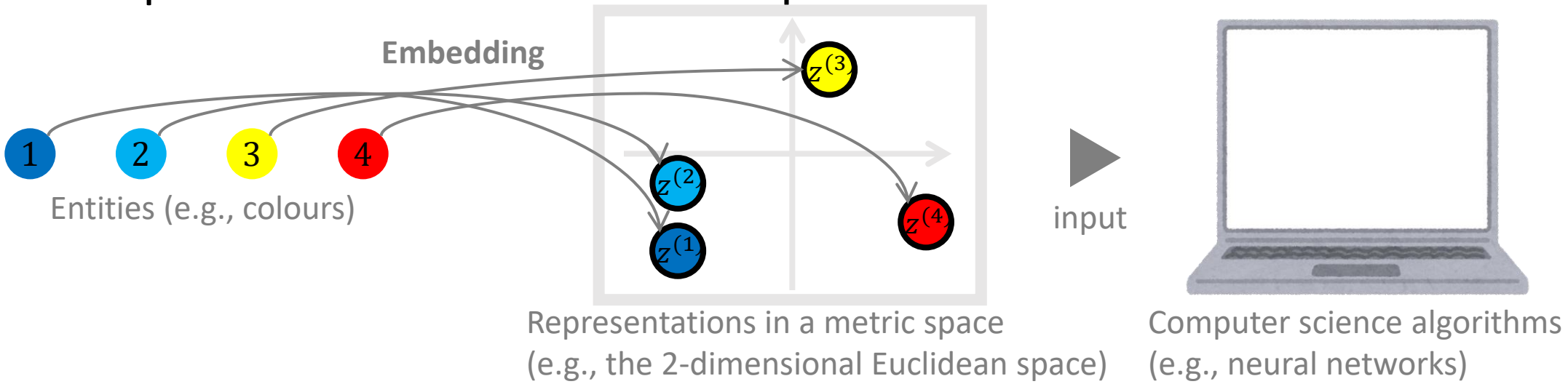
- Representation learning's motivation
- Why non-Euclidean, hyperbolic?
- How does hyperbolic space look like?
- Representation learning models using hyperbolic space

Outline

- **Representation learning's motivation**
- Why non-Euclidean, hyperbolic?
- How does hyperbolic space look like?
- Representation learning models using hyperbolic space

Embedding maps entities into points in metric space

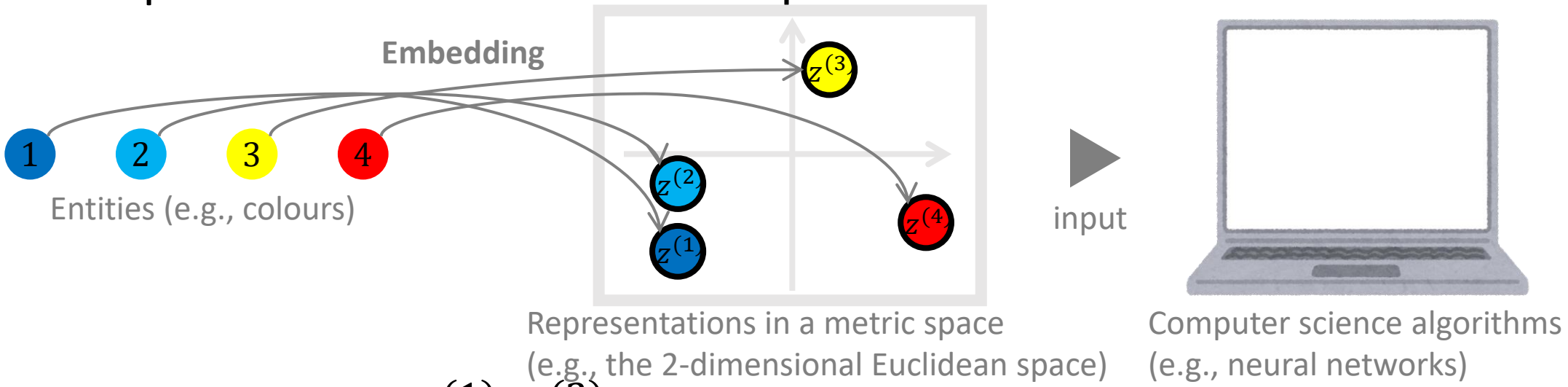
- Entities can be anything: e.g., people, cars, words, colours, etc.
- Computers need mathematical representations to handle entities.



- Embedding: obtaining representations of entities in a metric space
  - Notation: $z^{(i)}$ is the representation of entity i.

Embedding maps entities into points in metric space

- Entities can be anything: e.g., people, cars, words, colours, etc.
- Computers need mathematical representations to handle entities.



- Representations $z^{(1)}, z^{(2)}, \ldots$ are usually in ***metric space***
  - so we can measure the distance between entities

Notation

- $(\text{left}) \coloneqq (\text{right})$ means we define (left) by (right).
  - Examples: $\mathrm{i} \coloneqq \sqrt{-1}, \phi \coloneqq \frac{1+\sqrt{5}}{2}$.
- “iff” means “if and only if”
- $\ln\cdot$ is the natural logarithm.

Notation

- $\mathbb{Z}$: the set of integers. $\mathbb{Z} = \{0, \pm 1, \pm 2, \dots\}$.
- $\mathbb{Z}_{>0}$: the set of positive integers. $\mathbb{Z} = \{1,2, \dots\}$.
- $\mathbb{R}$: the set of real numbers.
  - Examples: $0, \sqrt{2}, \pi, \phi, -\frac{22}{7} \in \mathbb{R}$, but $\mathrm{i} \notin \mathbb{R}$.
- $\mathbb{R}_{\geq 0}$: the set of nonnegative real numbers.
  - Examples: $0, \pi, \phi \in \mathbb{R}_{\geq 0}$, but $-\frac{22}{7}, \mathrm{i} \notin \mathbb{R}_{\geq 0}$.
- $\mathbb{R}_{>0}$: the set of positive real numbers.
  - Examples: $\pi, \phi \in \mathbb{R}_{>0}$, but $0, -\frac{22}{7}, \mathrm{i} \notin \mathbb{R}_{>0}$

Notation (vectors)

- $\mathbb{R}^D$: the set of D-dimensional real vectors (for $D \in \mathbb{Z}_{>0}$).
  - Examples, $\begin{bmatrix} 2 \\ -1 \end{bmatrix} \in \mathbb{R}^2$, $\begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix} \in \mathbb{R}^3$.
  - Note: we denote most vectors as column vectors with square brackets.
- A real vector is denoted by a bold lower letter, e.g., $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$.
- The transpose of a vector is denoted by $\top$.
  - E.g., if $\boldsymbol{x} = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}$, then $\boldsymbol{x}^\top = [2 \quad -1 \quad 3]$.

Notation (maps, functions)

Let A and B be sets.

- $f: A \to B$ means that f is a map from the set A to the set B.
  - That is, $f(a)$ is defined iff $a \in A$ and $f(a) \in B$.
  - Examples: exp: $\mathbb{R} \to \mathbb{R}_{>0}$, $(\cdot)^2: \mathbb{R} \to \mathbb{R}_{\geq 0}$.
  - If B is a subset of $\mathbb{R}$, the map f is called a (real-valued) ***function*** on A.
- $A \times B$ is the direct product of A and B, the set of pairs of an element in A and an element in B.
  - That is, $A \times B = \{ (a, b) \mid a \in A, b \in B \}$.
- Hence, $f: A \times B \to \mathbb{R}$ means that f is a real-valued bivariate function that takes an element in A and an element in B as inputs.

Metric space (賦距空間) is a point set $\mathcal{Z}$ equipped with a distance function Δ

A metric space is denoted by the pair $(\mathcal{Z}, \Delta)$, where

- $\mathcal{Z}$ is a point set, and
- $\Delta: \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}_{\geq 0}$ is a ***distance function*** on $\mathcal{Z}$.

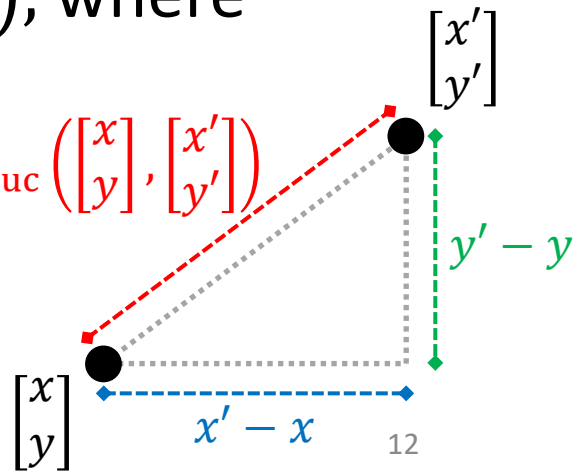
For $z, z' \in \mathcal{Z}$, the value $\Delta(z, z')$ is called the ***distance*** between z and z'.

In metric space, we can discuss whether two points are close or distant.

Example: the 2-dimensional Euclidean space $\left(\mathbb{R}^2, \Delta_{\mathbb{R}^2,\mathrm{Euc}}\right)$, where

- $\mathbb{R}^2$: the set of 2-dimensional real vectors.
- $\Delta_{\mathbb{R}^2,\mathrm{Euc}}: \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}_{\geq 0}$: the distance function defined by

$$\Delta_{\mathbb{R}^2,\mathrm{Euc}}\left(\begin{bmatrix} x \\ y \end{bmatrix}, \begin{bmatrix} x' \\ y' \end{bmatrix}\right) := \sqrt{(x' - x)^2 + (y' - y)^2}.$$



The distance function allows us to discuss two representations' relation quantitatively

A metric space $(\mathcal{Z}, \Delta)$ has a distance function $\Delta: \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}_{\geq 0}$.

The distance function defines the distance between two points.

In the representation learning context,

- A small value of $\Delta(z^{(i)}, z^{(j)})$ implies that i and j are strongly related.
- A large value of $\Delta(z^{(i)}, z^{(j)})$ implies that i and j are weakly related.

The distance function is essential to discuss the relation between entities quantitatively.

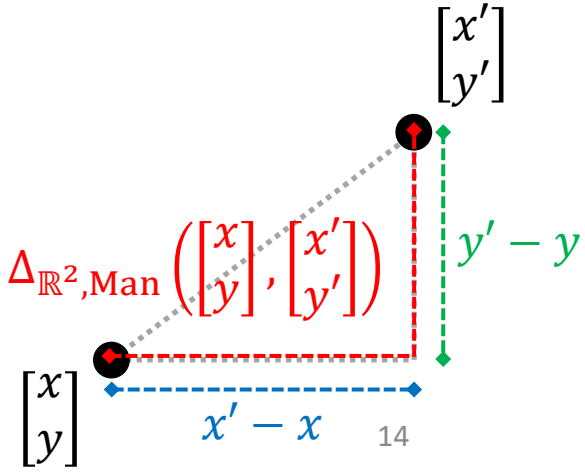
> Without the distance function, we could only discuss whether two entities are the same or not.

We can consider multiple distance functions on one point set.

Example: the Taxicab (Manhattan) geometry $\left(\mathbb{R}^2, \Delta_{\mathbb{R}^2,\mathrm{Man}}\right)$, where

- $\mathbb{R}^2$: the set of 2-dimensional real vectors.
- $\Delta_{\mathbb{R}^2,\mathrm{Man}}: \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}_{\geq 0}$: the distance function defined by

$$\Delta_{\mathbb{R}^2,\mathrm{Man}}\left(\begin{bmatrix} x \\ y \end{bmatrix}, \begin{bmatrix} x' \\ y' \end{bmatrix}\right) \coloneqq |x' - x| + |y' - y|.$$



The axiomata (公理) of a distance function

Strictly speaking, we call $(\mathcal{Z}, \Delta)$ a metric space iff for all z, z', z'',

- $\Delta(z, z') = 0 \Leftrightarrow z = z'$ (identity of indiscernibles 非退化性),
- $\Delta(z, z') \geq 0$ (nonnegativity 非負性),
- $\Delta(z, z') = \Delta(z', z)$ (symmetricity 對稱性),
- $\Delta(z, z') + \Delta(z', z'') \geq \Delta(z, z'')$ (triangle inequality 三角不等式).

Example of distance functions that satisfy the above:

- The Euclidean distance and ℓ^p-distance on real vector space
- The (geodesic) distance on a sphere
- The Hellinger distance between distributions
- The Wasserstein distance between distributions on a metric space

In the representation learning context, the axiomata are not always satisfied.

Example of distance functions that do **NOT** satisfy the axiomata:

- Squared Euclidean distance $\|\boldsymbol{z}' - \boldsymbol{z}\|_2^2$ (violates the triangle inequality)
- Negative inner product $-\boldsymbol{z}^\top \boldsymbol{z}'$(satisfies the symmetry only)
- Kullback-Leibler divergence between distributions $\mathbb{E}_P \log P(X) - \mathbb{E}_P \log P'(X)$ (satisfies the nonnegativity only)

In the representation learning context, the axiomata are not important.

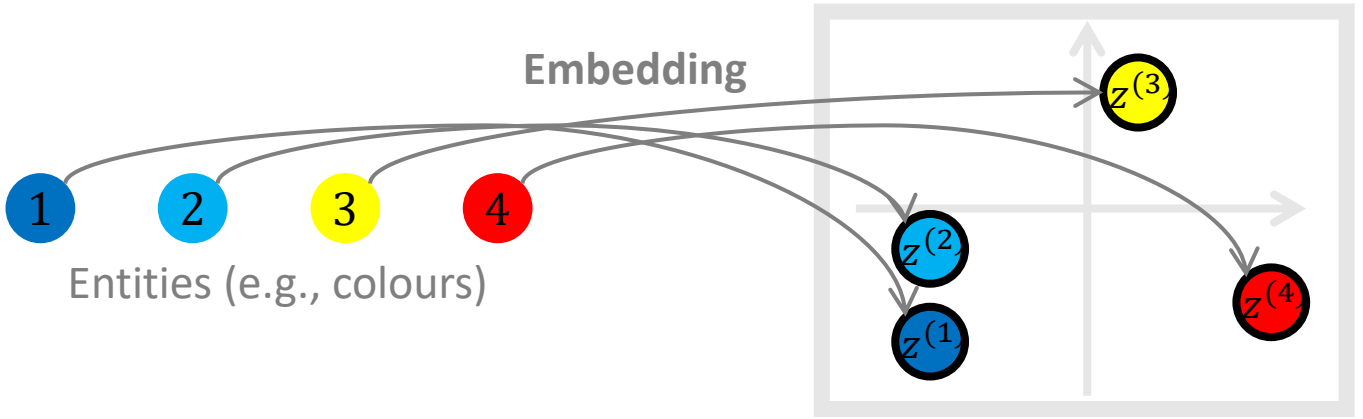
Essential is whether the following interpretation holds:

- A small distance ⇒ similar, related, close
- A large distance ⇒ dissimilar, irrelated, far

The objective of representation learning

Representation learning (embedding) on $(\mathcal{Z}, \Delta)$ aims to get representations $z^{(1)}, z^{(2)}, \ldots, z^{(N)} \in \mathcal{Z}$ that satisfy

- i and j are closely related $\Leftrightarrow \Delta_{\mathcal{Z}}\left(z^{(i)}, z^{(j)}\right)$ is small,
- i and j are NOT closely related $\Leftrightarrow \Delta_{\mathcal{Z}}\left(z^{(i)}, z^{(j)}\right)$ is large.



Application of embedding 1: visualisation [Nickel & Kiela, NeurIPS, 2017]

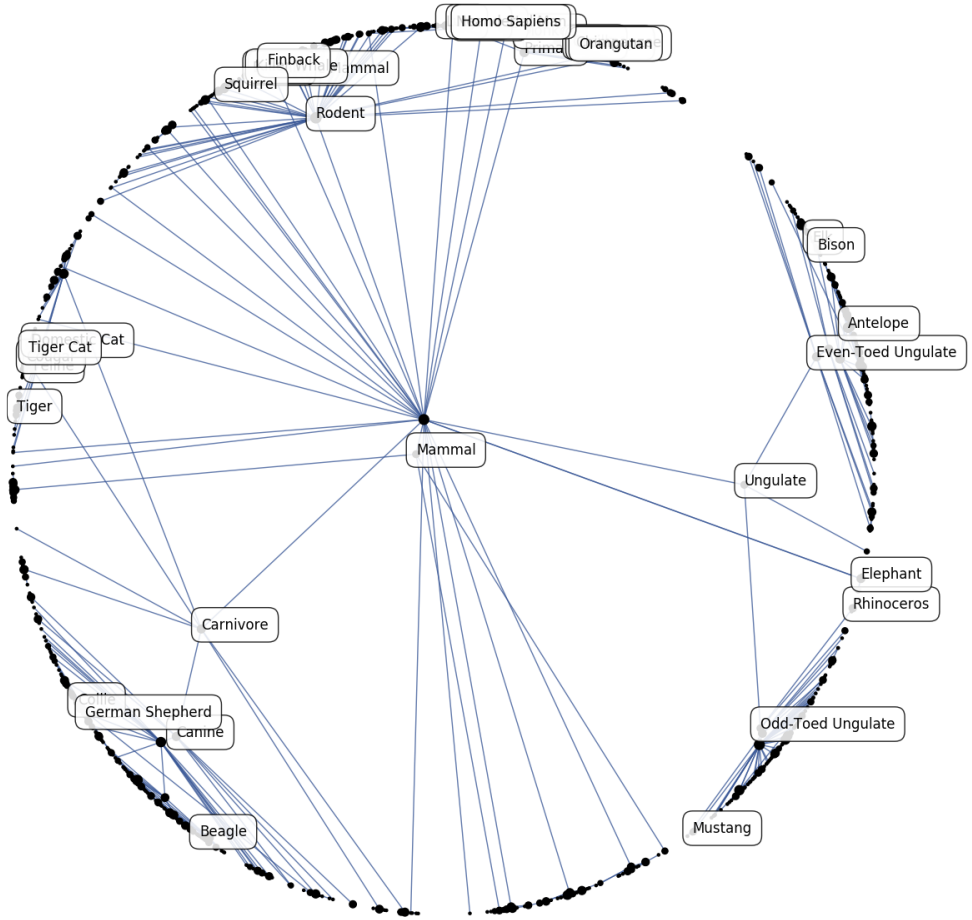


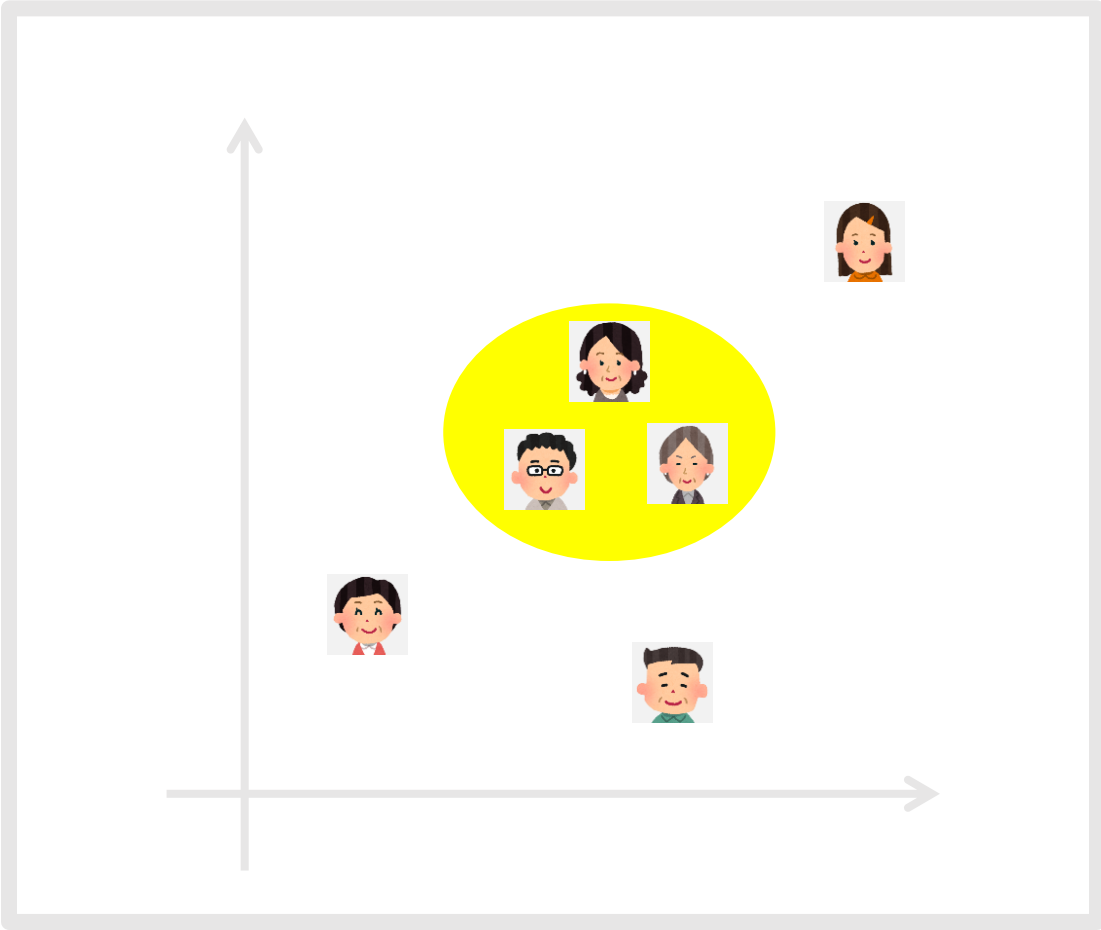
Figure is from [Nickel & Kiela, NeurIPS, 2017]

For visualisation, the representation space's dimension should be
2 or 3.

Application of embedding 2: recommendation (finding unknown links)

We can recommend them to one another if their representations are located close.

For recommendation, representations in a low-dimensional space is preferrable to save computational cost.



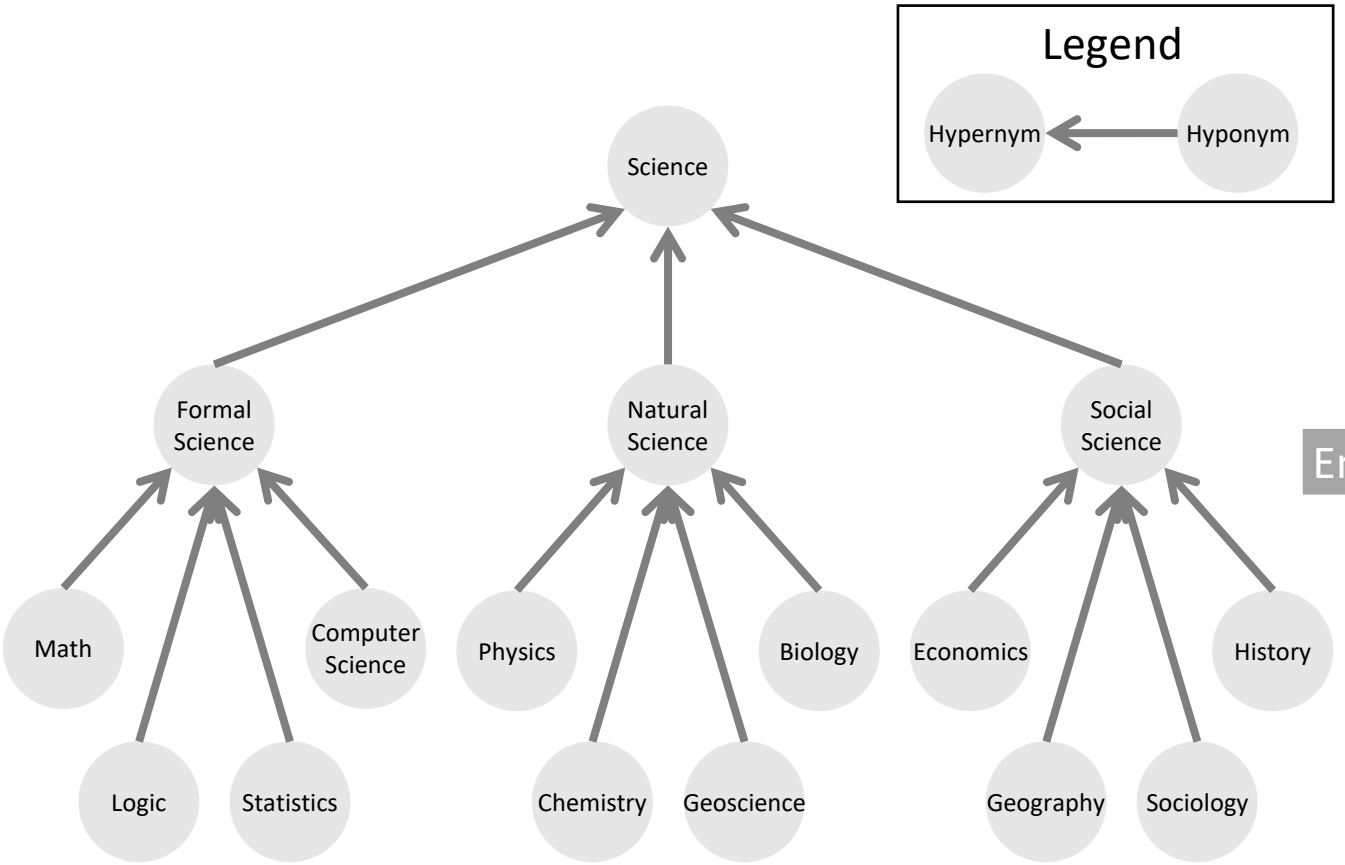
Outline

- **Representation learning's motivation**
  - **To obtain entities' representations that reflect the relation among them.**
- Why non-Euclidean, hyperbolic?
- How does hyperbolic space look like?
- Representation learning models using hyperbolic space

Outline

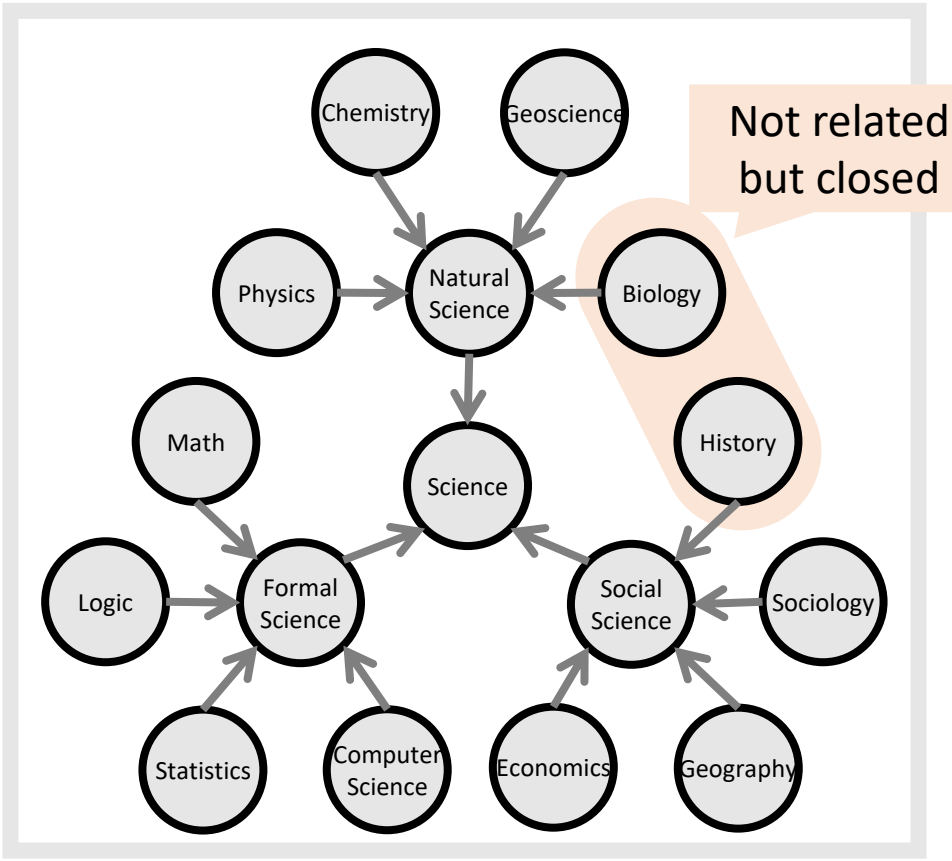
- ~~Representation learning's motivation~~
- **Why non-Euclidean, hyperbolic?**
- How does hyperbolic space look like?
- Representation learning models using hyperbolic space

Euclidean space is NOT suitable for handling hierarchical data.



Hierarchical data
(e.g., hypernym-hyponym relation in natural language words)

Embedding



Bad representations
(e.g., word embedding in $(\mathbb{R}^2, \Delta_{\mathbb{R}^2,\mathrm{Euc}})$)

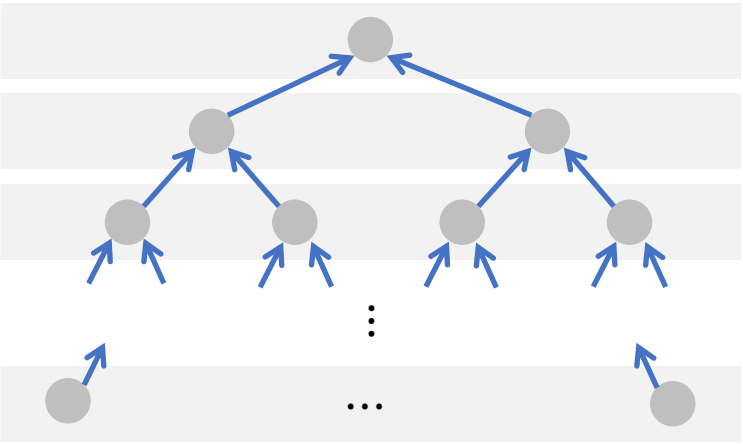
Why Euclidean and hierarchy not compatible? - Polynomial growth speed of space

- # entities in hierarchical data: **exponentially** grows w.r.t. layer depth.
- (surface area) $\propto r^{D-1}$ (**polynomial** w.r.t. radius r) in Euclidean space $\left(\mathbb{R}^D, \Delta_{\mathbb{R}^D,\mathrm{Euc}}\right)$

Radius	Circ. C	Layer dep.	# entities
0	0	0	1
1	2π	1	2
2	4π	2	4
⋮	⋮	⋮	⋮
r	$2\pi r$	H	2^H

polynomial ◀ incompatible ▶ exponential

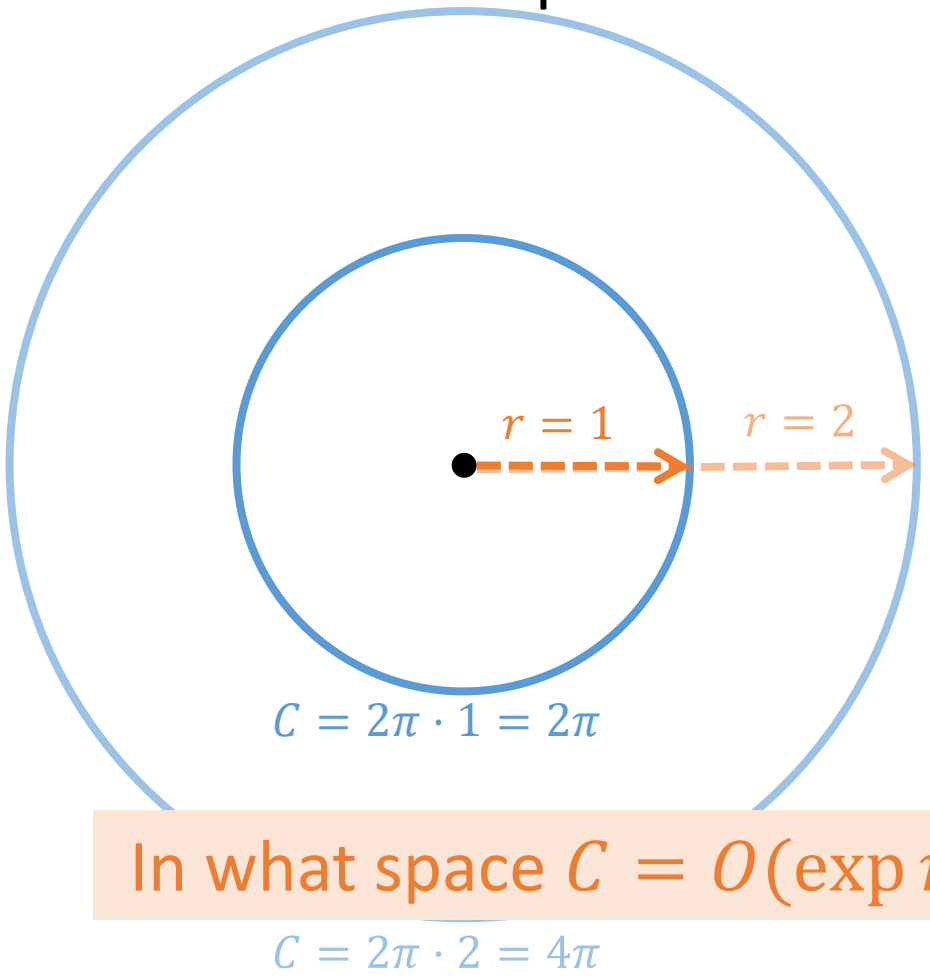
Hierarchical data
(e.g., complete binary tree)



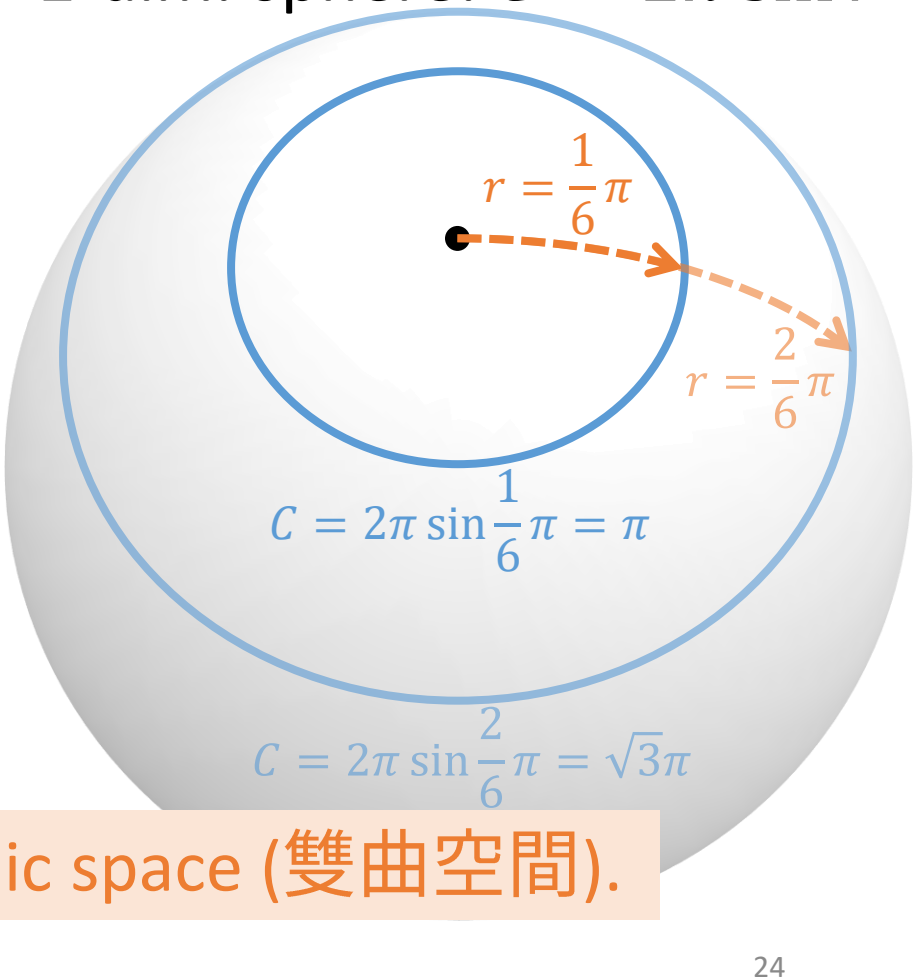
Where do we find a different growth speed?

Answer: in **non-Euclidean space**

2-dim. Euclidean space: $C = 2\pi r$



2-dim. sphere: $C = 2\pi \sin r$



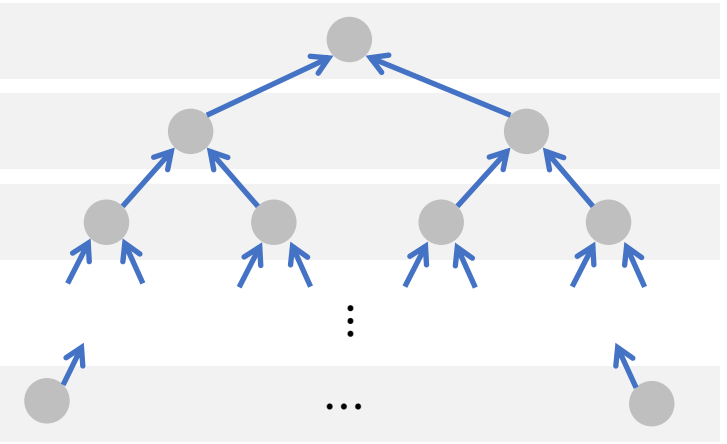
In what space $C = O(\exp r)$? – Hyperbolic space (雙曲空間).

Advantage of hyperbolic space

- # entities in hierarchical data: **exponentially** grows w.r.t. layer depth.
- (surface area) $\propto (\sinh r)^{D-1}$ (**exponential** w.r.t. radius r) in hyperbolic space

Radius	Circ. C	Layer dep.	# entities
0	0	0	1
1	$2\pi \sinh 1$	1	2
2	$2\pi \sinh 2$	2	4
⋮	⋮	⋮	⋮
r	$2\pi \sinh r$	H	2^H

Hierarchical data
(e.g., complete binary tree)



$\sinh r = O(\exp r)$ **exponential** ◀ **compatible** ▶ **exponentia**

Attracting much attention in ML from 2017 [Nickel & Kiela, 2017]

We can embed any tree in 2-dim'l hyperbolic space [Sarker+ 2011]

Outline

- ~~Representation learning's motivation~~
- **Why non-Euclidean, hyperbolic?**
  - **Hyperbolic space's exponential space growth speed matches data with a hierarchical structure.**
- How does hyperbolic space look like?
- Representation learning models using hyperbolic space

Outline

- ~~Representation learning's motivation~~
- ~~Why non-Euclidean, hyperbolic?~~
- **How does hyperbolic space look like?**
- Representation learning models using hyperbolic space

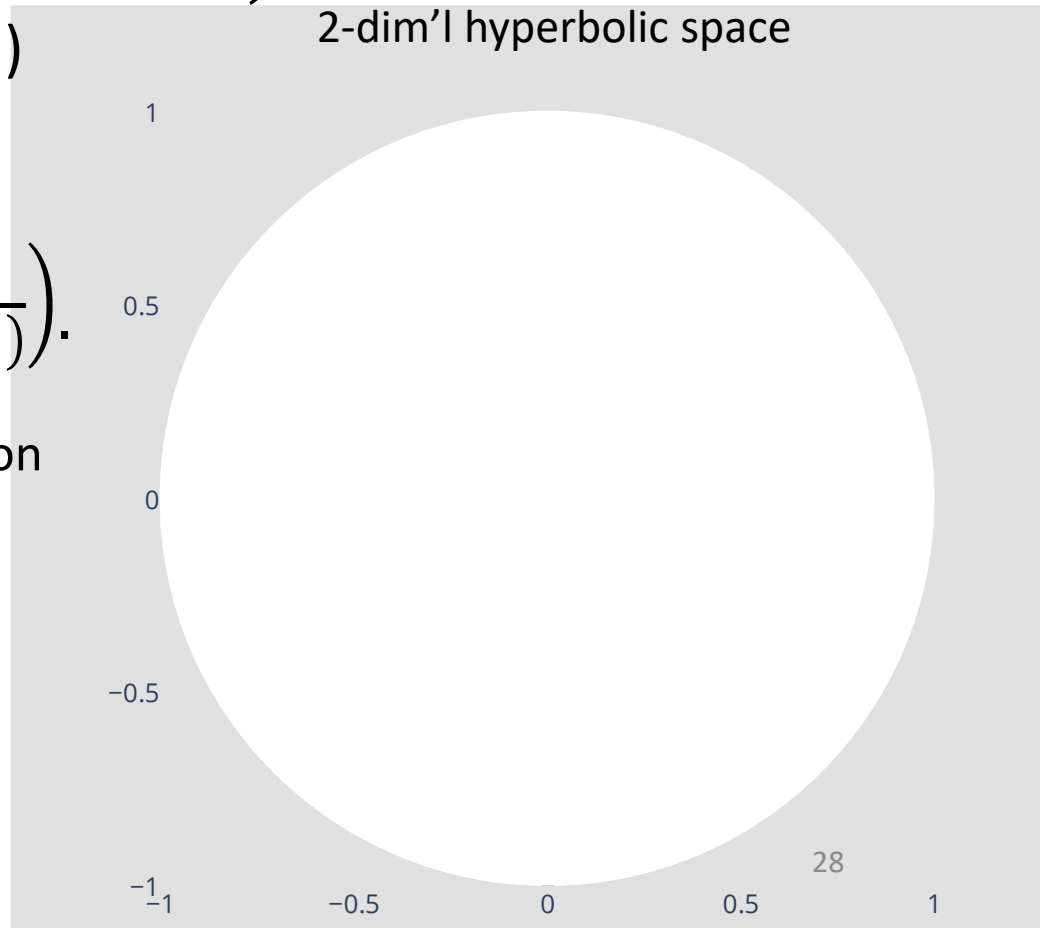
What is hyperbolic space?

D-dimensional hyperbolic space $\left(\mathbb{D}^D, \Delta_{\mathbb{D}^D,\mathrm{hyp}}(\boldsymbol{x},\boldsymbol{y})\right)$

- $\mathbb{D}^D = \{\boldsymbol{x} \in \mathbb{R}^D \mid \|\boldsymbol{x}\|_2 < 1\}$ (the unit open ball)
  - Where $\|\boldsymbol{x}\|_2 = \sqrt{\boldsymbol{x}^\top \boldsymbol{x}}$.
- Distance function:

$$\Delta_{\mathbb{D}^D,\mathrm{hyp}}(\boldsymbol{x},\boldsymbol{y}) = \cosh^{-1}\left(1 + \frac{2\|\boldsymbol{x}-\boldsymbol{y}\|_2^2}{(1-\|\boldsymbol{x}\|_2^2)(1-\|\boldsymbol{y}\|_2^2)}\right).$$

  - Where $\cosh x \coloneqq \frac{e^x+e^{-x}}{2}$, hyperbolic cosine function and its inverse $\cosh^{-1} x \coloneqq \ln\left(x + \sqrt{x^2-1}\right)$
- The above realization of hyperbolic space is called the Poincaré disk model.



Exercise on the distance function of 2-dimensional hyperbolic space.

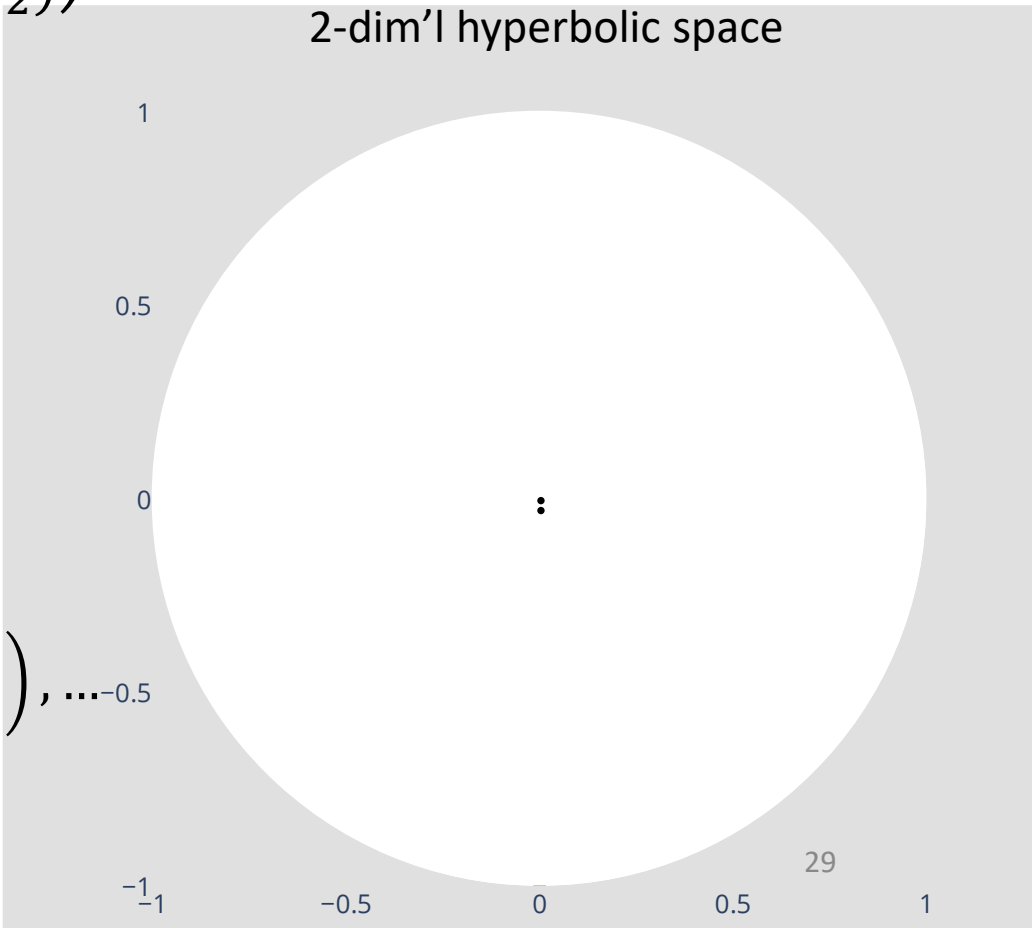
$$\Delta_{\mathbb{D}^D,\mathrm{hyp}}(\boldsymbol{x},\boldsymbol{y}) = \cosh^{-1}\left(1+\frac{2\|\boldsymbol{x}-\boldsymbol{y}\|_2^2}{(1-\|\boldsymbol{x}\|_2^2)(1-\|\boldsymbol{y}\|_2^2)}\right)$$

$$\approx \ln\left(\frac{2\|\boldsymbol{x}-\boldsymbol{y}\|_2^2}{(1-\|\boldsymbol{x}\|_2^2)(1-\|\boldsymbol{y}\|_2^2)}\right).$$

Exercise: consider the distance between $\begin{bmatrix}0\\c\end{bmatrix}$ and $\begin{bmatrix}0\\c-0.01\end{bmatrix}$ as $c \nearrow 1$.

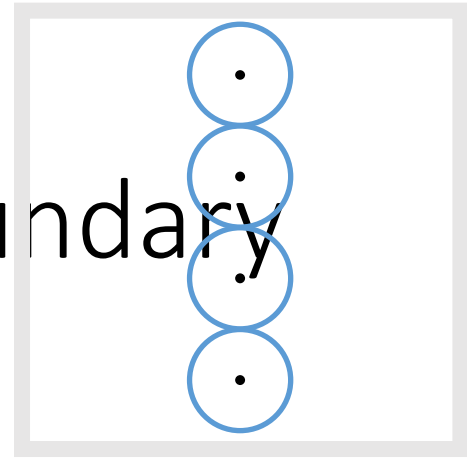
That is, we consider

$$\Delta_{\mathbb{D}^D,\mathrm{hyp}}\left(\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}0\\-0.01\end{bmatrix}\right), \Delta_{\mathbb{D}^D,\mathrm{hyp}}\left(\begin{bmatrix}0\\0.9\end{bmatrix},\begin{bmatrix}0\\0.89\end{bmatrix}\right),$$
$$\Delta_{\mathbb{D}^D,\mathrm{hyp}}\left(\begin{bmatrix}0\\0.99\end{bmatrix},\begin{bmatrix}0\\0.98\end{bmatrix}\right), \Delta_{\mathbb{D}^D,\mathrm{hyp}}\left(\begin{bmatrix}0\\0.999\end{bmatrix},\begin{bmatrix}0\\0.989\end{bmatrix}\right), \ldots$$



Example answer: the space is infinitely large around the disk boundary

2-dim'l Euclidean space



Consider $\Delta_{\mathbb{D}^D,\mathrm{hyp}}\left(\begin{bmatrix}0\\ c\end{bmatrix},\begin{bmatrix}0\\ c-0.01\end{bmatrix}\right)$.

Numerator is always $2 \cdot 0.01^2$

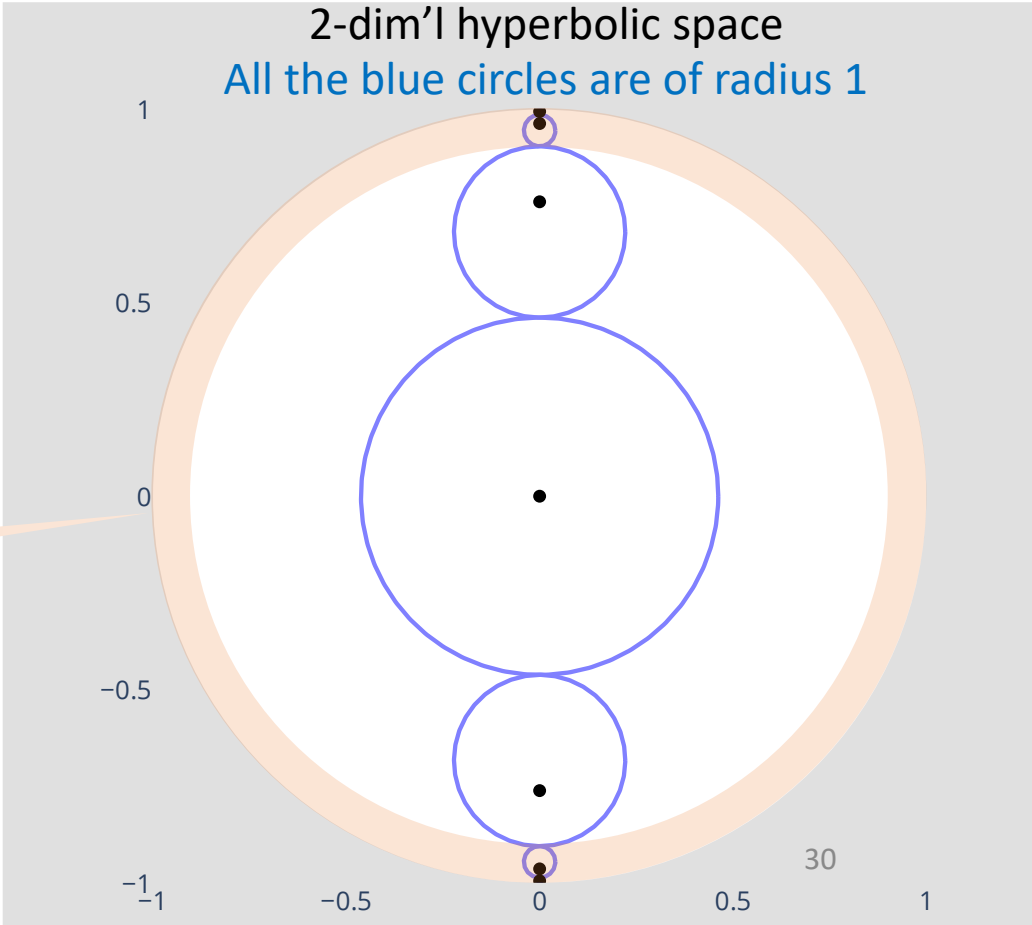
$$\Delta_{\mathbb{D}^D,\mathrm{hyp}}(\boldsymbol{x},\boldsymbol{y}) \approx \ln\left(\frac{2\|\boldsymbol{x}-\boldsymbol{y}\|_2^2}{(1-\|\boldsymbol{x}\|_2^2)(1-\|\boldsymbol{y}\|_2^2)}\right)$$

Denominator $\to 0$ as $\|\boldsymbol{x}\|_2^2, \|\boldsymbol{y}\|_2^2 \to 1$

$\to +\infty$ as $c \nearrow 1$.

2-dim'l hyperbolic space

All the blue circles are of radius 1



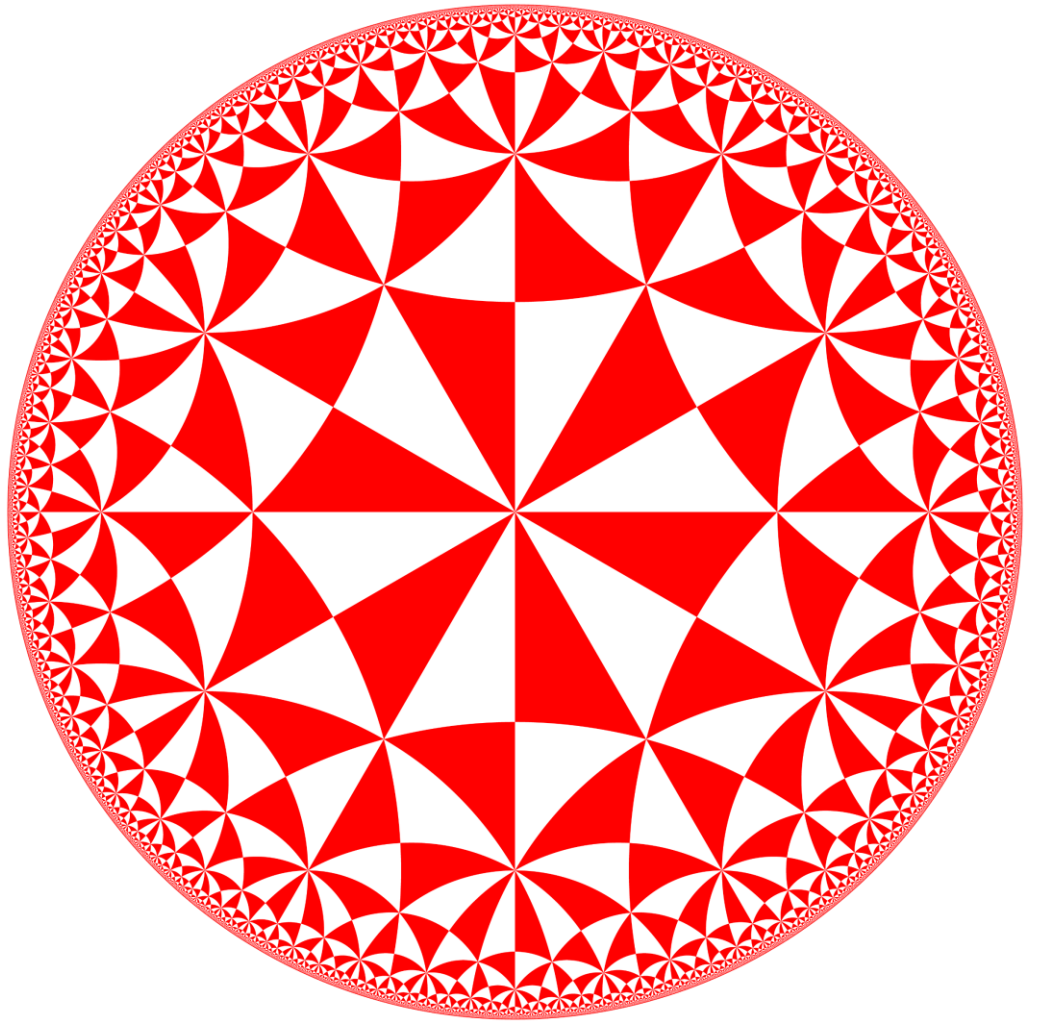
Around the disk boundary,
even a small coordinate difference may lead to a large distance

As a result, $C = O(\exp r)$

Visualization

https://en.wikipedia.org/wiki/File:Hyperbolic_domains_642.png (public domain)

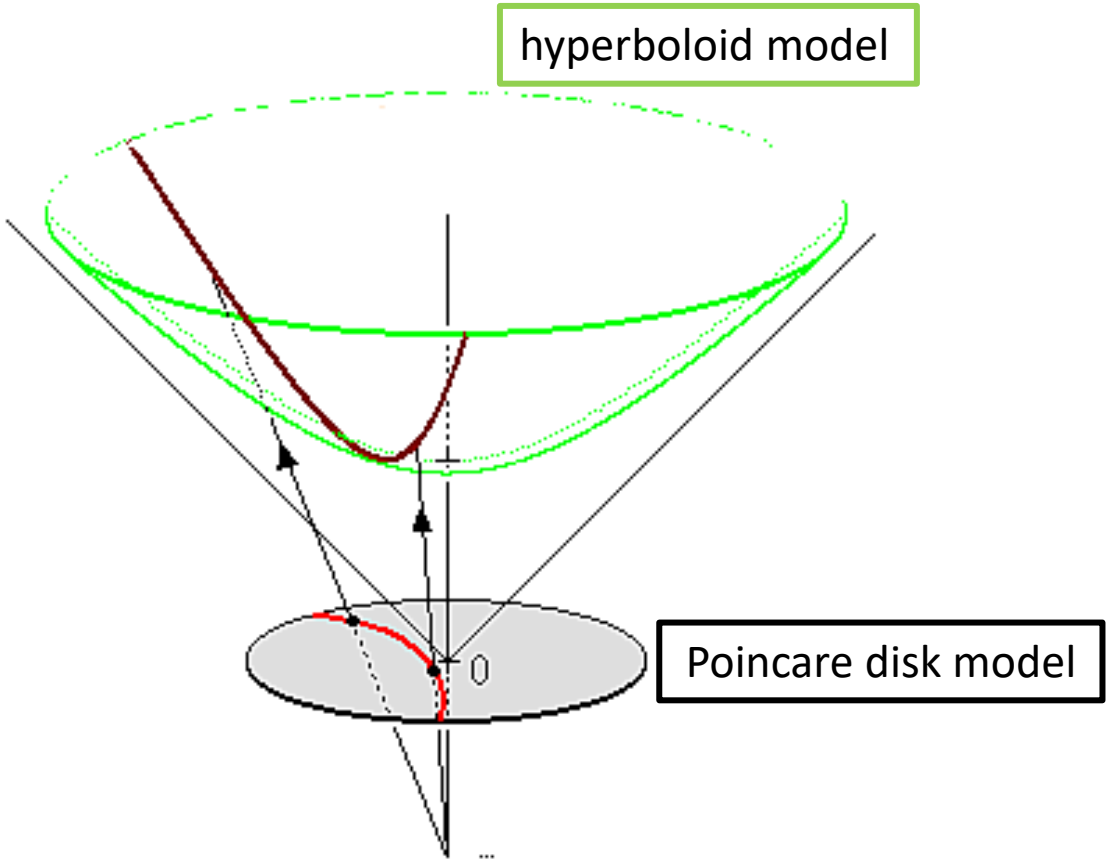
All the triangles have the same shape and size



Why is it called hyperbolic space (雙曲空間)?

We can realize the 2-dimensional hyperbolic space as a hyperboloid (雙曲面) $x^2 + y^2 - z^2 = -1, z > 0$ in the 3-dimensional Minkowski space.
The above realization is called the hyperboloid model of hyperbolic space.

- Minkowski space: the space where the distance function is given by $\sqrt{(x' - x)^2 + (y' - y)^2 - (z' - z)^2}$



https://commons.wikimedia.org/wiki/File:HyperboloidProjection.png (CC0)

Outline

- ~~Representation learning's motivation~~
- ~~Why non-Euclidean, hyperbolic?~~
- **How does hyperbolic space look like?**
  - **The space around the disk boundary is much larger than it looks (in the Poincare disk model)**
- Representation learning models using hyperbolic space

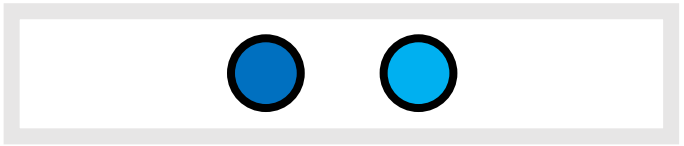
Outline

- ~~Representation learning's motivation~~
- ~~Why non-Euclidean, hyperbolic?~~
- ~~How does hyperbolic space look like?~~
- **Representation learning models using hyperbolic space**

Method design scheme

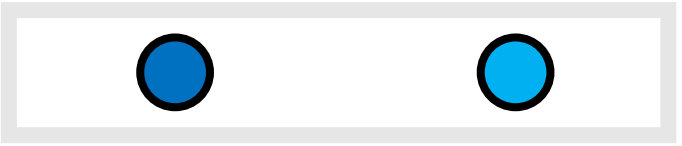
Basic ideas: designing a loss function that is:

- Increasing w.r.t. the distance between the representations of two entities strongly related to each other.



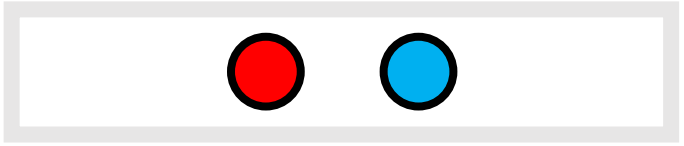
Strongly related and close: **small loss**

Increasing w.r.t. distance



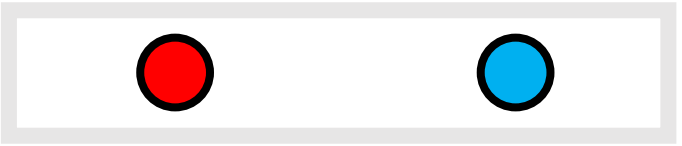
Strongly related and distant: **large loss**

- Decreasing w.r.t. to the distance between the representations of two entities weakly related to each other.



Weakly related and close: **large loss**

Decreasing w.r.t. distance



Weakly related and distant: **small loss**

Method design scheme

We can get good representations by minimizing the loss function. If we succeed in minimizing it, the resulting representations are expected to satisfy

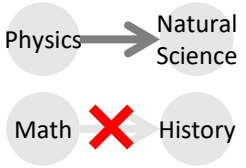
- Short distance between the representations of a strongly related pair.
- Long distance between the representations of a weakly related pair.

Random embedding example

Optimized embedding example

Example 1: positive/negative edges and graph embedding

- Positive/negative edges:
  - Positive edges: pairs of entities strongly related
  - Negative edges: pairs of entities weakly related



- Example hyperbolic model: Poincaré embedding [Nickel & Kiela, NeurIPS, 2017] where:
  - positive edges are generated by sampling edges in the graph,
  - negative edges are generated by randomly sampling a vertex pair from all the vertex pairs.

Loss function of Poincaré embedding [Nickel & Kiela, NeurIPS, 2017]

Input: a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the vertex set and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$.

The (stochastic) **loss function** is given by

$$\sum_{(i,j)\in\mathcal{E}} \left[-\Delta\left(z_i, z_j\right) + \log \sum_{k=1}^{n} \exp \Delta\left(z_i, z_{v^-(i)}\right) \right],$$

- Where $v^-(i)$ is a random variable uniformly distributed on the negative tails from i defined by $\mathcal{N}^-(i) = \{ j \mid (i,j) \neq \mathcal{E} \}$.

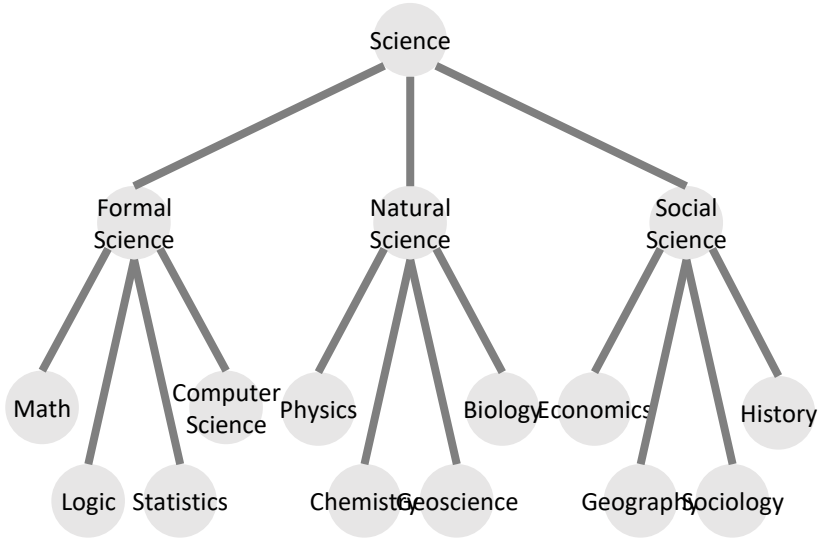
Loss function of Poincaré embedding [Nickel & Kiela, NeurIPS, 2017]

Input: a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the vertex set and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$.

The (stochastic) **loss function** is given by

$$\sum_{(i,j)\in\mathcal{E}} \left[-\Delta\left(z_i, z_j\right) + \log \sum_{k=1}^{n} \exp \Delta\left(z_i, z_{v^-(i)}\right) \right],$$

- Where $v^-(i)$ is a random variable uniformly distributed on the negative tails from i defined by $\mathcal{N}^-(i) = \{ j \mid (i,j) \neq \mathcal{E} \}$.
- **Example**: if $(i,j) = ($Formal Science$,$ Science$)$, then $v^-($Formal Science$)$ is uniformly distributed on...



Loss function of Poincaré embedding [Nickel & Kiela, NeurIPS, 2017]

Input: a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the vertex set and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$.

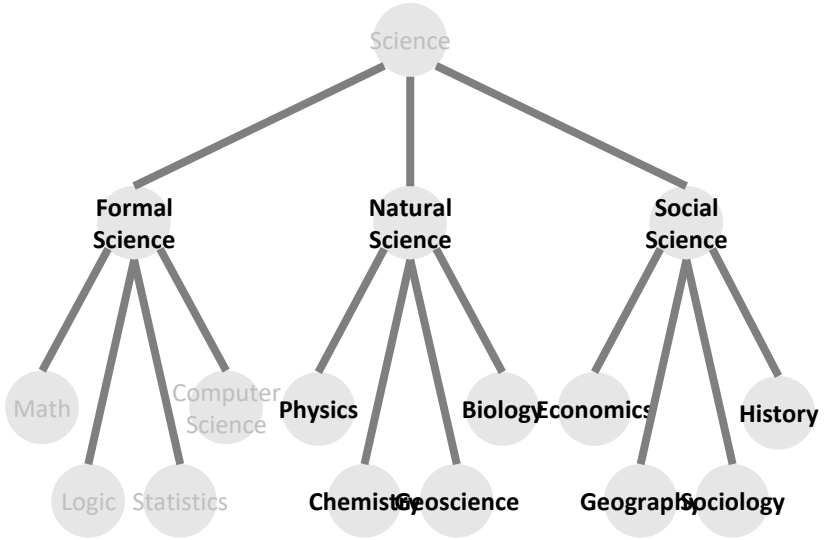
The (stochastic) **loss function** is given by

$$\sum_{(i,j)\in\mathcal{E}} \left[-\Delta\left(z_i, z_j\right) + \log \sum_{k=1}^{n} \exp \Delta\left(z_i, z_{v^-(i)}\right) \right],$$

- Where $v^-(i)$ is a random variable uniformly distributed on the negative tails from i defined by $\mathcal{N}^-(i) = \{ j \mid (i,j) \neq \mathcal{E} \}$.
- **Example**: if $(i,j) = ($Formal Science$,$Science$)$, then $v^-($Formal Science$)$ is uniformly distributed on...

The loss is

- decreasing w.r.t. distance btw positive edge reprs,
- increasing w.r.t. distance btw negative edge reprs.

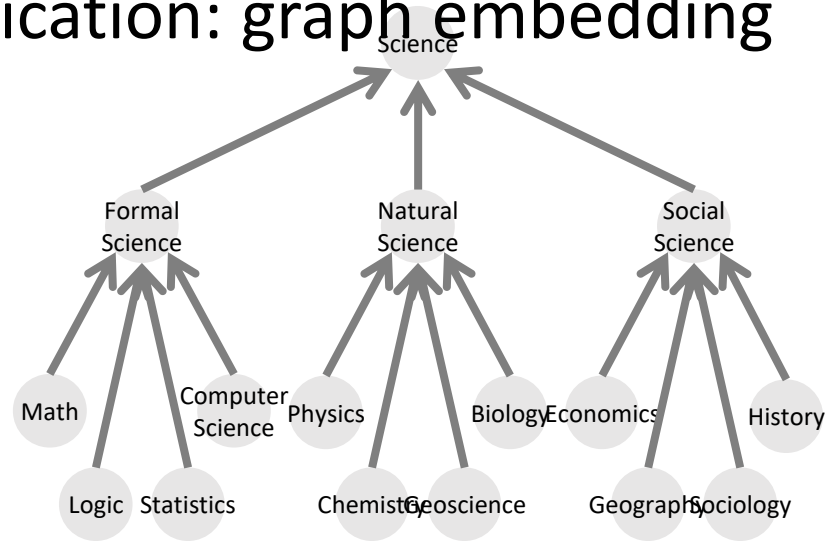


Two types of data input

Positive/negative edges

- Positive edges:
  pairs of entities closely related
- Negative edges:
  pairs of entities not related

Application: graph embedding



Ordinal data

- A triplet (i_s, j_s, k_s) with a negative label $y_s = -1$:
  j_s is more closely related to i_s than k_s.

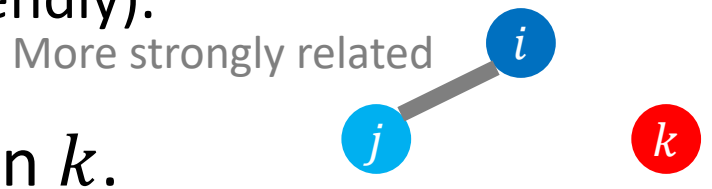
- A triplet (i_s, j_s, k_s) with a positive label $y_s = +1$:
  k_s is more closely related to i_s than j_s.

- Application: crowdsourcing

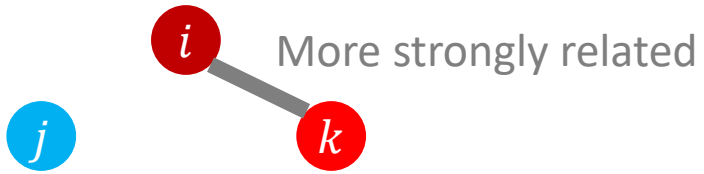
Example 2: ordinal data

Ordinal data consist of pairs of
- An entity triplet (i, j, k) and
- A label $y \in \{\pm 1\}$ (human annotation friendly).

- A negative label $y = -1$ indicates that j is more strongly related to i than k.



- A positive label $y = +1$ indicates that k is more strongly related to i than j.



- Example hyperbolic model: Hyperbolic ordinal embedding [Suzuki+, ACML, 2019]

Hyperbolic ordinal embedding's loss function (the simplest version)

Input $\big((i_1, j_1, k_1), y_1\big), \big((i_2, j_2, k_2), y_2\big), \dots, \big((i_S, j_S, k_S), y_S\big)$.

The loss function is given by

$$\sum_{s=1}^{S} \Big(\Big[y_s \Big(\Delta\big(x_{i_s}, x_{k_s}\big) - \Delta\big(x_{i_s}, x_{j_s}\big) \Big) + \delta \Big]_+ \Big).$$

The loss is

- decreasing w.r.t. distance btw representation pair more strongly related,
- increasing w.r.t. distance btw representation pair more weakly related.

Outline

- ~~Representation learning's motivation~~
- ~~Why non-Euclidean, hyperbolic?~~
- ~~How does hyperbolic space look like?~~
- **Representation learning models using hyperbolic space**
  - The loss function should be
    - Increasing w.r.t. the distance btw the reprs of two entities strongly related to each other.
    - Decreasing w.r.t. the distance btw the reprs of two entities weakly related to each other.

Summary

- Representation learning's motivation
  - To obtain entities' representations that reflect the relation among them.
- Why non-Euclidean, hyperbolic?
  - Hyperbolic space's exponential space growth speed matches data with a hierarchical structure.
- How does hyperbolic space look like?
  - The space around the disk boundary is much larger than it looks (in the Poincare disk model)
- Representation learning models using hyperbolic space
  - The loss function should be
    - Increasing w.r.t. the distance btw the reprs of two entities strongly related to each other.
    - Decreasing w.r.t. the distance btw the reprs of two entities weakly related to each other.

Take home message

If your data have a hierarchical or tree-like structure, hyperbolic space might help you.

Please join the anonymous feedback opportunity:

https://docs.google.com/forms/d/e/1FAIpQLSdfaP8RAaf96UjE96jyzpPPGLT_RIs8O0GuP0DDy-QWxZUcCQ/viewform?usp=sf_link

References

- [Sarker, 2011] Sarkar, Rik. "Low Distortion Delaunay Embedding of Trees in Hyperbolic Plane." In *Graph Drawing: 19th International Symposium, GD 2011, Eindhoven, The Netherlands, September 21-23, 2011, Revised Selected Papers*, vol. 7034, p. 355. Springer, 2011.
- [Nickel & Kiela, NeurIPS, 2017] Nickel, Maximillian, and Douwe Kiela. "Poincaré embeddings for learning hierarchical representations." *Advances in neural information processing systems* 30 (2017): 6338-6347.
- [Suzuki+, ACML, 2019] Suzuki, Atsushi, et al. "Hyperbolic Ordinal Embedding." *Asian Conference on Machine Learning*. PMLR, 2019.