Poincaré Embedding for Learning Hierarchical Representations

Marco Tsun Ting Lee

August 11, 2017

Overview

1 Background
- Objective
- Euclidean vs Hyperbolic
- Paper's Result

2 Dataset
- WordNet

3 Euclidean Space
- Training
- Evaluation

4 Hyperbolic Space
- Definition & Properties
- Training

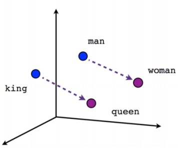
What is a Word Embedding?

cats $\rightarrow (0.8123, 0.7644, 82.2314, \ldots)$
dogs $\rightarrow (1.2323, 4.9712, 4.2790, \ldots)$
king $\rightarrow (6.1686, 78.1318, 6.4867, \ldots)$
queen $\rightarrow (34.1497, 6.4930, 3.2084, \ldots)$

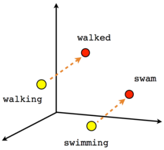
- Dimensionality Reduction (more efficient representation)
- Contextual Similarity (more expressive representation)
  - Words with similar context are clustered together.

Examples of Word Embedding

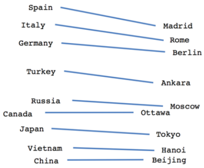
Traditional Example: $\Theta(king) - \Theta(queen) = \Theta(man) - \Theta(woman)$



Male-Female

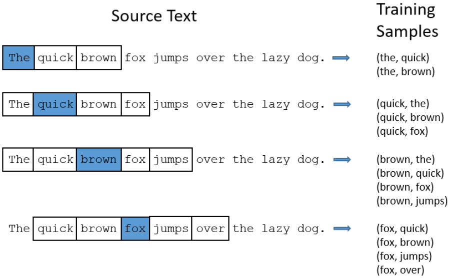


Verb tense



Country-Capital

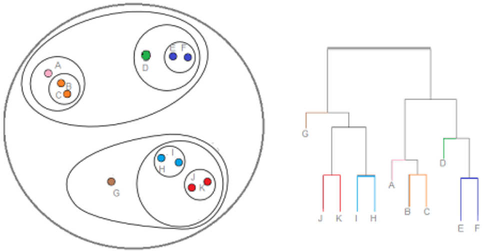
Word2Vec - Existing Word Embedding Method



see http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

Hierarchical Clustering - Method in Paper

Use Hierarchical Structure!



Hierarchical Clustering vs Word2Vec

Advantages:

- True context
- Rare words are not biased
- more efficient representation

Disadvantages:

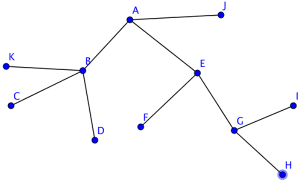
- Most data do not have hierarchical structure
- Hard to get dataset with hierarchical structure built in

Why use Hyperbolic space?

Motivation:
Any finite subset of an hyperbolic space "looks like" a finite tree!
Informally, hyperbolic space can be thought of as a continuous version of tree.

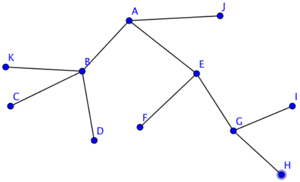
Example

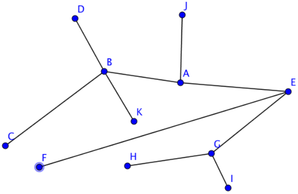
Imagine we have a tree...



Example

Assume each edge has length 1, we can define a discrete metric on the tree.





These two trees are the same!

Example

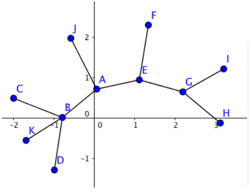
We only need the length between vertices to represent a tree!

```
#    weighted graph adjacency matrix
#    A  B  C  D  E  F  G  H  I  J  K
A = [0, 1, 2, 2, 1, 2, 2, 3, 3, 1, 2]
B = [1, 0, 1, 1, 2, 3, 3, 4, 4, 2, 1]
C = [2, 1, 0, 2, 3, 4, 4, 5, 5, 3, 2]
D = [2, 1, 2, 0, 3, 4, 4, 5, 5, 3, 2]
E = [1, 2, 3, 3, 0, 1, 1, 2, 2, 2, 3]
F = [2, 3, 4, 4, 1, 0, 2, 3, 3, 3, 4]
G = [2, 3, 4, 4, 1, 2, 0, 1, 1, 3, 4]
H = [3, 4, 5, 5, 2, 3, 1, 0, 2, 4, 5]
I = [3, 4, 5, 5, 2, 3, 1, 2, 0, 4, 5]
J = [1, 2, 3, 3, 2, 3, 3, 4, 4, 0, 3]
K = [2, 1, 2, 2, 3, 4, 4, 5, 5, 3, 0]
graph = [A, B, C, D, E, F, G, H, I, J, K]
```

Example

The optimal embedding in Euclidean space:

```
#    weighted graph adjacency matrix
#    A  B  C  D  E  F  G  H  I  J  K
A = [0, 1, 2, 2, 1, 2, 2, 3, 3, 1, 2]
B = [1, 0, 1, 1, 2, 3, 3, 4, 4, 2, 1]
C = [2, 1, 0, 2, 3, 4, 4, 5, 5, 3, 2]
D = [2, 1, 2, 0, 3, 4, 4, 5, 5, 3, 2]
E = [1, 2, 3, 3, 0, 1, 1, 2, 2, 2, 3]
F = [2, 3, 4, 4, 1, 0, 2, 3, 3, 3, 4]
G = [2, 3, 4, 4, 1, 2, 0, 1, 1, 3, 4]
H = [3, 4, 5, 5, 2, 3, 1, 0, 2, 4, 5]
I = [3, 4, 5, 5, 2, 3, 1, 2, 0, 4, 5]
J = [1, 2, 3, 3, 2, 3, 3, 4, 4, 0, 3]
K = [2, 1, 2, 2, 3, 4, 4, 5, 5, 3, 0]
graph = [A, B, C, D, E, F, G, H, I, J, K]
```




```
#              euclid dist adjacency matrix
#      A    B    C    D    E    F    G    H    I    J    K
A = [ 0.   1.1  2.1  2.3  1.1  2.   2.1  3.2  3.2  1.4  2.2]
B = [ 1.1  0.   1.3  1.3  2.1  3.1  3.   3.9  4.2  2.   1.1]
C = [ 2.1  1.3  0.   2.   3.1  3.8  4.2  5.2  5.2  2.1  1.1]
D = [ 2.3  1.3  2.   0.   3.1  4.3  3.7  4.3  4.9  3.3  1. ]
E = [ 1.1  2.1  3.1  3.1  0.   1.4  1.1  2.3  2.1  2.   3.2]
F = [ 2.   3.1  3.8  4.3  1.4  0.   1.9  3.   2.2  1.9  4.2]
G = [ 2.1  3.   4.2  3.7  1.1  1.9  0.   1.2  1.2  3.1  4.1]
H = [ 3.2  3.9  5.2  4.3  2.3  3.   1.2  0.   1.3  4.3  4.8]
I = [ 3.2  4.2  5.2  4.9  2.1  2.2  1.2  1.3  0.   3.9  5.2]
J = [ 1.4  2.   2.1  3.3  2.   1.9  3.1  4.3  3.9  0.   2.8]
K = [ 2.2  1.1  1.1  1.   3.2  4.2  4.1  4.8  5.2  2.8  0. ]
```

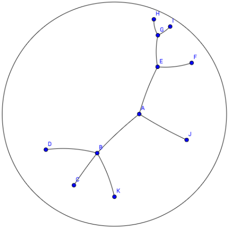
L2 Loss $= 7.166$

see https://github.com/marcoleewow/Find-Optimal-Space-Embedding-for-Trees

Example

The optimal embedding in Hyperbolic space:

```
#    weighted graph adjacency matrix
#    A  B  C  D  E  F  G  H  I  J  K
A = [0, 1, 2, 2, 1, 2, 2, 3, 3, 1, 2]
B = [1, 0, 1, 1, 2, 3, 3, 4, 4, 2, 1]
C = [2, 1, 0, 2, 3, 4, 4, 5, 5, 3, 2]
D = [2, 1, 2, 0, 3, 4, 4, 5, 5, 3, 2]
E = [1, 2, 3, 3, 0, 1, 1, 2, 2, 2, 3]
F = [2, 3, 4, 4, 1, 0, 2, 3, 3, 3, 4]
G = [2, 3, 4, 4, 1, 2, 0, 1, 1, 3, 4]
H = [3, 4, 5, 5, 2, 3, 1, 0, 2, 4, 5]
I = [3, 4, 5, 5, 2, 3, 1, 2, 0, 4, 5]
J = [1, 2, 3, 3, 2, 3, 3, 4, 4, 0, 3]
K = [2, 1, 2, 2, 3, 4, 4, 5, 5, 3, 0]
graph = [A, B, C, D, E, F, G, H, I, J, K]
```




```
#                hyp dist adjacency matrix
#      A    B    C    D    E    F    G    H    I    J    K
A = [ 0.   1.1  2.1  2.1  1.1  2.   2.1  3.   3.1  1.3  2. ]
B = [ 1.1  0.   1.1  1.3  2.1  3.1  3.   3.9  4.1  2.   1.3]
C = [ 2.1  1.1  0.   1.5  3.1  4.2  4.1  5.   5.1  3.   1.5]
D = [ 2.1  1.3  1.5  0.   3.   4.   3.8  4.7  4.9  3.2  2.4]
E = [ 1.1  2.1  3.1  3.   0.   1.2  1.1  2.1  2.1  2.   3.1]
F = [ 2.   3.1  4.2  4.   1.2  0.   2.   3.1  2.7  2.6  4. ]
G = [ 2.1  3.   4.1  3.8  1.1  2.   0.   1.2  1.1  3.1  4. ]
H = [ 3.   3.9  5.   4.7  2.1  3.1  1.2  0.   1.8  4.1  5. ]
I = [ 3.1  4.1  5.1  4.9  2.1  2.7  1.1  1.8  0.   4.   5.1]
J = [ 1.3  2.   3.   3.2  2.   2.6  3.1  4.1  4.   0.   2.6]
K = [ 2.   1.3  1.5  2.4  3.1  4.   4.   5.   5.1  2.6  0. ]
```

L2 Loss = 1.928!

see https://github.com/marcoleewow/Find-Optimal-Space-Embedding-for-Trees

Quasi-Isometry

In math terms,

Theorem

*$\forall n \in \mathbb{Z}, \delta > 0, \exists$ a constant C s.t. the following holds:
if $x_1, \ldots, x_n$ are points in a δ-hyperbolic space X,
$\exists$ a finite tree T and an embedding $f : T \to X$ s.t.
$x_i \in f(T) \forall i = 1, \ldots, n$ and
$\forall i, j : d_T(f^{-1}(x_i), f^{-1}(x_j)) \leq d_X(x_i, x_j) \leq d_T(f^{-1}(x_i), f^{-1}(x_j)) + C$*

The constant C can be taken to be $\delta \cdot h(n)$ with $h(n) = O(\log n)$ and this is optimal!

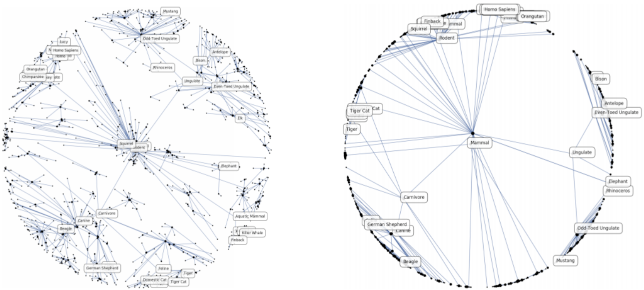
Hyperbolic vs Euclidean Results

		Dimensionality					
		5	10	20	50	100	200
Euclidean	Rank	3311.1	2199.5	952.3	351.4	190.7	81.5
	MAP	0.024	0.059	0.176	0.286	0.428	0.490
Poincaré	Rank	5.7	**4.3**	4.9	4.6	4.6	4.6
	MAP	0.825	0.852	0.861	**0.863**	0.856	0.855

Hyperbolic Advantages

- Outperform Euclidean embedding significantly
- Much more efficient representation!
- Did not add much computational intensity

Embedding Visualization



(a) Intermediate embedding after 20 epochs (b) Embedding after convergence

WordNet

- Clear latent hierarchical structure
- Easy to use
- Publicly available for download
- Enough data for experiments

```
>>> wn.synset('dog.n.01')
Synset('dog.n.01')
>>> print(wn.synset('dog.n.01').definition())
a member of the genus Canis (probably descended from the common wolf) that has been domesticated by man since prehistoric times; occurs in many breeds
>>> len(wn.synset('dog.n.01').examples())
1
>>> print(wn.synset('dog.n.01').examples()[0])
the dog barked all night
>>> wn.synset('dog.n.01').lemmas()
[Lemma('dog.n.01.dog'), Lemma('dog.n.01.domestic_dog'), Lemma('dog.n.01.Canis_familiaris')]
>>> [str(lemma.name()) for lemma in wn.synset('dog.n.01').lemmas()]
['dog', 'domestic_dog', 'Canis_familiaris']
>>> wn.lemma('dog.n.01.dog').synset()
Synset('dog.n.01')
```

http://www.nltk.org/howto/wordnet.html

All Noun Synsets

```
>>> for synset in list(wn.all_synsets('n'))[:10]:
...     print(synset)
...
Synset('entity.n.01')
Synset('physical_entity.n.01')
Synset('abstraction.n.06')
Synset('thing.n.12')
Synset('object.n.01')
Synset('whole.n.02')
Synset('congener.n.03')
Synset('living_thing.n.01')
Synset('organism.n.01')
Synset('benthos.n.02')
```

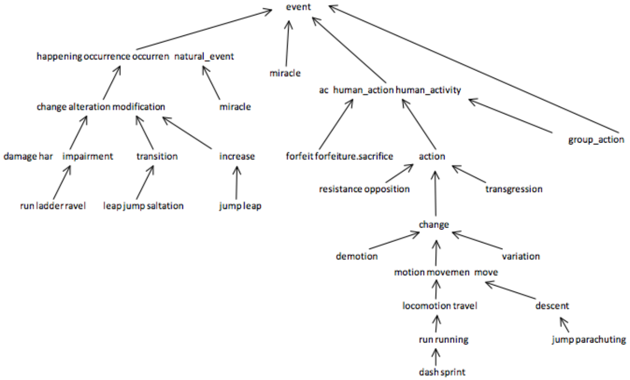
Hypernymy Relation

Hypernymy relation = 'is-a' relation
e.g. dog (hyponym) 'is-a' canine (hypernym)

Compute transitive closures of synsets

```
>>> dog = wn.synset('dog.n.01')
>>> hypo = lambda s: s.hyponyms()
>>> hyper = lambda s: s.hypernyms()
>>> list(dog.closure(hypo, depth=1)) == dog.hyponyms()
True
>>> list(dog.closure(hyper, depth=1)) == dog.hypernyms()
True
>>> list(dog.closure(hypo))
[Synset('basenji.n.01'), Synset('corgi.n.01'), Synset('cur.n.01'),
 Synset('dalmatian.n.02'), Synset('great_pyrenees.n.01'),
 Synset('griffon.n.02'), Synset('hunting_dog.n.01'), Synset('lapdog.n.01'),
 Synset('leonberg.n.01'), Synset('mexican_hairless.n.01'),
 Synset('newfoundland.n.01'), Synset('pooch.n.01'), Synset('poodle.n.01'), ...]
>>> list(dog.closure(hyper))
[Synset('canine.n.02'), Synset('domestic_animal.n.01'), Synset('carnivore.n.01'),
Synset('animal.n.01'), Synset('placental.n.01'), Synset('organism.n.01'),
Synset('mammal.n.01'), Synset('living_thing.n.01'), Synset('vertebrate.n.01'),
Synset('whole.n.02'), Synset('chordate.n.01'), Synset('object.n.01'),
Synset('physical_entity.n.01'), Synset('entity.n.01')]
```

Directed Acyclic Graph



WordNet Data Samples

- Total of 82,115 nouns and 743,241 hypernymy relations.
- Index the nouns: {0:cat, 1:dog, 2:canine,..., 82114:queen}
- Let $\mathcal{D} = \{(u, v)\}$ be the set of observed hypernymy relations between noun pairs.
- e.g. dog 'is-a' canine $=> (1, 2) \in \mathcal{D}$

Objective

- Euclidean Distance $d(\boldsymbol{u}, \boldsymbol{v}) = ||\boldsymbol{u} - \boldsymbol{v}||^2$
- Unit ball $\mathcal{B}^d = \{x \in \mathbb{R}^d : ||x|| < 1\}$

Let n be total number of nouns (in our case n = 82115).
We want to find $\Theta = \{\theta_i\}_{i=1}^n$, where $\theta_i \in \mathcal{B}^d$
To estimate Θ, we solve the optimization problem:

$$\Theta' \leftarrow \operatorname{argmin} \mathcal{L}(\Theta)$$

Negative Sampling Loss

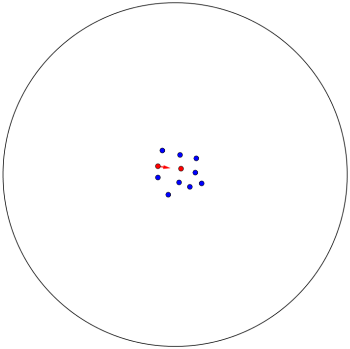
$$\mathcal{L}(\Theta) = \sum_{(u,v)\in\mathcal{D}} \log \frac{e^{-d(\boldsymbol{u},\boldsymbol{v})}}{\sum_{\boldsymbol{v'}\in\mathcal{N}(u)} e^{-d(\boldsymbol{u},\boldsymbol{v'})}}$$

$$\text{,where } \mathcal{N}(u) = \{v|(u,v) \notin \mathcal{D}\}$$

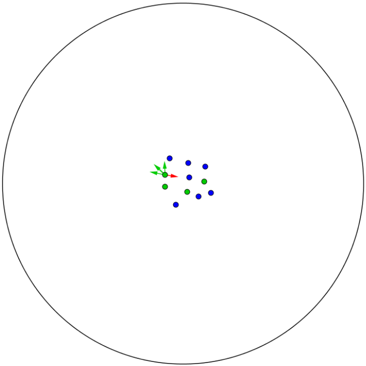
- soft ranking loss
- stochastically sample 10 negative samples
- differentiable
- update $\boldsymbol{u}$ only! All $\boldsymbol{v}$'s are fixed

Loss Function Visualization

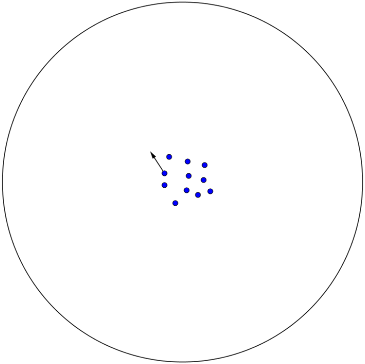
Initialization: $\Theta_{ij} \sim \mathcal{U}(-0.001, 0.001)$



Loss Function Visualization



Loss Function Visualization



SGD Optimizer

$$\mathrm{proj}(\theta) = \begin{cases} \frac{\theta}{||\theta||} - \epsilon & \text{if } ||\theta|| \geq 1 \\ \theta & \text{otherwise.} \end{cases}$$

$$\theta_{t+1} \leftarrow \mathrm{proj}(\theta_t - \eta_t \nabla_E \mathcal{L})$$

Link Prediction

- split data samples into 80% train, 10% valid and 10% test samples.
- Test generalization performance by hiding some links and check if we can predict them using distance in embedded space.

Rank Metric

For each pair of $(u, v) \in \mathcal{D}$, We rank its distance $d(\boldsymbol{u}, \boldsymbol{v})$ among the ground truth negative samples.

Mean Average Precision (MAP)

Rank nodes by closest distance to $\boldsymbol{u}$, and see if they are linked.
e.g.
number of links of $u = 4$.
links: 1, 0, 1, 1, 0, 1.
Precision: 1/1, 1/2, 2/3, 3/4, 3/5, 4/6
Average precision(take average across corrected links):
$(1/1 + 2/3 + 3/4 + 4/6)/4 = 0.77$
MAP: Take mean across all samples.

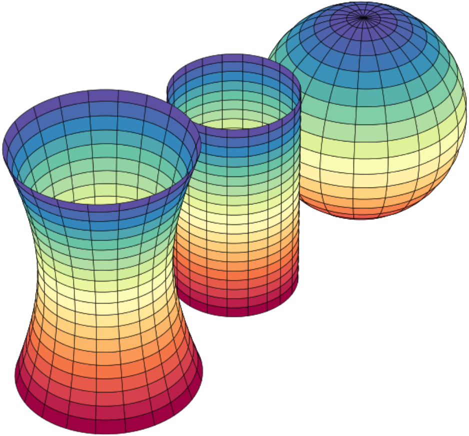
Results

		Dimensionality					
		5	10	20	50	100	200
Euclidean	Rank	3311.1	2199.5	952.3	351.4	190.7	81.5
	MAP	0.024	0.059	0.176	0.286	0.428	0.490

Why Not Euclidean?

- Curse of dimensionality
- Inefficient representation

Riemannian Manifold Examples



Poincaré Ball Model $\mathbb{B}^d$

Poincaré Ball Model $\mathbb{B}^d$:

$$\mathbb{B}^d = \{\boldsymbol{x} \in \mathbb{R}^d : ||\boldsymbol{x}|| < 1\}$$

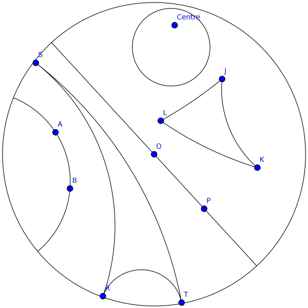
equipped with hyperbolic metric:

$$d(\boldsymbol{u}, \boldsymbol{v}) = \operatorname{arcosh}\left(1 + 2\frac{||\boldsymbol{u} - \boldsymbol{v}||^2}{(1 - ||\boldsymbol{u}||^2)(1 - ||\boldsymbol{v}||^2)}\right),$$

where $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{B}^d$.

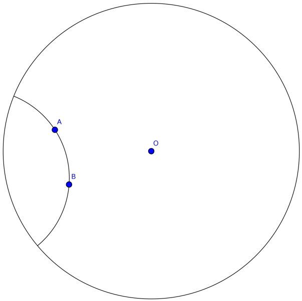
Hyperbolic space

Poincaré Ball Model $\mathbb{B}^2$



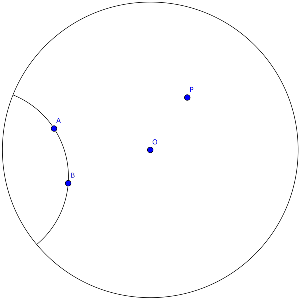
Hyperbolic Space Properties

Parallel Postulate



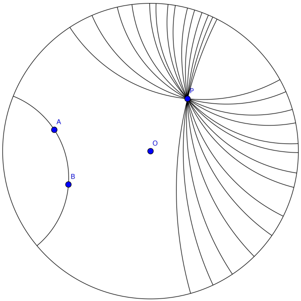
Hyperbolic Space Properties

Parallel Postulate



Hyperbolic Space Properties

Parallel Postulate



Hyperbolic Space Properties

Circle of radius r:

$$\text{Circumference} = 2\pi \sinh r > 2\pi r \text{ for } r > 0.$$
$$\text{Area} = 2\pi \cosh r - 1 > \pi r^2 \text{ for } r > 0.$$

Ball of radius r:

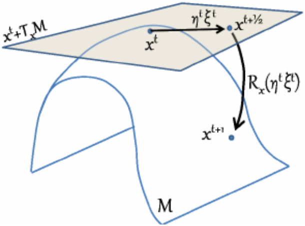
$$\text{Volume} = \pi(\sinh(2r) - 2r) > \frac{4}{3}\pi r^3 \text{ for } r > 0.$$

Intuition: "Stores more information" locally

Riemannian Gradient

$$\nabla_g = g_\theta^{-1} \nabla_E,$$

$$\text{where } g_\theta^{-1} = \frac{(1 - ||\theta||^2)^2}{4}$$



ignore notations in the above diagram!

Riemannian SGD

Full update equation:

$$\theta_{t+1} \leftarrow \mathrm{proj}(\theta_t - \eta_t \nabla_g \mathcal{L})$$

$$= \theta_{t+1} \leftarrow \mathrm{proj}(\theta_t - \eta_t g_{\theta_t}^{-1} \nabla_E \mathcal{L})$$

with hyperbolic distance for $\mathcal{L}$ instead of Euclidean!

Results

		Dimensionality					
		5	10	20	50	100	200
Euclidean	Rank	3311.1	2199.5	952.3	351.4	190.7	81.5
	MAP	0.024	0.059	0.176	0.286	0.428	0.490
Poincaré	Rank	5.7	**4.3**	4.9	4.6	4.6	4.6
	MAP	0.825	0.852	0.861	**0.863**	0.856	0.855

The End