Using hyperbolic large-margin classifiers for biological link prediction

SemDeep-5 @ IJCAI 2019

Asan Agibetov, Georg Dorffner, Matthias Samwald

Institute für Artificial Intelligence and Decision Support - Medical University of Vienna, Austria

Aug 12, 2019
Representing biological knowledge

1 “Shared hypothesis testing”, Agibetov et al., J. Biomed. Sem., 2018
Biological link prediction

$f_1 = $ TNF alpha overproduction

$f_2 = $ Cartilage degeneration

$f_3 = $ Joint deformation

$f_4 = $ Loss of collagen

$f_5 = $ Loss of proteoglycan

biological process results in
Link prediction as distance based inference in embedding space

How to learn $\phi$?

Graph domain

Embedding space $\mathbb{R}^n$

inference in embedding space based on distance

Points within r-ball centered around fixed point are connected in the graph domain

$$d(\phi(f_1), \phi(f_3)) \leq r \implies f_1 \text{ and } f_3 \text{ have a link}$$
Representation learning in Hyperbolic space

Nickel and Kiela. NIPS 2017
Hierarchical relationship from hyperbolic embeddings

Nickel and Kiela. ICML 2018
Clusters of proteins and age groups from hyperbolic coordinates

Lobato et al. Bioinformatics 2018
Hyperbolic embeddings

Same as in Euclidean case we try to learn a link estimator $Q(u, v) \mapsto [0, 1]$ ($u, v$ node pairs) with MLE

- $Pr(G) = \prod_{(u,v) \in E_{\text{train}}} Q(u, v) \prod_{(u,v) \notin E_{\text{train}}} 1 - Q(u, v)$
- If $Q$ perfect estimator then $Pr(x) = 1$ iff $x = G$ (i.e., graph can be fully reconstructed)

Embeddings are parameters $\Theta$ of link estimator $Q$; trained with cross-entropy loss $\mathcal{L}$ and negative sampling

- $\mathcal{L}(\Theta) = \sum_{(u,v)} \log \frac{e^{-d(u,v)}}{\sum_{v' \in \text{neg}(u)} e^{-d(u,v')}}$
- But we perform all computations in hyperbolic space
Backpropagation to learn embeddings

Nickel and Kiela. NIPS 2017
Link prediction for multi-relational biological knowledge graphs

\[
\begin{align*}
A_{\text{associated with}} &= \begin{bmatrix}
0 & 1 & 0 & 1 & 1 \\
1 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 \\
\end{bmatrix} \\
A_{\text{provokes}} &= \begin{bmatrix}
0 & 1 & 0 & 1 & 1 \\
1 & 0 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 \\
\end{bmatrix} \\
A_{\text{results in}} &= \begin{bmatrix}
0 & 1 & 0 & 1 & 1 \\
1 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 \\
\end{bmatrix}
\end{align*}
\]
Flattenning knowledge graphs

Turn KG into unlabelled directed graph, s.t., no pair of nodes is connected with more than one arc (directed edge)

Compensate reduced information with binary classifiers fine tuned for each relation type

| Dataset   | # pairs connected with > 1 relation types |
|-----------|------------------------------------------|
| WN11      | 124/93003 (0.133%)                       |
| FB15-237  | 23700/310116 (7.642%)                    |
| UMLS      | 1343/6527 (20.576%)                      |
| BIO-KG    | 0/1619239 (0%)                           |

Agibetov, Samwald. SemDeep-4@ISWC 2018
Agibetov, Samwald. SemDeep-4@ISWC 2018

Cho et al. arxiv 2018
## Performance evaluation

| Dataset  | # relation types | # entities | max # links per relation type | min # links per relation type | mean # links per relation type | # pairs connected with > 1 relation types |
|----------|------------------|------------|------------------------------|-------------------------------|-------------------------------|------------------------------------------|
| UMLS     | 46               | 137        | 1021                         | 1                             | 142                           | 1343/6527 (20.576%)                      |
| BIO-KG   | 9                | 346225     | 554366                       | 6159                          | 179915                        | 0/1619239 (0%)                          |

|                           | Euclidean embeddings |                     | Hyperbolic embeddings       |                     |
|---------------------------|----------------------|---------------------|-----------------------------|---------------------|
|                           | dim d                | Euc SVM             | Hyp SVM                      | Euc SVM             | Hyp SVM                      |
| UMLS                      | 2                    | 0.661 ± 0.023       | 0.616 ± 0.019               | 0.695 ± 0.026       | **0.703 ± 0.018**            |
|                           | 5                    | **0.780 ± 0.023**   | 0.743 ± 0.024               | **0.735 ± 0.030**   | **0.743 ± 0.024**            |
|                           | 10                   | **0.793 ± 0.025**   | 0.754 ± 0.022               | **0.767 ± 0.031**   | **0.742 ± 0.026**            |
| BIO-KG                    | 2                    | 0.692 ± 0.010       | 0.691 ± 0.010               | 0.613 ± 0.006       | 0.676 ± 0.009                |
|                           | 5                    | **0.776 ± 0.010**   | 0.771 ± 0.011               | **0.697 ± 0.008**   | **0.756 ± 0.011**            |
|                           | 10                   | 0.732 ± 0.009       | 0.723 ± 0.008               | 0.711 ± 0.010       | **0.763 ± 0.010**            |
Lessons learned

Benefit of learning hyperbolic embeddings

- fewer dimensions to capture latent semantic and hierarchical information
- scalability and interpretability (easier to visualize 2 or 3 dimensions)

From our preliminary results

- hyperbolic embeddings learn hierarchical relationships in UMLS better than Euclidean embeddings (lower dimensions)
- For complex and big graphs (BIO-KG) train hyperbolic embeddings for longer periods ($>500$ epochs)
Open issues and future directions

- even with recent advances in Riemannian SGD optimization \(^3\), learning hyperbolic embeddings still much slower than in the Euclidean case
- next steps should be focused on end-to-end hyperbolic embedding training (hyperbolic large-margin classifier loss is directly incorporated during the training process)
- code available at https://github.com/plumdeq/hsvm
- contact: asan.agibetov@meduniwien.ac.at

\(^3\)“Gradient descent in hyperbolic space”. Wilson and Leimeister, 2018
Why non-Euclidean space - (low-dim) manifolds

- Computing on a lower dimensional space leads to manipulating fewer degrees of freedom
- Non-linear degrees of freedom often make more intuitive sense
  - cities on the earth are better localized giving their longitude and latitude (2 dimensions)
  - instead of giving their position $x, y, z$ in the Euclidean 3D space
Learning graph embeddings

- Learn link estimate $Q(u, v) \mapsto [0, 1]$ ($u, v$ node pairs) and approximate graph structure (connectivity) with MLE (maximum likelihood estimation)\(^4\)

  $Pr(G) = \prod_{(u,v) \in E_{train}} Q(u,v) \prod_{(u,v) \notin E_{train}} 1 - Q(u,v)$

- If $Q$ perfect estimator then $Pr(x) = 1$ iff $x = G$ (i.e., graph can be fully reconstructed)

- $Q$ can be trained to estimate links at different orders, i.e., approximate $A^n$.

\(^4\)“Graph likelihood”, Haija, ... Perozzi, ..., CIKM17, NeurIPS 2018
Similar principle as word2vec \(^5\)

Similar words will have similar contexts

\[ \text{cat} \text{ ? } \text{kitty} \text{ ? } \text{amyloidosis} \text{ ?} \]

“All of the sudden a ____ jumped from a tree to chase a mouse.”

Weights of a Neural Network are the embeddings. Similar words embedded closer. Dissimilar farther.

\(^5\)“word2vec”, Mikolov et al., NIPS 2014
What’s so special about Riemannian geometry - curvature

**Negative Curvature**
Triangle angles add up to **less** than 180°

**Zero Curvature**
Triangle angles add up to **exactly** 180°

**Positive Curvature**
Triangle angles add up to **more** than 180°
Model of hyperbolic geometry
Properties of hyperbolic geometry
Computing lengths in hyperbolic geometry

Radius 1

Radius 2

Radius 3

Objects
- ID: line-1
  Start: (0.649, 0.746)
  End: (0.609, 0.706)
  Hyperbolic Length: 1.791

- ID: line-2
  Start: (0.074, 0.166)
  End: (-0.536, -0.319)
  Hyperbolic Length: 1.783
Approximation of graph distance