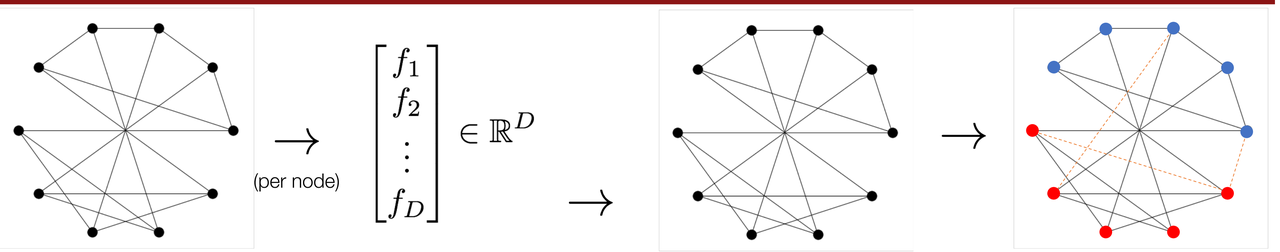
Learning Hyperbolic Representations in Real-World Networks

Manan Shah and Sagar Maheshwari
{manans, msagar}@stanford.edu

Overview



Step 1: How can we convert graphs into node-level vector representations?

Step 2: How can we use node representations for node classification and link prediction?

Our work develops a **supervised hybrid hyperbolic embedding framework** to approach embedding tasks (*step 1*) for arbitrarily complex graphs. We further generate models for node classification and link prediction provided node-level embeddings (*step 2*), and we evaluate our model on numerous real-world datasets, presenting numerical results and embedding visualizations. Our results indicate that **our hyperbolic embeddings vastly outperform traditional Euclidean embeddings** on both node classification and link prediction tasks. We further analytically compare the distribution of generated embeddings to conclude that hyperbolic embeddings better encode hierarchical structure.

Embedding Generation

Supervised Hyperbolic Embeddings. In order to modify current generation of hyperbolic embeddings to incorporate node labels, we alter the sampling procedure described in [1] to only generate negative samples between differently labeled nodes. We retain the Riemannian SGD update rule, where ∇_E represents the Euclidean gradient of the loss function.

$$\boldsymbol{\theta}_{t+1} \leftarrow \text{proj}\left(\boldsymbol{\theta}_t - \eta_t \frac{(1-\|\boldsymbol{\theta}_t\|^2)^2}{4}\nabla_E\right)$$

In particular, our novel supervised loss function solely incorporates updates from examples of different classes, so that

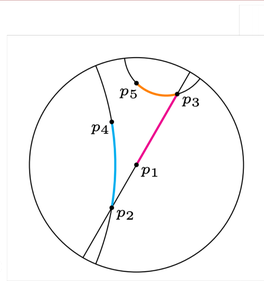
$$\mathcal{L}(\Theta) = \sum_{(u,v)\in\mathcal{D}} \log \frac{e^{-d(\boldsymbol{u},\boldsymbol{v})}}{\sum_{\boldsymbol{v}'\in\mathcal{N}(u)} e^{-d(\boldsymbol{u},\boldsymbol{v}')}}$$

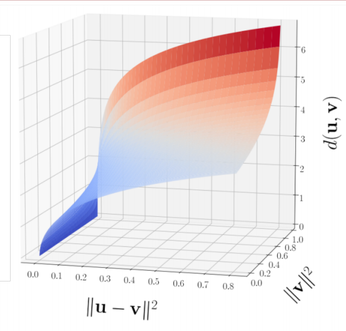
is defined with $\mathcal{N}(u) = \{v \mid (u,v) \notin E, c(u) \neq c(v)\} \cup \{u\}$ where c denotes a mapping from node to class.

Hyperbolic-Euclidean Embedding Fusion. For our embedding representations to include structural features from both Euclidean and hyperbolic space, we define a fusion procedure to combine hyperbolic and Euclidean generated vectors via Hadamard product and simple concatenation. The final vectors consist of learned representations from both paradigms.

Hyperbolic Embeddings

- Prior methods for node embeddings
  - DeepWalk uses unbiased random walks to generate node embeddings
  - node2vec uses biased 2nd order random walks to generate embeddings
- Recent work suggests node embeddings in hyperbolic space improve performance for networks with latent hierarchies
  - Poincaré models generate node embeddings in the n-dimensional Poincaré ball and use Riemannian Stochastic Gradient Descent to find optimal embeddings





In our work, we build upon existing unsupervised Poincaré node embedding frameworks to develop a supervised hybrid embedding framework. By utilizing representations obtained in both Euclidean and hyperbolic spaces, **our learned representations more effectively represent node-level hierarchies and transitive closure.**

Node Classification and Link Prediction

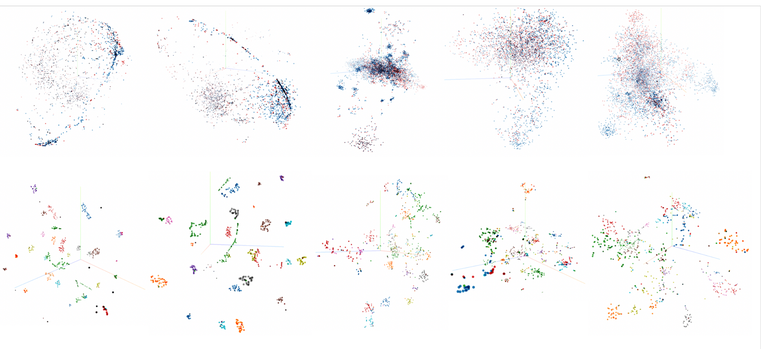
Node Classification. In order to generate classification outcomes after embeddings were learned, we trained **logistic regression, random forest, and support vector machine classifiers** on node embeddings, selecting the node label from our taxonomy associated with the highest probability. *5*-fold cross validation was performed to evaluate classification performance, so that 80% of nodes were used to train and 20% of nodes were used to evaluate at each iteration.

Link Prediction. To generate link prediction outcomes after embeddings were learned, we split our graph into training, validation, and test sets which can be viewed as versions of the graph different time intervals (where we wish to train on edges in time interval 0 and predict future edges to appear in time interval 1). We generate positive training examples by sampling from edges that exist in the graph, and negative examples from edges that are missing between nodes so that there is a perfect class balance between positive and negative examples. Embeddings are evaluated on the test set using **random forest** and **logistic regression classifiers**.

Datasets. We evaluate our framework on two real-world graphs with radically differing structure from the SNAP data hub.

- **Email EU Core.** This network was generated using email data from a large European research institution, where emails (edges) represent communication between members (nodes). The graph contains 1,005 nodes and 25,571 edges, and each node is labeled with the organization of the members of the organization to generate 42 classes.
- **CHG Miner.** This network represents a drug-target interaction network containing information on which genes are targeted by drugs on the US market. The graph contains 7,341 nodes and 15,138 edges, and each node is labeled with its class as either a drug or a target to generate 2 classes.

Experimental Evaluation



	Node Classification (Accuracy)		Link Prediction (Average Precision)	
	email-EU-core Train: 804 Test: 201	ChG-Miner Train: 5872 Test: 1468	email-EU-core Train: 19278 Test: 9638	ChG-miner Train: 18170 Test: 9082
DeepWalk	0.741	0.669	0.55 / 0.55	0.55 / 0.55
node2vec	0.761	0.693	0.57 / 0.55	0.57 / 0.56
Poincare	0.905	0.686	0.49 / 0.51	0.51 / 0.51
Hybrid	**0.915**	**0.710**	**0.57 / 0.56**	**0.58 / 0.57**

Above: Metrics for node classification and link prediction. Note that 5-fold CV metrics are reported for node classification, and train / test metrics are reported for link prediction.

Left: visualizations for embedding methods. (a) represents hyperbolic with burn-in, (b) represents hyperbolic without burn-in, (c) represents DeepWalk, (d) represents graph factorization, and (e) represents node2vec.

Conclusions. Our results indicate that our supervised hyperbolic embeddings vastly outperform traditional methods on both node classification and link prediction, indicating that leveraging graph hyperbolic structure provides significant benefits for overall performance. In particular, node classification results were bolstered by over 2 percent on both email-EU-core and ChG-miner, indicating that the incorporation of multiple unique aspects of graphical structure allowed for hybrid embeddings to excel in both cases. We further note from embedding PCA and t-SNE visualizations that hyperbolic embeddings closely model the Poincaré ball structure, as expected from the retraction update in our training procedure.

Future Work. In the future we hope to extend our work to more diverse and large datasets to further verify the benefits of hybrid embeddings on differing graphical structures. We further hope to identify more advanced methods of embedding fusion that may yield improved results.

[1] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In Advances in neural information processing systems, pages 6338–6347, 2017.