Evaluating the Quality of Graph Embeddings via Topological Feature Reconstruction

Stephen Bonner*, John Brennan*, Ibad Kureshi*, Georgios Theodoropoulos, Stephen McGough, Bogulsaw Obara*

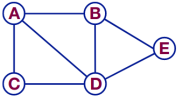
*Department of Computer Science, Durham University

December 11, 2017

Introduction

- A graph is a **representation of data** - *useful at capturing intrinsic links or relationships.*
- A graph $G = (V, E)$ set of vertices V and edges E.
- **Vertices:** an entity (person, object or document). **Edges:** relationship or link between two vertices.
- A graph's structure is called it's *topology*.
- **Can represent:** social media networks, road networks, citation networks, web hyper-link graphs, semantic web data and protein interaction networks.

Graph: $G = (\mathcal{V}, \mathcal{E})$



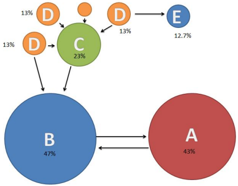
Adjacency matrix: $\mathbf{A}$

$$
\begin{array}{c}
\\ A \\ B \\ C \\ D \\ E
\end{array}
\begin{array}{c}
\begin{array}{ccccc} A & B & C & D & E \end{array} \\
\begin{pmatrix}
0 & 1 & 1 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 0
\end{pmatrix}
\end{array}
$$

Durham University

Graph Analysis

- The field of graph analysis is all about mining and making sense of these complex graph structures!
- Problems areas in network science include: **Link Prediction**, **Vertex Centrality Measurement**, **Classification**, **Community Detection** and **Temporal Evolution**.
- Traditionally these have been performed by custom graph-based algorithms - *Have issues with generality (domain specific algorithms) and scalability (graph focused parallel libraries).*



Research Problems

Analysing graph via machine learning faces some challenging issues:

- **Lack Of Data** - Compared to other ML-based fields, there is lack of *(labelled)* data!
- **Lack Of Benchmarks** - There is no *MNIST* or *CIFAR* for graphs! *Hard to compare results.*
- **Data Size** - Graphs can be massive - 10^8 vertices easily!
- **Data Representation** - How best to represent the graph itself before performing ML? *Traditionally performed using hand-crafted features.*

Durham University

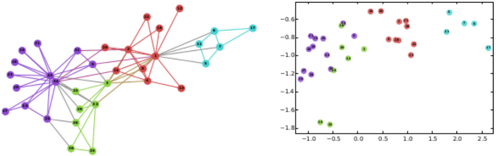
Graph Embeddings

Durham
University

Graph Embeddings

Graph Embeddings have recently become a popular method for analysing graph using *Machine Learning* - specifically using **Neural Networks**.

- Aim to solve the problem of graph data representation by mapping the graph into vector space - $f : V \to \mathbb{R}^d$
- Aim to capture as much information as possible from the *topology*.
- Can be **Supervised** and **Unsupervised**.



(a) Input: Karate Graph (b) Output: Representation

Evaluating Embedding Quality

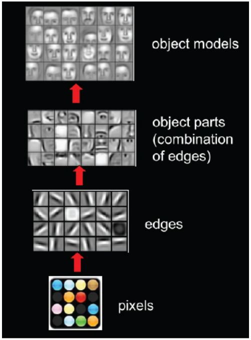
Key Idea: We propose to measure embedding quality by *predicting* a series of known **vertex-level topological features** from graph embeddings.

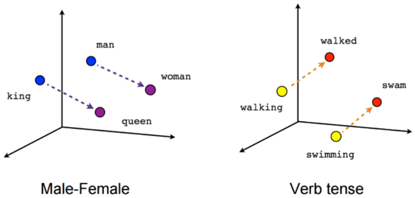
- Existing approaches do not necessarily mean that embedding approaches are measured against their ability to reconstruct the topology.
- Predicting topological features is a *task independent* way of measuring this!
- We turn the task into **Classification** by **binning** the features via a histogram.
- We provide a framework for future embedding approaches to measure their quality.

Durham University

What Do Embeddings Learn?

In other fields it's clear what *neural networks are learning:*





However: Not so clear with graph embeddings!

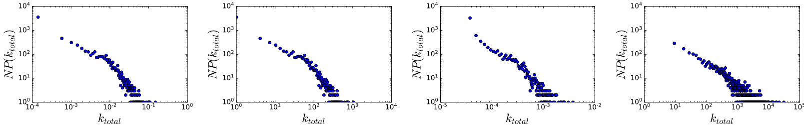
Evaluating Embedding Quality

Table 1: Topological Features for Measuring Embedding Quality.

Feature Name	Equation
PageRank Score (PR)	$PR(v) = \frac{1-d}{N} + d \sum_{u \in \Gamma^{-}(v)} \frac{PR(u)}{d^{+}(u)}$
Degree Centrality (DC)	$DC(v) = \frac{1}{\lvert V \rvert} \Gamma^{-}(v) + d^{+}(v)$
Local Clustering Score (CLU)	$CLU(v) = \frac{2\Phi}{d^{+}(v)(d^{+}(v)-1)}$
Number Of Triangles (TR)	$TR(v) = \Phi$

Unbalanced Distribution Of Features

Many graph features are *highly unbalanced:*



(a) Degree Centrality

(b) Degree

(c) Page Rank

(d) Triangle Count

Figure 1: Distribution of Topological Feature Values from the Wiki-Vote Dataset in Log Scale: (a) Degree Centrality Distribution, (b) Total Vertex Degree Distribution, (c) Page-Rank Distribution, (d) Distribution of Number Of Triangles for each Vertex.

Durham University

Graph Embedding Approaches Considered

Graph Embeddings

We select three state-of-the-art unsupervised neural network based graph embedding approaches:

- **Node2vec** - Sequences of vertices generated by random walks and fed into the Skip-gram model, with user controlled parameters for biasing the random walk *(Grover et al., KDD 2016)*.
- **Hyperbolic Poincaré Disk** - Similar to Node2Vec but using Hyperbolic geometry *(Chamberlain et al., MLG 2017)*.
- **Structural Deep Network Embedding** - Embeds vertices using a deep auto-encoder directly on the adjacency matrix *(Wang et al., KDD 2016)*.

Durham University

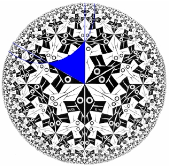
Graph Embeddings - Random Walks

Both **Node2Vec** and **Hyperbolic Poincaré Disk** are random walk based models, essentially optimising:

$$\frac{1}{|V|}\sum_{t=1}^{|V|}\sum_{i=0}^{n}\sum_{-c\leq j\leq c, j\neq 0}\log \mathbf{P}(w_{i+j}^{t}|w_{i}^{t}), \tag{1}$$

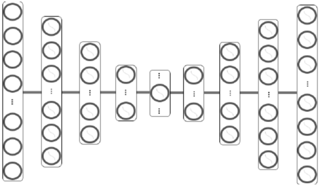
Intuition: vertices with similar neighbourhoods get similar embeddings.

Hyperbolic Poincaré Disk uses a hyperbolic space for the embedding - meaning that the embedding vector describes a point on a hyperbolic disc.



Graph Embeddings - Auto-Encoder

Structural Deep Network Embedding is based on the use of a deep *auto-encoder*.



A two part, *graph specific loss function* is created to produce the embeddings.

Intuition: vertices with similar first and second order structures get similar embeddings.

Experimental Evaluation

Experimental Setup

- All three approaches were implemented in *TensorFlow* and run on an NVIDIA Tesla K40.
- Embeddings were classified against varying amounts of training data using a *Logistic Regression.*
- Reported results are the mean of five replicated experiment, with the metrics being *Macro-F1* and *Micro-F1*.

Table 2: Key Hyper-Parameter Settings

Approach	**Opt**	**LR**	**Misc**
SNDE	RMSProp	0.01	α=500, b=10, epochs=500
Node2Vec	SGD	1.0	p=0.5, q=2, epochs=5
Poincaré Disk	SGD	0.1	”

Durham University

Datasets

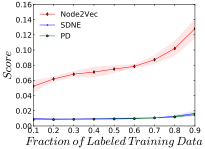
For evaluating embedding quality, we used a selection of dataset from different domains all taken from SNAP.

Table 3: Graph Datasets Used.

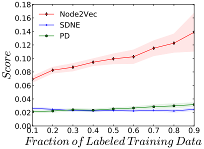
Dataset	$\|V\|$	$\|E\|$	*domain*
Ca-HepPh	12,008	118,521	*Collaboration*
ego-Facebook	4,039	88,234	*Social*
p2p-Gnutella04	10,876	39,994	$Peer - to - peer$
wiki-Vote	7,115	103,689	*Wiki*

Durham University

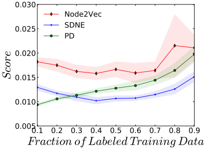
Results - Predicting Degree Centrality



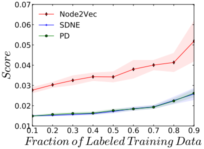
(a) Macro DC ca-HepPh



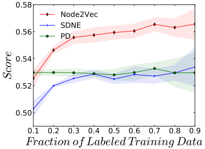
(b) Macro DC facebook-combined



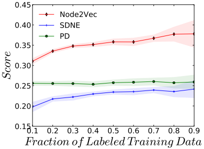
(c) Macro DC p2p-Gnutella04



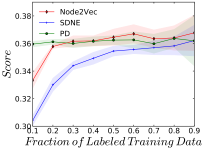
(d) Macro DC wiki-Vote



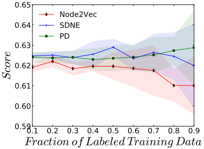
(e) Micro DC ca-HepPh



(f) Micro DC facebook-combined



(g) Micro DC p2p-Gnutella04



(h) Micro DC wiki-Vote

Figure 2: Predicting a Vertex's Degree Centrality (DC) Value.

Durham University

Results - Predicting Degree Centrality Wiki-Vote

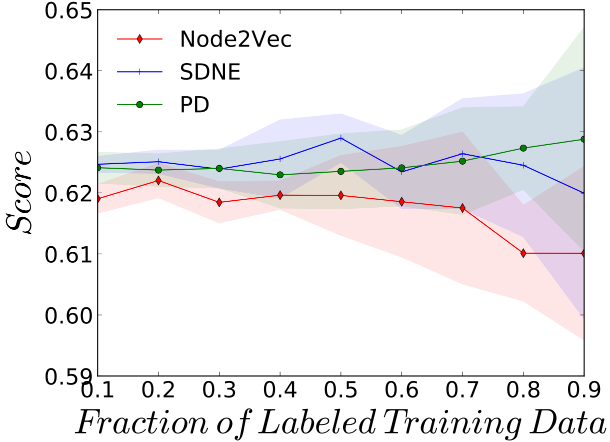
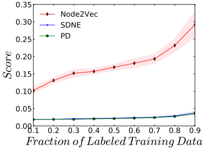


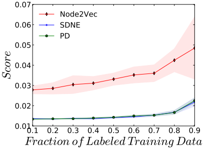
Figure 3: Micro-f1 Score Degree Centrality on Wiki-Vote

Durham University

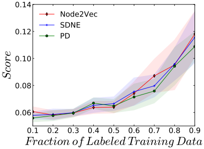
Results - Predicting Triangle Count



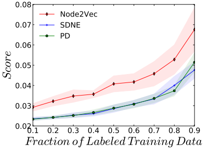
(a) Macro TR ca-HepPh



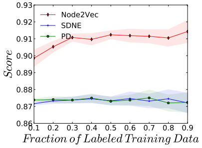
(b) Macro TR facebook-combined



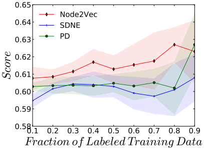
(c) Macro TR p2p-Gnutella04



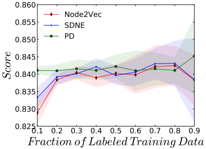
(d) Macro TR wiki-Vote



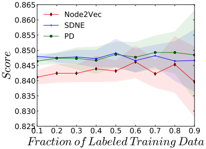
(e) Micro TR ca-HepPh



(f) Micro TR facebook-combined



(g) Micro TR p2p-Gnutella04

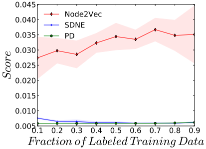


(h) Micro TR wiki-Vote

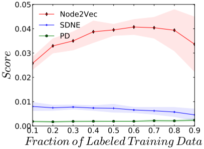
Figure 4: Predicting a Vertex's Triangle Count (TR).

Durham University

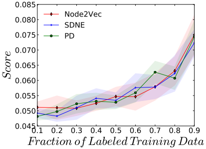
Results - Predicting Local Clustering Score



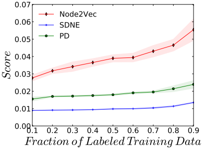
(a) Macro CLU ca-HepPh



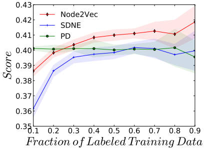
(b) Macro CLU facebook-combined



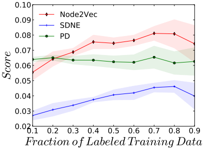
(c) Macro CLU p2p-Gnutella04



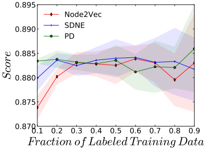
(d) Macro CLU wiki-Vote



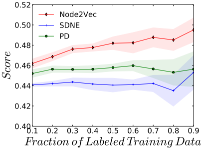
(e) Micro CLU ca-HepPh



(f) Micro CLU facebook-combined



(g) Micro CLU p2p-Gnutella04

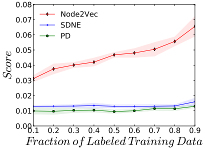


(h) Micro CLU wiki-Vote

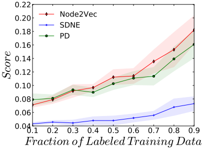
Figure 5: Predicting Vertex's Local Clustering (CLU).

Durham University

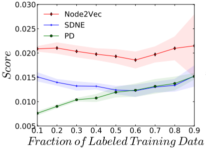
Results - Predicting PageRank



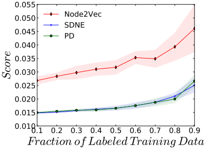
(a) Macro PR ca-HepPh



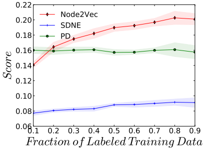
(b) Macro PR facebook-combined



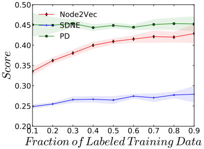
(c) Macro PR p2p-Gnutella04



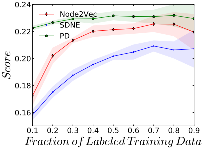
(d) Macro PR wiki-Vote



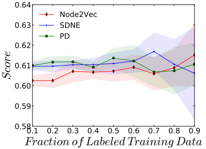
(e) Micro PR ca-HepPh



(f) Micro PR facebook-combined



(g) Micro PR p2p-Gnutella04



(h) Micro PR wiki-Vote

Figure 6: Predicting a Vertex's PageRank (PR).

Durham University

Results - Predicting PageRank

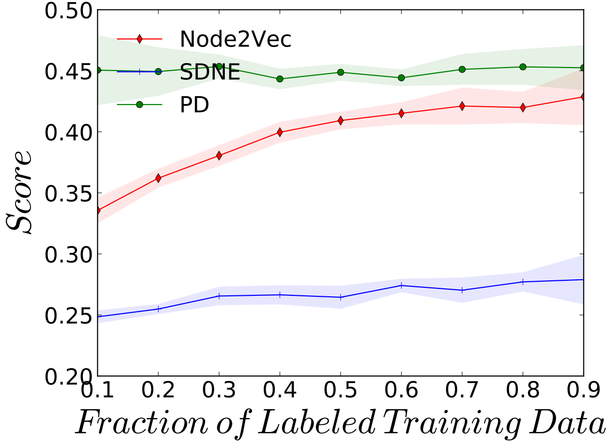


Figure 7: Micro-f1 Score Predicting PageRank on Facebook

Durham University

Conclusions

- **We propose a frame for assessing embedding quality via topological feature reconstruction.**
- This work seems to suggest that traditional graph features are *not always being learned.*
- No consistent pattern across datasets and features - *Different features seem to be learned across datasets.*
- *Node2Vec* seems best able to reconstruction topological from the embedding process.
- *Hyperbolic* approach performs well considering it is limited to 2D.
- **Future Work:** aim to expand the features we are analysing and see if optimising to predict features increases general performance.

Durham University

Thank You!

Durham University