A Deep Learning Approach to Link Weight Prediction

Yuchen Hou

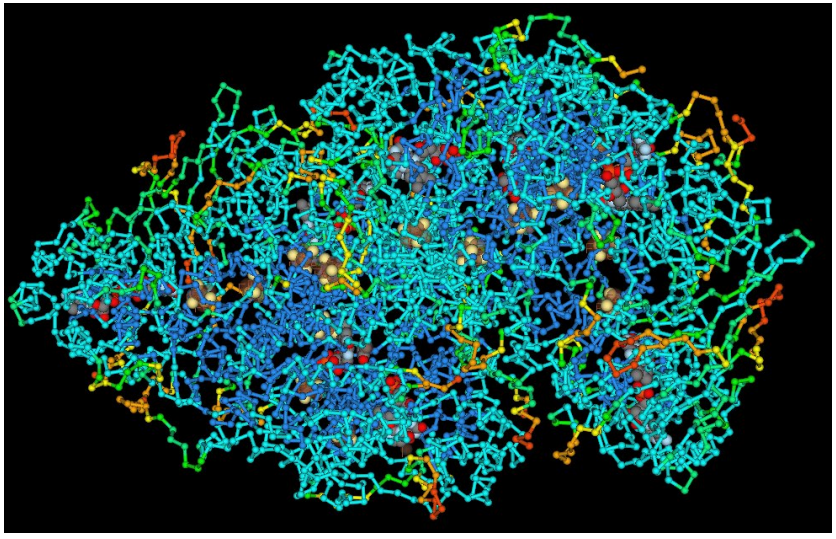
WASHINGTON STATE UNIVERSITY

Introduction: deep learning in different application domains

- Image recognition
- Speech recognition
- Natural language processing
- Graph mining

Background: graph mining problems

(a) Link prediction: predicting user connectivities in a social network.



(b) Graph classification: predicting the chemical activity of a macromolecule.

Figure: Example graph mining problems and their application scenarios.

Contribution: link weight prediction with deep learning

- The first deep learning approach to the link weight prediction problem.
- A unique supervised learning technique for node embedding.
- 73% more accurate than the state-of-the-art non deep learning approach.
- A generalized link weight prediction model using pretrained node embeddings.

Problem: link weight prediction

Problem example

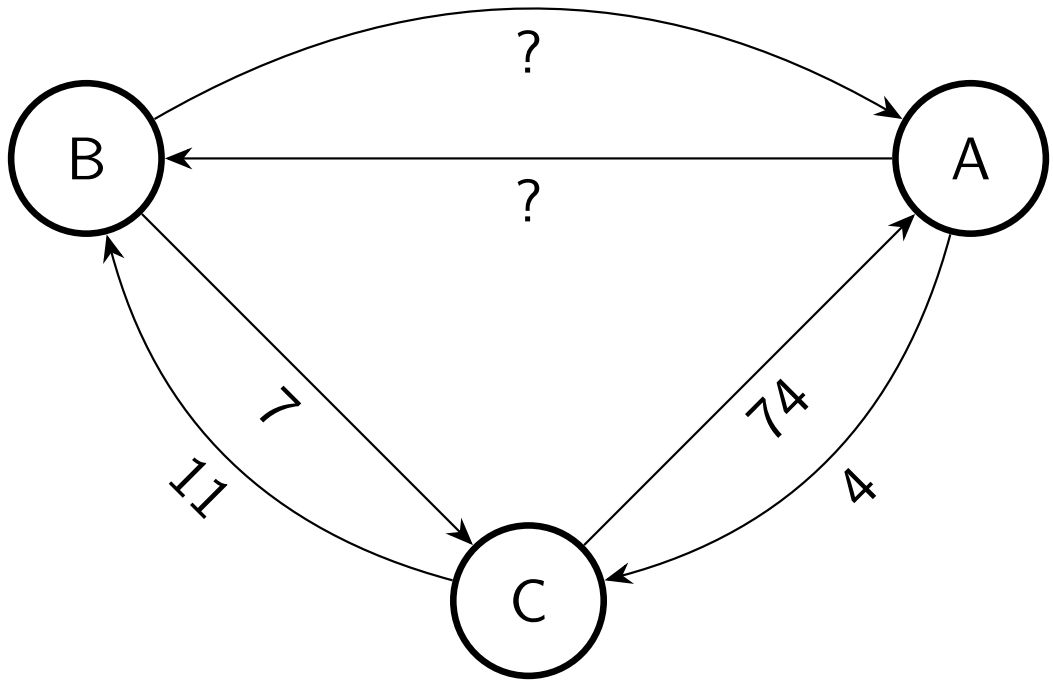


Figure: Message volume prediction in a social network of 3 users.

Problem: link weight prediction

Problem definition

- Given a weighted directed graph with the node set V and a link subset E
- Build a model weight = f(source, destination) to predict the weight of any link (source, destination) $\notin$ E

The state-of-the-art approach

pWSBM (pure Weighted Stochastic Block Model)

- Partition nodes into node groups of topologically similar nodes.
- Connect groups with bundles with normal weight distributions.
- A bundle represents all links connecting the two groups.
- A link has the same weight distribution as the bundle representing it.

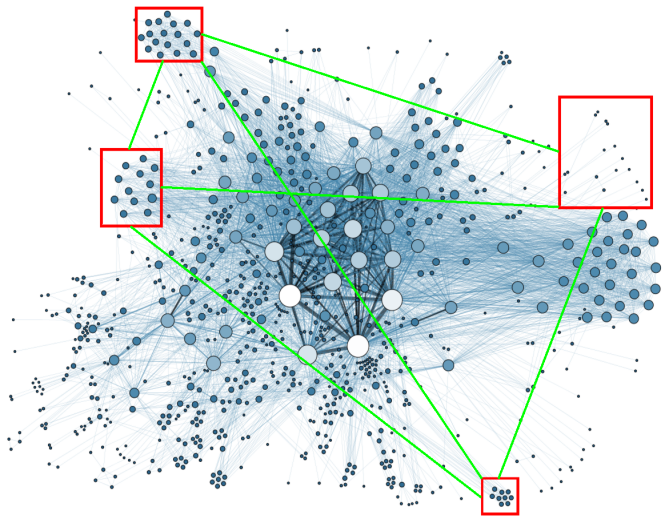


Figure: pWSBM approach to link weight prediction.

The state-of-the-art approach

pWSBM (pure Weighted Stochastic Block Model)

The SBM has the following parameters:

- z: the node group label vector
- μ: the bundle weight expectation matrix
- σ: the bundle weight standard deviation matrix

The weight of each link (i, j) A_{ij} has normal distribution:

$$A_{ij} \sim N(\mu_{z_i z_j}, \sigma^2_{z_i z_j})$$

The pWSBM fits parameter z, μ and σ to maximize the log likelihood of observation A:

$$\log(P(A|z, \mu, \sigma)) = \sum_{ij} (\frac{\mu_{z_i z_j}}{\sigma^2_{z_i z_j}} - \frac{1}{2\sigma^2_{z_i z_j}} - \frac{\mu^2_{z_i z_j}}{\sigma^2_{z_i z_j}})$$

Motivation: Skip-gram model

Model architecture

- word2vec: map words in sentences to vectors.
- item2vec: reduce orders (lists of items) to sentences.
- node2vec: reduce paths (lists of nodes) to sentences.

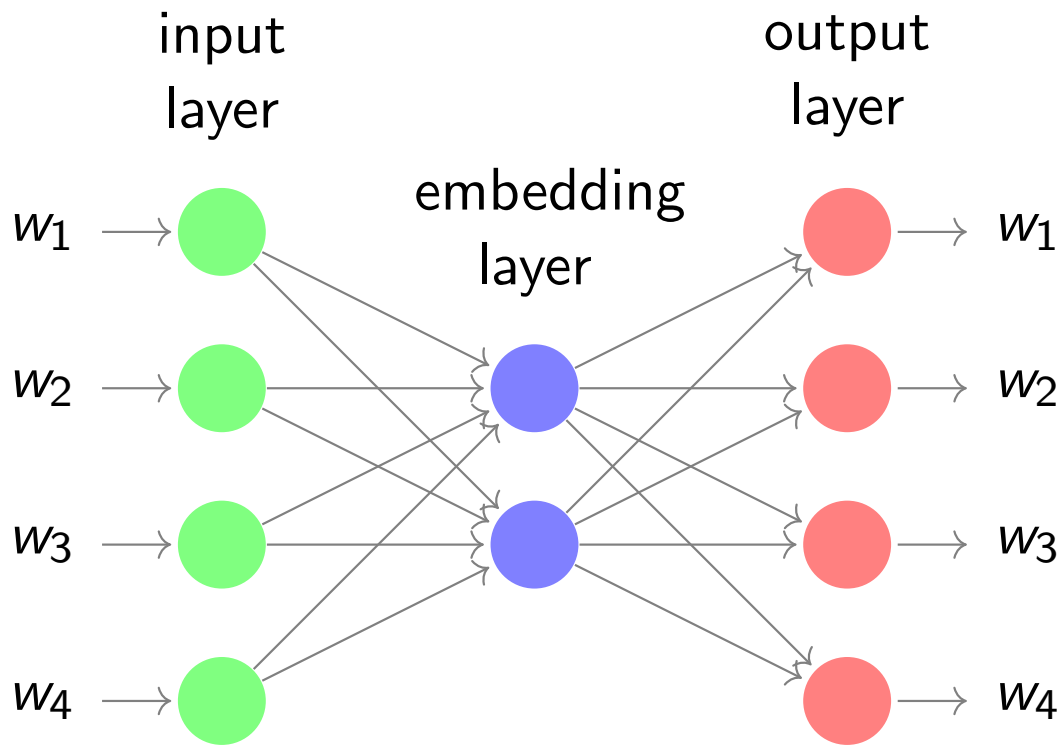


Figure: The skip-gram model with vocabulary size 4 and embedding size 2.

Motivation: Skip-gram model

Datasets

- Sentence: the quick brown fox jumps over the lazy dog
- Context radius: 2
- Training example: (word, context-word) pairs

Table: The words dataset for a natural language corpus.

Input = word	**Output = context-word**
fox	quick
fox	brown
fox	jumps
fox	over
jumps	brown
jumps	fox
...	...

Deep learning approach: Model R (R as in "relation")

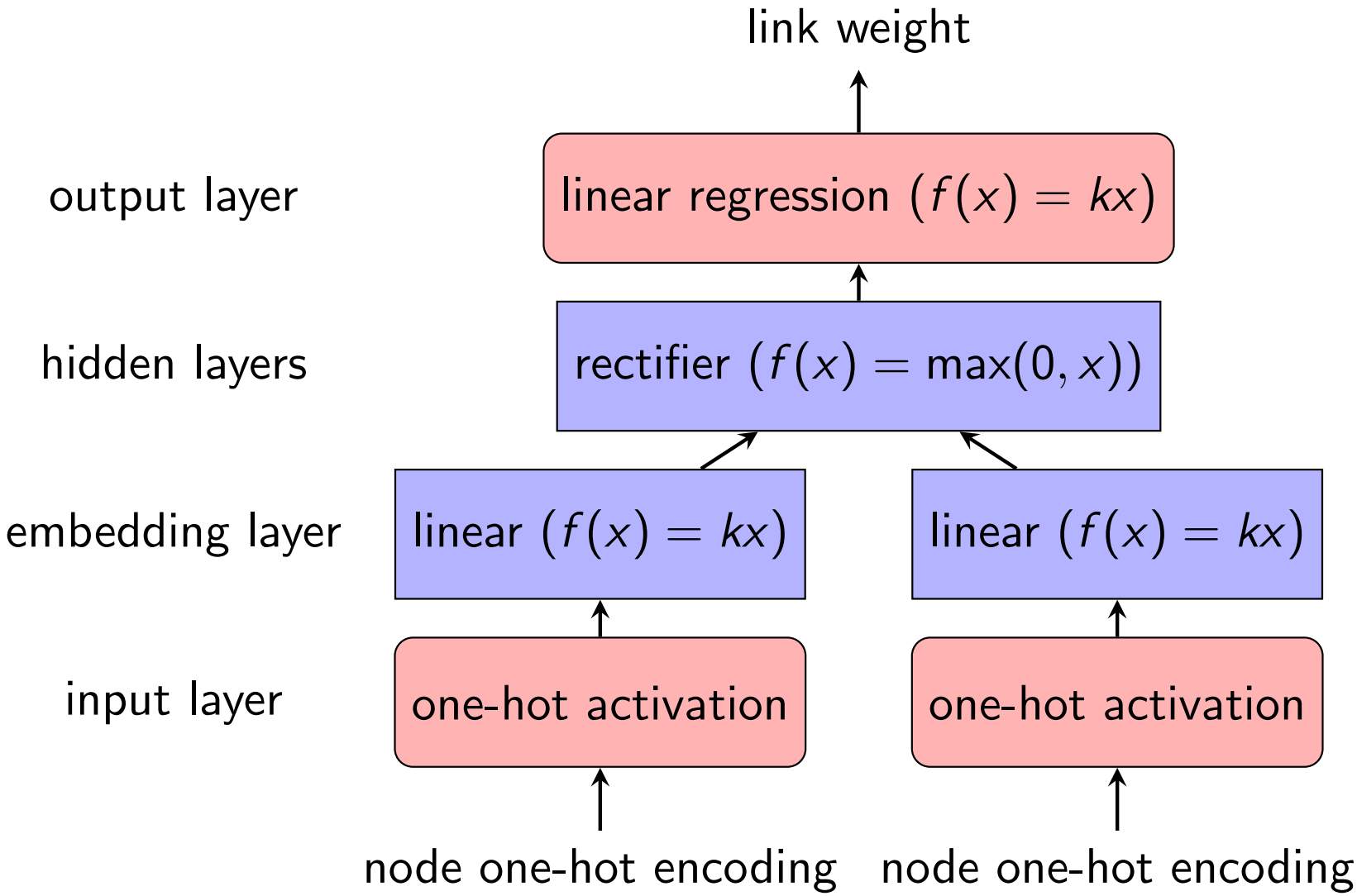


Figure: Model R with multiple hidden layers.

Comparison: pWSBM vs Model R

Table: The advantages Model R has over pWSBM in several aspects.

Aspect	**pWSBM**	**Model R**
model granularity	node group level	node level
distribution assumption	normal distribution	NA
model flexibility	low	high

Experiments

Baseline approaches

- SBM (Stochastic Block Model)
- pWSBM (pure Weighted Stochastic Block Model)
- bWSBM (balanced Weighted Stochastic Block Model)
- DCWBM (Degree Corrected Weighted Stochastic Block Model)

Experiments

Datasets

Table: The datasets used in experiments with weights scaled to [0, 1].

Dataset	**Nodes**	**Link weights**
Airport	airports	passengers delivered between airports
Collaboration	nations	paper collaborations between nations
Congress	committees	members shared between committees
Forum	users	messages exchanged between users

Experiments

Settings

- 25 independent trials on each dataset
- training set: 70%
- validation set: 20%
- testing set: 10%

Experiments

Results

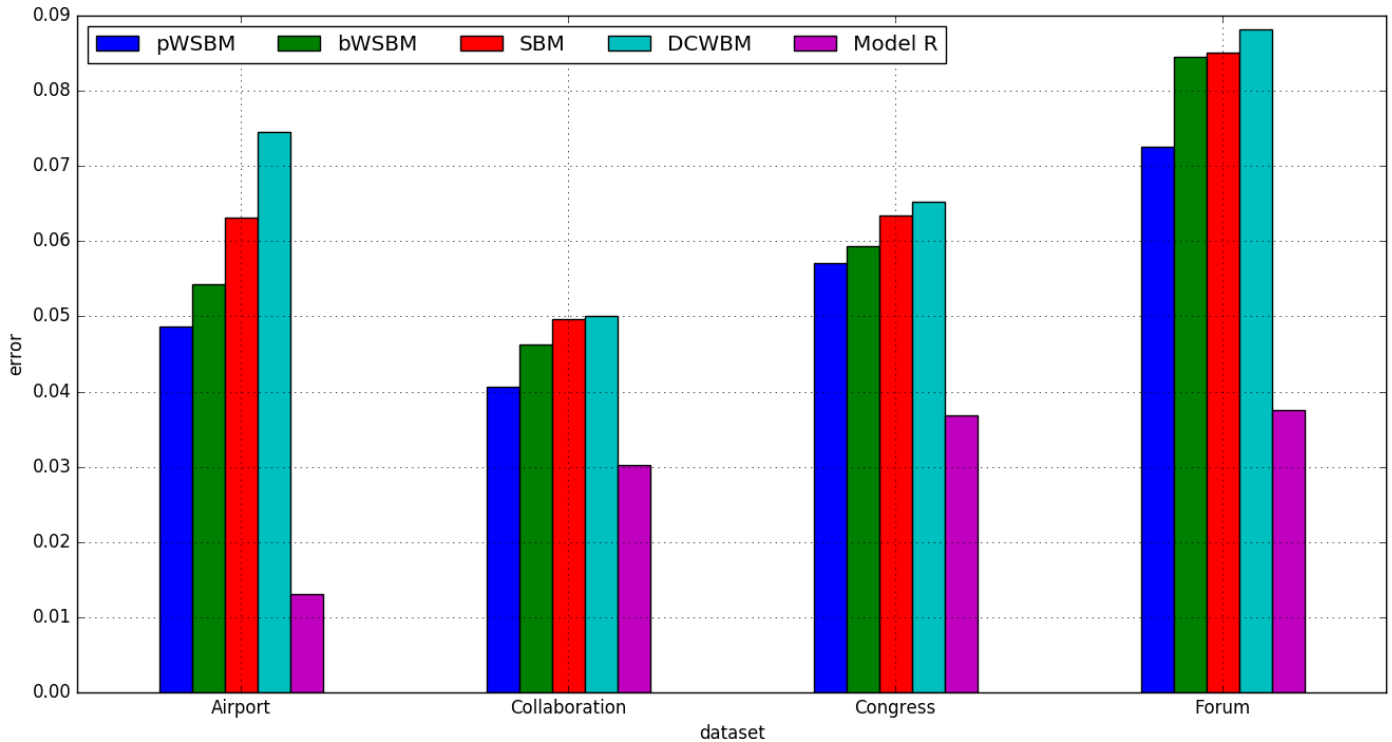


Figure: Model R has lower mean squared errors than 4 baseline approaches over 4 datasets consistently.

Node embedding analysis

Motivation

- Question: What knowledge does Model R learn?
- Hypothesis: It learns meaningful node embeddings (similar nodes are closer).
- Related work: Word2vec word embedding analysis.

Node embedding analysis

Dataset: MovieLens 100K

- Well understood domain: movie recommendation
- 100,000 ratings from 1000 users on 1700 movies.
- Bigraph: users and movies are nodes; ratings are link weights.

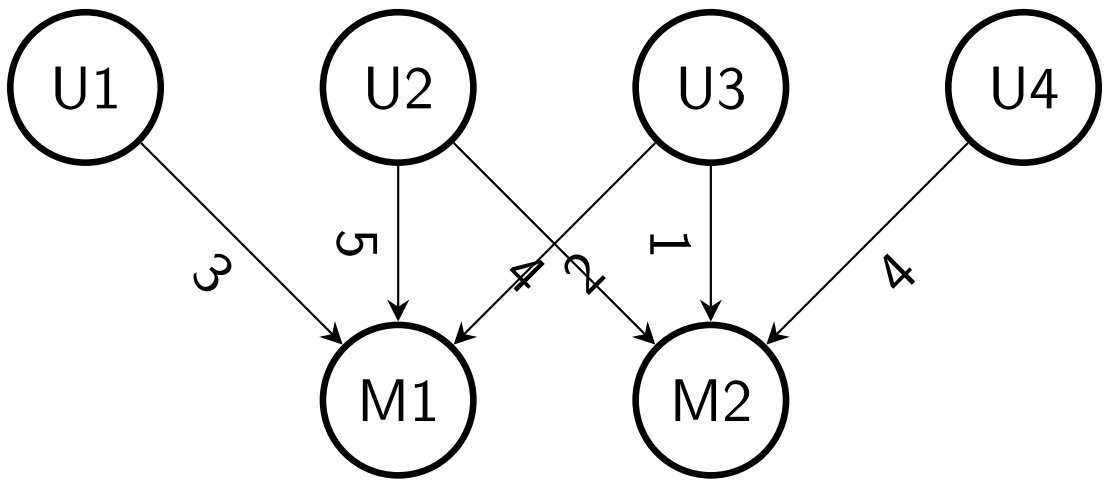


Figure: Movie rating illustration for a dataset of 4 users and 2 movies.

Node embedding analysis

Visualization

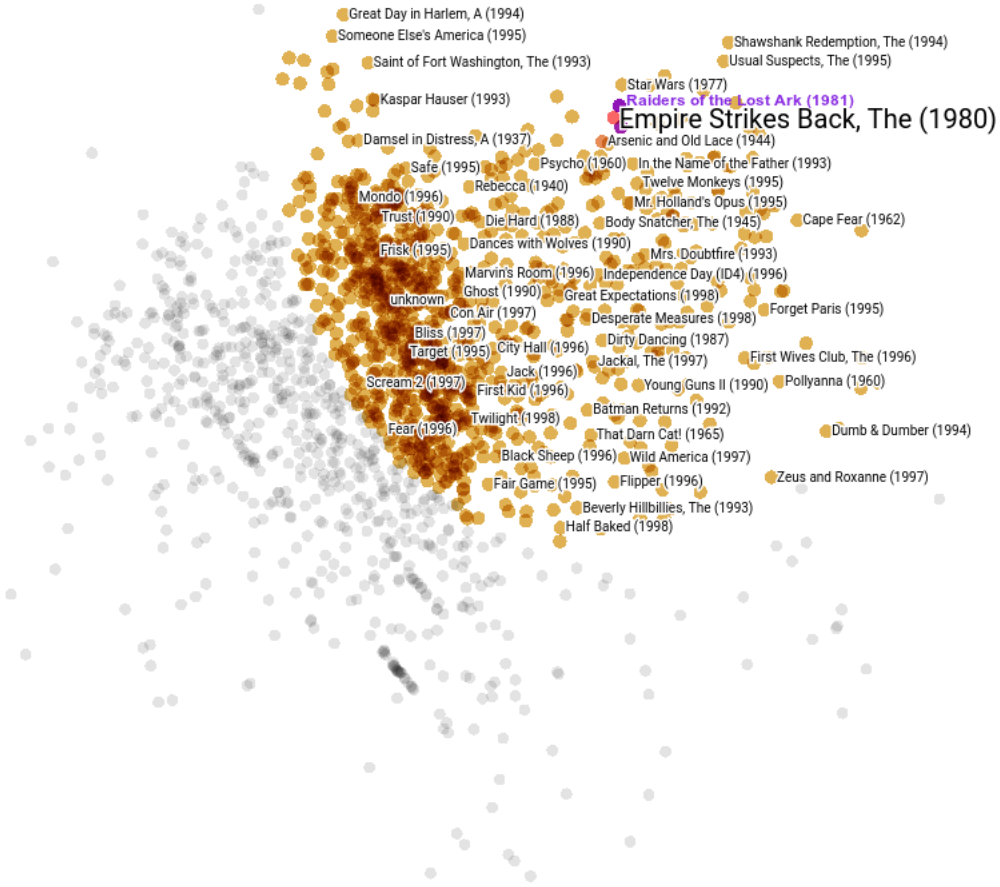


Figure: The embeddings of all movies in MovieLens 100K dataset. Dimensionality reduction of embeddings uses Principal Component Analysis.

Node embedding analysis

Distance and similarity

Table: The distances of movies to the reference movie for MovieLens 100K dataset. The distance from similar movies (Star Wars and Return of Jedi) to the reference movie (The Empire Strikes Back) are shorter than the median distance.

Movie	Distance	Similarity
The Empire Strikes Back (1980)	0	self (reference)
Raiders of the Lost Ark (1981)	0.012	most similar
Star Wars (1977)	0.047	more similar
Return of the Jedi (1983)	0.063	more similar
Children of the Revolution (1996)	0.256	median point
Tomorrow Never Dies (1997)	0.295	less similar
Ayn Rand: A Sense of Life(1997)	0.296	less similar
101 Dalmatians (1996)	0.335	least similar

Model S

Motivation

- Model S is an extension of Model R incorporating different types of embeddings.
- Decoupling node embedding learning and weight prediction learning.
- Investigating the effectiveness of other node embedding techniques.
- Adopting more advanced embedding techniques developed in the future.

Model S

Node embedding techniques

- LLE(locally linear embedding): nonlinear dimensionality reduction
- Node2vec: node embedding with skip-gram model
- Model R: node embedding learning supervised by link weight

Model S

LLE(locally linear embedding)

LLE is a manifold learning approach designed for dimensionality reduction consisting of 2 steps:

1. Linear approximation of data points X's in original space minimizing cost function:

$$cost(W) = \sum_i |X_i - \sum_j W_{ij} X_j|^2$$

2. Reconstruction of data points Y's in a low dimensional space minimizing cost function:

$$cost(Y) = \sum_i |Y_i - \sum_j W_{ij} Y_j|^2$$

Model S

Experiments

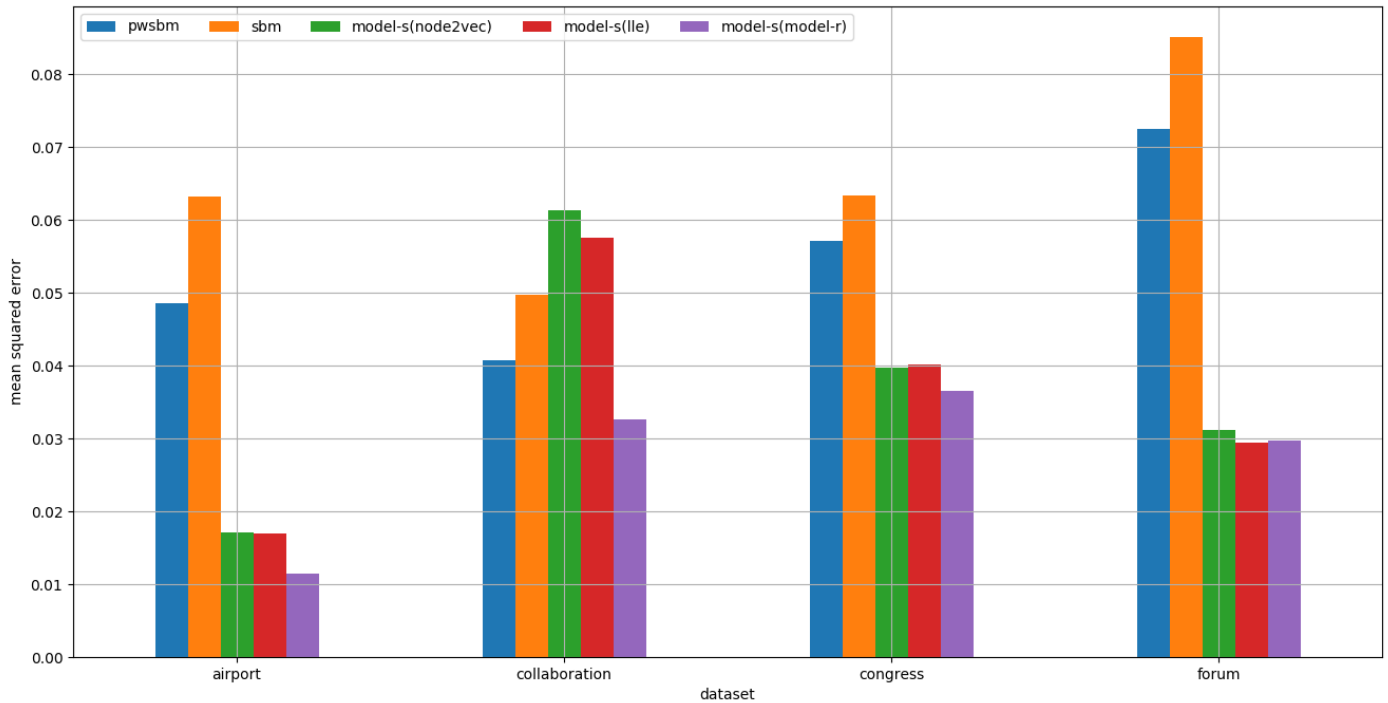


Figure: Model S with Model R embedding has the best overall performance. Model S generally performs better than SBM and pWSBM with 3 different embedding techniques (LLE, Model R and node2vec).

Conclusions

- Model R is more accurate than the state-of-the-art non deep learning approaches to the link weight prediction problem.
- Model R learns node embeddings and uses this information to predict unknown link weights.
- Deep learning can be successfully applied to link weight prediction problem.

Future work

- Unified node embedding: embedding nodes to only one space.
- Node embedding metrics: evaluation of embedding qualities.
- Complex graphs: taking advantage of rich node information.

Publications

- Comparative analysis of deep node embeddings for link weight prediction in graphs, TNNLS 2018 (in preparation)
- On graph mining with deep learning: introducing Model R for link weight prediction, JAISCR 2018
- Deep learning approach to link weight prediction, IJCNN 2017

Thank you!