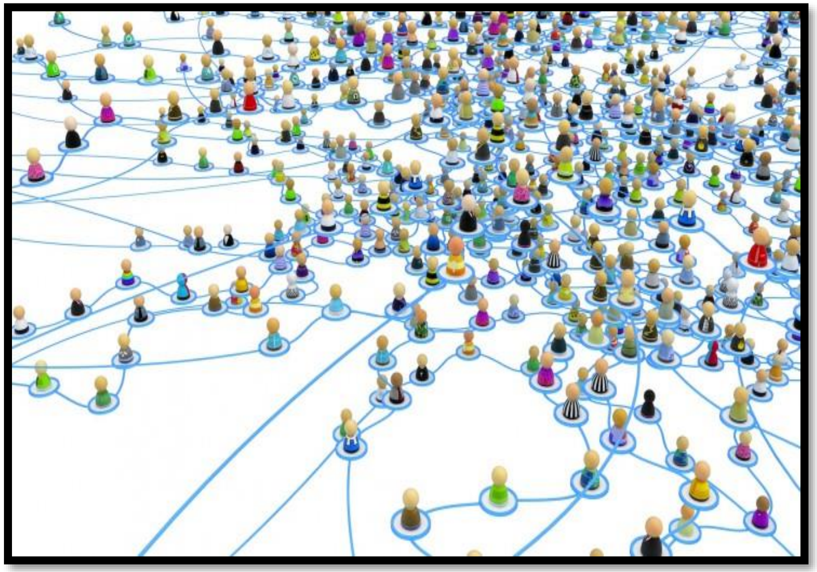
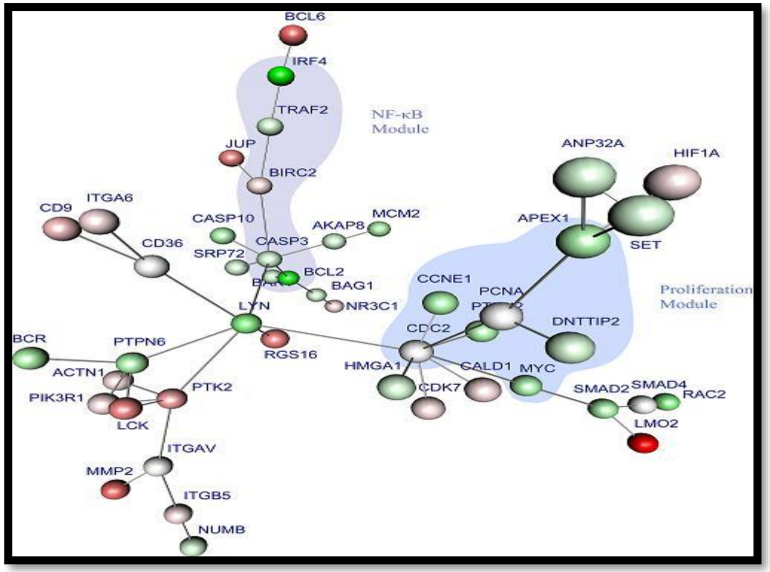
Deep Recursive Network Embedding with Regular Equivalence

Ke Tu	Peng Cui	Xiao Wang	Philip S. Yu	Wenwu Zhu
Tsinghua U	Tsinghua U	Tsinghua U	UIC&Tsinghua	Tsinghua U

Network Analytics



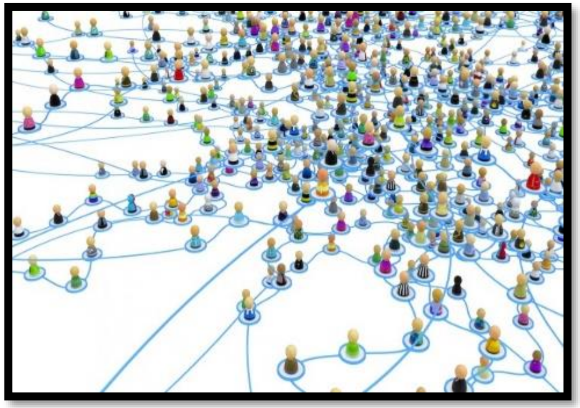


Social Networks

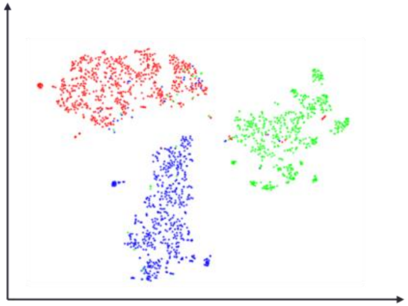
Biology Networks

Networks are widely used to represent the rich pairwise relationships of data objects

Network Embedding



Origin network

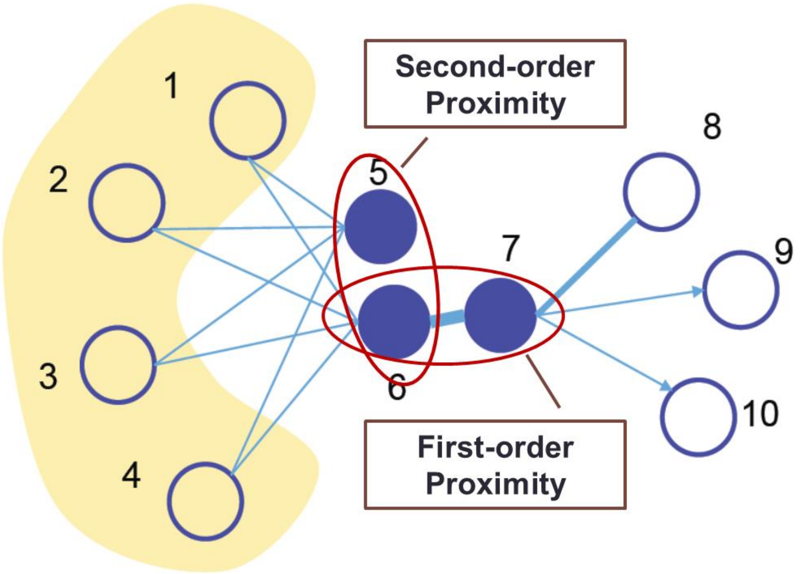


- in vector space

- ☐ Link Prediction
- ☐ Community Detection
- ☐ Node Classification
- ☐ Network Distance
- ☐ Node Importance
- ☐ …

Networks Embedding aims to learn a low-dimensional representation for each node

Existing Embedding Methods

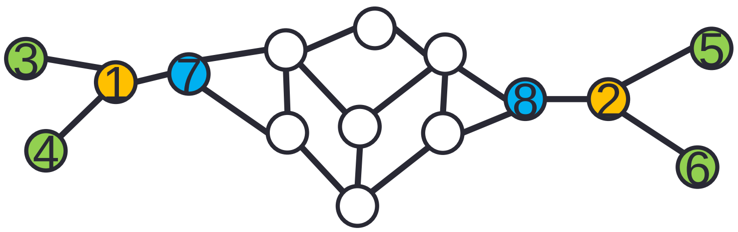


- By direct links(first-order) or common neighborhoods (second-order) between nodes
- They can only preserve local proximity (Structural equivalence), can not get the global position

Motivation

- Vertexes in different parts of the network may have similar roles(global position)

- Example：
  - Managers in the social network of a company
  - Outliers in a network in the task of anomaly detection

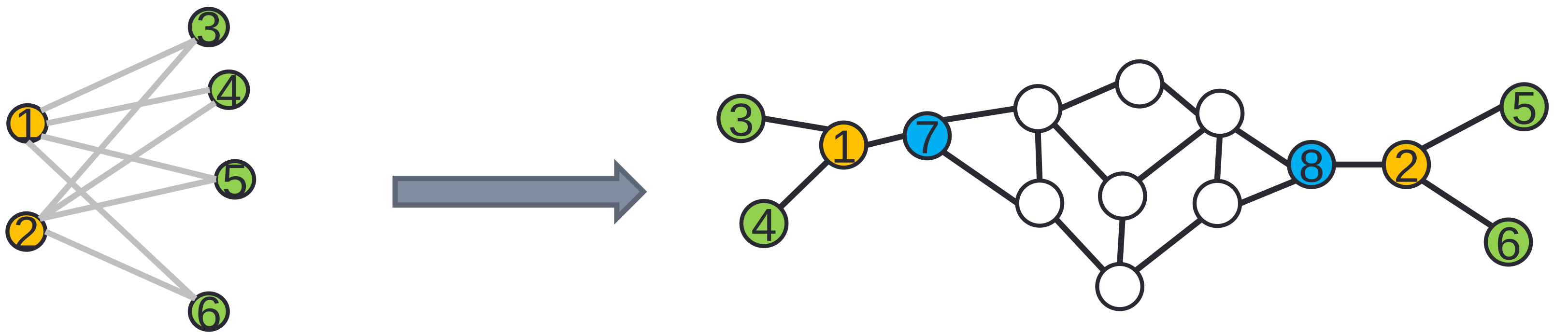


- Previous assumption of network embedding is not suitable for these node role-based tasks.

How to preserve the role or importance of a vertex in embedding space?

Regular Equivalence

Two vertexes are defined to be regularly equivalent if they have network neighbors which are themselves regularly equivalent.



- Structural equivalence s
  - $N(u) = N(v)$
  - Direct way
  - Common neighbors
- Regular equivalence r
  - $\{r(i) | i \in N(u)\} = \{r(j) | j \in N(u)\}$
  - Recursive way
  - Similar global position

We need to preserve Regular equivalence instead of Structural equivalence

Regular Equivalence

Basis: two regularly equivalent nodes should have similar embeddings

1. Explicitly calculate the regular equivalence of all vertex pairs
   - infeasible for large-scale networks
2. Replace regular equivalence into simpler graph theoretic metrics
   - Such as centrality measures
     - Only capture a specific aspect of network role
   - Some centrality measures also bear high computational complexity
     - Such as between centrality, closeness centrality

How to effectively and efficiently preserve regular equivalence in network embedding

Deep Recursive Network Embedding

- The definition of regular equivalence is recursive
  - Preserve the regular equivalence by aggregating neighbors' information in a recursive way

$$\mathcal{L}_1 = \sum_{v \in V} ||\mathrm{X}_v - Agg(\{\mathrm{X}_u | u \in \mathcal{N}(v)\})||_F^2,$$

- How to design the aggregating function
  - Variable length of neighbors
  - Highly nonlinear
  - → Agg = Layer-normalized LSTM

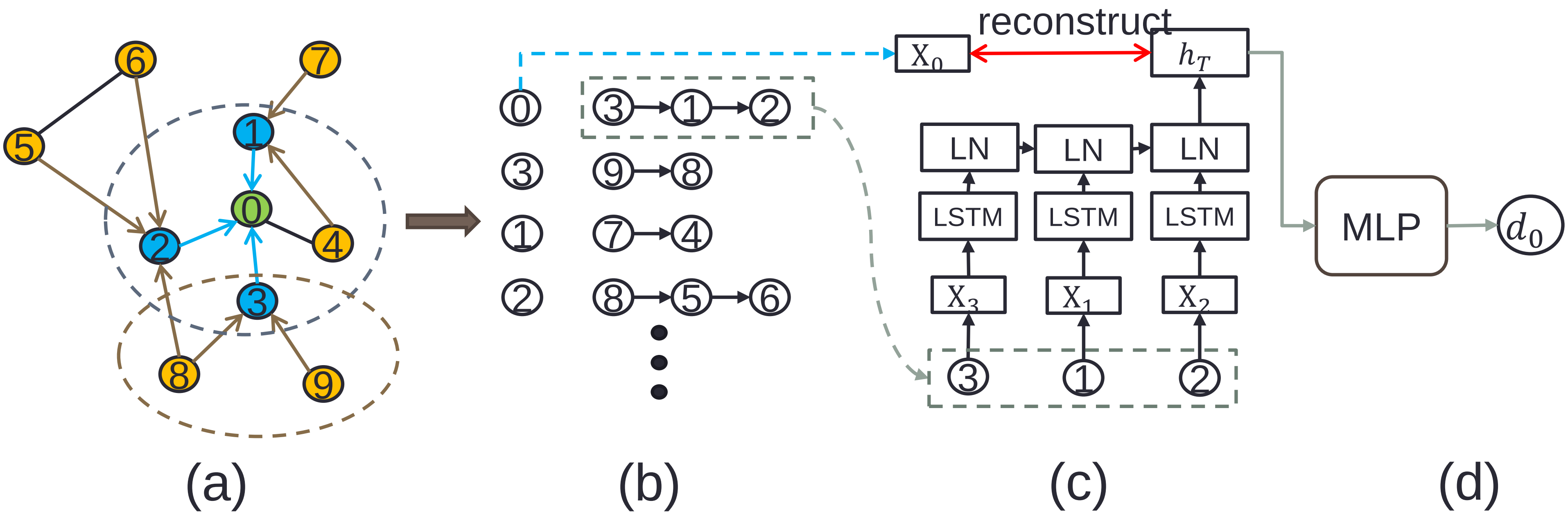
Deep Recursive Network Embedding

- Heavy-tailed distribution
  - Very long neighbor **sequence** → Neighborhood Sampling
  - Consider sample node by their degree
  - $P(v) \propto dv$

- The trivial solution
  - Regularization
  - Use node degree as the weakly guided information

$$\mathcal{L}_{reg} = \sum_{v \in V} \|\log(d_v + 1) - MLP(Agg(\{\mathrm{X}_u | u \in \mathcal{N}(v)\}))\|_F^2 ,$$

Deep Recursive Network Embedding



(a) Sampling neighborhoods
(b) Sorting neighborhoods by their degree
(c) Aggregate neighbors
(d) A Weakly guided regularizer

Theoretical Analysis

THEOREM 3.5. *If the centrality $C(v)$ of node v satisfies that $C(v) = \sum_{u \in \mathcal{N}(v)} F(u)C(u)$ and $F(v) = f(\{F(u), u \in \mathcal{N}(v)\})$ where f is any computable function, then $C(v)$ is one of the optimal solutions of our model.*

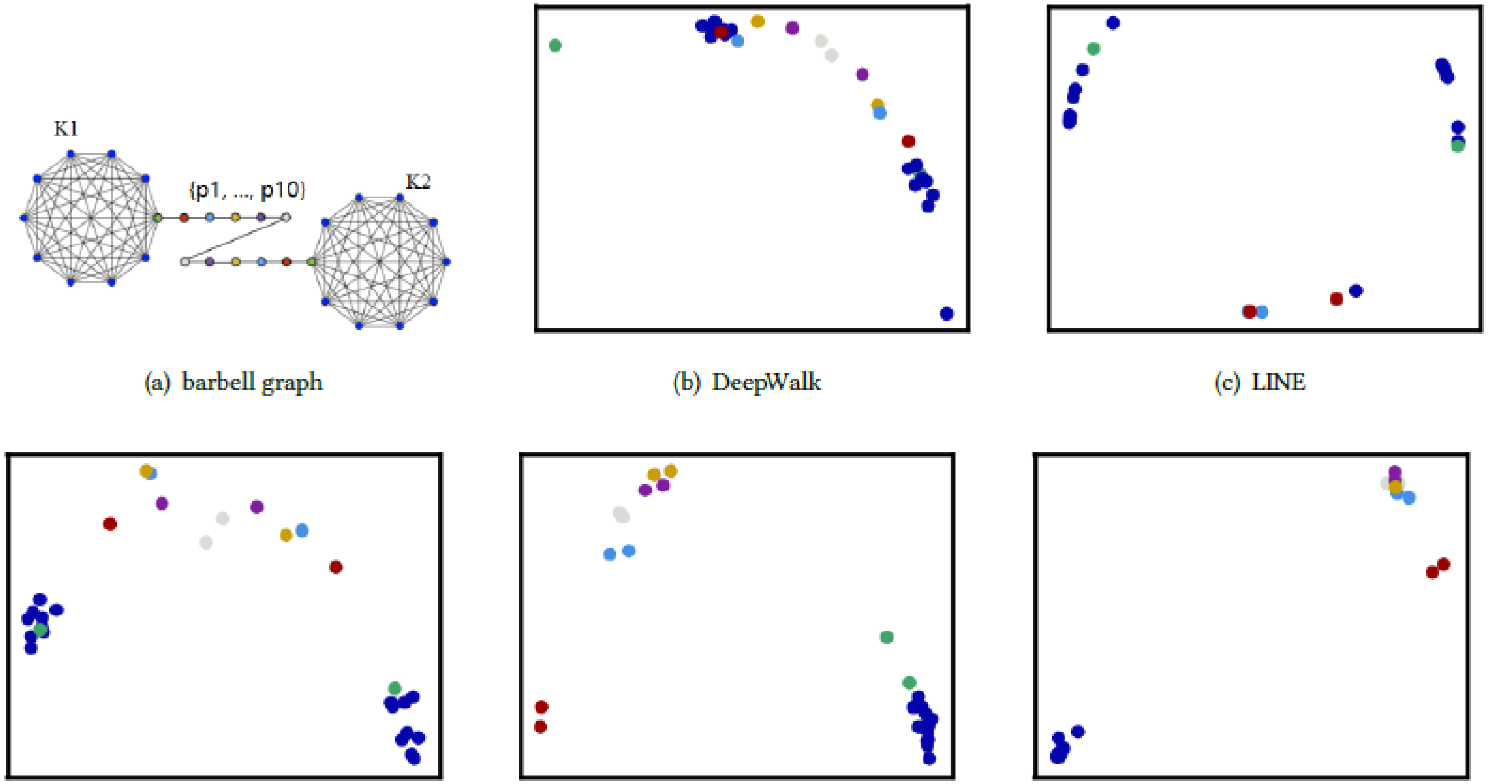
Centrality	Definition $C(v)$	$F(v)$	$f(\{x_i\})$
Degree	$d_v = \sum_{u \in \mathcal{N}(v)} I(d_u)$	$1/d_v$	$1/(\sum I(x_i))$
Eigenvector	$1/\lambda * \sum_{u \in \mathcal{N}(v)} C(u)$	$1/\lambda$	mean
PageRank	$\sum_{u \in \mathcal{N}(v)} 1/d_u * C(u)$	$1/d_v$	$1/(\sum I(x_i))$

Complexity Analysis

- The time complexity of training process is $O(NSk^2I)$
  - N is the number of nodes
  - S is the limited sample number.
  - k is the length of embedding, set as 32, 64, 128
  - I is the number of iterations(epochs)

- Conclusion: the complexity of training process is linear to the number of nodes n.

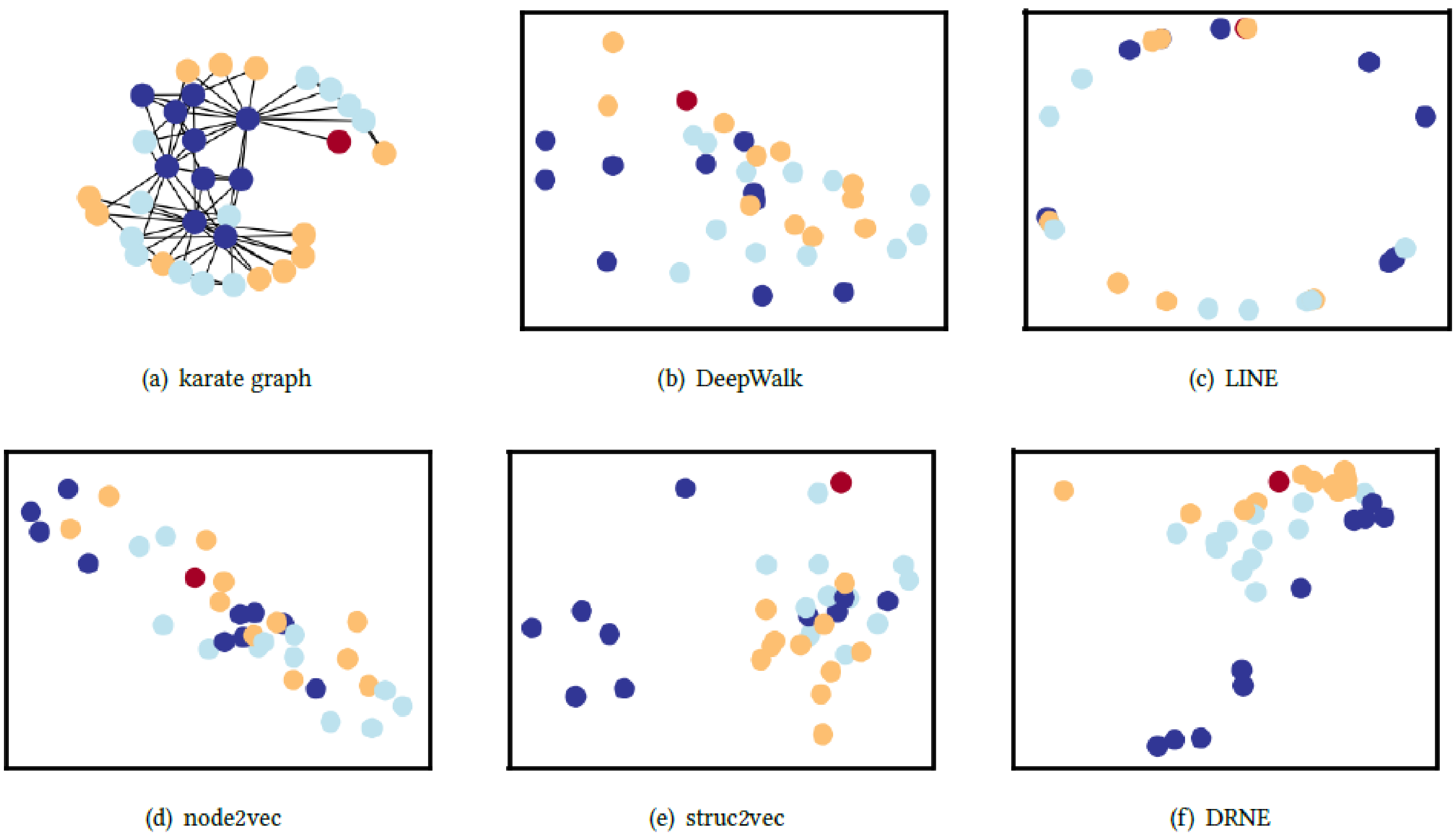
Experiment --- Network Visualization



(a) barbell graph (b) DeepWalk (c) LINE

(d) node2vec (e) struc2vec (f) DRNE

Experiment --- Network Visualization



Color: k-core

Experiment --- predict centrality

centrality	closeness	betweenness	eignvector	k-core
DeepWalk	0.6016	3.7188	2.1543	13.2755
LINE	0.5153	4.3919	1.5072	15.8179
node2vec	1.0489	3.4065	3.9436	39.2156
struc2vec	0.2365	0.25371	1.0544	9.0858
DRNE	**0.1909**	**0.1261**	**0.5267**	**5.5683**

The MSE value of predicting centralities on Jazz dataset ($*10^{-2}$)

centrality	closeness	betweenness	eignvector	k-core
DeepWalk	0.2982	1.7836	1.1194	19.7016
LINE	0.3979	1.8425	1.5167	34.9079
node2vec	0.3573	1.6958	1.1432	24.1704
struc2vec	0.2947	1.6018	1.0445	25.3047
DRNE	**0.1101**	**0.6676**	**0.3108**	**7.7210**

The MSE value of predicting centralities on BlogCatalog dataset ($*10^{-2}$)

Experiment --- Regular Equivalence Prediction

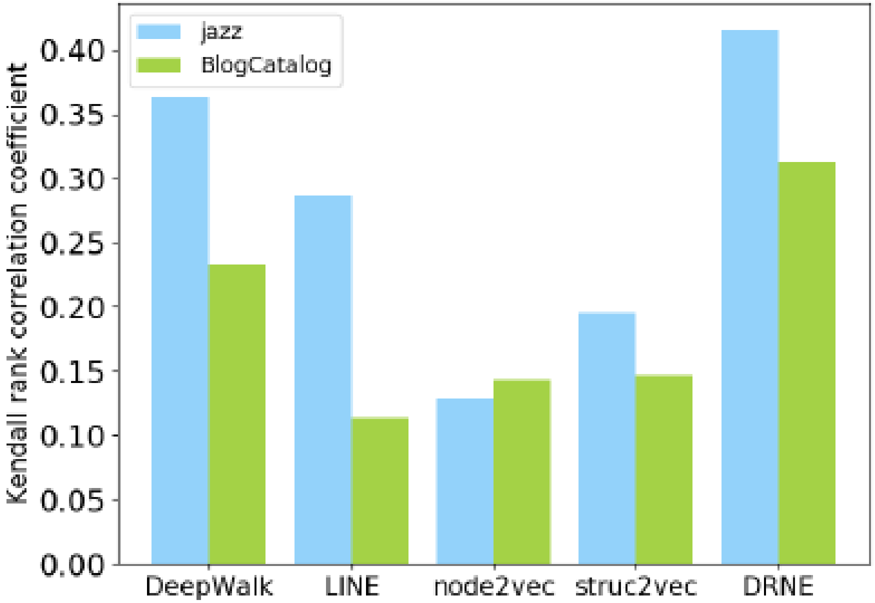
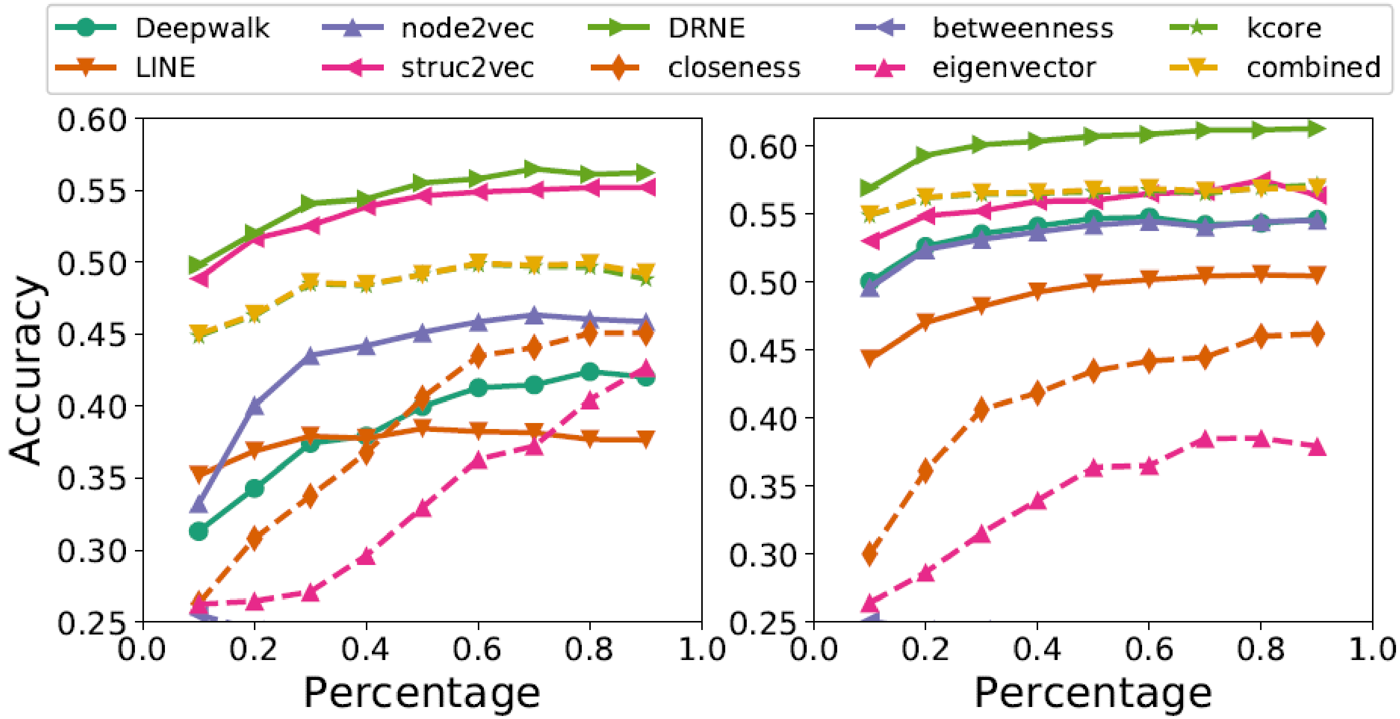


Figure 5: Kendall rank correlation coefficient by fitting regular equivalence on Jazz and BlogCatalog dataset.

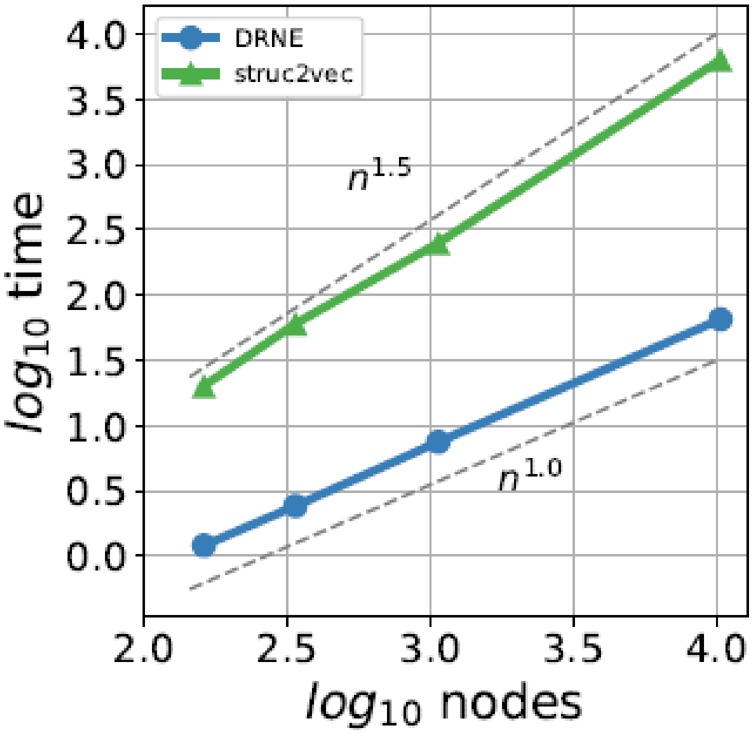
Experiment --- Structural Role Classification



Europe air-traffic

American air-traffic

Experiment --- Training Time



Linear training time

Summary

- Investigate a novel problem of learning node representations with regular equivalence
- Propose a novel deep model DRNE
  - Learn node representations by aggregating neighbors' representations recursively in a non-linear way
  - Theoretically prove that the learned representations can well reflect several popular and typical node centralities
  - Linear time complexity to the number of node
- Extensive experiments

Thanks!

Ke Tu, Tsinghua University

tuke1993@gmail.com