Intro to Network Embedding

Nate Russell

CS 512 UIUC Spring 17

Outline

Why Embed Network Vertices into Vector Space?

LINE: Large-scale Information Network Embedding
Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, Qiaozhu Mei

History and Overview of Network Embedding Methods

PROSNET: INTEGRATING HOMOLOGY WITH MOLECULAR NETWORKS FOR PROTEIN FUNCTION PREDICTION.
Wang S[1], Qu M, Peng J.

Why embed Networks into Vector Space

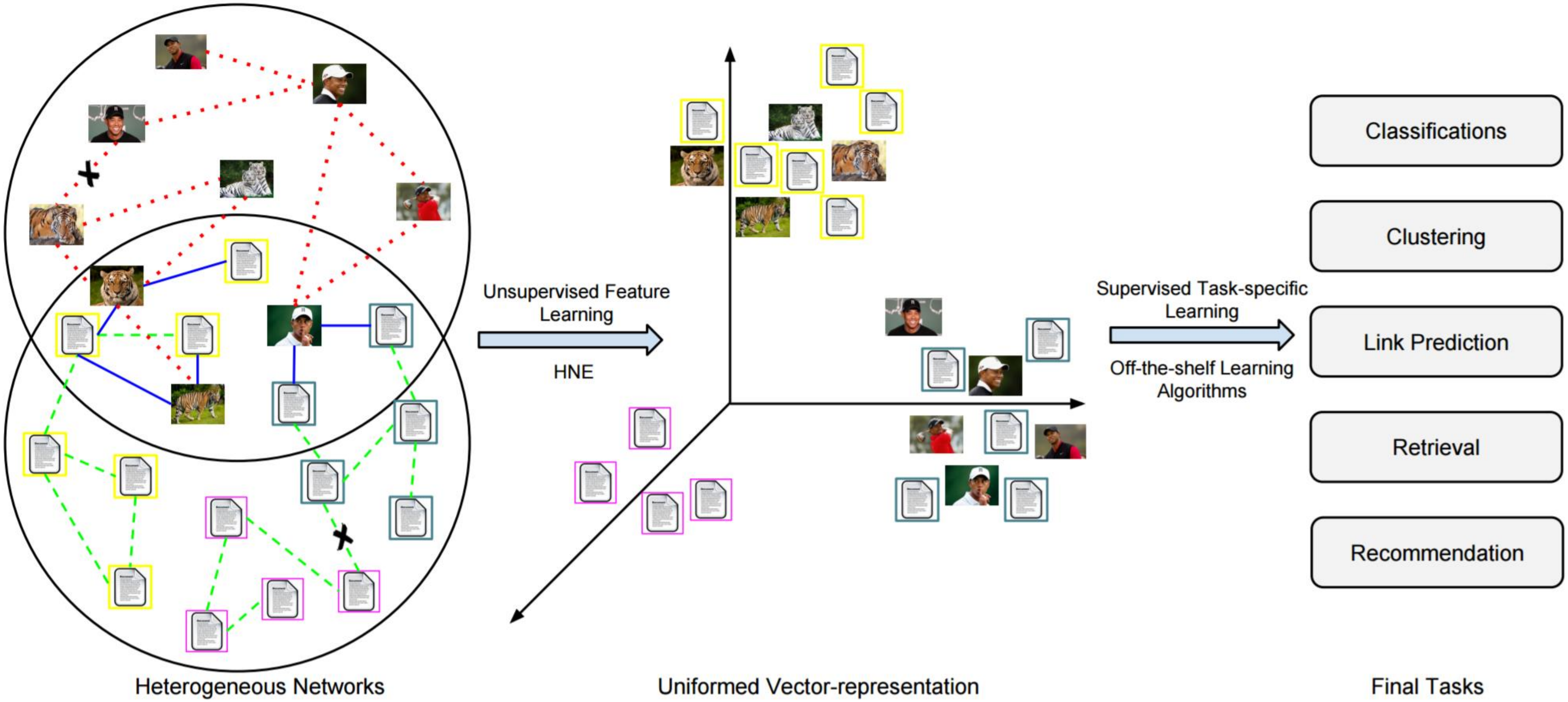


Figure 2: The flowchart of the proposed Heterogeneous Network Embedding (HNE) framework.

Chang et al KDD 2015

Formalization

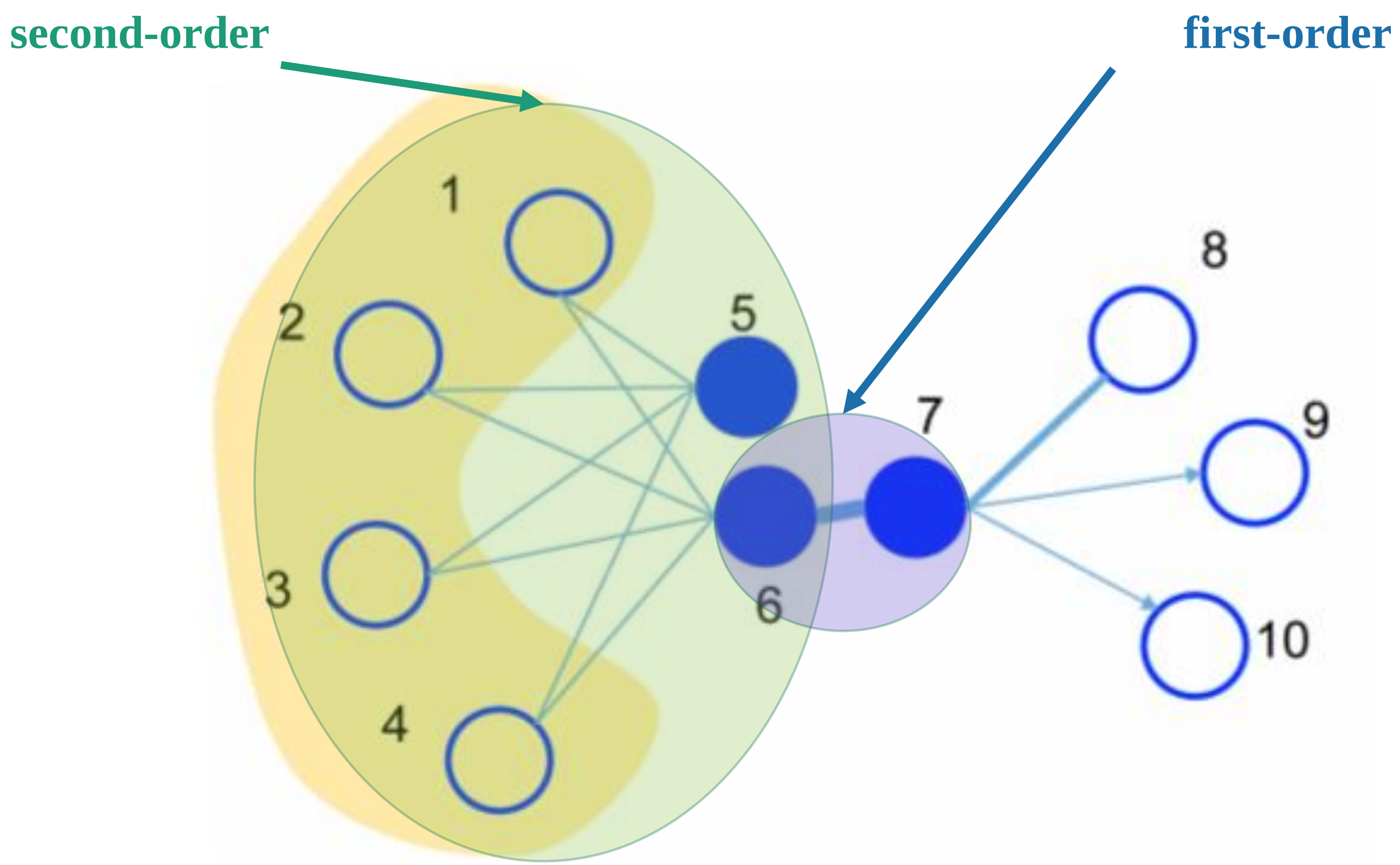
Given a large network G = (V, E)

Goal:
Represent each vertex v ∈ V into a low-dimensional space R d ,
i.e., learning a function f_G: V → R^d , where $d \ll |V|$.

For LINE:
In the space R^d, both the **first-order** proximity and the **second-order** proximity between the vertices are preserved.

In General:
In the space R^d, some combination of network distance or network topology is preserved.

LINE Similarity & Context



Directed Weighted Network

1st Order Model

Distribution over Vertex pairs

$$p_1(v_i, v_j) = \frac{1}{1 + \exp(-\boxed{\vec{u}_i^T} \cdot \vec{u}_j)}$$

$$\vec{u}_i \in R^d$$

Latent Vector

Empirical Distribution

$$\hat{p}_1(i, j) = \frac{w_{ij}}{\boxed{W}}$$

$$W = \sum_{(i,j)\in E} w_{ij}$$

Probability Mass = Normalized Edge Weights

1st Order Model

$$D_{\mathrm{KL}}(P\|Q) = \sum_i P(i)\,\log\frac{P(i)}{Q(i)}$$

KL Divergence

Empirical Distribution

Distribution over Vertex pairs

1st Order Loss Function

$$O_1 = \boxed{d}(\hat{p}_1(\cdot,\cdot), p_1(\cdot,\cdot))$$

Simplify and Remove Constants

$$O_1 = -\sum_{(i,j)\in E} w_{ij}\log p_1(v_i, v_j)$$

2nd Order Model

Distribution over
Vertex pairs

Context Latent

$$p_2(v_j|v_i) = \frac{\exp(\vec{u}_j'^T \cdot \vec{u}_i)}{\sum_{k=1}^{|V|} \exp(\vec{u}_k'^T \cdot \vec{u}_i)}$$

Empirical Distribution

$$\hat{p}_2(v_j|v_i) = \frac{w_{ij}}{\boxed{d_i}}$$

Weighted Out Degree

$$d_i = \sum_{k \in N(i)} w_{ik}$$

2^{nd} Order Model

2^{nd} Order Loss Function

$$\lambda_i = d_i$$

$$O_2 = \sum_{i \in V} \boxed{\lambda_i} \overbrace{d(\hat{p}_2(\cdot|v_i), p_2(\cdot|v_i))}^{\text{KL Divergence}}$$

Simplify and Remove Constants

$$O_2 = -\sum_{(i,j) \in E} w_{ij} \log p_2(v_j|v_i)$$

Simplify Density Estimation as Logistic Classification

$$\log \sigma({\vec{u}'_j}^{T} \cdot \vec{u}_i) + \sum_{i=1}^{K} E_{v_n \sim P_n(v)}[\log \sigma(-{\vec{u}'_n}^{T} \cdot \vec{u}_i)]$$

2nd Order Model

$$\overbrace{\log \sigma({\vec{u}'_j}^{T} \cdot \vec{u}_i)}^{\text{Observed}} + \overbrace{\sum_{i=1}^{K} E_{v_n \sim \boxed{P_n}(v)} [\log \sigma(-{\vec{u}'_n}^{T} \cdot \vec{u}_i)]}^{\text{Negative Sampling}}$$

Transform into Probability

$$\sigma(x) = 1/(1 + \exp(-x))$$

$$P_n(v) \propto {d_v}^{3/4}$$

d_v is the out-degree of vertex v.

High Out-degree, discounted but favored

Learn more about Negative Sampling

Popular Simplification: Q. Le and T. Mikolov. Distributed representations of sentences and documents. In Proceedings of The 31st International Conference on Machine Learning, pages 1188–1196, 2014

Original work on Noise Contrastive Estimation: Michael U Gutmann and Aapo Hyv¨arinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. The Journal of Machine Learning Research, 13:307–361, 2012.

1^{st} Order Model (With Negative Sampling)

$$O_1 = - \sum_{(i,j)\in E} w_{ij} \log p_1(v_i, v_j)$$

Trivial Solution Alert !

$$\boxed{u_{ik} = \infty, \text{ for i=1}, \ldots, |V| \text{ and } k = 1, \ldots, d.}$$

$\vec{u}_j'^T$ to $\vec{u}_j^T$ **Solution**

$$\log \sigma({\vec{u}_j'}^T \cdot \vec{u}_i) + \sum_{i=1}^{K} E_{v_n \sim P_n(v)}[\log \sigma(-{\vec{u}_n'}^T \cdot \vec{u}_i)]$$

Tricks

Gradient of 2nd order model

$$\frac{\partial O_2}{\partial \vec{u}_i} = w_{ij} \cdot \frac{\partial \log p_2(v_j|v_i)}{\partial \vec{u}_i}$$

High Variance ~ Exploding Gradient

Edge Imputation – Influence Assumption

$$w_{ij} = \sum_{k \in N(i)} w_{ik} \frac{w_{kj}}{d_k}$$

Alias Table Technique: A. Q. Li, A. Ahmed, S. Ravi, and A. J. Smola. Reducing the sampling complexity of topic models. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 891–900. ACM, 2014

LINE Performance

Table 1: Statistics of the real-world information networks.

	Language Network	Social Network		Citation Network	
Name	WIKIPEDIA	FLICKR	YOUTUBE	DBLP(AUTHORCITATION)	DBLP(PAPERCITATION)
Type	undirected,weighted	undirected,binary	undirected,binary	dircted,weighted	directed,binary
|V|	1,985,098	1,715,256	1,138,499	524,061	781,109
|E|	1,000,924,086	22,613,981	2,990,443	20,580,238	4,191,677
Avg. degree	504.22	26.37	5.25	78.54	10.73
#Labels	7	5	47	7	7
#train	70,000	75,958	31,703	20,684	10,398

Table 5: Results of multi-label classification on the FLICKR network.

Metric	Algorithm	10%	20%	30%	40%	50%	60%	70%	80%	90%
Micro-F1	GF	53.23	53.68	53.98	54.14	54.32	54.38	54.43	54.50	54.48
	DeepWalk	60.38	60.77	60.90	61.05	61.13	61.18	61.19	61.29	61.22
	DeepWalk(256dim)	60.41	61.09	61.35	61.52	61.69	61.76	61.80	61.91	61.83
	LINE(1st)	63.27	63.69	63.82	63.92	63.96	64.03	64.06	64.17	64.10
	LINE(2nd)	62.83	63.24	63.34	63.44	63.55	63.55	63.59	63.66	63.69
	LINE(1st+2nd)	**63.20****	**63.97****	**64.25****	**64.39****	**64.53****	**64.55****	**64.61****	**64.75****	**64.74****
Macro-F1	GF	48.66	48.73	48.84	48.91	49.03	49.03	49.07	49.08	49.02
	DeepWalk	58.60	58.93	59.04	59.18	59.26	59.29	59.28	59.39	59.30
	DeepWalk(256dim)	59.00	59.59	59.80	59.94	60.09	60.17	60.18	60.27	60.18
	LINE(1st)	62.14	62.53	62.64	62.74	62.78	62.82	62.86	62.96	62.89
	LINE(2nd)	61.46	61.82	61.92	62.02	62.13	62.12	62.17	62.23	62.25
	LINE(1st+2nd)	**62.23****	**62.95****	**63.20****	**63.35****	**63.48****	**63.48****	**63.55****	**63.69****	**63.68****

Significantly outperforms DeepWalk at the: ** 0.01 and * 0.05 level, paired t-test.

LINE Performance

Table 2: Results of word analogy on WIKIPEDIA data.

Algorithm	Semantic (%)	Syntactic (%)	Overall (%)	Running time
GF	61.38	44.08	51.93	2.96h
DeepWalk	50.79	37.70	43.65	16.64h
SkipGram	69.14	57.94	63.02	2.82h
LINE-SGD(1st)	9.72	7.48	8.50	3.83h
LINE-SGD(2nd)	20.42	9.56	14.49	3.94h
LINE(1st)	58.08	49.42	53.35	2.44h
LINE(2nd)	**73.79**	**59.72**	**66.10**	2.55h

Table 3: Results of Wikipedia page classification on WIKIPEDIA data set.

Metric	Algorithm	10%	20%	30%	40%	50%	60%	70%	80%	90%
Micro-F1	GF	79.63	80.51	80.94	81.18	81.38	81.54	81.63	81.71	81.78
	DeepWalk	78.89	79.92	80.41	80.69	80.92	81.08	81.21	81.35	81.42
	SkipGram	79.84	80.82	81.28	81.57	81.71	81.87	81.98	82.05	82.09
	LINE-SGD(1st)	76.03	77.05	77.57	77.85	78.08	78.25	78.39	78.44	78.49
	LINE-SGD(2nd)	74.68	76.53	77.54	78.18	78.63	78.96	79.19	79.40	79.57
	LINE(1st)	79.67	80.55	80.94	81.24	81.40	81.52	81.61	81.69	81.67
	LINE(2nd)	79.93	80.90	81.31	81.63	81.80	81.91	82.00	82.11	82.17
	LINE(1st+2nd)	**81.04****	**82.08****	**82.58****	**82.93****	**83.16****	**83.37****	**83.52****	**83.63****	**83.74****
Macro-F1	GF	79.49	80.39	80.82	81.08	81.26	81.40	81.52	81.61	81.68
	DeepWalk	78.78	79.78	80.30	80.56	80.82	80.97	81.11	81.24	81.32
	SkipGram	79.74	80.71	81.15	81.46	81.63	81.78	81.88	81.98	82.01
	LINE-SGD(1st)	75.85	76.90	77.40	77.71	77.94	78.12	78.24	78.29	78.36
	LINE-SGD(2nd)	74.70	76.45	77.43	78.09	78.53	78.83	79.08	79.29	79.46
	LINE(1st)	79.54	80.44	80.82	81.13	81.29	81.43	81.51	81.60	81.59
	LINE(2nd)	79.82	80.81	81.22	81.52	81.71	81.82	81.92	82.00	82.07
	LINE(1st+2nd)	**80.94****	**81.99****	**82.49****	**82.83****	**83.07****	**83.29****	**83.42****	**83.55****	**83.66****

Significantly outperforms GF at the: ** 0.01 and * 0.05 level, paired t-test.

LINE Performance

Table 6: Results of multi-label classification on the YOUTUBE network. The results in the brackets are on the reconstructed network, which adds second-order neighbors (i.e., neighbors of neighbors) as neighbors for vertices with a low degree.

Metric	Algorithm	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%
Micro-F1	GF	25.43 (24.97)	26.16 (26.48)	26.60 (27.25)	26.91 (27.87)	27.32 (28.31)	27.61 (28.68)	27.88 (29.01)	28.13 (29.21)	28.30 (29.36)	28.51 (29.63)
	DeepWalk	39.68	41.78	42.78	43.55	43.96	44.31	44.61	44.89	45.06	45.23
	DeepWalk(256dim)	39.94	42.17	43.19	44.05	44.47	44.84	45.17	45.43	45.65	45.81
	LINE(1st)	35.43 (36.47)	38.08 (38.87)	39.33 (40.01)	40.21 (40.85)	40.77 (41.33)	41.24 (41.73)	41.53 (42.05)	41.89 (42.34)	42.07 (42.57)	42.21 (42.73)
	LINE(2nd)	32.98 (36.78)	36.70 (40.37)	38.93 (42.10)	40.26 (43.25)	41.08 (43.90)	41.79 (44.44)	42.28 (44.83)	42.70 (45.18)	43.04 (45.50)	43.34 (45.67)
	LINE(1st+2nd)	39.01* **(40.20)**	41.89 **(42.70)**	43.14 **(43.94**)**	44.04 **(44.71**)**	44.62 **(45.19**)**	45.06 **(45.55**)**	45.34 **(45.87**)**	45.69** **(46.15**)**	45.91** **(46.33**)**	46.08** **(46.43**)**
Macro-F1	GF	7.38 (11.01)	8.44 (13.55)	9.35 (14.93)	9.80 (15.90)	10.38 (16.45)	10.79 (16.93)	11.21 (17.38)	11.55 (17.64)	11.81 (17.80)	12.08 (18.09)
	DeepWalk	28.39	30.96	32.28	33.43	33.92	34.32	34.83	35.27	35.54	35.86
	DeepWalk (256dim)	28.95	31.79	33.16	34.42	34.93	35.44	35.99	36.41	36.78	37.11
	LINE(1st)	28.74 (29.40)	31.24 (31.75)	32.26 (32.74)	33.05 (33.41)	33.30 (33.70)	33.60 (33.99)	33.86 (34.26)	34.18 (34.52)	34.33 (34.77)	34.44 (34.92)
	LINE(2nd)	17.06 (22.18)	21.73 (27.25)	25.28 (29.87)	27.36 (31.88)	28.50 (32.86)	29.59 (33.73)	30.43 (34.50)	31.14 (35.15)	31.81 (35.76)	32.32 (36.19)
	LINE(1st+2nd)	**29.85** (29.24)	31.93 **(33.16**)**	33.96 **(35.08**)**	35.46** **(36.45**)**	36.25** **(37.14**)**	36.90** **(37.69**)**	37.48** **(38.30**)**	38.10** **(38.80**)**	38.46** **(39.15**)**	38.82** **(39.40**)**

Significantly outperforms DeepWalk at the: ** 0.01 and * 0.05 level, paired t-test.

Table 7: Results of multi-label classification on DBLP(AUTHORCITATION) network.

Metric	Algorithm	10%	20%	30%	40%	50%	60%	70%	80%	90%
Micro-F1	DeepWalk	63.98	64.51	64.75	64.81	64.92	64.99	64.99	65.00	64.90
	LINE-SGD(2nd)	56.64	58.95	59.89	60.20	60.44	60.61	60.58	60.73	60.59
	LINE(2nd)	62.49 **(64.69*)**	63.30 **(65.47**)**	63.63 **(65.85**)**	63.77 **(66.04**)**	63.84 **(66.19**)**	63.94 **(66.25**)**	63.96 **(66.30**)**	64.00 **(66.12**)**	63.77 **(66.05**)**
Macro-F1	DeepWalk	63.02	63.60	63.84	63.90	63.98	64.06	64.09	64.11	64.05
	LINE-SGD(2nd)	55.24	57.63	58.56	58.82	59.11	59.27	59.28	59.46	59.37
	LINE(2nd)	61.43 **(63.49*)**	62.38 **(64.42**)**	62.73 **(64.84**)**	62.87 **(65.05**)**	62.93 **(65.19**)**	63.05 **(65.26**)**	63.07 **(65.29**)**	63.13 **(65.14**)**	62.95 **(65.14**)**

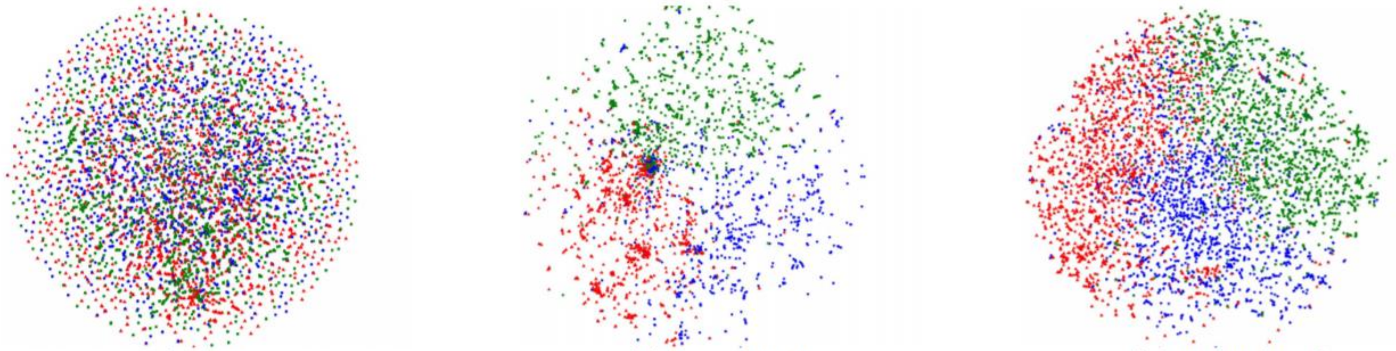
Significantly outperforms DeepWalk at the: ** 0.01 and * 0.05 level, paired t-test.

LINE Performance

Table 8: Results of multi-label classification on DBLP(PAPERCITATION) network.

Metric	Algorithm	10%	20%	30%	40%	50%	60%	70%	80%	90%
Micro-F1	DeepWalk	52.83	53.80	54.24	54.75	55.07	55.13	55.48	55.42	55.90
	LINE(2nd)	58.42	59.58	60.29	60.78	60.94	61.20	61.39	61.39	61.79
		(**60.10****)	(**61.06****)	(**61.46****)	(**61.73****)	(**61.85****)	(**62.10****)	(**62.21****)	(**62.25****)	(**62.80****)
Macro-F1	DeepWalk	43.74	44.85	45.34	45.85	46.20	46.25	46.51	46.36	46.73
	LINE(2nd)	48.74	50.10	50.84	51.31	51.61	51.77	51.94	51.89	52.16
		(**50.22****)	(**51.41****)	(**51.92****)	(**52.20****)	(**52.40****)	(**52.59****)	(**52.78****)	(**52.70****)	(**53.02****)

Significantly outperforms DeepWalk at the: ** 0.01 and * 0.05 level, paired t-test.



(a) GF (b) DeepWalk (c) LINE(2nd)

Figure 2: Visualization of the co-author network. The authors are mapped to the 2-D space using the t-SNE package with learned embeddings as input. Color of a node indicates the community of the author. Red: “data Mining,” blue: “machine learning,” green: “computer vision.”

Paper	Year	# of Citations
Distributed Large-scale Natural Graph Factorization	13	29
Translating Embeddings for Modeling Multi-relational Data (TransE)	13	234
DeepWalk: Online Learning of Social Representations	14	158
Combining Two And Three-Way Embeddings Models for Link Prediction in Knowledge Bases (Tatec)	15	7
Holographic Embeddings of Knowledge Graphs (HOLE)	15	10
Diffusion Component Analysis: Unraveling Functional Topology in Biological Networks	15	11
GraRep: Learning Graph Representations with Global Structural Information	15	15
Deep Graph Kernels	15	16
Heterogeneous Network Embedding via Deep Architectures	15	25
PTE: Predictive Text Embedding through Large-scale Heterogeneous Text Networks	15	30
LINE: Large-scale Information Network Embedding	15	90
A General Framework for Content-enhanced Network Representation Learning (CENE)	16	0
Variational Graph Auto-Encoders (VGAE)	16	0
PROSNET: INTEGRATING HOMOLOGY WITH MOLECULAR NETWORKS FOR PROTEIN FUNCTION PREDICTION.	16	0
Large-Scale Embedding Learning in Heterogeneous Event Data (HEBE)	16	0
AFET: Automatic Fine-Grained Entity Typing by Hierarchical Partial-Label Embedding	16	0
Deep Neural Networks for Learning Graph Representations (DNGR)	16	1
subgraph2vec: Learning Distributed Representations of Rooted Sub-graphs from Large Graphs	16	2
Walklets: Multiscale Graph Embeddings for Interpretable Network Classification	16	2
Asymmetric Transitivity Preserving Graph Embedding (HOPE)	16	3
Label Noise Reduction in Entity Typing by Heterogeneous Partial-Label Embedding (PLE)	16	6
Semi-Supervised Classification with Graph Convolutional Networks (GCN)	16	7
Revisiting Semi-Supervised Learning with Graph Embeddings (Planetoid)	16	10
Structural Deep Network Embedding	16	12
node2vec: Scalable Feature Learning for Networks	16	27

Network Embedding Overview

Algorithm	Weighted	Directed	Context Deifinition	Loss
LINE	●	◑	1st order and 2nd Order	Negative Sampling + L2 + Concat
Graph Factorization	●	○	Laplacian Eigenvectors	SVD
DeepWalk	○	●	Fixed Length R.W.	Negative Sampling + L2
DCA	●	○	R.W. with Restart	KL Divergence
SDNE	●	○	1st order and 2nd Order	Joint Loss AutoEncoder
GraRep	●	◑	1st order and Fixed Length R.W.	SVD
DNGR	●	◑	Regularized R.W. with Restart	Denoising Autoencoder
HOPE	●	●	HOPE similarity Metric	JDG-SVD
node2vec	●	●	Fixed Length Biased R.W.	Negative Sampling + Weighted L2
DGK	●	○	Subgraphs occuring at same degree	Negative Sampling + Multiplicative Combination
subgraph2vec	●	○	Subgraphs of different degress	Negative Sampling + L2
Walklets	●	●	Fixed Length R.W.	Negative Sampling + L2
Proposed	●	●	1st order and Fixed Length R.W.	Ladder Network

ProSNet

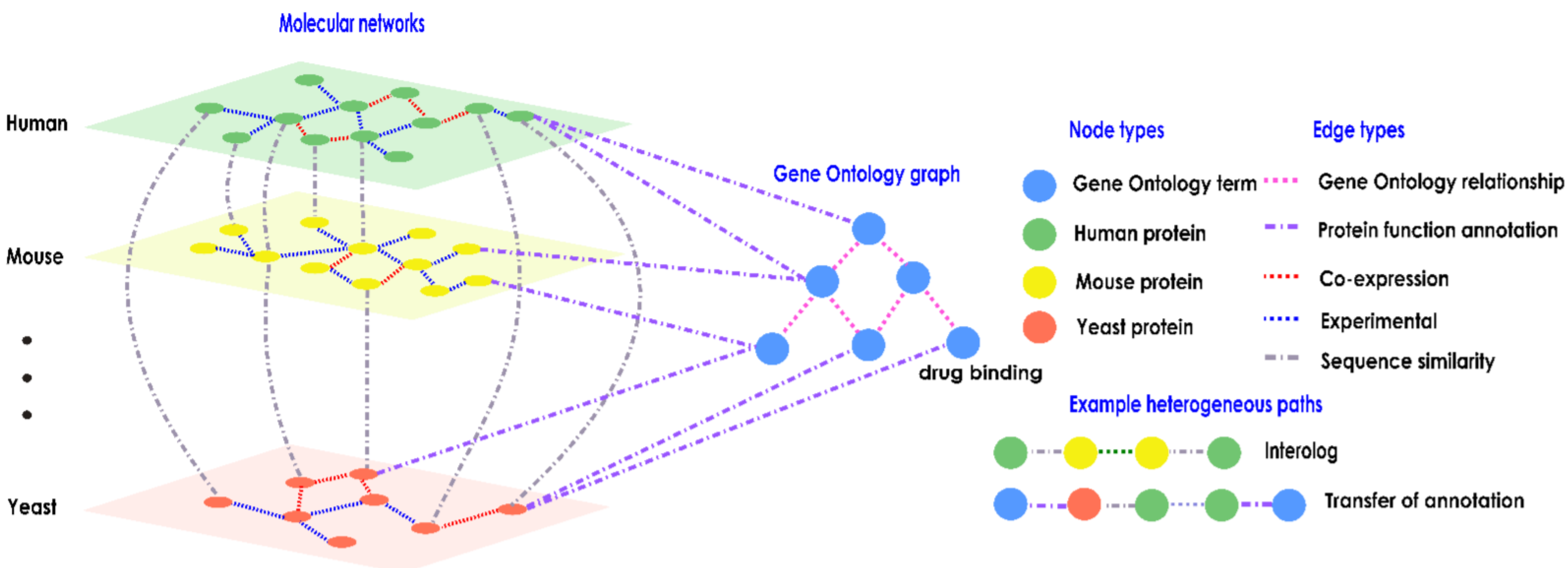


Fig. 1. An example of the heterogeneous biological network under our function prediction framework. The node set V consists of four types, {"*Human protein*", "*Yeast protein*", "*Mouse protein*", and "*Gene Ontology term*"}. The edge type set R consists of five types, {"*Sequence similarity*", "*Protein function annotation*","*Gene Ontology relationship*","*Experimental*", and "*Co-expression*" }. This HBN explicitly captures *interolog* and *transfer of annotation* through heterogeneous paths across different species.

Probability that v is connected to u by M

$$Pr(v|u,\mathcal{M}) = \frac{\exp(f(u,v,\mathcal{M}))}{\sum_{v'\in V}\exp(f(u,v',\mathcal{M}))}$$

Scoring Function (Inspired by GloVe)

$$f(u,v,\mathcal{M}) = \mu_{\mathcal{M}} + \underbrace{\mathbf{p}_{\mathcal{M}}{}^T\mathbf{x_u} + \mathbf{q}_{\mathcal{M}}{}^T\mathbf{x_v}}_{\text{Local Bias}} + \underbrace{\mathbf{x_u}{}^T\mathbf{x_v}}_{\text{Inner Product Similarity}}$$

Inner Product Similarity

$\mathbf{p}_{\mathcal{M}}$ and $\mathbf{q}_{\mathcal{M}} \in \mathbb{R}^d$

Local Bias

$\mu_{\mathcal{M}} \in \mathbb{R}$ is the global bias of the heterogeneous path $\mathcal{M}$

Glove: J. Pennington, R. Socher and C. D. Manning, Glove: Global vectors for word representation., in EMNLP, 2014

Path $\mathcal{P}_{e_1 \rightsquigarrow e_L} = \langle e_1 = \langle u_1, v_1, r_1 \rangle, \ldots, e_L = \langle u_L, v_L, r_L \rangle \rangle$

Follows $\mathcal{M} = \langle r_1, r_2, \ldots, r_L \rangle$

Probability that path of type M exists

Approximation enables recurrence and allows for dynamic programming to reduce computational burden

$$Pr(\mathcal{P}_{e_1 \rightsquigarrow e_L} | \mathcal{M}) \propto \overbrace{C(u_1, 1|\mathcal{M})^{\boxed{\gamma}} \times Pr(\mathcal{P}_{e_1 \rightsquigarrow e_L} | u_1, \mathcal{M})}$$

0.75 (Same as Word2Vec)
Discounts popular nodes

$$C(u, i|\mathcal{M})$$

Count of paths following M where the ith Node is u

$$Pr(\mathcal{P}_{e_1 \rightsquigarrow e_L} | u_1, \mathcal{M}) = \prod_{i=1}^{L} Pr(v_i | u_i, r_i)$$

Assume each node on path only depends on previous node

Conditional distributions are tractable but expensive

$$\begin{cases} Pr(v|u,\mathcal{M}) = \dfrac{\exp(f(u,v,\mathcal{M}))}{\sum_{v'\in V}\exp(f(u,v',\mathcal{M}))} \\[2ex] Pr(\mathcal{P}_{e_1\rightsquigarrow e_L}|\mathcal{M}) \propto C(u_1,1|\mathcal{M})^{\gamma} \times Pr(\mathcal{P}_{e_1\rightsquigarrow e_L}|u_1,\mathcal{M}) \end{cases}$$

Simplify Density Estimation as Logistic Classification

$$\overbrace{\frac{1}{\boxed{\theta}+1} Pr^{+}(\mathcal{P}_{e_1\rightsquigarrow e_L}|\mathcal{M})}^{\text{Observed}} + \overbrace{\frac{\theta}{\theta+1} Pr^{-}(\mathcal{P}_{e_1\rightsquigarrow e_L}|\mathcal{M})}^{\text{Negative Sampling Distribution}}$$

Negative Sampling Weight

$$Pr^{-}(\mathcal{P}_{e_1\rightsquigarrow e_L}|\mathcal{M}) \propto \prod_{i=1}^{L+1} C(u_i, i\,|\mathcal{M})^{\gamma}$$

The posterior probability that a given sample D came from positive path instance samples following the given heterogeneous path is

$$Pr(\boxed{D} = 1 | \mathcal{P}_{e_1 \rightsquigarrow e_L}, \mathcal{M}) = \frac{Pr^+(\mathcal{P}_{e_1 \rightsquigarrow e_L} | \mathcal{M})}{Pr^+(\mathcal{P}_{e_1 \rightsquigarrow e_L} | \mathcal{M}) + \theta \cdot Pr^-(\mathcal{P}_{e_1 \rightsquigarrow e_L} | \mathcal{M})}$$

$$D \in \{0, 1\}$$

$$Pr(\mathcal{P}_{e_1 \rightsquigarrow e_L} | \mathcal{M}) \xrightarrow{\text{Fit the Distribution}} Pr^+(\mathcal{P}_{e_1 \rightsquigarrow e_L} | \mathcal{M}).$$

Accomplished by Maximizing the Expected Log Likelihood

$$\begin{aligned}\mathcal{L}_{\mathcal{M}} =& \mathbb{E}_{Pr^+}\left[\log \frac{Pr(\mathcal{P}_{e_1 \rightsquigarrow e_L} | \mathcal{M})}{Pr(\mathcal{P}_{e_1 \rightsquigarrow e_L} | \mathcal{M}) + \theta \cdot Pr^-(\mathcal{P}_{e_1 \rightsquigarrow e_L} | \mathcal{M})}\right] \\ &+ \theta \cdot \mathbb{E}_{Pr^-}\left[\log \frac{\theta \cdot Pr^-(\mathcal{P}_{e_1 \rightsquigarrow e_L} | \mathcal{M})}{Pr(\mathcal{P}_{e_1 \rightsquigarrow e_L} | \mathcal{M}) + \theta \cdot Pr^-(\mathcal{P}_{e_1 \rightsquigarrow e_L} | \mathcal{M})}\right],\end{aligned}$$

Derived Loss Function

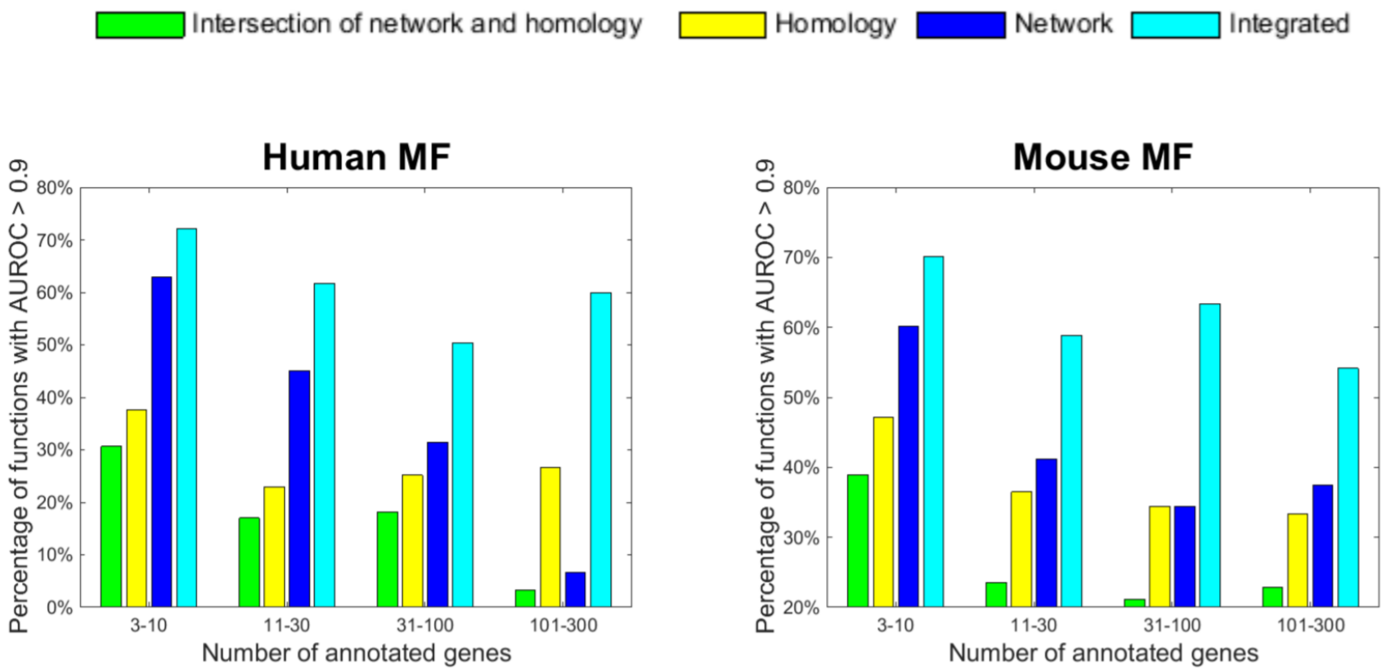
$$\mathcal{L}_{\mathcal{M}} \approx \sum_{\mathcal{P}_{e_1 \rightsquigarrow e_L} \text{ following } \mathcal{M}} \log \sigma\Big(\sum_{i=1}^{L} f(u_i, v_i, r_i)\Big) + \sum_{j=1}^{\theta} \mathbb{E}_{\mathcal{P}^j_{e_1 \rightsquigarrow e_L} \sim Pr^- | u_1, \mathcal{M}} \Big[\log\big(1 - \sigma(\textstyle\sum_{i=1}^{L} f(u_i^j, v_i^j, r_i))\big) \Big]$$

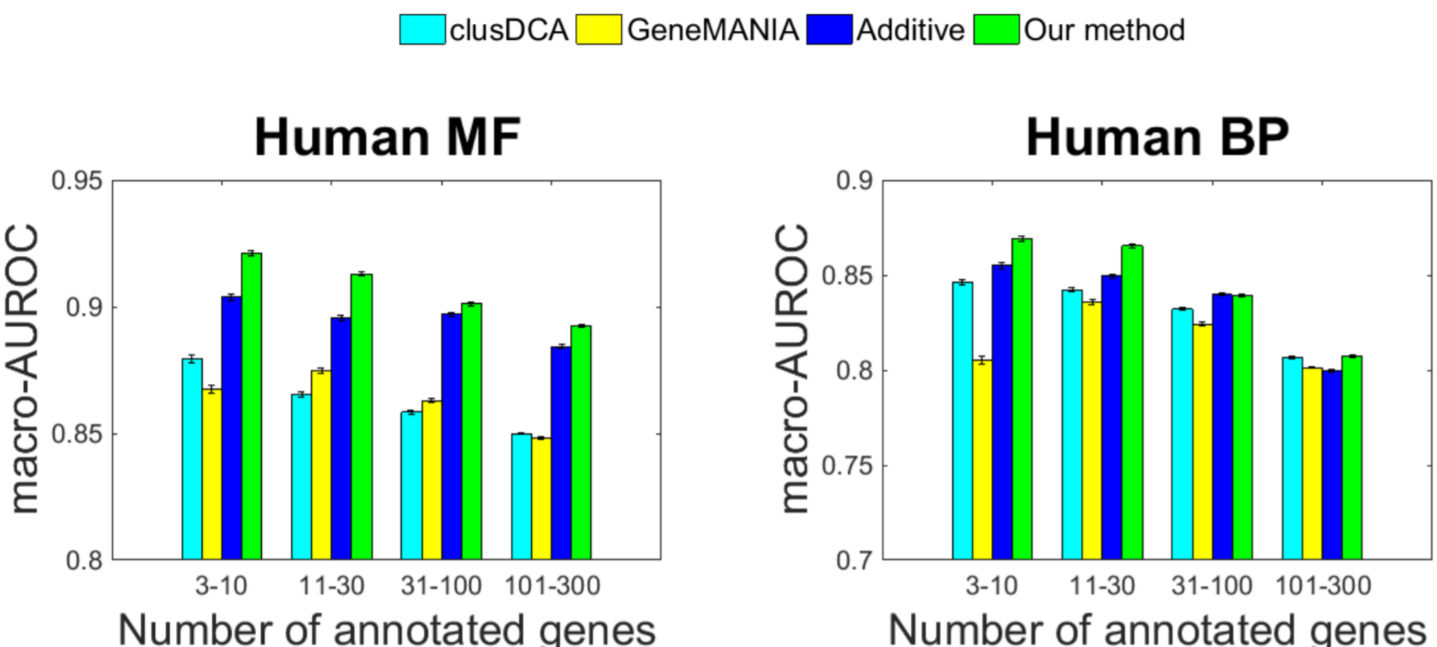
$$Pr(v|u, \mathcal{M}) = \frac{\exp(f(u, v, \mathcal{M}))}{\cancel{\sum_{v' \in V} \exp(f(u, v', \mathcal{M}))}}$$

"We can do this because the NCE objective encourages the model to be approximately normalized and recovers a perfectly normalized model if the model class contains the data distribution"

$$\sum_{i=1}^{L} f(u_i, v_i, r_i) - \cancel{\log\left(\theta \cdot Pr^-(\mathcal{P}_{e_1 \rightsquigarrow e_L} | \mathcal{M})\right)}$$

Dropped same way that original NCE work did


Intersection of network and homology
Homology
Network
Integrated
Human MF
Mouse MF
Percentage of functions with AUROC > 0.9
0%
10%
20%
30%
40%
50%
60%
70%
80%
3-10
11-30
31-100
101-300
Number of annotated genes


clusDCA
GeneMANIA
Additive
Our method
Human MF
Human BP
macro-AUROC
0.8
0.85
0.9
0.95
0.7
0.75
3-10
11-30
31-100
101-300
Number of annotated genes

Thank You for your Time

Questions?