

# Supplementary File for “Similarity Modeling on Heterogeneous Networks via Automatic Path Discovery”

Carl Yang, Mengxiong Liu, Frank He, Xikun Zhang, Jian Peng, and Jiawei Han

University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA  
{jiyang3, mliu60, shibihe, xikunz2, jianpeng, hanj}@illinois.edu

## 1 Related Work

We emphasize the novelty of our work through comparison with the related three lines of research.

### 1.1 Heterogeneous Network Modeling

Networks provide a natural and generic way of modeling real-world structured data. Among them, heterogeneous networks have drawn increasing research attention in the past decade, due to its capacity of retaining rich type information [31]. The rich representation of nodes and links facilitates various downstream applications including link prediction [16], classification [8], clustering [33], recommender systems [45], outlier detection [46], *etc.*

The modeling of heterogeneous networks is mainly around similarity learning under meta-paths [32, 39, 4, 19, 28, 38]. For example, [32] defines the proximity between two nodes by the normalized count of path instances following a user-specified meta-path, and shows that this measure captures better peer similarity semantics than random walk based similarity measures like SimRank [10] and PPR [11]. Other methods, while leveraging different techniques to combine or generate multiple meta-paths, all assume a given or enumerable set of useful meta-paths up to a certain empirically decided length, and none of them consider node contents.

Recently, network embedding algorithms based on the advances in neural network models like Skip-gram models [21] and convolutional neural networks [14] have been extremely popular [25, 36, 6, 23, 13, 7]. They aim to compute distributed representations of nodes that capture their network proximities regarding both neighborhood similarity and structural similarity [40, 26, 18]. The learned embedding vectors can then be leveraged for various network learning tasks including link prediction [44], node classification [43], community detection [42], *etc.* Following this trend, researchers have also been actively studying the embedding of heterogeneous networks [3, 27, 9, 5, 29, 37]. However, all existing heterogeneous network modeling methods are developed based on the assumption that proper meta-paths are given or can be enumerated, and none of them considers the integration of rich contents on the nodes.

In this work, we break free the requirements of given or enumerable meta-paths, and instead, we automatically explore and learn the useful paths with efficient reinforcement learning, based on a small set of example pairs of similar nodes. Furthermore, we integrate rich node contents through joint deep embedding, and thus are able to find discriminative paths among instances of the same meta-paths.

## 1.2 Reinforcement Learning

Reinforcement learning has been widely used in robot control, human-computer interaction and automated decision making, in which software agents need to consider interactions with a dynamic environment and take actions towards certain goals [34, 12]. They study the problems by learning policies to make decisions and maximize a reward signal from the environment. Some recent significant progress has been made by combining advances in deep learning for sensory processing with reinforcement learning [14], resulting in the super-human performance on high-dimensional game controls [22, 30].

As close to our work, several approaches have been proposed to solve the inherent path discovery problem for reasoning over knowledge bases with reinforcement learning [41, 2]. Policy gradient [35] is used based on the discrete state space encoded by [1] to learn a conditional stochastic policy for fact prediction in [41], while [2] utilizes LSTM to construct a policy network over discrete states without pretraining for query answering. These methods have achieved notable successes on knowledge bases. However, when it comes to our scenario of similarity modeling over heterogeneous networks, the algorithms are no longer appropriate, since we consider fewer node types but with more outgoing edges, as well as the probabilistic measure of paths together with node contents.

In this work, we connect continuous control policy gradient with network embedding to form a coherent semi-supervised learning framework. The training of policy network is supervised by example similar pairs of nodes, whereas the autoencoder based network embedding is unsupervised. The two objectives are jointly trained to automatically explore useful paths and rich semantics in content-rich heterogeneous networks, and enable inductive learning on unlabeled nodes. To the best of our knowledge, this is the first research effort to leverage reinforcement learning for heterogeneous network similarity modeling with efficient path exploration.

## 1.3 Deep Embedding

Embedding or representation based on neural networks has been intensively studied recently in various artificial intelligence related fields.

To empower the information exploration of reinforcement learning on content-rich heterogeneous networks, we propose to leverage deep embedding for content representation learning. As we discussed before, node contents on heterogeneous networks can be of various formats, so we want the embedding technique to be general and flexible. Moreover, since we aim to achieve inductive learning on the

network, *i.e.*, learn the important paths and contents on a small set of example pairs of nodes and infer the similarity among arbitrary pairs, we want the embedding of node attributes to be unsupervised.

To this end, we employ deep denoise autoencoder (DAE) [15], which has been proven advantageous in capturing the intrinsic features within high-dimensional sparse, noisy inputs in an unsupervised fashion. To deal with multiple types of inputs, we can insert various differentiable neural networks after the raw content vectors, and concatenate multiple outputs before sending to the generic DAE. For example, for categorical contents, we can insert a look-up embedding layer; for numerical contents, simple linear feedforward neural networks with nonlinear activations can be devised; for textual and visual contents, classic RNN [20] and CNN [14] can be leveraged, respectively. Finally, for nodes with a meaningful name but no contents in the network, we can also leverage their semantics by finding the embedding of the words in their names from precomputed word embeddings like [24], and then compute the average before sending to the DAE.

## 2 Discussions

### 2.1 Again, Why Continuous Space?

Most heterogeneous network modeling algorithms assume a given or enumerable set of meta-paths, including both directly computing the meta-path-based adjacency matrixes through path counting [32, 39, 4, 28, 38, 16, 8, 33, 45, 46] and network embedding through path sampling [3, 27, 9, 5, 29, 37]. Only until very recently have we found a few works that attempt to deal with the situation where meta-paths are not given. One of them [19], explicitly searches for the important paths based on a tree structure and a heuristic measure. However, the search space can be very large, which leads to problematic efficiency, as we have shown in the experiments. Also, there is no guarantee for finding the best paths, as there is no clear correlation between the heuristic measure and similarity modeling performance. Another work [4], which is very similar in spirit to [19], tries to grow larger meta-graphs from smaller ones, again suffers from efficiency problems even with the optimization tricks and has no guarantee towards optimal similarity model, because the meta-graph discovery and proximity learning steps are separate.

The natural end-to-end learning of useful paths is through reinforcement learning, which has been explored on knowledge bases [41, 2]. However, the two works both consider discrete policy gradient, which boils down to breadth-first search from one end or both ends of the paths, which, to be strict, is essentially not reinforcement learning. As we discussed in our preliminaries, while the node types can be modeled by discrete softmax functions, node contents are too complex for softmax, which naturally calls for continuous representations.

Our continuous policy gradient is smart in two folds. The first is it does not require explicit breadth-first search. Instead, the reinforcement learning agent starts with randomly exploring the paths, and gradually learns the preferable

paths by estimating and leveraging similar paths in the embedding space. The second is it can be naturally combined with content embedding through joint training, thus effectively differentiating instances of the same meta-paths, which might carry different semantic meanings. Therefore, we believe continuous reinforcement learning is the right solution for automatic path discovery on content-rich heterogeneous networks.

## 2.2 Meta-Paths or Semantic Paths?

In this paper, we leverage the power of heterogeneous networks in modeling multi-typed nodes and links. However, we also leverage the rich contents around nodes. To some extent, node types and contents are similar in nature: What should be modeled as *types* and what should be modeled as *contents* is actually an open question, and in practice, it really depends on what is *preferred* or *stressed* by the person who constructs the heterogeneous networks.

Careful readers might notice that in our model, by representing node contents and types together into the same embedding space, we intentionally weaken the difference between types and contents, because they are essentially similar things. However, we do pay special attention to the information that has been commonly modeled by node types through the discrimination loss. The implication of this treatment on the reinforcement learning agent is that, it is thus able to look for paths characterized by both types and contents, rather than just node types as all existing algorithms focus on. Therefore, it is indeed inaccurate to say our algorithm is automatically discovering meta-paths, but rather it is discovering certain semantic paths.

In our experiments, we find both the reconstruction loss and the discrimination loss are playing important roles, while the discrimination loss does matter more. This might also explain why people commonly model those node types as they are.

## 2.3 Shorter Paths?

Another outstanding advantage of our algorithm is we do not need to set a maximum length of paths to be explored. Rather, by allowing the reinforcement learning agent to go back to the start node at every step, we give it the chance to learn when it should stop exploring paths that are too long. We conjecture that this might lead the agent to always prefer shorter paths by keeping returning to the start since in this way it might get to the target node through fewer steps and thus get more rewards. However, we find in our experiments that the agent can be smart enough to decide on what networks to explore longer paths (such as DBLP) and on what else to prefer shorter ones (such as IMDb). Therefore, we believe our algorithm truly breaks free the requirement of given maximum length of paths. Such requirements are not practical at all but assumed by almost all existing frameworks.

## 2.4 Random Paths?

Finally, while we were unable to conduct comparison experiments with two very recent works on heterogeneous network proximity learning [16, 17], our work is conceptually different from theirs: Our algorithm starts with randomly exploring the paths, but efficiently learns to be able to choose the better paths along the way, while their algorithms always randomly sample the paths, which should be way less efficient and may be unable to scale to really large networks like the ones we use in our experiments. As we also notice, their experiments were all done on quite small networks with the largest one being a DBLP dataset with a few thousand authors and the smaller ones like a LinkedIn dataset and a Facebook dataset with a few hundred users, which might indicate a lack of scalability, which renders them hardly applicable to experimental settings.

## 3 Demo

We released our code with a demo function on Github<sup>1</sup>, and also included it together with this submission in the *Supplementary Materials*.

Since the submission website does not accept *.zip* files this year, we changed the suffix into *.doc*. Please change it back to *.zip* in order to properly access the code. If you have any problem opening the file, please find the code through our Github link.

Here we briefly introduce how to run our code and discuss about some results, while more details can be found through playing with our demo!

Our AUTOPATH pipeline is implemented with TensorFlow<sup>2</sup> on Python2. Please make sure you have the newest version of both of them. To train the model on our default IMDb dataset (provided together with the code), simply use the command `python2 train.py`. You may also change the parameters as you like in `config.py`. With our default parameter settings, training the model roughly takes a few minutes on CPU. Note that, you need to remove the `tmp` folder if you change the neural network structures before training a new model. If you have any problem running the code, please feel free to contact us.

After training the model, use the command `python2 demo.py` to play with the demo. This demo basically allows you to look for similar movies regarding genres. Note that finding movies with exactly the same set of genres is very challenging because genres can be multiple and ambiguous. Nonetheless, our results show that most movies returned by our model at least share a few genres with the queried movie. To play with the demo, input one or more movie ids (separated by space) at each time. In Figure 1, we provide some example inputs and outputs of the demo.

As we can see through this demo, our algorithm can efficiently capture the semantic similarities among entities in content-rich heterogeneous networks. The

---

<sup>1</sup> <https://github.com/yangji9181/AutoPath>

<sup>2</sup> <https://www.tensorflow.org/>

### Example Input

```
0
```

or

```
0 1 2
```

### Example Output

```
Top 10 related movies for <the_van>, ['Comedy', 'Drama']
<michael_collins>, ['Drama', 'Biography', 'Thriller', 'War']
<the_butcher_boy>, ['Comedy', 'Drama']
<his_girl_friday>, ['Comedy', 'Romance', 'Drama']
<the_madness_of_king_george>, ['Comedy', 'Drama', 'Biography', 'History']
<the_cook_the_thief_his_wife_&_her_lover>, ['Crime', 'Comedy', 'Horror', 'Romance', 'Drama']
<tin_men>, ['Comedy', 'Drama']
<the_beverly_hillbillies>, ['Comedy', 'Family']
<an_american_in_paris>, ['Romance', 'Musical']
<sweet_nothing>, ['Drama']
```

**Fig. 1.** Example inputs and outputs of our demo.

model effectively explores paths on the network, differentiating paths with different nodes, while avoiding the need to go through every possible path, thanks to our appropriate continuous reinforcement learning with deep embedding techniques. The model does not need to be fully trained (*e.g.*, till convergence) to produce reasonable results.

## Acknowledgement

This research was sponsored in part by the U.S. Army Research Laboratory under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), National Science Foundation IIS-1320617 and IIS 16-18481, and grant 1U54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative ([www.bd2k.nih.gov](http://www.bd2k.nih.gov)).

The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies of the U.S. Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

## References

1. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in neural information processing systems. pp. 2787–2795 (2013)
2. Das, R., Dhuliawala, S., Zaheer, M., Vilnis, L., Durugkar, I., Krishnamurthy, A., Smola, A., McCallum, A.: Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. In: ICLR (2018)

3. Dong, Y., Chawla, N.V., Swami, A.: metapath2vec: Scalable representation learning for heterogeneous networks. In: KDD. pp. 135–144 (2017)
4. Fang, Y., Lin, W., Zheng, V.W., Wu, M., Chang, K., Li, X.L.: Semantic proximity search on graphs with metagraph-based learning. In: ICDE. pp. 277–288 (2016)
5. Fu, T.y., Lee, W.C., Lei, Z.: Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In: CIKM. pp. 1797–1806 (2017)
6. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: SIGKDD. pp. 855–864. ACM (2016)
7. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems. pp. 1025–1035 (2017)
8. Hou, S., Ye, Y., Song, Y., Abdulhayoglu, M.: Hindroid: An intelligent android malware detection system based on structured heterogeneous information network. In: KDD. pp. 1507–1515 (2017)
9. Huang, Z., Mamoulis, N.: Heterogeneous information network embedding for meta path based proximity. arXiv preprint arXiv:1701.05291 (2017)
10. Jeh, G., Widom, J.: Simrank: a measure of structural-context similarity. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 538–543. ACM (2002)
11. Jeh, G., Widom, J.: Scaling personalized web search. In: Proceedings of the 12th international conference on World Wide Web. pp. 271–279. Acm (2003)
12. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: A survey. JAIR 4, 237–285 (1996)
13. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (2017)
14. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. pp. 1097–1105 (2012)
15. Le, Q.V.: Building high-level features using large scale unsupervised learning. In: ICASSP. pp. 8595–8598 (2013)
16. Liu, Z., Zheng, V.W., Zhao, Z., Zhu, F., Chang, K., Wu, M., Ying, J.: Semantic proximity search on heterogeneous graph by proximity embedding. In: AAAI. pp. 154–160 (2017)
17. Liu, Z., Zheng, V.W., Zhao, Z., Zhu, F., Chang, K., Wu, M., Ying, J.: Distance-aware dag embedding for proximity search on heterogeneous graphs. In: AAAI (2018)
18. Lyu, T., Zhang, Y., Zhang, Y.: Enhancing the network embedding quality with structural similarity. In: CIKM. ACM (2017)
19. Meng, C., Cheng, R., Maniu, S., Senellart, P., Zhang, W.: Discovering meta-paths in large heterogeneous information networks. In: WWW. pp. 754–764 (2015)
20. Mikolov, T., Karafiát, M., Burget, L., Černocký, J., Khudanpur, S.: Recurrent neural network based language model. In: Eleventh Annual Conference of the International Speech Communication Association (2010)
21. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS. pp. 3111–3119 (2013)
22. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. Nature 518(7540), 529 (2015)
23. Niepert, M., Ahmed, M., Kutzkov, K.: Learning convolutional neural networks for graphs. In: International conference on machine learning. pp. 2014–2023 (2016)

24. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: EMNLP. pp. 1532–1543 (2014)
25. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: SIGKDD. pp. 701–710. ACM (2014)
26. Ribeiro, L.F., Saverese, P.H., Figueiredo, D.R.: struc2vec: Learning node representations from structural identity. In: SIGKDD. pp. 385–394. ACM (2017)
27. Shang, J., Qu, M., Liu, J., Kaplan, L.M., Han, J., Peng, J.: Meta-path guided embedding for similarity search in large-scale heterogeneous information networks. arXiv preprint arXiv:1610.09769 (2016)
28. Shi, Y., Chan, P.W., Zhuang, H., Gui, H., Han, J.: Prep: Path-based relevance from a probabilistic perspective in heterogeneous information networks. In: KDD. pp. 425–434 (2017)
29. Shi, Y., Gui, H., Zhu, Q., Kaplan, L., Han, J.: Aspem: Embedding learning by aspects in heterogeneous information networks. In: SDM (2018)
30. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al.: Mastering the game of go without human knowledge. *Nature* 550(7676), 354 (2017)
31. Sun, Y., Han, J.: Mining heterogeneous information networks: principles and methodologies. *Synthesis Lectures on Data Mining and Knowledge Discovery* 3(2), 1–159 (2012)
32. Sun, Y., Han, J., Yan, X., Yu, P.S., Wu, T.: Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *VLDB* 4(11), 992–1003 (2011)
33. Sun, Y., Norick, B., Han, J., Yan, X., Yu, P.S., Yu, X.: Integrating meta-path selection with user-guided object clustering in heterogeneous information networks. *TKDD* 7(3), 11 (2013)
34. Sutton, R.S., Barto, A.G.: *Reinforcement learning: An introduction*, vol. 1. MIT press Cambridge (1998)
35. Sutton, R.S., McAllester, D.A., Singh, S.P., Mansour, Y.: Policy gradient methods for reinforcement learning with function approximation. In: NIPS. pp. 1057–1063 (2000)
36. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: Large-scale information network embedding. In: WWW. pp. 1067–1077 (2015)
37. Wan, M., Ouyang, Y., Kaplan, L., Han, J.: Graph regularized meta-path based transductive regression in heterogeneous information network. In: SDM. pp. 918–926 (2015)
38. Wang, C., Song, Y., Li, H., Zhang, M., Han, J.: Knowsim: A document similarity measure on structured heterogeneous information networks. In: ICDM. pp. 1015–1020 (2015)
39. Wang, C., Sun, Y., Song, Y., Han, J., Song, Y., Wang, L., Zhang, M.: Relsim: relation similarity search in schema-rich heterogeneous information networks. In: SDM. pp. 621–629 (2016)
40. Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: SIGKDD. pp. 1225–1234. ACM (2016)
41. Xiong, W., Hoang, T., Wang, W.Y.: DeepPath: A reinforcement learning method for knowledge graph reasoning. In: EMNLP (2017)
42. Yang, C., Liu, M., Wang, Z., Liu, L., Han, J.: Graph clustering with dynamic embedding. arXiv preprint arXiv:1712.08249 (2017)
43. Yang, Z., Cohen, W., Salakhutdinov, R.: Revisiting semi-supervised learning with graph embeddings. In: ICML (2016)

44. Zhang, M., Chen, Y.: Weisfeiler-lehman neural machine for link prediction. In: SIGKDD. pp. 575–583. ACM (2017)
45. Zhao, H., Yao, Q., Li, J., Song, Y., Lee, D.L.: Meta-graph based recommendation fusion over heterogeneous information networks. In: KDD. pp. 635–644 (2017)
46. Zhuang, H., Zhang, J., Brova, G., Tang, J., Cam, H., Yan, X., Han, J.: Mining query-based subnetwork outliers in heterogeneous information networks. In: ICDM. pp. 1127–1132 (2014)