InDNI: An Infection Time Independent Method for Diffusion Network Inference

Guoxin Chen[1], Yongqing Wang *[1], Jiangli Shao[1], Huawei Shen[1], and Xueqi Cheng[1]

Data Intelligence System Research Center, Institute of Computing Technology,
Chinese Academy of Sciences
gx.chen.chn@gmail.com, {wangyongqing, shaojiangli19z, shenhuawei, cxq}@ict.ac.cn

Abstract. Diffusion network inference aims to reveal the message propagation process among users and has attracted many research interests due to the fundamental role it plays in some real applications, such as rumor-spread forecasting and epidemic controlling. Most existing methods tackle the task with exact node infection time. However, collecting infection time information is time-consuming and labor-intensive, especially when information flows are huge and complex. To combat the problem, we propose a new diffusion network inference algorithm that only relies on infection states. The proposed method first encodes several observation states into a node infection matrix and then obtains the node embedding via the variational autoencoder(VAE). Nodes with the least Wasserstein distance of embeddings are predicted for existing propagation edges. Meanwhile, to reduce the complexity, a novel clustering-based filtering strategy is designed for selecting latent propagation edges. Extensive experiments show that the proposed model outperforms the state-of-the-art infection time independent models while demonstrating comparable performance over infection time based models.

Keywords: Diffusion Network Inference · Variational Autoencoder · Wasserstein Distance.

1 Introduction

The topology of a diffusion network reveals how information is propagated among users, intuitively illustrating potential information propagation paths. Promoting and preventing future diffusion on the network are of great importance[23]. Generally, observing the influence relationship between users in the real scenarios is difficult, thus researchers try to recover the relations with the historical propagation[9], which is known as the diffusion network inference task.

According to whether the node infection time is used or not, most of the existing diffusion network inference methods can be divided into two categories: the infection time dependent and infection time independent algorithms[11]. The infection time dependent algorithm mostly assume that previously infected nodes are potential parents of subsequently infected nodes. It infers the influence relationship between nodes by constructing and maximizing different likelihood functions based on the exact infection time[15]. However, in some realistic scenarios, monitoring and recording the node infection time is labor demanding and time consuming, limiting the performance and application of the algorithm. Therefore, works on reconstructing the diffusion network structure without infection time have emerged. Two existing methods[8,2] attempt to learn influence relationships between nodes from all fixed-length path trajectories or from initial and resulting infected node sets. However, the former requires obtaining all fixed-length path trajectories, the latter requires prior knowledge of the number of edges in the diffusion network. These are difficult to obtain in practice.

In order to address the above problems, we proposes a novel infection time independent network inference method called InDNI (An Infection Time **In**dependent Method for **D**iffusion **N**etwork **I**nference). InDNI takes final states of nodes (infected or not) in different propagation processes as inputs, which is easier to obtain than the exact infection time or infection sequences. During information propagation, user pairs with follower relationships tend to exhibit similar behaviors (infected or not infected at the same time). Therefore, the final infection state of a node can well reflect the influence relationship between nodes. InDNI utilizes variational auto-encoders[4] to extract the behavioral characteristics of nodes. Meanwhile, wasserstein distance[24] is utilized to measure the similarity between node pairs and further infer the adjacent edge relationship between nodes. Furthermore, to address the problem that nodes rarely involved in propagation will be aggregated within the embedding space and be misjudged as having adjacent edges, InDNI excludes node pairs with very low probability of adjacent edges by examining the correlation of the infection mutual information(IMI) metric [9] between nodes. In summary, the main contributions of this paper are summarized as follows:

- We propose a novel diffusion network inference algorithm named InDNI. Compared to existing algorithms, InDNI relies only on the final infection status of nodes, which is easier to obtain in practice.
- An effective filtering strategy based on infection mutual information is proposed for not only accelerating the prediction process but also reducing the bias led by infrequent nodes.
- Experimental results on synthetic and real-world dataset demonstrates that InDNI outperforms the state-of-the-art infection time independent models while showing comparable performance over infection time based models.

2 Related Work

According to whether the infection time is utilized, diffusion network inference can be divided into the following two categories: (1) The infection time-based algorithm; (2) The infection time-independent algorithms.

The infection time-based algorithm. Most of the infection time-based network inference algorithms are based on the following assumptions: nodes that are sequentially infected within a time interval have influence relationships, and previously infected nodes are considered as potential parents of subsequently infected nodes. Therefore, these methods rely on explicit temporal information for each node. Representative algorithms are: InfoPath[21], NetINF[7], REFINE[13], etc. Depending on the differences in the principles, this types of methods can be subdivided into: the convex programming-based approaches, the submodularity-based approaches, and the embedding-based approaches. Although the infection time-based algorithm is effective in some scenarios to solve the network inference problem, it is a very difficult task to obtain the data with the accurate infection time during the propagation. This undoubtedly challenges the applicability of the algorithm.

The infection time-independent algorithms. To solve the problem of time collection difficulties, some works try to learn the influence relationships between nodes from diffusion path traces (the Path approach[8]), based on lifting effects (the LIFT approach[2]) or the node infection sequence (the DeepINFER approach[12]). The PATH approach requires obtaining all fixed-length path trajectories. The LIFT approach requires a priori knowledge of the number of edges in the diffusion network, otherwise it infers a fully connected graph. All these methods have strong a priori assumptions that are not conducive to the application of the algorithm. Subsequently, DeepINFER proposes to compare the infection sequence context of a node with the textual context and learn the node representation by Skip-Gram model[20]. It gets rid of the dependence on infection time to a certain extent. However, due to the hysteresis of information propagation, there is not necessarily a direct influence relationship between the context node and the central node, which will make the model mistakenly believe that there is a high probability of adjacent edges between the context node and the central node.

3 Problem Statement

The diffusion network can be represented as a graph $G = \{V, E\}$, where $V = \{v_1, v_2, ..., v_n\}$ refers to the set of n nodes, and E denotes the set of m edges between nodes. In the diffusion network inference problem, the set of nodes is given and the set of edges is unknown and needs to be inferred. In this paper, we assume that the propagation result only contains the final infection state of the node, not the infection time or sequence. Therefore, the formal definition of our problem statement is given (Table 1 gives the main symbols in this paper):

Given: *On the diffusion network G, given a set of node infection status results $C = \{C^1, C^2, \ldots, C^k\}$ during k times of historical propagation. The element $C^l = \{c_1^l, c_2^l, \ldots, c_n^l\}$ is an n-dimensional vector (n is the number of nodes in G) which records the final infection status of each node in the l^{th} propagation, and $c_i^l = \{0, 1\}$ (0 refers to not infected, 1 refers to infected).*

Infer: *The unknown edge set E of the diffusion network G.*

Table 1. A brief summary of notions.

Symbol	Description
G	A diffusion Network.
V, E	The node set and edge set of network G.
n, m	The number of nodes and edges of network G.
C	The observed infection status of nodes in G during historical propagation.
X	Node infection matrix extracted from C.
S_{pair}	The set of candidate node pairs, i.e., node pairs that may have adjacent edges.

4 InDNI Algorithm

From the perspective of graph representation learning, the basic idea of the diffusion network inference is to learn the node representation according to the result of propagation, and judge whether there is an edge based on the similarity between nodes. Therefore, InDNI extracts the behavioral features of nodes in the process of reconstructing the infection state of nodes with the help of VAE. Then, InDNI describes the similarity between nodes through the Wasserstein distance for diffusion network inference. Meanwhile, in order to solve the problem of aggregation of nodes that are isolated or rarely involved in propagation in the embedding space, InDNI introduces IMI metrics for preliminary filtering to obtain candidate pairs of nodes that may have adjacent edges.

4.1 Node Representation Learning

Node initial features. Based on the assumption that adjacent nodes often have similar behaviors in information propagation, the following definition of the node infection matrix is given as input to the model:

Definition 1 (Node Infection Matrix). *On a diffusion network G containing n nodes, the set of node infection states $C = \{C^1, C^2, \ldots, C^k\}$ during k historical propagation is given. Extracting the node infection matrix from C is defined as $X \in \mathbb{R}^{n\times k}$. The i-th row and j-th column of the matrix X represent the infection of node v_i in the j-th propagation:*

$$X_{ij} = \begin{cases} 1, \, if \, c_i^j = 1 \\ 0, \, if \, c_i^j = 0 \end{cases} \tag{1}$$

The node infection matrix X only depends on the infection results of the propagation, which represents the behavioral tendency of nodes in this group of propagation.

Variational Autoencoder. InDNI maps node infection states to a low-dimensional space through a Variational Autoencoder (VAE) containing multiple non-linear layers, aiming to extract dense behavioral feature vectors of nodes from the node infection matrix X. VAE utilizes Gaussian distribution to describe the probability distribution of node features, whose powerful information extraction capability has been demonstrated in several fields such as graph representation learning[24] and graph generation[14].

Loss Function. The loss function of VAE mainly consists of two parts: reconstruction loss and distribution loss[4], as shown by the Eqs. 2. In the formula, $p(Z) = \prod_i p(z_i) = \prod_i \mathcal{N}(z_i|0, I)$ is a Gaussian prior distribution of the latent variable Z, and $Q(Z|X, \Phi)$ is the Encoder, Φ is the parameter of the Encoder, $P(\hat{X}|Z, \Theta)$ is the Decoder, Θ is the parameter of the Decoder, and KL refers to the KL divergence.

$$\mathcal{L} = E_{Q(Z|X,\Phi)}\left[\log P(\hat{X}|Z, \Theta)\right] - \mathrm{KL}\left[Q(Z|X, \Phi)\|p(Z)\right] \tag{2}$$

The reconstruction loss hopes that the features extracted by the Encoder can restore the input X well. It is worth mentioning that the node infection matrix X is usually sparse. In order to avoid the model from over-learning the sparse part, inspired by existing work, in the reconstruction loss, we assign more penalties to the loss caused by non-zero elements than the loss caused by zero elements. So the reconstruction loss can be defined as:

$$E_{Q(Z|X,\Phi)}\left[\log P(\hat{X}|Z, \Theta)\right] = \sum_{i=0}^{n} \|(X_i - \hat{X}_i) \odot P_i\|_2^2 = \|(X - \hat{X}) \odot P\|_F^2 \tag{3}$$

where $\odot$ denotes the Hadamard product, and the elements of row i and column j of the weight matrix P are defined as follows:

$$P_{ij} = \begin{cases} \rho, \, if \, X_i^j = 1 \\ 1, \, if \, X_i^j = 0 \end{cases} \tag{4}$$

where ρ is the hyperparameter of the model, which is used to adjust the weight of the reconstruction loss of non-zero elements. In the experimental part, the effect of different ρ on the results is explored.

The distribution loss can be thought of as a regularizer. Its goal is to reconstruct a meaningful output X even when the latent variable Z is sampled from a priori distribution $p(Z)$. We assume that the latent variable Z follows a Gaussian distribution, as shown by the Eqs. 5.

$$\begin{aligned} &\mathrm{KL}\left[Q(Z|X, \Phi)||p(Z)\right] = \mathrm{KL}\left[\mathcal{N}(\mu(X), \Sigma(X))\|\mathcal{N}(0, I)\right] \\ &= \frac{1}{2}(\mathrm{Tr}(\Sigma(X)) + \mu(X)^T\mu(X) - k + \log(\det(\Sigma(X)))) \end{aligned} \tag{5}$$

where k denotes the dimensionality of the distribution features, Tr is the trace of the matrix, det is the determinant of the matrix, and the parameters μ and Σ of the Gaussian distribution can be obtained by fitting a neural network while Σ is constrained to be a diagonal matrix.

4.2 Similarity Measure

After obtaining the node representation, a suitable distance metric is needed to describe the similarity between different nodes. Extensive work[24] has demonstrated that the Wasserstein distance is a good measure of the distance between two distributions. It does not suffer from problems similar to the KL divergence or JS divergence that give meaningless or constant results for two distributions that do not overlap at all. This ensures that it can stably describe the similarity between nodes. The q^{th} Wasserstein distance between two probability distributions P_1 and P_2 is defined as :

$$W_q(P_1, P_2) = (\inf \mathbb{E}\left[\mathrm{d}(X, Y)^q\right])^{1/q} \tag{6}$$

where $\mathbb{E}[Z]$ denotes the expected value of a random variable Z and the infimum is taken over all joint distributions of the random variables X and Y with marginals P_1 and P_2 respectively.

However, the general form of Wasserstein distance is limited by the large computational cost and it is difficult to apply it directly to the problem in this paper. To reduce the computational cost, in this paper, the 2nd Wasserstein distance (abbreviated as W_2) has a closed form solution to speed up the computational process since we use a Gaussian distribution as the node representation. In the meanwhile, we focus on diagonal covariance matrices[6][22], i.e., $\Sigma_1\Sigma_2 = \Sigma_2\Sigma_1$. Thus, Eqs. 6 can be simplified as Eqs. 7 [24]. The computational complexity of W_2 scales linearly with the embedding dimension.

$$W_2(\mathcal{N}(\mu_1, \Sigma_1), \mathcal{N}(\mu_2, \Sigma_2)) = (\|\mu_1 - \mu_2\|_2^2 + \|\Sigma_1^{1/2} - \Sigma_2^{1/2}\|_F^2)^{1/2} \tag{7}$$

4.3 Filtering candidate node pairs

There are often isolated nodes or nodes that rarely participate in the propagation in the diffusion network. Since their infection status is basically all zeros, these nodes cluster within the embedding space when projected onto the embedding space via the VAE. Obviously, by the above approach alone, the algorithm would naturally assume that there are adjacent edges between these nodes, which do not actually exist. To solve the above problem, InDNI introduces the Infection-MI metric for initial filtering to obtain the set of candidate node pairs for which adjacent edges may exist.

Infection-MI. Based on the assumption that neighboring nodes tend to have similar behaviors during information propagation, we argue that the higher the behavioral correlation, the higher the probability of the existence of adjacent edges. Inspired by the existing work[9], we introduce the Infection-MI metric (abbreviated as IMI) based on mutual information to measure node behavioral correlation.

$$\mathrm{IMI}(v_i, v_j) = \mathrm{MI}(v_i = 1, v_j = 1) - \mathrm{MI}(v_i = 1, v_j = 0) - \mathrm{MI}(v_i = 0, v_j = 1) \tag{8}$$

According to the above definition, when the behaviors of nodes v_i and v_j are highly negatively correlated, i.e., the values of $\mathrm{MI}(v_i = 1, v_j = 0)$ or $\mathrm{MI}(v_i = 0, v_j = 1)$ is significantly large, $\mathrm{IMI}(v_i, v_j)$ tends to be negative. When the behavior of v_i and v_j tend to be independent, the values of $\mathrm{MI}(v_i = 1, v_j = 1)$, $\mathrm{MI}(v_i = 1, v_j = 0)$ and $\mathrm{MI}(v_i = 0, v_j = 1)$ are very small and the value of IMI is close to 0. When the behavior of A and B have a high correlation, IMI is a relatively large positive value. Therefore, IMI can well reflect the correlation of behaviors between nodes.

Kmeans-based Filtering Method. In diffusion networks, each node v_i usually contains only a limited number of parents that tend to have a large positive correlation with node v_i. Except for a few nodes that have negative correlation with v_i, most of the nodes in the network have no influence relationship on v_i, which leads to a compact cluster with a mean value close to 0 for the IMI metric. To avoid setting hyperparameters and to distinguish weak positive correlation from positive correlation, we introduce a filtering method based on Kmeans clustering[10] to filter out the set of node pairs S_{pair} with possible adjacent edges based on IMI, with the following procedure.

Step 1: Calculate the IMI value between each node in the network and remove the node pairs in which the IMI value is negative.

Step 2: Executing the Kmeans algorithm for the remaining pairs of nodes (one-dimensional clustering), where $K = 2$, fixing one of the cluster centers to 0 and initializing the other cluster center to the maximum of all IMI values, iterating continuously until stability.

Step 3: The node pairs contained in clusters with non-zero cluster centers are treated as the set of candidate node pairs, denoted as S_{pair}, i.e., the node pairs with possible adjacent edges.

Algorithm 1 The InDNI Algorithm

```
Input: The node set $V$, the node infection matrix $X$.
Output: The edge set $E$.
for $i = 1$ to $n$ do
    for $j = i + 1$ to $n$ do
        Calculate $\mathrm{IMI}(v_i, v_j)$ by Eqs. 8.
    end for
end for
Based on IMI, $S_{pair}$ is obtained by Kmeans filtering method.
The node infection matrix $X$ is used as input and the node representation is obtained by iteratively optimizing the VAE parameters according to Eqs. 2.
for $(v_i, v_j) \in S_{pair}$ do
    Calculate $W_2(v_i, v_j)$ by Eqs. 7.
end for
By Eqs. 9, infer the edge set $E$.
```

4.4 Network Inference

Once the representation of each node and the set of candidate node pairs S_{pair} are obtained, the next step is to infer the set of edges E of the diffusion network[12]. We believe that the distributional difference between node pairs with adjacent edges should be smaller, and the distributional difference between node pairs without adjacent edges should be larger. Therefore, we can infer whether there is an edge by computing the similarity of candidate node pairs, which is expressed formally as Eqs. 9.

$$E = \{(u, v) : W_2(u, v) \leq \tau, (u, v) \in S_{pair}\} \tag{9}$$

Up to this point, all the steps have been clarified and the overall flow of InDNI is summarized in Algorithm 1.

5 Experiments

5.1 Experimental Setup

Datasets. To comprehensively observe the capabilities of the InDNI algorithm, we conduct experiments on different scenarios on synthetic and real datasets. The basic statistics of each dataset are shown in Table 2.

Table 2. A brief description about datasets

Type	Networks	n	m	k
Synthetic datasets	Kronecker Random	512	1024	5000
	Kronecker Hierarchical	512	1024	5000
	Kronecker Core-Periphery	512	1024	5000
Real world datasets	Dolphin	62	318	1000
	Polblogs	1490	33430	10000
	Facebook	4039	88234	50000

We first consider the Kronecker model[16] to simulate a real-world diffusion network, which can generate different network structure that approximates the real-world scenario based on different parameter inputs. In this paper, we consider the three most common network structures: Random[5] (parameter matrix: [0.5,0.5; 0.5,0.5]), Hierarchical[3] ([0.9,0.1; 0.1,0.9]) and Core-periphery[17] ([0.9,0.5; 0.5,0.3]).

Although the synthetic model can generate the desired network structure by controlling different parameters, which facilitates the in-depth analysis of the algorithm performance. However, synthetic datasets are not really a substitute for real-world propagation processes, which are often influenced by a large number of external factors and cannot be really accurately described. Therefore, to explore the capability of the algorithm in real-world scenarios, we conducted extensive experiments on real datasets as follows: (1)**Dolphin:**[19] A small undirected social network of frequent contact between 62 dolphins in a community near the New Zealand Strait; (2) **Polblogs:**[1] A network of hyperlinks between blogs about American politics recorded in 2005; (3) **Facebook:**[18] A social network from Facebook that has anonymized user ids.

Baseline methods. Among the existing network inference algorithms, it is mainly divided into infection time-based and infection time-independent network inference algorithms. The accuracy of infection time-based inference algorithms tends to be higher than that of infection time-independent inference algorithms, because the former uses more information and requires higher quality of data. We selected the most representative algorithms of the two major classes as the benchmark algorithms for our experiments, listed as InfoPath[21], Netinf[7], REFINE[13], DeepINFER[12]. To evaluate the accuracy of InDNI in diffusion network inference, we report the F1-score[9] as an evaluation metric.

5.2 Results and Discussion

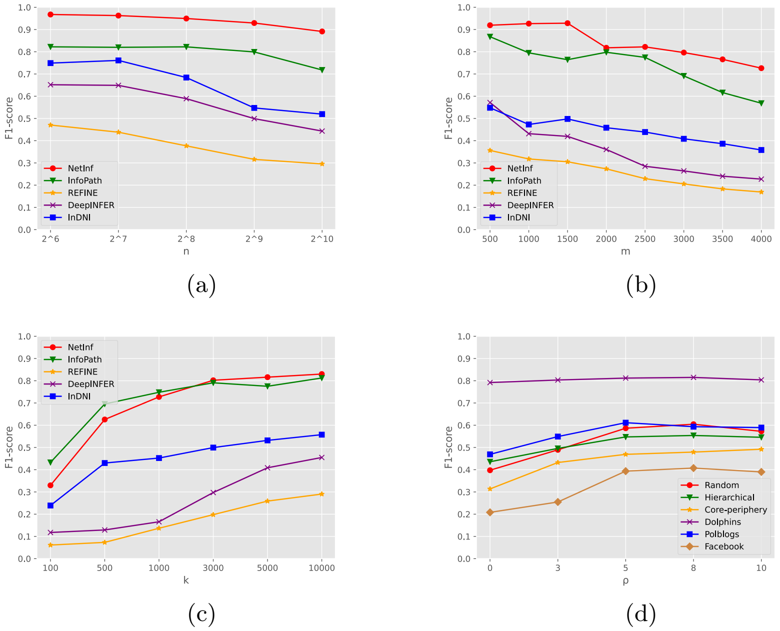
Diffusion network inference. Following the above settings, we compare the performance of the algorithms on six datasets, the results are presented in Table 3. As seen from the table, (1) the accuracy of infection time-based inference algorithms tends to be higher than that of infection time-independent inference algorithms, because the former uses more information and requires higher quality of data. (2) InDNI outperforms DeepINFER on all datasets, while approaching the infection time-based algorithm on some datasets, especially real-world datasets. (3) Comparing the performance of algorithms on synthetic datasets, even with the same network size, different network properties can lead to drastically different results.

Table 3. Diffusion network inference results (F1-score). (The first three methods are infection time-based methods and the last two methods are infection time-independent methods.)

Datasets Alg.	Random	Hierarchical	Core-periphery	Dolphin	Polblogs	Facebook
REFINE	0.316	0.315	0.270	0.520	0.372	0.289
InfoPath	0.796	0.799	0.778	0.752	0.634	0.403
NetINF	0.852	0.929	0.740	0.794	0.683	0.427
DeepINFER	0.464	0.499	0.381	0.792	0.574	0.335
InDNI	0.586	0.547	0.469	0.811	0.611	0.394
	+26.3%	+17.4%	+23.1%	+2.39%	+6.45%	+17.6%

Sensitivity analysis experiments. In this subsection, we explore the effects of the number of nodes, the number of edges, the number of the historical propagation and the hyperparameter ρ on the algorithm, respectively, as shown in Fig. 1.

All experiments are performed on synthetic datasets. We control the Kronecker model to generate Hierarchical networks with uniform network mechanisms but with different parameters. From the figure we can observe that (1) The infection time-independent algorithms are more sensitive to network size than the infection time-based algorithm. (2) Compared with the infection time-independent algorithms, the infection time-based algorithm can obtain the set of potential parent nodes through the node infection time, which helps them achieve better results. (3) When the number of propagation is small, the statistical effect of propagation cannot be fully reflected, and the results of each algorithm are poorer under the influence of biased data. However, InDNI, NetINF and InfoPath have lower requirements for the number of propagations, which is important for practical applications. (4) The loss weight ρ avoids focusing too much on the zero element, and the appropriate parameter make the model significantly improve on each dataset.



(a)

(b)

(c)

(d)

Fig. 1. Sensitivity Analysis Experiments.(a)Effects of the number of nodes;(b)Effects of the number of edges;(c)Effects of the number of propagation;(d)Effects of hyperparameter ρ.

Ablation experiment. To verify the effectiveness as well as the necessity of each part of the algorithm, we explored the performance of co-occurrence, MI, IMI, and InDNI without IMI versus InDNI, respectively, as shown in Fig. 2. From which we can observe that (1) The experimental results of IMI, InDNI without IMI and InDNI shows that filtering method, which plays an important role in the overall performance of the algorithm, does solve the problem of aggregation of isolated nodes in the embedding space. (2) The results of IMI are all better than MI, justifying the rationality and necessity of IMI. (3) The combination of node representation and filtering methods can achieve better results.

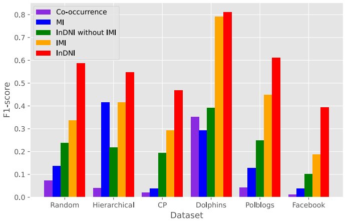


Fig. 2. Ablation Experiment.

6 Conclusion and Future Work

In this paper, we propose an algorithm InDNI for diffusion network inference from the final infection status of nodes only. Compared with existing algorithms, InDNI does not depend on infection time or infection sequence and does not have strong a priori assumptions. Experiments on a large number of synthetic and real-world datasets demonstrate that InDNI outperforms other algorithms that do not rely on infection time, and is able to approach the performance of infection time based algorithms on some datasets.

In the future, we will subsequently consider data collection error detection and complementary algorithms for more precise prediction. In addition, the InDNI algorithm still has a gap with the infection time based algorithm. We believe that the learning of node representation can be improved and will explore it in future work.

References

1. Adamic, L.A., Glance, N.: The political blogosphere and the 2004 us election: divided they blog. In: Proceedings of the 3rd international workshop on Link discovery. pp. 36–43 (2005) 5.1
2. Amin, K., Heidari, H., Kearns, M.: Learning from contagion (without timestamps). In: International Conference on Machine Learning. pp. 1845–1853. PMLR (2014) 1, 2
3. Clauset, A., Moore, C., Newman, M.E.: Hierarchical structure and the prediction of missing links in networks. Nature **453**(7191), 98–101 (2008) 5.1
4. Doersch, C.: Tutorial on variational autoencoders. arXiv preprint arXiv:1606.05908 (2016) 1, 4.1
5. Erdos, P., Rényi, A., et al.: On the evolution of random graphs. Publ. Math. Inst. Hung. Acad. Sci **5**(1), 17–60 (1960) 5.1
6. Givens, C.R., Shortt, R.M.: A class of wasserstein metrics for probability distributions. Michigan Mathematical Journal **31**(2), 231–240 (1984) 4.2
7. Gomez-Rodriguez, M., Leskovec, J., Krause, A.: Inferring networks of diffusion and influence. ACM Transactions on Knowledge Discovery from Data (TKDD) **5**(4), 1–37 (2012) 2, 5.1

8. Gripon, V., Rabbat, M.: Reconstructing a graph from path traces. In: 2013 IEEE International Symposium on Information Theory. pp. 2488–2492. IEEE (2013) 1, 2
9. Han, K., Tian, Y., Zhang, Y., Han, L., Huang, H., Gao, Y.: Statistical estimation of diffusion network topologies. In: 2020 IEEE 36th International Conference on Data Engineering (ICDE). pp. 625–636. IEEE (2020) 1, 4.3, 5.1
10. Hartigan, J.A., Wong, M.A.: Algorithm as 136: A k-means clustering algorithm. Journal of the royal statistical society. series c (applied statistics) **28**(1), 100–108 (1979) 4.3
11. Huang, H., Yan, Q., Gan, T., Niu, D., Lu, W., Gao, Y.: Learning diffusions without timestamps. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 582–589 (2019) 1
12. Kefato, N.Z., Montresor, A.: Deepinfer: Diffusion network inference through representation learning. In: Proc. 13th Int. Workshop Mining Learn. Graphs. p. 5 (2017) 2, 4.4, 5.1
13. Kefato, Z.T., Sheikh, N., Montresor, A.: Refine: Representation learning from diffusion events. In: International Conference on Machine Learning, Optimization, and Data Science. pp. 141–153. Springer (2018) 2, 5.1
14. Kipf, T.N., Welling, M.: Variational graph auto-encoders. arXiv preprint arXiv:1611.07308 (2016) 4.1
15. Kurashima, T., Iwata, T., Takaya, N., Sawada, H.: Probabilistic latent network visualization: Inferring and embedding diffusion networks. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 1236–1245 (2014) 1
16. Leskovec, J., Chakrabarti, D., Kleinberg, J., Faloutsos, C., Ghahramani, Z.: Kronecker graphs: an approach to modeling networks. Journal of Machine Learning Research **11**(2) (2010) 5.1
17. Leskovec, J., Lang, K.J., Dasgupta, A., Mahoney, M.W.: Statistical properties of community structure in large social and information networks. In: Proceedings of the 17th international conference on World Wide Web. pp. 695–704 (2008) 5.1
18. Leskovec, J., Mcauley, J.: Learning to discover social circles in ego networks. Advances in neural information processing systems **25** (2012) 5.1
19. Lusseau, D., Schneider, K., Boisseau, O.J., Haase, P., Slooten, E., Dawson, S.M.: The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. Behavioral Ecology and Sociobiology **54**(4), 396–405 (2003) 5.1
20. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems **26** (2013) 2
21. Rodriguez, M.G., Leskovec, J., Balduzzi, D., Schölkopf, B.: Uncovering the structure and temporal dynamics of information propagation. Network Science **2**(1), 26–65 (2014) 2, 5.1
22. Wang, H., Banerjee, A.: Bregman alternating direction method of multipliers. In: Advances in Neural Information Processing Systems. vol. 27. Curran Associates, Inc. (2014) 4.2
23. Xia, Y., Chen, T.H.Y., Kivelä, M.: Applicability of multilayer diffusion network inference to social media data. arXiv preprint arXiv:2111.06235 (2021) 1
24. Zhu, D., Cui, P., Wang, D., Zhu, W.: Deep variational network embedding in wasserstein space. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 2827–2836 (2018) 1, 4.1, 4.2, 4.2