DESTINE: Dense Subgraph Detection on Multi-Layered Networks

Zhe Xu*, Si Zhang*, Yinglong Xia†, Liang Xiong†, Jiejun Xu‡, and Hanghang Tong*
* University of Illinois at Urbana-Champaign, {zhexu3,sizhang2,htong}@illinois.edu
† Facebook, {yxia,lxiong}@fb.com
‡ HRL Laboratories, jxu@hrl.com

ABSTRACT

Dense subgraph detection is a fundamental building block for a variety of applications. Most of the existing methods aim to discover dense subgraphs within either a single network or a multi-view network while ignoring the informative node dependencies across multiple layers of networks in a complex system. To date, it largely remains a daunting task to detect dense subgraphs on multi-layered networks. In this paper, we formulate the problem of dense subgraph detection on multi-layered networks based on cross-layer consistency principle. We further propose a novel algorithm DESTINE based on projected gradient descent with the following advantages. First, armed with the cross-layer dependencies, DESTINE is able to detect significantly more accurate and meaningful dense subgraphs at each layer. Second, it scales linearly w.r.t. the number of links in the multi-layered network. Extensive experiments demonstrate the efficacy of the proposed DESTINE algorithm in various cases.

CCS CONCEPTS

• **Mathematics of computing** → **Graph algorithms**; • **Information systems** → **Data mining**.

KEYWORDS

dense subgraph detection; multi-layered network

ACM Reference Format:
Zhe Xu*, Si Zhang*, Yinglong Xia†, Liang Xiong†, Jiejun Xu‡, and Hanghang Tong*. 2021. DESTINE: Dense Subgraph Detection on Multi-Layered Networks. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3459637.3482083

1 INTRODUCTION

Dense subgraph detection which aims to extract tightly connected components from the underlying graph is a fundamental problem in many data mining applications, such as community detection [16], follower-buying service detection [14] and protein complexes detection [25]. Despite of extensive research, many existing works focus on detecting dense subgraphs on a single network such as densest subgraph detection [7, 13], α-quasi-clique detection [1, 6], non-negative matrix approximation [5, 10].

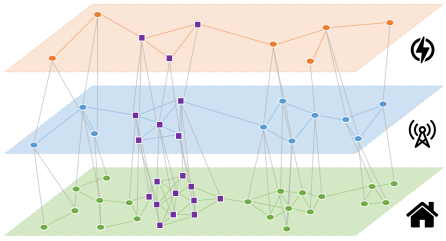

Figure 1: An illustrative example of the dense subgraph detection problem on a three-layered network. Orange, blue, and green nodes denote power plants, communication stations and properties, respectively. The purple rectangle nodes compose the target dense subgraph in each layer.

To leverage the rich side information (e.g., node/edge attributes) accompanied with many networks, some existing works [11, 18, 28] strive to detect dense subgraphs on multi-view networks where each view of the network is constructed by certain side information. However, these works naturally assume the same set of nodes across different views. In the meanwhile, a group of works [4, 23] study the dense subgraph detection problem in the setup of heterogeneous networks [19] to consider different types of nodes and edges. Yet, these methods often require domain expertise to design meta paths.

In many real-world scenarios, networked data can be modelled through inter-dependent layers with different sets of nodes [15, 26]. For illustration, we present a three-layered network in Figure 1 whose layers represent a power network, a communication network and a property network, respectively. In this case, nodes within a single layer are connected based on the geographical proximities, whereas the links across different layers indicate the *cross-layer dependencies*. For example, the cross-layer dependencies between the top two layers imply that a power plant supplies power to various communication stations, and a communication station can be supported by multiple power sources. The cross-layer dependencies between the bottom two layers show that a communication station provides services to multiple properties and residents in a property can choose different communication services. Moreover, dense subgraphs at different layers tend to depend on each other. In Figure 1, infrastructures support each other within a local area due to the supplies and demands for power, population, resources, etc. More importantly, if there is a dense subgraph on the power plant network, the subgraphs of the corresponding communication services and properties in this area are likely to be dense as well.

Unfortunately, existing methods cannot fully handle dense subgraph detection on the multi-layered networks due to the following two key challenges. First (*C1. Formulation*), it is not clear how to

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia

ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3482083

jointly encode both within-layer density and cross-layer dependency into the dense subgraph detection problem. Second (*C2. Algorithm*), the relaxed optimization problem underlying most dense subgraph detection problems is already non-convex, and introducing the cross-layer dependency information might make the problem even harder. In this paper, we propose a novel algorithm DESTINE that aims to address these two challenges. The main contributions of the paper are summarized as follows.

Problem Formulation. We formulate dense subgraph detection on multi-layered networks as an optimization problem, based on cross-layer consistency principle [15].

Algorithm and Analysis. We propose an efficient algorithm DESTINE and analyze its convexity and complexity.

Experiments. We conduct comprehensive experiments to demonstrate the effectiveness and scalability of DESTINE.

2 PROBLEM DEFINITION

We use bold uppercase letters for matrices (e.g., $\mathbf{A}$), bold lowercase letters for vectors (e.g., $\mathbf{u}$) where $\mathbf{u}(x)$ denotes the x-th element of vector $\mathbf{u}$, and lowercase letters for scalars (e.g., c). We denote the transpose of a matrix/vector by the superscript $'$ (e.g., $\mathbf{A}'$ as the transpose of $\mathbf{A}$). We follow the definition of multi-layered network in [8] and use the following notations to describe a multi-layered network with g layers. First, we denote a set of adjacency matrices $\mathcal{A} = \{\mathbf{A}_i | i = 1, ..., g\}$ to describe the network structure of each layer where $\mathbf{A}_i \in \{0,1\}^{n_i \times n_i}$, and n_i is the number of nodes at the i-th layer. Second, we represent the cross-layer dependency matrices as $C = \{\mathbf{C}_{i,j} | i, j = 1, ..., g, i \neq j\}$ where $\mathbf{C}_{i,j} \in \{0,1\}^{n_i \times n_j}$. If node-$x$ at the i-th layer depends on node-y at the j-th layer, $\mathbf{C}_{i,j}(x,y) = 1$. In addition, we use the selection vector $\mathbf{s}_i$ to represent the nodes in the extracted dense subgraph at the i-th layer. Specifically, if node-x at the i-th layer is included in the extracted dense subgraph, $\mathbf{s}_i(x) = 1$. Otherwise, $\mathbf{s}_i(x) = 0$. We formally define the dense subgraph detection problem on multi-layered networks as follows.

PROBLEM 1. *Dense Subgraph Detection on Multi-Layered Networks.*

Given: *a multi-layered network with (1) a set of adjacency matrices $\mathcal{A}$, (2) a set of cross-layer dependency matrices C.*

Find: *a set of selection vectors $\{\mathbf{s}_i | i = 1, ..., g\}$ indicating the detected dense subgraphs at each layer.*

3 METHODS

Objective Function. We formulate Problem 1 by Eq. (1) with two parts, including within-layer objective function L_d and cross-layer objective function L_r

$$\min_{\{\mathbf{s}_i\}} \underbrace{\sum_{i=1}^{g} L_d(\mathbf{s}_i)}_{\text{within-layer objective}} + \underbrace{\sum_{i,j=1}^{g} \gamma_{i,j} L_r^{i,j}(\mathbf{s}_i, \mathbf{s}_j)}_{\text{cross-layer objective}} \quad (1)$$

$$s.t. \quad \forall i,\ \mathbf{s}_i \in \{0,1\}^{n_i}$$

where i, j are layer indices, $\mathbf{s}_i$ is the node selection vector at the i-th layer and $\gamma_{i,j}$ is the hyper-parameter that denotes the importance of the regularization term between the i-th and the j-th layers.

Remarks. Compared with the methods which focus on a single network (e.g. power grid), our method utilizes the rich cross-layer information by L_r to obtain better performance. On the other hand, compared with methods mixing different layers into a heterogeneous network, our method explicitly tells apart the differences between within-layer links and cross-layer links to render the flexibility of integrating existing single network-based methods.

For the within-layer objective, we adopt the *edge surplus*-based objective function [21] which equivalently measures the density as the difference between the number of existing links and that of missing links in the detected subgraph [27], i.e.,

$$L_d(\mathbf{s}_i) = p\mathbf{s}_i'(\mathbf{1}_i\mathbf{1}_i' - \mathbf{I}_i - \mathbf{A}_i)\mathbf{s}_i - \mathbf{s}_i'\mathbf{A}_i\mathbf{s}_i \quad (2)$$

where $\mathbf{1}_i$ is a vector of length n_i whose elements are all set as 1s, $\mathbf{I}_i$ is an identity matrix of size $n_i \times n_i$ and p is the hyper-parameter denoting the penalty weight of missing links in the detected subgraph. When minimizing the above objective w.r.t. the selection vector $\mathbf{s}_i$, a small $L_d(\cdot)$ would imply a high density of the subgraph in terms of edge surplus.

As aforementioned, dense subgraphs at different layers tend to highly depend on each other. To formulate such information, we graft the *cross-layer consistency* principle which was originally proposed for ranking tasks on network of networks [15]. The core idea behind cross-layer consistency is that the influence (e.g., the ranking score) of a pair of nodes from two layers should be similar if they are strongly dependent on each other. We instantiate cross-layer consistency, in the context of dense subgraph detection, as a set of regularization terms between the selection vectors as follows.

$$L_r^{i,j}(\mathbf{s}_i, \mathbf{s}_j) = ||\mathbf{C}_{i,j} \odot (\mathbf{C}_{i,j} - \mathbf{s}_i\mathbf{s}_j' - (\mathbf{1}_i - \mathbf{s}_i)(\mathbf{1}_j - \mathbf{s}_j)')||_F^2 \quad (3)$$

where $\odot$ denotes the Hadamard product and $||\cdot||_F$ denotes the Frobenius norm. The key idea of Eq. (3) is that for observed cross-layer dependency $\mathbf{C}_{i,j}(x,y) = 1$, node-$x$ in the i-th layer and node-y in the j-th layer should be either *both* selected or not selected simultaneously as a member of the dense subgraph in the corresponding layer (i.e., $\mathbf{s}_i(x) = \mathbf{s}_j(y)$). Otherwise, $(\mathbf{C}_{i,j} - \mathbf{s}_i\mathbf{s}_j' - (\mathbf{1}_i - \mathbf{s}_i)(\mathbf{1}_j - \mathbf{s}_j)')(x,y) = 1$, which leads to the penalty of the objective function. In other words, if two nodes are inter-dependent with each other, their selection status should be similar as well, which aligns well with the spirit of the cross-layer consistency principle. In addition, through the Hadamard product, if two nodes are not linked by any cross-layer dependency (i.e., $\mathbf{C}_{i,j}(x,y) = 0$), Eq. (3) sets no constraint on their selection status.

Due to the binary constraints in Eq. (1), the optimization problem is an integer programming which is hard to solve. We relax the binary constraints and have the following optimization formula.

$$\min_{\{\mathbf{s}_i\}} \sum_{i=1}^{g} L_d(\mathbf{s}_i) + \sum_{i,j=1}^{g} \gamma_{i,j} L_r^{i,j}(\mathbf{s}_i, \mathbf{s}_j),\ s.t.\ \forall i,\ \mathbf{0}_i \leq \mathbf{s}_i \leq \mathbf{1}_i \quad (4)$$

where the relaxed selection vector $\mathbf{s}_i$ represents the node probabilities to be selected into the dense subgraphs. Next, we analyze the convexity results of the optimization problem in Eq. (4).

LEMMA 3.1. *If the largest eigenvalue of $\mathbf{A_i}$ satisfies $\lambda_1(\mathbf{A_i}) > \frac{\sum_j \gamma_{i,j} n_j + p(n_i - 1)}{p+1}$, the optimization problem in Eq. (4) is not convex.*

PROOF. We first explore the Hessian matrices of Eq. (4) w.r.t. to the selection vector $\mathbf{s}_i$ as $\frac{\partial^2 L}{\partial \mathbf{s}_i^2} = \frac{\partial^2 L_d}{\partial \mathbf{s}_i^2} + \sum_j \gamma_{i,j} \frac{\partial^2 L_r^{i,j}}{\partial \mathbf{s}_i^2}$, where

$$\frac{\partial^2 L_d}{\partial \mathbf{s}_i^2} = 2p(\mathbf{1}_i\mathbf{1}_i' - \mathbf{I}) - 2(p+1)\mathbf{A}_i$$

$$\frac{\partial^2 L_r^{i,j}}{\partial \mathbf{s}_i^2} = \text{diag}(\mathbf{C}_{i,j}(8(\mathbf{s}_j \odot \mathbf{s}_j) + 2\mathbf{1}_j - 8\mathbf{s}_j)) \quad (5)$$

where $\text{diag}(\mathbf{v})$ denotes a diagonal matrix corresponding to the vector $\mathbf{v}$. Clearly, both $\mathbf{1}_i\mathbf{1}_i' - \mathbf{I}$ and $-\mathbf{A}_i$ are symmetric but not positive

semi-definite, so they have negative eigenvalues. Then we consider the Hessian matrix in Eq. (5). Since $\mathbf{C}_{i,j}$ is an non-negative matrix, we study the elements of $8(\mathbf{s}_j \odot \mathbf{s}_j) + 2\mathbf{1}_j - 8\mathbf{s}_j$:

$$(8(\mathbf{s}_j \odot \mathbf{s}_j) + 2\mathbf{1}_j - 8\mathbf{s}_j)(x) = 8(\mathbf{s}_j(x) - 0.5)^2 \tag{6}$$

Due to the $[0, 1]$ constraint on entries of selection vectors, the entries of $(8(\mathbf{s}_j \odot \mathbf{s}_j) + 2\mathbf{1}_j - 8\mathbf{s}_j)$ lie in the range $[0, 2]$, so the eigenvalues of Eq. (5) are all non-negative.

According to Weyl's inequality theorem [24], for matrices $\mathbf{M}$, $\mathbf{H}$, and $\mathbf{P} \in \mathcal{H}$, where $\mathcal{H}$ is the set of $n \times n$ Hermitian matrices, if $\mathbf{M} = \mathbf{H} + \mathbf{P}$ and their eigenvalues are arranged in the order of $\lambda_1(\mathbf{M}) \geq \cdots \geq \lambda_n(\mathbf{M})$, $\lambda_1(\mathbf{H}) \geq \cdots \geq \lambda_n(\mathbf{H})$, and $\lambda_1(\mathbf{P}) \geq \cdots \geq \lambda_n(\mathbf{P})$, then we have $\lambda_n(\mathbf{P}) \leq \lambda_i(\mathbf{M}) - \lambda_i(\mathbf{H}) \leq \lambda_1(\mathbf{P})$, $\forall i = 1, \ldots, n$.

By defining $\mathbf{M} = \frac{\partial^2 L}{\partial \mathbf{s}_i^2}$, $\mathbf{H}_1 = 2p(\mathbf{1}_i\mathbf{1}_i' - \mathbf{I})$, $\mathbf{H}_2 = -2(p+1)\mathbf{A}_i$, and $\mathbf{P}_j = \text{diag}(\mathbf{C}_{i,j}(8(\mathbf{s}_j \odot \mathbf{s}_j) + 2\mathbf{1}_j - 8\mathbf{s}_j))$, since $\mathbf{H}_1$, $\mathbf{H}_2$, $\mathbf{M} \in \mathcal{H}$ and $\{\mathbf{P}_j\} \subset \mathcal{H}$, we have the following inequalities.

$$\lambda_{n_i}(\mathbf{M}) \leq \lambda_1(\mathbf{H}_1) + \lambda_{n_i}(\mathbf{H}_2) + \sum_j \gamma_{i,j}\lambda_1(\mathbf{P}_j) \tag{7}$$

where n_i is the number of nodes at the i-th layer. Based on Eq. (6), we estimate the eigenvalues of $\mathbf{P}_j$ as $0 \leq \lambda_{n_i}(\mathbf{P}_j) \leq \lambda_1(\mathbf{P}_j) \leq 2n_j$. Based on the characteristic equation $|\mathbf{1}_i\mathbf{1}_i' - \mathbf{I} - \lambda\mathbf{I}| = 0$, we have $\lambda_1(\mathbf{H}_1) = 2p(n_i - 1)$ and $\lambda_2(\mathbf{H}_1) = \cdots = \lambda_{n_i}(\mathbf{H}_1) = -2p$. By relaxing Eq. (7) we have $\lambda_{n_i}(\mathbf{M}) \leq 2p(n_i - 1) - 2(p+1)\lambda_1(\mathbf{A}_i) + \sum_j 2\gamma_{i,j}n_j$: Hence, if $\lambda_1(\mathbf{A_i}) > \frac{\sum_j \gamma_{i,j}n_j + p(n_i-1)}{p+1}$, the problem is not convex. □

Lemma 3.2. *The optimization problem in Eq.* (4) *is convex w.r.t.* $\mathbf{s}_i$ *if* $\sum_j \gamma_{i,j} \min(\mathbf{C}_{i,j}(8(\mathbf{s}_j \odot \mathbf{s}_j) + 2\mathbf{1}_j - 8\mathbf{s}_j)) \geq 2p + 2(p+1)\lambda_1(\mathbf{A}_i)$ *where* $\min(\cdot)$ *returns the minimum entry of the vector.*

Proof. Omitted for space. □

Lemma 3.1 reveals that the dense subgraph detection problem on multi-layered networks is inherently difficult and non-convex in certain regions of the parameter space. The left hand side of the convex condition in Lemma 3.2 is composed of three sub-conditions: (1) the connections between the i-th layer and other layers are close (i.e. dense $\{\mathbf{C}_{i,j}\}$ matrices); (2) the model pays enough penalty for the cross-layer objective functions (i.e. large $\{\gamma_{i,j}\}$); and (3) the selection status of the other layers (layers except the i-th layer) is distinctive (i.e., entries of $\mathbf{s}_j$ is close to either 0 or 1). Notice that $(8(\mathbf{s}_j \odot \mathbf{s}_j) + 2\mathbf{1}_j - 8\mathbf{s}_j)(x) = 8(\mathbf{s}_j(x) - 0.5)^2$. Therefore, the closer $\mathbf{s}_j(x)$ is to 0 or 1, the larger this term $8(\mathbf{s}_j(x) - 0.5)^2$ will be, and the optimization problem in Eq. (4) is more likely to be convex.

Optimization Algorithm. We first define $\mathbf{A}$ as a block diagonal matrix $\mathbf{A} = \text{diag}(\mathbf{A}_1, \ldots, \mathbf{A}_g)$. Then we define $\mathbf{C}$ as a block matrix with block $\mathbf{C}(i, j) = \mathbf{C}_{i,j}$ if $i \neq j$ and $\mathbf{C}(i, i)$ is a zero matrix for $i, j = 1, \cdots, g$. We include hyper-parameters $\gamma_{i,j}$ into a block matrix $\mathbf{R}$ whose block $\mathbf{R}(i, j) = \mathbf{R}_{i,j}$ for $i \neq j$ and $\mathbf{R}(i, i)$ is a zero matrix. Here, all entries of $\mathbf{R}_{i,j}$ have the values of $\sqrt{\gamma_{i,j}}$. We empirically set $\gamma_{i,j} = (\frac{\text{density of the i-th layer}}{\text{density of the j-th layer}})^2$. In order to punish the missing links in the selected subgraph within each layer separately, we define matrix $\hat{\mathbf{A}}$ whose diagonal blocks are the same as $\mathbf{A}$ but its off-diagonal blocks are all set as 1. In addition, we define an aggregated selection vector $\mathbf{s} = [\mathbf{s}_1', \ldots, \mathbf{s}_g']'$. With the above notations, we overload the functions L_d and L_r and rewrite Eq. (4) as follows.

$$\min_{\mathbf{s}} \quad L_d(\mathbf{s}) + L_r(\mathbf{s}) = p\mathbf{s}'(\mathbf{1}\mathbf{1}' - \mathbf{I} - \hat{\mathbf{A}})\mathbf{s} - \mathbf{s}'\mathbf{A}\mathbf{s} \tag{8}$$
$$+ ||\mathbf{R} \odot \mathbf{C} \odot (\mathbf{C} - \mathbf{s}\mathbf{s}' - (\mathbf{1} - \mathbf{s})(\mathbf{1} - \mathbf{s})')||_F^2 \quad s.t. \quad \mathbf{0} \leq \mathbf{s} \leq \mathbf{1}$$

Data	Layer	OQC	NMF	SNMF	FRAUDAR	Destine
ER	δ=0.1	0.54	**1.00**	**1.00**	0.03	**1.00**
	δ=0.2	0.03	**1.00**	**1.00**	0.03	**1.00**
	δ=0.6	0.03	0.05	0.06	0.03	**0.85**
SF	l=20	0.41	**1.00**	**1.00**	0.04	**1.00**
	l=40	0.29	0.90	0.87	0.03	**1.00**
	l=60	0.25	0.85	0.85	0.03	**1.00**

Table 1: F1 scores on the synthetic data.

Data	INFRA-5 [8]	INFRA-3 [8]	Aminer [20]	Bio [9, 17, 22]
# of layers	5	3	3	3
# of nodes	349	15,126	125,344	35,631
# of links	379	29,861	214,181	253,827
# of cross-layer links	565	28,023,500	188,844	75,456

Table 2: Statistics of the real-world datasets.

The gradient of Eq. (8) w.r.t. $\mathbf{s}$ is $\frac{\partial L}{\partial \mathbf{s}} = \frac{\partial L_d}{\partial \mathbf{s}} + \frac{\partial L_r}{\partial \mathbf{s}}$. By defining the notation $\mathbf{R}^2 = \mathbf{R} \odot \mathbf{R}$, we have:

$$\frac{\partial L_d}{\partial \mathbf{s}} = 2p(\mathbf{1}\mathbf{1}' - \mathbf{I} - \hat{\mathbf{A}})\mathbf{s} - 2\mathbf{A}\mathbf{s} \tag{9}$$

$$\frac{\partial L_r}{\partial \mathbf{s}} = 8\mathbf{s} \odot ((\mathbf{R}^2 \odot \mathbf{C})(\mathbf{s} \odot \mathbf{s})) + 2(\mathbf{R}^2 \odot \mathbf{C})\mathbf{s} \tag{10}$$
$$+ 2\mathbf{s} \odot ((\mathbf{R}^2 \odot \mathbf{C})\mathbf{1}) - 4(\mathbf{R}^2 \odot \mathbf{C})(\mathbf{s} \odot \mathbf{s}) - 8\mathbf{s} \odot ((\mathbf{R}^2 \odot \mathbf{C})\mathbf{s})$$

We update $\mathbf{s}$ by projected gradient descent: $\mathbf{s} \leftarrow \Pi_{[0,1]^{|\mathbf{s}|}}[\mathbf{s} - \alpha\frac{\partial L}{\partial \mathbf{s}}]$, where α is the learning rate, $|\cdot|$ represents cardinality, and $\Pi_{\mathcal{Y}}(\mathbf{x}) := \arg\min_{\mathbf{y} \in \mathcal{Y}} ||\mathbf{y} - \mathbf{x}||_2^2$. In experiments, we adopt Armijo line search [2] to adjust the learning rate. Finally, we set threshold 0.5 to return the relaxed selection vector $\mathbf{s}$ into a binary vector and recover it into a set of selection vectors of each layer $\{\mathbf{s}_i\}$ by deconcatenating the vector $\mathbf{s}$. In the following lemma, we show that the time complexity of the proposed Destine algorithm is linear w.r.t. the number of links of the multi-layered network.

Lemma 3.3. *The time complexity of* Destine *is* $O(t_{max}(t_{search} + 2)(m + c))$ *where* t_{search} *is the average number of iterations for searching Armijo condition;* t_{max} *is the maximal number of iterations;* m, c *are the total numbers of within-layer links and cross-layer links of the multi-layered networks respectively.*

Proof. Omitted for space. □

4 EXPERIMENTS

Metrics and Baseline Methods. We evaluate the proposed method in two scenarios. In the scenario where we manually inject cliques as ground-truths, we adopt F1-score as the metric. In the scenario without manually-injected cliques, we adopt the size (n), density ($m/\binom{n}{2}$), and triangle density ($l/\binom{n}{3}$) as the metrics where n, m, l indicate the number of nodes, edges and triangles respectively. We compare our method with following baseline methods: OQC [21], NMF [10], SNMF [5], FRAUDAR [14].

Evaluation on Synthetic Datasets. For the evaluation on synthetic datasets, the following testing protocol is conducted. A set of cliques are planted into each layer of a multi-layered network, and we test if the dense subgraph algorithms are able to detect parts or all of them. Here are the detailed dataset settings. (1) **Erdős-Rényi (ER) graphs.** We generate three Erdős-Rényi graphs [12] as three layers with 1800, 2400, 3000 nodes and link probabilities $\delta \in \{0.1, 0.2, 0.6\}$, respectively. We divide each of them into 60 subgraphs and match subgraphs from every layer into 60 matched subgraphs (i.e., view the i-th subgraphs from layer 1, 2, and 3 as a group of matched subgraphs). The cross-layer dependency links

Data		Density					Triangle Density					Size				
		OQC	NMF	SNMF	FRAUD.	DESTINE	OQC	NMF	SNMF	FRAUD.	DESTINE	OQC	NMF	SNMF	FRAUD.	DESTINE
INFRA-3	AP	0.60	0.92	0.98	0.48	**0.99**	0.28	0.80	0.94	0.18	**0.98**	82	44	37	102	**34**
	AS	0.51	0.76	0.86	0.55	**1.00**	0.16	0.46	0.66	0.20	**1.00**	13	7	7	12	**4**
	Power	0.54	0.80	0.87	0.50	**0.97**	0.18	0.53	0.66	0.15	**0.92**	52	23	20	56	**15**
INFRA-5	P1	0.53	**1.00**	0.67	0.24	**1.00**	0.05	**1.00**	0.00	0.01	**1.00**	6	**3**	**3**	14	**3**
	P2	0.47	0.18	**0.67**	0.08	**0.67**	**0.05**	0.00	0.00	0.00	0.00	6	11	**3**	32	**3**
	P3	0.50	1.00	0.67	0.12	**1.00**	0.00	**1.00**	0.25	0.00	**1.00**	5	**3**	4	20	**3**
	P4	0.67	0.20	0.00	0.19	**1.00**	0.00	0.00	0.00	0.01	**1.00**	4	10	**1**	16	3
	Net	0.50	0.25	0.33	0.23	**0.67**	0.00	0.00	0.00	0.00	0.00	4	8	**3**	12	4
Aminer	Paper	0.71	0.61	0.60	0.71	**0.90**	0.35	0.23	0.25	0.35	**0.70**	10	8	6	10	**5**
	Venue	0.53	**0.94**	**0.94**	0.40	**0.94**	0.18	0.73	**0.82**	0.10	**0.82**	48	20	**18**	65	**18**
	Auth	**1.00**	**1.00**	0.73	**1.00**	**1.00**	**1.00**	**1.00**	0.46	**1.00**	**1.00**	28	27	40	28	**26**
Bio	Chem.	**1.00**	**1.00**	**1.00**	0.48	**1.00**	**1.00**	**1.00**	**1.00**	0.22	**1.00**	**91**	**91**	**91**	183	**91**
	Gene	0.43	0.29	0.62	0.06	**0.97**	0.07	0.05	0.27	0.00	**0.92**	73	55	12	838	**9**
	DZ	0.71	0.89	0.99	0.74	**1.00**	0.43	0.74	0.99	0.48	**1.00**	119	89	64	113	**59**

Table 3: Evaluation on real-world data with metrics: size, density, and triangle density.

generating probability is set as 0.05 if the link connects two nodes from the matched subgraphs. Otherwise, it is set as 10^{-3}. We randomly set a group of matched subgraphs into cliques of size 30, 40, and 50, respectively. (2) **Scale-free (SF) graphs.** We generate three scale-free graphs by [3] with 1800, 2400, 3000 nodes. The parameter l is set as $\{20, 40, 60\}$ which is the number of links that are preferentially attached to existing high degree nodes. We execute the same settings as Erdős-Rényi graphs to generate cross-layer links and cliques in each layer.

Table 1 shows that our method consistently achieves the best performance compared with all baselines under the above two settings. Specifically, DESTINE not only performs well (i.e., one of the best) at the layers that are easier to detect dense subgraphs (e.g. ER with $\delta = 0.1$ *or* 0.2), but also significantly outperforms other baselines at the layer with more within-layer links (e.g. $\delta = 0.6$).

Evaluation on real-world datasets. We use a variety of real-world datasets to test our method, all of which are publicly available. Table 2 summarizes the statistics of these datasets. We present the properties of the detected subgraphs from perspectives of size, density, and triangle density in Table 3. As we can observe, in most cases DESTINE detects subgraphs with *the highest density, the highest triangle density, and the smallest size*. To be specific, FRAUDAR consistently extracts subgraphs of larger sizes, but both the densities and triangle densities are much smaller than the proposed method DESTINE and other baseline methods. In contrast, our method extracts the subgraphs of the highest density measures and of small sizes comparable to the baseline methods NMF and SNMF. In addition, our method detects dense subgraphs at all layers while the baseline methods may fail at some layers. For example, on the INFRA-5 dataset, the baseline method NMF can extract dense subgraphs at layers P1 and P3, but fails at layers P2, P4 and Net. This validates that the cross-layer consistency indeed helps the dense subgraph detection on multi-layered networks.

Scalability. We record wall-clock time of our method on a three-layered synthetic Erdős-Rényi graph with 10,000 nodes at each layer. We implement it with different within-layer and cross-layer link density. The results are presented in Figure 2. We observe that if one of the within-layer and cross-layer link densities is fixed, the running time is linear w.r.t. the other one. Note that for a multi-layered network with the fixed number of nodes, the link density is proportional to the number of links. In other words, the empirical results complies with the complexity analysis in Lemma 3.3.

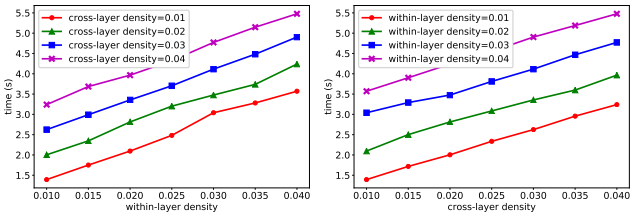

Figure 2: Wall-clock time versus the densities of within-layer and cross-layer links.

5 CONCLUSION

Dense subgraph detection is a fundamental building block behind a wealth of applications. However, most of the existing methods overlook the node dependencies across multiple networks, which could bear critical clues of detecting more comprehensive dense subgraphs. In this paper, we study the dense subgraph detection problem on multi-layered networks. The key idea is to instantiate the cross-layer consistency principle in the context of dense subgraph detection, by encoding cross-layer node dependencies as the regularization terms. We further propose an efficient algorithm named DESTINE based on projected gradient descent and conduct extensive experiments in various scenarios to validate both the effectiveness and efficiency of the proposed method.

ACKNOWLEDGEMENT

This work is supported by National Science Foundation under grant No. 1947135, and 2003924 by the NSF Program on Fairness in AI in collaboration with Amazon under award No. 1939725, by the United States Air Force and DARPA under contract number FA8750-17-C-0153 [1], and Army Research Office (W911NF2110088). The content of the information in this document does not necessarily reflect the position or the policy of the Government or Amazon, and no official endorsement should be inferred. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

[1]Distribution Statement "A" (Approved for Public Release, Distribution Unlimited)

REFERENCES

[1] James Abello, Mauricio GC Resende, and Sandra Sudarsky. 2002. Massive quasi-clique detection. In *Latin American symposium on theoretical informatics*. Springer, 598–612.

[2] Larry Armijo. 1966. Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of mathematics* 16, 1 (1966), 1–3.

[3] Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *science* 286, 5439 (1999), 509–512.

[4] Debaditya Barman, Subhayan Bhattacharya, Ritam Sarkar, and Nirmalya Chowdhury. 2019. k-Context Technique: A Method for Identifying Dense Subgraphs in a Heterogeneous Information Network. *IEEE Transactions on Computational Social Systems* 6, 6 (2019), 1190–1205.

[5] Melisew Tefera Belachew and Nicolas Gillis. 2017. Solving the Maximum Clique Problem with Symmetric Rank-One Non-negative Matrix Approximation. *Journal of Optimization Theory and Applications* 173, 1 (2017), 279–296.

[6] Mauro Brunato, Holger H Hoos, and Roberto Battiti. 2007. On effectively finding maximal quasi-cliques in graphs. In *International conference on learning and intelligent optimization*. Springer, 41–55.

[7] Moses Charikar. 2000. Greedy approximation algorithms for finding dense components in a graph. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*. Springer, 84–95.

[8] Chen Chen, Hanghang Tong, Lei Xie, Lei Ying, and Qing He. 2016. FASCINATE: fast cross-layer dependency inference on multi-layered networks. In *SIGKDD*. 765–774.

[9] Allan Peter Davis, Cynthia J Grondin, Robin J Johnson, Daniela Sciaky, Benjamin L King, Roy McMorran, Jolene Wiegers, Thomas C Wiegers, and Carolyn J Mattingly. 2017. The comparative toxicogenomics database: update 2017. *Nucleic acids research* 45, D1 (2017), D972–D978.

[10] Chris Ding, Tao Li, and Michael I Jordan. 2008. Nonnegative matrix factorization for combinatorial optimization: Spectral clustering, graph matching, and clique finding. In *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 183–192.

[11] Xiaowen Dong, Pascal Frossard, Pierre Vandergheynst, and Nikolai Nefedov. 2012. Clustering with multi-layer graphs: A spectral perspective. *IEEE Transactions on Signal Processing* 60, 11 (2012), 5820–5831.

[12] Paul Erdős and Alfréd Rényi. 1960. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci* 5, 1 (1960), 17–60.

[13] Andrew V Goldberg. 1984. *Finding a maximum density subgraph.* University of California Berkeley.

[14] Bryan Hooi, Hyun Ah Song, Alex Beutel, Neil Shah, Kijung Shin, and Christos Faloutsos. 2016. Fraudar: Bounding graph fraud in the face of camouflage. In *SIGKDD*. ACM, 895–904.

[15] Jingchao Ni, Hanghang Tong, Wei Fan, and Xiang Zhang. 2014. Inside the atoms: ranking on a network of networks. In *SIGKDD*. ACM, 1356–1365.

[16] Guo-Jun Qi, Charu C Aggarwal, and Thomas Huang. 2012. Community detection with edge content in social media networks. In *2012 IEEE 28th International Conference on Data Engineering*. IEEE, 534–545.

[17] Sabry Razick, George Magklaras, and Ian M Donaldson. 2008. iRefIndex: a consolidated protein interaction database with provenance. *BMC bioinformatics* 9, 1 (2008), 405.

[18] Yiye Ruan, David Fuhry, and Srinivasan Parthasarathy. 2013. Efficient community detection in large networks using content and links. In *Proceedings of the 22nd international conference on World Wide Web*. ACM, 1089–1098.

[19] Yizhou Sun and Jiawei Han. 2012. Mining heterogeneous information networks: principles and methodologies. *Synthesis Lectures on Data Mining and Knowledge Discovery* 3, 2 (2012), 1–159.

[20] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. Arnetminer: extraction and mining of academic social networks. In *SIGKDD*. ACM, 990–998.

[21] Charalampos Tsourakakis, Francesco Bonchi, Aristides Gionis, Francesco Gullo, and Maria Tsiarli. 2013. Denser than the densest subgraph: extracting optimal quasi-cliques with quality guarantees. In *SIGKDD*. ACM, 104–112.

[22] Marc A Van Driel, Jorn Bruggeman, Gert Vriend, Han G Brunner, and Jack AM Leunissen. 2006. A text-mining analysis of the human phenome. *European journal of human genetics* 14, 5 (2006), 535–542.

[23] Ruby W Wang, Shelia X Wei, and Y Ye Fred. 2021. Extracting a core structure from heterogeneous information network using h-subnet and meta-path strength. *Journal of Informetrics* 15, 3 (2021), 101173.

[24] Hermann Weyl. 1912. Das asymptotische Verteilungsgesetz der Eigenwerte linearer partieller Differentialgleichungen (mit einer Anwendung auf die Theorie der Hohlraumstrahlung). *Math. Ann.* 71, 4 (1912), 441–479.

[25] Min Wu, Xiaoli Li, Chee-Keong Kwoh, and See-Kiong Ng. 2009. A core-attachment based method to detect protein complexes in PPI networks. *BMC bioinformatics* 10, 1 (2009), 169.

[26] Zhe Xu, Si Zhang, Yinglong Xia, Liang Xiong, and Hanghang Tong. 2020. Ranking on Network of Heterogeneous Information Networks. In *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 848–857.

[27] Si Zhang, Dawei Zhou, Mehmet Yigit Yildirim, Scott Alcorn, Jingrui He, Hasan Davulcu, and Hanghang Tong. 2017. Hidden: hierarchical dense subgraph detection with application to financial fraud detection. In *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM, 570–578.

[28] Yang Zhou, Hong Cheng, and Jeffrey Xu Yu. 2009. Graph clustering based on structural/attribute similarities. *Proceedings of the VLDB Endowment* 2, 1 (2009), 718–729.