5-10-2019

Connected-Dense-Connected Subgraphs in Triple Networks

Dhara Shah

Recommended Citation

Shah, Dhara, "Connected-Dense-Connected Subgraphs in Triple Networks." Thesis, Georgia State University, 2019.
doi: https://doi.org/10.57709/14354991

CONNECTED-DENSE-CONNECTED SUBGRAPHS IN TRIPLE NETWORKS

by

DHARA SHAH

Under the Direction of Sushil Prasad, PhD, Yubao Wu, PhD

ABSTRACT

Finding meaningful communities - subnetworks of interest within a large scale network - is a problem with a variety of applications. Most existing work towards community detection focuses on a single network. However, many real-life applications naturally yield what we refer to as Triple Networks. Triple Networks are comprised of two networks, and the network of bipartite connections between their nodes. In this paper, we formulate and investigate the problem of finding Connected-Dense-Connected subgraph (CDC), a subnetwork which has the largest density in the bipartite network and whose sets of end points within each network induce connected subnetworks. These patterns represent communities based on the bipartite association between the networks. To our knowledge, such patterns cannot be detected by existing algorithms for a single network or heterogeneous networks. We show that finding CDC subgraphs is NP-hard and develop novel heuristics to obtain feasible solutions, the fastest of which is O(nlogn+m) with n nodes and m edges. We also study different variations of the CDC subgraphs. We perform experiments on a variety of real and synthetic Triple Networks to evaluate the effectiveness and efficiency of the developed methods. Employing these heuristics, we demonstrate how to identify communities of similar opinions and research interests, and factors influencing communities.

INDEX WORDS: Triple Networks, Unsupervised community detection , max-flow densest bipartite subgraph , NP-Hard, greedy node deletions , local search

CONNECTED-DENSE-CONNECTED SUBGRAPHS IN TRIPLE NETWORKS

by

DHARA SHAH

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Masters of Science

in the College of Arts and Sciences

Georgia State University

2019

CONNECTED-DENSE-CONNECTED SUBGRAPHS IN TRIPLE NETWORKS

by

DHARA SHAH

Committee Chair:	Sushil Prasad
	Yubao Wu
Committee:	Shihao Ji

Electronic Version Approved:

Office of Graduate Studies
College of Arts and Sciences
Georgia State University
May 2019

DEDICATION

Mom, Dad, Gallu and Elsha for being my shelter on the stormiest days

ACKNOWLEDGEMENTS

- Thank you Dr. Sushil Prasad for being my adviser
- Dr. Yubao Wu for structuring this work
- Dr. Shihao Ji for seeing me through the ML applications of this work
- Dr. Robert Harrison for encouragement, motivation and pointing me towards new developments
- Dr. Michael Stewart for mathematical proof reads
- Danial Aghajarian and Chris Freas for being my first reviewers of all new developments
- Dr. Saeid Belkasim and Dr. K.N. King for advising me towards the logistics
- Tammie Dudley and Jamie Hayes for keeping my progress on trek
- Brendan Benshoof and Andrew Rosen for being my first points of contact for any and all CS epistemology and metadata
- Jo Benshoof and Erin Gillman for endless English corrections

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

- CDC - Connected-Dense-Connected
- OCD - One Connected Dense

PART 1

INTRODUCTION

Community detection is a key primitive with a wide range of applications in real world [1]. Most existing work focuses on finding communities within a single network. In many real-life applications, we can often observe Triple Networks consisting of two networks and a third bipartite network representing the interaction between them. For example, in Twitter, users form a follower network, hashtags form a co-occurrence network, and the user-hashtag interactions form a bipartite network. The user-hashtag interactions represent a user's posts or tweets containing a hashtag. Figure 5.2 shows a real Twitter Triple Network. The nodes on the left part represent users and those on the right represent hashtags. The edges among the nodes on the left represent a user following other user. The edges among the nodes on the right represent two hashtags appearing in the same tweet. The edges in between represent a user interacting with tweets containing a hashtag. This Triple Network model can ideally represent many real world applications such as taxi pick-up-drop-off networks, Flixster user-movie networks, and author-paper citation networks.

In this paper, we study the problem of finding the Connected-Dense-Connected sub-

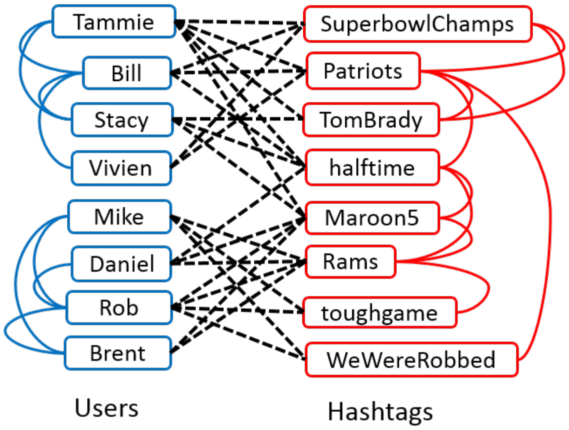


Figure (1.1) Twitter Triple Network

graphs (CDC) in Triple Networks. Given a Triple Network consisting of two graphs $G_a(V_a, E_a)$ and $G_b(V_b, E_b)$ and a bipartite graph $G_c(V_a, V_b, E_c)$, the CDC consists of two subsets of nodes $S \subset V_a$ and $T \subset V_b$ such that the induced subgraphs $G_a[S]$ and $G_b[T]$ are both connected and the density of $G_c[S, T]$ is maximized.

In the Twitter Triple Network in Figure 5.2, we observe two CDC subgraphs: the one at the top with $S_1 = \{$Tammie, Bill, Stacy, Vivien$\}$ and $T_1 = \{$Patriots, TomBrady, SuperbowlChamps, halftime$\}$, and the one at the bottom with $S_2 = \{$Mike, Daniel, Rob, Brent$\}$ and $T_2 = \{$Rams, WeWereRobbed, toughgame, Maroon5$\}$. In either of the two CDCs, the left and right networks are connected and the middle one is dense. These CDCs are meaningful. The CDC at the top shows that Patriots' fans are praising Tom Brady and are happy to be champions again. The CDC at the bottom shows that LA Rams' fans are disappointed to loose the game.

Our problem is different from finding co-dense subgraphs [2323] or coherent dense subgraphs [4545], whose goal is to find the dense subgraphs preserved across multiple networks with the same types of nodes and edges. In our problem, the left and right networks contain different types of nodes and the edges in the three networks represent different meanings. Our problem is also different than the densest connected subgraphs in dual networks [6]. Dual networks consist of one set of nodes and two sets of edges. Triple Networks consist of two sets of nodes and three sets of edges. Triple Networks can degenerate to dual networks when the two sets of nodes are identical and the bipartite links connect each node to its replica.

Previous work shows that finding the densest subgraph in a single network could be solved in polynomial time [7] and finding the densest connected subgraph in dual networks is NP-hard [6]. We show that finding CDC subgraph in Triple Networks is also NP-hard. We develop two heuristic approaches to find approximate solutions. The first approach finds CDC subgraphs of the densest bipartite subgraph. The second approach starts from large degree nodes and utilizes a local search heuristic to find CDC subgraphs. We further study variant problems with different connectivity and seed constraints, and also develop heuristics for the variant problems. We perform extensive empirical study using a variety of real and synthetic Triple Networks to demonstrate the effectiveness and efficiency of the developed algorithms.

The rest of the paper is organized as follows. Section 2 places our work among related work and contrasts with them. Section 3 defines CDC subgraphs and its variants, proving that finding these patterns is NP-Hard. Section 4 discusses heuristics to obtain these patterns. Section 5 illustrates effectiveness and of CDC subgraphs and variants, and efficiency of the heuristic methods deployed on real and synthetic networks. Section 6 concludes this work.

PART 2

BACKGROUND AND RELATED WORK

The problem of finding a densest subgraph of a graph has been well studied by data mining community. At the core, this problem asks for finding subgraphs with the highest average degree. This problem has been solved in polynomial time using max-flow min-cut approach [7]. Inspired by this approach, the problem of finding densest subgraph in a directed graph has also been solved in polynomial time [8]. The prohibitive cost of these polynomial time algorithms has been addressed with 2-approximation algorithm [9]. However, variations of densest subgraph problems, such as discovery of densest subgraph with k nodes, have been shown to be NP-hard [10]. On the other hand, the problem of finding densest subgraph with pre-selected seed nodes is solvable in polynomial time [11].

The solutions above are designed for homogeneous information network structure where the nodes and edges have just one type. Heterogeneous information networks [12] – the networks with multiple node and edge types – have been a new development in the field of data mining. Heterogeneous network structure provides a model for graph infusion with rich semantics. The Triple Networks introduced in this paper are a type of heterogeneous network with node types V_a and V_b, and edge types E_a, E_b and E_c. Our work can be categorized as unsupervised clustering in heterogeneous network. Parallel to our work, Boden et al. discuss a density based clustering approach of k-partite graphs in heterogeneous information structure [13]. In this work, two types of nodes V_a and V_b are considered. With node type specific hyper-parameters and the bipartite connections E_c, the connections E_a and E_b are inferred. This method of clustering is different from our work where E_a and E_b are part of the network, and the definition of density is hyper-parameter free. Boden et al. detect communities by subspace clustering on nodes' projection to attribute space. In contrast, our work of finding CDC subgraphs cannot be inferred as a subspace clustering technique.

Though both works produce iterative refinement algorithms, the former concentrates on improving inference of E_a and E_b iteratively.

The closest network schema to our work is dual networks [6], discovered by Wu et al. A dual network is comprised of two networks having the same set of nodes but different types of edges. These two networks are inferred as physical and conceptual networks. Wu et al. provide 2-approximation algorithms for NP-hard problem of finding subgraphs that are densest in conceptual network, and are connected in physical network. Though the network architecture and subgraph patterns are different, our work is inspired by the pruning methods and variants proposed in this work.

PART 3

TRIPLE NETWORK, CDC SUBGRAPHS AND VARIANTS

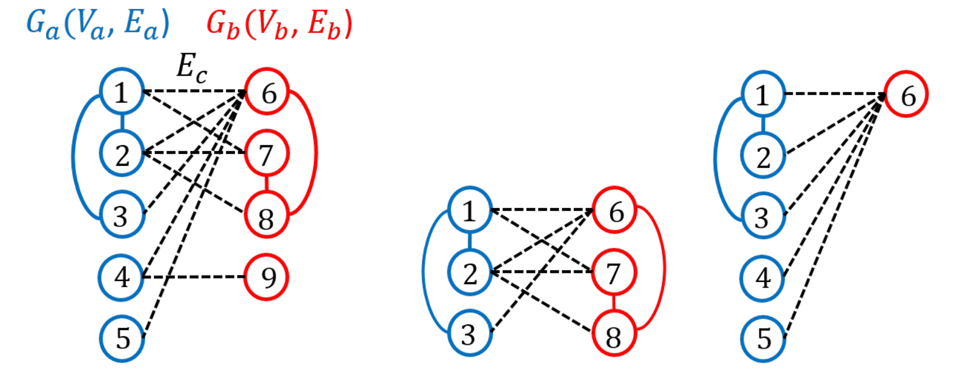
In this section we define Triple Network, CDC subgraph and its variants. We prove that finding CDC subgraph and variants from a Triple Network is NP-hard.

Definition 1 (Triple network)**.** *Let $G_a(V_a, E_a)$ and $G_b(V_b, E_b)$ represent graphs of two networks. Let $G_c(V_a, V_b, E_c)$ represent the bipartite graph between G_a and G_b. $G(V_a, V_b, E_a, E_b, E_c)$ is the Triple Network generated by G_a, G_b and G_c.*

We abbreviate a Triple Network as G. An example of Triple Network is illustrated in figure 3.1(a).

The subgraphs induced by $S_a \subset V_a$ and $S_b \subset V_b$ in networks G_a, G_b and G_c are denoted by $G_a[S_a]$, $G_b[S_b]$ and $G_c[S_a, S_b]$. For brevity, we denote this sub Triple Network, a set of three subgraphs, as $G[S_a, S_b]$.

Definition 2 (Density of a Triple Network)**.** *Given a Triple Network $G[S_a, S_b]$, its density is defined as $\rho(S_a, S_b) = \frac{|E_c(S_a,S_b)|}{\sqrt{|S_a||S_b|}}$, where $|E_c[S_a, S_b]|$ is the number of bipartite edges in*



(a) An example of a toy Triple Network

(b) CDC subgraphs of the toy Triple Network

(c) OCD subgraph of the toy Triple Network

Figure (3.1) Toy Triple Network and its CDC and OCD subgraphs *subgraph $G_c[S_a, S_b]$, $|S_a|$ is the number of nodes in $G_a[S_a]$ and $|S_b|$ is the number of nodes in $G_b[S_b]$.*

For example, the density of sub Triple Network in figure 3.1(b) with $S_a = \{1, 2, 3\}$ and $S_b = \{6, 7, 8\}]$ is $\rho(S_a, S_b) = \frac{|E_c(S_a,S_b)|}{\sqrt{|S_a||S_b|}} = \frac{6}{\sqrt{3*3}} = 2$.

By definition of density, only the bipartite edges of a Triple Network contribute to the density. Hence, the density of a Triple Network G is same as the density of its bipartite subgraph G_c.

3.0.1 Connected-Dense-Connected (CDC) subgraphs

Definition 3 (CDC subgraph)**.** *Given Triple Network $G(V_a, V_b, E_a, E_b, E_c)$, a CDC subgraph is a sub Triple Network $G[S_a, S_b]$ such that*

1. *$G_a[S_a]$ and $G_b[S_b]$ are connected subgraphs, and*
2. *the density $\rho(S_a, S_b)$ is maximized.*

For example, the density of each CDC subgraph in figure 3.1(b) is 2, higher than density of any other sub Triple Network of the Triple Network 3.1(a) that is connected in G_a and G_b. A Triple Network can have multiple CDC subgraphs.

Theorem 1. *Finding CDC subgraph in a triple network is NP Hard.*

Proof. We prove that finding CDC subgraph is a reduction of set-cover problem. Let $R = \{r_1, \cdots r_p\}$ be a set and and $C = \{C_1, \cdots C_q\}$ be its cover with $R = \cup_{i=1}^{q} C_i$. The aim of this set cover problem is to find minimum subset $C_{opt} \subset C$, known as optimal set-cover, such that each $r_j \in R$ belongs to at least one set of C_{opt}. This problem is proved to be NP complete.

Let $T = \{t_1, \cdots t_p\}$ be a set of points, having the same cardinality as R. Let $D = \{D_1, \cdots D_q\}$ be a set-cover of T, analogous to C, such that if $r_i \in C_j$, then $t_i \in D_j$. Hence, T, D can be considered as a copy of R, C.

We construct the triple network as follows. Let $V_a = \{h, r_1, \cdots r_p, C_1, \cdots C_q\}$, where node h is connected to every $C_i \in C$ and node r_i is connected to node C_j if $r_i \in C_j$ in the

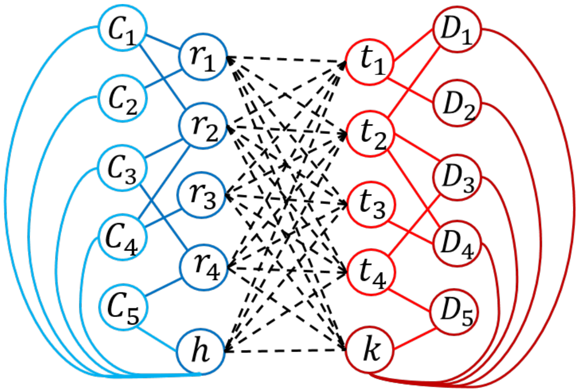


Figure (3.2) Triple Network from set-cover

set-cover problem. Similarly, let $V_b = \{k, t_1 \cdots t_p, D_1, \cdots D_q\}$ be the analogous set to V_a. We connect V_a and V_b by connecting all nodes $\{r_1, \cdots .r_p, h\}$ to all nodes $\{t_1, \cdots, t_p, k\}$.

Construction of such triple network is illustrated in figure 3.2 from an instance of set-cover problem $C_1 = \{r_1, r_2\}, C_2 = \{r_1\}, C_3 = \{r_2, r_4\}, C_4 = \{r_2, r_3\}, C_5 = \{r_4\}$.

Let $C_{opt} \subset C$ be an optimal solution to the set-cover problem of C and $|C_{opt}| = q^* \leq q$. Similarly, let D_{opt} be the analogous optimal solution to D and $|D_{opt}| = q^* \leq q$. Let $H = \{h, r_1, \cdots, r_p\}$ and $J = \{k, t_1, \cdots, t_p\}$. The subgraph induced by $S_a = H \cup C_{opt}$ is connected in V_a, and similarly, the subgraph induced by $S_b = J \cup D_{opt}$ is connected in V_b. Hence, the subgraph $G[S_a, S_b]$ has density $\rho(S_a, S_b) = \frac{(p+1)^2}{(p+q^*+1)}$.

Let S_1 and S_2 be any nonempty node sets where $G_a[S_1]$ and $G_b[S_2]$ are connected. In general, $S_1 = H' \cup C'$ where $H' \subset H$ and $C' \subset C$. Similarly, $S_2 = J' \cup D'$ where $J' \subset J$ and $D' \subset D$. We show that $\rho(S_1, S_2) \leq \rho(S_a, S_b)$, making $G[S_a, S_b]$ the CDC subgraph. Let $|H'| = p_1$, $|C'| = q_1$, $|J'| = p_2$ and $|D'| = q_2$. Hence, $\rho(S_1, S_2) = \frac{p_1 p_2}{\sqrt{(p_1+q_1)(p_2+q_2)}}$.

First, we consider the case when S_1 contains all the nodes of H and S_2 contains all the nodes of J. In this case, $p_1 = p_2 = p + 1$. Also, by definition of optimal set-cover, $q^* \leq q_1$ and $q^* \leq q_2$. Hence, $\rho(S_1, S_2) = \frac{(p+1)^2}{\sqrt{(p+q_1+1)(p+q_2+1)}} \leq \frac{(p+1)^2}{(p+q^*+1)} = \rho(S_a, S_b)$.

Second, we consider the case when S_1 contains a subset of nodes $H' \subset H$. In this case, we first show that adding elements from $H \setminus H'$ to S_1 will only increase its density.

If $h \notin S_1$, then after adding h to S_1, the resulting subgraph has density $\frac{(p_1+1)p_2}{\sqrt{(p_1+q_1+1)(p_2+q_2)}} > \frac{p_1 p_2}{\sqrt{(p_1+q_1)(p_2+q_2)}} = \rho(S_1, S_2)$. This subgraph is also connected in G_a, since h is connected to every $C_i \in C$. To add a node $r_j \in H \setminus H'$ and making it still connected, we need to add at most one node C_i to C' with $r_j \in C_i$. Hence, the density of this resulting subgraph is $\frac{(p_1+1)p_2}{\sqrt{(p_1+q_1+2)(p_2+q_2)}} > \frac{p_1 p_2}{\sqrt{(p_1+q_1)(p_2+q_2)}} = \rho(S_1, S_2)$. We can repeat this process by adding remaining nodes of $H \setminus H'$ to S_1, while density of the resulting subgraphs keeps increasing.

Similarly, adding elements from $J \setminus J'$ to S_2 increases density of the resulting subgraphs. Since we proved in the first case that the density $\rho(S_1, S_2)$ when $H \subset S_1$ and $J \subset S_2$, we have hence completed the proof of the second case.

In summary, we proved that for any nonempty sets $S_1 \subset V_a$ and $S_2 \subset V_b$, $\rho(S_1, S_2) \leq \rho(S_a, S_b)$, making $G[S_a, S_b]$ a CDC subgraph. Also,$G[S_a, S_b]$ is the solution inducted by optimal set covers, an instance being $S_a = \{r_1, r_2, r_3, r_4, h,\ C_1, C_3, C_4\}$ and $S_b = \{s_1, s_2, s_3, s_4, k, D_1, D_3, D_4\}$ hence proving that finding CDC subgraphs is NP hard. □

3.0.2 Variants of CDC subgraph

CDC subgraphs stipulate connectedness of $G_a(S_a)$ and $G_b(S_b)$. Alleviating this connectivity constraint, we define OCD subgraphs for which exactly one of $G_a(S_a)$ or $G_b(S_b)$ is connected.

Definition 4 (OCD subgraph). *Given a Triple Network $G(V_a, V_b, E_a, E_b, E_c)$ a OCD subgraph is a sub Triple Network $G[S_a, S_b]$ such that*

1. *Exactly one of $G_a[S_a]$ or $G_b[S_b]$ is connected, and*

2. *The density $\rho(S_a, S_b)$ is maximized.*

For example, the sub Triple Network $G[\{1, 2, 3, 4, 5\}, \{6\}]$ with the highest density 2.23 in figure 3.1(c) is an OCD subgraph as $G_a[\{5\}]$ is connected. A Triple Network can have multiple OCD subgraphs.

Finding OCD subgraph in triple network is NP hard

Proof. We prove that finding OCD subgraph is also reduction of the set cover problem. We first construct the triple network same as in theorem 1. Let $S_a = H$ and $S_b = J \cup D_{opt}$. The subgraph $G[s_a, S_b]$ hence has density $\rho(S_a, S_b) = \frac{(p+1)^2}{\sqrt{(p+1)(p+q^*+1)}}$ We claim that $G[S_a, S_b]$ is an OCD subgraph. We observe that $G[S_b]$ is connected.

Let S_1 and S_2 be any nonempty node sets where either $G[S_1]$ or $G[S_2]$ is connected. In general, $S_1 = H' \cup C'$ where $H' \subset H$. Similarly, $S_2 = J' \cup D'$ where $J' \cup J$. We show that $\rho(S_1, S_2) \leq \rho(S_a, s_b)$.

First, we consider the case when S_1 contains all the nodes of H and S_2 contains all the nodes of J. In this case, $p_1 = p_2 = p + 1$. Also, by definition of optimal set-cover, $q^* \leq q_1$ and $q^* \leq q_2$. Hence, $\rho(S_1, S_2) = \frac{(p+1)^2}{\sqrt{(p+q_1+1)(p+q_2+1)}} \leq \frac{(p+1)^2}{\sqrt{(p+q^*+1)(p+1)}} = \rho(S_a, S_b)$.

Second, we consider the case when S_1 contains a subset of nodes $H' \subset H$. In this case, we first show that adding elements from $H \setminus H'$ to S_1 will only increase its density. Suppose, $G_a[S_1]$ is not connected and $G_b[S_2]$ is connected. Then, after adding element from $H \setminus H'$, the resulting subgraph has density $\frac{(p_1+1)p_2}{\sqrt{(p_1+q_1)(p_2+q_2)}} > \frac{p_1 p_2}{(p_1+q_1)(p_2+q_2)} = \rho(S_1, S_2)$. This includes adding h to S_1 if $h \notin H'$, making resultant subgraph connected in V_a. Now suppose $G_a[S_1]$ is connected. Then, following the same case of theorem 1, we first add h if it is not in H' and then add element from $H \setminus H'$ and still show that the resultant subgraph is connected in V_a and its density increases. Similarly, we conclude that when S_2 contains a subset of nodes in $J' \subset J$, adding elements from $J' \setminus J$ also increases the density of the resultant subgraph.

At last, we observe that if $G_a[S_2]$ is connected, then the resultant subgraph obtained by removing elements from C' has density $\frac{p_1 p_2}{\sqrt{(p_1+q_1-1)(p_2+q_2)}} > \rho(S_1, S_2)$.

In summary, we have proved that for any nonempty sets $S_1 \subset V_a$ and $S_2 \subset V_b$ with either $G_a[S_1]$ or $G_b[S_2]$ connected has density $\rho(S_1, S_2) \leq \rho(S_a, S_b)$, making $G[S_a, S_b]$ an OCD subgraph. Also, $G[S_a, S_b]$ is the solution induced by optimal set cover, an instance being $S_a = \{r_1, r_2, r_3, r_4, h\}, S_b = \{s_1, s_2, s_3, s_4, k, D_1, D_3, D_4\}$ hence proving that finding OCD subgraphs is NP hard. □

3.1 Adding constraints to CDC and OCD subgraphs

We observe that CDC patterns are meaningful around pre-selected nodes in $G_a(S_a)$ or $G_b(S_b)$. We identify these pre-selected nodes as seeds. We introduce CDC and OCD subgraphs with seed constraints, where $G_a(S_a)$ or $G_b(S_b)$ should maintain their connectivity constraints while containing the seeds.

Definition 5. *(CDC_seeds).Given a Triple Network $G(V_a, V_b, E_a, E_b, E_c)$ and sets of seed nodes $V_1 \subset V_a$ and $V_2 \subset V_b$, the CDC_seeds subgraph consists of sets of nodes S_a, S_b such that $V_1 \subset S_a$, $V_2 \subset S_b$, $G_a[S_a]$ and $G_b[S_b]$ are connected and density of $G[S_a, S_b]$ is maximized.*

Definition 6. *(OCD_seed). Given a Triple Network $G(V_a, V_b, E_a, E_b, E_c)$ and a set of node V with $V \subset S$, the OCD_seed consists of sets of nodes such that either $G_a[S]$ or $G_b[S]$ is connected and the density of $G[S, E_c[S]]$ is maximized.*

Finding OCD, CDC_seeds and OCD_seed subgraphs in a Triple Network is NP-hard. Similar set-cover arguments as in Theorem 1 could be used to prove it. Please refer to [14] for details.

PART 4

HEURISTIC ALGORITHMS

Since mining CDC subgraphs is NP hard, we propose heuristic algorithms for finding feasible solutions by two approaches. In our first approach, we obtain the densest bi-partite subgraph $G_c[S_a, S_b]$ and then find the connected components of $G_a[S_a]$ and $G_b[S_b]$ using BFS. As a result, we obtain connected sub Triple Networks, with bi-partite edges in $G_c[S_a, S_b]$. We then choose the one with the highest density as a feasible CDC subgraph. Since the time complexity of obtaining the densest bi-partite subgraph is higher than that of BFS, algorithms in sections 4.1 and 4.2 focus on improving the complexity of finding the densest bi-partite subgraphs. In second approach, With given seed nodes from V_a and V_b, we build CDC subgraphs by adding nodes with highest bipartite degrees, while maintaining the connectedness in G_a and G_b. This Local Search algorithm is presented in section 4.3.

We observe that there can be multiple densest bi-partite subgraphs of a bi-partite graph, and real world Triple Networks are sparse in E_c. To use the sparsity of E_c as a leverage, we explore methods to divide the bipartite graph $G(V_a, V_b, E_c)$ in to smaller bi-partite subgraphs first and then apply the densest subgraph algorithms for some of these subgraphs. For an undirected graph, a connected densest subgraph exists. Following this intuition, we proved that the same is true for our formulation of the bi-partite graph.

Theorem 2. *Let $G(S_{a_1}, S_{b_1}, E(S_{a_1}, S_{b_1})), G(S_{a_2}, S_{b_2}, E(S_{a_2}, S_{b_2}))$ be bipartite subgraphs, with $S_{a_1} \cap S_{a_2} = \phi, S_{b_1} \cap S_{b_2} = \phi, E(S_{a_1}, S_{b_2}) = \phi, E(S_{a_2}, S_{b_1}) = \phi, E(S_{a_1}, S_{b_1}) \cap E(S_{a_2}, S_{b_2}) = \phi$. Let $|S_{a_1}| = a_1, |S_{a_2}| = a_2, |S_{b_1}| = b_1, |S_{b_2}| = b_2, |E(S_{a_1}, S_{b_1})| = e_1, |E(S_{a_2}, S_{b_2})| = e_2$.*
Let the density of this graphs defined by
$\rho(G(S_{a_1}, S_{b_1}, E(S_{a_1}, S_{b_1})) = \frac{e_1}{\sqrt{a_1 b_1}}$,
$\rho(G(S_{a_2}, S_{b_2}, E(S_{a_2}, S_{b_2})) = \frac{e_2}{\sqrt{a_2 b_2}}$,
$\rho(G(S_{a_1} \cup S_{a_2}, S_{b_1} \cup S_{b_2}, E(S_{a_1}, S_{b_1}) \cup E(S_{a_2}, S_{b_2})) = \frac{e_1+e_2}{\sqrt{(a_1+a_2)(b_1+b_2)}}$

Prove that $\frac{e_1+e_2}{\sqrt{(a_1+a_2)(b_1+b_2)}} \leq max\{\frac{e_1}{\sqrt{a_1b_1}}, \frac{e_2}{\sqrt{a_2b_2}}\}$

Proof. Without loss of generality, let $max\{\frac{e_1}{\sqrt{a_1b_1}}, \frac{e_2}{\sqrt{a_2b_2}}\} = \frac{e_1}{\sqrt{a_1b_1}}$.

This implies,

$$\frac{e_1}{\sqrt{a_1b_1}} \geq \frac{e_2}{\sqrt{a_2b_2}} \Leftrightarrow e_2 \leq e_1\frac{\sqrt{a_2b_2}}{\sqrt{a_1b_1}} \tag{4.1}$$

Now, under this assumption,

$$\frac{e_1+e_2}{\sqrt{(a_1+a_2)(b_1+b_2)}} \leq max\{\frac{e_1}{\sqrt{a_1b_1}}, \frac{e_2}{\sqrt{a_2b_2}}\} \tag{4.2}$$

$$\Leftrightarrow \frac{e_1+e_2}{\sqrt{(a_1+a_2)(b_1+b_2)}} \leq \frac{e_1}{\sqrt{a_1b_1}} \tag{4.3}$$

$$\tag{4.4}$$

Also, LHS of equation (4.2)=

$$\begin{aligned}\frac{e_1+e_2}{\sqrt{(a_1+a_2)(b_1+b_2)}} &\leq \frac{e_1+e_1\frac{\sqrt{a_2b_2}}{\sqrt{a_1b_1}}}{\sqrt{(a_1+a_2)(b_1+b_2)}} \text{ Because (4.1)}\\ &= \frac{e_1(\sqrt{a_1b_1}+\sqrt{a_2b_2})}{\sqrt{a_1b_1}\sqrt{(a_1+a_2)(b_1+b_2)}}\end{aligned}$$

Hence, if we prove

$$\frac{e_1(\sqrt{a_1b_1}+\sqrt{a_2b_2})}{\sqrt{a_1b_1}\sqrt{(a_1+a_2)(b_1+b_2)}} \leq \frac{e_1}{\sqrt{a_1b_1}} = \text{ RHS of equation (4.2)}$$

we prove (4.2). Here,

$$\begin{aligned}\frac{e_1(\sqrt{a_1b_1}+\sqrt{a_2b_2})}{\sqrt{a_1b_1}\sqrt{(a_1+a_2)(b_1+b_2)}} &\leq \frac{e_1}{\sqrt{a_1b_1}}\\ \Leftrightarrow (\sqrt{a_1b_1}+\sqrt{a_2b_2}) &\leq \sqrt{(a_1+a_2)(b_1+b_2)}\\ \Leftrightarrow (\sqrt{a_1b_1}+\sqrt{a_2b_2})^2 &\leq (a_1+a_2)(b_1+b_2)\\ \Leftrightarrow 2\sqrt{a_1b_1a_2b_2} &\leq a_1b_2+a_2b_1\\ \Leftrightarrow \sqrt{(a_1b_2)(a_2b_1)} &\leq \frac{a_1b_2+a_2b_1}{2}\end{aligned}$$

This is true since arithmetic mean of two non-negative real numbers is always greater than or equal to their geometric mean. Hence

$$\frac{e_1 + e_2}{\sqrt{(a_1 + a_2)(b_1 + b_2)}} \leq \frac{e_1(\sqrt{a_1 b_1} + \sqrt{a_2 b_2})}{\sqrt{a_1 b_1}\sqrt{(a_1 + a_2)(b_1 + b_2)}}$$
$$\leq \frac{e_1}{\sqrt{a_1 b_1}} = max\{\frac{e_1}{\sqrt{a_1 b_1}}, \frac{e_2}{\sqrt{a_2 b_2}}\}$$

□

This allows us to consider sub Triple Networks that are connected in E_c for the densest subgraph discovery, which significantly lowered the cost of our algorithms.

4.1 Maxflow Densest Subgraph (MDS)

MDS algorithm finds a densest bipartite subgraph of a Triple Network in polynomial time. Inspired by [8] and [7], we use the max-flow min-cut strategy to prove this.

Definition 7. *(Maximum density and densest subgraph in Triple Network) In a Triple Network $G(V_a, V_b, E_a, E_b, E_c)$, maximum density is $\rho^* = \max\limits_{S_a \subseteq V_a, S_b \subseteq V_b} \frac{|E_c(S_a, S_b)|}{\sqrt{|S_a||S_b|}}$.*
The subgraph $G[S_a^*, S_b^*]$ is a densest subgraph if $\rho(S_a^*, S_b^*) = \rho^*$.

Let $G_c[S_a, S_b]$ be a bi-partite subgraph of the Triple Network G. Consider the number $\lambda \in \mathbb{R}^+$ for which $|E_c(S_a, S_b)| - \lambda\sqrt{|S_a||S_b|} = 0$. λ, thus the density of this graph, depends on ratio $r = \frac{|S_a|}{|S_b|}$ and $|E_c(S_a, S_b)|$. Ratio r can take at most $|V_a||V_b|$ different values, and $\lambda \in (0, \sqrt{|V_a||V_b|}]$. It is evident from definition 7 that finding a densest subgraph of the Triple Network is equivalent to finding $\max\limits_{S_a \subset V_a, S_b \subset V_b} \{\lambda| \, |E_c(S_a, S_b)| - \lambda\sqrt{|S_a||S_b|} = 0\}$ over all subgraphs $G_a[S_a], G_b[S_b]$. Let $G(S_a^*, S_b^*, E_c^*)$ be the subgraph for which this maxima is achieved. Instead of enumerating all possible subgraphs $S_a \subset V_a$ and $S_b \subset V_b$, if we could guess λ and r. With these guessed values of λ and r, if there exists a subgraph $G[S_a, S_b]$ with $\frac{|S_a|}{|S_b|} = r$ and density greater than the current guess λ, then the densest subgraph would be the graph associated with maximum such λ. We argue that it is sufficient to guess r and λ to guess the density of the bi-partite graph as the following: By definition of λ and r,

$$\begin{aligned}
\lambda &= \frac{|E_c(S_a, S_b)|}{\sqrt{|S_a||S_b|}} \\
r &= \frac{|S_a|}{|S_b|} \Rightarrow |S_a| = r|S_b| \\
\Rightarrow \lambda &= \frac{|E_c(S_a, S_b)|}{|S_b|\sqrt{r}} \\
\Rightarrow \lambda\sqrt{r} &= \frac{|E_c(S_a, S_b)|}{|S_b|}
\end{aligned}$$

However, we assume that we only consider connected bi-partite graphs, meaning that for each $v_a \in S_a$ and $v_b\ inS_b$, we know that their bi-partite degrees $d(v_a)$ and $d(v_b)$ are nonzero. If that was not the case, then we will have dropped those elements, and have gotten better density. Also, we proved this as a theorem.

Hence, we can safely say that $|E_c(S_a, S_b)| = \sum_{v_b \in S_b} d(v_b)$, with $d(v_b) \geq 1$. This means that $\frac{|E_c(S_a,S_b)|}{|S_b|}$ represents average degree of $|S_b|$, that can be approximated as k, $k \in [1, \infty)$. Hence,

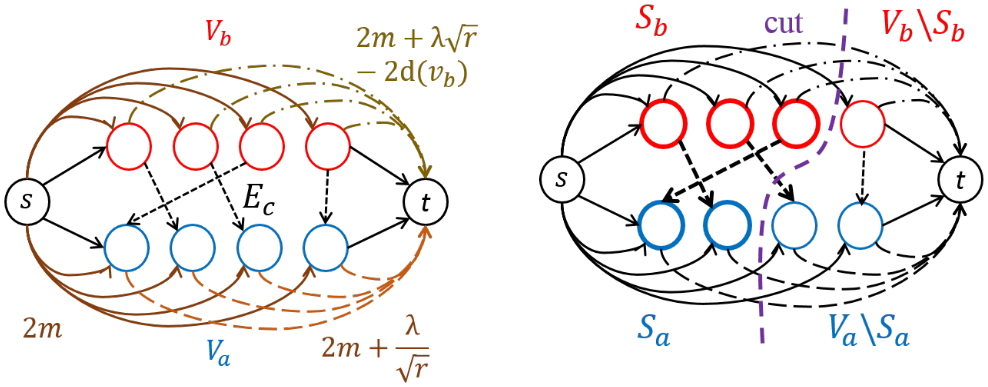
$$\lambda\sqrt{r} = \frac{k|S_b|}{|S_b|} = k$$

so, by guessing r and λ, we try to see if there is a subgraph $G[S_a, S_b]$ having average degree k in S_b

Given the values of λ and r, we construct the following flow network using the Triple Network G. This flow network yields a subgraph $G[S_a, S_b]$ of density greater than λ if such subgraph exists in G. Else it yields an empty set.

1. Initialize weighted directed graph $G'(V', E')$ with $V' = V_a \cup V_b$, $E' = \phi$, and a constant $m = |E_c|$

2. For all edges $\{v_a, v_b\} \in E_c$, add (v_b, v_a) with weight 2 to E'



(a) Construction of the flow graph for finding a densest subgraph of the Triple Network $G(V_A, V_b, E_c)$

(b) Finding the minimum cut for given ratio guess r and iteratively adjusting the bounds of maximum density renders a densest subgraph $G(S_a, S_b)$

Figure (4.1) MDS algorithm: Flow construction and iterations

3. Add source node s and sink node t to V'
4. For all vertices $v \in V_a \cup V_b$, add edge (s, v) with weight $2m$ to E'
5. For all vertices $v_a \in V_a$, add edge (v_a, t) with weight $2m + \frac{\lambda}{\sqrt{r}}$ to E'
6. For all vertices $v_b \in V_b$, add edge (v_b, t) with weight $2m + \sqrt{r}\lambda - 2d(v_b)$ to E', where $d(v_b)$ is the degree of v_b in G

Now, we apply the MDS algorithm 1 to this graph.

Theorem 3. *MDS algorithm yields a densest subgraph of the Triple Network.*

Proof. Let $G(V_a, V_b, E_c)$ be a Triple Network with $V_a \neq \phi, V_b \neq \phi$. Let $G'(V', E')$ be the weighted directed flow network constructed from this network as mentioned above. Let S, T be the minimum s-t cut of this flow network. From figure 4.1(a), as a base line, if $S = \{s\}$ and $T = V_a \cup V_b \cup \{t\}$, then the value the cut is $2m(|V_a| + |V_b|)$. However, if $S = \{s\} \cup S_a \cup S_b$

and $T = V_a \setminus \{S_a\} \cup V_b \setminus \{S_b\} \cup \{t\}$ then the value of a cut in this flow network is

$$2m|V_a| + 2m|V_b| - \sum_{v_a \in V_a \setminus S_a} 2m - \sum_{v_b \in V_b \setminus S_b} 2m + \sum_{v_a \in S_a} (2m + \frac{\lambda}{\sqrt{r}})$$

$$+ \sum_{v_b \in S_b} (2m + \sqrt{r}\lambda - 2d(v_b)) + \sum_{\substack{\{v_b, v_a\} \in E \\ , v_b \in S_b, \\ v_a \in V_a \setminus S_a}} 2$$

$$= 2m(|V_a| + |V_b|) + \lambda\sqrt{r}|S_b| + \frac{\lambda}{\sqrt{r}}|S_a| - 2|E_c(S_a, S_b)|$$

$$= 2m(|V_a| + |V_b|) - 2(|E_c(S_a, S_b)| - \lambda\sqrt{|S_a||S_b|})(\because r = \frac{|S_a|}{|S_b|})$$

This non-trivial s-t cut, if exists, is minimal. Hence the value of this cut is less than the value of trivial cut. In other words, $2m(|V_a| + |V_b|) \geq 2m(|V_a| + |V_b|) - 2(|E_c(S_a, S_b)| - \lambda\sqrt{|S_a||S_b|}$

Hence, for a non-trivial s-t cut, $|E_c(S_a, S_b)|\lambda\sqrt{|S_a||S_b|} < 0$. So if, for given values of λ and r, the flow network renders a non-trivial s-t cut S, T; then the subgraph $S \setminus \{s\} = (S_a, S_b, E(S_a, S_b))$ has density λ such that
$|E_c(S_a, S_b)| - \lambda\sqrt{|S_a||S_b|} < 0$. Which implies that the density of the subgraph $(S_a, S_b, E(S_a, S_b)) \geq \lambda$. Hence, maximum density has to be higher than the current guess of λ. However, if the flow network renders a trivial s-t cut, no subgraph of G has density λ with given r. Hence, maximum density has to be lower than current guess of λ. By repeating this process as a binary search, eventually we will find the smallest λ with $|E_c(S_a, S_b)| - \lambda\sqrt{|S_a||S_b|} = 0$ for the given r. By iterating on possible values of r, the maximum value of such λ is found. This value is maximum density and the corresponding subgraph is a densest subgraph of G. □

Theorem 4. *MDS algorithm is a polynomial time algorithm.*

Proof. The density difference of any two subgraphs of a bi-partite graph $G(V_a, V_b, E_c)$ is $\left|\frac{m}{\sqrt{v_1 v_2}} - \frac{m'}{\sqrt{v_1' v_2'}}\right| \geq \frac{1}{|V_a|^2|V_b|^2}$ with $0 \leq m, m' \leq |E_c|, 1 \leq v_1, v_1' \leq |V_a|, 1 \leq v_2, v_2' \leq |V_b|$. This guarantees that the search for maximum density in the range $(0, \sqrt{|V_a||V_b|}]$ can be performed with step size $\frac{1}{|V_a|^2|V_b|^2}$, halting in $O(|V_a|^{3/2}|V_b|^{3/2})$ iterations.

```
Input: Triple Network $G(V_a, V_b, E_c)$,with $V_a \neq \phi, V_b \neq \phi$
Output: A densest subgraph $G[S_a, S_b]$ of $G$
  $possible\_ratios = \{\frac{i}{j} | i \in [1, \cdots |V_a|], j \in [1, \cdots |V_b|]\}$
  $densest\_subgraph = \phi, maximum\_density = 0$
  for $ratio\ guess\ r \in possible\_ratios$ do
      $low \leftarrow 0, high \leftarrow \sqrt{|V_a||V_b|}, g = \phi$
      while $high - low \geq \frac{1}{|V_a|^2|V_b|^2}$ do
          $mid = \frac{high+low}{2}$
          construct a flow graph $G'$ as described in 1 - 6 and find the minimum s-t cut
            $S, T$
          $g' = S \setminus \{\text{source node } s\}$
          if $g' \neq \phi$ then
              $g \leftarrow g'$
              $low = mid$
          else $high = mid$
          if $maximum\_density < low$ then
              $maximum\_density = low$
              $densest\_subgraph = g$
```

Algorithm 1 Maxflow Densest Subgraph (MDS)

Within each iteration of this binary search, the minimum cut of the flow graph is calculated in $O(|V_a| + |V_b|)^2(2(|V_a| + |V_b|) + |E_c|))$. Hence, the complexity of algorithm 1 is $O(|V_a|^{4.5}|V_b|^{4.5})$. Adding the cost of BFS as stage II, the upper-bound still remains unchanged. □

4.2 Greedy Node Deletions

Due to high time complexity, MDS algorithm is infeasible for large Triple Networks. In this section, we present heuristics to obtain a dense bi-partite subgraph with a reduced time complexity.

The first heuristic to obtain a dense bipartite subgraph is to iteratively delete the nodes with the lowest bipartite degree while keeping track of the subgraph with the highest density obtained in the process. This algorithm of Greedy Node Deletion using degrees (GND) is formalized as Algorithm 2, where criterion in line 2 is node degree.

However, degree is not the best measure of a node's impact on density. Figure 4.2

```
Input: Triple Network $G(V_a, V_b, E_c)$,with $V_a \neq \phi, V_b \neq \phi$,
    criterion to delete nodes
Output: A densest subgraph $G[S_a, S_b]$ of $G$
  $S_a = V_a, S_b = V_b, maximim\_density = \rho(V_a, V_b)$
  while $V_a \neq \phi$ and $V_b \neq \phi$ do
      $v =$ node with minimum criterion in $V_a \cup V_b$
      $V_a = V_a \setminus \{v\}, V_b = V_b \setminus \{v\}$
      if $maximum\_density < \rho(V_a, V_b)$ then
          $S_a = V_a, S_b = V_b, E_c = E[V_a, V_b]$
  return $G[S_a, S_b]$
```

Algorithm 2 Greedy Node Deletions

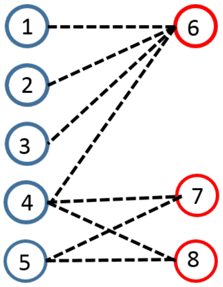


Figure (4.2) GND misses the densest subgraph by deleting the nodes $\{1, 2, 3\}$

illustrates that GND delets the nodes $\{1, 2, 3\}$ iteratively. Iteratively deleting the lowest degree neighbors of the higher degree nodes may lead to missing the densest bi-partite subgraph $[\{1, 2, 3, 4\}, \{6\}]$.

Instead of accounting for the connections of a node, the percent of the possible connections of that node may serve as a better measure of the node's impact on density. With this intuition, we define rank of a node.

Definition 8 (Rank)**.** *Let* $G(V_a, V_b, E_a, E_b, E_c)$ *be a Triple Network. For* $v_a \in V_a, rank(v_a) = \frac{d(v_a)}{|V_b|}$ *and for* $v_b \in V_b, rank(v_b) = \frac{d(v_b)}{|V_a|}$.

Using the lowest rank as the deletion criterion, we modify Algorithm 2 and formulate Greedy Rank Deletion (GRD) Algorithm 2, where the criterion of deletion in line 2 is rank.

A drawback of GRD is that the deletion of nodes is sequential and one at a time and hence slow. To expedite this process, for each iteration, we delete all the nodes satisfying the deletion criterion in bulk. This does not lower the time complexity upper-bound, but

```
Input: Triple network $G(V_a, V_b, E_c)$,with $V_a \neq \phi, V_b \neq \phi$
Output: A densest subgraph $G[S_a, S_b]$ of $G$
  $S_a = V_a, S_b = V_b, maximim\_density = \rho(V_a, V_b)$
  while $V_a \neq \phi$ and $V_b \neq \phi$ do
      $v$ = node with minimum rank in $V_a \cup V_b$
      $V_a = V_a \setminus \{v\}, V_b = V_b \setminus \{v\}$
      if $maximum\_density < \rho(V_a, V_b)$ then
          $S_a = V_a, S_b = V_b, E_c = E[V_a, V_b]$
  return $G[S_a, S_b]$
```

Algorithm 3 Greedy Node Deletion by using node ranks(GRD)

the running time decreases exponentially. Fast Rank Deletion (FRD) is hence formulated as 4. This algorithm could be tuned by choosing different ϵ values from $(-1, 1)$ with the lower to higher being less to more deletions per iteration.

```
Input: Triple Network $G(V_a, V_b, E_c)$,with $V_a \neq \phi, V_b \neq \phi$,
    value of $\epsilon \in (-1, 1)$
Output: A densest subgraph $G[S_a, S_b]$ of $G$
  $S_a = V_a, S_b = V_b, maximim\_density = \rho(V_a, V_b)$
  while $V_a \neq \phi$ and $V_b \neq \phi$ do
      $\bar{r}$ = average node rank in $G$
      $\bar{V} = \{v \in V_a \cup V_b \mid rank(v) < (1 + \epsilon)\bar{r}\}$
      $V_a = V_a \setminus \bar{V}, V_b = V_b \setminus \bar{V}$
      if $maximum\_density < \rho(V_a, V_b)$ then
          $S_a = V_a, S_b = V_b, E_c = E[V_a, V_b]$
  return $G[S_a, S_b]$
```

Algorithm 4 Fast Rank Deletion (FRD)

4.2.1 Time complexity of Greedy Node Deletions

By maintaining two {degree:node} Fibonacci heaps and an index on the nodes, the time complexity of these greedy deletion algorithms is $O((V_a + V_b)log(V_a + V_b) + E_c)$. Adding the cost of BFS for stage II, the total time complexity for obtaining CDC subgraphs is $O((V_a + V_b)log(V_a + V_b) + E_c + E_a + E_b)$

```
Input: $G(V_a, V_b, E_c)$,with $V_a \neq \phi, V_b \neq \phi$
    $seedS_a$ = Set of seeds in $V_a$
    $seedS_b$ = Set of seeds in $V_b$
Output: A subgraph $G[S_a, S_b]$ of $G$
    $S_a$ = Connected component of $seedS_a$ in $G_a$
    $S_b$ = Connected component of $seedS_b$ in $G_b$
    $\delta(S_a)$ = Boundary of $S_a$ in $G_a$
    $\delta(S_b)$ = Boundary of $S_b$ in $G_b$
    $nbhd$, the adjacency list of $V_a$ in $G_a$ and $V_b$ in $G_b$
  $max\_density = \rho(G[S_a, S_b])$
  do
      $v$ = node in $\delta(S_a) \cup \delta(S_b)$ with the highest degree in $G_c[S_a, S_b]$
      $S_a = S_a \cup v$ if $v \in V_a$, $S_b = S_b \cup v$ if $v \in V_b$
      $\delta(S_a) \cup \delta(S_b) = \delta(S_a) \cup \delta(S_b) \cup nbhd(v) \setminus \{v\}$
      $max\_density = max(max\_density, \rho(G[S_a, S_b]))$
  while $\rho(G_c[S_a, S_b]) \geq max\_density$ and $\delta(S_a) \cup \delta(S_b) \neq \phi$;
    return $G[S_a, S_b]$
```

Algorithm 5 Local Search (LS)

4.3 Local Search

In practice, given a Triple Network, CDC subgraphs around pre-selected seeds are very informative. For example, given a set of research interests, a list of research groups that densely publish in these areas; or given a list of people, hot topics of discussion among their friend-circles. To capture this intuition, in this section we introduce a bottom-up approach for obtaining feasible CDC subgraphs, namely Local Search.

Given connected components S_a and S_b containing desired seeds in V_a and V_b, the local search algorithm finds CDC subgraph by adding nodes that increase the density while maintaining connectedness of S_a and S_b. More precisely, outlined as algorithm 5, this algorithm iteratively includes previously un-included boundary node of $S_a \cup S_b$ with the maximum adjacency value to the set of included nodes.

4.3.1 Time complexity of Local Search

After calculating 2-approximation Steiner trees and maintaining degree:node binary heap of the boundary $\delta(S_a) \cup \delta(S_b)$, the time complexity of Local Search algorithm is

$O(|V_a|^2 log(V_a) + |V_b|^2 log(V_b) + E_c)$. However, in practice, the search stops in a few iterations and hence imperially the fastest algorithm.

4.4 Algorithms of variants

OCD subgraphs are bi-products of mining CDC subgraphs for the top-down algorithms. The stage I of finding the densest bi-partite subgraph remains the same, but we apply stage II connected components of V_a or V_b and find the resultant OCD subgraph with highest density. In bottom-up approach, we use LS algorithm with either S_a or S_b to be empty. The resultant subgraphs rendered by LS more effective, but smaller in size in comparison to the top-down algorithms. We obtain CDC_seeds and OCD_seed subgraphs using LS algorithm.

PART 5

EXPERIMENTS RESULTS

In this section, we evaluate the effectiveness and efficiency of the proposed methods through comprehensive experiments on real and synthetic datasets. We demonstrate the effectiveness of CDC and OCD subgraphs by illustrating novelty of the information obtained from these subgraphs on real Triple Networks. We demonstrate the efficiency of our algorithms by measuring the running times of the algorithms as well as the density ratios of the resultant CDC subgraphs.

The experiments are coded in Python 2.7 and run on 8 cores Intel Core i7 3.6Gz CPU with 32G memory.

5.1 Real Triple Networks

We constructed several Triple Networks form a variety of application domains, here we present networks constructed from Twitter, NYC taxi data, Flixter and ArnetMiner coauthor datasets.

5.1.1 NYC Taxi data

New York City (NYC) yellow cab taxi data is a public dataset [15] where each taxi trip's pick-up and drop-off point has geographic location in decimal degrees. We consider the trips from June 2016 to construct a Triple Network. The geographic location accuracy of this dataset is thresholded up-to 5 decimal points, preserving granularity to different door-entrances. As a result, we obtain $|E_c| = 2,066,569$ taxi trips with $|V_a| = 733,896$ distinct pick-up and $|V_b| = 794,085$ distinct drop-off points. We consider the points within haversine distance of 50 meters to be connected, and obtain $G_a(V_a, E_a)$ of and pick-up points and $G_b(V_b, E_b)$ of drop-off points with $|E_a| = 31,513,503$ and $|E_b| = 13,465,065$ respectively.

5.1.2 Twitter network

Twitter is a social media for micro-blogging, where users can follow each-other for updates. To extract meaningful user-follower relationships, we choose the most popular news networks, namely CNN, Huffington Post and Fox News and randomly chose a few thousand of their intersecting followers. We iteratively grow this network by including followers of existing nodes, with certain number of recent tweets, and number of their friends and followers. Using Twitter's REST API, we construct a 5-hop network with $|V_a| = 61,726$ users and $|E_a| = 7,008,491$ edges. We collect $|V_b| = 3,679,824$ different hashtags from these users' most recent tweets, with the users posting these hashtags $|E_c| = 48,269,139$ times. We considered two hashtags related if they appeared in the same tweet. We also keep a count of the number of tweets per hash-tag co- occurrence. By obtaining $|E_b| = 2,896,925$ hashtag co-occurrence relations, we concluded building the Twitter Triple Network.

5.1.3 ArnetMiner Coauthor dataset

ArnetMiner Coauthor dataset is comprised of two types of relations: $|V_a| = 1,712,433$ authors and their $|V_b| = 3,901,018$ research interests,with $|E_c| = 2,581,981$ relations of authors to their research interests, and $|E_a| = 4,258,946$ co-author relationships. We consider two interests linked if they co-occur in the list of research interests of an author. We keep a count of number of authors per research interest co-occurrence. We obtain $|E_b| = 953,490$ such edges.

5.1.4 Flixter dataset

Flixter[16] is a social network of users and their movie ratings. In the Flixter dataset, there are $|V_a| = 786,963$ users, and $|E_a| = 7,058,819$ edges representing their friend-circle. The user-item ranking matrix is comprised of $|E_c| = 8,184,462$ user rankings for $|V_b| = 48,794$ movies, with rating scale from 1 to 5 with 0.5 increment. With no sufficient information, we have $|E_b| = 0$ edges relating these movies.

Table (5.1) The real triple-networks on NY Taxi data (TX), Twitter (TW), ArnetMiner (AM), and Flixter (FX) datasets

Data	$\lvert V_a\rvert$	$\lvert E_a\rvert$	$\lvert V_b\rvert$	$\lvert E_b\rvert$	$\lvert E_c\rvert$
TX	$733,896$	$31,513,503$	$794,085$	$13,465,065$	$2,066,569$
TW	61726	7008491	3679824	2896925	48269139
AM	1712433	4258946	3901018	953490	12589981
FX	786936	7058819	48794	0	8196077

Table (5.2) Logistics of Synthetic Random and R-MAT networks

$\lvert V_a\rvert$	$\lvert E_a\rvert$	$\lvert V_b\rvert$	$\lvert E_b\rvert$	$\lvert E_c\rvert$
2^{19}	5×10^6	2^{19}	5×10^6	10^7
2^{20}	10^7	2^{20}	10^7	2×10^7
2^{21}	2×10^7	2^{21}	2×10^7	4×10^7
2^{22}	4×10^7	2^{22}	4×10^7	8×10^7

The table 5.1 describes the statistics of the real Triple Networks.

5.2 Synthetic Triple Networks

In order to evaluate efficiency of our algorithms, we construct two types of synthetic Triple Networks.

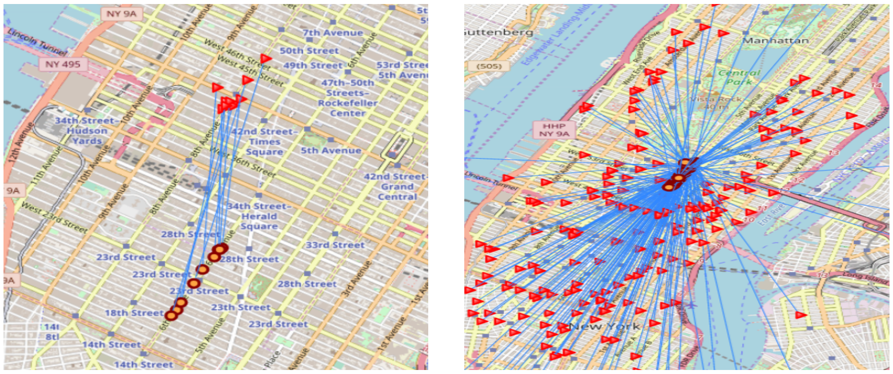
We generate Random Networks, with synthetic generation of G_a, G_b and G_c having random edges.

To approximate real world Triple Networks synthetically, we also generate R-MAT Networks with G_a and G_b having R-MAT edges [17181718] and G_c having random edges.

We generate four different configurations of synthetic graphs for Random and R-MAT networks, mentioned in table 5.2.

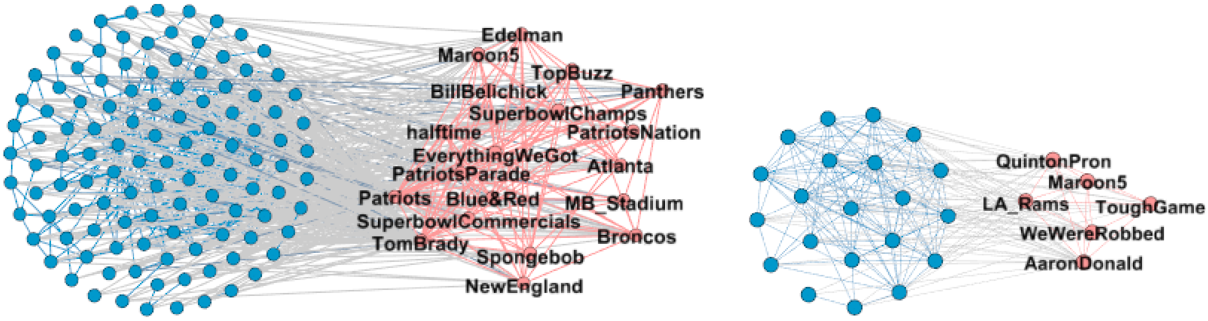
5.2.1 Effectiveness Evaluation on Real Networks

We illustrate the effectiveness of CDC subgraphs and variants by emphasizing the knowledge gain from these patterns obtained from real networks. These figures demonstrate that



(a) CDC subgraph yielding directional flow of human migration in 1 hour period (b) OCD subgraph yielding drop-off hot-spots on a street in 4 hours period

Figure (5.1) CDC and OCD subgraphs from NY Taxi data. Traingles and circles represent pick-up and drop-off points respectively



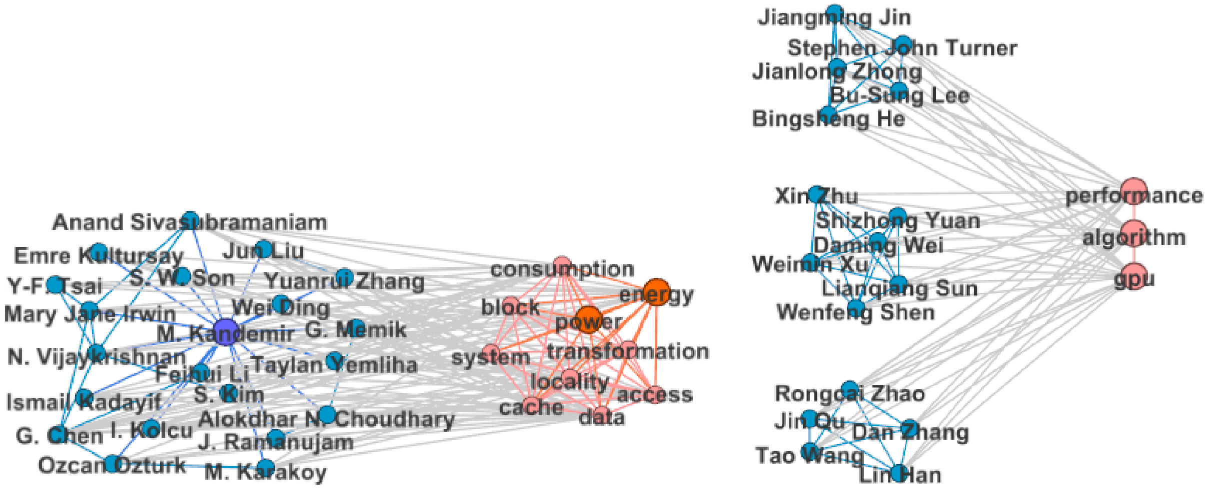
(a) CDC subgraph representing Patriots' fans (b) CDC subgraph representing Rams' fans

Figure (5.2) CDC subgraphs from Twitter. Users-followers networks on the left and hashtag networks on the right.

CDC subgraphs and variants are communities detected by the strong associations to their attributes. These subgraphs identify similar opinions, research interests and factors influencing communities. They are also effective tools for hot-spot detection and fraud detection.

NYC Taxi data Figure 5.1 illustrates CDC and OCD subgraphs with pick-up and drop-off points as triangles and circles respectively.

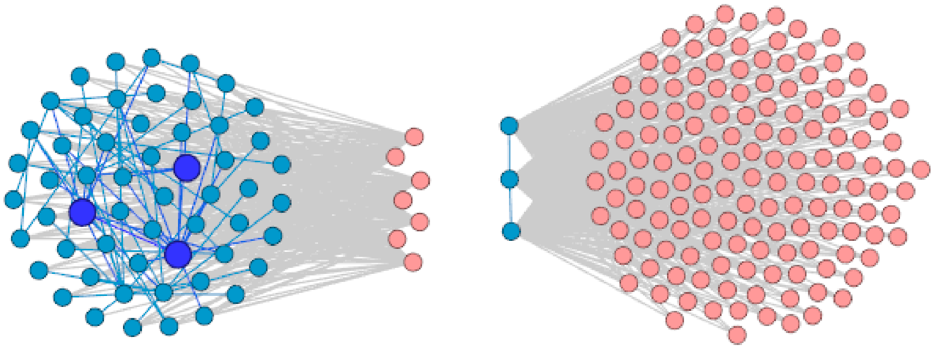
Figure 5.1(a) illustrates the CDC subgraph with pick-up locations on 6th Avenue between 18th and 27th street populated with food and shopping destinations, and drop-of locations on 8th Avenue. This CDC subgraph is generated by observing the 6:00-7:00 pm traffic on June 4, 2016. The drop-off points are clustered near 42nd street Port Authority bus terminals of city transit. This CDC subgraph gives a directional flow of human migration



(a) CDC_seeds subgraph with author and research-interests seeds

(b) OCD_seed subgraph with research-interest seeds

Figure (5.3) CDC and OCD subgraphs from ArnetMiner. Co-author networks on the left and research-interest networks on right.



(a) OCD_seed subgraph of user seeds influenced by movies

(b) OCD subgraph of a possible fraud

Figure (5.4) OCD subgraphs from Flixter. User networks on the left and movie networks on the right.

in a short distance during a specific time-frame. Figure 5.1(b) illustrates OCD subgraph with pick-up seeds near 5th Avenue and Central Park South. This subgraph is generated by observing 4:00-8:00 pm traffic on June 1, 2016. The pick-up points are scattered along Manhattan and the drop-off points are clustered around Pennsylvania Station, a public transit hub. Thus, OCD subgraphs could be equivalents to hot-spot detection.

Twitter Network Figure 5.2 represents CDC subgraphs obtained from Twitter Network. Left and right subgraphs represent users-followers and hashtag networks. We remove usernames to protect user privacy. These figures represent twitter users and their opinions about SuperBowl contenders, Patriots and LA Rams. Hence, CDC subgraphs can identify communities with contrasting opinions.

ArnetMiner coauthor data Figure 5.3 depicts CDC_seeds and OCD_seed subgraphs from ArnetMiner Triple Network. Left and right subgraphs represent author-coauthor and research-interest networks.

Figure 5.3(a) is a CDC_seeds subgraph with randomly chosen author seed {M.Kandimir} and interest seeds {power,energy}. This pattern yields author seed's associates working on related research topics of interest seeds. Figure 5.3(b) is OCD_seed subgraph with interest seeds chosen as {algorithm, gpu, performance}. This patterns yields 16 authors and their respective co-author networks with publications related to interests seeds. Thus, even with the given seeds, the CDC and OCD subgraphs are different from supervised community detection.

Flixter data Figure 5.4 depicts OCD subgraphs illustrating influence of movies on users. Left and right subgraphs represent the users' social networks and the movies networks, The users networks are connected.

Figure 5.4(a) is an OCD_seed subgraph with users seeds, chosen at random. The right network represents movies with 5 star rankings by the users on the left. This pattern hence finds the movies influencing the friend-circle of the seed users. An OCD subgraph in figure 5.4(b) depicts a suspicious ranking activity, where the 3 users on the left give a 5 star ranking to 144 movies on the right. CDC and OCD subgraphs hence illustrate the power of potential fraud detection.

5.2.2 Efficiency evaluation

We evaluate the efficiency of our heuristic algorithms by their running-time and the quality of the resulting CDC subgraphs from real and synthetic networks.

Table (5.3) CDC subgraph densities from random networks

nodes	2^{20}	2^{21}	2^{22}	2^{23}
DBP	19.083	19.095	19.094	19.086
GND	18.713	18.705	18.691	18.720
GRD	18.901	18.836	18.837	18.698
FRD	7.401	7.389	7.402	7.401

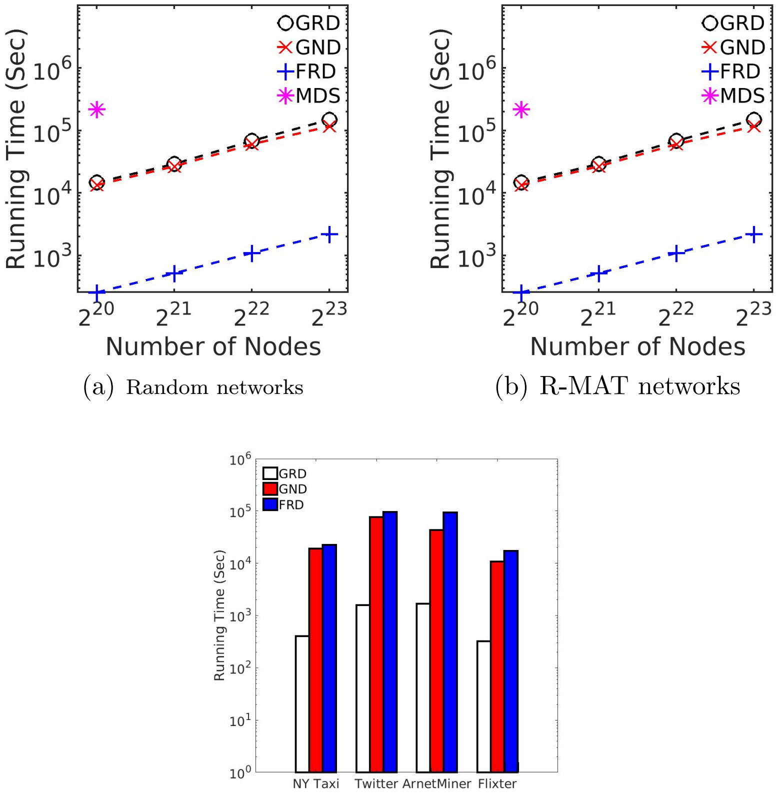
Table (5.4) CDC subgraph densities from R-MAT networks

nodes	2^{20}	2^{21}	2^{22}	2^{23}
DBP	19.071	19.065	19.073	19.072
GND	17.028	16.761	17.019	16.627
GRD	17.201	17.002	17.046	16.689
FRD	6.612	6.610	6.509	6.501

Greedy node deletions The running-times of MDS, GND, GRD, FRD algorithms on real, random and R-MAT networks are depicted in Figure 5.5. The x axis represents the number of nodes in $V_a \cup V_b$ and the y axis represents log scale of seconds. Each point represents running-time of the algorithm for given network. The running-time of MDS algorithm for larger networks is more than 24 hours, when we halted the algorithm computations. Running-times increase with network size, but vary a little for random and R-MAT graphs of the same size. FRD with $\epsilon = 0$ is the fastest algorithm.

We discover that GRD yields the densest bipartite subgraph among all algorithms. The densities of CDC subgraphs obtained by GND, GRD and FRD from random and R-MAT networks are presented in table 5.3 and 5.4. For each graph, DBP represents the density of the densest bipartite graph obtained by GRD, without being connected in G_a or G_b. The ratio, DBP/CDC densitiy, varies a little with the network size. This trend is observed across all network types and algorithms. GRD produces the best and FRD with $\epsilon = 0$ produces the least accurate results.

Local Search (LS) Given the seeds of V_a and V_b, LS produces meaningful, locally dense CDC patterns. We evaluate the efficiency of LS algorithm by measuring its running-times with 2, 4 and 8 seeds. Figure 5.6 presents the running-times of LS. The x axis represents the number of nodes in $V_a \cup V_b$ and the y axis represents running-times in seconds. Each point represents running-time of FRD for given network and seed configuration. The seeds are chosen randomly in the same connected components. The boundaries $\delta(S_a)$ and $\delta(S_b)$ grow larger with increase in the number of seeds. Hence the running-time of LS increases



(a) Random networks

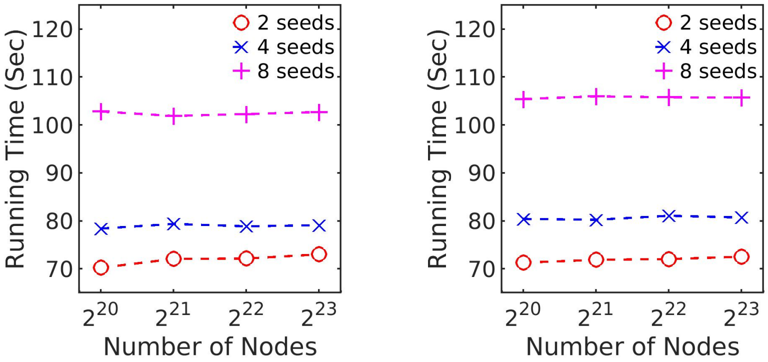
(b) R-MAT networks

(c) Real Networks

Figure (5.5) Running-times for MDS, GND, GRD and FRD

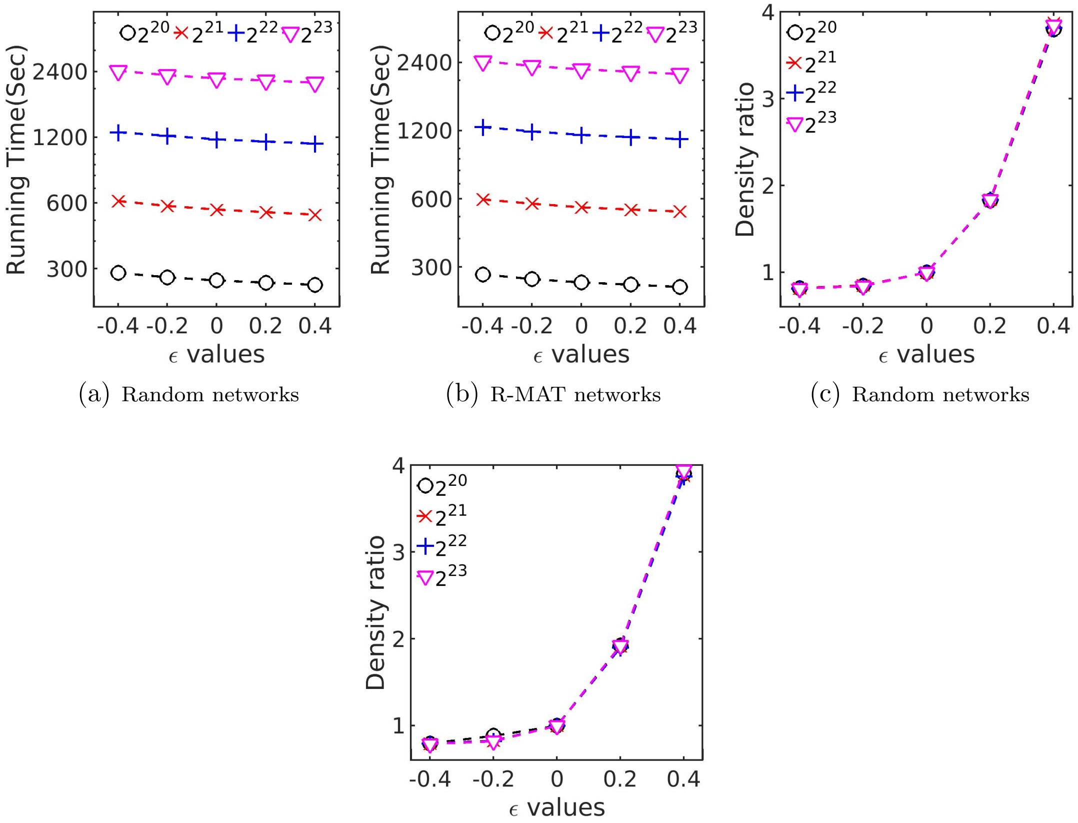
with the number of seeds. We observe similar trends from real networks. In synthetic networks, for a given number of seeds, LS running-times vary a little across different network sizes. This is because LS halts when the density of the current CDC subgraph starts decreasing, which depends only on the local topologies of G_a and G_b.

Fast Rank Deletion (FRD) The purpose of FRD is to obtain feasible CDC subgraphs faster. This is achieved by deleting all the nodes with degree less than $(1 + \epsilon) *$ *average degree* at each pass. However, lower ϵ values result in fewer deletions per pass, defying the purpose of FRD. Higher ϵ values result in more deletions per pass, lowering the densities of the resulting CDC subgraphs. Hence the meaningful results are



(a) Random networks (b) R-MAT networks

Figure (5.6) LS running-times with 2,4 and 8 seeds



(a) Random networks (b) R-MAT networks (c) Random networks

(d) R-MAT networks

Figure (5.7) FRD evaluations for $\epsilon \in [-0.4, 0.4]$

obtained with ϵ values in the range of interval $[-0.4, 0.4]$.

Figures 5.7(a) and 5.7(b) represent the running-times of FRD. The x axis represents different ϵ values and the y axis represents running-times in log scale of seconds. Each point represents running-time of FRD for given network and ϵ configurations. Increase in ϵ value causes higher amount of deletion per pass, resulting in fewer passes. Hence, the running-times decrease with the increase of ϵ.

Figures 5.7(c) and 5.7(d) represent the density change of resultant CDC subgraphs for given ϵ value, with respect to $\epsilon = 0$. The x axis represents different ϵ values, and the y axis represents the ratio, Density of CDC for $\epsilon = 0$/Density of CDC with given ϵ. Each point represents this density ratio obtained by FRD, for given network and ϵ configurations. Higher ϵ values result in more deletions per pass, lowering the densities of the resulting CDC subgraphs. Hence, the density ratio increases as the ϵ value decreases. We observe similar trends from real networks. The densities of resultant CDC subgraphs obtained by FRD depend on network topologies. Hence, for the same type of synthetic networks with the same ϵ value, the variance in the density ratio is low.

PART 6

CONCLUSION

In this paper, we introduce Triple Network, its CDC subgraph problem and its variants. We provide heuristics to find feasible solutions to these patterns, otherwise NP-Hard to find. We conclude that CDC subgraphs yield communities with similar charasteristics by illustrating the information gain of these patterns in NYC taxi, Twitter, ArnetMiner, and Flixter networks. We demonstrate the efficiency of our algorithms on large real and synthetic networks by observing running-time and density trends in real and synthetic networks.

PART 7

FUTURE WORK

As future work, we propose the following:

- Use a parallel and distributed implementation of max-flow min-cut algorithm to extend MDS algorithm to large graphs of the size 2^{20} to 2^{23}
- Compare the results of MDS algorithms to see if GND is a 2 approximation of CDC subgraphs, if not what is the relation between the MDS baseline and CDC results of our heuristics
- Prove 2-approximation guarantees of GND and GRD algorithms
- Provide parallel-and distributed versioons of these algorithms
- Compare the results of our heuristics to results of clustering with graph embeddings

REFERENCES

[1] S. Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, no. 3-5, pp. 75–174, 2010.

[2] R. Kelley and T. Ideker, "Systematic interpretation of genetic interactions using protein networks," *Nature biotechnology*, vol. 23, no. 5, p. 561, 2005.

[3] J. Pei, D. Jiang, and A. Zhang, "On mining cross-graph quasi-cliques," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining.* ACM, 2005, pp. 228–238.

[4] H. Hu, X. Yan, Y. Huang, J. Han, and X. J. Zhou, "Mining coherent dense subgraphs across massive biological networks for functional discovery," *Bioinformatics*, vol. 21, no. suppl_1, pp. i213–i221, 2005.

[5] W. Li, H. Hu, Y. Huang, H. Li, M. R. Mehan, J. Nunez-Iglesias, M. Xu, X. Yan, and X. J. Zhou, "Pattern mining across many massive biological networks," in *Functional coherence of molecular networks in bioinformatics.* Springer, 2012, pp. 137–170.

[6] Y. Wu, R. Jin, X. Zhu, and X. Zhang, "Finding dense and connected subgraphs in dual networks," in *Data Engineering (ICDE), 2015 IEEE 31st International Conference on.* IEEE, 2015, pp. 915–926.

[7] A. V. Goldberg, *Finding a maximum density subgraph.* University of California Berkeley, CA, 1984.

[8] S. Khuller and B. Saha, "On finding dense subgraphs," in *International Colloquium on Automata, Languages, and Programming.* Springer, 2009, pp. 597–608.

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

[10] A. Bhaskara, M. Charikar, E. Chlamtac, U. Feige, and A. Vijayaraghavan, "Detecting high log-densities: an o (n 1/4) approximation for densest k-subgraph," in *Proceedings of the forty-second ACM symposium on Theory of computing.* ACM, 2010, pp. 201–210.

[11] B. Saha, A. Hoch, S. Khuller, L. Raschid, and X.-N. Zhang, "Dense subgraphs with restrictions and applications to gene annotation graphs," in *Annual International Conference on Research in Computational Molecular Biology.* Springer, 2010, pp. 456–472.

[12] Y. Sun, Y. Yu, and J. Han, "Ranking-based clustering of heterogeneous information networks with star network schema," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2009, pp. 797–806.

[13] B. Boden, M. Ester, and T. Seidl, "Density-based subspace clustering in heterogeneous networks," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases.* Springer, 2014, pp. 149–164.

[14] D. Shah, S. Prasad, and Y. Wu, *Finding Connected-Dense-Connected Subgraphs and variants is NP-Hard.* Department of Computer Science, Georgia State University, Atlanta, GA, 2019. [Online]. Available: https://scholarworks.gsu.edu/computer_science_technicalreports/2/

[15] NYC taxi & limousine commission - trip record data. [Online]. Available: https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

[16] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proceedings of the fourth ACM conference on Recommender systems.* ACM, 2010, pp. 135–142.

[17] D. Chakrabarti, Y. Zhan, and C. Faloutsos, "R-mat: A recursive model for graph mining," in *Proceedings of the 2004 SIAM International Conference on Data Mining.* SIAM, 2004, pp. 442–446.

[18] D. A. Bader and K. Madduri, “Gtgraph: A synthetic graph generator suite,” *Atlanta, GA, February*, 2006.