Improving Togetherness Using Structural Entropy

Siyu Zhang[1], Jiamou Liu[3], Yiwei Liu[1], Zijian Zhang[2,3(✉)], and Bakhadyr Khoussainov[4]

[1] School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
{3120181073,yiweiliu}@bit.edu.cn
[2] School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing, China
zhangzijian@bit.edu.cn
[3] School of Computer Science, The University of Auckland, Auckland, New Zealand
jiamou.liu@auckland.ac.nz
[4] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China
bmk@uestc.edu.cn

Abstract. A major theme in the study of social dynamics is the formation of a community structure on a social network, i.e., the network contains several densely connected region that are sparsely linked between each other. In this paper, we investigate the network integration process in which edges are added to dissolve the communities into a single unified network. In particular, we study the following problem which we refer to as togetherness improvement: given two communities in a network, iteratively establish new edges between the communities so that they appear as a single community in the network. Towards an effective strategy for this process, we employ tools from structural information theory. The aim here is to capture the inherent amount of structural information that is encoded in a community, thereby identifying the edge to establish which will maximize the information of the combined community. Based on this principle, we design an efficient algorithm that iteratively establish edges. Experimental results validate the effectiveness of our algorithm for network integration compared to existing benchmarks.

Keywords: Togetherness · Social network · Structural entropy

1 Introduction

1.1 Background

A social network consists of a set of individuals and their social bonds, forming a graph that exhibits a range of non-trivial statistical properties. *Community*

Z. Zhang—This paper is supported by National Natural Science Foundation of China No. 62172040, No. U1836212, No. 61872041.

S. Deng et al. (Eds.): MobiCASE 2021, LNICST 434, pp. 85–98, 2022.
https://doi.org/10.1007/978-3-030-99203-3_6

structure amounts to one of the most prevalent properties of a social network. This property asserts that the graph can be broadly viewed as consisting of several densely-connected subgraphs, called *communities*, that are loosely connected between each other. Understanding the community structure provides key insights on the social compositional of the society. As social bonds can be seen as channels of interactions and passages of information and knowledge, communities are decisive in determining important social and behavioral traits such as social cohesion, self-identity, and the emergence of norms and trust [6,16,17].

A perpetual theme in studies on social networks is the dynamics of social ties. Countless scenarios exist where needs arise for updating the social bonds between individuals, thereby changing the graph structure. Many of these scenarios would involve intensions to integrate two communities into one. Take for example, business mergers and acquisitions which see two companies dissolve into a single entity, the marriage between two large families with complex relations, the assimilation process when immigrants arrive at a new country, and the healing process between two divided political fractions of a government. Creating a smooth unification between two disjoint communities in these scenarios is a common desirable outcome, which enhances the building of common grounds, mutual understanding, cooperation and conflict resolution.

The concept of *togetherness* [21] aims to capture the level of unity between two communities. This concept is defined in the context of *network integration*. In a nutshell, network integration corresponds to a process where communication channels are built, creating opportunities for interactions between individuals who otherwise belong to different communities. More formally, take a graph $G = (V, E)$ which represents a social network, i.e., V is the set of individuals and E is a set of (undirected) social ties. Suppose V is the union of several disjoint communities and C_1, C_2 are two of these communities. The network integration process adds a sequence of new edges $e_1, e_2, \ldots, e_\ell$ between nodes in C_1 and C_2 to form a graph G' in which members of $C_1 \cup C_2$ integrate into a single community. To be more precise, the authors of [21] introduced a number of methods to measure togetherness of two communities in an integrated network. These notions are based on the distances between nodes. In particular, the strongest notion among them is Δ-*togetherness*, which refers to the reciprocal of graph diameter in the integrated network. Several strategies were proposed in [20,21] to perform the network integration process to boost Δ-togetherness.

We argue that there is a need for another togetherness notion for network integration. Indeed, the *diameter* of a graph refers to the longest distance between any pair of nodes in the graph. Δ-togetherness assumes that the diameter of the integrated graph is the sole indicator for a united community. This is not accurate as there are situations where two communities are far from being fully integrated when the updated network reaches a small diameter (i.e., the diameter of the graph obtained in the "combined community" is no more than the diameters of any of the original communities). Consider, e.g., graph $G = (V, E)$ whose $V = \{v, u\} \cup U_1 \cup U_2$ where $U_1 = \{v_1, \ldots, v_n\}$ and $U_2 = \{u_1, \ldots, u_n\}$ where $n \in \mathbb{N}$, and E contains edges $\{v_i, v_j\}, \{u_i, u_j\}$ for any $i \neq j \in \{1, \ldots, n\}$ and

$\{v_i, u_j\}$ for any $i, j \in \{1, \ldots, n\}$ (thus U_1, U_2 forms a complete graph of size $2n$), and edges $\{v, v_i\}$, $\{u, u_i\}$ for any $i \in \{1, \ldots, n\}$. Now define two disjoint communities C_1, C_2 each of which has a graph structure that is isomorphic to G. To integrate these two communities, we add two new edge, the first between the copy of v in C_1 and the copy of v_1 in C_2, the second between the copy of u in C_2 and the copy of u_1 in C_1. Analyzing this graph it is apparent that $C_1 \cup C_2$ is far from being unified, despite achieving a good Δ-togetherness as:

1. Both the combined community (over $C_1 \cup C_2$) and the original communities (over C_1 and C_2, respectively) have diameter 3, suggesting high Δ-togetherness.
2. Each community C_i ($i \in \{1, 2\}$) has a *intra-density* (defined as the number of edges over nodes *within* the community) $(n(2n-1)+2n)/(2n+2) = (2n^2+n)/(2n+2)$ which tends to ∞ as $n \to \infty$.
3. The communities C_1 and C_2 have a *inter-density* (defined as the number of edges over nodes *between* C_1 and C_2) $2/(4n+4)$ which tends to 0 as $n \to \infty$.

To derive a more appropriate form of togetherness, it is necessary to recall our original motivation, namely, integrating two communities so that they eventually *appear* as a single community. The maturity of algorithms for *community detection* [12] means that one may assess togetherness through these algorithms. However, community detection algorithms vary vastly and can give inconsistent results. Going one step deeper, we look at what is behind the success of community detection algorithms, that is, they reveal regions of the graph that provide the most information regarding the network structure as a whole. The key to defining togetherness thus lies in revealing structural information of a graph.

1.2 Contributions

Motivated by the discussions above, we investigate network integration and togetherness in a graph that has a salient community structure. Our contribution include: (1) We invoke Li-Pan *structural information theory* [15] and borrow from their work the notion of *structural entropy* to quantify the amount of uncertainty within a graph. This notion allows us to determine how much information is gained through a community structure, which leads to our togetherness improvement problem. (2) We discover an interesting phenomenon, namely, *maximum degree principle* that link information gain through community structure with the degrees of nodes in the communities. Through this finding, we propose an efficient algorithm (TIE) for solving the togetherness improvement problem. (3) Finally, we demonstrate the effectiveness of our algorithm using experiments on both real-world and synthetic network datasets.

1.3 Related Work

Network Integration and Togetherness. Network integration was introduced as optimization problems by Moskvina and Liu in [19], who proposed two

types of network integration problem: integrating a newcommer into a network (the so-called *network building* process), and combining two (sub-)networks into one through optimizing some global measure. Both of these problems involve adding a number of new edges to the graph iteratively and the objectives are to maximize certain notion of *social capital*, which refers to structure-based measurement of efficacies such as influence and cohesion [13,25]. In [19], the objectives are defined in terms of distances between nodes. Along the pathway of network building, [26,27] investigated building networks for a newcomer in a dynamic network, [7] added concerns to the cost of adding social ties, [24] generalized the objective from distance-based to other type of centrality measures, [8] further discussed the concept of social capital and proposed bonding and bridging social capital, and [28] focused on the situation where only partial observation is made by the newcomer.

While integrating a newcomer into a network mainly concerns social capital gained by the newcomer, the task of combining networks targets at global notions of social capital. In particular, *togetherness* embodies this type of social capital. There, Moskvina and Liu in [20] proposed a kind of network integration problem by establishing edges between two networks to minimize the diameter in the combined network. [21] proposed the notions of $\exists$-togetherness, $\forall$-togetherness, as well Δ-togetherness. They proposed the central-periphery (CtrPer) algorithm to facilitate the selection of new edges to be added to improve togetherness in the combined network.

We mention the two paradigms that persist in network science for quantifying properties of networks: The first is a *distance-based* approach which evaluates nodes based on the minimum number of hops, e.g., eccentricity and closeness centrality; the second is a *volume-based* approach which evaluates nodes based not only on distance but also on the number of paths connecting nodes [3]. Notions such as network flow and betweenness are all examples of tools within the latter paradigm. As discussed above, the togetherness notions introduced above are predominantly distance-based metrics, which is insufficient to correctly capture the community structure of a network. In this paper, we adopt a notion of structural entropy which is inherently a volume-based property.

Community Detection. Community detection amounts to one of the most enduring themes in network science [12]. Due to complexity of real-world networks and inconsistencies among the intended uses of communities, there has not been a universally-accepted notion of a community structure. The closest to a generally-agreeable notion of communities is through Newman's *modularity* [22], which quantifies the level of deviation between the level of connectivity within communities and the expected level of connectivity in a null model that has the same degree distribution. Communities are defined as those that maximizes this deviation. However, there are known insufficiencies to this formulation such as the resolution limit [11]. Nevertheless, modularity-based community detection methods such as the Louvain method [2] are widely adopted due to their superior performance.

2 Togetherness Based on Structural Entropy

2.1 Network Integration and Togetherness

By a *network* we mean an undirected and unweighted graphs $G = (V, E)$. A *community structure* over a network G is a partition of the node set V into a collection of subsets $\{C_1, C_2, \ldots, C_L\}$, called *communities*, where $\cup_{i=1}^{k} C_i = V$ and $\bigwedge_{i \neq j} C_i \cap C_j = \varnothing$. We need the following notions: N_v denotes the *neighborhood* of $v \in V$, d_v denotes the *degree* $|N_v|$ of a node $v \in V$, ν_i denotes the *volume* $\sum_{v \in C_i} d_v$ of C_i where $1 \leq i \leq L$, g_i denotes $|\{\{v, u\} \in E \mid v \in C_i, u \in V \backslash C_i\}|$. In general, if a community structure is given as input, we assume each community induces a densely-connected subgraph of G that is sparsely linked to other communities.

Network integration is a process that introduces a sequence of new edges to the graph to improve togetherness. More formally, fix two communities C_i, C_j, the process iteratively adds edges $e_1, e_2, \ldots$ that connect members of C_i with members of C_j. Our goal is to add as few edges as possible before $C_i \cup C_j$ can be regarded as a single community (as testified by running different community detection algorithms). Figure 1 illustrates an example of network integration using Zachary's karate club[1], a standard benchmark dataset for community detection. Communities in the graphs are identified using the Louvain method. It is seen that nodes are originally separated into four communities (as shown in different colors). After establishing 9 new edges (shown in red) between the communities in blue and in purple, a new community is formed that contains members from both original communities (shown on the right).

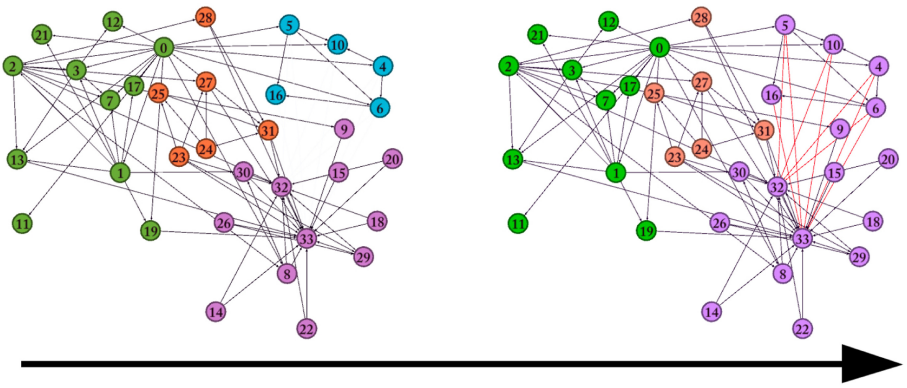

Fig. 1. Integrating two communities in Zachary's karate club network. After adding 9 edges between the blue and purple community, the two communities integrate as validated by the Louvain algorithm. (Color figure online)

[1] http://www-personal.umich.edu/~mejn/netdata/.

2.2 Structural Entropy

To give a formal definition of the problem, it is necessary to fix a notion of community. Here, we invoke structural information theory. The ability of quantify information embodied in structures has been a "grand challenge in the half-a-century-old computer science" as asserted by Brooks in [5]. For community detection, it has been conjectured that entropy-based methods would play a major role [9]. Many notions of entropy are defined on graphs [1,4,10,18], all of which are essentially Shannon entropy applied to different types of distributions, which measures many aspects of the graph but not about encoding of the graph structure itself and certainly not about a community structure.

Towards building a structural information theory, Li and Pan developed a framework in [15] to quantify structure information of graphs. We adopt this framework in this paper. Different from earlier notions of structural entropy, the framework by Li and Pan proposes a hierarchical structure to represent a graph G, with the root of the tree (top-most level) being the entire graph, the next level being the different communities that partition G which form the children of the root, and the level below that being sub-communities that partition a community, and so on, where each level downwards represent a collection of sets of nodes that forms a partition of the set above it. The leaves of this tree represent singleton node sets, i.e., individual nodes in the graph G. We call this hierarchy an *encoding tree* of G. At each level of this hierarchy, a graph encoding scheme is defined. We interpret Li and Pan's framework as follows which is different from their original description. Let $G = (V, E)$ be a graph with communities $\mathcal{P} = \{C_1, \ldots, C_L\}$. We study two questions:

- The first inquires about the minimum description length of G *without* community structure.
- The second inquires about the minimum description length of G *with* community structure.

For the first question, we consider a depth-1 encoding tree $\mathcal{T}_1$ where all children of the root are singleton nodes (hence no community information). In this scenario a *one-dimensional coding function* is a mapping $f : V \to \{0,1\}^\star$ that assigns every leaf of the tree, i.e., a node in V, to a binary codeword. Under this encoding, an edge $\{v, u\} \in E$ has codeword $f(v)f(u)$ (for some fixed order between v, u). Now consider a codeword of the graph G by listing the codewords of all edges in G as $f(v_1)f(u_1), f(v_2)f(u_2), \ldots$ (every edge appears once and the order does not matter). In this encoding, every node $v \in V$ occurs exactly d_v times, among $2|E|$ node occurrences in total. It is thus reasonable to seek a coding function f that produces the shortest length description of all edges. Shannon entropy gives the lowerbound on the average length of nodes, which we call *one-dimensional structural entropy* of G. This value is our answer to the first goal.

Definition 1 [15]**.** *The* 1D structural entropy *of* G *is*

$$\mathcal{H}^1(G) = -\sum_{i=1}^{n} \frac{d_i}{2m} \cdot \log_2 \frac{d_i}{2m}. \tag{1}$$

For the second question, we consider a depth-2 encoding tree $\mathcal{T}_2$ where the children of the root denote communities $C_1, \ldots, C_L$, and the leaves (i.e., individual nodes) are children of their corresponding communities. In this case, a *two-dimensional coding function* is a pair of mappings $f_1 \colon V \to \{0,1\}^\star$ and $f_2 \colon \mathcal{P} \to \{0,1\}^\star$. In other words, f_1 assigns a codeword to every node in G and f_2 assigns a codeword to every community in $\mathcal{P}$. An edge $\{v,u\} \in E$ under this coding function can be considered as the shortest path that goes from the leaf v to leaf u on the tree:

- if u, v belong to the same community C_i, then this shortest path simply goes from v to C_i, and then from C_i to u. Thus we use $\langle\{v,u\}\rangle = f_1(v)f_1(u)$ to code this edge.
- if $u \in C_i$ and $v \in C_j$, then this shortest path goes from v to C_i, to the root, then to C_j, and finally to u. Thus we use $\langle\{v,u\}\rangle = f_1(v)f_2(C_i)f(C_j)f(u)$ to code this edge $\{v,u\}$.

Since the communities are explicitly coded in this scheme, the codeword $f_1(v)$ for each node can be made within its own community C_i. Thus the shortest $f_1(v)$ has expected length $\log_2 \frac{d_v}{\nu_i}$. When coding the community C_i, one also takes into account the frequency that a node belongs to C_i occurs in the list of all edges, and thus the shortest codeword of C_i has expected length $\log_2 \frac{\nu_i}{2|E|}$. Summing up, we define the two-dimensional structural entropy as our answer to the second question:

Definition 2 [15]**.** *The* 2D structural entropy $\mathcal{H}^{\mathcal{P}}(G)$ *is*

$$\sum_{j=1}^{L} \left[- \sum_{v_i \in C_j} \frac{d_i}{2|E|} \log_2 \frac{d_i}{\nu_j} - \frac{g_j}{2|E|} \log_2 \frac{\nu_j}{2|E|} \right]. \tag{2}$$

Here, $\mathcal{H}^{\mathcal{P}}(G)$ captures the average number of bits to encode nodes in E with the presence of community structure $\mathcal{P} = \{C_1, C_2, C_3, \ldots, C_L\}$. It is easy to see that $\mathcal{H}^1(G) \geq \mathcal{H}^{\mathcal{P}}(G)$ for any partition $\mathcal{P}$. Furthermore, $\mathcal{H}^1(G) = \mathcal{H}^{\mathcal{P}}(G)$ if and only if either $\mathcal{P} = \{V\}$ or $\mathcal{P} = \cup_{v \in V}\{\{v\}\}$.

2.3 Togetherness Improvement Through Entropy

Informally speaking, $\mathcal{H}^1(G)$ and $\mathcal{H}^{\mathcal{P}}(G)$ can be viewed as the amount of uncertainty within a graph before and after community detection, respectively. It is thus natural to compute the information gain as the amount of information revealed by the community structure $\mathcal{P}$.

Definition 3. *The* information gain *of G given community structure $\mathcal{P}$ is*

$$\rho^{\mathcal{P}}(G) = (\mathcal{H}^1(G) - \mathcal{H}^{\mathcal{P}}(G))/\mathcal{H}^1(G). \tag{3}$$

We are now ready to formally define the problem that we aim to solve in the paper. Suppose our goal is to integrate the two communities C_1 and C_2 in $\mathcal{P}$ by iteratively adding edges. In every step, we would like to add a new edge

$\{v, u\}$ such that the community structure $\mathcal{P}' = \{C_1 \cup C_2, C_3, C_4, \ldots, C_L\}$ has the maximum information gain. Let $C_1 \otimes C_2 = \{\{v, u\} \mid v \in C_1, u \in C_2\}$ and $G \oplus \{u, v\}$ be $(V, E \cup \{\{u, v\}\})$. The *togetherness improvement problem* seeks to

$$\text{Maximize}_{\{v,u\} \in (C_1 \otimes C_2) \setminus E} \ \rho^{\mathcal{P}}(G \oplus \{v, u\}). \tag{4}$$

3 The TIE Algorithm

Solving the problem above brute-force takes time $O(|C_1||C_2|)$ which is inefficient if the communities are large and the procedure needs to iterate. We now present a faster algorithm with a simple observation: namely that one can maximize $\rho^{\mathcal{P}}(G \oplus \{u, v\})$ simply by comparing the degrees of nodes in the communities.

Let $G = (V, E)$ be an connected undirected graph, $P = (C_1 \cup C_2, C_3, \ldots, C_L)$ be the original community structure of $G = (V, E)$ and d_v is the degree of $v \in V$. Assume C_1 and C_2 are the communities to be merged, then the expected community structure is $P = (C_1 \cup C_2, C_3, \ldots, C_L)$ We say that a non-edge $\{u, v\} \in C_1 \otimes C_2$ is RE-Max for $C_1 \otimes C_2$ if $\rho^P(G \oplus \{u, v\}) \geq \rho^P(G \oplus \{x, y\})$ for any non-edge $\{x, y\} \in C_1 \otimes C_2$. For the candidate non-edge $\{u, v\}$, we have the following max-degree theorem about RE-Max:

Theorem 1 (max-degree principle). *If $\{u, v\} \in (C_1 \otimes C_2) \setminus E$ is an optimal solution to the togetherness improvement problem(RE-Max), then $d_u \geq d_{u'}$ for any node $u' \in C_1$ with $\{u', v\} \in (C_1 \otimes C_2) \setminus E$, and $d_v \geq d_{v'}$ for any node $v' \in C_2$ with $\{u, v'\} \in (C_1 \otimes C_2) \setminus E$.*

Proof. If there exists a $\{u', v\} \in (C_1 \otimes C_2) \setminus E$ satisfying $d_{u'} > d_u$ and $\rho^{\mathcal{P}'}(G \oplus \{u', v\}) > \rho^{\mathcal{P}'}(G \oplus \{u, v\})$. By (1) and (2), the value of $\mathcal{H}^1(H) - \mathcal{H}^{\mathcal{P}}(H)$ when $H = G \oplus \{u, v\}$ or $H = G \oplus \{u', v\}$ both equal to

$$-\sum_{j \neq 1,2} \frac{\nu_j - g_j}{2|E| + 2} \log_2 \frac{\nu_j}{2|E| + 2} - \sum_{j \in \{1,2\}} \frac{\nu_j - g_j}{2|E| + 2} \log_2 \frac{\nu_j + 1}{2|E| + 2}.$$

Then Definition 3 implies that $\mathcal{H}^1(G \oplus \{u, v\}) < \mathcal{H}^1(G \oplus \{u', v\})$. Note that $\mathcal{H}^1(G \oplus \{u, v\}) - \mathcal{H}^1(G \oplus \{u', v\}) = -\frac{1}{2|E|+2}(F(d_u) - F(d_{u'}))$ where $F \colon \mathbb{R} \to \mathbb{R}$ is defined by $F(x) = (x + 1)\log_2(x + 1) - x \log_2 x$. Since F is monotonically increasing and $d_{u'} > d_u$, we have $\mathcal{H}^1(G \oplus \{u, v\}) > \mathcal{H}^1(G \oplus \{u', v\})$, contradiction with the assumption. Therefore $d_u \geq d_{u'}$ for any node $x \in C_2$ and $\{x, v\} \notin E$. The same proof can be applied to d_v.

Denote $\{u, v\} \in C_1 \otimes C_2$ or $\{u, v\} \in C_2 \otimes C_1$. We define a non-edge $\{u, v\} \in C_1 \& C_2$ as a critical edge for u if v has the maximal degree among all non-edge $\{u, y\} \in C_1 \& C_2$. From the max-degree principle in Theorem 1, if a non-edge u, v is RE-Max, the edge must be a critical edge, then we only need search RE-Max among the critical edges $\{u, v\}$ for $u \in C_1 \cup C_2$. The max-degree principle asserts that any new edge $\{u, v\}$ that amounts to an optimal solution to the togetherness improvement problem must satisfy that $\{u, v\} \in Cand$, where $Cand$ is a set of

critical edges. With this edge set, we can implement this algorithm iteratively. Based on this idea, we refer to this algorithm as **T**ogetherness **I**mprovement thru **E**ntropy (TIE):

ALGORITHM 1. Togetherness improvement algorithm TIE

input: $G = (V, E)$, budget k, expect community stucture $P = (C_1 \cup C_2, C_3, \ldots, C_L)$
output: a non-edge set E'

1: $E' = \phi$, $\rho_{max} = 0$
2: Create a candidate edge set $Cand = \{(x, y) \mid x \in C_1 \cup C_2\}$ so that $\{x, y\}$ is the critical non-edge of C_1&C_2
3: **for** $i = 1 \longrightarrow k$ **do**
4: **for** $\{u, v\} \in Cand$ **do**
5: **if** $\rho^P(G \oplus \{u, v\}) > \rho_{max}$ **then**
6: $u^* \leftarrow u$, $v^* \leftarrow v$, $\rho_{max} = \rho^P(G \oplus \{u, v\})$
7: **end if**
8: **end for**
9: $E' = E' \cup \{u^*, v^*\}$, $E = E \cup \{u^*, v^*\}$
10: $\rho_{max} = 0$,update $Cand$ by function UPDATE
11: **end for**
12: **return** E'
13:
14: **function** UPDATE($Cand, \{u^*, v^*\}$)
15: **for** $\{u, v\} \in Cand$ **do**
16: **if** $u = u^*$ or $v = v^*$ **then**
17: recalculate critical edge $\{u, v'\}$ for u
18: $Cand = Cand \cup \{u, v'\} \setminus \{u, v\}$
19: **else if** $\delta\{u, v^*\} = 0$ and $d_v < d_{v^*}$ and $\{u, v^*\} \notin E$ **then**
20: $Cand = Cand \cup \{u, v^*\} \setminus \{u, v\}$
21: **else if** $\delta\{u, u^*\} = 0$ and $d_u < d_{u^*}$ and $\{u, u^*\} \notin E$ **then**
22: $Cand = Cand \cup \{u, u^*\} \setminus \{u, v\}$
23: **end if**
24: **end for**
25: **return** $Cand$
26: **end function**

To simple, we denote $\delta\{u, v\} = 1$ if u,v belong to the same community and $\delta\{u, v\} = 1$ if they belong to different communities in Algorithm 1. This Algorithm contains two main steps: the computing step and the update step. In the computing step (line 4–8), TIE computes information gain $\rho^{\mathcal{P}}(G \oplus \{v, u\})$ for all the candidate edges to search the maximal one RE-Max. Since we only save one critical edge for each node $u \in C_1 \cup C_2$, then this step takes at most $|C_1|+|C_2|$. times. In the update step (line 14–26), TIE will update the candidate edges. Specially, if u is the endpoint of the optimal edge $\{u^*, v^*\}$ in computing step, the function recalculate the critical edge for u, and if the remain candidate edge is not critical, replace one of the endpoint with u^* or v^*. This step takes at

most $2(|C_1| + |C_2|)$. It means that the procedure of adding one edge takes time $O(|C_1| + |C_2|)$, it's far better than brute-force implementation that takes time $O(|C_1||C_2|)$.

Therefore, by iterating k times, the time complexity of TIE is $O(|C_1||C_2| + k(|C_1| + |C_2|))$, where $|C_1||C_2|$ is the time by creating a candidate edge set.

4 Experiment

We evaluate our solution to the togetherness improvement problem aiming to answer three questions: 1) how does TIE compare with existing benchmark w.r.t. improving togetherness when evaluated using an established community detection algorithm? and 2) How efficient is the TIE algorithm?

Table 1. The key statistics of real networks

Name	$\|V\|$	$\|E\|$	$(\|C_1\|, \|E(C_1)\|)$	$(\|C_2\|, \|E(C_2)\|)$
filmtrust	874	1853	(68, 94)	(61, 137)
email	1133	5451	(121, 345)	(127, 348)
cora	2708	5278	(205, 458)	(196, 312)
facebook	14113	52309	(114, 402)	(107, 364)

4.1 Experimental Settings

Datasets. We answer the questions above by performing network integration on real-world and synthetic datasets. Let E_i denote the edge set inside community C_i for $i \in \{1, 2\}$. Table 1 provides key statistics of the used real-world dataset: **filmtrust** (http://konect.cc/networks/librec-filmtrust-trust/) is the user-user trust network of the FilmTrust project, **email** (http://konect.cc/networks/arenas-email/) is the email communication network at the University Rovira i Virgili in Spain, **cora** (https://paperswithcode.com/dataset/cora) is the citation network of 2708 scientific publications, and **facebook** (http://networkrepository.com/fb-pages-company.php) is the "mutually-liked" network among facebook pages. We also apply the well-established *LFR model* to generate a number of synthetic networks [14] for which Table 2 lists key statistics. The parameter K of the LFR model refers to the desired average degree and μ the fraction of inter-community edges. The real-world graphs do not have a specified community structure. We therefore we choose $\mathcal{P}$ produced by the community detection algorithm. Due to space limitation, we display our results only using the Louvain method due to its prevalence use in literature. Results for different community detection algorithms will be provided in an extended version of the paper.

Table 2. The key statistics of synthetic networks generated by LFR

$\|V\|$	$\|E\|$	K	μ	$(\|C_1\|, \|E_1\|)$	$(\|C_2\|, \|E_2\|)$
100	261	5	0.8	(14, 22)	(20, 36)
500	3059	15	0.8	(78, 214)	(83, 201)
1000	6463	15	0.8	(139, 349)	(141, 372)
10000	299115	8	0.95	(944, 5776)	(931, 5664)

Benchmark Algorithms. Three edge creation strategies are chosen as benchmarks. All these algorithms were adopted in [21] where CtrPer has demonstrated effectiveness:

- **Random:** Adding edges randomly between C_1 and C_2;
- **Min-deg:** Adding edge $\{u, v\}$ if u has the minimal degree in C_1 and v has the minimal degree among all non-edge $\{u, v\} \in C_1 \otimes C_2$.
- **Max-bet:** adding edge $\{u, v\}$ if u has the maximal betweenness in C_1 and v has the maximal betweenness among all non-edge $\{u, v\} \in C_1 \otimes C_2$.
- **CtrPer:** add edges by optimizing $\forall$-togetherness in network integration.

Performance Index. Let the edge set E' be a set of added edges and $\mathcal{P}' = \{C_1', C_2', C_3', \ldots, C_{L'}'\}$ be the community structure detected on $G \oplus E'$. Denote

$$S_{\max}(\mathcal{P}') = \max \left\{ \frac{|C \cap C_y'|}{\sqrt{|C| \times |C_y'|}} \mid C_y' \in \mathcal{P}' \right\} \tag{5}$$

Then $S_{\max}(\mathcal{P}')$ measures the similarity between $C = C_1 \cup C_2$ and communities identified. A higher $S_{\max}$ value indicate better performance. We also consider F1-score which quantifiers the chance that edges in C_1 and C_2 still lie in the combined community $C = C_1 \cup C_2$ [23].

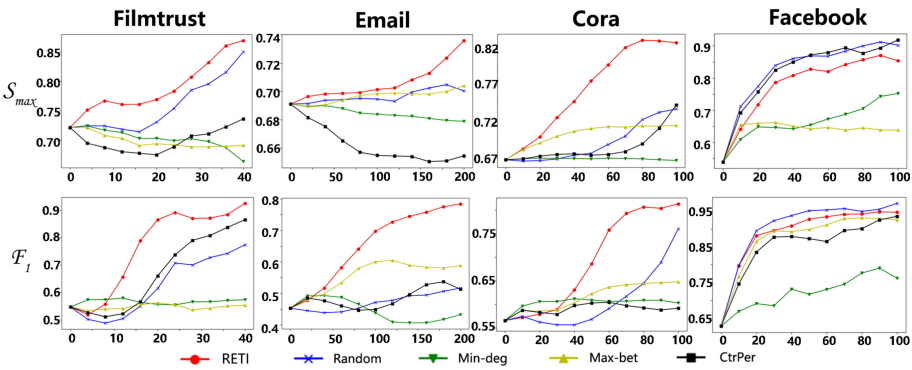

Fig. 2. Results on real world network. The horizontal axis indicates the number of new edges added to each graph. Each node is the average over 100 runs.

4.2 Results and Analysis

Togetherness Improvement. Figure 2 compares performance to togetherness improvement as more edges are added: As more edges are added, both $S_{\max}$ and F1-score increases for the TIE algorithm. TIE clearly outperforms the benchmarks w.r.t. both metrics on graphs *filmtrust*, *email* and *cora*, and comparable with the best strategy on *facebook*. On *facebook*, several algorithm all quickly achieve good performance and therefore the advantage of TIE is not clearly shown. On synthetic LFR networks with 100, 500, 1000, 10000 nodes, we add a certain ratio of edges between the communities. Table 3 lists the resulting $S_{\max}$ scores. TIE clearly outperforms benchmarks in all cases. Max-bet was not shown due to its high computation cost and lack of space.

Efficiency. Table 4 lists the running time versus number of edges added. Each value of time is the average of 50 repeated experiments. Random adding edges uses the shortest time. TIE runs in reasonable time for all cases.

Table 3. The resulting $S_{\max}$ index of network integration on LFR networks of various sizes

	TIE			Random			Min-deg			CtrPer		
LFR	10%	20%	30%	10%	20%	30%	10%	20%	30%	10%	20%	30%
100	**0.68**	**0.72**	**0.78**	0.61	0.64	0.70	0.58	0.62	0.70	0.59	0.63	0.71
500	**0.45**	**0.57**	**0.64**	0.38	0.41	0.47	0.39	0.45	0.55	0.37	0.40	0.47
1000	**0.39**	**0.48**	**0.56**	0.35	0.39	0.45	0.38	0.43	0.51	0.34	0.37	0.41
10000	**0.20**	**0.28**	**0.37**	0.19	0.19	0.19	0.19	0.23	0.27	**0.20**	0.21	0.21

Table 4. The time cost of five algorithm by adding E'

Graph	$\lvert E'\rvert$	TIE	Random	Min-deg	Max-bet	CtrPer
filmtrust	40	0.148 s	0.006 s	0.032 s	1.387 s	0.041 s
email	200	6.838 s	0.050 s	1.314 s	32.473 s	2.425 s
cora	100	2.628 s	0.042 s	0.565 s	109.29 s	0.817 s
facebook	100	2.408 s	0.349 s	0.645 s	1358.8 s	0.788 s
LFR100	50	0.033 s	0.002 s	0.005 s	0.027 s	0.007 s
LFR500	100	1.147 s	0.016 s	0.206 s	3.830 s	0.297 s
LFR1000	200	4.744 s	0.067 s	0.951 s	23.379 s	1.234 s
LFR10000	100	114.80 s	2.656 s	33.061 s	4657.9 s	51.424 s

5 Conclusion and Future Work

This paper studies the togetherness improvement problem that aims to integrate communities in a social network by employing tools from structural information theory. The integration of communities can be facilitated by structural entropy where information gain provides the key to improving togetherness. We design an efficient TIE algorithm for the task which is validated on both real-world and synthetic datasets. Future work includes extending the problem and method to directed, attributed, or weighted networks as well as considering other strategies, e.g., rewiring for network integration.

References

1. Anand, K., Bianconi, G.: Entropy measures for networks: toward an information theory of complex topologies. Phys. Rev. E **80**(4), 045102 (2009)
2. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech: Theory Exp. **2008**(10), P10008 (2008)
3. Borgatti, S.P., Everett, M.G.: A graph-theoretic perspective on centrality. Soc. Netw. **28**(4), 466–484 (2006)
4. Braunstein, S.L., Ghosh, S., Mansour, T., Severini, S., Wilson, R.C.: Some families of density matrices for which separability is easily tested. Phys. Rev. A **73**(1), 012320 (2006)
5. Brooks, F.P., Jr.: Three great challenges for half-century-old computer science. J. ACM (JACM) **50**(1), 25–26 (2003)
6. Bruhn, J.: The concept of social cohesion. In: Bruhn, J. (ed.) The Group Effect, pp. 31–48. Springer, Boston (2009). https://doi.org/10.1007/978-1-4419-0364-8_2
7. Cai, Y., Zheng, H., Liu, J., Yan, B., Su, H., Liu, Y.: Balancing the pain and gain of hobnobbing: utility-based network building over atributed social networks. In: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, pp. 193–201 (2018)
8. Chen, Q., Su, H., Liu, J., Yan, B., Zheng, H., Zhao, H.: In pursuit of social capital: upgrading social circle through edge rewiring. In: Shao, J., Yiu, M.L., Toyoda, M., Zhang, D., Wang, W., Cui, B. (eds.) APWeb-WAIM 2019. LNCS, vol. 11641, pp. 207–222. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26072-9_15
9. Chunaev, P.: Community detection in node-attributed social networks: a survey. Comput. Sci. Rev. **37**, 100286 (2020)
10. Dehmer, M.: Information processing in complex networks: graph entropy and information functionals. Appl. Math. Comput. **201**(1–2), 82–94 (2008)
11. Fortunato, S., Barthelemy, M.: Resolution limit in community detection. Proc. Natl. Acad. Sci. **104**(1), 36–41 (2007)
12. Fortunato, S., Lancichinetti, A.: Community detection algorithms: a comparative analysis: invited presentation, extended abstract. In: Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools, pp. 1–2 (2009)
13. Jiang, H., Carroll, J.M.: Social capital, social network and identity bonds: a reconceptualization. In: Proceedings of the Fourth International Conference on Communities and Technologies, pp. 51–60 (2009)

14. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. Phys. Rev. E **78**(4), 046110 (2008)
15. Li, A., Pan, Y.: Structural information and dynamical complexity of networks. IEEE Trans. Inf. Theory **62**(6), 3290–3339 (2016)
16. Liu, J., Wei, Z.: Network, popularity and social cohesion: a game-theoretic approach. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31 (2017)
17. Liu, Y., et al.: From local to global norm emergence: dissolving self-reinforcing substructures with incremental social instruments. In: International Conference on Machine Learning, pp. 6871–6881. PMLR (2021)
18. Liu, Y., Liu, J., Zhang, Z., Zhu, L., Li, A.: REM: from structural entropy to community structure deception. Adv. Neural. Inf. Process. Syst. **32**, 12938–12948 (2019)
19. Moskvina, A., Liu, J.: How to build your network? A structural analysis. arXiv preprint arXiv:1605.03644 (2016)
20. Moskvina, A., Liu, J.: Integrating networks of equipotent nodes. In: Nguyen, H.T.T., Snasel, V. (eds.) CSoNet 2016. LNCS, vol. 9795, pp. 39–50. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42345-6_4
21. Moskvina, A., Liu, J.: Togetherness: an algorithmic approach to network integration. In: 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 223–230. IEEE (2016)
22. Newman, M.E.: Modularity and community structure in networks. Proc. Natl. Acad. Sci. **103**(23), 8577–8582 (2006)
23. Shalev-Shwartz, S., Ben-David, S.: Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press, Cambridge (2014)
24. Tang, Y., Liu, J., Chen, W., Zhang, Z.: Establishing connections in a social network. In: Geng, X., Kang, B.-H. (eds.) PRICAI 2018. LNCS (LNAI), vol. 11012, pp. 1044–1057. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-97304-3_80
25. Vitak, J., Ellison, N.B., Steinfield, C.: The ties that bond: re-examining the relationship between Facebook use and bonding social capital. In: 2011 44th Hawaii International Conference on System Sciences, pp. 1–10. IEEE (2011)
26. Yan, B., Chen, Y., Liu, J.: Dynamic relationship building: exploitation versus exploration on a social network. In: Bouguettaya, A., et al. (eds.) WISE 2017. LNCS, vol. 10569, pp. 75–90. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68783-4_6
27. Yan, B., Liu, Y., Liu, J., Cai, Y., Su, H., Zheng, H.: From the periphery to the center: information brokerage in an evolving network. arXiv preprint arXiv:1805.00751 (2018)
28. Zhao, H., Su, H., Chen, Y., Liu, J., Zheng, H., Yan, B.: A reinforcement learning approach to gaining social capital with partial observation. In: Nayak, A.C., Sharma, A. (eds.) PRICAI 2019. LNCS (LNAI), vol. 11670, pp. 113–117. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29908-8_9