SIAM J. COMPUT.
Vol. 45, No. 2, pp. 197–215

DECOMPOSITIONS OF TRIANGLE-DENSE GRAPHS*

RISHI GUPTA†, TIM ROUGHGARDEN†, AND C. SESHADHRI‡

Abstract. High triangle density—the graph property stating that a constant fraction of two-hop paths belongs to a triangle—is a common signature of social networks. This paper studies triangle-dense graphs from a structural perspective. We prove constructively that significant portions of a triangle-dense graph are contained in a disjoint union of dense, radius 2 subgraphs. This result quantifies the extent to which triangle-dense graphs resemble unions of cliques. We also show that our algorithm recovers planted clusterings in approximation-stable k-median instances.

Key words. graph algorithms, social and information networks, clustering

AMS subject classifications. 05C85, 91D30

DOI. 10.1137/140955331

1. Introduction.

Can the special structure possessed by social networks be exploited algorithmically? Answering this question requires a formal definition of "social network structure." Extensive work on this topic has generated countless proposals but little consensus (see, e.g., [CF06]). The most often mentioned (and arguably most validated) statistical properties of social networks include heavy-tailed degree distributions [BA99, BrKu+00, FFF99], a high density of triangles [WS98, SaCaWiZa10, UKBM11] and other dense subgraphs or "communities" [For10, GN02, New03, New06, LLDM08], and low diameter and the small world property [Kle00a, Kle00b, Kle02, New01].

Much of the recent mathematical work on social networks has focused on the important goal of developing generative models that produce random networks with many of the above statistical properties. Well-known examples of such models include preferential attachment [BA99] and related copying models [KuRa+00], Kronecker graphs [CZF04, LeChKlFa10], and the Chung–Lu random graph model [CL02a, CL02b]. A generative model articulates a hypothesis about what "real-world" social networks look like and is directly useful for generating synthetic data. Once a particular generative model of social networks is adopted, a natural goal is to design algorithms tailored to perform well on the instances generated by the model. It can also be used as a proxy to study the effect of random processes (like edge deletions) on a network. Examples of such results include [AJB00, LiAm+08, MS10].

This paper pursues a different approach. In lieu of adopting a particular generative model for social networks, we ask, *Is there a combinatorial assumption weak enough to hold in every "reasonable" model of social networks, yet strong enough to permit useful structural and algorithmic results?*

*Received by the editors February 3, 2014; accepted for publication (in revised form) November 19, 2015; published electronically March 1, 2016. A preliminary version of this paper appeared in *Proceedings of the 5th Innovations in Theoretical Computer Science Conference*, 2014.
http://www.siam.org/journals/sicomp/45-2/95533.html

†Stanford University, Stanford, CA 94305 (rishig@cs.stanford.edu, tim@cs.stanford.edu). The first author was supported in part by the ONR PECASE award. The second author's research was supported in part by NSF awards CCF-1016885 and CCF-1215965, an AFOSR MURI grant, and an ONR PECASE award.

‡Sandia National Labs, Livermore, CA 94551 (scomand@sandia.gov). Sandia National Laboratories is a multiprogram laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Specifically, we seek structural results that apply to *every* reasonable model of social networks, including those yet to be devised.

Triangle-dense graphs. We initiate the study of *triangle-dense graphs*. Let a *wedge* be a two-hop path in an undirected graph.

DEFINITION 1 (triangle-dense graph). *The* triangle density *of an undirected graph $G = (V, E)$ is $\tau(G) := 3t(G)/w(G)$, where $t(G)$ is the number of triangles in G and $w(G)$ is the number of wedges in G (conventionally, $\tau(G) = 0$ if $w(G) = 0$). The class of* ϵ-triangle-dense graphs *consists of the graphs G with $\tau(G) \geq \epsilon$.*

Since every triangle of a graph contains three wedges, and no two triangles share a wedge, the triangle density of a graph is between 0 and 1. In the social sciences, triangle density is usually called the *transitivity* of a graph [WF94] and also the *(global) clustering coefficient*. We use the term triangle density because "transitivity" already has strong connotations in graph theory.

As an example, the triangle density of a graph is 1 if and only if it is the union of cliques. The triangle density of an Erdös–Renyi graph, drawn from $G(n, p)$, is concentrated around p. Thus, only dense Erdös–Renyi graphs have constant triangle density (as $n \to \infty$). Social networks are generally sparse and yet have remarkably high triangle density; the Facebook graph, for instance, has triangle density 0.16 [UKBM11]. Large triangle density—meaning much higher that what the edge density would suggest—is perhaps the least controversial signature of social networks (see related work below).

The class of ϵ-triangle-dense graphs becomes quite diverse as soon as ϵ is bounded below 1. For example, the complete tripartite graph is triangle dense with $\epsilon \approx \frac{1}{2}$. Every graph obtained from a bounded-degree graph by replacing each vertex with a triangle is triangle dense, where ϵ is a constant that depends on the maximum degree. Adding a clique on $n^{1/3}$ vertices to a bounded-degree n-vertex graph produces a triangle-dense graph, where again the constant ϵ depends on the maximum degree. We give a litany of examples in section 4. Can there be interesting structural or algorithmic results for this rich class of graphs?

Our results: A decomposition theorem. Our main decomposition theorem quantifies the extent to which a graph with large triangle density resembles a union of cliques. The next definition gives our notion of an "approximate union of cliques." We use $G|_S$ to denote the subgraph of a graph G induced by a subset S of vertices. Also, the *edge density* of a graph $G = (V, E)$ is $|E|/\binom{|V|}{2}$.

DEFINITION 2 (tightly knit family). *Let $\rho > 0$. A collection $V_1, V_2, \ldots, V_k$ of disjoint sets of vertices of a graph $G = (V, E)$ forms a* ρ-tightly knit family *if*

- *each subgraph $G|_{V_i}$ has both edge density and triangle density at least ρ,*
- *each subgraph $G|_{V_i}$ has radius at most 2.*

When ρ is a constant (as the graph size tends to infinity), we often refer simply to a tightly knit family. The "clusters" (i.e., the V_i's) of a tightly knit family are dense in edges and in triangles. In the context of social networks, an abundance of triangles is generally associated with meaningful social structure. There is no a priori restriction on the number of clusters in a tightly knit family.

Our main decomposition theorem states that every graph with constant triangle density contains a tightly knit family that captures a constant fraction of the graph's triangles (with the constants depending on the triangle density).

RESULT 1 (main decomposition theorem). *There exists a polynomial $f(\epsilon) = \epsilon^c$ for constant $c > 0$ such that for every ϵ-triangle-dense graph G, there exists an $f(\epsilon)$-tightly knit family that contains an $f(\epsilon)$ fraction of the triangles of G.*

We emphasize that Result 1 requires only that the input graph G has constant triangle density—beyond this property, it could be sparse or dense, low- or high-diameter, and possess an arbitrary degree distribution. Graphs without constant triangle density, such as sparse Erdös–Renyi random graphs, do not generally admit nontrivial tightly knit families (even if the triangle density requirement for each cluster is dropped).

Our proof of Result 1 is constructive. Using suitable data structures, the resulting algorithm can be implemented to run in time proportional to the number of wedges of the graph; a working C++ implementation is available from the authors. This running time is reasonable for many social networks. Our preliminary implementation of the algorithm requires a few minutes on a commodity laptop to decompose networks with millions of edges.

Note that Result 1 is nontrivial only because we require that the tightly knit family preserve the "interesting social information" of the original graph, in the form of the graph's triangles. Extracting a single low-diameter cluster rich in edges and triangles is easy—large triangle density implies that typical vertex neighborhoods have these properties. But extracting such a cluster carelessly can do more harm than good, destroying many triangles that only partially intersect the cluster. Our proof of Result 1 shows how to repeatedly extract low-diameter dense clusters while preserving at least a constant fraction of the triangles of the original graph.

A graph with constant triangle density need not contain a tightly knit family that contains a constant fraction of the graph's *edges*; see the examples in section 4. The culprit is that triangle density is a "global" condition and does not guarantee good local triangle density everywhere, allowing room for a large number of edges that are intuitively spurious. Under the stronger condition of constant local triangle density, however, we can compute a tightly knit family with a stronger guarantee.

DEFINITION 3 (Jaccard similarity). *The* Jaccard similarity *of an edge $e = (i, j)$ of a graph $G = (V, E)$ is the fraction of vertices in the neighborhood of e that participate in triangles:*

$$J_e = \frac{|N(i) \cap N(j)|}{|N(i) \cup N(j) \setminus \{i, j\}|}, \tag{1}$$

where $N(x)$ denotes the neighbors of a vertex x in G.

DEFINITION 4 (everywhere triangle dense). *A graph is* everywhere ϵ-triangle dense *if $J_e \geq \epsilon$ for every edge e, and there are no isolated vertices.*

Though useful conceptually, we would not expect graphs in practice to be everywhere triangle dense for a large value of ϵ. The following weaker definition permits graphs that have a small fraction of edges with low Jaccard similarity.

DEFINITION 5 (μ, ϵ-triangle dense). *A graph is* μ, ϵ-triangle dense *if $J_e \geq \epsilon$ for at least a μ fraction of the edges e.*

We informally refer to graphs with constant ϵ and high enough μ as *mostly everywhere triangle dense*. An everywhere ϵ-triangle-dense graph is μ, ϵ-triangle dense for every μ. An everywhere ϵ-triangle-dense graph is also ϵ-triangle dense.

The following is proved as Theorem 15.

RESULT 2 (stronger decomposition theorem). *There are polynomials $\mu(\epsilon) = \epsilon^{c_1}$ and $f(\epsilon) = \epsilon^{c_2}$ with $c_1, c_2 > 0$ such that for every $\mu(\epsilon), \epsilon$-triangle-dense graph G, there exists an $f(\epsilon)$-tightly-knit family that contains an $f(\epsilon)$-fraction of the* edges and *triangles of G.*

Applications to planted cluster models. We give an algorithmic application of our decomposition in section 5, where the tightly knit family produced by our algorithm is meaningful in its own right. We consider the approximation-stable metric k-median instances introduced by Balcan, Blum, and Gupta [BBG13]. By definition, every solution of an approximation-stable instance that has near-optimal objective function value is structurally similar to the optimal solution. They reduce their problem to clustering a certain graph with "planted" clusters corresponding to the optimal solution. We prove that our algorithm recovers a close approximation to the planted clusters, matching their guarantee.

1.1. Discussion.

Structural assumptions versus generative models. Pursuing structural results and algorithmic guarantees that assume only a combinatorial condition (namely, constant triangle density), rather than a particular model of social networks, has clear advantages and disadvantages. The class of graphs generated by a specific model will generally permit stronger structural and algorithmic guarantees than the class of graphs that share a single statistical property. On the other hand, algorithms and results tailored to a single model can lack robustness: they might not be meaningful if reality differs from the model and are less likely to translate across different application domains that require different models. Our results for triangle-dense graphs—meaning graphs with constant triangle density—are relevant for every model of social networks that generates such graphs with high probability, and we expect that all future social network models will have this property. And of course, our results can be used in any application domain that concerns triangle-dense graphs, whether motivated by social networks or not.

Beyond generality and robustness, a second reason to prefer a combinatorial assumption to a generative model is that the assumption can be easily verified for a given data set. Since computing the triangle density of a network is a well-studied problem, both theoretically and practically (see [SPK13] and the references therein), the extent to which a network meets the triangle density assumption can be quantified. By contrast, it is not clear how to argue that a network is a typical instance from a generative model, other than by verifying various statistical properties (such as triangle density). This difficulty of verification is amplified when there are multiple generative models vying for prominence, as is currently the case with social and information networks (e.g., [CF06]).

Given the prevalence of triangles in social networks, it is considered an important property for generative models to match [CF06]. Comparative studies of such generative models explicitly compared the clustering coefficients to real data and showed that classic models like the preferential attachment model, the copying model, and the stochastic kronecker model do not generate enough triangles [SaCaWiZa10, PSK12]. Recent models have explicitly tried to remedy this by creating many triangles, examples being the forest fire, block two-level Erdős–Rényi, and transitive Chung–Lu models [LKF07, SKP12, IFMN12].

Why triangle density? Social networks possess a number of statistical signatures, as discussed above. Why single out triangle density? First, there is tremendous empirical support for large triangle density in social networks. This property has been studied for decades in the social sciences [HL70, Col88, Bur04, Fau06, FWVDC10], and recently there have been numerous large-scale studies on online social networks [SaCaWiZa10, UKBM11, SPK13]. Second, in light of this empirical evidence, generative models for social and information networks are explicitly designed

to produce networks with high triangle density [WS98, CF06, SaCaWiZa10, VB12]. Third, the assumption of constant triangle density seems to impose more exploitable structure than the other most widely accepted properties of social and information networks. For example, the property of having small diameter indicates little about the structure of a network—every network can be rendered small-diameter by adding one extra vertex connected to all other vertices. Similarly, merely assuming a power-law degree distribution does not seem to impose significant restrictions on a graph [FPP06]. For example, the Chung–Lu model [CL02a] generates power-law graphs with no natural decompositions. While constant triangle density is not a strong enough assumption to exclude all "obviously unrealistic graphs," it nevertheless enables nontrivial decomposition results. Finally, we freely admit that imposing one or more combinatorial conditions other than triangle density could lead to equally interesting results, and we welcome future work along such lines. For example, recent work by Ugander, Backstrom, and Kleinberg [UBK13] suggests that constraining the frequencies of additional small subgraphs could produce a refined model of social and information networks.

Why tightly knit families? We have intentionally highlighted the existence and computation of tightly knit families in triangle dense graphs, rather than the (approximate) solution of any particular computational problem on such graphs. Our main structural result quantifies the extent to which we can "visualize" a triangle-dense graph as, approximately, a union of cliques. This is a familiar strategy for understanding restricted graph classes, analogous to using separator theorems to make precise how planar graphs resemble grids [LT79], tree decompositions to quantify how bounded-treewidth graphs resemble trees [RS86], and the regularity lemma to describe how dense graphs are approximately composed of "random-like" bipartite graphs [Sze78]. Such structural results provide a flexible foundation for future algorithmic applications. We offer a specific application to recovering planted clusterings and leave as future work the design of more applications.

2. An intuitive overview. We give an intuitive description of our proof. Our approach to finding a tightly knit family is an iterative extraction procedure. We find a single member of the family, remove this set from the graph (called the *extraction*), and repeat. Let us start with an everywhere ϵ-triangle-dense graph G and try to extract a single set S. It is easy to check that every vertex neighborhood is dense and has many triangles and would qualify as a set in a tightly knit family. But for vertex i, there may be many vertices outside $N(i)$ (the neighborhood of i) that form triangles with a single edge contained in $N(i)$. By extracting $N(i)$, we could destroy too many triangles. We give examples in section 4 where such a naïve approach fails.

Here is a simple greedy fix to the procedure. We start by adding $N(i)$ and i to the set S. If any vertex outside $N(i)$ forms many triangles with the edges in $N(i)$, we just add it to S. It is not clear that we solve our problem by adding these vertices to S, since the extraction of S could still destroy many triangles. We prove that by adding at most d_i vertices (where d_i is the degree of i) with the highest number of triangles to $N(i)$, this "destruction" can be bounded. In other words, $G|_S$ will have a high density, obviously has radius 2 (from i), and will contain a constant fraction of the triangles incident to S.

Naturally, we can simply iterate this procedure and hope to get the entire tightly knit family. But there is a catch. We crucially needed the graph to be *everywhere* ϵ-triangle dense for the previous argument. After extracting S, this need not hold. We therefore employ a *cleaning* procedure that iteratively removes edges of low Jaccard

similarity and produces an everywhere ϵ-triangle-dense graph for the next extraction. This procedure also destroys some triangles, but we can upper bound this number. As an aside, removing low Jaccard similarity edges has been used for sparsifying real-world graphs by Satuluri, Parthasarathy, and Ruan [SPR11].

When the algorithm starts with an arbitrary ϵ-triangle-dense graph G, it first cleans the graph to get an everywhere ϵ-triangle-dense graph. We may lose many edges during the initial cleaning, and this is inevitable, as examples in section 4 show. In the end, this procedure constructs a tightly knit family containing a constant fraction of the triangles of the original ϵ-triangle-dense graph.

When G is everywhere or mostly everywhere ϵ-triangle-dense, we can ensure that the tightly knit family contains a constant fraction (depending on ϵ) of the *edges* as well. Our proof is a nontrivial charging argument. By assigning an appropriate weight function to triangles and wedges, we can charge removed edges to removed triangles. This (constructively) proves the existence of a tightly knit family with a constant fraction of edges and triangles.

3. Extracting tightly knit families. In this section we walk through the proof outlined in section 2 above. We first bound the losses from the cleaning procedure in section 3.2. We then show how to extract a member of a tightly knit family from a cleaned graph in section 3.3. We combine these two procedures in Theorem 13 of section 3.4 to obtain a full tightly knit family from a triangle-dense graph. Finally, Theorem 15 of section 3.5 shows that the procedure also preserves a constant fraction of the edges in a mostly everywhere triangle-dense graph.

3.1. Preliminaries. We begin with some notation. Consider a graph $G = (V, E)$. We index vertices with $i, j, k, \ldots$ and say vertex i has degree d_i. We repeatedly deal with subgraphs H of G and use the $\ldots(H)$ notation for the respective quantities in H. So, $t(H)$ denotes the number of triangles in H, $d_i(H)$ denotes the degree of i in H, and so on. Also, if S is a set of vertices, let $G|_S$ denote the induced subgraph on G.

We conclude the preliminaries with a simple lemma on the properties of everywhere ϵ-triangle-dense graphs.

LEMMA 6. *If H is everywhere ϵ-triangle dense, then $d_i \geq \epsilon\, d_j$ for every edge (i, j). Furthermore, $N(i)$ is ϵ-edge dense for every vertex i.*

Proof. If $d_i \geq d_j$ we are done. Otherwise

$$\epsilon \leq J_{(i,j)} = \frac{|N(i) \cap N(j)|}{|(N(i) \setminus \{j\}) \cup (N(j) \setminus \{i\})|} \leq \frac{d_i - 1}{d_j - 1} \leq \frac{d_i}{d_j},$$

as desired. To prove the second statement, let $S = N(i)$. The number of edges in S is at least

$$\frac{1}{2}\sum_{j \in S} |N(i) \cap N(j)| \geq \frac{1}{2}\sum_{j \in S} J_{(i,j)}(d_i - 1) \geq \frac{\epsilon d_i(d_i - 1)}{2} = \epsilon \binom{d_i}{2}. \qquad \square$$

3.2. Cleaning a graph. An important ingredient in our constructive proof is a "cleaning" procedure that constructs an everywhere ϵ-triangle-dense graph.

DEFINITION 7. *Consider the following procedure $clean_\epsilon$ on a graph H that takes input $\epsilon \in (0, 1]$. Iteratively remove an arbitrary edge with Jaccard similarity less than ϵ, as long as such an edge exists. Finally, remove all isolated vertices. We call this* ϵ-cleaning *and denote the output by $clean_\epsilon(H)$.*

The output $clean_\epsilon(H)$ is dependent on the order in which edges are removed, but our results hold for an arbitrary removal order. Satuluri, Parthasawarthy, and Ruan [SPR11] use a more nuanced version of cleaning for graph sparsification of social networks. They provide much empirical evidence that removal of low Jaccard similarity edges does not destroy interesting graph structure, such as its dense subgraphs. Our arguments below may provide some theoretical justification.

CLAIM 8. *The number of triangles in $clean_\epsilon(H)$ is at least $t(H) - \epsilon w(H)$, where $w(H)$ is the number of wedges in H.*

Proof. The process $clean_\epsilon$ removes a sequence of edges $e_1, e_2, \ldots$. Let W_l and T_l be the set of wedges and triangles that are removed when e_l is removed. Since the Jaccard similarity of e_l *at this stage* is at most ϵ, $|T_l| \leq \epsilon(|W_l| - |T_l|) \leq \epsilon|W_l|$. All the W_l's (and T_l's) are disjoint. Hence, the total number of triangles removed is $\sum_l |T_l| \leq \epsilon \sum_l |W_l| \leq \epsilon w(H)$. ∎

We get an obvious corollary by noting that $t(H) = \tau(H) \cdot w(H)/3$.

COROLLARY 9. *The graph $clean_\epsilon(H)$ is everywhere ϵ-triangle dense and has at least $(\tau(H)/3 - \epsilon)w(H)$ triangles.*

3.3. Finding a single cluster. Suppose we have an everywhere ϵ-triangle dense graph H. We show how to remove a single cluster of a tightly knit family. Since the entire focus of this subsection is on H, we drop the $\ldots(H)$ notation.

For a set S of vertices, let t_S denote the number of triangles which have at least one vertex in S, and let $t_S^{(I)} = t(H|_S)$ denote the number of triangles which have all three vertices in S (the I is for "internal"). For $\rho \in (0, 1]$, we say that a set S is *ρ-extractable* if $H|_S$ is ρ-edge dense, ρ-triangle dense, $H|_S$ has radius 2, and $t_S^{(I)} \geq \rho t_S$. We define the following *extract* procedure that finds a single extractable cluster in the graph H.

The extraction procedure *extract*. Let i be a vertex of maximum degree. For every vertex j, let θ_j be the number of triangles incident on j whose other two vertices are in $N(i)$. Let R be the set of d_i vertices with the largest θ_j values. Output $S = \{i\} \cup N(i) \cup R$.

It is not necessary to start with a vertex of maximum degree, but doing so provides a better dependence on ϵ. (Also, strictly speaking, the $\{i\}$ above is redundant; a simple argument shows that $i \in R$.)

We start with a simple technical lemma.

LEMMA 10. *Suppose $x_1 \geq x_2 \geq \cdots > 0$ with $\sum x_j \leq \alpha$ and $\sum x_j^2 \geq \beta$. For all indices $r \leq 2\alpha^2/\beta$, $\sum_{j \leq r} x_j^2 \geq \beta^2 r/4\alpha^2$.*

Proof. If $x_{r+1} \geq \beta/2\alpha$, then $\sum_{j \leq r} x_j^2 \geq \beta^2 r/4\alpha^2$ as desired. Otherwise,

$$\sum_{j>r} x_j^2 \leq x_{r+1} \sum_j x_j \leq \beta/2.$$

Hence, $\sum_{j \leq r} x_j^2 = \sum x_j^2 - \sum_{j>r} x_j^2 \geq \beta/2 \geq \beta^2 r/4\alpha^2$, using the bound given for r. ∎

The main theorem of the section follows.

THEOREM 11. *Let H be an everywhere ϵ-triangle dense graph. The procedure extract outputs an $\Omega(\epsilon^4)$-extractable set S of vertices. Furthermore, the number of edges in $H|_S$ is an $\Omega(\epsilon)$-fraction of the edges incident to S.*

Proof. Let $\epsilon > 0$, i a vertex of maximum degree, and $N = N(i)$.

We have $|S| \leq 2d_i$. By Lemma 6, $H|_N$ has at least $\epsilon\binom{d_i}{2}$ edges, so $H|_S$ is $\Omega(\epsilon)$-edge dense. By the size of S and maximality of d_i, the number of edges in $H|_S$ is an

$\Omega(\epsilon)$-fraction of the edges incident to S. It is also easy to see that $H|_S$ has radius 2. It remains to show that $H|_S$ is $\Omega(\epsilon^4)$-triangle dense and that $t_S^{(I)} = \Omega(\epsilon^4)t_S$.

For any j, let η_j be the number of edges from j to N, and let θ_j be the number of triangles incident on j whose other two vertices are in N. Let $x_j = \sqrt{2\theta_j}$.

Lemma 10 tells us that if we can (appropriately) upper bound $\sum_j x_j$ and lower bound $\sum_j x_j^2$, then the sum of the largest few x_j^2's is significant. This implies that $H|_S$ has sufficiently many triangles. Using appropriate parameters, we show that $H|_S$ contains $\Omega(\text{poly}(\epsilon) \cdot d_i^3)$ triangles, as opposed to trivial bounds that are quadratic in d_i.

CLAIM 12. *We have $\sum_j x_j \leq \sum_{k\in N} d_k$, and $\sum_j x_j^2 \geq \frac{\epsilon}{2}\sum_{k\in N} d_k(H|_N)\ d_k$, where $d_k(H|_N)$ is the degree of vertex k within $H|_N$.*

Proof. We first upper bound $\sum_j x_j$:

$$\sum_j x_j \leq \sum_j \sqrt{2\binom{\eta_j}{2}} \leq \sum_j \eta_j = \sum_{k\in N} d_k.$$

The first inequality follows from $\theta_j \leq \binom{\eta_j}{2}$. The last equality is simply stating that the total number of edges to vertices in N is the same as the total number of edges from vertices in N.

Let t_e be the number of triangles that include the edge e. For every $e = (k_1, k_2)$, $t_e \geq J_e \cdot \max(d_{k_1} - 1, d_{k_2} - 1) \geq \epsilon \cdot \max(d_{k_1} - 1, d_{k_2} - 1)$. Since $\epsilon > 0$, each vertex is incident on at least 1 triangle. Hence all degrees are at least 2, and $d_k - 1 \geq d_k/2$ for all k. This means

$$t_e \geq \frac{\epsilon \cdot \max(d_{k_1}, d_{k_2})}{2} \geq \frac{\epsilon(d_{k_1} + d_{k_2})}{4} \qquad \text{for all } e = (k_1, k_2).$$

We can now lower bound $\sum_j x_j^2$. Abusing notation, $e \in H|_N$ refers to an edge in the induced subgraph. We have

$$\sum_j x_j^2 = \sum_j 2\theta_j = \sum_{e\in H|_N} 2t_e \geq \sum_{(k_1,k_2)\in H|_N} \frac{\epsilon}{2}(d_{k_1} + d_{k_2}) = \frac{\epsilon}{2}\sum_{k\in N} d_k(H|_N)\ d_k.$$

The two sides of the second equality are counting (twice) the number of triangles "to" and "from" the edges of N. ∎

We now use Lemma 10 with $\alpha = \sum_{k\in N} d_k$, $\beta = \frac{\epsilon}{2}\sum_{k\in N} d_k(H|_N)\, d_k$, and $r = d_i$. We first check that $r \leq 2\alpha^2/\beta$. Note that $d_i \geq d_k \geq \epsilon d_i$ for all $k \in N$, by Lemma 6 and by the maximality of d_i. Hence,

$$\frac{2\alpha^2}{\beta} = \frac{4}{\epsilon}\frac{\left(\sum_{k\in N} d_k\right)^2}{\sum_{k\in N} d_k(H|_N)\, d_k} \geq \frac{4}{\epsilon}\frac{\epsilon d_i |N| \sum_{k\in N} d_k}{d_i \sum_{k\in N} d_k} \geq 4d_i \geq r,$$

as desired. Let R be the set of $r = d_i$ vertices with the highest value of θ_j or, equivalently, with the highest value of x_j^2. By Lemma 10, $\sum_{j\in R} x_j^2 \geq \beta^2 r/4\alpha^2$, or $\sum_{j\in R} \theta_j \geq \beta^2 r/8\alpha^2$. We compute

$$\frac{\beta}{\alpha} = \frac{\epsilon}{2}\frac{\sum_{k\in N} d_k(H|_N)\ d_k}{\sum_{k\in N} d_k} \geq \frac{\epsilon}{2}\min_{k\in N} d_k(H|_N) \geq \frac{\epsilon^2 d_i}{4},$$

which gives $\sum_{j\in R} \theta_j \geq \epsilon^4 d_i^3/128$. For the first inequality above, think of the $d_k/\sum d_k$ as the coefficients in a convex combination of $d_k(H|_N)$'s. For the last inequality, $d_k(H|_N) = t_{(i,k)} \geq J_{(i,k)}(d_i - 1) \geq \epsilon d_i/2$ for all $k \in N$.

Recall $S = N \cup R$ and $|S| \leq 2d_i$. We have

$$t_S^{(I)} \geq \frac{\sum_{j \in R} \theta_j}{3} \geq \frac{\epsilon^4 d_i^3}{384},$$

since triangles contained in N get overcounted by a factor of 3. Since both t_S and the number of wedges in S are bounded above by $|S|\binom{d_i}{2} = \Theta(d_i^3)$, $H|_S$ is $\Omega(\epsilon^4)$-triangle dense, and $t_S^{(I)} = \Omega(\epsilon^4)t_S$, as desired. ∎

3.4. Getting the entire family in a triangle-dense graph. We start with a ϵ-triangle-dense graph G and explain how to get the desired entire tightly knit family. Our procedure—called the decomposition procedure—takes as input a parameter ϵ.

The decomposition procedure. Clean the graph with $clean_\epsilon$, and run the procedure *extract* to get a set S_1. Remove S_1 from the graph, run $clean_\epsilon$ again, and *extract* another set S_2. Repeat until the graph is empty. Output the sets $S_1, S_2, \ldots$.

We now prove our main theorem, Result 1, restated for convenience.

THEOREM 13. *Consider a τ-triangle dense graph G and $\epsilon \leq \tau/4$. The decomposition procedure outputs an $\Omega(\epsilon^4)$ tightly knit family with an $\Omega(\epsilon^4)$-fraction of the triangles of G.*

Proof. We are guaranteed by Theorem 11 that $G|_{S_i}$ is $\Omega(\epsilon^4)$-edge and $\Omega(\epsilon^4)$-triangle dense and has radius 2. It suffices to prove that an $\Omega(\epsilon^4)$-fraction of the triangles in G are contained in this family.

Consider the triangles that are *not* present in the tightly knit family. We call these the destroyed triangles. Such triangles fall into two categories: those destroyed in the cleaning phases and those destroyed when an extractable set is removed. Let C be the triangles destroyed during cleaning, and let D_k be the triangles destroyed in the kth extraction. By the definition of extractable subsets and Theorem 11, $t(G|_{S_k}) = \Omega(\epsilon^4|D_k|)$. Note that C, D_k, and the triangles in $G|_{S_k}$ (over all k) partition the total set of triangles. Hence, we get that $\sum_k t(G|_{S_k}) = \Omega(\epsilon^4(t - |C|))$.

We now bound $|C|$. This follows the proof of Claim 8. Let $e_1, e_2, \ldots$ be all the edges removed during cleaning phases. Let W_l and T_l be the set of wedges and triangles that are destroyed when e_l is removed. Since the Jaccard similarity of e_l at the time of removal is at most ϵ, $|T_l| \leq \epsilon(|W_l| - |T_l|) \leq \epsilon|W_l|$. All the W_ls (and T_ls) are disjoint. Hence, $|C| = \sum_l |T_l| \leq \epsilon \sum_l |W_l| = \epsilon w = 3\epsilon t/\tau \leq 3t/4$, and $\sum_k t(G|_{S_k}) = \Omega(\epsilon^4 t)$, as desired. ∎

We also give a quick runtime analysis. Recall that $w(G)$ is the number of wedges in G.

THEOREM 14. *The decomposition can be obtained in time proportional to $w(G) + |V|$.*[1]

Proof. We maintain five hash tables/sets for the course of the algorithm: a hash table from each node to its incident edges, a hash table from each edge to the degrees of its endpoints, a hash table from each edge e to the set of triangles containing e, and hash tables from integers d to the set of all vertices of degree d and the set of all edges with Jaccard similarity less than ϵ. We also keep track of the maximum degree.

We assume constant time insert, delete, and lookup in all hash tables and sets. We can initialize the data structures by enumerating over all wedges and edges, which takes $O(w(G) + |V|)$ time. When an edge e is deleted, we can update the hash tables

[1] Note that the algorithm here uses $O(w(G)+|V|)$ space as well. A slight variant of the algorithm runs in $O(w(G) + |V|)$ time and only $O(|E|)$ space but takes longer to analyze—see Appendix A of [GRS14] for details.

in time proportional to the number of wedges containing e. Since edges are deleted but never added by the decomposition procedure, we spend a total of $O(w(G) + |V|)$ time maintaining the data structures over the course of the procedure.

Now, the three operations performed by the procedure are $clean_\epsilon$, *extract*, and removing a set of nodes from the graph. The first and third are accounted for by our data structures, as is finding a vertex of maximum degree from which to *extract*.

For an *extract* operation from vertex i, we enumerate over all pairs (u, v) of neighbors of i, which takes time proportional to the number of wedges at i. For each pair that is an edge, we enumerate over all triangles that involve this pair/edge, and by hashing appropriately, we can find the d_i vertices with the largest θ_j values. Every such enumerated triangle is deleted when the extracted set is removed, so the total time spent here is again at most $O(w(G) + |E|) = O(w(G) + |V|)$, giving the desired result. □

3.5. Preserving edges in a mostly everywhere triangle-dense graph. For a mostly everywhere triangle-dense graph, the decomposition procedure can also preserve a constant fraction of the *edges*. This requires a more subtle argument. The aim of this subsection is to prove the following (cf. Result 2).

THEOREM 15. *Consider a μ, γ-triangle dense graph G, for $\mu \geq 1 - \gamma^2/32$. The decomposition procedure, with $\epsilon \leq \gamma^3/12$, outputs an $\Omega(\epsilon^4)$ tightly knit family with an $\Omega(\epsilon^4)$ fraction of the triangles of G and an $\Omega(\epsilon\gamma)$ fraction of the edges of G.*

The proof appears at the end of the subsection. The tightly knit family and triangle conditions follow directly from Theorem 13, so we focus on the edge condition. By Theorem 11, the actual removal of the clusters preserves a large enough fraction of the edges. The difficulty is in bounding the edge removals during the cleaning phases.

We first give an informal description of the argument. We would like to charge lost edges to lost triangles and piggyback on the fact that not many triangles are lost during cleaning. More specifically, we apply a weight function to triangles (and wedges), such that losing or keeping an edge corresponds to losing or keeping roughly one unit of triangle (and wedge) weight in the graph. Most edges (i, j) belong to roughly $d_i + d_j$ triangles and wedges, and so intuitively we weight each of those triangles (and wedges) by roughly $1/(d_i + d_j)$. This intuition breaks down if $d_i \ll d_j$, but $d_i \approx d_j$ for edges with high Jaccard similarity.

The rest of the argument follows the high-level plan of the ϵ-triangle dense case (cf. the argument to bound $|C|$ in Theorem 13), though work is needed to replace triangles and wedges with their weighted counterparts. The original graph G has high triangle density, which under our weight function is enough to imply a comparable amount of triangle weight and wedge weight. Only edges with low Jaccard similarity are removed during cleaning, and each of these removed edges destroys significantly more wedge weight than triangle weight. Hence, at the end of the process, a lot of triangle weight must remain. There is a tight correspondence between edges and triangle weight, and so a lot of edges must also remain.

We now start the formal proof. We use E, W, and T to denote the sets of edges, wedges, and triangles in G. W_e and T_e denote the sets of edges and triangles that include the edge e. We use E^c, W^c, and T^c to denote the respective sets destroyed during the cleaning phases, and we use W_e^c and T_e^c to denote the corresponding local versions. If an edge e is removed during cleaning, then $W_e^c \subseteq W_e$, but the sets are not necessarily equal, since elements of W_e may have been removed prior to e being cleaned. Let $T^s = T \setminus T^c$. Let E^s and V^s denote the edges and vertices, respectively, included in at least one triangle of T^s. For ease of reading, let $d_i' = d_i - 1$ be one less than the degree of vertex i.

Call an edge e *good* if $J_e \geq \gamma$ in the original graph G, and *bad* otherwise. We use g_i to denote the number of good edges incident to vertex i. Call a wedge good if it contains at least one good edge, and bad otherwise. By hypothesis, a μ fraction of edges are good. We make the following observation.

CLAIM 16. *For every good edge $e = (i, j)$, $d'_i \geq \gamma d'_j$.*

Proof. We have

$$\gamma \leq J_e = \frac{t_e}{d'_i + d'_j - t_e} \leq \frac{d'_i}{d'_j},$$

where the last inequality comes from $t_e \leq d'_i$. ∎

We now define a *weight* function r on triangles and wedges, as per the informal argument above. For a triangle $\mathcal{T} = (i_1, i_2, i_3)$ with at least two good edges, let $r(\mathcal{T}) = 1/d'_{i_1} + 1/d'_{i_2} + 1/d'_{i_3}$. If $\mathcal{T}$ has only one good edge (i_1, i_2), let $r(\mathcal{T}) = 1/d'_{i_1} + 1/d'_{i_2}$. If $\mathcal{T}$ has no good edges, let $r(\mathcal{T}) = 0$. For a good wedge w with central vertex i, let $r(w) = 1/d'_i$; otherwise, let $r(w) = 0$. Let $r(X) = \sum_{x \in X} r(x)$. Note that weights are always with respect to the degrees in the original graph G and do not change over time.

In the next two claims we show that the total triangle weight in G is comparable to the total wedge weight in G and is also comparable to $|E|$.

CLAIM 17. $r(T) \geq \gamma\mu|E|$.

Proof. Let t_i^g be the number of triangles $(i, j, k) \in T$ for which at least one of $(i, j), (i, k)$ is good. Since the good edges each have Jaccard similarity $\geq \gamma$, we have $t_i^g \geq g_i \gamma d'_i/2$. Thus,

$$r(T) = \sum_i \frac{t_i^g}{d'_i} \geq \sum_i \frac{g_i \gamma}{2} = \gamma\mu|E|. \qquad \blacksquare$$

CLAIM 18. $r(W) \leq 2\mu|E|$.

Proof. Let w_i^g be the number of good wedges which have i as their central vertex. Then

$$r(W) = \sum_i \frac{w_i^g}{d'_i} \leq \sum_i g_i = 2\mu|E|. \qquad \blacksquare$$

The next two claims bound the triangle weight lost by cleaning any particular edge.

CLAIM 19. *If a good edge $e = (i, j)$ is removed during cleaning, then $r(T_e^c) \leq (3\epsilon/\gamma^2) r(W_e^c)$.*

Proof. Assume that $d_i \geq d_j$. Let $d = d_i$. We first lower bound $r(W_e^c)$ as a function of $|W_e^c|$. For any $w \in W_e^c$, w has at least one good edge and has either i or j as its central vertex. Hence $r(w) \geq \min\{1/d'_i, 1/d'_j\} = 1/d'$, and

$$r(W_e^c) \geq \frac{|W_e^c|}{d'}.$$

We now upper bound $r(T_e^c)$ as a function of $|T_e^c|$. Consider triangle $t = (i, j, k) \in T_e^c$. If (i, j) is the only good edge in t, then $r(t) = 1/d'_i + 1/d'_j \leq 2/d'\gamma$, since $d'_j \geq d'\gamma$ by Claim 16. If t has at least two good edges, then k is at most two good edges away from i, and $d'_k \geq d'\gamma^2$. This gives $r(t) = 1/d'_i + 1/d'_j + 1/d'_k \leq 3/d'\gamma^2$. Hence

$$r(T_e^c) \leq \max\left\{\frac{3}{d'\gamma^2}, \frac{2}{d'\gamma}\right\} |T_e^c| = \frac{3}{d'\gamma^2}|T_e^c|.$$

Now, $|T_e^c| \leq \epsilon |W_e^c|$, since $J_e \leq \epsilon$ at the time of cleaning. Hence we have

$$r(T_e^c) \leq \frac{3}{d'\gamma^2}|T_e^c| \leq \frac{3\epsilon}{d'\gamma^2}|W_e^c| \leq \frac{3\epsilon}{\gamma^2}\, r(W_e^c)$$

as desired. □

CLAIM 20. *If a bad edge $e = (i, j)$ is removed during cleaning, $r(T_e^c) \leq 4/\gamma$.*

Proof. The only triangles with nonzero weight in T_e^c have a good edge to i and/or a good edge to j. Let m_i and m_j be the minimum degrees of any vertex connected by a good edge to i and j, respectively. It is not too hard to see that

$$r(T_e^c) \leq g_i \left(\frac{1}{d_i'} + \frac{1}{m_i'}\right) + g_j \left(\frac{1}{d_j'} + \frac{1}{m_j'}\right).$$

Plugging in $m_i' \geq \gamma d_i'$ (Claim 16) and $g_i \leq d_i'$ gives the desired result. □

We now combine the observations above to show that cleaning cannot remove all the triangle weight.

CLAIM 21. $r(T^s) \geq \gamma|E|/4$.

Proof. We have

$$\begin{aligned}
r(T^c) &= \sum_{\text{good } e} r(T_e^c) + \sum_{\text{bad } e} r(T_e^c) \\
&\leq \sum_{\text{good } e} \frac{3\epsilon}{\gamma^2} r(W_e^c) + \sum_{\text{bad } e} \frac{4}{\gamma} && \text{by Claim 19 and Claim 20} \\
&\leq \frac{3\epsilon}{\gamma^2} r(W) + \frac{4}{\gamma}(1-\mu)|E| \\
&\leq \frac{6\epsilon\mu|E|}{\gamma^2} + \frac{4(1-\mu)|E|}{\gamma} && \text{by Claim 18} \\
&\leq \frac{\gamma\mu|E|}{2} + \frac{\gamma|E|}{8},
\end{aligned}$$

where the last inequality follows from the bounds on ϵ and μ in the theorem statement. Hence

$$\begin{aligned}
r(T^s) &= r(T) - r(T^c) \\
&\geq \gamma\mu|E| - \left(\frac{\gamma\mu|E|}{2} + \frac{\gamma|E|}{8}\right) && \text{by Claim 17} \\
&\geq \gamma|E|/4,
\end{aligned}$$

since $\mu \geq 3/4$. □

Finally, we show that if a subgraph of G has high triangle weight, it must also have a lot of edges. Though the claim is stated in terms of T^s, the proof would hold for any $H \subset G$. This can be thought of as a moral converse to Claim 17.

CLAIM 22. $r(T^s) \leq |E^s|$.

Proof. Let $H = (V, E^s)$. The triangles of H are exactly T^s. We have

$$r(T^s) \leq \sum_{(i,j,k)\in T(H)} \frac{1}{d_i'(G)} + \frac{1}{d_j'(G)} + \frac{1}{d_k'(G)} = \sum_i \frac{t_i(H)}{d_i'(G)},$$

where $t_i(H)$ is the number of triangles in H incident to i. From here, we compute

$$\sum_i \frac{t_i(H)}{d'_i(G)} \leq \frac{\binom{d_i(H)}{2}}{d'_i(G)} \leq \sum \frac{d_i(H)}{2} = |E^s|$$

as desired. □

The last two claims together imply that the cleaning phase does not destroy too many edges. The rest of the proof is nearly identical to that of Theorem 13 from the ϵ-triangle dense case.

Proof of Theorem 15. As noted above, the tightly knit family and triangle conditions follow directly from Theorem 13.

Let D_k be the edges destroyed in the kth extraction, and let E_k be the edges in $G|_{S_k}$. By Theorem 11, $|E_k| = \Omega(\epsilon|D_k|)$. Since E^c, D_k, and E_k (over all k) partition E, we have $\sum_k |E_k| = \Omega(\epsilon(|E| - |E^c|))$. Since $|E^c| + |E^s| \leq |E|$, we have $\sum_k |E_k| = \Omega(\epsilon|E^s|)$. Finally, by Claims 21 and 22, $|E^s| = \Omega(\gamma|E|)$, and so $\sum_k |E_k| = \Omega(\epsilon\gamma|E|)$ as desired. □

4. Triangle-dense graphs: The rogues' gallery. This section provides a number of examples of graphs with constant triangle density. These examples show, in particular, that radius-1 clusters are not sufficient to capture a constant fraction of an ϵ-triangle-dense graph's triangles and that tightly knit families cannot always capture a constant fraction of an ϵ-triangle-dense graph's edges.

• *Why radius* 2*?* Consider the complete tripartite graph. This is everywhere ϵ-triangle-dense with $\epsilon \approx \frac{1}{2}$. If we removed the 1-hop neighborhood of any vertex, we would destroy a $1 - \Theta(1/n)$-fraction of the triangles. The only tightly knit family (with constant ρ) in this graph is the entire graph itself.

• *More on* 1*-hop neighborhoods.* All 1-hop neighborhoods in an everywhere triangle-dense graph are edge dense, in the sense of Lemma 6. Maybe we could just take the 1-hop neighborhoods of an independent set to get a tightly knit family? Of course, the clusters would only be edge disjoint (not vertex disjoint).

We construct a family of everywhere ϵ-triangle-dense graph with constant ϵ where this does not work. There are $m + 1$ disjoint sets of vertices, $A_1, \ldots, A_m, B$ each of size m. The graph induced on $\cup_k A_k$ is just a clique on m^2 vertices. Each vertex $b_k \in B$ is connected to all of A_k. Note that B is a maximal independent set, and the 1-hop neighborhoods of B contain $\Theta(m^4)$ triangles in total. However, the total number of triangles in the graph is $\Theta(m^6)$.

• *Why we can't preserve edges.* Result 1 only guarantees that the tightly knit family contains a constant fraction of the graph's triangles, not its edges. Consider a graph that has a clique on $n^{1/3}$ vertices and an arbitrary (or, say, a random) constant-degree graph on the remaining $n - n^{1/3}$ vertices. No tightly knit family (with constant ρ) can involve vertices outside the clique, so most of the edges must be removed. Of course, most edges in this case have low Jaccard similarity.

In general, the condition of constant triangle density is fairly weak and is met by a wide variety of graphs. The following two examples provide further intuition for this class of graphs.

• *A family of triangle-dense graphs far from a disjoint union of cliques.* Define the graph Bracelet(m, d), for m nodes of degree d, when $m > 4d/3$, as follows: Let $B_1, \ldots, B_{3m/d}$ be sets of $d/3$ vertices each put in cyclic order. Note that $3m/d \geq 4$. Connect each vertex in B_k to each vertex in B_{k-1}, B_k and B_{k+1}. Refer to Figure 1. This is an everywhere ϵ-triangle-dense d-regular graph on m vertices, with ϵ a constant

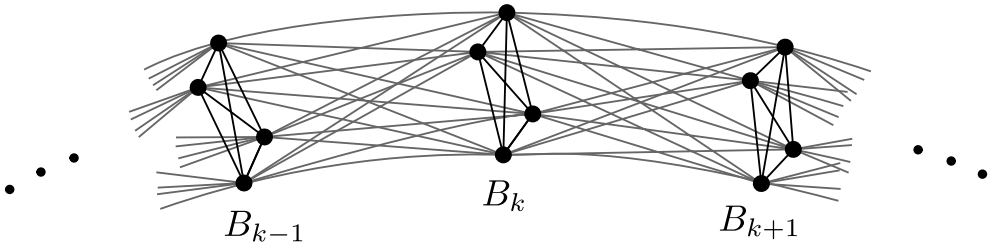

FIG. 1. *Bracelet graph with* $d/3 = 4$.

as $m \to \infty$. Nonetheless, it is maximally far (i.e., $O(md)$ edges away) from a disjoint union of cliques. A tightly knit family is obtained by taking $B_1 \cup B_2 \cup B_3$, $B_4 \cup B_5 \cup B_6$, etc.

• *Hiding a tightly knit family.* Start with $n/3$ disjoint triangles. Now, add an arbitrary bounded-degree graph (say, an expander) on these n vertices. The resulting graph has constant triangle density, but most of the structure is irrelevant for a tightly knit family.

5. Recovering a planted clustering. This section gives an algorithmic application of our decomposition procedure to recovering a "ground truth" clustering. We study the planted clustering model defined by Balcan, Blum, and Gupta [BBG13] and show that our algorithm gives guarantees similar to theirs. We do not subsume the results in [BBG13]. Rather, we observe that a graph problem that arises as a subroutine in their algorithm is essentially that of finding a tightly knit family in a triangle-dense graph. Their assumptions ensure that there is (up to minor perturbations) a unique such family.

The main setting of [BBG13] is as follows. Given a set of points V in some metric space, we wish to k-cluster them to minimize some fixed objective function, such as the k-median objective. Denote the optimal k-clustering by $\mathcal{C}$ and the value by OPT. The instance satisfies (c, ϵ)-*approximation-stability* if for any k-clustering $\mathcal{C}'$ of V with objective function value at most $c \cdot OPT$, the "classification distance" between $\mathcal{C}$ and $\mathcal{C}'$ is at most ϵ. Thus, all solutions with near-optimal objective function value must be structurally close to $\mathcal{C}$.

A summary of the argument in [BBG13] is as follows. The first step converts an approximation-stable k-median instance into an unweighted undirected graph by including an edge between two points if and only if the distance between them (in the k-median instance) is at most a judiciously chosen threshold τ. In [BBG13, Lemma 3.5] it is proved that this *threshold graph* $G = (V, E)$ contains k disjoint cliques $\{X_a\}_{a=1}^k$ such that the cliques do not have any common neighbors. These cliques correspond to clusters in the ground-truth clustering, and their existence is a consequence of the approximation stability assumption. The aim is to get a k-clustering sufficiently close to $\{X_a\}$. Formally, a k-clustering $\{S_a\}$ of V is Δ-*incorrect* if there is a permutation σ such that $\sum |X_{\sigma(a)} \setminus S_a| \leq \Delta$.

Let $B = V \setminus \bigcup_a X_a$. The second step of the argument in [BBG13] proves that when $|B|$ is small, good approximations to $\{X_a\}$ can be found efficiently. We give a different algorithm for implementing this second step; correctness follows by adapting the arguments in [BBG13]. Intuitively, the connection between our work and the setting of [BBG13] is that, when $|B|$ is much smaller than $\sum_a |X_a|$, the threshold graph G output by the first step has high triangle density. Furthermore, as we prove

below, the clusters output by the procedure *extract* of Theorem 11 are very close to the X_a's of the threshold graph.

In more detail, to obtain a k-clustering of the threshold graph G identified in [BBG13], our algorithm iteratively uses the procedure *extract* (from section 3.3) k times to get clusters $S_1, S_2, \ldots, S_k$. In particular, recall that at each step we choose a vertex s_i with the current highest degree d_i. We set N_i to be the d_i neighbors of s_i at this time and R to be the d_i vertices with the largest number of triangles to N_i. Then, $S_i = \{i\} \cup N_i \cup R$. The exact procedure of Theorem 13, which includes cleaning, also works fine. Forgoing the cleaning step does necessitate a small technical change to *extract*: instead of adding all of R to S, we add only the elements of R which have a positive number of triangles to N_i.

We use the notation $N^*(U) = N(U) \cup U$. So $N^*(X_a) \cap N^*(X_b) = \emptyset$, when $a \neq b$. Unlike [BBG13], we assume that $|X_a| \geq 3$. The following parallels the main theorem of [BBG13, Theorem 3.9], and the proof has the same high-level structure.

THEOREM 23. *The output of the clustering algorithm above is $O(|B|)$-incorrect on G.*

Proof. We first map the algorithm's clustering to the true clustering $\{X_a\}$. Our algorithm outputs k clusters, each with an associated "center" (the starting vertex). These are denoted $S_1, S_2, \ldots$, with centers $s_1, s_2, \ldots$, in order of extraction. We determine if there exists some true cluster X_a such that $s_1 \in N^*(X_a)$. If so, we map S_1 to X_a. (Recall the $N^*(X_a)$'s are disjoint, so X_a is unique if it exists.) If no X_a exists, we simply do not map S_1. We then perform this for $S_2, S_3, \ldots$, except that we do not map S_k if we would be mapping it to an X_a that has previously been mapped to. We finally end up with a subset $P \subseteq [k]$ such that for each $a \in P$, S_a is mapped to some $X_{a'}$. By relabeling the true clustering, we can assume that for all $a \in P$, S_a is mapped to X_a. The remaining clusters (for $X_{a \notin P}$) can be labeled with an arbitrary permutation of $[k] \setminus P$.

Our aim is to bound $\sum_a |X_a \setminus S_a|$ by $O(|B|)$.

We perform some simple manipulations.

$$\begin{aligned}
\bigcup_a (X_a \setminus S_a) &= \bigcup_{a \in P} (X_a \setminus S_a) \cup \bigcup_{a \notin P} (X_a \setminus S_a) \\
&= \bigcup_{a \in P} (X_a \cap \bigcup_{b<a} S_b) \cup \bigcup_{a \in P} (X_a \setminus \bigcup_{b \leq a} S_b) \cup \bigcup_{a \notin P} (X_a \setminus S_a) \\
&\subseteq \bigcup_a (X_a \cap \bigcup_{b<a} S_b) \cup \bigcup_{a \in P} (X_a \setminus \bigcup_{b \leq a} S_b) \cup \bigcup_{a \notin P} X_a.
\end{aligned}$$

So we get the following sets of interest:

- $L_1 = \bigcup_a (X_a \cap \bigcup_{b<a} S_b) = \bigcup_b (S_b \cap \bigcup_{a>b} X_a)$ is the set of vertices that are "stolen" by clusters before S_a.
- $L_2 = \bigcup_{a \in P} (X_a \setminus \bigcup_{b \leq a} S_b)$ is the set of vertices that are left behind when S_a is created.
- $L_3 = \bigcup_{a \notin P} X_a$ is the set of vertices that are never clustered.

Note that $\sum_a |X_a \setminus S_a| = |\bigcup_a (X_a \setminus S_a)| \leq |L_1| + |L_2| + |L_3|$. The proof is completed by showing that $|L_1| + |L_2| + |L_3| = O(|B|)$. This will be done through a series of claims.

We first state a useful fact.

CLAIM 24. *Suppose for some $b \in \{1, 2, \ldots, k\}$, $s_b \in N(X_b)$. Then N_b is partitioned into $N_b \cap X_b$ and $N_b \cap B$.*

Proof. Any vertex in $N_b \setminus X_b$ must be in B. This is because N_b is contained in a two-hop neighborhood from X_b, which cannot intersect any other X_a. ∎

CLAIM 25. *For any b, $|S_b \cap \bigcup_{a>b} X_a| \leq 6|S_b \cap B|$.*

Proof. We split into three cases. For convenience, let U be the set of vertices $S_b \cap \bigcup_{a>b} X_a$. Recall that $|S_b| \leq 2d_b$.

• For some c, $s_b \in X_c$. Note that $c \leq b$ by the relabeling of clusters. Observe that S_b is contained in a two-hop neighborhood of s_b and hence cannot intersect any cluster X_a for $a \neq c$. Hence, U is empty.

• For some (unique) c, $s_b \in N(X_c)$. Again, $c \leq b$. By Claim 24, $d_b = |N_b| = |N_b \cap X_c| + |N_b \cap B|$. Suppose $|N_b \cap B| \geq d_b/3$. Then $|S_b \cap B| \geq |N_b \cap B| \geq d_b/3$. We can easily bound $|S_b| \leq 2d_b \leq 6|S_b \cap B|$.

Suppose instead $|N_b \cap B| < d_b/3$, and hence $|N_b \cap X_c| > 2d_b/3$. Note that $|N_b \cap X_c|$ is a clique. Each vertex in $N_b \cap X_c$ makes $\binom{|N_b \cap X_c|-1}{2} \geq \binom{\lfloor 2d_b/3 \rfloor}{2}$ triangles in N_b. On the other hand, the only vertices of N_b that any vertex in X_a for $a \neq c$ can connect to is in $N_b \cap B$. This forms fewer than $\binom{\lfloor d_b/3 \rfloor}{2}$ triangles in N_b. If $\binom{\lfloor d_b/3 \rfloor}{2} > 0$, then $\binom{\lfloor 2d_b/3 \rfloor}{2} > \binom{\lfloor d_b/3 \rfloor}{2}$.

Consider the construction of S_b. We take the top d_b vertices with the most triangles to N_b, and say we insert them in decreasing order of this number. Note that in the modified version of the algorithm, we only insert them while this number is positive. Before any vertex of X_a ($a \neq b$) is added, all vertices of $N_b \cap X_c$ must be added. Hence, at most $d_b - |N_b \cap X_c| = |N_b \cap B| \leq |S_b \cap B|$ vertices of $\cup_{a \neq b} X_a$ can be added to S_b. Therefore, $|U| \leq |S_b \cap B|$.

• The vertex s_b is at least distance 2 from every X_c. Note that $N_b \subseteq S_b \cap B$. Hence, $|S_b| \leq 2d_b \leq 2|S_b \cap B|$. ∎

CLAIM 26. *For any $a \in P$, $|X_a \setminus \bigcup_{b \leq a} S_b| \leq |S_a \cap B|$.*

Proof. Since $a \in P$, either $s_a \in X_a$ or $s_a \in N(X_a)$. Consider the situation of the algorithm after the first $a-1$ sets $S_1, S_2, \ldots, S_{a-1}$ are removed. There is some subset of X_a that remains; call it $X'_a = X_a \setminus \bigcup_{b<a} S_b$.

Suppose $s_a \in X_a$. Since X'_a is still a clique, $X'_a \subseteq N_a$, and $(X_a \setminus \bigcup_{b \leq a} S_b)$ is empty.

Suppose instead $s_a \in N(X_a)$. Because s_a has maximum degree and X'_a is a clique, $d_a \geq |X'_a| - 1$. Note that $|X'_a \setminus S_a|$ is what we wish to bound, and $|X'_a \setminus S_a| \leq |X'_a \setminus N_a|$. By Claim 24, N_a partitions into $N_a \cap X_a = N_a \cap X'_a$ and $N_a \cap B$. We have $|X'_a \setminus N_a| = |X'_a| - |N_a \cap X_a| \leq d_a + 1 - |N_a \cap X_a| = |N_a \cap B| + 1 \leq |S_a \cap B|$. ∎

CLAIM 27. *$|L_3| \leq |B| + |L_1|$.*

Proof. Consider some X_a for $a \notin P$. Look at the situation when $S_1, \ldots, S_{a-1}$ are removed. There is a subset X'_a (forming a clique) left in the graph. All the vertices in $X_a \setminus X'_a$ are contained in L_1. By maximality of degree, $d_a \geq |X'_a| - 1$. Furthermore, since $a \notin P$, $N_a \subseteq B$ implying $d_a \leq |S_a \cap B| - 1$. Therefore, $|X'_a| \leq |S_a \cap B|$. We can bound $\bigcup_{a \notin P}(X_a \setminus X'_a) \subseteq L_1$, and $\sum_{a \notin P} |X'_a| \leq |B|$, completing the proof. ∎

To put it all together, we sum the bound of Claims 25 and 26 over $b \in [k]$ and $a \in P$, respectively, to get $|L_1| \leq 6|B|$ and $|L_2| \leq |B|$. Claim 27 with the bound on $|L_1|$ yields $|L_3| \leq 7|B|$, completing the proof of Theorem 23. ∎

6. Conclusions. This paper proposes a "model-free" approach to the analysis of social and information networks. We restrict attention to graphs that satisfy a combinatorial condition—constant triangle density—in lieu of adopting a particular generative model. The goal of this approach is to develop structural and algorithmic results that apply simultaneously to all reasonable models of social and information

networks. Our main result shows that constant triangle density already implies significant graph structure: every graph that meets this condition is, in a precise sense, well approximated by a disjoint union of clique-like graphs.

Our work suggests numerous avenues for future research.

1. Can the dependence of the intercluster edge and triangle density on the original graph's triangle density be improved?
2. The relative frequencies of four-vertex subgraphs also exhibit special patterns in social networks—for example, there are usually very few induced four-cycles [UBK13]. Is there an assumption about four-vertex induced subgraphs, in conjunction with high triangle density, that yields a stronger decomposition theorem?
3. Are there interesting additional conditions under which the decomposition into a tightly knit family is essentially unique?
4. Are stronger decomposition results possible for graphs that are also dense with larger cliques, such as 4-cliques?
5. Which computational problems are easier for triangle-dense graphs than for arbitrary graphs? Just as planar separator theorems lead to faster algorithms and better heuristics for planar graphs than for general graphs, we expect our decomposition theorem to be a useful tool in the design of algorithms for triangle-dense graphs.

Acknowledgments. We are grateful for the helpful comments provided by Jon Kleinberg, Johan Ugander, and the anonymous reviewers.

REFERENCES

[AJB00] R. Albert, H. Jeong, and A.-L. Barabási, *Error and attack tolerance of complex networks*, Nature, 406 (2000), pp. 378–382.

[BA99] A.-L. Barabasi and R. Albert, *Emergence of scaling in random networks*, Science, 286 (1999), pp. 509–512.

[BBG13] M.-F. Balcan, A. Blum, and A. Gupta, *Clustering under approximation stability*, J. ACM, 60 (2013), pp. 1068–1077.

[BrKu+00] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener, *Graph structure in the web*, Computer Networks, 33 (2000), pp. 309–320.

[Bur04] R. S. Burt, *Structural holes and good ideas*, Amer. J. Sociology, 110 (2004), pp. 349–399.

[CF06] D. Chakrabarti and C. Faloutsos, *Graph mining: Laws, generators, and algorithms*, ACM Comput. Surv., 38 (2006).

[CL02a] F. Chung and L. Lu, *The average distances in random graphs with given expected degrees*, Proc. Nat. Acad. Sci. USA, 99 (2002), pp. 15879–15882.

[CL02b] F. Chung and L. Lu, *Connected components in random graphs with given degree sequences*, Ann. Combin., 6 (2002), pp. 125–145.

[Col88] J. S. Coleman, *Social capital in the creation of human capital*, Amer. J. Sociology, 94 (1988), pp. 95–120.

[CZF04] 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, Philadelphia, 2004, pp. 442–446.

[Fau06] K. Faust, *Comparing social networks: Size, density, and local structure*, Metodoloski zvezki, 3 (2006), pp. 185–216.

[FFF99] M. Faloutsos, P. Faloutsos, and C. Faloutsos, *On power-law relationships of the internet topology*, in Proceedings of SIGCOMM, 1999, pp. 251–262.

[For10] S. Fortunato, *Community detection in graphs*, Phys. Rep., 486 (2010), pp. 75–174.

[FPP06] A. Ferrante, G. Pandurangan, and K. Park, *On the hardness of optimization in power law graphs*, in Proceedings of Conference on Computing and Combinatorics, 2006, pp. 417–427.

[FWVDC10] B. Foucault Welles, A. Van Devender, and N. Contractor, *Is a friend a friend?: Investigating the structure of friendship networks in virtual worlds*, in Extended Abstracts on Human Factors in Computing Systems, ACM, 2010, pp. 4027–4032.

[GN02] M. Girvan and M. Newman, *Community structure in social and biological networks*, Proc. Natl. Acad. Sci. USA, 99 (2002), pp. 7821–7826.

[GRS14] R. Gupta, T. Roughgarden, and C. Seshadhri, *Decompositions of triangle-dense graphs*, in Proceedings of the 5th Conference on Innovations in Theoretical Computer Science, ACM, 2014, pp. 471–482.

[HL70] P. W. Holland and S. Leinhardt, *A method for detecting structure in sociometric data*, Amer. J. Sociology, 76 (1970), pp. 492–513.

[IFMN12] J. J. Pfeiffer III, T. La Fond, S. Moreno, and J. Neville, *Fast generation of large scale social networks while incorporating transitive closures*, in Proceedings of the International Conference on Privacy, Security, Risk, and Trust (PASSAT), 2012, pp. 154–165.

[Kle00a] J. M. Kleinberg, *Navigation in a small world*, Nature, 406 (2000), 845.

[Kle00b] J. M. Kleinberg, *The small-world phenomenon: An algorithmic perspective*, in Proceedings of the Symposium on Theory of Computing, 2000, pp. 163–170.

[Kle02] J. M. Kleinberg, *Small-world phenomena and the dynamics of information*, in Advances in Neural Information Processing Systems, Vol. 1, MIT Press, Cambridge, MA, 2002, pp. 431–438.

[KuRa+00] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal, *Stochastic models for the web graph*, in Proceedings of Foundations of Computer Science, 2000, pp. 57–65.

[LiAm+08] H. Lin, C. Amanatidis, M. Sideri, R. M. Karp, and C. H. Papadimitriou, *Linked decompositions of networks and the power of choice in Polya urns*, in Proceedings of the Symposium on Discrete Algorithms, 2008, pp. 993–1002.

[LeChKlFa10] J. Leskovec, D. Chakrabarti, J. M. Kleinberg, C. Faloutsos, and Z. Ghahramani, *Kronecker graphs: An approach to modeling networks*, J. Mach. Learn. Res., 11 (2010), pp. 985–1042.

[LKF07] J. Leskovec, J. Kleinberg, and C. Faloutsos, *Graph evolution: Densification and shrinking diameters*, ACM Trans. Knowledge Discovery Data, 1 (2007).

[LLDM08] J. Leskovec, K. Lang, A. Dasgupta, and M. Mahoney, *Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters*, Internet Math., 6 (2008), pp. 29–123.

[LT79] R. J. Lipton and R. E. Tarjan, *A separator theorem for planar graphs*, SIAM J. Appl. Math., 36 (1979), pp. 177–189.

[MS10] A. Montanari and A. Saberi, *The spread of innovations in social networks*, Proc. Natl. Acad. Sci. USA, 107 (2010), pp. 20196–20201.

[New01] M. E. J. Newman, *The structure of scientific collaboration networks*, Proc. Natl. Acad. Sci. USA, 98 (2001), pp. 404–409.

[New03] M. E. J. Newman, *Properties of highly clustered networks*, Phys. Rev. E, 68 (2003), 026121.

[New06] M. E. J. Newman, *Finding community structure in networks using the eigenvectors of matrices*, Phys. Rev. E, 74 (2006), 036104.

[PSK12] A. Pinar, C. Seshadhri, and T. G. Kolda, *The similarity between Stochastic Kronecker and Chung-Lu graph models*, in Proceedings of the 2012 SIAM International Conference on Data Mining, SIAM, Philadelphia, 2012.

[RS86] N. Robertson and P. D. Seymour, *Graph minors* III: *Planar tree-width*, J. Combin. Theory Ser. B, 36 (1986), pp. 49–64.

[SaCaWiZa10] A. Sala, L. Cao, C. Wilson, R. Zablit, H. Zheng, and B. Y. Zhao, *Measurement-calibrated graph models for social network experiments*, in Proceedings of the World Wide Web Conference, ACM, 2010, pp. 861–870.

[SKP12] C. Seshadhri, T. G. Kolda, and A. Pinar, *Community structure and scale-free collections of Erdös-Rényi graphs*, Phys. Rev. E, 85 (2012), 056109.

[SPK13] C. Seshadhri, A. Pinar, and T. G. Kolda, *Fast triangle counting through wedge sampling*, in Proceedings of the 2013 SIAM International Conference on Data Mining, SIAM, Philadelphia, 2013.

[SPR11] V. Satuluri, S. Parthasarathy, and Y. Ruan, *Local graph sparsification for scalable clustering*, in Proceedings of ACM SIGMOD, 2011, pp. 721–732.

[Sze78] E. Szemerédi, *Regular partitions of graphs*, Problèmes Combinatoires théorie Graphes, 260 (1978), pp. 399–401.

[UBK13] J. Ugander, L. Backstrom, and J. Kleinberg, *Subgraph frequencies: Mapping the empirical and extremal geography of large graph collections*, in Proceedings of World Wide Web Conference, 2013, pp. 1307–1318.

[UKBM11] J. Ugander, B. Karrer, L. Backstrom, and C. Marlow, *The Anatomy of the Facebook Social Graph*, arXiv:1111.4503, 2011.

[VB12] J. Vivar and D. Banks, *Models for networks: A cross-disciplinary science*, Wiley Interdiscip. Rev. Comput. Statist., 4 (2012), pp. 13–27.

[WF94] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*, Cambridge University Press, Cambridge, UK, 1994.

[WS98] D. Watts and S. Strogatz, *Collective dynamics of 'small-world' networks*, Nature, 393 (1998), pp. 440–442.