2016 IEEE 16th International Conference on Data Mining

On Dense Subgraphs in Signed Network Streams

Jose Cadena*, Anil Kumar Vullikanti*, and Charu C. Aggarwal†
* Biocomplexity Institute and Department of Computer Science
Virginia Tech, Blacksburg, VA. {jcadena,akumar}@vbi.vt.edu
†IBM T.J. Watson Research Center, NY, 10598, USA. charu@us.ibm.com

***Abstract*—Signed networks remain relatively under explored despite the fact that many real networks are of this kind. Here, we study the problem of subgraph density in signed networks and show connections to the event detection task. Notions of density have been used in prior studies on anomaly detection, but all existing methods have been developed for unsigned networks. We develop the first algorithms for finding dense subgraphs in signed networks using semi-definite programming based rounding. We give rigorous guarantees for our algorithms, and develop a heuristic EGOSCAN which is significantly faster. We evaluate the performance of EGOSCAN for different notions of density, and observe that it performs significantly better than natural adaptations of prior algorithms for unsigned networks. In particular, the improvement in edge density over previous methods is as much as 85% and usually over 50%. These results are consistent across signed and unsigned networks in different domains. The improvement in performance is even more significant for a constrained version of the problem involving finding subgraphs containing a subset of query nodes.**

We also develop an event detection method for signed and unsigned networks based on subgraph density. We apply this to three different temporal datasets, and show that our method based on EGOSCAN significantly outperforms existing approaches and baseline methods in terms of the precision-recall tradeoff (by as much as 25-50% in some instances).

I. INTRODUCTION

Event and anomaly detection using network data are fundamental problems with applications in a large number of areas, such as computer security [1], [2], social networks [3], and finance and insurance [4]. Such settings typically involve a sequence of networks, which evolve over time. For instance, social networks often involve interactions between pairs of users at different points of time. These problems are commonly formalized using changes in different types of network features, such as distances, density, community structure and spectral properties—see [5] for a detailed discussion on these problems in different domains. Most of the previous work in this area has focused only on the simpler case of unsigned networks. However, many event detection settings require considering the more difficult case of signed networks, in which edge weights can be positive or negative, e.g., [6], [7].

A common subproblem that arises in event detection is that of dense subgraph mining, which is an important problem in its own right [8], [9], [10], [11], [12], [13]. There are many notions of density, such as the *average degree*, *edge density* and *triangle density* [13] (see Section II for definitions). The two latter ones can be computed more efficiently; however, as discussed in [13], these notions might not give the densest subgraph in real networks, and they propose a different notion called the *Optimal QuasiClique*, which does much better at finding dense networks. One limitation of all existing algorithms and notions of density is that they are restricted to unsigned networks.

In this paper, we propose methods for finding dense subgraphs in signed networks and present its application to event detection. Our contributions are summarized as follows (see Table VI for a comparison with related work).

1) *Formalizing density problems and event detection in signed networks*: We introduce the Generalized Optimal Quasiclique (GOQC) problem for dense subgraph mining in signed networks, and show that the problem is NP-complete. We also show that event detection in network streams can be naturally formalized using GOQC.
2) *Algorithms for GOQC*: We develop an algorithm called DENSDP for GOQC, using semidefinite programming (SDP) based rounding, which gives an $O(\log n)$ approximation to the optimal solution under certain conditions (Theorem 2). In practice, the approximation factor is much better. Our method is based on the approach of [14]; however, we find that a different rounding approach based on [15] performs better in practice. Furthermore, DENSDP can be easily modified to admit additional constraints of practical interest, such as finding dense subgraphs containing a specific set of query nodes. Although DENSDP runs in polynomial time, it does not scale very well to large networks. Motivated by the low diameter of dense subgraphs, we design another algorithm, EGOSCAN, with significantly faster running time, by modifying DENSDP using heuristics for pruning and scanning neighborhoods (ego-networks) of bounded size.
3) *Detection of dense subgraphs in signed and unsigned networks*: We evaluate DENSDP and EGOSCAN in more than 20 real networks and observe that they find solutions with very high density, significantly improving on adaptations of the best prior methods for both signed and unsigned networks. With a variant of EGOSCAN that optimizes the edge density instead of the GOQC score, we obtain EGOSCAN-δ, an algorithm that gives solutions with much higher edge and triangle density than all earlier methods. The improvement in edge density over previous methods is as much as 85% and usually over 50%. These results are consistent across signed and unsigned networks in different domains. The

2374-8486/16 $31.00 © 2016 IEEE
DOI 10.1109/ICDM.2016.148

improvement in performance is even more significant for the constrained version involving finding subgraphs containing a subset of query nodes.

4) *Event detection using GOQC*: We use our approach for event detection in three real datasets (ICEWS, Traffic, and Enron), for which we have suitable ground truth events (described in Section V-D). We find that our method based on EGOSCAN significantly outperforms existing approaches and baseline methods in terms of the precision-recall tradeoff (by as much as 25-50% in some instances).

II. DEFINITIONS AND PROBLEM SETTING

We assume that we have a signed network stream, which is defined as a time series of undirected signed networks $\mathcal{G} = \{G^{(1)}, G^{(2)}, \cdots, G^{(T)}\}$. The network at time t is denoted by $G^{(t)}(V, E^{(t)})$, where V is the set of nodes —this is constant across time steps— and $E^{(t)}$ is the set of edges at time t. Each edge $e = (u, v)$ in $E^{(t)}$ has a weight $w^{(t)}(u, v)$ indicating the strength of the interaction between u and v at that time step—this can be positive or negative. If e is not present at a given time instant, its weight is 0.

Our focus is on detecting *surprising* interactions between the nodes of a network compared to historical interactions. Let $\alpha^{(t)}(e)$ be the *expected weight* of the edge at time t—this is inferred from the already-observed snapshots (i.e, 1 to $t-1$). We are interested in detecting a subset of nodes of $G^{(t)}$ whose total weight is much higher than expected. We formalize this problem below.

Problem 1 (Event Detection in Signed Networks (EDSN)) *Given a signed network stream $G^{(t)}(V, E)$ and values $w^{(t)}(u, v)$ and $\alpha^{(t)}(u, v)$ for each pair of nodes (u, v), find a subset of nodes $S \subseteq V$ that maximizes*

$$f^{(t)}(S) = \sum_{u,v \in S} (w^{(t)}(u, v) - \alpha^{(t)}(u, v)). \tag{1}$$

Our approach for event detection is to solve Problem 1 for each time step t. If there exists a subset S with $f^{(t)}(S)$ above a threshold level, we say that there is an event. For a single time step, Problem 1 is a generalization of the Optimal Quasiclique (OQC) problem proposed by [12]. For a fixed time step t, we drop the superscript and denote the weight and expected weight of edge (u, v) by $w(u, v)$ and $\bar{\alpha}(u, v)$, respectively, which gives us the GOQC problem:

Problem 2 (Generalized Optimal Quasiclique (GOQC)) *Given a signed network $G(V, E)$, a weight function $w(\cdot)$ and a penalty function $\bar{\alpha}(\cdot)$, the goal is to find a subset of nodes S that maximizes $f_{\bar{\alpha}}(S) = \sum_{u,v \in S} w(u, v) - \bar{\alpha}(u, v)$.*

When we have a parameter α such that $\bar{\alpha}(u, v) = \alpha$ and $w(u, v) = 1$, for all edges $(u, v) \in E$, the above function becomes

$$\begin{aligned} f_\alpha(S) &= \sum_{u,v \in S} (w(u, v) - \alpha(u, v)) \\ &= \sum_{u,v \in S} w(u, v) - \sum_{u,v \in S} \alpha \\ &= |E[S]| - \alpha \left(\frac{|S| \cdot (|S| - 1)}{2} \right), \end{aligned}$$

where $E[S]$ denotes the edges in the subgraph induced by S. This is precisely the OQC function of [12] restricted to uniform α and edge weights of 1. We also consider a variant of the EDSN problem based on finding interactions that are either too high or too low compared to the expectation, as defined below.

Problem 3 (Event Detection in Signed Networks using Total Deviation (EDSN-TD)) *Given a network $G^{(t)}(V, E)$ and values $w^{(t)}(u, v)$ and $\alpha^{(t)}(u, v)$ for each pair of nodes (u, v), find a subset of nodes $S \subseteq V$ that maximizes*

$$f^{(t)}(S) = \left| \sum_{u,v \in S} (w^{(t)}(u, v) - \alpha^{(t)}(u, v)) \right|. \tag{2}$$

The EDSN-TD problem differs from EDSN in that it considers the absolute value of the score. This variant is useful when we are interested in activity that is either too high or too low compared to historical observations.

We use the following definitions in the rest of the paper. For a subset S, we define the *average degree* as $deg(S) = \frac{|E[S]|}{|S|}$, *density* as $\delta(S) = \frac{|E[S]|}{\binom{|S|}{2}}$, and triangle density as $\tau(S) = \frac{\text{\#triangles in } S}{\binom{|S|}{3}}$.

III. PROPOSED METHODS

We start by observing that the GOQC problem is computationally hard in general.

Theorem 1: The GOQC problem is NP-complete. Further, it is NP-hard to approximate the GOQC value within a factor of $O(n^{1/2-\epsilon})$.

Proof: We show that the k-Clique problem is polynomial-time reducible to GOQC. Let $G = (V, E)$ be an instance of the k-Clique problem. We construct an instance $(G, w, \bar{\alpha})$ of GOQC in the following manner on graph G. We define $w(e) = 2$ for all $e \in E$ and $\bar{\alpha}(u, v) = 1$ if $(u, v) \in E$, else $\bar{\alpha}(u, v) = N$, where $N > n(n-1)$. This implies that for a subset $S \subseteq V$, if there exists $u, v \in S$ with $(u, v) \notin E$, we would have $f_{\bar{\alpha}}(S) < 0$. On the other hand, if S is a clique, $w(u, v) - \bar{\alpha}(u, v) = 1$ for all $u, v \in S$. Therefore, $f_{\bar{\alpha}}(S) \geq 0$ if and only if S is a clique. Furthermore, if S is a clique, $f_{\bar{\alpha}}(S) = \binom{|S|}{2}$. Therefore, G has a clique of size at least k if and only if there is a solution in the GOQC instance $(G, w, \bar{\alpha})$ of value at least $k(k-1)/2$. This completes the proof. ∎

In contrast, we note that the complexity of the OQC problem is not known [13]. In light of this hardness, we focus on approximation algorithms.

A. Algorithm DENSDP for the GOQC problem

We start with the following quadratic programming formulation for an instance of GOQC with inputs $G = (V, E)$, w, and $\bar{\alpha}$.

$$\text{(QP) max} \quad \sum_{(u,s)\in E} w(u,s)\left(\frac{1+x_ux_0+x_sx_0+x_ux_s}{4}\right) - \sum_{u,s\in V, u\neq s} \bar{\alpha}(u,s)\left(\frac{1+x_ux_0+x_sx_0+x_ux_s}{4}\right)$$

Subject to

$$x_0, x_u \in \{-1, 1\} \qquad \text{for all } u \in V$$

Here, each variable x_u, except for x_0, corresponds to a node $u \in V$. Lemma 1 shows that the above program solves the GOQC problem.

Lemma 1: The program (QP) is equivalent to the GOQC problem.

Proof: Given a set of nodes $S \subseteq V$, we obtain a feasible solution $\mathbf{x}$ to the quadratic programming problem above by setting: $x_u = 1$ for all $u \in S$, $x_0 = 1$, and $x_v = -1$ for all $v \notin S$. We observe below that the objective value of this solution $\mathbf{x}$ equals the GOQC score for this subset. For this definition, $1 + x_ux_0 + x_sx_0 + x_ux_s$ takes the value of either 4 or 0; it has the value 4 if and only if $u \in S$. Therefore, an edge $(u, s) \in E$ contributes to the first sum if and only if $u, s \in S$; also, all the $\binom{|S|}{2}$ possible pairs of nodes in S contribute to the second sum, and, thus, the value of the objective function in the quadratic program is equivalent to $f_\alpha(S)$. The converse follows by considering the set $S = \{u : x_u = x_0\}$. ∎

We use a semidefinite relaxation of the problem:

$$\text{(SDP) max} \quad \sum_{(u,s)\in E} w(u,s)\left(\frac{1+v_uv_0+v_sv_0+v_uv_s}{4}\right) - \sum_{u,s\in V, u\neq s} \bar{\alpha}(u,s)\left(\frac{1+v_uv_0+v_sv_0+v_uv_s}{4}\right)$$

Subject to

$$v_u^T \cdot v_u = 1 \qquad \text{for all } u \in V$$
$$v_0, v_u \in \mathbb{R}^{n+1} \qquad \text{for all } u \in V$$

In this case, each v_u is an $(n + 1)$-dimensional vector constrained to have unit norm. It is easy to show that the optimal solution to this relaxation, OPT_{SDP}, is an upper bound on the optimal solution of the corresponding instance of GOQC. However, the solution to SDP is *high-dimensional* and needs to be *rounded* in order to get a solution to QP (and therefore, GOQC). We use the rounding approach of [14]. Once we obtain a subset of nodes S' from rounding, we refine this solution by using the local search algorithm proposed by [12] for OQC. We take S' and add a node u to the set if $f_{\bar{\alpha}}(S' \cup \{u\}) > f_{\bar{\alpha}}(S')$. When no more nodes can be added, we remove a node u from S' if $f_{\bar{\alpha}}(S' - \{u\}) > f_{\bar{\alpha}}(S')$. These two steps are repeated until there is no improvement in the score. Our algorithm DENSDP is summarized in Algorithm 1.

Algorithm 1 DENSDP$(G(V, E), w, \bar{\alpha})$.

Input: Signed network $(G(V, E)$, weight function w, and penalty function $\bar{\alpha})$
Output: $S \subseteq V$, a solution to GOQC

(1) SDP Step
Construct an instance of (SDP) using G, w, $\bar{\alpha}$
Solve (SDP), obtaining a vector v_u for each $u \in V$
(2) Rounding Step
Sample $r \sim \mathcal{N}(0_{(n+1)}, I_{(n+1)\times(n+1)})$
For each i, let $z_i = v_i \cdot r/T$, where $T = \sqrt{4 \log n}$.
For each i, if $|z_i| > 1$, set $y_i = z_i/|z_i|$, else $y_i = z_i$.
For each i, set $x_i = 1$ with probability $\frac{1+y_i}{2}$ and $x_i = -1$ with probability $\frac{1-y_i}{2}$
Let $S' \leftarrow \{u | x_u = x_0\}$
$S \leftarrow$ LOCALSEARCH$(G, w, \bar{\alpha}, S')$
return S

Theorem 2: If $w(\cdot)$ and $\bar{\alpha}(\cdot)$ are symmetric, and $\sum_e w(e) - \bar{\alpha}(e) \geq 0$, then the set S returned by algorithm DENSDP satisfies $f_{\bar{\alpha}}(S) = \Omega(OPT/\log n)$.

Proof: The program (SDP) is equivalent to maximizing

$$\phi(v_0, \ldots, v_n) = \sum_{e=(u,s)\in E} w(e)\left(\frac{v_uv_0+v_sv_0+v_uv_s}{4}\right) - \sum_{u,s\in V, u\neq s} \bar{\alpha}(u,s)\left(\frac{v_uv_0+v_sv_0+v_uv_s}{4}\right).$$

We assume that $V = \{1, \ldots, n\}$, and consider an $(n + 1) \times (n + 1)$-dimensional matrix $A = (a_{ij})$ defined in the following manner: (1) for all $u \in V$, $a_{0u} = a_{u0} = \sum_{v\in N(u)} w_{(u,v)}/4 - \sum_{s\in V} \bar{\alpha}(u, s)/4$, (2) for all $(u, s) \in E$, $a_{us} = (w_{(u,s)} - \bar{\alpha}(u, s))/4$, (3) for all $u, s \in V$ such that $(u, s) \notin E$, $a_{us} = -\bar{\alpha}(u, s)/4$, (4) for all $u \in V$, $a_{uu} = 0$. Then, maximizing the function $\phi(\cdot)$ above is equivalent to maximizing $\sum_{ij} a_{ij} v_i \cdot v_j$, with the matrix A having zeros on all diagonal entries—this corresponds precisely to the SDP formulation of [14]. Therefore, for $T = \sqrt{4 \log n}$, it follows that $\sum_{ij} a_{ij} x_i x_j = \Omega(OPT_{SDP}/\log n)$, where $x \in \{-1, +1\}^{n+1}$ is the vector resulting from the rounding step in Algorithm 1, and OPT_{SDP} denotes the optimum SDP objective value. Since $\sum_e w(e) - \bar{\alpha}(e) \geq 0$ and $OPT_{SDP} \geq OPT$, it follows that this solution x gives an $O(\log n)$ approximation to (QP). Finally, by Lemma 1, $f_{\bar{\alpha}}(S)$ equals the value of (QP), and the theorem follows. ∎

Remark. We note that the condition $\sum_e w(e) - \bar{\alpha}(e) \geq 0$ in Theorem 2 does not imply that the solution is trivially the entire graph. Consider a graph $G(V, E)$ with $V = \{1, 2, 3, 4, 5\}$, where $\{1, \cdots, 4\}$ form a clique while 5 is an isolated node. Also, assume a fixed $\alpha = 1/3$ for all edges. In this case, $f_\alpha(V) = 6 - (1/3)(10) = 2.6667 > 0$, but the optimal solution is given by the clique $S = \{1, 2, 3, 4\}$ and has value $f_\alpha(S) = 6 - (1/3)(6) = 4$.

B. Alternative rounding approach

An alternative rounding approach is based on [15]. The $(n + 1)$-dimensional vector r is chosen as before, so that

each component is normally distributed. We include a node u in S if and only if the corresponding vector v_u satisfies: $sgn(v_0^T \cdot r) = sgn(v_u^T \cdot r)$, where $sgn(\cdot)$ is the sign function. The algorithm DENSDP-FW is summarized in Algorithm 2, and performs much better than DENSDP in our experiments.

Algorithm 2 DENSDP-FW$(G(V,E), w, \bar{\alpha})$.

Input: Signed network $(G(V,E)$, weight function w, and penalty function $\bar{\alpha})$
Output: $S \subseteq V$, a solution to GOQC

(1) SDP Step
Construct an instance of (SDP) using G, w, and $\bar{\alpha}$
Solve (SDP), obtaining a vector v_u for each $u \in V$
(2) Rounding Step
Sample $r \sim \mathcal{N}(0_{(n+1)}, I_{(n+1)\times(n+1)})$
Let $S' \leftarrow \{u | sgn(v_0^T \cdot r) = sgn(v_u^T \cdot r)\}$
$S \leftarrow$ LOCALSEARCH$(G, w, \bar{\alpha}, S')$
return S

C. EGOSCAN: A scalable SDP-based approach

Though Theorem 2 gives a rigorous guarantee for algorithm DENSDP, and it runs in polynomial time, it does not scale very well. Using the observation that dense subgraphs typically have low diameter, we propose EGOSCAN, a local-search approach that divides a large network into subgraphs of small size for which we can run DENSDP quickly. For a node u, we define G_u^d to be the subgraph induced by u and its neighbors within d hops (i.e., the d-neighborhood of u). We also refer to this subgraph as the *ego network* of u. Given a graph G and a parameter d, our algorithm EGOSCAN iterates over all the nodes of G; for each node u, we solve GOQC on G_u^d, which is much smaller than the entire graph. Our final solution is the subgraph with highest GOQC score over all the G_u^d networks; as before, we use local search to refine the solution. EGOSCAN is described in detail in Algorithm 3. In the pseudocode, $UB(G_u^d)$ is an upper bound on the GOQC score of G_u^d, which we describe below.

Algorithm 3 EGOSCAN$(G(V,E), w, \bar{\alpha}, d)$.

Input: Signed network $(G(V,E)$, weight function w, penalty function $\bar{\alpha})$, and parameter d
Output: $S \subseteq V$, a solution to GOQC

Let $S' \leftarrow \emptyset$
for $u \in V$ **do**
  Compute G_u^d
  if $UB(G_u^d) \geq f_{\bar{\alpha}(S)}$ **then**
    Let $S_u \leftarrow$ DENSDP-FW$(G_u, w, \bar{\alpha})$
    if $f_{\bar{\alpha}}(S_u) > f_{\bar{\alpha}(S)}$ **then**
      $S' \leftarrow S_u$
    end if
  end if
end for
$S \leftarrow$ LOCALSEARCH$(G, w, \bar{\alpha}, S')$
return S

Pruning and parallelization. We can further speed up EGOSCAN by reducing the number of calls to DENSDP. Given the subgraph G_u^d for a node u, the sum of the weights of the positive edges gives an upper bound on the optimal GOQC score for G_u^d. We call this upper bound UB. In our algorithm, if the UB value for G_u^d is less than the best solution seen so far, we can discard G_u^d without any loss in quality. As we show in our experiments, pruning helps significantly in removing large portions of the search space. Moreover, each ego network can be processed independently, so that the algorithm can be parallelized easily.

IV. GOQC WITH MEMBERSHIP CONSTRAINTS

A natural extension of the GOQC problem is the *Constrained Generalized Optimal Quasi-Clique (CGOQC)*. The objective here is to find a solution that contains a specific set of query nodes.

Problem 4 (Constrained Generalized Optimal Quasiclique (CGOQC)) *Given a signed network $G(V,E)$, a weight function $w : E \to \mathbb{R}$, a penalty function $\bar{\alpha} : E \to \mathbb{R}$, and a subset of nodes $Q \subset V$, the goal is to find a subset of nodes $S \supseteq Q$ that maximizes $f_{\bar{\alpha}}(S) = \sum_{u,v \in S} w(u,v) - \bar{\alpha}(u,v)$.*

We can modify program (QP) described in section III to produce solutions containing a query set, thus obtaining an algorithm for GCOQC. For each node u in the query set, we add a constraint $x_u x_0 = 1$. Intuitively, the constraint forces the node to be in the solution returned by the quadratic program. The program (QP) can be rewritten as follows:

$$\text{Maximize} \quad \sum_{(u,s)\in E} w(u,s)\left(\frac{1 + x_u x_0 + x_s x_0 + x_u x_s}{4}\right) - \sum_{u,s\in V, u\neq s} \bar{\alpha}(u,s)\left(\frac{1 + x_u x_0 + x_s x_0 + x_u x_s}{4}\right)$$

Subject to

$$x_u x_0 = 1 \qquad \text{for all } u \in Q$$
$$x_u \in \{-1, 1\} \qquad \text{for all } u \in V$$

As before, we obtain a semidefinite relaxation to the problem, which we call C-SDP.

(C-SDP)

$$\text{Maximize} \quad \sum_{(u,s)\in E} w(u,s)\left(\frac{1 + v_u v_0 + v_s v_0 + v_u v_s}{4}\right) - \sum_{u,s\in V, u\neq s} \alpha\left(\frac{1 + v_u v_0 + v_s v_0 + v_u v_s}{4}\right)$$

Subject to

$$v_u^T \cdot v_u = 1 \qquad \text{for all } u \in V$$
$$v_u^T \cdot v_0 = 1 \qquad \text{for all } u \in Q$$
$$v_u \in \mathbb{R}^{n+1} \qquad \text{for all } u \in V$$

Our algorithm for CGOQC is analogous to Algorithm 1, but we solve (C-SDP) instead of (SDP). The rounding step remains unchanged, and we obtain the following approximation guarantee.

Theorem 3: If $w(\cdot)$ and $\bar{\alpha}(\cdot)$ are symmetric, and $\sum_e w(e) - \bar{\alpha}(e) \geq 0$, then the above algorithm finds a solution S for the CGOQC problem, satisfying $f_{\bar{\alpha}}(S) = \Omega(OPT/\log n)$ in polynomial time in n, for any given query set Q.

V. Analysis and Experiments

Our experimental analysis addresses the following questions:

1. Detection of dense subgraphs in signed and unsigned networks. Does EgoScan find dense subgraphs? Are the results consistent across signed and unsigned networks in different domains? How does EgoScan compare to existing methods?

2. Approximation guarantee in practice. How far from optimal is the DenSDP solution in real networks, compared to the worst case bound of $O(\log n)$ we prove in Theorem 2? (Section V-A).

3. Comparing DenSDP and EgoScan. How much speedup does EgoScan give us over DenSDP? How effective is the pruning in reducing the search space? (Section V-C).

4. Event Detection Performance. What is the precision-recall tradeoff of our methods for event detection in real datasets? How does it compare to existing methods? (Section V-D).

5. Constrained GOQC. What is the quality (defined below) of our SDP-based algorithm for CGOQC? How does it compare with existing methods? (Section V-F).

A. Finding Dense Subgraphs

We compare EgoScan to the greedy algorithms proposed in [12], which we call Greedy and LS below. There are no algorithms for dense subgraph mining in signed networks. Therefore, for comparison purposes, we use the local search algorithm from [12]. The local search starts at a node v that maximizes the average degree of the ego-network of v: $\frac{\sum_{u,s \in ego(v)} w(u,s)}{|ego(v)|}$, where $ego(v)$ is the set of nodes in the ego-network of v.

Table I shows our results on real-world networks from different domains, using a uniform penalty of $\alpha = 1/3$ — as justified by [12]— for all the edges in most of the graphs. Networks marked with an asterisk (*) are signed and networks marked with two asterisks (**) are signed with non-uniform α. This last set of networks and the penalties used are described in Section V-D below. We also show results for a variant of EgoScan in which the algorithm returns the subgraph with highest density instead of highest GOQC score; we call this variant EgoScan-δ.

We report the size of the subgraph found ($|S|$), its density (δ), triangle density (τ), and GOQC objective (f_α). We find that, in most cases, EgoScan discovers subgraphs of higher density and similar size compared to Greedy and LS. EgoScan-δ is able to find subgraphs with notably higher density, more than 0.70 for most networks. The difference in triangle density is even more pronounced, with EgoScan-δ achieving scores above 0.70 in most cases, even in large instances.

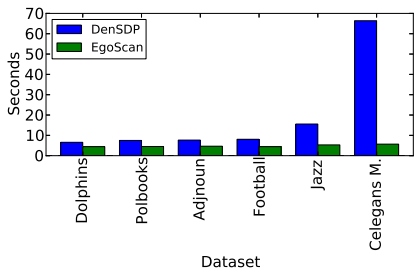

Fig. 1. Running time of DenSDP and EgoScan for various datasets. For EgoScan, we plot the average time to process the ego networks of all nodes (without considering the pruned neighborhoods). The average running time for EgoScan remains almost constant for increasing graph sizes.

B. Approximation guarantee for DenSDP in practice

We compare the objective value $f_\alpha(S_{app})$ of the solution S_{app} from DenSDP with the *fractional* SDP solution, OPT_{SDP} in Table II for some of the networks considered in Table I. This ratio $f_\alpha(S_{app})/OPT_{SDP}$ is a lower bound on the approximation quality of DenSDP, which is shown to be $\Omega(1/\log n)$ in Theorem 2. We observe that the ratios are at least 0.65 and above 0.70 for most networks, which implies that the algorithm performs better than the theoretical worst case bound.

TABLE II
The ratio $f_\alpha(S_{app})/OPT_{SDP}$, where S_{app} is the solution found by DenSDP, and OPT_{SDP} is the fractional SDP objective value. DenSDP finds solutions that have GOQC score *at least* 0.65 times the score of the optimal; in most cases, the approximation is above 0.70.

Dataset	S_{app}/S_{SDP}
Dolphins	0.70
Polbooks	0.70
Adjnoun	0.79
Football	0.65
Jazz	0.88
Celegans M.	0.82

C. Speedup from EgoScan

We compare DenSDP and EgoScan in terms of scalability. In Figure 1, we show the running time of both algorithms for different instances. In the case of EgoScan, we plot the average time to solve the semidefinite program for one ego network. We see that for different instances the average time to evaluate an ego network remains almost constant, which allows EgoScan to process larger instances much faster, especially when we add pruning and parallelization, as discussed in Section III.

We now analyze the pruning power of the UB upper bound. Table III reports the number of nodes of the networks we evaluated and the number of ego networks that are discarded. Even in the small networks, we are able to discard at least 50% of the search space for EgoScan, except for the Football

TABLE I

OPTIMAL QUASICLIQUES EXTRACTED FROM REAL NETWORKS BY GREEDY METHODS AND OUR SDP-BASED ALGORITHMS. WE REPORT THE SIZE ($|S|$) OF THE DISCOVERED SUBGRAPH, ITS DENSITY (δ), TRIANGLE DENSITY (τ), AND GOQC OBJECTIVE (f_α). EGOSCAN AND EGOSCAN-δ ARE ABLE TO FIND DENSER SUBGRAPHS THAN EXISTING GREEDY METHODS (δ). THESE RESULTS ARE CONSISTENT OVER NETWORKS OF DIFFERENT SIZES AND DIFFERENT DOMAINS. ONE ASTERISK (*) DENOTES THAT THE NETWORK IS SIGNED. TWO ASTERISKS (**) DENOTE THAT THE NETWORK IS SIGNED AND HAS DIFFERENT PENALTIES FOR EACH EDGE; OTHERWISE, THE PENALTY IS 1/3 FOR ALL EDGES.

	$\lvert S\rvert$				δ				τ				f_α			
	Greedy	LS	EgoScan	EgoScan-δ	Greedy	LS	EgoScan	EgoScan-δ	Greedy	LS	EgoScan	EgoScan-δ	Greedy	LS	EgoScan	EgoScan-δ
Dolphins	12	9	8	5	0.50	0.64	0.64	1.00	0.13	0.26	0.38	1.00	11.0	11.0	11.0	6.7
Polbooks	14	17	15	7	0.63	0.58	0.58	0.90	0.24	0.21	0.25	0.74	26.7	33.7	33.67	12.0
Adjnoun	15	12	15	4	0.50	0.58	0.48	0.83	0.13	0.20	0.13	0.50	18.0	16.0	18.0	3.0
Football	9	9	9	9	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	24.0	24.0	24.0	24.0
Jazz	57	59	50	30	0.55	0.54	0.62	1.00	0.23	0.24	0.33	1.00	348.0	351.7	354.67	290.0
Celeg. M	26	27	28	7	0.57	0.55	0.54	0.95	0.22	0.21	0.19	0.86	76.7	77.0	77.0	13.0
Wiki-Vote	132	133	130	60	0.48	0.48	0.48	0.59	0.13	0.13	0.13	0.24	1265.0	1250.0	1264.0	451.0
ca-AstroPh	57	81	98	56	1.00	0.75	0.62	1.00	1.00	0.51	0.34	1.00	1064.0	1357.0	1377.67	1026.7
AS-24july06	63	63	58	6	0.52	0.52	0.52	0.93	0.17	0.17	0.18	0.80	372.0	372.0	372.0	9.0
email-Enron	111	106	96	8	0.48	0.50	0.51	0.96	0.14	0.15	0.18	0.89	890.0	914.0	902.0	17.7
web-Google	104	66	66	17	0.48	0.85	0.85	1.00	0.22	0.64	0.64	1.00	769.7	1103.0	1103.0	90.7
AS-Skitter	318	319	276	18	0.54	0.53	0.53	0.87	0.19	0.19	0.22	0.71	10200.0	10096.0	10196.0	82.0
wikiElec.ElecBs3*	-	87	103	43	-	0.46	0.49	0.63	-	0.27	0.32	0.72	-	480.0	804.0	272.0
soc-sign-Slashdot081106*	-	144	144	23	-	0.55	0.55	0.66	-	0.43	0.43	0.64	-	2259.0	2259.0	83.67
soc-sign-Slashdot090216*	-	147	148	6	-	0.55	0.55	0.73	-	0.42	0.42	0.80	-	2329.0	2329.0	6.0
icews-countries-201407*	-	40	40	24	-	0.51	0.51	0.66	-	0.44	0.44	0.93	-	137.0	137.0	91.0
icews-countries-201412*	-	43	43	20	-	0.51	0.51	0.73	-	0.42	0.43	0.90	-	159.0	161.0	74.67
wiki-rfa-2006*	-	108	105	79	-	0.48	0.49	0.50	-	0.31	0.32	0.35	-	836.70	836.70	505.50
wiki-rfa-2012*	-	51	50	45	-	0.50	0.50	0.53	-	0.29	0.29	0.35	-	214.0	214.0	195.0
icews-brazil**	-	24	15	15	-	0.03	0.08	0.08	-	0.05	0.15	0.15	-	7.55	7.83	7.83
icews-mexico**	-	24	18	18	-	0.03	0.06	0.06	-	0.07	0.12	0.12	-	8.92	9.42	9.42
icews-venezuela**	-	17	11	11	-	0.03	0.09	0.09	-	0.07	0.19	0.19	-	4.67	5.17	5.17

network. In the larger networks (Wiki-Vote and below), we are able to prune more than 99% of the search space.

TABLE III

NUMBER OF EGO NETWORKS PRUNED USING THE UB UPPER BOUND. WITH UB, WE DISCARD MOST OF THE SEARCH SPACE. IN THE LARGER GRAPHS, WE ARE ABLE TO DISCARD MORE THAN 99% OF THE NETWORK.

Dataset	**Total nodes**	**Pruned**
Dolphins	62	31
Polbooks	105	68
Adjnoun	112	76
Football	115	15
Jazz	198	123
Celegans M.	453	391
Wiki-Vote	7,115	6,872
ca-AstroPh	18,772	18,474
AS-24july06	22,602	22,527
email-Enron	36,692	36,430
web-Google	875,713	874,159

D. Event Detection

1) Datasets: We use the EDSN and EDSN-TD problems from Section II for event detection on real-world network streams. A summary of the datasets is provided in Table IV.

ICEWS. The Integrated Crisis Early Warning System (ICEWS) [16] is a dataset of political events around the world, automatically extracted from news sources. Every entry in the dataset corresponds to an interaction between two social *actors*. An actor may be as general as a country or an ethnic group, or as specific as a particular person. Each actor falls in one of 32 classes[1] defined in the CAMEO coding convention [17]. Additionally, each interaction has an *intensity* score. The score ranges from -10 to $+10$, where negative numbers indicate increasingly hostile events, and positive numbers indicate increasingly cooperative events. We use ICEWS to detect protest events in Latin American cities. We choose three Latin American capitals: Brasilia (ICEWS Brazil), Mexico City (ICEWS Mexico), and Caracas (ICEWS Venezuela). For each city, we build signed networks where the nodes are the 32 CAMEO actor classes. In a given week, an edge $e = (u, v)$ has weight $w_e^{(t)} = c_e^{(t)}$, where $c_e^{(t)}$ is the number of events between nodes u and v. The historical weight, $\alpha_e^{(t)}$, is the average number of events per week in a recent time window: $\alpha_e^{(t)} = \sum_{i=t-W}^{t-1} \frac{c_e^{(i)}}{W}$.

For the evaluation, we use the Gold Standard Report (GSR) dataset presented in [18]. The GSR is a compilation of civil unrest events in 10 Latin American countries. This dataset also has information about which events are considered *surprising* or unexpected protests. For the evaluation, our goal is to correctly identify if there is an unexpected protest at a given week.

Traffic. We use the highway network of Los Angeles County, California[2] and its activity on May, 2014. Nodes in the graph correspond to sensors on the highway that measure the average speed and number of vehicles on the road. The sampling rate is 5 minutes, but we aggregate the data to intervals of 30 minutes. An edge in the graph represents a highway segment between two sensors. Our goal is to detect traffic congestion in this network. At time t, an edge $e = (u, v)$ has weight $w_e^{(t)} = -s_e^{(t)}$, where $s_e^{(t)}$ is be the average speed recorded by sensors u and v at time t. The historical weight for the edge is the average speed on (u, v) at the same time of the day in the last W days: $\alpha_e^{(t)} = \sum_{i \in \{t-48W, t-48(W-1), \ldots, t-48\}} -\frac{s_e^{(i)}}{W}$. Intuitively, an edge with a positive value indicates that the speed observed at the current timestamp is lower than in previous history, which is a signal of a traffic accident in the

[1]Some example of these classes are GOV (Goverment), CVL (Civilian), MED (Media)

[2]http://pems.dot.ca.gov/

corresponding highway segment.

The ground truth consists of reports emitted by the California Highway Patrol in May 2014 (collected from the PEMS website). As events of interest, we consider the subset of congestion-sensitive events corresponding to traffic collisions, car fires, hit-and-run reports, wrong-way driver incidents, and closure of roads. We ignored other types of events in the data, such as traffic hazards, which do not cause traffic congestion. Often accidents result in slow movement of traffic. By our choice of edge weights, dense subgraphs correspond to parts of the network with unexpectedly slow traffic. In a time window, there can be more than one accident at different parts of the network, so we find the top-30 densest subgraphs in each timestamp. We say that we detect an event if the reported location is within a radius of 2 miles of our dense subgraphs and occurred on the same 30 minute period. In figure 3, we show the precision-recall tradeoff for $k = 1$ to 30.

Enron. The Enron corpus[3] consists of the email directories of 151 Enron employees from May 1999 and July 2002 [19]. We consider each employee as a node in the graph; in a given week t, there is an edge between two employees if they exchanged emails during that week. The weight of the edge is the number of exchanged emails, $c_e^{(t)}$, and the penalty is given by $\alpha_e^{(t)} = \sum_{i=t-W}^{t-1} \frac{c_e^{(i)}}{W}$. For the ground truth, we use a timeline of important events related to the Enron corporation[4].

TABLE IV
DATASETS USED IN OUR EVENT DETECTION EXPERIMENTS.

Dataset	Nodes	Edges	Timestamps	Resolution
ICEWS Brazil	32	120,203	48	1 week
ICEWS Mexico	32	120,203	48	1 week
ICEWS Venezuela	32	120,203	48	1 week
Traffic	1870	2,965,584	1,488	30 minutes
Enron	151	7,444	162	1 week

2) Performance Evaluation: Our method outputs, for each timestamp, the subgraph with highest GOQC score found and the score of this subgraph. These scores induce a ranking of the likelihood of an event at each time t (i.e. higher scores indicate higher evidence for an event of interest). For our quantitative evaluation, we use the scores to generate a precision-recall curve for each dataset and compute the area under the curve of this plot –the average precision of the method.

To the best of our knowledge, the only methods for event detection in temporal networks that consider positive and negative weights are MEDEN [20] and its variant Netspot [21]. In both papers, the authors formulate the event detection task as the Heaviest Subgraph problem, which is based on Steiner connectivity instead of density. We compare our results to these methods. Additionally, to compare our results to existing methods based on density, we convert our datasets to unweighted networks and find the optimal quasiclique in each timestamp using the LocalSearchOQC algorithm of [12].

[3]https://www.cs.cmu.edu/%7Eenron/

[4]http://www.agsm.edu.au/bobm/teaching/BE/Enron/timeline.html

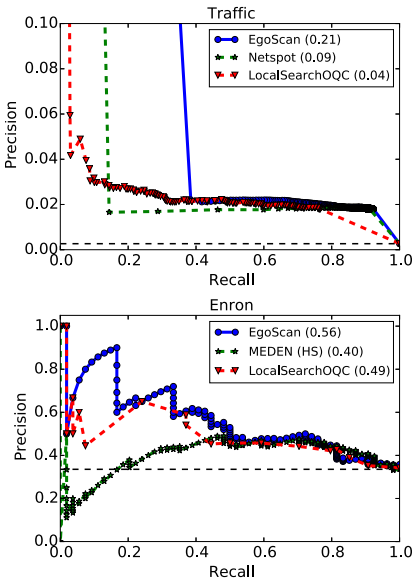

Fig. 3. **Precision-Recall tradeoff in the traffic network.** DS has better performance than the NetSpot method for temporal anomaly detection. The absolute precision and recall are low because traffic congestions can occur by events other than accidents, such as regular rush hour, sports games, concerts, etc.

Figure 2 shows the precision-recall plots for ICEWS, with the black dotted line illustrating the precision that we would obtain by flipping a fair coin (null model). For the three countries that we consider, our algorithm achieves higher precision than the other two methods at the same level of recall. Furthermore, we point that in the case of Brazil and Mexico, both MEDEN and LocalSearchOQC have performance close to the null model for most levels of recall, whereas our method shows significant precision.

Figure 3 shows the precision-recall results for the Traffic and Enron datasets. For the Traffic dataset, all the methods have low absolute recall (below 0.10). A similar result is reported by [21] for this dataset. However, our algorithm improves significantly over Netspot and LocalSearchOQC. In Enron, we achieve precision well above the null model and improve over the other two models for different levels of recall.

E. Qualitative Analysis

ICEWS. During February 2014, Venezuela was in a state of heightened civil unrest due to lack of public safety and general dissatisfaction with the central government of president Nicolas Maduro. Figure 4 shows a time series of the number of events in Caracas, Venezuela from December 15, 2013 to March 10, 2014. The dense subgraphs found by EGOSCAN are able to capture this increase in protest activity. Figure 5 shows the dense subgraph reported in the week of February 17, 2014. This week has many interactions between actors. In particular, CVL (civilians), GOV (Government), and OPP (opposition) are connected to almost every other node in the subgraph. In contrast, the same set of nodes in the previous month is disconnected.

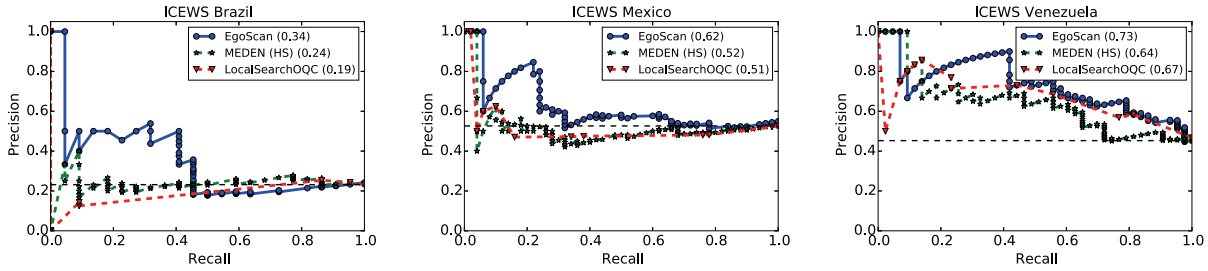

Fig. 2. **Precision-Recall plots for ICEWS.** The number in parenthesis is the area under the curve for each method. Our algorithm achieves higher precision at the same levels of recall compared to existing methods and a baseline.

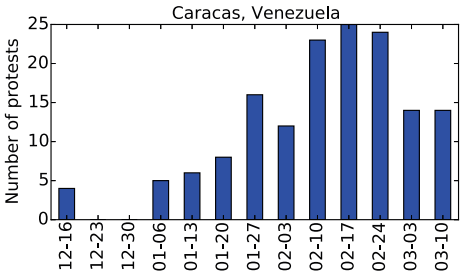

Fig. 4. Timeseries of protests in Caracas, Venezuela.

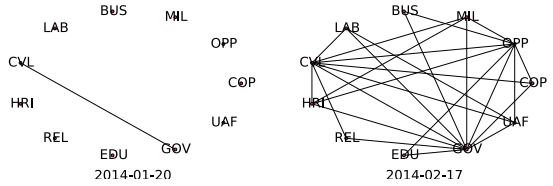

Fig. 5. Dense Subgraph for February 17, 2014 in ICEWS Venezuela (right) compared with the corresponding graph from the prior month (left). The increased activity between the actors during February is consistent with nationwide protests in Venezuela.

Enron. In August 14, 2001, Enron CEO Jeff Skilling anounced his resignation from the company. One week prior to this event, the densest subgraph reported by EGOSCAN shows increased email activity among Enron officials, including Kenneth Lay (former Chairman and successor of Skilling as CEO), Dave Delainey (Chairman and CEO after Lay), Greg Whalley (former president and COO), Louise Kitchen (president of Enron Online), Mark Haedicke (Managing Director of Enron Wholesale), and Steven Kean (VP and Chief of Staff), among others.

Traffic. We computed the number of times that each node appears in a dense subgraph reported by our algorithm. We find that the nodes that are most often reported correspond to the intersection of Freeways 710 and 105 in the highway system –near Lynwood, CA. This road is a known hotspot for truck accidents, with 5.8 accidents per mile per year reported in 2015[5].

[5]http://www.latimes.com/local/california/la-me-california-commute-20150602-story.html

F. Constrained GOQC

Lastly, we evaluate our SDP algorithm for constrained GOQC in different instances. For each instance, we generate a random query set of 5 to 7 nodes. As in Section V-A, we compare our method to the local search algorithm proposed in [12] using the same four criteria. Results are reported in table V. We notice that DENSDP finds subgraphs with notably higher GOQC score and density. The difference between the two algorithms is more pronounced in this variant compared to the unconstrained GOQC problem.

TABLE V
PERFORMANCE OF DENSDP FOR THE CONSTRAINED GOQC PROBLEM. DENSDP FINDS SUBGRAPHS WITH GOQC SCORE UP TO 2 TIMES BETTER THAN A LOCAL SEARCH ALGORITHM. THE EDGE AND TRIANGLE DENSITY ARE ALSO BETTER FOR ALL NETWORK INSTANCES.

	$\|S\|$		δ		τ		f_α	
	LS	DenSDP	LS	DenSDP	LS	DenSDP	LS	DenSDP
Dolphins	6	13	0.13	0.31	0.00	0.07	-3.0	-2.0
Polbooks	12	16	0.41	0.42	0.14	0.16	5.0	11.0
Adjnoun	15	14	0.37	0.38	0.08	0.09	4.0	4.67
Football	14	14	0.54	0,54	0.21	0.21	18.67	18.67
Jazz	61	54	0.47	0.54	0.15	0.26	241.0	300.0
Celeg. M	30	30	0.39	0.41	0.12	0.13	25.0	33.0

VI. RELATED WORK

We divide the related work into three categories: signed networks, dense subgraph mining, and graph-based event detection. Table VI shows a comparative summary of our work with existing methods and formulations.

Signed Networks. In a signed network, edge weights are positive or negative, representing friendship or conflict, respectively. [32] proposed a theory of balance in relationships using signed networks, which was formalized in terms of the structural balance theory [33]. The theory captures the colloquial notions of the "the friend of my friend is my friend" and "the enemy of my friend is my enemy", and it has been verified among tribal groups and countries [34], [35]. [36], [37] study variants of structural balance in social networks. Recent work has focused on link prediction [38], [39], [40], community detection [30], [41], and clustering [31], [7], and finding cohesive groups in a small part of a signed network [**?**]. Instead, we consider density and event detection problems, which have not been directly addressed yet (Table VI). We refer the reader to [42] for a survey on signed networks.

TABLE VI
COMPARATIVE SUMMARY OF THE RELATED WORK

Category	Method	Static	Temporal	Event Detection	Dense Subgraph	Signed	Unsigned	Weighted	Theoretical guarantees	Scalable
Dense Subgraph Mining	Densest Subgraph [22], [11], [9]	✓	✗	✗	✓	✗	✓	✓	✓	✓
	Densest-k Subgraph [10], [15]	✓	✗	✗	✓	✗	✓	✓	✓	✗
	Densest At-Least (At-Most) k subgraph [8]	✓	✗	✗	✓	✗	✓	✓	✓	✓
	Optimal Quasiclique [23], [13]	✓	✗	✗	✓	✗	✓	✓	✗	✓
Graph-based Event Detection	OddBall [24]	✓	✗	✓	✓	✗	✓	✓	✗	✓
	GraphScope, COM2 [25], [26]	✓	✓	✓	✓	✗	✓	✓	✗	✓
	Meden, NetSpot [20], [21]	✓	✓	✓	✗	✓	✓	✓	✗	✓
	Graph Scan Statistics [27], [28], [29]	✓	✓	✓	✗	✗	✓	✓	✗	✓
Signed Networks	Community Detection [30], [6]	✓	✗	✗	✗	✓	✗	✗	✗	✓
	Spectral Clustering [31]	✓	✗	✗	✓	✓	✓	✓	✗	✓
	Low-Rank Modeling [7]	✓	✗	✗	✗	✓	✗	✗	✓	✓
Our contributions	DenSDP	✓	✓	✓	✓	✓	✓	✓	✓	✗
	EgoScan	✓	✓	✓	✓	✓	✓	✓	✗	✓

Dense Subgraph Mining. We briefly touch upon them existing formulations for finding dense subgraphs. We refer the reader to [12], [13] for more details.

In the *Densest Subgraph* problem, we are given a graph $G(V, E)$, and the goal is to find a subset of nodes $S \subseteq V$, such that the *average degree* of the graph induced by S, $\frac{E(S)}{|S|}$ is maximized. This problem can be solved in polynomial time using Goldberg's flow-based algorithm [22]. There is also a linear-time greedy algorithm that yields a $\frac{1}{2}$-approximation [11], [9]; in practice, this algorithm finds subgraph with average degree close to optimal. In real-world networks, subgraphs with maximal average degree have been found to be large —sometimes trivially spanning the entire node set V— and not very dense [12]. When we want to control the size of the subgraphs discovered, we can add a constraint k to the densest subgraph formulation. In the *Densest-k Subgraph* problem [43], the goal is to find a subset S of size k with maximum number of edges. This problem is NP-Hard. Asahiro et al. propose a $O(k/n)$ greedy approximation algorithm for any value of k [10]; for particular values of k, Feige and Langberg are able to obtain better approximations using semidefinite programming [15]. When the constraint is to find a set S of size at least k, we obtain the *Densest-at-least-k Subgraph* problem; when the constraint is $|S| \leq k$, we have the *Densest-at-most-k Subgraph* problem. Both variants (also NP-Hard) were proposed by Andersen and Chellapilla [8]. Recently, [13] proposed the k-*Clique* densest subgraph as an extension to the classical *densest subgraph* problem. In this formulation, the goal is to find a set of nodes that maximizes $\frac{G_k(S)}{|S|}$, where G_k is the number of k-cliques induced by the nodes in S. For $k = 3$, we obtain the *triangle-densest subgraph* problem; the authors show this latter formulation discovers graphs that are denser than the ones found by maximizing the average degree.

Dense subgraphs also have connections with cliques; a k-clique is the densest graph of size k. Then, an alternative way to find dense subgraphs is to look for large cliques. However, there are two big challenges with this approach. First, the clique problem cannot be approximated to a constant factor, unless P = NP. Second, the clique definition is too restrictive because all of the edges have to be present. In [12], Tsourakakis et al. proposed to the *Optimal Quasiclique* problem, where the goal is to find a set of nodes that maximizes $f_\alpha(S) = E(S) - \alpha\binom{|S|}{2}$, where α is a parameter. The authors propose a greedy method based on [11] and a local search algorithm to optimize the objective. They also show that subgraphs with a high f_α score have high edge and triangle density, and have small diameter —all of these are desirable properties for dense subgraphs.

Despite the extensive literature in graph density, the problem of finding dense subgraphs in signed networks has not been explored yet. As we discuss later, signed networks bring a new set of challenges when it comes to density problems; the current methods and formulations assume the weights of the graph are non-negative, and there is no simple way to extend these methods to the signed case.

Graph-based Event Detection. [5] gives a good survey on graph-based anomaly and event detection. They classify the different approaches in dynamic graphs depending on the graph characteristic that is used and the kind of events that are detected. One line of work formalizes anomalous patterns in the graph within a time window, which is used for identifying anomalies in the entire stream [20], [21]. Information theoretic approaches have also been proposed [25], [26]. In these formulations, anomalies are subgraphs that have high encoding cost. A related approach examines changes in community structure, e.g., [44], [3]. However, as discussed in [42], there has been limited work on event detection in signed networks.

VII. CONCLUSIONS

We propose the GOQC problem as the first formalization of dense subgraphs in signed networks. We also find an interesting connection between the GOQC problem and event

detection in signed networks. This connection leads to a window-based method for event detection for both signed and unsigned network streams. We also develop the first efficient methods with rigorous approximation guarantees and good empirical performance for the GOQC problem. Our results show that semidefinite programming based methods are able to find dense subgraphs in many different domains. These methods can be scaled up to achieve practical algorithms by the heuristics that we develop here.

Acknowledgements. Charu Aggarwal's research was sponsored in part by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The work of Jose Cadena and Anil Vullikanti has been partially supported by the following grants: DTRA CNIMS Contract HDTRA1-11-D-0016-0010, NSF BIG DATA Grant IIS-1633028 and NSF DIBBS Grant ACI-1443054.

REFERENCES

[1] W. Eberle and L. Holder, "Graph-based approaches to insider threat detection," in *Proc. of CSIIRW*, 2009.

[2] Q. Ding, N. Katenka, P. Barford, E. D. Kolaczyk, and M. Crovella, "Intrusion as (anti)social communication: characterization and detection," in *Proc. of the 18th ACM SIGKDD*, 2012, pp. 886–894.

[3] C. Aggarwal, Y. Zhao, and P. Yu, "Outlier detection in graph streams," in *ICDE*, 2011.

[4] M. Kumar *et al.*, "Data mining to predict and prevent errors in health insurance claims processing," in *Proc. of ACM SIGKDD*, 2010, pp. 65–74.

[5] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: a survey," *Data Mining and Knowledge Discovery*, 2014.

[6] P. Bogdanov, N. D. Larusso, and A. Singh, "Towards community discovery in signed collaborative interaction networks," in *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*. IEEE, 2010, pp. 288–295.

[7] K.-Y. Chiang *et al.*, "Prediction and clustering in signed networks: a local to global perspective," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1177–1213, 2014.

[8] R. Andersen and K. Chellapilla, "Finding dense subgraphs with size bounds," in *Algorithms and Models for the Web-Graph*. Springer, 2009, pp. 25–37.

[9] Y. Asahiro, R. Hassin, and K. Iwama, "Complexity of finding dense subgraphs," *Discrete Applied Mathematics*, vol. 121, no. 1, pp. 15–26, 2002.

[10] Y. Asahiro, K. Iwama, H. Tamaki, and T. Tokuyama, "Greedily finding a dense subgraph," *Journal of Algorithms*, vol. 34, no. 2, pp. 203–221, 2000.

[11] M. Charikar, "Greedy approximation algorithms for finding dense components in a graph," in *APPROX*, 2000.

[12] C. Tsourakakis *et al.*, "Denser than the densest subgraph: extracting optimal quasi-cliques with quality guarantees," in *Proc. of ACM SIGKDD*, 2013, pp. 104–112.

[13] C. Tsourakakis, "The k-clique densest subgraph problem," in *Proc. of WWW*, 2015, pp. 1122–1132.

[14] M. Charikar and A. Wirth, "Maximizing quadratic programs: extending grothendick's inequality," in *IEEE FOCS*, 2004.

[15] U. Feige and M. Langberg, "Approximation algorithms for maximization problems arising in graph partitioning," *J. Algorithms*, 2001.

[16] D. J. Gerner *et al.*, "Machine coding of event data using regional and international sources," *Inter. Studies Quarterly*, pp. 91–119, 1994.

[17] ——, "Conflict and mediation event observations (cameo): A new event data framework for the analysis of foreign policy interactions," *International Studies Association*, 2002.

[18] N. Ramakrishnan *et al.*, "'beating the news' with embers: Forecasting civil unrest using open source indicators," in *Proc. of ACM SIGKDD*, 2014, pp. 1799–1808.

[19] B. Klimt and Y. Yang, "Introducing the enron corpus." in *CEAS*, 2004.

[20] P. Bogdanov, M. Mongiovì, and A. Singh, "Mining heavy subgraphs in time-evolving networks," in *ICDM*, 2011.

[21] M. Mongiovı *et al.*, "Netspot: Spotting significant anomalous regions on dynamic networks," in *Proc. of SDM*, 2013.

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

[23] J. Abello, M. G. Resende, and S. Sudarsky, "Massive quasi-clique detection," in *LATIN 2002: Theoretical Informatics*. Springer, 2002, pp. 598–612.

[24] L. Akoglu, M. McGlohon, and C. Faloutsos, "Oddball: Spotting anomalies in weighted graphs," in *Advances in Knowledge Discovery and Data Mining*. Springer, 2010, pp. 410–421.

[25] J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu, "Graphscope: parameter-free mining of large time-evolving graphs," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2007, pp. 687–696.

[26] M. Araujo, S. Papadimitriou, S. Günnemann, C. Faloutsos, P. Basu, A. Swami, E. E. Papalexakis, and D. Koutra, "Com2: fast automatic discovery of temporal (?comet?) communities," in *Advances in Knowledge Discovery and Data Mining*. Springer, 2014, pp. 271–283.

[27] C. E. Priebe, J. M. Conroy, D. J. Marchette, and Y. Park, "Scan statistics on enron graphs," *Computational & Mathematical Organization Theory*, vol. 11, no. 3, pp. 229–247, 2005.

[28] S. Speakman, Y. Zhang, and D. B. Neill, "Dynamic pattern detection with temporal consistency and connectivity constraints," in *Data Mining (ICDM), 2013 IEEE 13th International Conference on*. IEEE, 2013, pp. 697–706.

[29] F. Chen and D. B. Neill, "Non-parametric scan statistics for event detection and forecasting in heterogeneous social media graphs," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 1166–1175.

[30] V. Traag and J. Bruggeman, "Community detection in networks with positive and negative links," *Physical Review E*, vol. 80, no. 036115, pp. 1–6, 2009.

[31] J. Kunegis, S. Schmidt, A. Lommatzsch, J. Lerner, E. W. De Luca, and S. Albayrak, "Spectral analysis of signed graphs for clustering, prediction and visualization." in *SDM*, vol. 10. SIAM, 2010, pp. 559–559.

[32] F. Heider, "Attitudes and cognitive organization," *The Journal of psychology*, vol. 21, no. 1, pp. 107–112, 1946.

[33] D. Cartwright and F. Harary, "Structural balance: a generalization of heider's theory." *Psychological review*, vol. 63, no. 5, p. 277, 1956.

[34] K. E. Read, "Cultures of the central highlands, new guinea," *Southwestern Journal of Anthropology*, pp. 1–43, 1954.

[35] T. Antal, P. L. Krapivsky, and S. Redner, "Social balance on networks: The dynamics of friendship and enmity," *Physica D: Nonlinear Phenomena*, vol. 224, no. 1, pp. 130–136, 2006.

[36] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Predicting positive and negative links in online social networks," in *Proc. of WWW*, 2010, pp. 641–650.

[37] ——, "Signed networks in social media," in *Proc. of SIGCHI*, 2010, pp. 1361–1370.

[38] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American society for information science and technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[39] C.-J. Hsieh *et al.*, "Low rank modeling of signed networks," in *Proc. of ACM SIGKDD*, 2012, pp. 507–515.

[40] J. Tang, S. Chang, C. Aggarwal, and H. Liu, "Negative link prediction in social media," in *Proc. of ACM WSDM*, 2015, pp. 87–96.

[41] L. Tang and H. Liu, "Community detection and mining in social media," *Synthesis Lectures on Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 1–137, 2010.

[42] J. Tang, Y. Chang, C. C. Aggarwal, and H. Liu, "A survey of signed network mining in social media," *CoRR*, vol. abs/1511.07569, 2015.

[43] U. Feige, M. Seltser *et al.*, *On the densest k-subgraph problem*. Citeseer, 1997.

[44] L. Peel and A. Clauset, "Detecting change points in the large-scale structure of evolving networks," 2014, coRR, abs/1403.0989.