Using Strong Triadic Closure to Characterize Ties in Social Networks

Stavros Sintos
Department of Computer Science and Engineering
University of Ioannina
Ioannina, Greece
cs101919@cs.uoi.gr

Panayiotis Tsaparas
Department of Computer Science and Engineering
University of Ioannina
Ioannina, Greece
tsap@cs.uoi.gr

ABSTRACT

In the past few years there has been an explosion of social networks in the online world. Users flock these networks, creating profiles and linking themselves to other individuals. Connecting online has a small cost compared to the physical world, leading to a proliferation of connections, many of which carry little value or importance. Understanding the strength and nature of these relationships is paramount to anyone interesting in making use of the online social network data. In this paper, we use the principle of Strong Triadic Closure to characterize the strength of relationships in social networks. The Strong Triadic Closure principle stipulates that it is not possible for two individuals to have a strong relationship with a common friend and not know each other. We consider the problem of labeling the ties of a social network as strong or weak so as to enforce the Strong Triadic Closure property. We formulate the problem as a novel combinatorial optimization problem, and we study it theoretically. Although the problem is NP-hard, we are able to identify cases where there exist efficient algorithms with provable approximation guarantees. We perform experiments on real data, and we show that there is a correlation between the labeling we obtain and empirical metrics of tie strength, and that weak edges act as bridges between different communities in the network. Finally, we study extensions and variations of our problem both theoretically and experimentally.

Categories and Subject Descriptors

J.4 [**Computer Applications**]: Social and behavioral sciences; H.2.8 [**Database Applications**]: Data Mining; H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

Strong Triadic Closure, Social Networks, Approximation Algorithms

KDD'14, August 24–27, 2014, New York, NY, USA.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2956-9/14/08 ...$15.00.
http://dx.doi.org/10.1145/2623330.2623664.

1. INTRODUCTION

The past few years have been marked by the emergence and explosive growth of online social networks. Facebook, LinkedIn, and Twitter are three prominent examples of such online networks, which have become extremely popular, engaging hundreds of millions of users all over the world. Online social networks grow much faster than physical social networks since the "cost" of creating and maintaining connections is much lower. The average user in Facebook has a few hundreds of friends, and a sizeable fraction of the network has more than a thousands friends [20]. The social circle of the average user contains connections with true friends, but also with forgotten high-school classmates, distant relatives, and acquaintances made through brief encounters. Many of the online connections correspond to weak, or no relationships in the physical world.

Understanding the strength and nature of online relationships is paramount to anyone interested in extracting some utility out of the online social network data. For monetization purposes, knowing which relationships correspond to true friendships is of critical importance to advertisers who want to profile users based of their social circle and initiate viral marketing campaigns. For sociologists, knowing the relative importance of online relationships can have a significant effect on the way they model and interpret dynamics and norms in the social network. For friendship suggestion algorithms, knowing which friends matter more can have an important impact on the produced link recommendations.

The problem of understanding the strength and nature of social ties has been studied in the past [12, 7, 6, 22]. Previous approaches rely on user characteristics in order to estimate the true affinity between two users. In this work we use solely the graph structure in order to derive the characterization of the ties within a social network. To this end we make use of the *Strong Triadic Closure* (STC) principle [4]. The STC principle has its roots into early works in Psychology [3, 15, 8], and it has been used in the study of social networks [4, 8]. Informally, the STC principle assumes that there are two types of ties, strong and weak, and it stipulates that it is not possible for two individuals to have a strong relationship with a common friend and not know each other. That is, it is not possible to have an open triangle in the network graph where both edges of the triangle are labeled strong.

We use the STC property to characterize the ties of a social network by asking for a labeling of the edges of the social graph into strong and weak such that the STC property is satisfied. There is a trivial solution to this problem which is to label all edges weak. However, we believe that creating strong relationships is the main motivation for users to join, and actively engage with a social network, online or otherwise. Therefore, we look for a labeling that also maximizes the number of strong ties (or minimizes the number of weak ties).

We thus obtain the following two problems: the MAXSTC problem where we ask for a labeling of the graph such that the STC property holds and the number of strong edges is maximized, and the MINSTC problem where we seek to minimize the number of weak edges. These are two novel combinatorial optimization problems that are of independent theoretical interest. Both problems are NP-hard, and we thus look for efficient algorithms with approximation guarantees. We show that this is not possible for the MAXSTC problem. For the MINSTC problem, we show that it can be expressed as a graph vertex cover problem on an appropriately defined graph, a problem known to have a constant factor approximation algorithm.

We also extend our formulation to capture more complex problems where new edges may be added to the graph, or ties in the network may be of different types. Of particular interest is the MINMULTISTC problem where we seek to enforce a variant of the STC property in the presence of multiple *types* of strong edges. The problem of understanding the "type" of an edge is something that arises naturally in practice. Although there are specialized online social networks catering to different needs of the users (social, professional, informational), the boundaries between these different circles are not always clear. It is often the case that there are multiple types of relationships in a single network. Therefore, it is important to be able to not only distinguish between strong and weak ties, but also to differentiate between different types of relationships and social circles.

We test our algorithms experimentally on real datasets. Our experiments demonstrate that the labeling we obtain based on the structural graph property of Strong Triadic Closure correlates well with empirical measures of tie strength. Furthermore, our labeling agrees with the celebrated "strength of weak ties" observation [8]. The edges between different communities in the social network are usually labeled weak, while the strong edges concentrate among the nodes of the communities.

In summary, in this work we make the following contributions.

- We formulate the problem of characterizing the edges of a social network as a novel optimization problem where we seek to enforce the Strong Triadic Closure property while maximizing the number of strong edges in the graph (or minimizing the number of weak edges).

- We show that our optimization problem is NP-hard. For the minimization problem there exists an efficient approximation algorithm with constant approximation ratio, while there is no good approximation algorithm for the maximization problem.

- We propose and study two extensions to our problem. The first allows the addition of new edges in the graph in order to enforce the STC property. The second allows for multiple types of strong edges.

- We study our algorithms experimentally on real datasets. We show that there is a correlation between the label of an edge and naturally defined notions of tie strength, and that the weak edges act like bridges between different communities in the network.

The rest of the paper is structured as follows. In Section 2 we review some of the related work. In Section 3 we formally define the problems we will study. In Section 4 we study the complexity of our problem, and in Section 5 we consider approximation algorithms. Section 6 considers extensions to the basic problem. Section 7 contains the experimental evaluation, and Section 8 concludes the paper.

2. RELATED WORK

In this paper we build upon the Strong Triadic Closure principle from Psychology. Strong Triadic Closure was first defined by Granovetter [8] in his seminal paper "The Strength of Weak Ties". Previously, Davis [3] and Newcomb [15] discuss some evidence that this property exists in social networks. The Strong Triadic Closure is discussed in detail in the book of Easley and Kleinberg [4]. They discuss the effect of the property on the structure of the network, and possible relaxations, but they do not consider the problem of labeling the edges of the graph to enforce the property. In the discussion they also consider recent experiments [10, 16] which demonstrate a correlation between structural properties of an edge and a notion of strength measured in practice. We perform similar experiments in Section 7 where we study the correlation between the label of an edge and the empirical tie strength.

Recent work has considered the problem of assessing the link strength in a social network, using data from e-mails [14], phone calls [16], and social media [7]. Kahanda and Neville [12] develop a supervised learning approach to predict link strength from transactional information (communication, file transfers, etc) and differentiate between strong and weak relationships in large-scale social networks. Gilbert and Karahalios [7] develop a model for characterizing ties in a social network using features about the similarity and interaction between users. They validate their model on small-scale data collected with questionnaires. In a follow-up work, Gilbert [6] explores how well a tie-strength model developed for one social medium adapts to another in order to find relationships which transcend a particular medium. In addition, Xiang et al. [22] develop an unsupervised model to estimate relationship strength from interaction activity and user similarity. They also handle heterogenous relationship strength (e.g. acquaintances, best friends). Their work is motivated by the theory of homophily from sociology, which postulates that people tend to form ties with other people who have similar characteristics. Jones et al. [11] found that the frequency of online interaction is a good predictor of strong ties.

Another direction of related research focuses on characterizing the type of a relationship between users. Tang, et al. [19] use user and link characteristics to build a generative model which assigns the most likely type to a specific relationship. In a follow-up work, Tang et al. [18] extend their model for classifying the type of social relationships by learning across heterogenous networks. Their model incorporates ideas from social theories such as structural balance and social status. Backstrom et al. [1] use only the structure of the Facebook graph to identify the romantic partner of a user. They propose dispersion as a new network measure for estimating tie strength.

Our work is similar to this line of research in the sense that we are also trying to characterize the strength and type of social ties. However, prior work relies heavily on user and link characteristics to derive this characterization. In our case we only make use of the graph structure. We use the Strong Triadic Closure principle to formulate our labeling problem as a discrete optimization problem.

3. PROBLEM DEFINITION

Let $G = (V, E)$ be an undirected graph that represents a social network, where the set of vertices V corresponds to individuals, and the set of edges E corresponds to the connections (ties) between these individuals. The goal is to produce a *labeling* of the ties in the social network as either *strong* or *weak*. We will denote this labeling as a function $L_G : E \rightarrow \{W, S\}$, which maps each edge $e \in E$ to a label W (Weak), or S (Strong). Abusing the notation, we will sometimes use L_E to refer to the labeling of the set of edges in E.

The goal is to find a labeling that satisfies the *Strong Triadic Closure (STC)* property, which is defined as follows.

Definition 1 (Strong Triadic Closure). *Given a graph G, a labeling L_G of the graph satisfies the Strong Triadic Closure (STC) property, if there exists no pair of edges (u, v) and (u, w), such that $L_G(u, v) = S$ and $L(u, w) = S$, and $(v, w) \notin E$.*

Informally, the STC property requires that for every node u, it is never the case that u has strong ties with both v and w, yet there is no tie between v and w.

We want to label the edges of the graph into strong or weak, such that the labeling satisfies the STC property. It is easy to see that a trivial solution to this problem is to label all edges in the network as weak. However, we believe that people build social networks with the goal to create strong ties with other people, therefore, we ask for a labeling that satisfies the STC property while maximizing the number of strong ties. Equivalently, we can ask to minimize the number of weak ties in the network, while satisfying the STC property. Let $S(L_G)$ and $W(L_G)$ denote the number of strong and weak ties respectively produced by the labeling L_G. We are interested in the following two problems.

Problem 1 (Maximum Strength STC (maxSTC)). *Given a graph G, find a labeling L_G that satisfies the STC property and maximizes $S(L_G)$.*

Problem 2 (Minimum Weakness STC (minSTC)). *Given a graph G, find a labeling L_G that satisfies the STC property and minimizes $W(L_G)$.*

In the following we show that the problems are NP-hard, and we consider approximation algorithms.

4. COMPLEXITY ANALYSIS

To establish hardness it is sufficient to consider one of the two variants. We will show that the maxSTC problem is NP-hard, since the proof is easier.

Before going into the hardness proof, we introduce some notation and provide some intuition about the problem. Let $(u, v), (u, w) \in E$ denote pair of edges in the graph G that share a common endpoint u. We say that the edges define an *open triangle* $\langle (u, v), (u, w) \rangle$ *incident* on u, if $(v, w) \notin E$. The first observation is that, according to the definition of the STC property, a labeling L_G *violates* the STC property if and only if there exists at least one open triangle $\langle (u, v), (u, w) \rangle$ such that both (u, v) and (u, w) are labeled strong. In this case, we say that the open triangle *violates* the STC property; otherwise, we say that the open triangle *satisfies* the STC property. It is clear that a labeling L_G satisfies the STC property, if there is no open triangle that violates the STC property. Furthermore, it is also clear an edge (u, v) that does not belong to any open triangle should be labeled strong. The labeling of (u, v) does not affect the labeling of the remaining edges, since it cannot cause the STC property to be violated. Thus, when looking for the optimal solution, we only need to consider edges that participate in at least one open triangle.

For our reduction we will consider a special type of network: the *ego-network* G_u of a user u. Assuming an underlying social network G, consider a single user $u \in V$ of the network, and let N_u be the friends of u. Let E_u denote the set of edges from u to the nodes in N_u, and E_N the set of edges between the nodes in N_u. The ego-network of a user u is defined as $G_u = (\{u\} \cup N_u, E_u \cup E_N)$.

We will now define a simpler variant of the maxSTC problem, which we will show that it is as hard as the maxSTC problem. In our new problem, given an ego-network G_u we ask for a labeling of just the edges E_u incident on the node u such that the STC property is not violated. That is, there is no open triangle incident on u that has both edges labeled as strong.

Problem 3 (maxEgoSTC). *Given the ego-graph G_u of user u, find a labeling L_{E_u} of the edges incident on node u that satisfies the STC property and maximizes $S(L_{E_u})$.*

It is easy to show that the maxSTC problem is at least as hard as the maxEgoSTC problem.

Lemma 1. *There is a polynomial-time reduction from the* maxEgoSTC *problem to the* maxSTC *problem.*

Proof. Let G_u be the ego-network of node u that is given as input to the maxEgoSTC problem. The reduction is straightforward: we create an instance of the maxSTC problem, using the the graph G_u as input and asking for a labeling L_E that maximizes the number of strong edges. The key observation is that whether or not the labeling L_{E_u} of the edges in E_u satisfies the STC property is independent of the labeling L_{E_N} of the edges in E_N. This follows from the fact that there exists no open triangle in G_u that contains an edge from E_u and an edge from E_N. Such a triangle would have to be of the form $\langle (u, v), (v, w) \rangle$, missing the edge (u, w). However this is not possible, since by construction $(u, w) \in E_u$. Therefore, all open triangles in G_u contain either two edges from E_u, or two edges from E_N. Thus, we can label the edges E_u and E_N independently. Finding a labeling L_E that maximizes the number of strong edges in G_u while respecting the STC property requires to find labelings L_{E_u} and L_{E_N} that maximize the number of strong edges in E_u and E_N respectively. The labeling L_{E_u}

is a solution of the maxEgoSTC problem. Note also that finding a labeling L_{E_N} that maximizes the number of strong edges in E_N is an instance of the maxSTC problem, with the graph $G_N = (N_u, E_N)$ as input

More formally, consider the decision problem for the maxEgoSTC problem, where given a graph G_u, we ask if there is a labeling L_{E_u} that has $S(L_{E_u}) \geq k$. Consider also the decision version of the maxSTC problem, where given a graph G, we ask if there is a labeling L_G that has $S(L_G) \geq \ell$. Given the graph G_u we create the graph G_N and using binary search on the value of ℓ we find the labeling $L^*_{G_N}$ that maximizes $S(L_{G_N})$. Let $S(L^*_{G_N}) = \mu$. Now we give the graph G_u as input to the maxSTC problem and we ask if there is a labeling L_{G_u} such that $S(L_{G_u}) \geq k + \mu$. Since the labeling of E_u and E_N are independent, there is a labeling L_{E_u} with $S(L_{E_u}) \geq k$, if and only if there is a labeling L_{G_u} with $S(L_{G_u}) \geq k + \mu$. □

Lemma 2. *The* maxEgoSTC *problem is NP-hard.*

Proof. We will now show that the maxEgoSTC problem is NP-hard by reducing the maxClique problem to it. Given an input graph $G = (V, E)$ and a value k, the decision version of the maxClique problem asks if there exists a subset $V_c \subseteq V$ of vertices of size at least k, such that the induced subgraph $G_c = (V_c, E_c)$ forms a clique, where $E_c = \{(u, v) \in E : u \in V_c, v \in V_c\}$.

Given the input graph $G = (V, E)$ to the maxClique problem, we create an instance of the maxEgoSTC problem, by creating an ego-network G_u consisting of an additional node u and edges E_u that connect node u to all the nodes V of G. That is, $G_u = (\{u\} \cup V, E_u \cup E)$. We ask if there is a solution of the maxEgoSTC problem of size at least k.

Let $S \subseteq E_u$ be the subset of edges in E_u that are labeled strong according to a labeling L_{E_u}. Each edge $(u, v) \in S$ in the ego-network defines a unique vertex $v \in V$. Let $V_S \subseteq V$ denote the set of vertices defined by the set of edges S. The labeling L_{E_u} satisfies the STC property, if and only if the set of vertices V_S defines a clique in the graph G. If the labeling of the edges in S satisfies the STC property, then no pairwise combination of edges from S can create an open triangle. Therefore, for every pair of edges $(u, v), (u, w) \in S$, we have that $(v, w) \in E$, and thus V_S defines a clique. On the other hand, if the labeling of the edges in S does not satisfy the STC property, then there must exist at least one pair of edges $(u, v), (u, w) \in S$ that define an open triangle. Therefore, the edge $(v, w) \notin E$ and hence the set V_S does not define a clique.

Therefore, there exists a labeling L_{E_u} for the maxEgoSTC problem such that $S(L_{E_u}) \geq k$, if and only if, there exists a clique of size at least k in graph G. □

5. APPROXIMATING MIN-STC

Given that the two problems we consider are NP-hard, we look for approximation algorithms. The reduction from maxClique to maxEgoSTC preserves the approximation, so, following the result in [9] we cannot approximate the maxEgoSTC solution within a factor better than $O(n^{1-\epsilon})$. Fortunately, we can do better for the case of the minSTC problem.

Theorem 1. *There exists a 2-approximation algorithm for the* minSTC *problem.*

Proof. Recall that a labeling L_G satisfies the STC property, if there exists no open triangle that violates the STC property. That is, there is no open triangle $\langle (u, v), (u, w) \rangle$ such that both (u, v) and (u, w) are labeled strong. Therefore, for every open triangle, at least one of the edges of the open triangle must be labeled weak. We say that this edge *covers* the open triangle. The goal is to find the minimum set of edges that cover all open triangles in the graph.

Using this intuition we will show how the minSTC problem can be mapped to the *Minimum Vertex Cover* (minVertexCover) problem. Given a graph $G = (V, E)$, a subset of vertices $C \subseteq V$ is a *vertex cover* of the graph G, if for every edge $(u, v) \in E$, $u \in C$ or $v \in C$. The minVertexCover problem, given a graph G looks for a vertex cover of G with the smallest number of vertices.

The mapping from minSTC to minVertexCover proceeds as follows. Given a graph $G = (V, E)$ that is input to the minSTC problem, let T denote the set of all open triangles in G. We create a *dual* graph $G_T = (V_E, E_T)$ that is input to minVertexCover as follows. For every edge $e \in E$ we create a vertex $v_e \in V_E$. For every open triangle $\langle e_1, e_2 \rangle \in T$, we create an edge $(v_{e_1}, v_{e_2}) \in E_T$.

Given a labeling L_G we define the set C to be the set of vertices $v_e \in V_E$ such that the corresponding edge $e \in E$ is labeled weak. If L_G satisfies the STC property, then for every triangle $\langle e_1, e_2 \rangle \in T$ at least one of the edges e_1, or e_2 must be labeled weak. Therefore, for every edge $(v_{e_1}, v_{e_2}) \in E_T$ at least one of the two endpoints is included in the set C, and hence C is a vertex cover for the graph G_T.

Furthermore, given a minimum vertex cover $C \subseteq V_E$ of the graph G_T we can create a labeling L_G by labeling as weak every edge $e \in E$, such that $v_e \in C$, and the remaining edges as strong. Since C is a vertex cover for G_T, by construction of G_T it follows that every open triangle in G is covered by at least one edge labeled weak. Therefore, the labeling L_G respects the STC property. If C is the minimum vertex cover, then L_G is the labeling with the minimum number of weak edges. If C is an α-approximate solution for the minimum vertex cover, then L_G is an α-approximation of the minimum number of weak edges. It is well known that there is a 2-approximation algorithm for the minVertexCover problem [21, 2], which implies a 2-approximation algorithm for the minSTC problem. □

In our experiments we consider two different approximation algorithms for the minSTC problem: A 2-approximation algorithm that relies on finding a *maximal matching* for the dual graph G_T; A greedy $O(\log n)$-approximation algorithm that constructs a vertex cover of G_T by always selecting the node that covers the most uncovered edges. We discuss the details of the algorithms in Section 7.

6. EXTENSIONS AND VARIATIONS

We will now discuss some extensions and variations to the basic minSTC problem.

6.1 STC with edge additions

Consider the case that the graph G that we want to label consists of a full clique of n nodes, missing a single edge (u, v). Then, the best labeling we can obtain has $n - 2$ weak edges (all the edges incident to either u or v). However, we could obtain a labeling with all the edges labeled strong if we simply added the missing edge (u, v). Thus, we consider a new minimization problem, where, in order to guarantee the STC property, except for labeling existing edges as weak, we can also add new edges to the graph. The goal is to minimize the number edges added to the graph, and the number of edges in the original graph that are labeled weak. We refer to this problem as MINSTC+.

PROBLEM 4 (MINSTC+). *Given a graph $G = (V, E)$, identify a set of* additional *edges $E' \subseteq V \times V \setminus E$ and a labeling $L_{G'}$ of the graph $G' = (V, E \cup E')$ that satisfies the STC property such that $W(L_E) + |E'|$ is minimized.*

The MINSTC+ problem is also NP-hard. However, we can again view it as a coverage problem, and exploit the fact that there is a known approximation algorithm.

LEMMA 3. *There is a $O(\log n)$-approximation algorithm for the* MINSTC+ *problem.*

PROOF. We will show that our problem can be modelled as an instance of the Minimum Hitting Set (MINHITSET) problem. The minimum hitting set problem is defined as follows. Given a universe of elements U and a collection of subsets of U, $\mathcal{S} = \{S_1, ..., S_n\}$, we want to find a subset $C \subseteq U$ of minimum size, such that for each $S_i \in \mathcal{S}$, $S_i \cap C \neq \emptyset$, that is, each set $S_i \in \mathcal{S}$ is *hit* by C.

We can transform an instance of the MINSTC+ problem to an instance MINHITSET problem using a construction very similar to the one we used for transforming MINSTC to the MINVERTEXCOVER problem. Given the graph $G = (V, E)$ the universe U is defined as the set of all pairs of the form (u, v) where $u, v \in V$. For every open triangle $t = \langle (u, v), (u, w) \rangle$ of the graph G, we create a set $S_t = \{(u, v), (u, w), (v, w)\}$. The goal is to find the smallest subset C of pairs in U such that we hit all the sets S_t. Given a hitting set, we define the set E' as the set of pairs $(w, u) \in C$ such that $(w, u) \notin E$. The remaining pairs in C correspond to edges in E and are labeled weak. The labeling of graph G' satisfies the STC property since every open triangle in G is either covered by a weak edge, or it is closed by an edge in E'. We can assume that the additional edges are labeled weak, therefore, they do not create any new violating open triangles.

Given a solution to the MINSTC+ problem, we can define a hitting set, by adding to the set C the pairs in E', and the edges in E that are labeled weak. Since all open triangles in G are covered, this defines a hitting set.

The MINHITSET problem is NP-hard, but the simple greedy algorithm that always selects the element that hits most sets that are not already hit is known to have a $O(\log n)$ approximation ratio. Therefore, there exists a $O(\log n)$ approximation algorithm for the MINSTC+ problem. □

6.2 STC with multiple relationship types

In the MINSTC (or MAXSTC) problem, there are only two types of edges: strong and weak. We assume that the weak edges are the “less important” ones, and we want to produce a labeling that maximizes the strong ones. We now consider the scenario where there are multiple types of strong ties. For example, when considering the social network of a specific individual it would be useful to understand which links correspond to strong family ties, strong work ties, or strong friendship ties. In this case we want to identify the strong edges of each type.

We model this problem using a natural extension of the STC property. Similar to before, the goal is to have as many strong edges as possible, such that there is no violating open triangle with both edges labeled strong. The difference is that with multiple types of strong edges, an open triangle is violating if both its edges are labeled strong *and* they are both of the same type. That is, it is ok for a user u to have strong relationships with users v, w and v, w to not be connected, as long as the type of relationship of (u, v) and (u, w) is different.

More formally, we assume a fixed number k of strong relationship types. We can view these types as k labels $\{S_1, ..., S_k\}$. We also have the additional label W for the weak edges. Given graph $G = (V, E)$, we want to produce a labeling $L_E : E \to \{W, S_1, ..., S_k\}$ of the edges of a graph G. The labeling must satisfy the *multi-Strong Triadic Closure* property (multi-STC) which is defined as follows.

DEFINITION 2 (MULTI-STRONG TRIADIC CLOSURE). *Given a graph G, a labeling $L_E : E \to \{W, S_1, ..., S_k\}$ satisfies the multi-Strong Triadic Closure (multi-STC) property, if there exists no pair of edges (u, v) and (u, w), such that $(v, w) \notin E$ and $L(u, v) = L(u, w) = S_i$, for some $i \in [1, k]$.*

Similar to the STC property, we can trivially satisfy the multi-STC property by labeling all edges as weak. Our goal is again to maximize the number of edges labeled strong (of any type), or minimize the number of edges labeled weak. The maximization problem runs again into the problem of finding the maximum clique, so we study the minimization problem.

PROBLEM 5 (MINMULTISTC). *Given a graph G, and k strong edge types, find a labeling L_E that satisfies the multi-STC property and minimizes $W(L_E)$.*

We can now prove the following theorem.

THEOREM 2. *The* MINMULTISTC *problem is NP-hard for any $k \geq 2$. There is a $O(\log n)$-approximation algorithm for $k = 2$. The problem is hard to approximate for $k \geq 3$, unless $P = NP$.*

PROOF. For the proof, we will make use of the dual graph G_T that we constructed in Section 5. We can model our problem as a coloring problem on the dual graph G_T, where we want to color the nodes of G_T with $k + 1$ colors. There are k “strong” colors $\{S_1, ..., S_k\}$, one for each strong label, plus an additional “white” color W for the weak label. We want a *legal* coloring of the nodes of the graph G_T, where no two adjacent nodes can be colored with the same strong color. It is ok if we have two adjacent nodes having a white color. We want a legal coloring that minimizes the number of white nodes.

For $k = 2$ our problem is equivalent to the odd-cycle traversal problem [17], which given a graph asks for the minimum number of vertices to be removed, so that the resulting graph becomes bipartite. This problem is also known to be NP-hard, but there is a $O(\log n)$-approximation algorithm [5].

For $k \geq 3$ we can show that the problem is not only NP-hard, but also hard to approximate unless $P = NP$. The proof follows easily by observing that for a k-colorable graph, the optimal solution to the MINMULTISTC problem has cost zero, that is, we do not need to use the white color. This implies that if there was an algorithm with bounded approximation ratio, then for an input instance for which the optimal algorithm has cost zero, the algorithm would be able to produce a solution with zero cost as well; otherwise the approximation ratio is infinite. However, for $k \geq 3$, finding a k-coloring of a k-colorable graph is NP-hard. Therefore, it is hard to decide if there is a solution to the minMulti-STC problem that has cost greater than zero. Therefore, the problem is hard to approximate, unless $P = NP$. □

We note that for $k = 2$, the $O(\log n)$-approximation algorithm makes use of linear programming for deriving the solution. We propose a simpler heuristic in Section 7.

7. EXPERIMENTS

The goal of the experiments is to study if the labeling we obtain by enforcing the STC property correlates with an intuitive measure of tie strength in practice. We perform a variety of experiments towards this end. Our experiments are on real data, and demonstrate the practical utility of our formulation and of the proposed algorithms.

7.1 Datasets

We use five different datsets in our experiments: *Actors*, *Authors*, *Les Miserables*, *Karate Club* and *Amazon Books*. Table 1 shows some statistics about our datasets. The column "Weights" indicates whether we can compute weights for the edges of the graph. The weight of an edge corresponds to the empirical strength of the connection. The column "Community Structure" indicates whether there exists a known community structure in the graph.

Table 1: Datasets Statistics.

Dataset	Nodes	Edges	Weights	Community structure
Actors	1,986	103,121	Yes	No
Authors	3,418	9,908	Yes	No
Les Miserables	77	254	Yes	No
Karate Club	34	78	No	Yes
Amazon Books	105	441	No	Yes

We now describe the datasets in detail.

The *Actors* dataset: We create a graph from a movie dataset collected from IMDB[1], consisting of 3,125 movies made from 1945 to 2010, and 2,171 actors that participate in these movies. The actor graph contains a node for each actor in the data, and there is an edge between two actors if they have collaborated in at least one movie. For each node of the graph we also have information about the set of movies in which the actor has played. We prune actors who participated in less than 5 movies since we do not consider them to be significant members of the network.

The *Authors* dataset: This dataset was obtained from data downloaded from the DBLP site[2]. It consists of a collection of authors that have published papers in one of the major Data Mining, Databases or Theory conferences during the period between 1994 and 2013. The author graph contains a node for each author in the data, and there is an edge between two authors if they have collaborated in at least one paper. For each node in the graph we also have information about the set of papers the author has written. We prune authors who wrote less than 3 papers since we do not consider them to be significant members of the network.

The *Les Miserables* dataset: This dataset contains the network of co-appearances of characters in Victor Hugo's novel "Les Miserables" [13]. Nodes represent characters of the novel, and there is an edge between two nodes if the pair of characters appear in the same chapter of the book. For each edge we have the number of such co-appearances between the two characters.

The *Karate Club* dataset: Zachary's Karate Club dataset [23] is a social network of friendships between 34 members of a karate club at a US university in the 1970s. The information about the friendship was derived by questionnaires filled out by the members of the club.

The *Amazon Books* dataset: This dataset contains a set of books about US politics published around the time of the 2004 presidential election which are sold by the online bookseller Amazon.com[3]. Edges between books represent frequent co-purchasing of the books. In addition, each node (book) is labeled as "liberal", "neutral", or "conservative", depending on its political viewpoint. There are 43 liberal, 13 neutral and 49 conservative books in this dataset.

7.2 Algorithms

In Section 5, we proved that minSTC problem on the graph G can be mapped to the minVertexCover problem on the *dual* graph G_T. Given the graph G, the dual graph G_T is constructed by creating a node for every edge of G, and connecting two nodes if the corresponding edges form an open triangle. The algorithms we consider work by constructing an approximate solution to the minVertexCover problem. We now describe them in detail.

The Greedy Algorithm: The input to the algorithm is the graph G and its dual G_T, and the output is a labeling of the edges of the graph G as strong or weak. The algorithm works by constructing a vertex cover of graph G_T in a greedy fashion. Recall that a vertex cover of a graph is a set of vertices such that every edge of the graph has at least one endpoint in the set. Let C denotes the set of nodes which are selected by our algorithm. Initially $C = \emptyset$. At every step the algorithm selects the node v with the maximum degree in G_T, and adds it to the set C. It then deletes node v and all edges incident on v from graph G_T. The process is repeated until there are no more edges in the graph G_T. Given the set of nodes in C, we label the corresponding edges of graph G as weak. The remain edges are labeled strong. This algorithm is known to be a $O(\log n)$-approximation algorithm [21].

If at any step of the algorithm more than one nodes have the same degree, we break ties by choosing the node that corresponds to the edge in G that participates in the fewest *closed triangles* in the graph G. This way, our algorithm tends to label as weak edges that participate in many open triangles and few closed triangles, a principle that agrees with our intuition of what a weak edge should be.

The MaximalMatching Algorithm: The MaximalMatching algorithm also produces a vertex cover of the graph G_T, by constructing a maximal matching for the dual graph G_T. A matching of a graph is a collection of non-adjacent edges of the graph, while a maximal matching is one where no additional edges can be added. The algorithm constructs the matching one edge at the time. Let M denote the set

[1] http:www.imdb.com

[2] http://dblp.uni-trier.de/xml/

[3] Available by V. Krebs at http://www.orgnet.com/.

of edges selected by our algorithm. Initially $M = \emptyset$. The algorithm selects the next edge to add to the set M by first selecting the node u with the highest degree in G_T and then the neighbor v of u with the highest degree. If more than one nodes have the same degree then we break ties in the same way as in the Greedy Algorithm. We add edge (u, v) to M, and delete u, v and all edges incident on u or v from G_T. The algorithm terminates when there are no more edges in the graph G_T. Let C denote the set of vertices that are endpoints of the edges in M. Similar to before, we label as weak the corresponding edges of G, while the remaining edges are labeled as strong. This algorithm is known to be a 2-approximation algorithm [21].

Note that for both algorithms if there are vertices in the graph G_T that have no incident edges, then the corresponding edges in the graph G will be labeled strong. These correspond to edges that participate only in closed triangles, or that are isolated in the graph G.

Table 2 shows the number of edges labeled weak and strong for the two algorithms on the five datasets we consider in this paper. Despite the better approximation ratio the MaximalMatching algorithm always produces a larger number of weak edges.

Table 2: Number of strong and weak edges for Greedy and MaximalMatching algorithms.

	Greedy		MaximalMatching	
	Strong	Weak	Strong	Weak
Actors	11,184	91,937	8,581	94,540
Authors	3,608	6,300	2,676	7,232
Les Miserables	128	126	106	148
Karate Club	25	53	14	64
Amazon Books	114	327	71	370

7.3 Measuring Tie Strength

In this section we study the relationship between the assigned labels and a notion of tie strength measured in practice. Our experiments follow the line of experimentation in prior work [16, 10] where they study how structural features of an edge correlate with empirical tie strength.

For this experiment, we use the three datasets for which we can compute weights for the edges: the *Actors* dataset, the *Les Miserables* dataset and the *Authors* dataset. The weights on the edges correspond to the strength of the relationships: a strong and enduring collaboration between two nodes in the case of the *Actors* and *Authors* datasets, and high affinity in the storyline of the novel in the case of the *Les Miserables* dataset. Specifically, for the *Actors* dataset, the weight of an edge is the number of times that the two actors have collaborated; for the *Authors* dataset it represents the number of papers that they have written together; for the *Les Miserables* dataset, it is the number of co-appearances between two characters in the same chapter. The goal of this experiment is to test the validity of the edge labeling, by examining if there is a correlation between the assigned label and the weight of the edge. Mathematically, we will show that there is a statistically significant difference between the mean weight of strong and weak edges.

Table 3 shows the mean weight for the strong and weak edges for all the three datasets, using the Greedy and MaximalMatching algorithms. Clearly, for all of the datasets the strong edges have higher weight than the weak ones. The t-test reveals that the difference is statistically significant at a 5% confidence level. We can thus conclude that the labeling of our algorithm agrees with the “true” strength of the network ties.

Table 3: Mean count weight for strong and weak edges for Greedy and MaximalMatching algorithms.

	Greedy		MaximalMatching	
	S	W	S	W
Actors	1.4	1.1	1.3	1.1
Authors	1.341	1.150	1.362	1.167
Les Miserables	3.83	2.61	3.87	2.76

The frequency of common activity (e.g. collaboration) between two users is obviously a strong indicator of tie strength. However it may also be an artifact of the general frequent activity of the two users. For example, two highly prolific researchers may collaborate on higher-than-average number of papers, but this may be simply due to the fact that they produce a lot of publications in general. An alternative measure of tie strength is the fraction of the activity of the two users that is devoted to their relationship. We use Jaccard similarity to capture this idea. Recall that Jaccard similarity between two sets is defined as the ratio of their intersection over their union. In our case the sets correspond to the sets of activities in which the two users engage (e.g., movies, publications, etc), and the Jaccard similarity measures the fraction of their activities that are common.

For this experiment we use the *Actors* and the *Authors* datasets. For the *Actors* dataset the weight of an edge between two actors is the number of movies in which they have played together, over the total number of movies in which at least one of the two actors has participated. Similarly, the weight of an edge between two authors is defined as the number of papers that they have written together over the total number of their papers. We cannot compute Jaccard similarity for the *Les Miserables* dataset, since we do not have the exact chapter appearances for each character.

Table 4 shows the mean Jaccard similarity for the strong and weak edges using Greedy and MaximalMatching algorithms. Again, for all of the datasets the strong edges have higher weight than the weak ones and the t-test reveals that this difference is statistically significant at a 5% confidence level. We note that in the case of Jaccard similarity, the gap between strong and weak edges is larger than before. It seems that our labeling is more adept at capturing this focused measure of tie strength.

Table 4: Mean Jaccard similarity for strong and weak edges for Greedy and MaximalMatching algorithms.

	Greedy		MaximalMatching	
	S	W	S	W
Actors	0.06	0.04	0.06	0.04
Authors	0.145	0.084	0.155	0.088

Comparing the MaximalMatching and the Greedy algorithm we observe that they behave very similarly in terms of the mean weights of strong and weak edges. However, the Greedy algorithm produces consistently a larger number of strong edges, and it is intuitively more appealing.

7.4 Weak edges as bridges

Granovetter, in his seminal paper [8], demonstrated the importance of weak social ties in connecting individuals with information that is not readily available in their close social circle, such as new work opportunities. A possible explanation to this observation is nicely articulated in the book of David Easley and Jon Kleinberg [4], where they postulate that weak ties act as *bridges* between communities in the graph. Communities hold different types of information, and the only way for an individual to obtain access to information from a community different than her own is through weak ties.

In accordance to this interpretation, given a labeling of the edges of a graph with known community structure, we would like most of the inter-community edges to be labeled weak, while most of the strong labels to be confined to intra-community edges. That is, edges that bridge communities should be labeled weak, while strong edges should serve as a backbone of the communities.

Formally, let $G = (V, E)$ denote the input graph, and let $\mathcal{C} = \{C_1, ..., C_k\}$ denote a partition of the nodes of the graph into k communities, which is also given as part of the input. Let E_{inter} denote the set of edges (u, v) such that $u \in C_i$ and $v \in C_j$ for some $i \neq j$, and let E_{intra} denote the set of edges (u, v) such that $u, v \in C_i$ for some i. Also given the labeling L_G of the graph G let W denote the set of edges labeled weak, and let S denote the set of edges labeled strong. We define the precision P_W and recall R_W for the weak edges as follows:

$$P_W = \frac{|W \cap E_{inter}|}{|W|} \qquad \text{and} \qquad R_W = \frac{|W \cap E_{inter}|}{|E_{inter}|}$$

Similarly, we define precision P_S and recall R_S for strong edges as follows:

$$P_S = \frac{|S \cap E_{intra}|}{|S|} \qquad \text{and} \qquad R_S = \frac{|S \cap E_{intra}|}{|E_{intra}|}$$

The numbers we are mostly interested in are R_W and P_S, that is, we want the bridging edges to be labeled weak, and the strong edges to be confined within the communities.

To test our hypothesis we need graphs with known community structure. To this end, we use the *Karate Club* and *Amazon Books* datasets. For the *Karate Club* dataset it is well known [4] that there were two fractions within the members of the club, centered around the two trainers, that eventually led to the breakup of the club. For the *Amazon Books* dataset the communities are given by the political viewpoint of the books.

Table 5: Precision and Recall for strong and weak edges for Greedy and MaximalMatching algorithms.

Greedy				
	P_S	R_S	P_W	R_W
Karate Club	1	0.37	0.19	1
Amazon Books	0.81	0.25	0.15	0.69
MaximalMatching				
	P_S	R_S	P_W	R_W
Karate Club	1	0.2	0.16	1
Amazon Books	0.73	0.14	0.14	0.73

Table 5 shows the results for the two datasets for the Greedy and MaximalMatching algorithms. The two algorithms behave similarly, but the Greedy algorithm performs better overall in terms of both precision and recall. We now study the labeling of the Greedy algorithm in more detail.

For the *Karate Club* dataset we observe that we have perfect precision for the strong edges, and perfect recall for the weak edges. We visualize the results of the Greedy algorithm in Figure 1. The nodes are colored white and gray depending on the community to which they belong. The thick red edges correspond to the edges labeled strong, and the thin blue edges to the edges labeled weak. We can see that strong edges appear only between nodes of the same group, while all edges that cross communities are labeled weak.

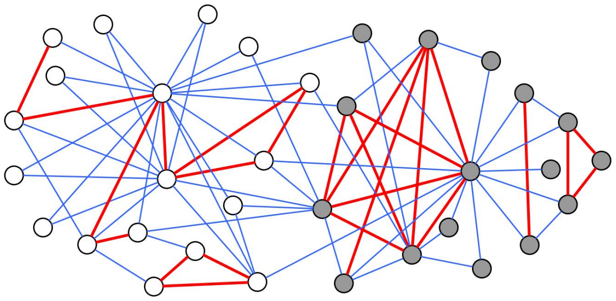

Figure 1: Karate Club graph. Blue light edges represent the weak edges, while red thick edges represent the strong edges.

For the *Amazon Books* dataset the Greedy algorithm characterizes 114 edges as strong, out of which 92 connect books of the same type, thus yielding precision $P_S = 0.81$. On the other hand, there are 70 edges that connect nodes from different groups, and 48 of those are labeled weak, yielding recall $R_W = 0.69$. Of the remaining 22 edges that cross communities and are labeled strong, 20 are edges with one of the two endpoints being a book labeled as neutral. It is intuitive that people would co-purchase books of neutral viewpoint with liberal or conservative books, thus leading to strong connections. There are only two edges that connect a liberal and a conservative pair of books, and are labeled strong by our algorithm. These pairs are: ("America Unbound", "Rise of the Vulcans"), and ("The Choice", "Rise of the Vulcans"). After some investigation, we found out that, for the first pair, although the books "America Unbound" and "Rise of the Vulcans" belong to different categories (liberal and conservative respectively), they are both about the exact same issue: George W. Bush's foreign policy. Therefore, there is a different latent dimension that groups them together, which can explain the strong relationship between them.

7.5 STC with added edges

In this section we conduct experiments for the MINSTC+ problem, where except for labeling edges as strong or weak, we can also add edges to the graph. To this end we use the greedy algorithm we described in Section 6. The algorithm works iteratively. At each step of the algorithm a pair of nodes (u, v) is selected which covers the most remaining open triangles. This pair is either an edge not currently in the graph, which, when added, closes the most remaining open triangles, or an existing edge, which, when labeled weak, covers the most remaining open triangles. We refer to this algorithm as the Greedy+ Algorithm.

Table 6 shows the number of strong, weak and added edges using the Greedy+ algorithm. We can see that, as expected, using added edges the number of strong edges increases. Table 7 shows the mean weight for the strong and weak edges for the Greedy+ algorithm. It is still the case that strong edges have higher mean weight than the weak edges, however, compared to the results in Table 3, the strong edges have lower mean weight while weak edges have higher mean weight. Therefore, although the Greedy+ algorithm labels more edges as strong, it seems that some of these edges are of low weight.

Table 6: Number of strong, weak and added edges for the Greedy+ algorithm.

	Strong	Weak	Added
Actors	12095	91026	537
Authors	4041	5867	343
Les Miserables	165	89	14
Karate Club	29	49	1
Amazon Books	158	283	29

Table 7: Mean count weight for strong and weak edges for the Greedy+ algorithm.

	S	W
Actors	1.403	1.191
Authors	1.332	1.142
Les Miserables	3.345	3.011

To obtain further insight into the effect of added edges we look at the labeling produced by the Greedy+ algorithm for networks with known community structure. Table 8 shows the results for the *Amazon Books* and *Karate Club* datasets. We observe that the recall value R_S of the Greedy+ algorithm is higher compared to that of the Greedy algorithm, while the P_S value is almost the same. This means that the algorithm is successful at introducing strong edges within the communities as it was supposed to. We also compute the precision P_A of the added edges, shown in Table 8. We define P_A as the fraction of added edges that fall within the community. Clearly, the added edges serve the purpose of "strengthening" an existing community.

Table 8: Precision and Recall for strong and weak edges for the Greedy+ algorithm

	P_S	R_S	P_W	R_W	P_A
Karate Club	1	0.43	0.2	1	1
Amazon Books	0.79	0.37	0.13	0.53	0.72

We also visualize the effect of the added edges for the case of the *Karate Club* dataset in Figure 2. For this dataset, the Greedy+ algorithm adds only one edge (the dashed black edge in the figure). Compared to Figure 1, the algorithm maintains all the inter-community weak edges, and all the intra-community strong edges produced by the Greedy algorithm, and it adds an additional four strong edges within the grey community. The added edge reveals the existence of a near-clique of seven users in the network, which can now be labeled strong.

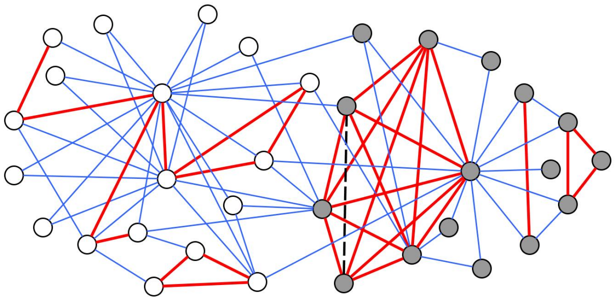

Figure 2: Karate Club graph. Blue light edges represent the weak edges, red thick edges represent the strong edges, and the black dashed edge represents the added edge

Therefore, we can conclude that the Greedy+ algorithm is better than the Greedy algorithm in revealing the backbone of an existing community in the graph. However, this comes at the price of labeling as strong more low-weight edges.

7.6 STC with multiple relationship types

We now consider the minMultiSTC problem where we have k different types of strong ties. For this problem, there is an approximation algorithm for $k = 2$, however, it makes use of Linear Programming, making it complex to implement. There is no known algorithm for $k > 2$.

We propose a heuristic algorithm that works for any k by making iterative calls to the Greedy algorithm. The algorithm starts by running the Greedy algorithm on the graph G, producing sets S_1 and W_1 of strong and weak edges respectively. We label the edges in S_1 as "Strong 1". Given the set W_1 we compute the subgraph G_{W_1} induced by the edges in W_1. We can now repeat the same process on the graph G_{W_1} to obtain a new set of edges S_2 to label "Strong 2", and a new subgraph G_{W_2}. We continue iteratively until all k labels have been utilized.

More formally, the i-th iteration of the algorithm takes the graph $G_{W_{i-1}}$ as input (where $G_{W_0} = G$), runs the Greedy algorithm, and produces the sets of edges S_i and W_i. The set S_i is labeled as "Strong i", and the set W_i is used to produce the graph G_{W_i} for the next iteration. This process is repeated until k iterations are completed. The set W_k in the final iteration is labeled as "Weak". We refer to this algorithm as the MultiGreedy Algorithm.

We now experiment with the MultiGreedy algorithm. For simplicity we only consider the case where $k = 2$, that is, we only have two types of strong edges. For this experiment, we use as input graph an *ego-graph*, that is, the relationships of a single individual, and the edges between them. This is a common scenario in online social networks, where we want to be able to discriminate between different types of relationships of a specific individual.

Using the *Authors* dataset, we create two ego-networks centered around Jon Kleinberg, and Ravi Kumar, two researchers with diverse interests and collaborations. We prune co-authors with whom the author has less than 3 papers together, so that we focus on the more meaningful collaborations. Tables 9 and 10 show the results. For Kleinberg we observe that the "Strong 1" edges correspond to collaborations related to the early Web research, and his association with IBM Almaden, while the "Strong 2" collaborations correspond to more theoretical publications. For Ravi Kumar, the "Strong 1" ties correspond to his time at IBM Almaden, while the "Strong 2" ties to the time at Yahoo. Our algorithm is able to differentiate between these two distinct types of relationships.

Table 9: J. Kleinberg's ego-network.

Type	Names
Strong 1	R. Kumar, S. Rajagopalan, A. Tomkins, A. Sahai, P. Raghavan
Strong 2	M. Sudan, D. P. Williamson
Weak	E. Tardos, F. T. Leighton, L. Backstrom, D. P. Huttenlocher, A. Kumar, J. Leskovec, L. Lee

Table 10: R. Kumar's ego-network.

Type	Names
Strong 1	A. Tomkins, D. Sivakumar, E. Upfal, P. Raghavan, S. Rajagopalan
Strong 2	V. Josifovski, S. Vassilvitskii, A. Z. Broder
Weak	R. Rubinfeld, K. S. McCurley, M. Mitzenmacher, A. Dasgupta, A. Panconesi, K. Punera, V. Rastogi, J. Novak, J. M. Kleinberg, M. Mahdian, E. Vee, S. Lattanzi, B. Reed, A. Sahai, R. Krauthgamer, T. S. Jayram, S. Pandey, B. Pang, R. V. Guha

8. CONCLUSIONS

In this paper we addressed the problem of characterizing the connections in a social network. We made use of the Strong Triadic Closure property, and we formulated a novel optimization problem where we look for a labeling of the edges of a graph into strong or weak, such that the STC property is satisfied and the number of weak edges is minimized. We studied the complexity of the ensuing problems, and we showed that for the minimization problem we can provide a constant approximation algorithm. We also considered extensions of the basic formulation, to account for edge additions and networks where ties may have several different types. The experimental results demonstrate that the labeling we obtain makes sense in practice.

Our work leaves room for further research. More specifically, it would be interesting to consider further relaxations of the STC property. It would also be interesting to have a stochastic model where each edge has a probability of being strong or weak, rather than belonging exclusively to one class or another. These questions become more interesting when we have more than one types of strong edges.

9. ACKNOWLEDGMENTS

This work is supported by the Marie Curie Reintegration Grant project titled JMUGS which has received research funding from the European Union.

10. REFERENCES

[1] L. Backstrom and J. M. Kleinberg. Romantic partnerships and the dispersion of social ties: a network analysis of relationship status on facebook. In *CSCW*, pages 831–841, 2014.

[2] K. L. Clarkson. A modification of the greedy algorithm for vertex cover. *IPL*, 16(1):23–25, 1983.

[3] J. A. Davis. Clustering and hierarchy in interpersonal relations. *American Sociological Review*, page 845.

[4] D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010.

[5] N. Garg, V. V. Vazirani, and M. Yannakakis. Multiway cuts in directed and node weighted graphs. In *ICALP*, volume 820, pages 487–498, 1994.

[6] E. Gilbert. Predicting tie strength in a new medium. In *CSCW*, pages 1047–1056, 2012.

[7] E. Gilbert and K. Karahalios. Predicting tie strength with social media. In *CHI*, pages 211–220, 2009.

[8] M. Granovetter. The strength of weak ties. *American Journal of Sociology*, 78:1360–1380, 1973.

[9] J. Hastad. Clique is hard to approximate within $n^{(1-\epsilon)}$. In *Acta Mathematica*, pages 627–636, 1996.

[10] B. A. Huberman, D. M. Romero, and F. Wu. Social networks that matter: Twitter under the microscope. *First Monday*, 14, 2009.

[11] J. J. Jones, J. E. Settle, R. M. Bond, C. J. Fariss, C. Marlow, and J. H. Fowler. Inferring tie strength from online directed behavior. *PLoS ONE 8*, 2013.

[12] I. Kahanda and J. Neville. Using transactional information to predict link strength in online social networks. In *ICWSM*, 2009.

[13] D. E. Knuth. The stanford graphbase: A platform for combinatorial computing. *Journal of Anthropological Research*, 1993.

[14] G. Kossinets and D. Watts. Empirical analysis of an evolving social network. *Science 311*, pages 88–90, 2006.

[15] T. M. Newcomb. *The Acquaintance Process*. New York: Holt, Rinehart & Winston, 1961.

[16] J.-P. Onnela, J. Saramaki, J. Hyvonen, G. Szabo, D. Lazer, K. Kaski, J. Kertesz, and A.-L. Barabasi. Structure and tie strengths in mobile communication networks. *PNAS USA*, 104:7332–7336, 2007.

[17] B. Reed, K. Smith, and A. Vetta. Finding odd cycle transversals. *Operations Research Letters*, 32(4):299 – 301, 2004.

[18] J. Tang, T. Lou, and J. Kleinberg. Inferring social ties across heterogenous networks. In *WSDM*, pages 743–752, 2012.

[19] W. Tang, H. Zhuang, and J. Tang. Learning to infer social ties in large networks. In *ECML/PKDD*, pages 381–397, 2011.

[20] J. Ugander, B. Karrer, L. Backstrom, and C. Marlow. The anatomy of the facebook social graph. *CoRR*, abs/1111.4503, 2011.

[21] V. V. Vazirani. *Approximation algorithms*. Springer, 2001.

[22] R. Xiang, J. Neville, and M. Rogati. Modeling relationship strength in online social networks. In *WWW Conference*, pages 981–990, 2010.

[23] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977.