Measuring Network Centrality Using Hypergraphs

Sanjukta Roy
Chennai Mathematical Institute
sanjukta@cmi.ac.in

Balaraman Ravindran
Indian Institute of Technology Madras
ravi@cse.iitm.ac.in

ABSTRACT

Networks abstracted as graph lose some information related to the super-dyadic relation among the nodes. We find natural occurrence of hyperedges in co-authorship, co-citation, social networks, e-mail networks, weblog networks etc. Treating these networks as hypergraph preserves the super-dyadic relations. But the primal graph or Gaifmann graph associated with hypergraphs converts each hyperedge to a clique losing again the n-ary relationship among nodes. We aim to measure Shapley Value based centrality on these networks without losing the super-dyadic information. For this purpose, we use co-operative games on single graph representation of a hypergraph such that Shapley value can be computed efficiently[1]. We propose several methods to generate simpler graphs from hypergraphs and study the efficacy of the centrality scores computed on these constructions.

Categories and Subject Descriptors

G.2.2 [**Hypergraphs**]: Network Centrality

Keywords

Hypergraph, Shapley value, Centrality

1. INTRODUCTION

The study of networks represents an important area of multidisciplinary research involving Physics, Mathematics, Chemistry, Biology, Social sciences, and Information sciences. These systems are commonly represented using simple graphs in which the nodes represent the objects under investigation, e.g., people, proteins, molecules, computer systems etc., and the edges represent interactions between the nodes. But, there are occasions in which interactions involve more than two actors. For example, deliberations that take place in a committee are multi-way interactions [2]. In such circumstances, using simple graphs to represent complex networks does not provide complete description of the real- world systems. For example, if we model a collaboration network as a simple graph, we would have to reduce the interaction into a set of two way interactions, and if there are multiple actors collaborating on the same project, the representation will be a clique on all those actors. As we will show, we lose information about the set of actors collaborating on the same project in the clique construction. A natural way to model these interaction is through a hypergraph. In a simple graph, an edge is represented by a pair of vertices, whereas an hyperedge is a non-empty subset of vertices.

Multi-way interactions, known as super-dyadic relations, characterize many real world applications and there has been much interest lately in modeling such relations, especially using hypergraphs [3, 4, 5, 6]. While there have been several approaches proposed for using various graph properties, such as connectivity, centrality, modularity, etc., for modeling interactions between actors, the extensions of these notions to hypergraphs is still an active area of research. The typical approach is to construct a simple graph from the hypergraph and compute the properties on the regular graph. The approaches differ in the construction of the simple graph.

A hypergraph of tag co-occurrences has been used for detecting communities using betweenness centrality [3]. Puzis et al. [4] used betweenness as a centrality measure on hypergraphs and gave an efficient approach using Brandes algorithm. Zhou et al. [7] pointed out the information loss occurs when hypergraphs are treated as simple graphs and utilised hypergraphs to device clustering technique. Katherine Faust [5] used non-dyadic affiliation relationships which allowed to quantify the centrality of a subset of actors or a subset of events. She calculated an actor's centrality as a function of the centrality of hyperedges of events to which the actor belongs and calculated the centrality of events as a function of centrality of the actors. Hypergraphs have been used to understand team assembly mechanisms to determine collaboration [8, 9, 10]. Jain et al. [11] studied Indian Rail network using hypergraph.

Of particular interest to us in this work are centrality measures. A centrality measure identifies nodes that are in the 'center'of a network. In practice, identifying exactly what we mean by 'center'depends on the application of interest. In this work, we mainly look at this from the view point of information propagation. *Independent Cascade* and *Linear Threshold* are the two information diffusion models proposed in [12]. To find a seed set of k nodes that maximizes influence propagation is in general a hard problem and [12] gave an $(1 - 1/e - \epsilon)$ approximation under the assumption that

CODS '15, March 18 - 21, 2015, Bangalore, India
 978-1-4503-3436-5/15/03 $15.00.
http://dx.doi.org/10.1145/2732587.2732595

the activation functions are submodular. We use the diffusion models of [12] to define measures of effectiveness of the different centrality scores that we investigate.

There have been some earlier attempts at computing centrality scores on hypergraphs. Faust [5] looked at a hypergraph as a bipartite graph[6] between actors and events when computing closeness centrality. Puzis et al.[4] essentially converted each hyperedge to a clique and computed betweenness centrality on the modified graph efficiently. Representing each hyperedge as a clique converts the hypergraph to a general graph. We call this conversion as *clique construction*. Clique construction is the method that is widely used for computations on hypergraph. But in this work we argue that the clique construction loses the non-dyadic information represented by the hyperedges. Another, slightly different way of looking at the hypergraph is using the edge multiplicity between two nodes as the weight on the edge between those nodes in the clique construction. This conversion we call as *wt-clique construction*.

The following is a toy example where we look at a small network so that we can easily measure influence of each node. This example shows that the traditional computations on hypergraphs indeed give away some information.

Example 1.

Let $H_1 = (V, E_1)$ be the hypergraph with $V = \{i, j, k, l\}$ and
$E_1 = \{\{i, j\}, \{i, k\}, \{j, k\}, \{i, k, l\}, \{i, j, k\}, \{j, l\}\}$ and $H_2 = (V, E_2)$ be another hypergraph with
$E_2 = \{\{i, j, k\}, \{i, j, k, l\}, \{i, k\}\}$.

These two hypergraphs can be thought of as two collaboration networks where i, j, k, l are authors and each edge represents a document the authors have written. For example, in the first network (given by H_1), the first edge i, j represents a document written by author i and j together. Notice here, in both the networks each author has written a document with every one else but the number of times two authors have written a document together is different. For example, in the second network (given by H_2) i has written one document with l and three documents with k. Let us assume we want to measure the centrality of nodes in these two networks to calculate fluency of information propagation between two authors. Our philosophy is that it must be easy to propagate information from k to i than from l to i.

If we convert the hyperedges of H_1 and H_2 to clique (clique construction) for both H_1 and H_2, we get the same graph, a 4-clique. Apparently, we lose the edge multiplicities. If we put the multiplicities as edge weights in the clique construction (wt-clique construction), both H_1 and H_2 again translate into same graph as in figure 1, though the above two hypergraphs are not the same.

Here we propose three different reduced representations of hypergraph which use the non-dyadic information of the hyperedges efficiently and show how ranking of nodes changes according to their influence on the network.

As stated before, finding maximum spread is a hard problem. [13] uses a game theoretic approach to find the ranking of the nodes which is comparable to that given by the natural greedy algorithm[12] but is much more time efficient that the greedy algorithm [1]. It captures the marginal contribution that each player makes to the dynamics of the game. Consider a cooperative game with transferable utilities, (N, ν), where $N = 1, 2, \ldots, n$ is the set of players and $\nu : 2^N \to \mathcal{R}$ is a characteristic function, that assigns a value to each coalition (subset of N) , with $\nu(\phi) = 0$. The Shapley value (SV) of player $i \in N$, is

$$SV(i) = \frac{1}{n!} \cdot \sum_{\pi \in \Omega} \nu(S_i(\pi) \cup \{i\}) - \nu(S_i(\pi)) \qquad (1)$$

where Ω is the set of all permutations over N , and $S_i(\pi)$ is the set of players appearing before the ith player in permutation π.

We use the same game (*Topk Game*) that is described in [13]. Then we extend the game for multi-hop on weighted graphs as done in *Topk Game with weights* in [1]. We measure influence of a node using Shapley Value (SV) of the node as in [13, 1]. This computation is fast on graphs. As shown in [1], it is linear in number of edges and vertices. We give an equally efficient algorithm for computing ranking on hypergraphs. We compute these ranking using clique construction; wt-clique construction and three of our proposed methods and compare the results using diffusion models.

The main contributions of this work are: (1) Different methods of reducing hypergraphs to simpler forms for the purposes of centrality computations. These reductions could possibly yield more efficient mechanisms for computing other network properties; (2) Computation of game-theoretic centrality measures on super-dyadic relations. These have been shown to be efficient to compute on simple graphs. And, this extension enables us to compute centrality scores on large super-dyadic networks; and (3) Empirical study of how centrality scores are distributed over the nodes of the network under different graph constructions. We establish that traditional constructions of simple graphs from hypergraphs lead to inefficient centrality distributions. We also introduce the notion of dominance for comparing various centrality measures that allows us to identify which measures are truly different in their ranking of the nodes.

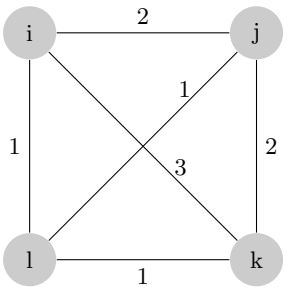

Figure 1: wt-clique construction for H_1 and H_2: $W(H_1)$ and $W(H_2)$ are the same

2. REDUCED REPRESENTATION OF HYPERGRAPH

We already talked about two types of reduced graph formation from hypergraphs viz, clique construction and wt-clique construction. We give three other constructions which, we will show, do better in terms of centrality measurement based on SV. For completeness we briefly describe clique construction and wt-clique construction as well.

2.1 Clique Construction

This is also known as primal of the hypergraph. Let $H = (V(H), E(H))$ be a hypergraph with the vertex set $V(H)$ and the edge set $E(H)$. A hyperedge $e \in E(H)$ is a subset of $V(H)$. The primal graph, also called the Gaifmann graph is defined as follows, $P(H) = (V', E')$ such that $V' = V(H)$ and $E' = \{\{u, v\} \mid e \in E(H)\ and\ u, v \in e\}$
This is also known as 2-section of hypergraph. Hypergraphs are represented as its primal graph in many problems like partition connectedness, betweenness computation etc.

2.2 Wt-Clique Construction

Let $H = (V(H), E(H))$ be a hypergraph with the vertex set $V(H)$ and the edge set $E(H)$. we say, $u, v \in V(H)$ share x hyperedges if there are x hyperedges, $e_1, e_2, \ldots, e_x$ such that $e_i \in E(H)$ and $u, v \in e_i \forall i \in \{1, 2, \ldots, x\}$.
Let $W(H)$ be the reduced representation of H using wt-clique construction. $W(H) = (V', E', w)$ such that $V' = V(H)$ and $E' = \{\{u, v\} \mid e \in E(H)\ and\ u, v \in e\}$ *and* $w : E' \to \mathcal{R}$ *with* $w(u, v) =$ number of hyperedges u, v share.
A similar kind of approach is used in [4] for computing betweenness centrality on hypergraphs where they call it one mode projection of hypergraph.

2.3 Line Graphs

This is also known as the dual of the hypergraph. Let $L(H)$ be the line graph of the hypergraph, H. $L(H) = (V', E')$, where $V' = E(H)$ and $E' = \{\{e_1, e_2\} \mid e_1, e_2 \in E(H), e_1 \cap e_2 \neq \phi\}$

2.3.1 Why Line Graph

We expect the line graph will give a better centrality measure than the clique construction on hypergraph. This is because, in clique construction we lose some information like edge multiplicities of the vertices. Observe, even in wt-clique construction where edge multiplicities do appear as weights on the edges, we do not consider whether two edges are contribution of two different hyperedges or the same hyperedge. For instance in *Example 1*, consider the edges (j, l) and (i, j) in the weighted clique construction of graphs H_1 and H_2 that is $W(H_1)$ and $W(H_2)$ respectively. The hyperedge $\{i, j, k, l\} \in E_2$ is responsible for the edge between j and l in $W(H_2)$. The same hyperedge adds to the weight of the edge between i and j in $W(H_2)$. But in $W(H_1)$, (j, l) is present because of the hyperedge $\{j, l\} \in E_1$. And the same hyperedge is not responsible for the edge $(i, j) \in W(H_1)$. To summarise, it can not be said from $W(H_1)$ and $W(H_2)$ whether the same hyperedge is responsible for two different edges in the wt-clique.
The above observation is crucial in centrality measurement because from $W(H_1)$ and $W(H_2)$ it can be said that i, j and k are equally central for both H_1 and H_2. Though this can be true for H_1 but for H_2, it can be clearly seen that nodes i and k appear more often than j.Later our results on line graph shows that this is correctly predicted by the computation using $L(H)$.

Let us now look at the line graph construction of the two hypergraphs of *Example 1*. This gives us some insight why Shapley Value measure computed on hypergraph using line graph can be more accurate compared to the one computed using clique or wt-clique construction. Figure 2 shows the line graphs for H_1 and H_2.

2.4 Weighted Line Graph

We also look at weighted line graph where weight on each edge is the size of the intersection between the two hyperedges. We compare the centrality measure with other reduced representations of hypergraph and show which gives a better measure for centrality.
Let $L_w(H)$ be the weighted line graph for H. $L_w(H) = (V', E', w)$, where $V' = E(H)$, $E' = \{\{e_1, e_2\} \mid e_1, e_2 \in E(H), e_1 \cap e_2 \neq \phi\}$ and $w : E' \to \mathcal{R}$, w is defined as follows. Weight of an edge $\{e, e'\} \in L_w(H)$ is $w(\{e, e'\}) = \frac{1}{|e \cap e'|}$, where $e, e' \in E$.
This representation keeps the hyperedges intact as in the line graph representation. Also it uses the information that how many nodes are common between two hyperedges. Two hyperedges which has more common nodes are closer. That is, it is easier to pass some information since more nodes (rational agents) can be involved in passing the information.

2.5 Multi-Graph

This is one variant of clique construction for hypergraph. Here, instead of putting the edge multiplicities as edge weights, multiple edges are kept between two nodes.
Define multi-graph of hypergraph, H, $M(H) = (V', E')$ such that $V' = V(H)$ and E' is a multiset, $E' = \{\{u, v\} \mid e \in E(H)\ and\ u, v \in e\}$. If two vertices $u, v \in V(H)$ share x hyperedges in H then the multi-graph of H, $M(H)$ has x edges between u and v.
Though this construction is similar to wt-clique construction, for example, both H_1, H_2 of *example 1* have same multi-graph construction, we will show it fares better in SV computation using the definition of fringe[1] (value of a coalition) given in the next section. Proof of the next claim given in the following section gives an intuitive idea why it performs better than $W(H)$ or sometimes even $L(H)$.

CLAIM 1. *If u, v belongs to many hyperedges then they marginally affect each other more in case of $M(H)$ compared to $W(H)$.*

3. SHAPLEY VALUE BASED CENTRALITY MEASURE ON HYPERGRAPH

We try to find the centrality measure on hypergraph using the concept of Shapley Value. But we define the game on $L(H)$ instead of H and find a ranking for the nodes of $L(H)$. That is, we first measure the centrality of each of the hyperedges than the vertices then translate it for the vertices of H. We have to maintain collective rationality and individual rationality when converting the Shapley values of the hyperedges to vertices of hypergraph. Let x_i be the value player i gets. Then,

$$\sum_{i=1}^{|N|} x_i = \nu(N)\ and\ \forall\ player\ j, x_j \geq v(\{j\}) \qquad (2)$$

must hold for the nodes of hypergraph. The value of the grand coalition, $\nu(N)$ is 1.
To achieve this, we normalise the Shapley values we get for the hyperedges and produce normalised values for the vertices.

We model the games presented in [1] for hypergraphs. Then we compute the Shapley value for these games when they are played on line graph and weighted line graph of the

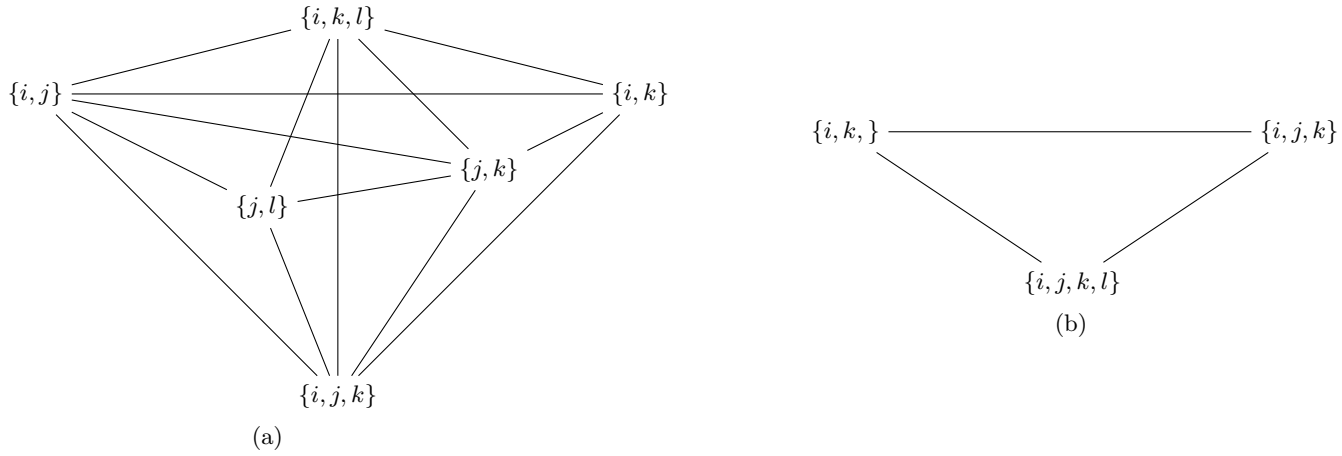

Figure 2: (a)$L(H_1)$ (b)$L(H_2)$

hypergraph.
Let H = (V,E) be a hypergraph. Define the neighbourhood of a hyperedge, $e \in E$ in the line graph of H, $L(H)$ as follows,

$$Ng(e) = \{e' \mid e' \in E, e \cap e' \neq \phi\} \tag{3}$$

We use the following equation to translate the values to the nodes of H from the Shapley value $sv_L(.)$ of the hyperedges. For node $v_i \in V$

$$sv(v_i) = \sum_{e \in E, v_i \in e} \frac{sv_L(e)}{|e|} \tag{4}$$

Next, we briefly introduced two games, one on unweighted graphs and another on weighted graphs (Game 1 and Game 3 from [1]). These games are defined over undirected network graphs. Then, we describe how we use these two games in case of hypergraphs.

3.1 TopK Game: $\nu_1(C)$ = #agents at-most 1 hop away

Given an unweighted, undirected network $G(V, E)$. Fringe of a subset $C \subseteq V(G)$ is defined as the set
$\{v \in V(G) : v \in C(or) \exists u \in C \; such \; that \; (u, v) \in E(G)\}$.
Based on the fringe, the cooperative game $g1(V(G), \nu_1)$ is defined with respect to the network $G(V, E)$ by the characteristic function $\nu_1 : 2^V(G) \to \mathcal{R}$ given by

$$\nu_1(C) = \begin{cases} 0 & if \; C = \phi \\ size(fringe(C)) & otherwise \end{cases} \tag{5}$$

Using g1 on line graph, $L(H)$ Shapley value $i \in E$ is given as follows,

$$sv_L(i) = \sum_{j \in E, i \cap j \neq \phi} \frac{1}{1 + |Ng(i)|} \tag{6}$$

This is computed using algorithm 1 of [1] which runs in $O(V + E)$ and finds SV for each of the hyperedges.

3.2 TopK Game with Weights: $\nu_3(C)$ = #agents at-most d_{cutoff} away

This is an extension of TopK game for weighted undirected networks $G(V, E, W)$, where $W : E \to \mathcal{R}^+$ is the weight function. Here the fringe of a set, $s \subseteq C$ is defined as the vertices which are atmost d_{cutoff} away from some node in s. Let $g3(V(G), \nu_3)$ be the game defined on the network, $G(V, E, W)$. The definition of fringe gives the following definition for ν_3,

$$\nu_3(C) = \begin{cases} 0 & if \; C = \phi \\ size(\{v_i : \\ \exists v_j \in C(or) distance(v_i, v_j) \leq d_{cutoff}\}) & otherwise \end{cases} \tag{7}$$

for each coalition $C \subseteq V(G)$. [1] gives an exact formula for SVs using ν_3. However, in this case the algorithm for implementing the formula has complexity $O(VE + V^2 log(V))$.

For weighted line graph, $L_w(H)$, g3 is used to find the Shapley Value of the hyperedges.
For TopK Game with Weights as in [1], we define the extended neighbourhood as follows,

$$extNg(e) = \{e' \mid d_{ee'} < d_{cutoff}\} \tag{8}$$

where, $d_{ee'}$ = shortest path distance between e and e' in $L_w(H)$.

Shapley value of $i \in E$ in $L_w(H)$ is

$$sv_L(i) = \sum_{j \in \{i\} \cup extNg(i)} \frac{1}{1 + |extNg(i)|} \tag{9}$$

As we see in the final equations, the marginal contribution of each node depends mainly on its neighbourhood. When looking at clique construction, any two vertices which share a common edge are neighbours. Line graph gives a dual of this relation. So it is important to find the actual neighbourhood for each node in the line graph that can influence the marginal contribution of a node of the line graph. Here what we mean by actual neighbourhood is the neighbours who can influence or take part in the computation of marginal contribution of this node, in other the words, neighbours who can change the node's influence over the network. The physical inference of Equation 7 tells the same notion mathematically. Next we analyse the neighbourhood for line graph and give an analogous equation which is more appropriate.

3.2.1 Neighbourhood Analysis: Line Graph

Instead of considering the degree as the size of the neighborhood we define volume that is analogous to degree for the nodes of line graph $L(H)$. Volume of a node $v, vol(v)$

is defined as the sum of weights of its neighbours. And, we do not consider a node in a line graph has a contribution of one, instead we take the cardinality of the corresponding hyperedge. Let us explain why a different definition of size is required.

Let us assume e_1, e_2, e_3 be three different edges of the hypergraph, H such that $i, j \in e_1$, $k, j \in e_2$ and $j, l \in e_3$. Let $v_{l_1}, v_{l_2}, v_{l_3}$ be the vertices of $L(H)$ corresponding to e_1, e_2 and e_3 respectively. Using Equation 6, each node in neighbourhood of v_{l_2} reduces the probability that v_{l_1} brings v_{l_2} in the coalition. Hence reduces the marginal contribution of v_{l_1}. So, v_{l_3} reduces MC of v_{l_1}.

Observe, a vertex in L(H), corresponding to a hyperedge with large cardinality or having influential nodes of H, gets high Shapley value and the other vertices which are connected to it gets comparatively low Shapley value. If l is an influential node that means v_{l_3} becomes more influential lowering the influence of v_{l_1}. Therefore, though l is not directly connected to i it can reduce SV of i. To eliminate this, we define $vol_i(.)$ as the size of intersection of a vertex of $L(H)$ with i. To maintain consistency, instead of counting i as 1 as in Equation 6, we take the contribution of i to the coalition as the size of i.

What did we say intuitively in the above argument? Did we say that the clique construction gives the correct neighbourhood? No, We showed that the neighbourhood must depend not only on how the vertices share an edge, if they do share, but also how many edges they share and most importantly which vertices are connecting two edges. This gives us the basic idea why line graph construction works better.

Therefore $vol : 2^{V(G)} \rightarrow \mathcal{R}$ and is defined as, $vol(v) = \sum_{u \in Ng(v)} \mid u \cap v \mid$. We propose the following equation based on the above argument.

$$sv(i) = \sum_{v \in Ng(i)} \frac{1}{vol(v) + size(v)} \tag{10}$$

3.3 Games with Multi-Graph

3.3.1 Proof of Claim 1

PROOF. If two agents have been together in k hyperedges then they contribute k times to each others degree. Let us assume there are k edges between u and v. Further assume, w is connected to v by an edge. Since marginal contribution (MC) depends inversely on the degree of the neighbors, probability that w brings v into a coalition is less in this scenario compared to the case when u and v are connected by a single edge. Thus, adding more to the degree of a neighbor decreases the MC of any other node which are less strongly connected to it. Hence it increases the expected MC of node u. □

3.3.2 TopK Game on Multi-Graph

As stated in the beginning, multi-graph construction is similar to wt-clique construction. So the Shapley value of $v_i \in V(H)$ using TopK Game from [1, 13] on M(H) is given in the following equation.

$$SV_{topk}(v_i) = \sum_{v_j \in \{v_i\} \cup N_{M(H)}(v_i)} \frac{w(v_i, v_j)}{1 + deg_{M(H)}(v_j)} \tag{11}$$

where $w(v_i, v_j)$ is the edge multiplicity between v_i and v_j, $N_{M(H)}(v_i)$ is the neighbourhood of v_i in $M(H)$ which is same as neighbourhood of v_i in H and $deg_{M(H)} = \mid N_{M(H)}(v_i) \mid$.

4. DIFFUSION

Let $f(S)$ be the expected number of nodes active (influenced) at the end if set S is targeted for initial activation. For example S can be the top k nodes,for some constant k. We pick this top k from the ranking we computed, which are not directly connected by an edge as done in the spin algorithm[13]. We want to maximise $f(S)$.

This optimization problem is NP-hard as proved in [12]. Hence we look for an approximate solution for this problem. Kempe et al.[12] generalised two diffusion models viz, Independent Cascade and Linear Threshold. They showed that greedy hill climbing algorithm using generalisation of both the models gives $(1 - 1/e - \epsilon)$ approximation. In fact, as shown by them, both the generalised models are equivalent.

To compare the rankings we get for the nodes of H using $P(H)$, $W(H)$, $L(H)$, $L_w(H)$ and $M(H)$, we use these two methods. These models only look at the rankings and not how the scores are distributed among the nodes. That is useful when the scores do not have any other inference. We come up with a heuristic using the concept of dominance from cooperative games to compare two set of SVs we get by different reduced representations. Dominance is a fundamental concept in cooperative game theory and is used for deciding whether a vector,x from the imputation space is preferred over another vector, y from the same imputation space.

4.1 Measure of Diffusion using Game Theory

The solution concepts in cooperative games give a measure of power (centrality score) of each player. Imputation Space is the set $I_v = \{x \in \mathcal{R}^n \mid x = (x_1, x_2, \ldots, x_n), \sum_{i=1}^{n} x_i = 1, x_i \geq 0\}$. It is the set of all possible assignment of values to each of the players. Since this set can contain uncountably many vectors we try to find a smaller set. We use the rules of the game to get a smaller set such that each vector from this subset assigns some meaningful values to the players. Shapley Value is the solution concept that we are using in our work to find the centrality scores (power) of the players. Marginal contribution network enables us to find the Shapley Value scores in polynomial time[1]. Now, we want to use the idea of dominant imputation to show which method is better in assigning the scores.

4.1.1 Motivation

Idea: Core is another solution concept like Shapley value to measure power of each of the agents. Finding core is a computationally hard problem too. It is defined as follows.

Definition 1. Core.
Core is the set of all undominated imputation.

Let $x, y \in I_v$, I_v = set of all imputations.

Definition 2. Dominance.
We say $x \succ y$ (x dominates y or x is preferred to y), if there exists a non-empty coalition $S \subseteq N$ such that $\forall i \in S, x_i > y_i$ and $\sum_{i \in S} x_i \leq v(S)$

Where N is the set of all players and $v(S)$ denotes the value of the coalition S.

Checking if an imputation (centrality score) is in the core is NP-hard[14](reduction from vertex cover) even with MC-Net representation. So instead of checking if an imputation is undominated, the domination of the imputation has been

checked. Domination is easy to compute when we know two imputations (say x and y as in the definition). If x dominates y and the agents who are getting more score in x are amongst the top agents according to both the imputations then x correctly assigns high score to the important nodes and low score to less important nodes since value of grand coaltion is 1.

4.1.2 Why We Need

When we know some vectors from the core or we know what is the best way to divide the value of the grand coalition, we have an optimal possible imputation. We can compute which centrality score is better between two of them by comparing them with the optimal one (undominated imputation). Precisely, we look at how much each deviates from the optimal one, co-ordinate wise. But in lack of that standard or optimal measure we might want to compare two rankings(imputations) to check which one is giving a better estimate of ranking. Here is where the idea of dominance can be used.

4.1.3 Dominance to Diffusion

Let X and Y are two imputations. S be a set of agents such that $\forall i \in S\ x_i > y_i$. We can check the marginal contribution of x_is in general over the sum of the values given to x_is in the said imputation. If marginal contribution is these x_is are high and X assigns high score to these x_is, that implies X is more appropriate measure than Y. Computing diffusion with general diffusion model is hard. This gives us an efficiently computable method to compare two centrality measures. But since domination is not transitive to compare n different measures dominance need to be checked for $O(n^2)$ times.

5. RESULTS

We use the five different reduced representations of a hypergraph described earlier and compute the centrality of the nodes in each reduced representation. We performed our experiments on three datasets, viz., JMLR, Citeseer, and Cora. The hypergraph on the JMLR dataset is a co-authorship network where each hyperedge corresponds to a paper published in JMLR and links the authors of that paper. The other two are co-citation network where all the papers cited in a particular paper form a hyperedge. As mentioned before we compare the five methods using (i) two diffusion models[12], Linear Threshold and Independent Cascade and (ii) the concept of dominance from co-operative game theory.

We performed three different experiments on the different datasets: (1) As done in [13], we use top k nodes with spin algorithm to find diffusion using Linear Threshold and Independent Cascade, where, k is our budget of how many nodes we activate initially. We compared the different centrality scores on the extent of the diffusion at the end of the process.

(2) Let us assume, we do not want to care about the budget but want to activate atleast p% of the nodes at the end of diffusion. Say, p is 90%. To achieve this we fix a SV, *val* such that all nodes having SV more than *val* is activated initially. We call *val*, the limiting SV. We observe as we decrease the limiting SV p increases rapidly in the beginning but after a certain value of limiting SV change in p in negligible. We also observe that for networks with thousand nodes the limiting value is 10^{-4} for $p = 85$.

Method	H_1	H_2	Summary
Unweighted Clique	i, j, k, l	i, j, k, l	Gives equal importance to all the vertices where as l is much less central than i or k
score	.25, .25, .25, .25	.25, .25, .25, .25	
Weighted Clique	k, j, i, l	k, j, i, l	
score	.27, .27, .27, .18	.27, .27, .27, .18	
Line Graph	j, k, i, l	k, i, j, l	correctly assigns lower fraction to l
score	.33, .29, .29, .08	.4, .4, .15, .03	
Weighted Line Graph	k, i, j, l	k, i, j, l	correctly assigns lower fraction to l
score	.31, .31, .3, .08	.4, .4, .15, .03	
Multi-Graph	k, i, j, l	k, i, j, l	note the difference of sore from wt-clique construction
score	.28, .28, .24, .17	.28, .28, .24, .17	

Table 1: No of nodes influenced for example 1

Dataset	No of nodes	No of edges	No of connected components	Size of largest component
JMLR	3217	1159	11	2275
Citeseer [15]	3312	2240	438	2110
Cora [15]	2708	2223	78	2485

Table 2: Description of the Datasets Used

(3) We examined how the SV is distributed as function of the ranking of the nodes. If a larger fraction of the SV is concentrated on the higher ranked nodes, then the measure discriminates better between the truly important nodes and the rest.

Most influential nodes can be prominently identified by any of the methods.But as we try to extend the reach of diffusion, we increase k that is we activate more nodes in the beginning. We observed that the top 10 nodes we get employing different methods have at least 70% intersection. But when we look at how much power has been assigned to each of these nodes the methods differ largely.

5.1 Example 1

Table 1 shows the empirical results that we get for the five mentioned reduced representation of hypergraph for the hypergraphs of *Example 1*. It shows the distribution of score, for the four nodes, produced by the five methods. It clearly shows that the ranking using $P(H)$ (unweighted clique) wrongly assigns same score to all the nodes. Also, observe the difference in scoring pattern of the methods $W(H)$ (weighted clique) and $M(H)$ (multi-graph).

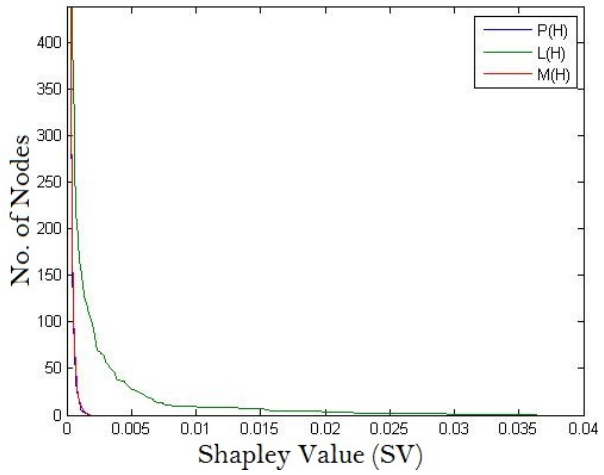

Figure 3: Compare SV distribution on JMLR Data

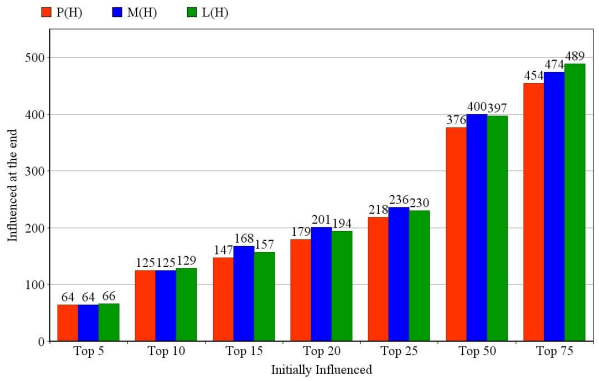

Figure 4: Diffusion using P(H), M(H) and L(H) on JMLR Data

Table 2 describes the three datasets on which we have performed the experiments. But before looking at the results on the datasets, let us look at the results on Example 1 that we introduced in introduction.

5.2 JMLR Data

From JMLR data we extract the hypergraph of the co-authorship network that is, in this network each node represents an author and each paper they published is a hyperedge consisting of the authors of that paper. Statistics of the network is given in table 2 in row 1.

We compare the distribution of SV among the nodes, computed from the JMLR hypergraph using the five mentioned representations, in figure 3. It plots the number of nodes, having SV more than a particular value v, vs v. The trend that we observe in figure 3 is that, the Shapley value computation through line graph always assigns very low (almost zero) value to powerless nodes where as the value, for important nodes go as high as 0.01. On the other hand, the highest value assigned to the most important node using P(H) or even M(H) is 0.0016 and 0.0018 respectively. The idea of dominance clearly shows this difference.

Let us look at the top nodes given by $P(H)$, $L(H)$ and $M(H)$. We apply Linear Threshold model to measure diffusion. In Linear Threshold model, threshold for each node is kept as 1/3 of degree of that node and we find the diffusion when top 10 nodes are activated initially. The number of nodes that are activated in the beginning is written in brackets in the heading of the columns in the table. The result is shown in figure 4. Notice, when top 15 or top 20 nodes are activated in the beginning $M(H)$ outperforms $L(H)$ but as we take more nodes for initial activation the green bar for $L(H)$ stands taller.

5.3 Citeseer Data

It is a co-citation network that is, each publication is a node and the authors who co-authored a publication form an hyperedge. Statistics of the network is given in table 2 in row 2.

Table 3 compares three of our proposed single graph representations of hypergraph with primal graph of hypergraph, $P(H)$ and $W(H)$.We followed spin algorithm[13] for picking

Graph Representation	Linear Threshold (top 10)	Independent Cascade (top5)	Independent Cascade (top10)	time (ms)
$P(H)$	698	832	1002.13	754
$W(H)$	627	800.86	1039.12	2583
$L(H)$	709	837.68	914.15	1573
$L_w(H)$	691	833.34	933.73	3042
$M(H)$	698	837.68	966.04	751
$P(H)$ (Betweenness)	625	901.55	951.81	18520
$L(H)$ (Betweenness)	664	791.35	1009.29	15731

Table 3: Diffusion using Linear Threshold and Independent Cascade model on Citeseer

up the top k nodes, for initial activation, from the rankings produced by the five methods. As stated earlier, it is easier to find out nodes which are important (e.g., the top node or top 5 nodes). Here, both $L(H)$ and $M(H)$ gives the same set of top 5 nodes. Table 3 shows the results of between-ness centrality too, when found using primal graph and our proposed line graph method. For between-ness too the performance of line graph is better both in terms of time requirement and ranking. Computations on line graph takes the advantage of the fact that in general these hypergraphs are sparse compared to general graphs.

Figure 5 compares the SV assigned to nodes using $L(H)$ and $M(H)$. It plots number of nodes, having Shapley Value above v, vs v. The most important node gets SV 0.2 and 0.0089 using $L(H)$ and $M(H)$ respectively. In the figure the purple line for $M(H)$ is almost grounded to $x - axis$ after 2×10^{-3} where as many nodes have been assigned SV more than 5×10^{-3} by $L(H)$. Figure 6 compares the distribution of SV assigned to the nodes by all the five methods. Table 4 gives an analogous analysis in tabular format. It also puts together the number of nodes activated at the end when all

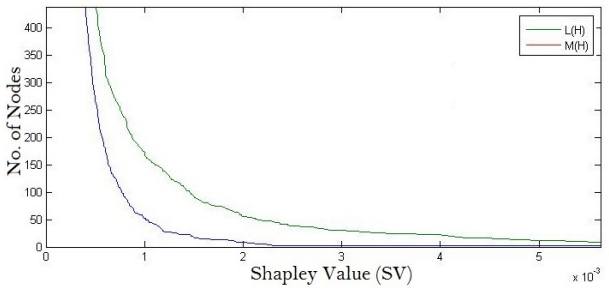

Figure 5: Compare SV distribution on Citeseer Data: L(H) vs M(H)

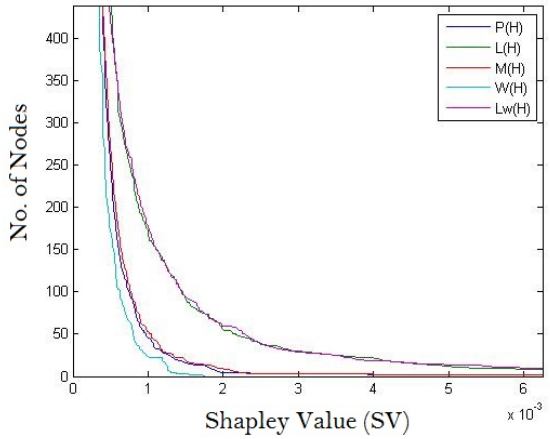

Figure 6: Compare SV distribution on Citeseer data

nodes, having SV above the limiting value v, are activated initially. For example, when the sum of Shapley value of the nodes picked is fixed at 0.07273128, we get only 3 nodes by $L(H)$ compared to 68 nodes by $M(H)$ and the limiting value is only 0.01 for $L(H)$ compared to 6.14×10^{-4} for $M(H)$. The sum of SV of top 1022 nodes, chosen by spin algorithm, manages to reach 0.3421102 only for $M(H)$.

If a method M_1 dominates another method M_2 via x nodes, more value of x implies stronger M_2, less value of x implies M_2 has not assigned more power to more important nodes when compared to M_1. Observe table 5, it takes only 26% nodes in favour of $L_w(H)$ when dominating $M(H)$ compared to 62% when $L(H)$ dominates $L_w(H)$. This is because, important nodes are assigned less score using $M(H)$. So, they go in favour of $L_w(H)$. But, when comparing $L_w(H)$ to $L(H)$ all the important nodes are assigned high value so it takes more nodes to sum up to a score of 0.609.

5.4 Cora Data

It is again a co-citation network. Statistics of the network is given in table 2 in row 3.
Total number of nodes activated at the end when top 10 nodes are activated at the start is shown in figure 7 for the five methods discussed. The diffusion model employed here is linear threshold. Table 7 gives comparison of three of our proposed representations with $P(H)$ and $W(H)$ for the co-

Method	Dominates	Dominated By
$L(H)$	$M(H)$ via 660 nodes of total score 0.685, $P(H)$ via 702 nodes of total score 0.705	None
$L_W(H)$	$M(H)$ via 717 nodes of total score 0.709, $P(H)$ via 748 nodes of total score 0.728	$L(H)$ via 1695 nodes of total score 0.609

Table 5: Comparison of Diffusion Using Dominance on Citeseer

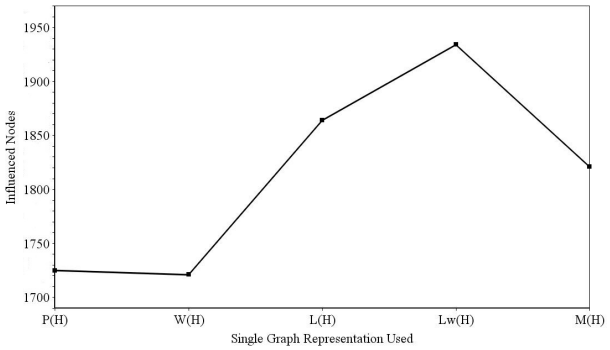

Figure 7: Diffusion using linear threshold model on Cora data

citation network from Cora Data. It gives the diffusion for between-ness too, when computed using $P(H)$ and $L(H)$. Performance of later is better when compared with respect to diffusion. Notice, since this hypergraph is not as sparse as in the case of Citeseer, the computation time when using $L(H)$ is more compared to $P(H)$ in case of between-ness measurement.

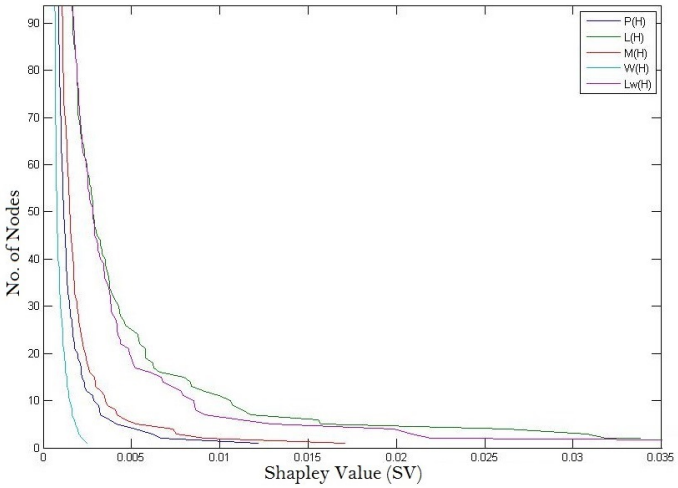

Figure 8: Compare SV distribution on Cora data

	Using Line Graph				Using Multi-Graph			
Sum of SV	limiting value of SV	Nodes Started with	Diffusion (Linear Threshold)	Diffusion (Independent Cascade)	limiting value of SV	Nodes Started with	Diffusion (Linear Threshold)	Diffusion (Independent Cascade)
0.4164433	10^{-6}	990	3067	2788.81				
0.41616583	10^{-5}	945	3060	2775.0				
0.4039153	10^{-4}	644	2864	2629.12	7.47112E-5 (sum 0.3421102)	1022	3107	2831.49
0.22036676	0.001	84	1216	1787.81	3.02×10^{-4}	472	2518	2438.02
0.07273128	0.01	3	608	1596.57	6.14×10^{-4}	68	1212	1757.99
0.04144367	0.05	1	541	1573.94	8.87×10^{-4}	24	769	1622.98

Table 4: Diffusion: Linear Threshold and Independent Cascade on Citeseer Data

	Using Line Graph				Using Multi-Graph			
Sum of SV	limiting value of SV	Nodes Started with	Diffusion (Linear Threshold)	Diffusion (Independent Cascade)	limiting value of SV	Nodes Started with	Diffusion (Linear Threshold)	Diffusion (Independent Cascade)
0.37252015	10^{-6}	752	2686	2489.74				
0.371984	10^{-5}	678	2678	2476.13				
0.35889158	10^{-4}	309	2639	2374.86	1.087832E-4 sum 0.2821837	722	2686	2489.13
0.27076006	0.001	49	2255	2186.8	1.5×10^{-4}	636	2686	2478.1
0.14932592	0.01	4	994	2127.4	3.96×10^{-4}	153	2466	2254.25
0.09491563	0.05	1	416	2128.83	7.59×10^{-4}	51	2227	2168.42

Table 6: Diffusion: Linear Threshold and Independent Cascade on Cora data

Figure 8 gives the SV distribution for different methods. The y-axis gives the number of nodes having Shapley Value more than the value of x-axis. The line corresponding to the SV distribution produced using $W(H)$ ends even before 0.005 which means no nodes has been assigned value above 0.005. Where as the lines for $L(H)$ and $L_w(H)$ goes beyond 0.05 and has many nodes with values more than that. This depicts the important nodes are also assigned low values by $W(H)$, $P(H)$ and even $M(H)$. Therefore it is hard to conclude which are the more powerful node from the scores in case of $W(H)$, $P(H)$ and $M(H)$. A similar analysis between $L(H)$ and $M(H)$ is shown in table 6 where the corresponding diffusion values are also given.

As we analysed for Citeseer Data, observe from Table 8, it takes only 14% nodes in favour of $L_w(H)$ when dominating $M(H)$ compared to 61% when dominating $L(H)$. This is because, important nodes are assigned less score using $M(H)$ so they go in favour of $L_w(H)$. But when comparing $L_w(H)$ to $L(H)$, all the important nodes are assigned high value so it takes more nodes to sum upto a score of 0.877 that can give preference to $L_w(H)$. Since, we do not care about these less important nodes, we can safely employ any one of $L(H)$ or $L_w(H)$ to get the ranking as well as the Shapley value based scores.

Graph Representation	Linear Threshold Threshold (top 10)	Independent Cascade (top 10)	Independent Cascade (top 5)	time
$P(H)$	1725	1654.73	1588.34	840
$W(H)$	1721	1670.13	1596.37	7998
$L(H)$	1864	1656.82	1625.5	7981
$L_W(H)$	1934	1685.98	1625.5	16438
$M(H)$	1821	1659.83	1598.67	2523
$P(H)$ Betweenness	1831	1663.82	1606.36	12226
$L(H)$ Betweenness	1934	1680.68	1623.73	22932

Table 7: No of nodes influenced on Cora data

6. CONCLUSION

We established that computation of game theoretic centrality on hypergraphs using the clique construction indeed loses some information which is preserved by line graph construction. The results also demonstrate that the clique construction does not perform well compared to Weighted Line Graph and Multi-Graph construction. Looking at the results on all the three datasets we conclude that Weighted Line Graph and Multi-graph reductions of a hypergraph lead

Method	Dominates	Dominated By
$L(H)$	$M(H)$ via 403 nodes of total score 0.845, $P(H)$ via 442 nodes of total score 0.873, $W(H)$ via 457 nodes of total score 0.891	$L_w(H)$ via 1669 nodes of total score 0.877
$L_W(H)$	$M(H)$ via 399 nodes of total score 0.853, $P(H)$ via 443 nodes of total score 0.887, $W(H)$ via 493 nodes of total score 0.909, $L(H)$ via 1669 nodes of total score 0.877	None

Table 8: Comparison of Diffusion Using Dominance on Cora

to more meaningful centrality scores.
It would be an interesting line of work to investigate if these two reductions of a hypergraph yield better algorithms/ computations in other domains, like, community detection. While it was more or less straightforward to extend the topK game to directed graphs, it is not clear how to extend it to directed hypergraphs. In the context of biological networks, directed hypergraphs are a more natural model and hence exploring the use of these centrality measures in directed hypergraphs is also a promising future line of investigation.

7. REFERENCES

[1] K. V. Aadithya, B. Ravindran, T. Michalak, and N. R. Jennings, "Efficient computation of the shapley value for centrality in networks." in *Proceedings of the Sixth Workshop on Internet and Network Economics (WINE)*. Springer-Verlag, 2010, pp. 1–13.

[2] P. Bonacich, A. C. Holdren, and M. Johnston, "Hyper-edges and multidimensional centrality," *Social Networks*, vol. 26, pp. 189–203, 2004.

[3] C. Bothorel and M. Bouklit, "An algorithm for detecting communities in folksonomy hypergraphs." in *IICS*, ser. LNI, vol. P-186. GI, 2011, pp. 159–168.

[4] R. Puzis, M. Purohit, and V. Subrahmanian, "Betweenness computation in the single graph representation of hypergraphs," *Social Networks*, vol. 35, pp. 561–572, 2013.

[5] K. Faust, "Centrality in affiliation networks," *Social Networks*, vol. 19, pp. 157–191, 1997.

[6] C. Berge, *Hypergraphs: Combinatorics of Finite Sets*. North-Holland, 1989.

[7] D. Zhou, J. Huang, and B. Schölkopf, "Learning with hypergraphs: clustering, classification, and embedding." in *NIPS*. MIT Press, 2006, pp. 1601–1608.

[8] A. Lungeanu, M. Murase, D. Carter, and N. Contractor, "A hypergraph approach to understanding the assembly of scientific research teams." in *Sunbelt XXXII conference, Redondo Beach, CA*, 2012.

[9] M. Zhu, A. Wax, L. DeChurch, and N. Contractor, "Teamwork at the hyper-edge: Impact of team hyperedge structures on performance." in *Sunbelt XXXII conference, Redondo Beach, CA*, 2012.

[10] M. A. Ahmad, B. Keegan, D. Williams, J. Srivastava, and N. Contractor, "Trust amongst rogues? A hypergraph approach for comparing clandestine trust networks in MMOGs." in *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media*, 2011.

[11] S. K. Jain, S. N. Satchidanand, A. K. Maurya, and B. Ravindran, "Studying indian railways network using hypergraphs," in *Proceedings of the Social Networking Workshop at COMSNETS 2014*. Springer, 2014.

[12] D. Kempe, J. M. Kleinberg, and É. Tardos, "Influential nodes in a diffusion model for social networks," in *32nd International Colloquium on Automata, Languages and Programming (ICALP 2005)*, 2005.

[13] R. Narayanam and Y. Narahari, "Determining the top-k nodes in social networks using the Shapley value." in *AAMAS (3)*. IFAAMAS, 2008, pp. 1509–1512. [Online]. Available: http://dblp.uni-trier.de/db/conf/atal/aamas2008-3.html#SuriN08

[14] X. Deng and C. Papadimitriou, "On the complexity of cooperative solution concepts," *Mathematics of Operations Research*, vol. 19(2), 1994.

[15] P. Sen, G. M. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective classification in network data," *AI Magazine*, vol. 29, no. 3, pp. 93–106, 2008.