Weighted Simplicial Complex: A Novel Approach for Predicting Small Group Evolution

Ankit Sharma[1,3], Terrence J. Moore[2],
Ananthram Swami[2], and Jaideep Srivastava[3]
Email: ankit@cs.umn.edu, jsrivastava@hbku.edu.qa
{terrence.j.moore,ananthram.swami}.civ@mail.mil

[1] University of Minnesota, USA
[2] U.S. Army Research Laboratory, USA
[3] Qatar Computing Research Institute, Qatar

Abstract. The study of small collaborations or teams is an important endeavor both in industry and academia. The social phenomena responsible for formation or evolution of such small groups is quite different from those for dyadic relations like friendship or large size guilds (or communities). In small groups when social actors collaborate for various tasks over time, the actors common across collaborations act as bridges which connect groups into a network of groups. Evolution of groups is affected by this network structure. Building appropriate models for this network is an important problem in the study of group evolution. This work focuses on the problem of group recurrence prediction. In order to overcome the shortcomings of two traditional group network modeling approaches: hypergraph and simplicial complex, we propose a hybrid approach: *Weighted Simplicial Complex (WSC)*. We develop a *Hasse diagram* based framework to study WSCs and build several predictive models for group recurrence based on this approach. Our results demonstrate the effectiveness of our approach.

1 Introduction

With the advent of high-speed internet, collaborations are no longer restricted by physical proximity. A group of individuals, irrespective of their demographics or location, can perform a task online. This task might be writing software code (or a Wikipedia article or a Google Doc) by a group of coders (or editors), or can be a business meeting involving video chat with colleagues or collaborations in writing paper [14,19,20,15] or teaming up in online games [13,11,22]. Understanding the dynamics of such small (social) groups is of increasing research interest in various sub-disciplines in the social sciences [9], and is of interest to applications that require high efficiency in the performance of human groups [5,18].

This paper addresses the problem of *group evolution*, with specific focus on understanding the causal factors driving the evolution. The overall objective is to build a model that can predict how a group will evolve in the future, based on its history. One aspect of special interest is *group recurrence*, which can be stated thus: *Which group(s) (or its subgroup(s)) among the groups observed so far, will continue to function as a group, i.e. perform some task again in the near future?*

Prior studies have demonstrated the significance of recurrence in network structure (see [27] and references therein). Most work on group evolution in social networks focuses on the evolution of arbitrary size communities or groups [8]. The sizes of these groups are usually large and the boundaries of the community depends on the definition of membership used. In this paper we study well-defined small groups which typically have size ≤ 20. A key difference between large groups and small groups is that membership in the former is largely based on identity, i.e. a member identifying himself with the group. In contrast, a small group is defined principally by the (regular) interaction between group members, often driven by some purpose, professional or personal. The focus of this paper is to study the evolution of small groups and, in contrast to classical social science literature, the objective is to build models that can predict future behavior, with the final goal of identifying potential causal mechanisms for small group evolution.

In contrast to prior work, we highlight the distinct nature of small groups and develop models inspired from social science theories of small groups [9]. A group can be formed depending on the requirement (fiat teams) or a set of actors can make an autonomous decision to work together (self assembly [3]). In either case, individuals find it easier to work with familiar actors [5], making *frequency* of activity by a group an important metric. Also, over time, actors build new relationships while working in different groups. A shared collaboration history is therefore created, where the same individuals are part of multiple groups, acting as bridges between groups, and resulting in a *network of groups (NOG)* (Figure 1). This is the *network* perspective of small groups [6] where the network of groups plays a central role in the group formation process. Moreover, group formation motives and group communication processes, which are *task centered*, are very different from those involved in building friendship ties in a friendship network or joining a community, e.g., joining a news interest group, being part of a Facebook community, subscribing to a Youtube channel, or publishing within a particular research discipline [21,17,16]. Recently, some attempts have been made to model networks as higher order relational structures such as simplicial complexes [10,7] and hypergraphs [14,12]. A hypergraph is a generalized graph where edges, now called hyperdges, instead of representing a relationship between a pair of vertices, represent a relationship between a set of vertices. If the relationship holds for every subset of the hyperedge, the hypergraph is called a *simplicial complex*. Although hypergraphs are more general, if the problem or the data has a special structure then simplicial complexes are more appropriate. For the *group recurrence* problem, we need to predict recurrence of not just observed groups but also the subgroups. Thus, simplicial complexes are more applicable to our problem

For the *group recurrence* problem we also want our model to capture any prior knowledge associated with each group or subgroup that might indicate cohesion among group members, or the context associated with the group. We use the concept of a *weighted simplicial complex*, which is a simplicial complex where each simplex has a prior weight associated with it. We develop several schemes to generate these prior weights, modeling different prior knowledge scenarios.

We observe that a simplicial complex, from a frequent pattern mining perspective [1], is the trivial set of all the frequent patterns of frequency equal to one, mined from the transactions database of hyperedges. This motivates the use of a Hasse diagram (Figure 1) [23] (similar to enumeration trees in pattern mining) as a graph representation for the simplicial complex. If we associate a weight with each node (representing simplicies) of the Hasse diagram it represents a weighted simplicial complex. We hypothesize that the topology of these groups plays a critical role in how past occurrences influence future occurrences of other (sub)groups.

Using the Hasse diagram, we apply a modification of the HyperPrior algorithm [24], for generating label diffusion-based machine learning models, as well as develop hierarchical label spreading algorithms for recurrence prediction. These algorithms make use of the weighted simplicial complex topology while exchanging the occurrence information between the subgroup nodes in the Hasse diagram. Our experimental analysis, conducted using the DBLP and EverQuest II datasets, shows the efficacy of the techniques developed. The main contributions of this study are:

- We present machine learning models to predict recurrence of already observed groups, which takes into account the higher order topology.
- We present a Hasse diagram-based framework to study simplicial complexes, hypergraphs, and frequent pattern mining in a unified manner.
- We show that frequent patterns can be considered as topological entities, with relationships between them guided by higher-order topological properties. To the best of our knowledge this has not been done before.

The rest of the paper is structured as follows. In Section 2 we describe the models of network of groups and the problem statement. Methods proposed are illustrated in Section 3 and Section 4 has experimental analysis.

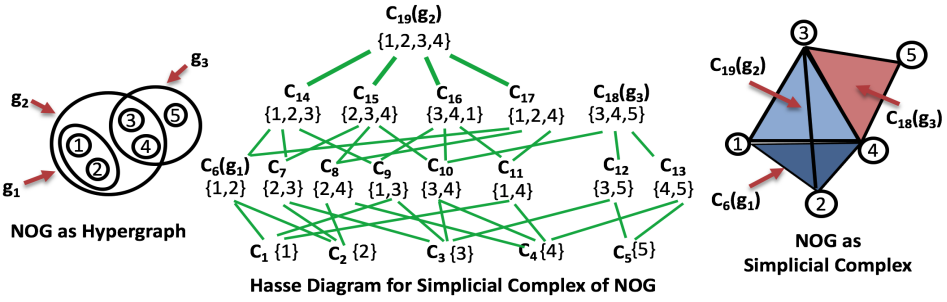

Fig. 1. Example illustrating a network of groups hypergraph (left) as a simplicial complex (right) and as a Hasse diagram (middle) corresponding to the simplicial complex, for a scenario where the actors {1,2,3,4,5} have collaberated in the past as groups: $g_1 = \{1, 2\}$, $g_2 = \{1, 2, 3, 4\}$ and $g_3 = \{3, 4, 5\}$.

2 Problem Statement and Preliminaries

2.1 Models for Network of Groups

We have a set of n *actors* $V = \{v_1, v_2, ..., v_n\}$. A subset of these *actors* can form a *group*. We have a collection of m such groups observed in the past, denoted

by $G = \{g_1, g_2, ..., g_m\}$ where $g_i \subseteq V$ represents the i^{th} *group*. The cardinality $c_i = |g_i|$ of a group is the number of actors in it. We let $R(g)$ denote the number of times group $g \in G$ has occurred. The network of groups can be modeled as a hypergraph [2] $H = (V, G)$ where the observed groups G are the hyperedges over the vertex set V of actors. We denote by $S_i = \{s_k^i, \forall k \in \{1, 2, ..., 2^{|g_i|}-2\}\}$ the set of all proper subsets of each group $g_i \in G$. If we consider the union of all subsets of the sets in G along with G itself, i.e., $C = \{G \cup (\bigcup_{i=1}^{m} S_i)\}$, then we have a (abstract) simplicial complex C and each element $c \in C$ is a simplex which represents a group or subgroup. If we also associate a weight $W(c) \in \mathbb{R}, \forall c \in C$, then we attain a *weighted simplicial complex* $\diamondsuit = (C, W)$. For convenience we also define the set containing the subgroups in C that were never observed in the past, i.e., $C_s = \{c|(c \in C) \wedge (c \notin G)\} = (C - G)$. Each $c \in C_s$ also has a set of groups $Q(c) \subseteq G$, of which it is a subgroup of, i.e., $Q(c) = \{x|(x \in G) \wedge (c \subset x)\}$. We define an occurrence function O which gives the occurrence count to all the groups in C as follows:

$$O(c) = \begin{cases} R(c) + \left(\sum_{x \in Q(c)} R(x) \right) & \text{when } c \in G \\ \sum_{x \in Q(c)} R(x) & \text{when } c \in C_s \end{cases} \tag{1}$$

In words, for an observed group we simply take the number of times it has occurred, $R(c)$, and also add the counts of the groups it has been a subset of. In the case of subgroups (those groups that haven't occurred in the past) we simply add the counts of the groups it has been a subset of. For a simplex (or (sub)group) $\alpha \in C$ we define its dimension as $dim(\alpha) = |\alpha| - 1$. If K_{max} is the maximum cardinality of any simplex in C then $(K_{max} - 1)$ is the maximum dimension of any simplex in C or simply the dimension of C.

The set of simplices of cardinality k within the simplicial complex C are defined by the set: $\pi^{\mathbf{k}} = \{\sigma|\sigma \in C \wedge |\sigma| = k\}, \forall k \in \{1, ..., K_{max}\}$. For the example in Figure 1, $C = \{C_1, ..., C_{19}\}$, $G = \{g_1, g_2, g_3\} = \{C_6, C_{19}, C_{18}\}$ and $C_s = (C - G)$.

We also define a *Hasse diagram*, T, for the simplicial complex C. The level in the diagram (Figure 1) determines the poset relation. We use the undirected graph derived from the Hasse diagram T over the vertex set $V(T) = C$ and with a set of undirected edges $E(T) = \{(x, y) \cup (y, x)|(x, y \in V(T)) \wedge (y \subset x) \wedge (|y| = |x| - 1)\}$. In the case of a weighted simplicial complex $\diamondsuit = (C, W)$, we associate with each vertex the weight of the corresponding simplex it represents, i.e., $W(v), \forall v \in V(T)$. Note, we can also associate a weight with the edges but in this study we assume all edges have a unit weight. We denote $\mathbf{A}$ to be the adjacency matrix of size $(|C| \times |C|)$ associated with the graph T.

2.2 Problem Statement

We are interested in prediction of groups formed by two processes: group recurrence and subgroup recurrence. In group recurrence, a group $g_i \in G$, called a *recurring group*, observed in the past can again occur in the future. Our first

problem is to predict a score for each of the groups in G. This score reflects the possibility of the given group occurring again in the future. In subgroup recurrence, a group $c_i \in C_s$ which has never been observed as a group in the past, might occur in the future. We refer to such groups as *recurring subgroups*. Our second problem is to predict a score for each of the groups in C_s, which reflects its possibility to be formed in future. We restrict ourselves to the prediction of only the recurring groups and subgroups and not groups composed of entirely new actors.

3 Methods

In this section, we first enumerate several ways of assigning prior weights. We then describe three different methods (along with several variants) to solve the problems described in the previous section. Each method models the tendency of a given group to be formed in the near future by assigning a score $\mathbf{S}(c)$ to each group in $c \in C$, returning a final vector of scores $\mathbf{S}$. The first method uses a simple group count-based approach and the next two methods consider the hierarchical structure of the higher order topology within the Hasse diagram.

3.1 Schemes for assigning initial weights

Several studies on small groups have shown that social actors tend to collaborate with actors with whom they have already developed strong working relationships [5] and that repeated ties within a group positively affect its performance [3]. There are a number of ways to assign a prior weight to represent the strength of the relationships between group members. Kapoor et al. [4] defined several weights for the problem of node centrality, of which we utilize two. The first, shown in (2), corresponds to a frequency-based definition and simply counts the number of times a group has performed some task together. The second, shown in (3), enforces that the average attachment of any two individuals (or the attention span of a member towards each other member) in a group decreases in proportion to the size of the group.

$$\mathbf{W}(c) = O(c), \forall c \in C \qquad (2) \qquad\qquad \mathbf{W}(c) = \frac{\log(O(c)) + 1}{|c|}, \forall c \in C \qquad (3)$$

The weights in (2) and (3) initialize all groups (observed) as well as subgroups (unobserved), i.e., all the simplices. We, therefore, also design slightly different variants where we only initialize the observed groups, which emphasizes the hypergraph model of the network:

$$\mathbf{W}(c) = \begin{cases} O(c) & \text{if } c \in G \\ 0 & \text{if } c \in C_s \end{cases} \qquad (4) \qquad \mathbf{W}(c) = \begin{cases} \dfrac{\log(O(c)) + 1}{|c|} & \text{if } c \in G \\ 0 & \text{if } c \in C_s \end{cases} \qquad (5)$$

In the following sections we will define several algorithms which will use these four initialization schemes. We will use the suffixes: **Simp-C**, **Simp-W**, **Hyp-C**, and **Hyp-W** to refer to the initializations in (2)-(5), respectively.

3.2 Count Based Scores (CBS)

We build the first set of scores using only the occurrence information available. For this we simply take the score vector $\mathbf{S}$ as the weight defined in (2) and (3), denoted the **CBS-C** score and **CBS-W** score, respectively. The **CBS-C** score, gives each group a value which is determined by the number of times the group members have worked together in past. Whereas, **CBS-W** assigns score based upon the cohesion among the group members.

3.3 Hasse Diagram based Models

CBS scores utilize counts of group recurrences, wherein each group was considered in isolation but do not consider the network of groups. This network encodes information about the observed groups, the unobserved groups, and the topological relations between them. Occurrences of a group affect the probability of other groups in the network to collaborate in the future. We develop two approaches applied to a Hasse diagram representation of a weighted simplicial complex to capture the local and global relational information.

Algorithm 1 GetHDSScores $(T, \mathbf{y}, K_{max}, \alpha)$

$\mathbf{f} \leftarrow \mathbf{y}$, $C \leftarrow V(T)$ {Get the simplicial complex corresponding to the Hasse diagram}
for $k = K_{max} - 1$ to 1 **do**
  for all $c \in \pi^{\mathbf{k}}$ **do**

$$\mathbf{f}(c) \leftarrow \mathbf{f}(c) + \alpha \left(\sum_{x \in (Q(c) \cap \pi^{\mathbf{k+1}})} y(x) \right)$$

  return $\mathbf{f}$

Hasse diagram spread-based scores (HDS Scores) This class of methods is based upon the intuition that observed groups in the Hasse diagram influence the subgroups below it in the hierarchy. Influence spread can happen in a variety of ways. There are several possible counter-intuitive group phenomena. We model these in a holistic fashion by spreading scores over the Hasse diagram. We propose that if we observe a node g_i in the Hasse diagram then it spreads its score (s_i) down the hierarchy. It can send the same, more, or less of its score to its children. In general, it can send αs_i ($\alpha \geq 0$) score to its children. These children update their scores and spread the score down the hierarchy recursively. This is shown in Algorithm 1. We initialize the algorithm using the vectors ($\mathbf{y} = \mathbf{W}$) in equations (2)-(5) to get four different scores, $\mathbf{S} = GetHDSScores(T, \mathbf{y}, K_{max}, \alpha)$, which we denote as **HDSSimp-C**, **HDSSimp-W**, **HDSHyp-C** and **HDSHyp-W**, respectively.

Hasse diagram diffusion-based scores: The spread-based scores are local in the sense that the final score of a node is only determined by its initial score and the scores of its parent(s). But, in general, the nodes representing groups in the network are connected by many pathways. Therefore, it is reasonable to assume that a potential group may be affected by occurrence of non-parent groups in the network. In order to take into account this structure of the entire Hasse diagram, we apply a modification of the graph label propagation algorithm HyperPrior [24].

Each vertex (group) is initialized with a label, which encodes prior information about the recurring tendency of that node. These labels (information) then diffuse (exchange information) via random-walks through the Hasse diagram network structure. After the random-walks stabilize, the final label for each vertex is the score indicating its recurrence possibility. The final label at a given vertex represents the chances that a random walk originating from other nodes ends at this vertex. Hence, this score is a combination of both the group's initial tendency to occur plus an adjustment based on the knowledge from other groups in the network, i.e., the random walk outcomes. This adjustment models a network guided similarity between the vertex and the other nodes. Vertices that are near in the network should end up receiving similar labels/scores.

More formally, let $\mathbf{y}$ be the vector of initial labels for the vertices in the Hasse diagram T with incidence matrix $\mathbf{A}$. Vector $\mathbf{y}$ is initialized by any of the weights in (2)-(5). As in a graph-based learning task, we learn the final label (score) vector $\mathbf{f}$ by taking into account the competing aims of similar labels for vertices connected by an edge in the Hasse diagram and of similar labels between the initial and final vectors. We capture these competing aims in the following cost minimization objective:

$$\min_{\mathbf{f}} \mathbf{f}^T \mathbf{L}\mathbf{f} + \beta \|\mathbf{f} - \mathbf{y}\|^2 \tag{6}$$

where, $\mathbf{L} = \mathbf{I} - \mathbf{D_v^{-1/2}} \mathbf{A} \mathbf{D_v^{-1/2}}$ is the normalized graph Laplacian [26] and $\mathbf{D_v}$ is a diagonal matrix consisting of the vertex degrees. The first term in (6) is a smoothing term which ensures that vertices (groups) sharing an edge (having common group members) have similar scores. This term therefore, enforces the Hasse diagram structure while learning the labels. The second term measures the difference between the given initial labels and the final vertex scores. It can be shown [26] that the solution to (6) is equivalent to the solution of the following linear system:

$$\mathbf{f}^* = (1 - \mu)(\mathbf{I} - \mu\theta)^{-1}\mathbf{y}, \tag{7}$$

where $\mu = 1/(1 + \beta)$, $\theta = \mathbf{D_v^{-1/2}} \mathbf{A} \mathbf{D_v^{-1/2}}$, and $\mathbf{f}^*$ is the vector of final labels of the group nodes. Note that, $\mathbf{f}^*(c)$ is the aggregate tendency $\mathbf{S}(c)$ of a group $c \in C$ to reoccur. Therefore, we have: $\mathbf{S} = \mathbf{f}^*$.

Similar to spread-based scores, we denote the scores here by **HDDSimp-C**, **HDDSimp-W**, **HDDHyp-C**, and **HDDHyp-W** when intialized using (2)-(5). Our aim is to predict a score for the recurring groups (i.e., $g \in G$) and recurring subgroups (i.e., $c \in C_s$). For each of the methods above, we get a final vector that contains the scores for all the groups. We partition the vector $\mathbf{S}$ into two vectors $\mathbf{S_{rg}}$ and $\mathbf{S_{rs}}$ of sizes $|G|$ and $|C_s|$, respectively, such that $\mathbf{S_{rg}}(c) = \mathbf{S}(c), \forall c \in G$ and $\mathbf{S_{rs}}(c) = \mathbf{S}(c), \forall c \in C_s$. In summary, we obtain three score vectors $\mathbf{S_{rs}}$, $\mathbf{S_{rg}}$ and $\mathbf{S}$ for each of the above methods.

4 Experimental Analysis

4.1 Dataset and Statistics

Datasets: The first dataset we apply our methods to is a massive multiplayer online role-playing game (MMORPG) dataset obtained from the Sony's EverQuest II (EQ II) game (www.everquest2.com). The game provides an online

environment where multiple players can log in and collaborate in groups to perform various quests and missions. The server logs from this game, provided by Sony, were used to extract group interactions. Here, we treat a set of players performing a task or mission as a group in the EQ II network. The EQ II data contains logs for 21 weeks of data for training and testing. We divide them into seven training/testing splits, each of which has a two-week long training period followed by a one-week testing period.

The second dataset is the DBLP dataset (obtained from www.aminer.org) containing computer science publications from 1930-2015. The set of co-authors on a paper form a group in the DBLP network. Note that in both EQII and DBLP networks, the groups can perform multiple game tasks or co-author multiple papers. We make eleven train-test splits as follows: $(1992-95/96-98)$, $(1993-95/96-98)$, $(1993-95/96-99)$, $(1991-97/98-10)$, $(1997-00/01-03)$, $(1998-00/01-03)$, $(1998-00/01-04)$, $(2002-05/06-08)$, $(2003-05/06-08)$, $(2003-05/06-09)$ and $(2001-07/08-10)$; following the format: *(train period start year−train period end year/ test period start year−test period end year)*. These splits were designed to observe the effect of varying training and testing period lengths as well as varying the entire train/test evaluation period. We have evaluated other variations of period lengths and other decades in the DBLP data, but in this paper we limit our discussion to the train/test periods we just described.

Table 1. Recurrence Statistics of the various Train/Test Periods

Dataset	Training Actors	Testing Actors	% Old Actors in Testing	% New Actors in Testing	Training Groups	Testing Groups	% Recurring Groups in Testing	% New Groups in Testing	% Groups with Old Actors	% Groups with New Actors
EQ II	3051	2215	81.67	18.33	1775	1219	67.92	32.08	88.93	11.07
Avg.			**84.06**	**15.94**			**74.01**	**25.99**	**90.51**	9.49
DBLP	677K	640K	40	60	549K	433K	12.06	87.94	84.53	15.47
Avg.			**34.73**	**65.27**			**11.65**	**88.35**	**81.17**	18.83

Statistics: Recall that we have two kinds of groups: (1) recurring groups that are observed in training and observed again in testing and (2) recurring subgroups that are observed in testing but are only observed as a subgroup of some group that occurred in training. We shall refer to the former set as RG, the latter set as RS, and the combined set as (RG+RS). Table 1 contains several statistics for (RG+RS). However, due to space constraints, we only show statistics for the last split from each dataset, as well as the average statistics across the splits. In Table 1, an actor in the testing phase is considered "**old**" if it was observed in the training period, otherwise it is considered "**new**". Note that for any group with new actors in the testing phase, we can only test whether the subgroup with old actors is a recurring group or subgroup from the training

Table 2. Different Dimension Face Recurrence Statistics

	% Train Groups	% Test Groups	% Train Groups	% Test Groups	% Train Groups	% Test Groups	% Train Groups	% Test Groups
	EQ II Splits				**DBLP Splits**			
Simplex Dimension	RG+RS (Exact)		RG+RS (New Vertices)		RG+RS (Exact)		RG+RS (New Vertices)	
≥ 1	15.57	**21.25**	15.70	**21.43**	4.63	**3.09**	6.28	**4.20**
≥ 2	8.97	**12.72**	9.02	**12.80**	2.78	**1.79**	3.41	**2.19**
	RS (Exact)		RS (New Vertices)		RS (Exact)		RS (New Vertices)	
≥ 1	0.70	**0.71**	0.76	**0.77**	1.70	**0.89**	3.08	**1.61**
≥ 2	0.20	**0.23**	0.25	**0.29**	0.76	**0.38**	1.24	**0.63**
	RG (Exact)		RG (New Vertices)		RG (Exact)		RG (New Vertices)	
≥ 1	57.18	**20.55**	57.50	**20.66**	14.89	**2.21**	17.53	**2.60**
≥ 2	45.68	**12.49**	45.77	**12.51**	10.02	**1.41**	11.17	**1.57**

period. These statistics are based on the distinct groups from the testing and training periods, so as to avoid any bias from the multiplicity of certain group interactions. We observe that on an average around 90% of the EQ II network groups and around 81% of the DBLP network groups formed in the test period contain at least one old actor. Only within these groups can we possibly search for recurring groups or subgroups. Note, 74% of the EQ II groups and around 12% of DBLP groups in testing period are exact recurrences and included in the set RG. This demonstrates that the recurring group process is more common in the EQ II network, whereas the recurring subgroup process is the more common feature in the DBLP network.

In Table 2, we record the statistics of the groups in training that recur in testing and of the groups in testing that are recurring groups or subgroups. We only consider groups of size ≤ 6 (i.e., faces of dimension ≤ 5) and also omit vertex recurrences since those are reported in Table 1.

For dimensions ≥ 1, the set RG+RS accounts for 20% of the testing groups in the EQ II network and $3-4\%$ in the DBLP network. For dimensions ≥ 2, the set RG+RS accounts for approximately 12% of the testing groups in the EQ II network and only 2% in the DBLP network. These subtle observations indicate that GR and SR processes are responsible for a significant portion of future formed groups. Therefore, modeling these processes is an important step towards higher order link prediction.

4.2 Evaluation Methodology and Experimental Setup

We evaluate the performance of these methods as classifiers using the area under the curve (AUC) statistic of the receiver operating characteristics (ROC) [25]. Using the three score vectors as the model output we calculated AUC scores for two sets of prediction test scenarios. The first set includes the exact occurrences found in the testing period (referred to as **"(Exact)"**) and the other set includes occurrences found with new vertices in the testing period (referred to as **"(New Vertices)"**). The following six scenarios are considered for each set:

1. **RG+RS(v)**: Predicting both recurring groups and subgroups that are dyadic edges or other higher order faces. Note that for any group with new actors in the testing phase, we can only test whether the subgroup with old actors is a recurring group or subgroup from the training period.
2. **RG+RS(v+e)**: Predicting both recurring groups and subgroups that are only triangles or other higher order faces. We only consider groups of size ≤ 6 and also omit vertex recurrences since those are reported in Table 1.
3. **RS(v)**: Predicting only recurring subgroups that are edges or other higher order faces.
4. **RS(v+e)**: Predicting only recurring subgroups that are triangles or other higher order faces.
5. **RG(v)**: Predicting only recurring groups that are edges or other higher order faces.
6. **RG(v+e)**: Predicting only recurring groups that are triangles or other higher order faces.

The optimal parameters were chosen for each split separately via grid search on the following parameter space: $\alpha = \{0.01, 0.1, 0.3,\ 0.5, 0.7, 0.9, 1, 2, 5, 10, 20\}$ and $\mu = \{10^{-7}, 10^{-6}, 10^{-5},\ 10^{-4}, 10^{-3},\ 0.01, 0.1, 0.3, 0.5, 0.7, 0.9, 0.99\}$. All the Hasse diagrams considered in the above methods have un-weighted edges.

Table 3. AUC Scores for **EQ II** and **DBLP**

	EQ II											
	Exact						**New Vertices**					
Method	RG+RS (v)	RG+RS (v+e)	RS (v)	RS (v+e)	RG (v)	RG (v+e)	RG+RS (v)	RG+RS (v+e)	RS (v)	RS (v+e)	RG (v)	RG (v+e)
HDDHyp-W	**0.96**	**0.98**	**0.67**	**0.87**	0.79	0.83	**0.96**	0.97	**0.68**	**0.86**	0.79	0.83
HDDHyp-C	**0.96**	**0.98**	0.63	0.78	**0.83**	**0.87**	**0.96**	**0.98**	0.64	0.78	0.82	**0.86**
HDDSimp-W	0.83	0.81	0.63	0.67	0.78	0.81	0.83	0.81	0.62	0.64	0.78	0.81
HDDSimp-C	0.78	0.75	0.52	0.6	0.83	0.86	0.78	0.75	0.52	0.59	**0.83**	0.85
CBS-W	0.77	0.68	0.6	0.54	0.78	0.81	0.76	0.68	0.59	0.5	0.78	0.81
CBS-C	0.76	0.72	0.5	0.49	0.82	0.85	0.76	0.72	0.5	0.47	0.82	0.85
HDSHyp-W	**0.96**	0.97	0.65	0.71	0.79	0.82	**0.96**	0.97	0.65	0.7	0.79	0.82
HDSSimp-W	0.7	0.59	0.58	0.52	0.76	0.8	0.7	0.59	0.58	0.48	0.76	0.8
HDSHyp-C	0.95	0.97	0.58	0.63	**0.83**	0.86	0.95	0.97	0.59	0.63	0.82	**0.86**
HDSSimp-C	0.68	0.61	0.49	0.48	0.82	0.85	0.67	0.6	0.48	0.44	0.82	0.85
	DBLP											
HDDHyp-W	**0.9**	**0.89**	**0.8**	**0.78**	0.69	0.68	**0.82**	**0.85**	0.73	**0.72**	0.7	0.68
HDDHyp-C	0.89	**0.89**	0.79	0.78	0.69	0.68	**0.82**	**0.85**	0.73	**0.72**	0.69	0.68
HDDSimp-W	0.77	0.79	0.73	0.73	0.7	0.69	0.77	0.77	**0.74**	0.71	0.71	0.69
HDDSimp-C	0.75	0.76	0.71	0.72	**0.71**	**0.7**	0.74	0.74	0.73	0.7	**0.72**	**0.7**
CBS-W	0.67	0.64	0.65	0.61	0.69	0.66	0.69	0.64	0.7	0.63	0.7	0.66
CBS-C	0.65	0.63	0.59	0.58	0.64	0.62	0.65	0.62	0.63	0.59	0.65	0.62
HDSHyp-W	0.89	0.88	0.75	0.73	0.69	0.66	**0.82**	0.84	0.73	0.7	0.7	0.66
HDSSimp-W	0.59	0.53	0.6	0.54	0.69	0.66	0.63	0.55	0.67	0.57	0.7	0.66
HDSHyp-C	0.88	0.87	0.72	0.71	0.64	0.62	0.8	0.83	0.68	0.67	0.65	0.62
HDSSimp-C	0.49	0.43	0.5	0.47	0.64	0.61	0.54	0.46	0.57	0.51	0.65	0.62

4.3 Results and Discussion

We compare the twelve different AUC scores, described in the prior section, for the ten methods developed in this paper. Results are reported in Table 3 for the EQ II and DBLP data. We have three different kinds of scores: CBS (Section 3.2), HDS (Section 3.3) and HDD (Section 3.3). Both the **CBS-W** and **CBS-C** scores are only count based and don't take into account any topological relationship between groups. On the other hand, the HDS and HDD methods

take into account topology by exchanging information locally and globally, respectively. One of our main hypotheses is that topological structure affects the group recurrence behavior. We are also unaware of any methods for small group recurrence and therefore chose **CBS-W** and **CBS-C** scores as our baseline. Note, as described in Section 3.1, all the three genre of methods can be either count based (referred using suffix **-C**), or cohesion metric based (denoted by suffix **-W**). The count based variants do not take into account the cardinality of the (sub)groups whereas the cohesion metrics are cardinality based.

Effect of Topology: We observe from Table 3 (the best scores are highlighted in bold) that the Hasse diagram-based methods consistently outperform the count-based methods. This supports our hypothesis that Hasse diagram-based methods, which take into account topology, indeed, are more informative about the group recurrence process.

We also compare the methods against four criteria: (a) How do the prediction methods fare for recurring subgroups as compared to recurring groups?; (b) How well do the methods predict at dimensions of dyadic edges and above, i.e., the -(v) cases, compared with how well they predict at dimensions of triadic groups and above, i.e., the -(v+e) cases?; (c) How well do the methods predict the "Exact" occurrences versus the "New Vertices" occurrences?; and (d) How do the count-based **"-C"** methods compare with the cohesion-based **"-W"** methods?

We observe that in order to predict recurring subgroups, **HDDHyp-W** outperforms all other methods whether the subgroup was an "exact" occurrence or a "new vertices" occurrence in testing. This suggests that exchange of information from the groups observed in the past to the groups not observed in the past via the Hasse diagram topology and the global-based label diffusion process is more crucial for influencing the appearance of subgroups not observed in the past. In fact, the poor accuracy of the **HDDSimp** methods indicates that weights placed on (possibly unobserved) subgroups of observed groups used as prior information cause bias and hurt the predictive power of the model. Given that **HDDHyp-W** is initialized using the cohesion weights in (4), the normalization of counts only on the prior observed group occurrences in the diagram is important for recurring subgroup prediction. Moreover, the performance of predicting triangles or higher order groups (RS(v+e)) is higher for the EQ II data and comparable for the DBLP data to that of predicting dyadic edges or higher (RS(v)) across all HDD methods, implying the important role played by the Hasse diagram structure for higher order group prediction.

On the other hand for recurring group prediction the count-based methods **HDDHyp-C** and **HDDSimp-C** performed best, suggesting that the likelihood of recurrence of already-observed groups is determined more by the simple counts of past concurrences. The count-based HDS methods also give results comparable with that of the HDD methods. This implies that even the local spread of count information is sufficient for recurring group predictions. These **-Simp**-based methods using (1), which take into account the subgroup counts of the groups that occurred in training, provide good results, suggesting that the unobserved subgroups have an important influence on the potential of groups to re-occur.

Finally, we note that across both the datasets and across all the twelve experiments, the **HDD** methods generally perform better than or as good as **HDS** methods. Further results and details shall be made available in a future technical report.

5 Conclusions

We consider the problem of predicting small group evolution and focus on the sub-problem on group and subgroup recurrence. We highlight two important group recurrence processes and capture them using weighted simplicial complexes. We use a Hasse diagram corresponding to the simplicial complex as a graph whose nodes correspond to subgroups in the complex. We then build semi-supervised models on top of this graph for group recurrence prediction. We have shown that frequent patterns like small groups can be considered as topological entities, with relationships between them guided by higher order topological properties.

References

1. Aggarwal, C.C., Han, J.: Frequent Pattern Mining. Springer (2014)
2. Berge, C., Minieka, E.: Graphs and hypergraphs. North-Holland publishing company Amsterdam (1973)
3. Contractor, N.: Some assembly required: leveraging web science to understand and enable team assembly. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 371(1987) (2013)
4. Kapoor, K., Sharma, D., Srivastava, J.: Weighted node degree centrality for hypergraphs. In: IEEE Network Science Workshop. pp. 152–155. IEEE (2013)
5. Lungeanu, A., Huang, Y., Contractor, N.S.: Understanding the assembly of interdisciplinary teams and its impact on performance. Journal of Informetrics 8(1), 59–70 (2014)
6. Monge, P.R., Contractor, N.S.: Theories of communication networks. Oxford University Press (2003)
7. Moore, T.J., Drost, R.J., Basu, P., Ramanathan, R., Swami, A.: Analyzing collaboration networks using simplicial complexes: A case study. In: Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on. pp. 238–243. IEEE (2012)
8. Patil, A., Liu, J., Shen, J., Brdiczka, O., Gao, J., Hanley, J.: Modeling attrition in organizations from email communication. In: IEEE Int'l Conf. on Social Computing (SocialCom). pp. 331–338 (2013)
9. Poole, S., Poole, M.S., Hollingshead, A.B.: Theories of small groups: Interdisciplinary perspectives. Sage Publications (2004)
10. Ramanathan, R., Bar-Noy, A., Basu, P., Johnson, M., Ren, W., Swami, A., Zhao, Q.: Beyond graphs: Capturing groups in networks. In: IEEE INFOCOM Workshops. pp. 870–875 (2011)
11. Roy, A., Singhal, A., Srivastava, J.: Formation and reciprocation of dyadic trust. ACM Transactions on Internet Technology (TOIT) 17(2), 15 (2017)
12. Sharma, A., Kuang, R., Srivastava, J., Feng, X., Singhal, K.: Predicting small group accretion in social networks: A topology based incremental approach. In: IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). pp. 408–415 (2015)

13. Sharma, A., Srivastava, J.: Group analysis using machine learning techniques. In: Group Processes, pp. 145–180. Springer (2017)
14. Sharma, A., Srivastava, J., Chandra, A.: Predicting multi-actor collaborations using hypergraphs. arXiv preprint arXiv:1401.6404 (2014)
15. Simmons, M., Singhal, A., Lu, Z.: Text mining for precision medicine: bringing structure to ehrs and biomedical literature to understand genes and health. In: Translational Biomedical Informatics, pp. 139–166. Springer (2016)
16. Singhal, A., Kasturi, R., Sharma, A., Srivastava, J.: Leveraging web resources for keyword assignment to short text documents. arXiv preprint arXiv:1706.05985 (2017)
17. Singhal, A., Kasturi, R., Srivastava, J.: Automating document annotation using open source knowledge. In: Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on. vol. 1, pp. 199–204. IEEE (2013)
18. Singhal, A., Roy, A., Srivastava, J.: Understanding co-evolution in large multi-relational social networks. In: Information Reuse and Integration (IRI), 2014 IEEE 15th International Conference on. pp. 733–740. IEEE (2014)
19. Singhal, A., Simmons, M., Lu, Z.: Text mining for precision medicine: automating disease-mutation relationship extraction from biomedical literature. Journal of the American Medical Informatics Association 23(4), 766–772 (2016)
20. Singhal, A., Simmons, M., Lu, Z.: Text mining genotype-phenotype relationships from biomedical literature for database curation and precision medicine. PLoS computational biology 12(11), e1005017 (2016)
21. Singhal, A., Srivastava, J.: Leveraging the web for automating tag expansion for low-content items. In: Information Reuse and Integration (IRI), 2014 IEEE 15th International Conference on. pp. 545–552. IEEE (2014)
22. Singhal, A., Subbian, K., Srivastava, J., Kolda, T.G., Pinar, A.: Dynamics of trust reciprocation in multi-relational networks. In: Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. pp. 661–665. ACM (2013)
23. Skiena, S.: Hasse diagrams. Implementing Discrete Mathematics: Combinatorics and Graph Theory With Mathematica p. 163 (1990)
24. Tian, Z., Hwang, T., Kuang, R.: A hypergraph-based learning algorithm for classifying gene expression and arraycgh data with prior knowledge. Bioinformatics 25(21), 2831–2838 (2009)
25. Van Trees, H.L.: Detection, Estimation and Modulation Theory. Wiley, New York (1968)
26. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B.: Learning with local and global consistency. Advances in neural information processing systems (NIPS) 16(16), 321–328 (2004)
27. Zuckerman, E.W.: Do firms and markets look different? repeat collaboration in the feature film industry, 1935–1995. Unpublished (2004)