*

Predicting Multi-actor collaborations using Hypergraphs

Ankit Sharma
Dept. of Computer Science
University of Minnesota
200 Union Street
Minneapolis, MN 55455
ankit@cs.umn.edu

Jaideep Srivastava
Dept. of Computer Science
University of Minnesota
200 Union Street
Minneapolis, MN 55455
srivasta@cs.umn.edu

Abhishek Chandra
Dept. of Computer Science
University of Minnesota
200 Union Street
Minneapolis, MN 55455
chandra@cs.umn.edu

ABSTRACT

Social networks are now ubiquitous and most of them contain interactions involving multiple actors (groups) like author collaborations, teams or emails in an organizations, etc. *Hypergraphs* are natural structures to effectively capture multi-actor interactions which conventional dyadic graphs fail to capture. In this work the problem of predicting collaborations is addressed while modeling the collaboration network as a hypergraph network. The problem of predicting future multi-actor collaboration is mapped to hyperedge prediction problem. Given that the higher order edge prediction is an inherently hard problem, in this work we restrict to the task of predicting edges (collaborations) that have already been observed in past. In this work, we propose a novel use of hyperincidence temporal tensors to capture time varying hypergraphs and provides a tensor decomposition based prediction algorithm. We quantitatively compare the performance of the hypergraphs based approach with the conventional dyadic graph based approach. Our hypothesis that hypergraphs preserve the information that simple graphs destroy is corroborated by experiments using author collaboration network from the DBLP dataset. Our results demonstrate the strength of hypergraph based approach to predict higher order collaborations (size>4) which is very difficult using dyadic graph based approach. Moreover, while predicting collaborations of size>2 hypergraphs in most cases provide better results with an average increase of approx. 45% in F-Score for different sizes $\in \{3, 4, 5, 6, 7\}$ (Figure 6).

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

*A full version of this paper is available as *Author's Guide to Preparing ACM SIG Proceedings Using LaTeX2ε and BibTeX* at `www.acm.org/eaddress.htm`

Keywords

Collaboration networks, social networks, link prediction, tensors, hypergraphs, team formation

1. INTRODUCTION

The problem of understanding group dynamics is central to the field of Social Sciences. Moreover, the increasing use of internet has led to an exponential increase in amount of online interaction data. As examples, social networking sites like Facebook or Twitter, group communication tools like Skype, Google Hangout, Google Docs, Massive Online multi-player games such as World of Warcraft, etc., are generating social networking data at a massive scale. These social datasets provides minute by minute account of interaction along with the structure and the content of these relationships [23].

In the domain of Social Science, a lot of studies have been conducted to understand how groups form, their static as well as dynamic attributes and structures, and how they evolve over time [4]. The research collaborations in scientific community are an excellent example of social networks in which individuals of various expertise collaborate to solve a research problem. Collaboration networks from scientific research community have been extensively used for studying team dynamics [14][17][2]. Group dynamics has real life applications as well for example in building emergency response teams for natural disasters management, automation of team selection for military operations, etc.

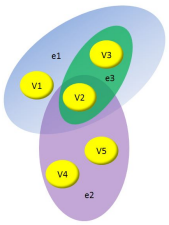

Figure 1: Hypergraph

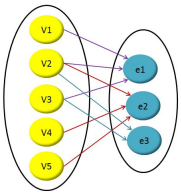

Figure 2: Bipartite of hypergraph (Fig.1)

The above examples reveal that there can be multiple overlapping collaborations which form a network of collaborations. Modeling such collaborations in dynamic settings where relationship between actors is evolving over time is a challenging task. Unfortunately, most of the prior research in social network analysis deals with dyadic interactions or small well-defined groups [19] rather than at the

group level. There are some studies that have dealt with group interactions by collapsing the group into dyadic links [17] and therefore, fail to keep the group level information intact. Ghoshal et. al. [9] have used tripartite regular hypergraph whcih captures folksonomy data but is too restrictive to capture variable size social collaborations. GuimerÃă et al. [11] attempts attempts to model group using node and group attributes which can explain the network structure but fails to deal with individual group evolution. Bipartite graphs (figure 2) as network models have also been used to capture groups [8] where one set of nodes represent event/collectives/groups and the other second set of nodes represent the actors. But in this model the relations between the actors have to be derived as the group relation is not represented in an intact manner [13].

Hypergraphs are generalization of graphs, which can have more than two node in an edge (rather than simple graphs where only 2 nodes are part of an edge). Therefore, hypergraphs can easily capture the coexistence of more than two entities in a single relation. Figure 1 shows a hypergraph with five nodes and three edges.

1.1 Related Work

Hypergraphs can easily capture the higher-order relationships while incorporating both group and node level attributes. Moreover, research has shown that several social, biological, ecological and technological systems can be better modeled using hypergraphs than using dyadic proxies [6]. There is an abundant literature of hypergraph theory in past [3] and many work in the spectral theory of hypergraphs recently [18][24]. The past decade has also seen an increasing interest for hypergraphs in machine learning community [25][22]. Hypergraphs have been used to model complex networks in different fields including biology [15], databases [7] and data mining [12]. In the domain of social sciences, Kapoor et a.l [13] have proposed with centrality metrics for weighted hypergraphs. Tramasco et al. [21] propose hypergraphs based metrics to evaluate various hypothesis, both semantic and structural, regarding team formation.

Although a lot of work done has been regarding mathematical formulation of hypergraphs, very few works (as stated above) capture the full potential of hypergraph models for real world applications. In this paper, we address the problem of higher order collaboration predictions by modeling it as a hyperedge prediction problem. The collaboration network is modeled as a hypergraph network. Given a previous history of the collaborations we predict collaborations using an supervised approach for hyperedge prediction. In order to capture the time varying hypergraphs and graphs we propose with a novel application of tensor in the form of incidence or hyper-incidence tensors. Our results show that graphs give significantly lower F-Score for higher order groups of size= $\{4, 5, 6, 7\}$ in comparison to hypergraph for most of the cases. In predicting collaborations (hyperedges) higher than size two i.e. more than two entities, hypergraphs in most cases provide better results with an average increase of approx. 45% in F-Score for different sizes $\in \{3, 4, 5, 6, 7\}$ (Figure 6). The main contributions of the paper are summarized as follows:

Our results demonstrate the strength of hypergraph based approach to predict higher order collaborations (size>4) which is very difficult using dyadic graph based approach. Moreover, while predicting collaborations of size>2 hypergraphs in most cases provide better results with a (25-150)% increase in F-Score for different sizes $\in \{4, 5, 6, 7\}$ and various training-test splits.

- We show a quantitative comparison between graphs and hypergraphs from an applications perspective. This to the best of our knowledge is a pioneer work.
- We propose a novel application of tensors to capture hypergraphs and use the decomposed tensor's factors to come up with a prediction model for higher order groups.
- We have successfully addressed the problem of predicting collaborations of higher order which has been rearely dealt with in the past research as it is a problem of considerable complexity.

The rest of the paper is as follows: Section 2 nails down the various hyperedge prediction problems, Section 3 proposes our hypothesis to be evaluated, Section 4 talks about the tensors based algorithm to capture this hypothesis, Section 5 talks about the experiments conducted and results are discussed, which is followed by conclusion and future work.

2. HYPEREDGE PREDICTION PROBLEMS AND PRELIMINARIES

In this section we describe how higher order collaboration prediction can be mapped to hyperedge prediction problem.

2.1 Problem Statement

In this paper we have used research collaborations where a set of authors (*actors*) collaborate for research. Each of these collaboration results in a *publication* and each of these publications represents an instance of this collaboration. This means that the same *collaboration* might result in multiple publications. Each of these *collaboration* is modeled as a hyperedge in a hypergraph whose each vertex represent an *actor*. The problem of *collaboration* prediction can then be treated as a problem of predicting a hyperedge. This further splits in two sub-problems:

- Problem of predicting the hyperedges already observed in past i.e. *old edge* prediction.
- Problem of predicting the hyperedges that have never been observed in past i.e. *new edge* prediction.

To the best of our knowledge these problems have not yet been addressed explicitly. In this paper we are restricting ourselves to the former problem of *old edge* prediction.

2.2 Problem Definition

Let $V = \{v_1, v_2, ..., v_{N_a}\}$ be a set of vertices (*actors*). We represent the hypergraph of *collaboration*s using $HG(V, H)$ where H is the incidence matrix of hypergraph which we term as *hyper-incidence matrix*. This matrix H represent the set of hyperedges (*collaboration*s) $\{h_1, h_2, ..., h_{N_h}\}$ where each hyperedge $h_k = \{v_1^{h_k}, ..., v_{|h_k|}^{h_k}\} \subseteq V$. We divide time into small snapshots (of size w as shown in Figure 4) with t as its index. N_c^t is the number of *publications* occurred in snapshot t and there are N_t number of snapshots in past. We denote the i^{th} *publication* in the t^{th} snapshot by $c_i^{(t)}$

$\forall i = \{1, 2, ..., N_c^{(t)}\}$. Each of this *publication* $c_i^{(t)}$ represents some *collaboration* (hyperedge) h_k. A mapping function ϕ is defined which returns the *collaboration* (hyperedge) represented by a given *publication* such that $\phi(c_i^{(t)}) = h_k \ \forall k = \{1, ..., N_h\}$ where N_h are the number of distinct *collaborations* (hyperedges) in the past. Size of H is therefore, $N_h \times N_a$ and we call s_k as the cardinality (No. of vertices inside h_k) of the hyperedge h_k i.e. $s_k = |h_k|$.

The problem of *old link* prediction is now defined as follows: Given a past history of collaborations $C_{hist} = \{c^{(t)}\}_{t=1}^{N_t}$ our goal for the problem of *old link* prediction is to predict the likelihood of future occurrence of each of the hyperedges $h_k \ \forall k = \{1, ..., N_h\}$ (i.e. *collaborations* already observed in past).

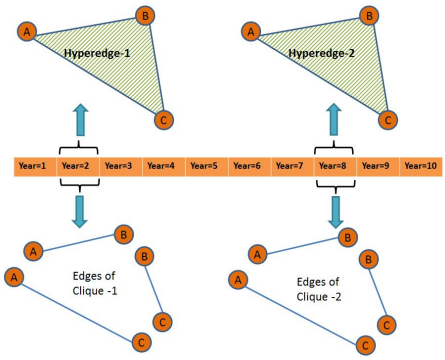

Figure 3: Toy Example showing of two *publications* published by *collaboration* (A, B, C) in year=2 and year=8 with their hyperedge (top) and clique of dyadic edges (bottom) representation.

3. HYPOTHESIS

In this section we state the hypothesis which is evaluated in this work. We claim that modeling social collaborations or interactions as hypergraph is likely to conserve a lot of information that is destroyed when modeled as dyadic graphs. The claim is supported by the following examples:

- *Independent dyadic interactions fail to predict higher order interactions :* For example, if A and B talk to each other often and similarly does, the pair (B-C) and (C-A). But this is unable to capture the same information nor can it give a sufficient prediction that (A-B-C) in a group will be interacting together. Whereas if we have seen (A-B-C) together several times this information is completely different than what we can attain from just observing the individual interaction independently. Thus, there is a blatant need for capturing higher order interaction is a form other than dyadic interactions.

- *Higher order interactions are captured in a much better manner using hypergraphs than a corresponding dyadic clique based representation:* For example as show in Figure 3, a *collaboration* of authors A,B and C produced couple of *publications* in a time window of ten years. Our aim is to predict future likelihood (P(A-B-C)) of this *collaboration* A-B-C reoccurring. If we use hyperedge representation then P(A-B-C) = 2/10. Whereas, on splitting the hyperedges as cliques of dyadic links, P(A-B-C) = P(A-B)xP(B-C)xP(C-A)= (2/10)x(2/10)x(2/10) = (8/1000) which is clearly less than the probability using the hyperedge. Hypergraph simply keeps the joint probability information intact.

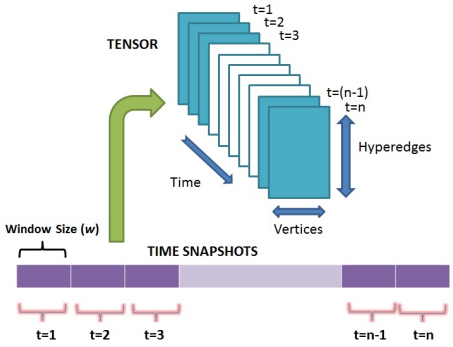

Figure 4: A tensor representation of the temporal information (snapshot size=w). Each snapshot data is fed in the corresponding hyper-incidence matrix.

4. PROPOSED APPROACH

This section describes the approach used to capture the above intuition and build a platform to conduct comparative analysis between graphs and hypergraphs. This section is divided into two sections. In the first section the hypergraph modeling using tensors is explained and the next section described the supervised hyperedge prediction.

4.1 Collaboration Modeling

4.1.1 Tensor and Incidence matrix representations

A tensor is a multidimensional, or N-way, array [18] and has proven to capture multi-dimensional data effectively [1]. For example Tensors allow to handle time as a separate dimension. This provides more flexibility to creatively manipulate the temporal dimension. Moreover, the temporal patterns can be captured using tensors to predict future patterns rather than just immediate future. Recently tensors have already proved effective in predicting temporal link prediction by Dunlavy et al [5]. This has encouraged us to capture hypergraphs and graphs using 3-way tensors where the first two dimensions capture the hypergraph/graph incidence matrix and the third dimension captures the temporal information. Keeping the same incidence matrix representation for both graph and hypergraph allows to have a parity comparison between the two models. We denote the tensor for graph and hypergraph using $\mathcal{Z}_g$ and $\mathcal{Z}_h$ which represent array of the snapshots of incidence or the hyper-incidence matrix respectively (Figure 4). Snapshot t refers to a time period $T = (w * (t-1), w * t)$.

Similar to hypergraph we represent graph as $G(V, E)$ where the graph incidence matrix is E represent the set of edges $\{e_1, e_2, ..., e_{N_g}\}$. Each edge contains a pair of vertices i.e. $e_k = \{v_i^{e_k}, v_j^{e_k}\} \subseteq V$ (Section 4.1.2 describes the method to obtain these edges). For the snapshot t we represent incidence matrix for graph as $E^{(t)}$ and use $H^{(t)}$ for the hyper-incidence matrix. Here, $E^{(t)}$ has the dimension $(N_g \times N_a)$ where N_g is the number of distinct dyadic edges between any two actors that have been observed uptil current time.

Similarly, the dimension of $H^{(t)}$ is $(N_h \times N_a)$ where N_h is the number of distinct multi-actor *collaborations* (hyperedges) between the actors that are observed till now. Note that in $E^{(t)}$ and $H^{(t)}$ only information of publications in snapshot t is stored but they have same dimension for all values of t.

: PREDICT-COLLAB(C_{hist}, $isHypergraph$)

[1]

$\mathcal{Z}_h$ tensor (size $N_h \times N_a \times N_t$) initialized to all zeros.

$\mathcal{Z}_g$ tensor (size $N_g \times N_a \times N_t$) initialized to all zeros.

$isHypergraph$ $c^{(t)} \in C_{hist}$ $c_i^{(t)} \in c^{(t)}$Find k s.t. $\phi(c_i) == h_k$ j s.t. $v_j \in \{v_1^{h_k}, ..., v_{|h_k|}^{h_k}\} = h_k$ $\mathcal{Z}_h(k,j,t) = \mathcal{Z}_h(k,j,t) + \frac{1}{s_k}$

$\mathbf{S}_h$ = BUILD-SIMILARITY-MATRIX ($\mathcal{Z}_h$, N_h, N_a)

return HYPERGRAPH-PROB-VECTOR ($\mathbf{S}_h$, N_h, N_a)

$c^{(t)} \in C_{hist}$ $c_i^{(t)} \in c^{(t)}$Find k s.t. $\phi(c_i) == h_k$ s_k is the cardinality of hyperedge h_k. Each of the $\binom{s_k}{2}$ dyadic links, d_p of the hyperedge h_k as a clique. Find k' s.t. dyadic edge d_p represents the same subset as c_i j s.t. $v_j \in \{v_1^{d_p}, v_2^{d_p}\} = d_p$ $\mathcal{Z}_g(k', j, t) = \mathcal{Z}_g(k', j, t) + \frac{1}{s_k}$

$\mathbf{S}_g$ = BUILD-SIMILARITY-MATRIX ($\mathcal{Z}_g$, N_g, N_a)

return GRAPH-PROB-VECTOR ($\mathbf{S}_g$, N_g, N_a)

: BUILD-SIMILARITY-MATRIX ($\mathcal{Z}$, N_a, N_b)

[1]

$\mathbf{S}$ similarity matrix of size $N_a \times N_b$ initialized with all zeros. K is the number of components. $[\lambda; \mathbf{A}, \mathbf{B}, \mathbf{C}]$ = CP-ALS($\mathcal{Z}$)

$k \in \{1, 2, ..., K\}$ $\mathbf{S} = \mathbf{S} + \lambda_k \gamma_k \mathbf{a_k} \mathbf{b_k}^\top$

return S

: HYPERGRAPH-PROB-VECTOR ($\mathbf{S}_h$, N_h, N_a)

[1]

$\mathbf{p_h}$ is the probability vector for hyperedge likelihood of length N_h initialized to all one.

$i \in \{1, 2, ..., N_h\}$ p s.t. $v_p \in h_i$ $\mathbf{p_h}(i) = \mathbf{p_h}(i) * \mathbf{S}_h(i, p)$

return $\mathbf{p_h}$

Therefore, $\mathcal{Z}_g(:,:,t) = E^{(t)}$ and $\mathcal{Z}_h(:,:,t) = H^{(t)}$ both representing array of snapshots of respective incidence matrices. Dimension of $\mathcal{Z}_g$ finally becomes $N_g \times N_a \times N_t$ and $\mathcal{Z}_h$ becomes $N_h \times N_a \times N_t$ dimensional.

4.1.2 Loading Tensors

Next step is to extract effective modeling information from historical publication data C_{hist} and feed it into both the graph and hypergraph tensors. The following couple of subsections describe this process for graphs and hypergraphs separately. We are using the following terms interchangeably: hyperedge and collaboration, occurrence of hyperedge and publication; and vertex and actor.

Hypergraph Case (Line (3-11) of Algorithm 1): All hyper-incidence matrices $H^{(t)}$ $\forall t$ have the same dimension and thus, the same number, N_h, of unique hyperedges. Each one of these hyperedges h_k $\forall k \in \{1, 2, ..., N_h\}$ represent a unique collaboration between a subset of actors (vertices) i.e. $h_k \subseteq V$. For each of the publication $c_i^{(t)} \in c^t$ for $i = \{1, 2, ..., N_c^{(t)}\}$ find the $k \in \{1, 2, ..., N_h\}$ such that $c_i^{(t)}$ represents the same subset of vertices as h_k i.e. $\phi(c_i^{(t)}) == h_k$. For this index k of the hyperedge, the tensor is filled as $\mathcal{Z}_h(k, j, t) = \frac{m_k}{s_k}$ where j is the index of each vertex which is the part of the hyperedge h_k, s_k is the cardinality of the hyperedge h_k and m_k is the multiplicity of the hyperedge h_k. Multiplicity is calculated as the log (No. of times h_k occurred in t), in other words how many times a particular *collaboration* published some work in snapshot t. This process captures the weight of the hyperedge h_k in the hypergraph tensor. The weight of the hyperedge is modeled as $(\frac{m_k}{s_k})$, as this definition of hyperedge weights is shown to give the best results by Kapoor et al.[13]. This whole process is repeated for all the time snapshots.

Graph Case (Line (14-25) of Algorithm 1): In case of graph also the graph-incidence matrices $G^{(t)}$ $\forall t$ have the same dimension and same number, N_g, of unique edges. Each of these edges g_k represent a unique set (dyadic collaboration) between two vertices (actors), $g_k = \{v_i^{g_k}, v_j^{g_k}\} \subseteq V$. For each publication $c_i^{(t)} \in c^t$ for $i = \{1, 2, ..., N_c^{(t)}\}$ find the $k \in \{1, ..., N_h\}$ such that $\phi(c_i^{(t)}) == h_k$. This hyperedge h_k is broken in to $\binom{s_c}{2}$ dyadic edges and let us denote each of the dyadic link by d_p. For each of the d_p find the index $k^{'} \in \{1, 2, ..., N_g\}$ for which the d_p represents the same edge as g_k. For this index $k^{'}$ the tensor is filled as $\mathcal{Z}_g(k^{'}, j, t) = \frac{m_k}{s_k}$ where j is the index of each vertex which is the part of the edge d_p, s_k is the cardinality of the hyperedge h_k and m_k is the multiplicity of the hyperedge h_k. Thus we model the dyadic link of the clique to get the weight of the original hyperedge [13]. Again, this whole process is repeated for all the time snapshots.

4.2 Decomposing the tensors (Algorithm 2)

Next step in the process is to decompose the tensors (loaded in the previous section). These decomposed factors are used in next section for doing *collaboration* prediction. The method proposed in this paper for decomposition is inspired by CP Scoring using Heuristic (CPH) method of Dunlavy et al. [5] which has already proven successful. This method is based uses the well know CANDECOMP/PARAFAC (CP) [16] tensor decomposition which is analogous to Singular Value Decomposition (SVD) [10] and it converts a tensor into sum of rank one tensors. Given a three dimensional tensor $\mathcal{X}$ with size $J_a \times J_b \times J_c$ its CP decomposition is given by:

$$\mathcal{X} \approx \sum_{f=1}^{F} \lambda_f \mathbf{a}_f \circ \mathbf{b}_f \circ \mathbf{c}_f \tag{1}$$

where $\lambda_f \in R^+$, $\mathbf{a}_f \in R^{J_a}$, $\mathbf{b}_f \in R^{J_b}$, and $\mathbf{c}_f \in R^{J_c}$. Each of the products $\lambda_f \mathbf{a}_f \circ \mathbf{b}_f \circ \mathbf{c}_f$ is called the *components* whereas $\mathbf{a}_f$, $\mathbf{b}_f$ and $\mathbf{c}_f$ are called the $factors$ of the decomposition. Note that though $\|\mathbf{a}_f\|$=$\|\mathbf{b}_f\|$=$\|\mathbf{c}_f\|$=1 but these factors are not orthogonal to each other as it is the case in SVD. Also λ_f is the weight for the f^{th} component. The decomposition is unique, unlike other tensor decomposition methods, resulting in an attractive method for prediction as the factors can be used directly [5]. Note that matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ contain the factors $\mathbf{a}_f$,$\mathbf{b}_f$ and $\mathbf{c}_f$ as column vectors.

: GRAPH-PROB-VECTOR ($\mathbf{S}_g$, N_g, N_a)

[1]

$\mathbf{p_g}$ is the probability vector for edge likelihood of length N_g initialized to all one.

$i \in \{1, 2, ..., N_g\}$ s_i is the cardinality of hyperedge h_i. Each of the $\binom{s_i}{2}$ dyadic links d_p, of the hyperedge h_i as a clique. p s.t. $v_p \in d_p$ $\mathbf{p_g}(i) = \mathbf{p_g}(i) * \mathbf{S}_g(i, p)$

return $\mathbf{p_g}$

Based upon CPH the similarity between the object i and j is contained in a similarity matrix $\mathbf{S}$ as the entry at (i, j). This matrix is defined as follows:

$$\mathbf{S} = \sum_{k=1}^{K} \gamma_k \lambda_k \mathbf{a_k} \mathbf{b_k}^\top \tag{2}$$

where

$$\gamma_k = \frac{1}{T_{buf}} \left(\sum_{t=T-T_{buf}+1}^{T} \mathbf{c}_k(t) \right) \tag{3}$$

$\mathbf{a_k}\mathbf{b_k}^\top$ for the component k basically represent the similarity between the object pairs in in the k^{th} component. Let the similarity matrix for graph be $\mathbf{S_g}$ (from decomposition of $\mathcal{Z}_g$) and for hypergraph be $\mathbf{S_h}$ (from decomposition of $\mathcal{Z}_h$). Compression over T_{buf} number of past years (buffer) captures the intuition that only the recent past publications are relevant for prediction.

4.3 Predicting Collaborations

In this step the similarity matrices $\mathbf{S}_g$ and $\mathbf{S}_h$ are used for predicting the edges or hyperedges. Interpretation of the similarity matrix in our approach is as follows. $\mathbf{S}_g(i, j)$ is the likelihood of the i^{th} dyadic edge occurring in future and also contains vertex j. Similarly, for case of hypergraph $\mathbf{S}_h(i, j)$ is the likelihood of the i^{th} hyperedge along with vertex j inside it. In short after the tensor decomposition (and the subsequent compression along time dimension) our method outputs a similarity value for all the *actors* for each *collaboration* indicating how likely each of these *actors* can start working with this *collaboration*.

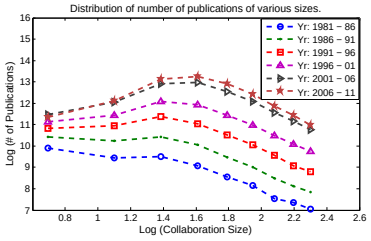

Figure 5: Log-Log Plot depicting No. of publications over different sizes of collaboration

4.3.1 *Hypergraph Case (Algorithm 3):*

If the reassurance of i^{th} hyperedge reoccurs and also contain j^{th} vertex is an event. Assuming that all these events for a particular i^{th} hyperedge for each of the vertices are independent the probability of i^{th} hyperedge reoccurs is defined as:

$$\mathbf{p_h}(i) = \prod_{p \in h_k} \mathbf{S}_h(i, p) \tag{4}$$

4.3.2 *Graph Case (Algorithm 4):*

Similarly, in case of graphs the probability of i^{th} edge reoccurring in future is:

$$\mathbf{q_g}(i) = \prod_{p \in g_k} \mathbf{S}_g(i, p) \tag{5}$$

The probability of i^{th} hyperedge reoccurring using the dyadic edge probabilities is:

$$\mathbf{p_g}(i) = \prod_{q \in D} \mathbf{q_g}(q) = \prod_{q \in D} \prod_{p \in g_k} \mathbf{S}_g(q, p) \tag{6}$$

where D is the set of dyadic edges that are contained in the clique representation of the i^{th} hyperedge.

The outcome of this whole process (Section 4) is these two vectors: $\mathbf{p_g}$ and $\mathbf{p_h}$. The i^{th} values of $\mathbf{p_g}$ and $\mathbf{p_h}$ are the likelihood of collaboration represented by the i^{th} hyperedge occurring in future as outputted by graph and hyperegraph models respectively. These vectors are used to generate the top-N list as detailed in the Section 5.

5. EXPERIMENTAL ANALYSIS

In this section we discuss the experimental setup used to evaluate the performance of the proposed approach. First section describes the dataset, data preprocessing and experimental setup. In the second section, we discuss the various experiments conducted and their analysis.

5.1 Dataset and Experimental Setup

We have evaluated the performance of the proposed approach using the popular DBLP dataset [20] containing publications from years 1930-2011. For the experiments the dataset is divided into training and test periods (*splits*) as shown in the Table 1 and Table 2. As shown in Table 1 the *splits* are designed with constant training period but variable testing periods. Table 2 contains *splits* with variable training periods and fixed length testing periods. Table 3 provides the statistics of the training and test set. It provides the total sum of edge counts across all the splits in two different ranges of splits: Split A.1 to A.5 and Split B.1 to B.5 as mentioned. However, only the No. of training and No. of old edges are useful statistics about the data for the proposed experiments.

The distribution Figure 5 is a *log-log* plot showing the distribution of publication counts of various collaboration sizes for different 5 year time periods of DBLP dataset. We observe that the distribution (Figure 5) across the different intervals follow a similar pattern. This shows that *splits* that were designed are equivalent as far as conducting experiments is concerned and no bias is involved.

As a preprocessing step, all the single author papers were removed since they do not capture relationships between authors.

For the CP Decomposition (CP-ALS) (that is required for Algorithm 1) Tensor Toolbox [1] is used. To find the parameter K for the CP-ALS algorithm we use the ensemble method approach proposed by Dunlavy et al [5] with $K = \{20, 40, ...200\}$. Also the parameter $T_{buf} = 3$ years is taken [5]. We have used the term graph and dyadic graph interchangeably.

5.2 Evaluation

In this section four experiments are described that evaluate our proposed approach and provide comparative analysis between dyadic and hypergraphs models. Each of the experiment below is conducted using some of the *splits*. The training period of each *split* is used to train the dyadic Graph or Hypergraph models using Algorithm 1. The algorithm is run for both graph and hypergraph case to return the edge

Table 1: Training-Testing Splits for variable Testing periods

Split No.	Training Size = 5 yrs	Test Size = 3 yrs	Test Size = 4 yrs	Test Size = 5 yrs
A.1	1981-85	1986-88	1986-89	1986-90
A.2	1986-90	1991-93	1991-94	1991-95
A.3	1991-95	1996-98	1996-99	1996-2000
A.4	1996-2000	2001-03	2001-04	2001-05
A.5	2001-05	2006-08	2006-09	2006-10

Table 2: Training-Testing Splits for variable Testing periods

Split No.	Training Size = 3 yrs	Training Size = 4 yrs	Training Size = 5 yrs	Test Size = 3 yrs
B.1	1983-85	1982-85	1981-85	1986-88
B.2	1988-90	1987-90	1986-90	1991-93
B.3	1993-95	1992-95	1991-95	1996-98
B.4	1998-2000	1997-2000	1996-2000	2001-03
B.5	2003-05	2002-05	2001-05	2006-08

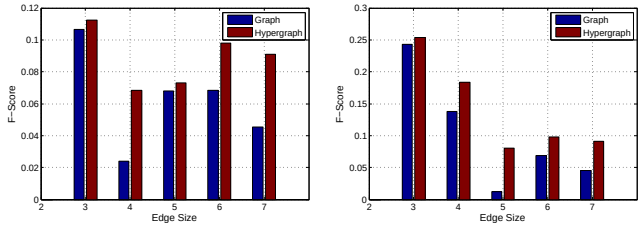

Figure 6: Experiment A: (a) AvgF-Score@100 (b) AvgF-Score@1000

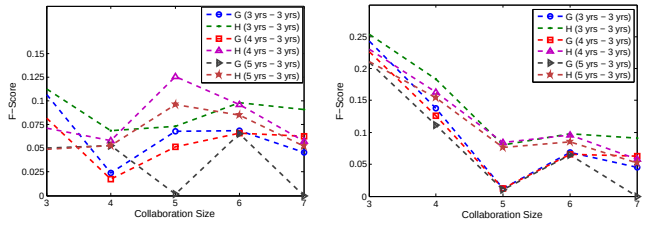

Figure 7: Experiment B (Variable Training Size): (a) AvgF-Score@100 (b) AvgF-Score@1000

($\mathbf{P_g}$) and hyperedge probability ($\mathbf{P_h}$) vectors. These probability vector contains likelihood values for collaborations of different sizes. Each vector is sorted in descending order and the list of top-N elements for each size is extracted. Out of these top-N elements the performance for each size collaboration over test set (old test edges in Table 3) is compared using the following metrics:

$$\text{Precision@N (Size-}h) = \frac{\text{\# of size 'h' collaborations correctly predicted from size 'h' top-}N\text{ list}}{N} \quad (7)$$

$$\text{Recall@N (Size-}h) = \frac{\text{\# of size 'h' collaborations correctly predicted from size 'h' top-}N\text{ list}}{\text{\# of actual size 'h' collaborations}} \quad (8)$$

Table 3: Total Edges for all the splits for different sizes and variable testing or training periods

Group Size	Split B.1-5: Training Size (Years)	Split B.1-5: No. of Training Edges	Split B.1-5: No. of Old Test Edges	Split B.1-5: No. of New Test Edges	Split A.1-5: Test Size (years)	Split A.1-5: No. of Training Edges	Split A.1-5: No. of Old Test Edges	Split A.1-5: No. of New Test Edges
2	3	192652	49537	210719	3	281203	52887	207369
3	3	129719	21242	176313	3	182562	22269	175286
4	3	60847	6121	93987	3	83421	6397	93711
5	3	23386	1502	39044	3	31907	1567	38979
6	3	9625	442	15990	3	13199	456	15976
7	3	4239	153	6851	3	5733	159	6845
2	4	239635	51746	208510	4	281203	61190	301689
3	4	158564	21964	175591	4	182562	24726	253978
4	4	73281	6319	93789	4	83421	6929	136111
5	4	28150	1546	39000	4	31907	1666	56502
6	4	11620	451	15981	4	13199	494	23266
7	4	5023	155	6849	4	5733	168	9921
2	5	281203	52887	207369	5	281203	65816	385474
3	5	182562	22269	175286	5	182562	25853	320284
4	5	83421	6397	93711	5	83421	7178	170603
5	5	31907	1567	38979	5	31907	1702	70431
6	5	13199	456	15976	5	13199	501	29005
7	5	5733	159	6845	5	5733	170	12339

$$\text{AvgPrecision@N (Size-}h) = \frac{\text{Sum of Precision@N (Size-}h\text{) for all splits}}{\text{Total \# of splits}} \quad (9)$$

$$\text{AvgRecall@N (Size-}h) = \frac{\text{Sum of Recall@N (Size-}h\text{) for all splits}}{\text{Total \# of splits}} \quad (10)$$

$$\text{AvgF-Score@N (Size-}h) = \frac{2 * \text{AvgPrecision@N (Size-}h) * \text{AvgRecall@N (Size-}h)}{\text{AvgPrecision@N (Size-}h) + \text{AvgRecall@N (Size-}h)} \quad (11)$$

This study considers collaborations of size $= \{2, 3, 4, 5, 6, 7\}$ as we are interested only in higher order collaborations (size=2 is used in the analysis as the trivial dyadic case). AverageF-Score@N and AverageF-Score@N are used in the experiments only for collaborations of size $= \{3, 4, 5\}$ across all the *splits* (over which the experiment is conducted). Collaboration of size $= \{6, 7\}$ the number of predictions are quiet less as compared to size $= \{3, 4, 5\}$ case. Therefore, for these cases all the predictions (rather than top-N) are used using the following metrics:

$$\text{Precision (Size-}h) = \frac{\text{\# of size 'h' collaborations correctly predicted}}{\text{Total \# of size 'h' predicted}} \quad (12)$$

$$\text{Recall (Size-}h) = \frac{\text{\# of size 'h' collaborations correctly predicted}}{\text{\# of actual size 'h' collaborations}} \quad (13)$$

$$\text{AvgPrecision (Size-}h) = \frac{\text{Sum of Precision (Size-}h\text{) for all splits}}{\text{Total \# of splits}} \quad (14)$$

$$\text{AvgRecall (Size-}h) = \frac{\text{Sum of Recall (Size-}h\text{) for all splits}}{\text{Total \# of splits}} \quad (15)$$

$$\text{AvgF-Score (Size-}h) = \frac{2 * \text{AvgPrecision (Size-}h) * \text{AvgRecall (Size-}h)}{\text{AvgPrecision (Size-}h) + \text{AvgRecall (Size-}h)} \quad (16)$$

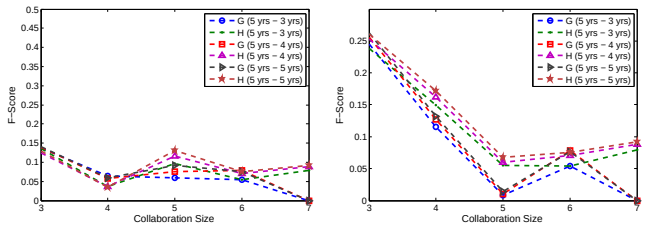

Figure 8: Experiment C (Variable Test Size): (a) AvgF-Score@100 (b) AvgF-Score@1000

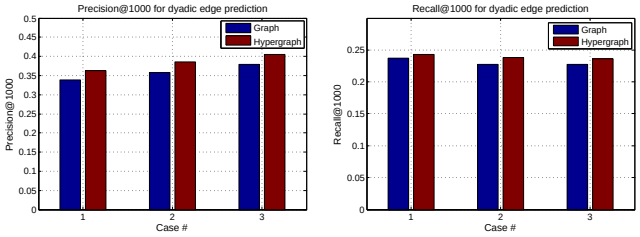

Figure 9: Experiment D (Dyadic Link Prediction): (a) AvgPrecision@1000 (b) AvgRecall@1000

In the expriments below AverageF-Score (Size-h) is used as a metric to evaluate the collaboration of size $= \{6, 7\}$ across all the *splits* (over which the experiment is conducted).

5.2.1 Experiment A

This experiment is conducted over the splits A.1 to A.5 for a fixed test period of 3 years (i.e. from Table 1 column 2 are the training periods and column 3 are the corresponding testing periods). AvgF-Score@100 and AvgF-Score@1000 are show in the Figure 6 (a),(b) for size $= \{3, 4, 5\}$. As shown in Figure 6(a),(b) for size= 3, graphs perform comparably with hypergraphs however for size= 4 prediction using hypergraphs show approx. 150% and 40% increase in F-Score for @100 and @1000 cases respectively. For size>5 Figure 6(a) and Figure 6(b) are identical showing the AvgF-Score. For size>5 graphs show similar trends as for size= 4 with performance degrading as size increases. As shown in figure 6(a) the F-Score for hypergraph perform better with an increase ranging from 25% for size= 5 to almost 100% for size= 7. This indicates that Hypergraphs maintain the higher order group information intact. However, owing to the limited training set for higher order (size> 6) collaborations hypergraph performance is reduced.

5.2.2 Experiment B

This experiment compares the prediction power of the two models: graph and hypergraphs, when trained over variable size training periods. The time period splits used in this case are B.1 to B.4 (which has fixed test period of 3 years) over training period size from 3 to 5 years as show in the Table 2. For size= $\{3, 4, 5\}$ AvgF-Score@100/1000 and AveragrF-Score for size> 5 curves for different training periods are shown in Figure 7(a),(b). As shown in Fig 7(a),(b) the F-Score curves for graph model are always lower than hypergraph curves for all size collaborations. Another interesting thing to note is that green curves of hypergraph are above pink and pink is above maroon for most sizes in both Figure 7(a),(b). Similar case is there for graphs (blue above red and red above grey). Thus, increasing the training period in several cases results in decrease in prediction power for both graphs and hypergraphs. This shows that the information about past can act as a noise and thus, decrease prediction accuracy.

5.2.3 Experiment C

To get further confidence in the prediction power of hypergraphs we ran experiments with predictions over variable testing periods from three to five years using the splits A.1 to A.4 (Table 1) and fixed training period size= 5 years. For size $= \{3, 4, 5\}$ the AvgF-Score@100 and AvgF-Scorel@1000 curves for different testing periods are shown in Figure 8(a),(b). As shown in Figure 8(a),(b), for size $= \{3, 4\}$ the graph model (curves colored blue, red and gray) is comparable to the green, pink and maroon curves (which represent hypergraph). However at higher order collaborations hypergraph outperform graphs (as inferred from the AvgF-Scores for size >= 5 shown in Figure 8(a),(b)).

5.2.4 Experiment D

This experiment analyzes the trivial case of predicting dyadic links. This experiment consists of three sub-experiments with the following testing and training combinations: A.1-A.5 with training of size = 5 years and test period size= 4 years (Case 1); B.1-B.4 with training period size = 4 years and test period size = 3 years (Case 2); and last, training using B.1-B.4 with training period of 3 years and testing using A.1-A.4 with test period size = 4 years (Case 3). These cases evaluate the dyadic link prediction under various combination of test and training periods. Results of this experiment are shown in the Figure 9(a),(b). It is clearly visible that the maroon bars (hypergraph) for different sub-experiments (Case 1 to 3) are always aslightly higher than blue bars (graphs). This shows that the performance of graphs and hypergraphs is comparable. Although, graphs are themselves sufficiently capture the information needed to predict dyadic links, the proposed tensor model for hypergraph is robust even to predict dyadic links.

5.2.5 Discussion

The above mentioned experiments corroborate about hypergraphs being a better and robust model for higher order collaboration than graphs. Results from these experiments demonstrate higher order collaboration (size>4) prediction is very difficult using dyadic graph based approach. Also collaborations of size> 2 hypergraphs in most cases provide better results with an average increase of approx. 45% in F-Score for different sizes $\in \{3, 4, 5, 6, 7\}$ (Figure 6). In fact our approach is robust for dyadic link prediction as well (Experiment D). Also combined inference from Experiment B and C shows that using the recent (past 3 years) publications we can prediction of a collaboration working together for an extended period of future 5 years.

6. CONCLUSION AND FUTURE WORK

In this work we have formulated the problem of higher order collaboration prediction. We show that these problems are much harder than the dyadic edge predictions. We make a pioneering attempt to address the *old edge* prediction problem. For this we propose a novel tensor decomposition based approach. We show that tensors are an excellent way to capture temporal hypergraphs since they perform much better in predicting collaborations of size greater than three (in general higher order hyperedges) in compar-

ison to the dyadic graph representation. Moreover, it also turns out that the hyper-incidence tensor model is robust for dyadic edge prediction as well. In this way we also provide a much needed explicit comparison between graphs and hypergraphs.

In the future we can work on modifying tensor based approach to address the harder problem of *new link* prediction. We can also use the power of tensors to predict exact future patterns (eg: giving a likelihood of publications n^{th} year in future) by using methods similar to [5]. Another interesting direction is to try out modeling K-size collaboration as a K-ary Tensors model.

Acknowledgment

The authors would also like to thank Dr. Tamara G. Kolda for her support and also for providing us with the code for Tensor Toolbox [1].

7. INTRODUCTION

The *proceedings* are the records of a conference. ACM seeks to give these conference by-products a uniform, high-quality appearance. To do this, ACM has some rigid requirements for the format of the proceedings documents: there is a specified format (balanced double columns), a specified set of fonts (Arial or Helvetica and Times Roman) in certain specified sizes (for instance, 9 point for body copy), a specified live area (18 × 23.5 cm [7" × 9.25"]) centered on the page, specified size of margins (1.9 cm [0.75"]) top, (2.54 cm [1"]) bottom and (1.9 cm [.75"]) left and right; specified column width (8.45 cm [3.33"]) and gutter size (.83 cm [.33"]).

The good news is, with only a handful of manual settings[1], the LaTeX document class file handles all of this for you.

The remainder of this document is concerned with showing, in the context of an "actual" document, the LaTeX commands specifically available for denoting the structure of a proceedings paper, rather than with giving rigorous descriptions or explanations of such commands.

8. THE *BODY* OF THE PAPER

Typically, the body of a paper is organized into a hierarchical structure, with numbered or unnumbered headings for sections, subsections, sub-subsections, and even smaller sections. The command `\section` that precedes this paragraph is part of such a hierarchy.[2] LaTeX handles the numbering and placement of these headings for you, when you use the appropriate heading commands around the titles of the headings. If you want a sub-subsection or smaller part to be unnumbered in your output, simply append an asterisk to the command name. Examples of both numbered and unnumbered headings will appear throughout the balance of this sample document.

Because the entire article is contained in the **document** environment, you can indicate the start of a new paragraph with a blank line in your input file; that is why this sentence forms a separate paragraph.

8.1 Type Changes and *Special* Characters

We have already seen several typeface changes in this sample. You can indicate italicized words or phrases in your text with the command `\textit`; emboldening with the command `\textbf` and typewriter-style (for instance, for computer code) with `\texttt`. But remember, you do not have to indicate typestyle changes when such changes are part of the *structural* elements of your article; for instance, the heading of this subsection will be in a sans serif[3] typeface, but that is handled by the document class file. Take care with the use of[4] the curly braces in typeface changes; they mark the beginning and end of the text that is to be in the different typeface.

You can use whatever symbols, accented characters, or non-English characters you need anywhere in your document; you can find a complete list of what is available in the *LaTeX User's Guide*[**?**].

8.2 Math Equations

You may want to display math equations in three distinct styles: inline, numbered or non-numbered display. Each of the three are discussed in the next sections.

8.2.1 Inline (In-text) Equations

A formula that appears in the running text is called an inline or in-text formula. It is produced by the **math** environment, which can be invoked with the usual `\begin. . .\end` construction or with the short form `$. . .$`. You can use any of the symbols and structures, from α to ω, available in LaTeX[**?**]; this section will simply show a few examples of in-text equations in context. Notice how this equation: $\lim_{n\to\infty} x = 0$, set here in in-line math style, looks slightly different when set in display style. (See next section).

8.2.2 Display Equations

A numbered display equation – one set off by vertical space from the text and centered horizontally – is produced by the **equation** environment. An unnumbered display equation is produced by the **displaymath** environment.

Again, in either environment, you can use any of the symbols and structures available in LaTeX; this section will just give a couple of examples of display equations in context. First, consider the equation, shown as an inline equation above:

$$\lim_{n\to\infty} x = 0 \tag{17}$$

Notice how it is formatted somewhat differently in the **displaymath** environment. Now, we'll enter an unnumbered equation:

$$\sum_{i=0}^{\infty} x + 1$$

and follow it with another numbered equation:

$$\sum_{i=0}^{\infty} x_i = \int_0^{\pi+2} f \tag{18}$$

just to demonstrate LaTeX's able handling of numbering.

[1]Two of these, the `\numberofauthors` and `\alignauthor` commands, you have already used; another, `\balancecolumns`, will be used in your very last run of LaTeX to ensure balanced column heights on the last page.

[2]This is the second footnote. It starts a series of three footnotes that add nothing informational, but just give an idea of how footnotes work and look. It is a wordy one, just so you see how a longish one plays out.

[3]A third footnote, here. Let's make this a rather short one to see how it looks.

[4]A fourth, and last, footnote.

Table 1: Frequency of Special Characters

Non-English or Math	Frequency	Comments
Ø	1 in 1,000	For Swedish names
π	1 in 5	Common in math
\$	4 in 5	Used in business
Ψ_1^2	1 in 40,000	Unexplained usage

8.3 Citations

Citations to articles [**?**, **?**, **?**, **?**], conference proceedings [**?**] or books [**?**, **?**] listed in the Bibliography section of your article will occur throughout the text of your article. You should use BibTeX to automatically produce this bibliography; you simply need to insert one of several citation commands with a key of the item cited in the proper location in the `.tex` file [**?**]. The key is a short reference you invent to uniquely identify each work; in this sample document, the key is the first author's surname and a word from the title. This identifying key is included with each item in the `.bib` file for your article.

The details of the construction of the `.bib` file are beyond the scope of this sample document, but more information can be found in the *Author's Guide*, and exhaustive details in the *LaTeX User's Guide*[**?**].

This article shows only the plainest form of the citation command, using `\cite`. This is what is stipulated in the SIGS style specifications. No other citation format is endorsed or supported.

8.4 Tables

Because tables cannot be split across pages, the best placement for them is typically the top of the page nearest their initial cite. To ensure this proper "floating" placement of tables, use the environment **table** to enclose the table's contents and the table caption. The contents of the table itself must go in the **tabular** environment, to be aligned properly in rows and columns, with the desired horizontal and vertical rules. Again, detailed instructions on **tabular** material is found in the *LaTeX User's Guide*.

Immediately following this sentence is the point at which Table 1 is included in the input file; compare the placement of the table here with the table in the printed dvi output of this document.

To set a wider table, which takes up the whole width of the page's live area, use the environment **table*** to enclose the table's contents and the table caption. As with a single-column table, this wide table will "float" to a location deemed more desirable. Immediately following this sentence is the point at which Table 2 is included in the input file; again, it is instructive to compare the placement of the table here with the table in the printed dvi output of this document.

8.5 Figures

Like tables, figures cannot be split across pages; the best placement for them is typically the top or the bottom of the page nearest their initial cite. To ensure this proper "floating" placement of figures, use the environment **figure** to enclose the figure and its caption.

This sample document contains examples of **.eps** and **.ps** files to be displayable with LaTeX. More details on each of these is found in the *Author's Guide*.

Figure 10: A sample black and white graphic (.eps format).

Figure 11: A sample black and white graphic (.eps format) that has been resized with the `epsfig` command.

As was the case with tables, you may want a figure that spans two columns. To do this, and still to ensure proper "floating" placement of tables, use the environment **figure*** to enclose the figure and its caption. and don't forget to end the environment with figure*, not figure!

Note that either **.ps** or **.eps** formats are used; use the `\epsfig` or `\psfig` commands as appropriate for the different file types.

8.6 Theorem-like Constructs

Other common constructs that may occur in your article are the forms for logical constructs like theorems, axioms, corollaries and proofs. There are two forms, one produced by the command `\newtheorem` and the other by the command `\newdef`; perhaps the clearest and easiest way to distinguish them is to compare the two in the output of this sample document:

This uses the **theorem** environment, created by the `\newtheorem` command:

THEOREM 1. *Let f be continuous on $[a, b]$. If G is an antiderivative for f on $[a, b]$, then*

$$\int_a^b f(t)dt = G(b) - G(a).$$

The other uses the **definition** environment, created by the `\newdef` command:

Definition 1. If z is irrational, then by e^z we mean the unique number which has logarithm z:

$$\log e^z = z$$

Two lists of constructs that use one of these forms is given in the *Author's Guidelines*.

There is one other similar construct environment, which is already set up for you; i.e. you must *not* use a `\newdef` command to create it: the **proof** environment. Here is a example of its use:

PROOF. Suppose on the contrary there exists a real number L such that

$$\lim_{x\to\infty} \frac{f(x)}{g(x)} = L.$$

Then

$$l = \lim_{x\to c} f(x) = \lim_{x\to c} \left[gx \cdot \frac{f(x)}{g(x)}\right] = \lim_{x\to c} g(x) \cdot \lim_{x\to c} \frac{f(x)}{g(x)} = 0 \cdot L = 0,$$

which contradicts our assumption that $l \neq 0$. □

Complete rules about using these environments and using the two different creation commands are in the *Author's*

Figure 13: A sample black and white graphic (.ps format) that has been resized with the `psfig` command.

Table 2: Some Typical Commands

Command	A Number	Comments
`\alignauthor`	100	Author alignment
`\numberofauthors`	200	Author enumeration
`\table`	300	For tables
`\table*`	400	For wider tables

Figure 12: A sample black and white graphic (.eps format) that needs to span two columns of text.

Guide; please consult it for more detailed instructions. If you need to use another construct, not listed therein, which you want to have the same formatting as the Theorem or the Definition[**?**] shown above, use the `\newtheorem` or the `\newdef` command, respectively, to create it.

A *Caveat* for the TEX Expert

Because you have just been given permission to use the `\newdef` command to create a new form, you might think you can use TEX's `\def` to create a new command: *Please refrain from doing this!* Remember that your LATEX source code is primarily intended to create camera-ready copy, but may be converted to other forms – e.g. HTML. If you inadvertently omit some or all of the `\def`s recompilation will be, to say the least, problematic.

9. CONCLUSIONS

This paragraph will end the body of this sample document. Remember that you might still have Acknowledgments or Appendices; brief samples of these follow. There is still the Bibliography to deal with; and we will make a disclaimer about that here: with the exception of the reference to the LATEX book, the citations in this paper are to articles which have nothing to do with the present subject and are used as examples only.

10. ACKNOWLEDGMENTS

This section is optional; it is a location for you to acknowledge grants, funding, editing assistance and what have you. In the present case, for example, the authors would like to thank Gerald Murray of ACM for his help in codifying this *Author's Guide* and the **.cls** and **.tex** files that it describes.

11. ADDITIONAL AUTHORS

Additional authors: John Smith (The Thørväld Group, email: `jsmith@affiliation.org`) and Julius P. Kumquat (The Kumquat Consortium, email: `jpkumquat@consortium.net`).

12. REFERENCES

[1] B. W. Bader and T. G. Kolda. Efficient matlab computations with sparse and factored tensors. *SIAM Journal on Scientific Computing*, 30(1):205–231, 2007.

[2] A.-L. Barabâsi, H. Jeong, Z. Néda, E. Ravasz, A. Schubert, and T. Vicsek. Evolution of the social network of scientific collaborations. *Physica A: Statistical Mechanics and its Applications*, 311(3):590–614, 2002.

[3] C. Berge and E. Minieka. *Graphs and hypergraphs*, volume 7. North-Holland publishing company Amsterdam, 1973.

[4] J. S. Coleman. Social capital in the creation of human capital. *American journal of sociology*, pages S95–S120, 1988.

[5] D. M. Dunlavy, T. G. Kolda, and E. Acar. Temporal link prediction using matrix and tensor factorizations. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(2):10, 2011.

[6] E. Estrada and J. A. Rodriguez-Velazquez. Complex networks as hypergraphs. *arXiv preprint physics/0505137*, 2005.

[7] R. Fagin. Degrees of acyclicity for hypergraphs and relational database schemes. *Journal of the ACM (JACM)*, 30(3):514–550, 1983.

[8] K. Faust. Centrality in affiliation networks. *Social networks*, 19(2):157–191, 1997.

[9] G. Ghoshal, V. Zlatić, G. Caldarelli, and M. Newman. Random hypergraphs and their applications. *Physical Review E*, 79(6):066118, 2009.

[10] G. H. Golub and C. Reinsch. Singular value decomposition and least squares solutions. *Numerische Mathematik*, 14(5):403–420, 1970.

[11] R. Guimerà, B. Uzzi, J. Spiro, and L. A. N. Amaral. Team assembly mechanisms determine collaboration network structure and team performance. *Science*, 308(5722):697–702, 2005.

[12] E.-H. Han, G. Karypis, V. Kumar, and B. Mobasher. Hypergraph based clustering in high-dimensional data sets: A summary of results. *IEEE Data Eng. Bull.*, 21(1):15–22, 1998.

[13] K. Kapoor, D. Sharma, and J. Srivastava. Weighted node degree centrality for hypergraphs. In *Network Science Workshop (NSW), 2013 IEEE 2nd*, pages 152–155. IEEE, 2013.

[14] J. S. Katz and B. R. Martin. What is research collaboration? *Research policy*, 26(1):1–18, 1997.

[15] S. Klamt, U.-U. Haus, and F. Theis. Hypergraphs and cellular networks. *PLoS computational biology*, 5(5):e1000385, 2009.

[16] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[17] M. E. Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences*, 98(2):404–409, 2001.

[18] K. J. Pearson and T. Zhang. On spectral hypergraph theory of the adjacency tensor. *arXiv preprint arXiv:1209.5614*, 2012.

[19] L. Putnam and C. Stohl. Bona fide groups. *Communication and group decision making*, pages 147–178, 1996.

[20] J. Tang, D. Zhang, and L. Yao. Social network

extraction of academic researchers. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 292–301. IEEE, 2007.

[21] C. Taramasco, J.-P. Cointet, and C. Roth. Academic team formation as evolving hypergraphs. *Scientometrics*, 85(3):721–740, 2010.

[22] Z. Tian, T. Hwang, and R. Kuang. A hypergraph-based learning algorithm for classifying gene expression and arraycgh data with prior knowledge. *Bioinformatics*, 25(21):2831–2838, 2009.

[23] A. Vazquez, R. Dobrin, D. Sergi, J.-P. Eckmann, Z. Oltvai, and A.-L. Barabási. The topological relationship between the large-scale attributes and local interaction patterns of complex networks. *Proceedings of the National Academy of Sciences*, 101(52):17940–17945, 2004.

[24] J. Xie and A. Chang. H-eigenvalues of signless laplacian tensor for an even uniform hypergraph. *Frontiers of Mathematics in China*, 8(1):107–127, 2013.

[25] D. Zhou, J. Huang, and B. Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. In *Advances in Neural Information Processing Systems*, pages 1601–1608, 2006.

APPENDIX

A. HEADINGS IN APPENDICES

The rules about hierarchical headings discussed above for the body of the article are different in the appendices. In the **appendix** environment, the command **section** is used to indicate the start of each Appendix, with alphabetic order designation (i.e. the first is A, the second B, etc.) and a title (if you include one). So, if you need hierarchical structure *within* an Appendix, start with **subsection** as the highest level. Here is an outline of the body of this document in Appendix-appropriate form:

A.1 Introduction

A.2 The Body of the Paper

A.2.1 Type Changes and Special Characters

A.2.2 Math Equations

Inline (In-text) Equations.

Display Equations.

A.2.3 Citations

A.2.4 Tables

A.2.5 Figures

A.2.6 Theorem-like Constructs

A Caveat for the TEX Expert

A.3 Conclusions

A.4 Acknowledgments

A.5 Additional Authors

This section is inserted by LaTeX; you do not insert it. You just add the names and information in the `\addition-alauthors` command at the start of the document.

A.6 References

Generated by bibtex from your .bib file. Run latex, then bibtex, then latex twice (to resolve references) to create the .bbl file. Insert that .bbl file into the .tex source file and comment out the command `\thebibliography`.

B. MORE HELP FOR THE HARDY

The sig-alternate.cls file itself is chock-full of succinct and helpful comments. If you consider yourself a moderately experienced to expert user of LaTeX, you may find reading it useful but please remember not to change it.