Community Discovery via Metagraph Factorization

YU-RU LIN
Arizona State University

JIMENG SUN
IBM T.J. Watson Research Center

HARI SUNDARAM
Arizona State University

AISLING KELLIHER
Arizona State University

PAUL CASTRO
IBM T.J. Watson Research Center

RAVI KONURU
IBM T.J. Watson Research Center

This work aims at discovering community structure in rich media social networks through analysis of time-varying, multi-relational data. Community structure represents the *latent* social context of user actions. It has important applications such as search and recommendation. The problem is particularly useful in the enterprise domain where extracting emergent community structure on enterprise social media can help in forming new collaborative teams, in expertise discovery, and in the long term reorganization of enterprises based on collaboration patterns. There are several unique challenges: (a) In social media, the context of user actions is constantly changing and co-evolving; hence the social context contains time-evolving multi-dimensional relations. (b) The social context is determined by the available system features and is unique in each social media platform; hence the analysis of such data needs to flexibly incorporate various system features. In this article we propose MetaFac (MetaGraph Factorization), a framework that extracts community structures from dynamic, multi-dimensional social contexts and interactions. Our work has three key contributions: (1) *metagraph*, a novel relational hypergraph representation for modeling multi-relational and multi-dimensional social data; (2) an efficient multi-relational factorization method for community extraction on a given metagraph; (3) an on-line method to handle time-varying relations through incremental metagraph factorization. Extensive experiments on real-world social data collected from an enterprise and the public Digg social media website suggest that our technique is scalable and is able to extract meaningful communities per social media contexts. We illustrate the usefulness of our framework through two prediction tasks: (1) in the enterprise dataset, the task is to predict users' future interests on tag usage, and (2) in the Digg dataset, the task is to predict users' future interests on voting and commenting Digg stories. Our prediction significantly outperforms baseline methods (including aspect model and tensor analysis), indicating the promising direction of using metagraphs for handling time-varying social relational contexts.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications—*Data mining*; H.3.3 [**Information Storage and Retrieval**]: *Information Search and Retrieval*—Information filtering; H.3.5 [**Information Storage and Retrieval**]: *Online Information Services*---Web-based services; I.5.3 [**Pattern Recognition**]: Clustering; J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Economics*
General Terms: Experimentation, Measurement, Algorithms, Theory, Human Factors
Additional Key Words and Phrases: MetaFac, metagraph factorization, relational hypergraph, non-negative tensor factorization, community discovery, dynamic social network analysis

---

1. INTRODUCTION

This work aims at discovering community structure in rich media social networks through analysis of the time-varying multi-relational data. Today, users routinely produce and consume media (e.g. blogs, YouTube, Digg) as well as interact with each other on social media platforms (e.g. Flickr, Facebook). These platforms allow a wide array of actions for managing and sharing media objects such as uploading photos, submitting and commenting on news stories, bookmarking and tagging, posting documents and creating web-links. These social media websites additionally support actions for interacting directly and indirectly with others, for example by sharing media and links with friends or commenting on photos uploaded by other users. These sites enable rich interactions between media and users, as well as complex social interactions among users. Some of the interactions can be implicit – two users may share similar tags, be interested in the same media themes, or even read a common member's posts. Understanding the context of these interactions – how they relate to other actions, users and media objects, can lead to improved functionality of the social media platforms as well as provide insight into the design of future online collaborative services for system developers.

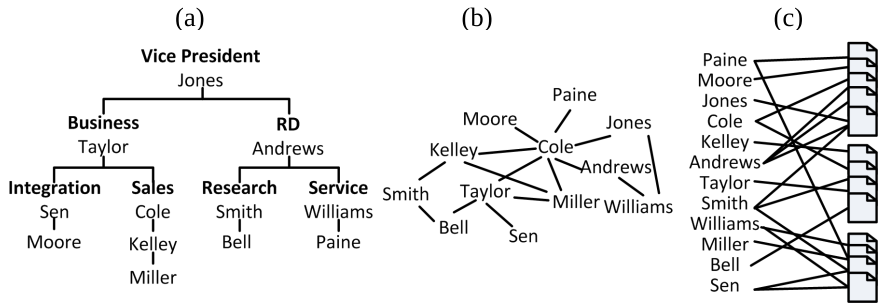

Figure 1: An example of various social relations in an enterprise: (a) formal organizational structure, (b) network of instant messaging, and (c) network of document sharing via bookmarking system.

As a motivating application, let us consider the use of social media in enterprises, which have increasingly embraced social media applications in an attempt to promote workplace collaboration. Such social media, including wikis, blogs, bookmark sharing, instant messaging, emails and calendar sharing can foster dynamic collaboration patterns that deviate from the formal organizational structure (e.g. cooperate departments, geographical places, etc.). As illustrated in Figure 1, people who are close in the formal organizational structure might be far apart in the instant messaging network (e.g. Sen and Moore). On the other hand, users' document access patterns might be related to their corporate roles as well as personal interests. The complex and dynamic interplay of various social relations and interactions in an enterprise reflects the day-to-day practice of collaboration. This process requires consideration of multiple aspects including how people assemble to tackle a task, how ideas are shared, what communication means are deployed, how task experts are identified, or how relevant information is found.

Understanding the latent community structure in an enterprise can have significant impact. For example:

- *Context-sensitive document search and recommendation*: Communities can reflect clustering phenomena in the dynamic heterogeneous social relations involving people and different types of information pieces, e.g. engineers may routinely access technical documents, while sales people may frequently read news about competitors. The clustering structures can be augmented in a search or recommendation system, e.g. when recommending a document, the document's rank can be boosted if the document has been recently viewed by people working closely with the given user.
- *Context-sensitive expert identification*: A main challenge in organizational learning is to leverage the expertise across organizational or even divisional boundaries, in an accurate and timely fashion [Monge and Contractor 2001; Powell et al. 1996]. This requires one to be able to find someone who has relevant expertise and can be easily reached, e.g. through various effective communication channels [Borgatti and Cross 2003]. The community structure extracted by our method, from multi-relational data such as organizational structure, daily communications and document access, can help identify experts located within the community of the information seeker.
- *Organizational study and reform*: Human organizations have been studied extensively through qualitative fieldwork (e.g. interviews) and quantitative participant surveys [Scandura and Williams 2000; Simsek and Veiga 2000; Stanton and Rogelberg 2001]. In such studies, the data is expensive to collect and the self-reported information may be inaccurate or biased by the participants' perception. In data-driven community discovery, the data is collected from social media, which capture more fine-grained human interactions and temporal information. Hence, the detected structures can be considered as patterns naturally arising from day-to-day communication means. These patterns can be used to simplify the amount of data in the heterogeneous social relations, for further organizational analysis including the functions and performance of sub-organizations. Such analysis can guide enterprise reorganization consistent with collaboration practice.

Discovering community structure from social media data has several challenges. *First*, in social media, the context of user actions is constantly changing and co-evolving with respect to other users' actions, emergent concepts and users' historic preferences. Hence the social context contains time-evolving multi-dimensional relations. *Second*, the social context is determined by the available system features that allow interactions on media objects and among people. Hence the social context is unique in each social media and the analysis of such data needs to flexibly incorporate various system features. When dealing with social media networks, there are very few studies (e.g. [Banerjee et al. 2007; Kemp et al. 2006]) that consider such data characteristics arising from social media.

In the following subsection, we give an overview of the problem and our solution approach.

1.1 Overview of the Problem

We are interested in the problem of discovering latent community structure in time-varying multi-relational social network data, which has three technical challenges:

(1) *Relational learning adaptable to different social media contexts*: A user's social media context is determined by the available system features that allow interactions on media objects and among people. The system features can vary across social media platforms (e.g. Digg vs. Facebook) or change with time, which requires the analysis of social media data to flexibly incorporate various combinations of relations. Most social network studies only consider fixed network modes, e.g. an author-paper network.

(2) *Evolutionary characterization of communities in time-varying social networks*: Given the time-varying network data, the extracted community structure needs to be able to explain the longitudinal human interaction patterns as well as the significant changes at certain times. Community evolution and dynamic social networks have not been studied in depth until recently (e.g. [Chi et al. 2007; Lin et al. 2008; Tantipathananandh et al. 2007; Yang et al. 2009]).

(3) *Analysis of multi-dimensional data*: Social media platforms usually archive action records consisting of different types of objects, e.g. a bookmarking record contains a user, a bookmarked URL, one or more tags, and a timestamp. Such records can be used to infer implicit interactions among people, provided the analysis can deal with multi-dimensional networked data. Existing high dimensional data mining techniques are usually computational intensive and not suitable for dealing with large scale social networked data.

Notions of community. There are different notions of a community. At the conceptual level, examples include communities of scientists working on similar areas of research [Girvan and Newman 2002] or authors of home pages who have some common interests [Adamic and Adar 2003]. At the operational level, community detection considers identifying the modular structure of a network where nodes represent individuals and links represent the interaction or similarity between individuals. Modules or communities are subset of nodes within which the links are dense, and between which the links are sparse. Based on such a definition, many community detection algorithms have been proposed (see section 2.1 for a brief survey).

We note that the prior definitions of a community are often too restrictive for analyzing rich-context social networks. First, people are observed as related to each other explicitly (e.g. direct collaborations or emails between people) and implicitly (e.g. having access to common web pages) at the same time. With multiple relations, it is unlikely to completely partition people into non-overlapping subsets. In this work, a *community* refers to a group of people who interact with objects (e.g. bookmarks) as well as with each other in a coherent manner – community members (including people and objects) are more likely to link to nodes also linked to by members within the community, and the links represent multiple relations. The main distinction between our definition and previous notions is that, in prior work, community identifications are based on a specific characteristic and rely on structures existing in a single type of relation (either interaction, common interests or similarity between individuals, e.g. [Adamic and Adar 2003; Girvan and Newman 2002]), while in our work, communities are identified based on structures across multiple types of relations. Our definition is motivated by social embeddedness [Granovetter 1985] which indicates the choices of individuals depend on how they are integrated in dense clusters or multiplex relations of social networks.

Our goal is to discover such communities from time-varying multi-relational social data. Based on the technical challenges discussed above, we identify three research questions addressed in this work:

(1) How to model such multi-relational social data?

(2) How to reveal the underlying communities that are consistent across multiple relations?

(3) How to track those communities over time?

1.2 Our Approach

We propose MetaGraph Factorization (MetaFac), a framework that extracts latent community structures from various social interactions. There are three key ideas in our framework:

(1) We propose *metagraph* (ref. Figure 2), a novel relational hypergraph representation for modeling multi-relational and multi-dimensional social data. A metagraph is a *multi-relational hypergraph* where each vertex represents a *facet* (i.e. a set of objects or entities of the same type), and each hyperedge represents the *relation* among facets. We use a metagraph to configure the relational context (a particular combination of facets and relations), which is the key to making our community analysis adaptable to various social contexts.

(2) We propose an efficient multi-relational factorization method for latent community extraction on a given metagraph. Given a social context, we represent multi-relational data as multiple conjunct data tensors where the conjunction is defined by a metagraph (e.g. based on the metagraph shown in Figure 2(b), the data tensor representing the "bookmark" relation is in conjunction with another data tensor representing the "join-project" relation due to the shared user facet). We formalize the latent community extraction as an optimization problem where the goal is to factorize the conjunct data tensors into a nonnegative superdiagonal core tensor multiplied by a nonnegative factor matrix along each facet. The optimization objective is defined as a function of the metagraph and the metagraph defines a combination of generalized KL-divergences, each of which corresponds to a relation represented by a data tensor on the given metagraph. The latent communities are determined by the factorization where the core tensor indicates the prior probability of each community and the factor matrices indicate the probability of each facet element given a community. We provide an efficient iterative algorithm that guarantees convergence to a local optimal solution, with the time complexity per iteration linear in the number of non-zero elements in all data tensors.

(3) We provide an on-line method to handle time-varying relations through incremental metagraph factorization. We incorporate an evolutionary clustering criterion in our metagraph-based optimization function so that the new community structure (of time t) to be extracted is consistent with prior community structure (of time t-1) and new observations (of time t). We introduce an additional cost to indicate how the new community structure deviates from the previous structure in terms of the generalized KL-divergence, and we provide an efficient iterative algorithm to search new community structures that do not significantly deviate from prior community structures.

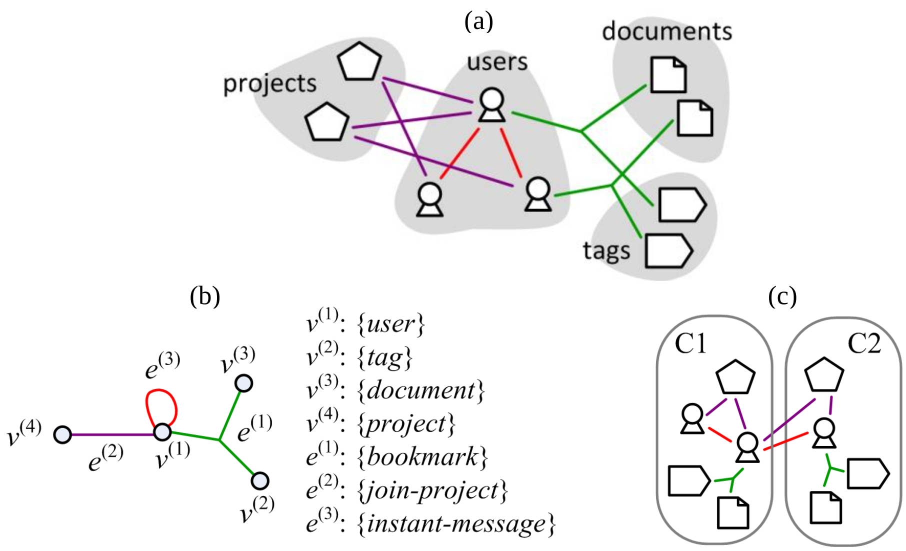

Figure 2: We propose a metagraph representation to model various social contexts – (a) users and related objects in an collaborative environment; (b) a metagraph representation for (a); (c) latent community structure in (a), where C1 and C2 are two clusters we want to discover.

We have conducted extensive experiments on real-world social media data collected from an enterprise (denoted as "ENTERPRISE" data) and the public Digg social media website. The results suggest that our technique is scalable and is able to extract meaningful communities based on the given social media context. We have found meaningful communities from the ENTERPRISE dataset. The communities exhibit distinct behavior corresponding to the engineering and sales subcultures within the enterprise. From the Digg data, we have found communities with distinct topical interests (e.g. gaming industry news, election news, world news, etc.) and their evolution corresponds to significant world or political events, such as the 2008 Summer Olympics and the Russia-Georgia conflict.

We further illustrate the usefulness of our framework through prediction tasks – to predict users' future interests in using specific bookmarking tags, as well as voting and commenting on Digg stories. The prediction performance is evaluated in terms of P@10 (the precision of the top 10 results) and NDCG (Normalized Discount Cumulative Gain), and we compare the prediction given by our community discovery framework with several baseline methods suitable for individual prediction tasks, including a widely-adopted collective filtering method (the probabilistic latent semantic analysis [Hofmann 1999] or pLSA) and a higher-order tensor decomposition method (PARAFAC). Our tag prediction in the ENTERPRISE dataset outperforms the baseline methods by 43-81% (P@10) or 27-72% (NDCG) on the average. We found that by leveraging cooperative relations (the department and directory of an employee), 19.46% of the users' future interests can be predicted by our method. Our voting and commenting prediction results in the Digg data outperforms the baseline methods by an order of magnitude. Specifically, our method outperforms the best baseline by 43% (P@10), 45% (NDCG), and 73% (P@10), 89% (NDCG), respectively. We further show that our prediction in both datasets can be further improved by incorporating historic community structure through incremental metagraph factorization and leveraging other relations through a metagraph. Although the experiment results indicate predicting users' future interests based on historic data is non-trivial, these results still suggest the utility of leveraging metagraphs to handle time-varying social relational contexts.

This article is a significant extension of our prior work [Lin et al. 2009b] and is our first comprehensive discussion on this subject. In this article, we include new experimental results, detailed algorithms and proofs. In particular there are several major extensions over prior work [Lin et al. 2009b].

(1) *Extended discussion on the problem*: In section 1, we use the proliferation of social media in enterprises to motivate the problem – how new collaborative structures can emerge through the use of social media and why extracting the emergent latent structures is important and challenging. In the problem formulation (section 4), we use an extended enterprise example to illustrate how diverse social contexts are modeled through the metagraph representation.

(2) *Details and proof of the algorithms*: In section 5, we formulate the community discovery on metagraph (metagraph factorization) as an optimization problem and provide an example to explain the formulation. We provide an iterative algorithm for solving the metagraph factorization problem and provide the proof of the convergence of this algorithm. In addition, we provide a pictorial view for illustrating the steps of finding a solution to the example problem and further discuss the probability interpretation of the solution.

(3) *New and extensive experiment results*: In section 7, we provide more experimental results from a real world enterprise dataset. We examine the effectiveness of our method on this dataset by using case studies and a prediction task. We further employ a forward-feature selection approach to select the best combination of

relations for prediction. The results suggest the applicability of our algorithm for extracting latent collaborative structures from dynamic social contexts.

This work is also related to our recent published work [Lin et al. 2009a], in which we propose the first generative model to extract communities based on both observed networked data and historic community structure. This work adopts a similar evolutionary clustering criterion to develop an on-line algorithm for extracting smoothly evolving community structures. The novel idea in this work is that we leverage the social embeddedness theory [Granovetter 1985] in sociology and develop a framework to deal with multiplex relations of social networks, which results in richer analytics and applications. Specifically, this work extends the prior work [Lin et al. 2009a] in three new aspects: (a) Multi-relations: Unlike our prior work [Lin et al. 2009a] that focuses on the pairwise relations between entities, this work considers a more general multi-relational networked data observed in rich-context social media. By introducing the metagraph representation, the algorithms proposed in this work can deal with multiple relations when they share the same set of entities. (b) Multi-dimensions: By using tensor (multilinear) algebra rather than matrix representation, the algorithms proposed in this work can handle arbitrarily many dimensions in a relation. (c) Rich analytics: Instead of using a symmetric matrix factorization, we extract community structures as a core tensor and a set of facet factor matrices which summarize communities from various dimensions (such as users, tags, feeds and comments) via different facet factors. The facet factors obtained across time can be used to detect community changes in different dimensions. They also allow an estimation of interactions between data entities in any two dimensions. We have employed this feature to predict users' potential interests in media objects such as tags and stories.

The experiments of this work focus on examining the proposed method in multi-relational network cases. Detecting communities in dynamic, uni-relational networks has been extensively discussed in our prior work [Lin et al. 2009a], where, by using synthetic datasets, we illustrate that our algorithm is capable of assigning meaningful community membership to a node to indicate the level of the node belonging to a community. We compare our algorithm with non-evolutionary as well as evolutionary algorithms in different noise conditions and show that our algorithm clearly outperforms baseline algorithms. For more results in uni-relational networks, we refer readers to our prior work [Lin et al. 2009a].

The rest of the paper is organized as follows. Section 2 reviews the related work. Section 3 introduces preliminaries and section 4 formalizes the problem. Sections 5 and 6 present our community extraction method on both static and dynamic multi-relational data. Section 7 describes experiments. Section 8 discusses the open issues of the presented approach and section 9 concludes.

2. RELATED WORK

Community discovery in rich media social networks deals with a constantly changing "mishmash" of interrelated users and media objects, which involves three aspects: (1) evolutionary characterization of communities in time-varying social networks (section 2.1), (2) analysis of multi-dimensional data (section 2.2), and (3) relational learning adaptable to different social contexts (section 2.3). To the best of our knowledge, our work is the first unified attempt to address all three aspects within a single problem.

2.1 Evolutionary Community Characterization

Communities in static networks. Community discovery has been extensively studied in social network analysis and other research areas. Many approaches, such as clique-based, degree-based and matrix-perturbation-based, have been proposed to extract cohesive subgroups from social networks [Wasserman and Faust 1994]. Recently, many effective algorithms have been proposed to find clustering structures from networked data, including spectral clustering algorithms [Chung 1997; Shi and Malik 2000] based on the eigenvectors of certain normalized similarity matrices. Newman and Girvan [2004] propose community extraction algorithms based on a modularity measure that quantifies the strength of community structure in a network. Yu et al. [2005] propose a soft clustering algorithm on graphs where the cluster memberships are assigned in a probabilistic way. This algorithm is closely related to the mixture model proposed by Newman and Leicht [2007] and the stochastic block model proposed by Holland et al. [1981]. Researchers have extended the stochastic block model in different ways. Airoldi et al. [2008] propose a mixed-membership stochastic block model. Kemp et al. [2004] propose a model that allows an unbounded number of clusters. Hofman et al. [2008] propose a Bayesian approach based on the stochastic block model to infer module assignments and to identify the optimal number of modules. Besides monopartite graphs, there is a growing body of work on community detection in bipartite graphs [Barber et al. 2008; Grujic et al. 2009; Harada et al. 2007]. A comprehensive review on community detection has been provided by Fortunato [Fortunato 2010].

Evolutionary Communities. Recent research based on the statistical properties of online social networks provides important insight regarding the structure and evolution of social behavior [Backstrom et al. 2006; Leskovec et al. 2008]. The structure of social interactions among people have been studied through unipartite or bipartite graphs, in which the community structure can be characterized by clustering methods [Sun et al. 2007] or a latent space model [Sarkar and Moore 2005], and the evolution of community structure is captured in terms of various criteria.

Kumar et al. [2006] study the evolution of the blogosphere in terms of the change of graph statistics and the "burstiness" of extracted communities. Aggarwal and Yu [2005] discuss an online approach to detect community changes in the graph streams based on the changes of edges (surrounding a set of seeded nodes) over a pre-defined time period. Berger-Wolf and Saia [2006] propose "metagroup statistics" to quantify the dynamics of social network structures based on interactions between groups of nodes over time. Spiliopoulou et al. [2006] propose a framework, MONIC, to monitor cluster transitions over time. They define a set of external transitions such as survive, split and disappear, to model transactions among different clusters and a set of internal transitions such as size and location transitions, to model changes within a community. Asur et al. [2007] introduce a family of events on both communities and individuals to characterize evolution of communities. The work of Palla et al. [2007] and Falkowski et al. [2006] use a two-step approach that extracts groups per time slice and then quantifies their evolution based on membership differences.

Sarkar and Moore [2005] propose a method that embeds nodes into latent spaces where the node coordinates at consecutive timesteps are regularized to avoid dramatic changes. Sun et al. [2007] use the Minimum Description Length principle to extract communities and to detect their changes. Tantipathananandh et al. [2007] propose an optimization-based approach for modeling dynamic community structure such that individuals do not change their "home community" too frequently and tend to interact with the home community most of the time, with a condition that only transitive interactions (across time) are allowed. Chi et al. [2007] use an evolutionary clustering criterion [Chakrabarti et al. 2006] and propose the first evolutionary spectral clustering algorithms for extracting clusters that smoothly evolving over time. They used graph cut as a metric for measuring community structures and community evolutions. Lin et al. [2009b] use a similar idea to extract community structures based on both observed networked data and historic community structure. They propose the first generative model to extract communities and their evolutions in a unified process. Yang et al. [2009] extend the generative model in the work of Lin et al. [2009b] and model the changes in community memberships over time explicitly by transition parameters, and a Bayesian treatment of parameter estimation is employed to improve point estimation results.

All these works restrict themselves to pair-wise relations between entities (e.g. user-user or user-document). In rich-context online social media, networked data consists of multiple co-evolving dimensions, e.g. users, tags, feeds, comments, etc. Collapsing such multi-way networks into pairwise networks results in the loss of valuable information, and the analysis of temporal correlation among multi-dimensions is difficult.

2.2 Multi-dimensional Mining

Existing techniques include tensor based analysis [Bader et al. 2006; Chi et al. 2008; Sun et al. 2007] or multi-graph mining [Zhu et al. 2007]. Tensor factorization is a generalized approach for analyzing multi-way interactions among entities. Note that a tensor represents complete interactions among all involved entities, which is a very strong assumption in social media since there might be events involving some but not all dimensions. Multi-graph mining considers joint factorization over two or more matrices. The combination of such matrices is domain-specific, e.g. in text mining, Zhu et al. [2007] propose a joint matrix factorization combining both linkage and document-term matrices to improve the hypertext classification. In social media, relations depend on the system features, which might consist of heterogeneous relations. Moreover, these relations may change over time in a social media website, which requires a more flexible relational model.

2.3 Relational Learning

Relational techniques such as Probabilistic Relational Models (PRMs) [Friedman et al. 1999] or Relational Markov Networks (RMNs) [Taskar et al. 2002] extend graph models to deal with various combinations of probabilistic dependency among entities. Such techniques can be computationally expensive, and may not scale to the large amount of data present in social media platforms. There have been relational learning techniques through pairwise relationships among entities [Bekkerman et al. 2005; Long et al. 2007; Singh and Gordon 2008; Tang et al. 2008; Wang et al. 2006]. For example, Singh and Gordon [2008] present a collective matrix factorization model that simultaneously factors several matrices with sharing factor parameters for entities participating in multiple pairwise relations. It shows that the collective matrix factorization, by exploiting the correlations between relations, can achieve higher prediction accuracy than factoring each matrix separately. The idea of exploiting correlation among multiple relations is similar to our work. However, one issue in their model is that each data matrix is factorized by two factors shared among relations. The stochastic constraint that each row of the factor matrices sums to one, usually leads to inconsistent interpretation for the cluster posterior derived from the shared factors of matrix factorization (i.e. the same factor may be interpreted as prior or posterior probability depending on whether it is the left or right factor of a matrix). In contrast, consistent cluster posterior can be derived from our model in a straightforward manner. Another common issue in such matrix factorization approach is that, when data has higher-order interactions, transforming the data into matrices incurs loss of information. Our work shares the same advantages as Kemp et al. [2006] and Banerjee et al. [2007], which can handle multiple higher order relations via tensor algebra, but their settings are different from ours.

Our unique contribution. In sum, social media analysis requires a flexible and scalable framework that exploits relational context defined by the system features of individual social media platforms. Such relational context is multi-dimensional, sparse (not all dimensions are involved in an event), and evolving over time. We propose the first graph-based tensor factorization algorithm to analyze the dynamics of heterogeneous social networks. Our method involves a novel “metagraph” representation based on hypergraphs [Berge 1976].

A hypergraph [Berge 1976] is a graph in which more than two vertices are linked by the same edge, and hence allowing for the manipulation of "sets of different types of objects." The theory of hypergraphs has been used to analyze data structures in several areas [Rugg 1984; Seidman 1981]. Specifically, the term “metagraph” has been used in a different context [Basu and Blanning 2007]. Basu and Blanning [2007] introduce metagraphs to depict the process of decision support systems and workflow management systems. The principal difference between their definition of a metagraph and ours is that their metagraph is defined to be a set of elements, along with a set of pairwise directed edges, and the manipulation of their metagraph is based on matrix algebra. By our definition, a metagraph is a hypergraph (with undirected hyperedges; see section 4.2 for a precise definition) and the formulation is based on tensor algebra. We use the term metagraph because of the following reason. In our metagraph, nodes are facets, and hyperedges refer to relations between the facets. In the traditional use of the term hypergraph, a hyperedge refers to a relation between specific instances (nodes refer to a certain facet instance, not a facet). So in our metagraph, there can be a hyperedge amongst, e.g., users, locations and jobs, whereas in a familiar hypergraph use, there is an edge amongst something specific, e.g., john, New York and secretary.

In the next section, we provide background on tensors.

3. PRELIMINARIES ON TENSORS

This section provides notations and essential background on tensors (section 3.1) and some basic operations (section 3.2). Tensor notations and operations provide a compact language that allows us to derive a formal representation of the heterogeneous social networks. For a more comprehensive discussion on tensors, we refer readers to Bader and Kolda's review [2006].

3.1 Tensors

A tensor is a mathematical representation of a multi-way array. The *order* of a tensor is the number of modes (or ways). A first-order tensor is a vector, a second-order tensor is a matrix, and a higher-order tensor has three or more modes. We use $\mathbf{x}$ as a vector, $\mathbf{X}$ as a matrix, and $\mathcal{X}$ as a tensor. The *dimensionality* of a mode is the number of elements in that mode. We use I_q to denote the dimensionality of mode q. E.g., the tensor $\mathbf{X} \in \Re_{+}^{I_1 \times I_2 \times I_3}$ has 3 modes with dimensionalities of I_1, I_2 and I_3, respectively. $\Re_+$ indicates that all elements of the tensor $\mathcal{X}$ have nonnegative values, which is usually the case for a data tensor. The (i_1,i_2,i_3)-element of a third-order tensor is denoted by $x_{i_1 i_2 i_3}$. Indices range from 1 to their capital version, e.g. $i_1=1,\ldots,I_1$.

3.2 Basic Operations

Mode-d matricization or *unfolding*: Matricization is the process of reordering the elements of an M-way array into a matrix. The mode-d matricization of a tensor $\mathbf{X} \in \Re^{I_1 \times \cdots \times I_M}$ is denoted by $\mathbf{X}_{(d)}$, i.e. $unfold(\mathbf{X}, d) = \mathbf{X}_{(d)} \in \Re^{I_d \times \prod_{q \in \{1,\ldots,M\}, q \neq d} I_q}$. Unfolding a tensor on mode d results in a matrix with height I_d and its width is the product of dimensionalities of all other modes.

The inverse operation is denoted as $\mathbf{X} = fold(\mathbf{X}_{(d)}) \in \Re^{I_1 \times \cdots \times I_M}$.

In general the unfolding operation can be defined on multiple modes. For example, we can define mode-(c,d) unfolding as $unfold(\mathbf{X}, (c,d)) = \mathbf{X}_{(c,d)} \in \Re^{(I_c \times I_d) \times \left(\prod_{q \in \{1,\ldots,M\}, q \neq c,d} I_q\right)}$. Unfolding a tensor on two modes c and d results in a three-way tensor. Similarly, we can define a *vectorization* operation $\mathbf{x}=vec(\mathcal{X})$, which linearizes the tensor into a vector.

Mode-d product: The mode-d matrix product of a tensor $\mathbf{X} \in \Re^{I_1\times\cdots\times I_M}$ with a matrix $\mathbf{U} \in \Re^{J\times I_d}$ is denoted by $\boldsymbol{\mathcal{X}}\times_d\mathbf{U}$ and results in a tensor of size $I_1\times\ldots\times I_{d-1}\times J\times I_{d+1}\times\ldots\times I_M$. Elementwise, we have $(\mathbf{X}\times_d\mathbf{U})_{i_1\cdots i_{d-1} j i_{d+1}\cdots i_M} = \sum_{i_d=1}^{I_d} x_{i_1 i_2\cdots i_M} u_{j i_d}$.

Mode-d accumulation: A mode-d accumulation or summation is defined as $acc(\mathbf{X}, d) = \mathbf{X}_{(d)}\mathbf{1} \in \Re^{I_d}$. The operation sums up all entries across all modes except for mode d, which results in a vector of length I_d. Accumulating a tensor on mode d can be obtained by unfolding the tensor on mode d into a matrix and then multiplying the matrix with an all-one vector. Like the unfolding operation, accumulation can be defined on multiple modes, e.g. a mode-(c,d) accumulation $acc(\mathbf{X}, (c, d)) = \mathbf{X}_{(c,d)}\times_3\mathbf{1} \in \Re^{I_c\times I_d}$, which will result in a matrix of size $I_c\times I_d$.

The *Kronecker product* of matrices $\mathbf{A}\in\Re^{I\times J}$ and $\mathbf{B}\in\Re^{K\times L}$ is denoted by $\mathbf{A}\otimes\mathbf{B}$. The result is a matrix of size $(IK)\times(JL)$ and defined by

$$\mathbf{A}\otimes\mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1J}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2J}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}\mathbf{B} & a_{I2}\mathbf{B} & \cdots & a_{IJ}\mathbf{B} \end{bmatrix}.$$

The *Khatri-Rao product* is the "matching columnwise" Kronecker product. The Khatri-Rao product of matrices $\mathbf{A}\in\Re^{I\times K}$ and $\mathbf{B}\in\Re^{J\times K}$ is denoted by $\mathbf{A}*\mathbf{B}$ and defined by $\mathbf{A}*\mathbf{B} = [\mathbf{a}_1\otimes\mathbf{b}_1 \quad \mathbf{a}_2\otimes\mathbf{b}_2 \quad \cdots \quad \mathbf{a}_K\otimes\mathbf{b}_K]$, where $\mathbf{a}_k$ and $\mathbf{b}_k$ are the k^{th} column vectors of $\mathbf{A}$ and $\mathbf{B}$ respectively.

Tensor decomposition or *factorization* decomposes a tensor into a core tensor multiplied by a matrix along each mode. Thus, in the three-way case where $\boldsymbol{\mathcal{X}}\in\Re^{I\times J\times K}$, we have $\boldsymbol{\mathcal{X}}\approx\boldsymbol{\mathcal{Z}}\times_1\mathbf{A}\times_2\mathbf{B}\times_3\mathbf{C}$, which means each element of the tensor $\boldsymbol{\mathcal{X}}$ is the product of the corresponding matrix elements multiplied by weight z_{pqr}, i.e. $x_{ijk} \approx \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R} z_{pqr} a_{ip} b_{jq} c_{kr}$. Here, $\mathbf{A}\in\Re^{I\times P}$, $\mathbf{B}\in\Re^{J\times Q}$ and $\mathbf{C}\in\Re^{K\times R}$ are called *factor matrices* or *factors*. The tensor $\boldsymbol{\mathcal{Z}}\in\Re^{P\times Q\times R}$ is called the *core tensor* and its elements show the level of interaction between different components. A special case of tensor decomposition is referred as CP/PARAFAC decomposition [Carroll and Chang 1970; Harshman 1970; Hitchcock 1927], where the core tensor is superdiagonal and P=Q=R. (A tensor $\mathbf{X} \in \Re^{I_1\times\cdots\times I_M}$ is called superdiagonal if $x_{i_1\cdots i_M} \neq 0$ only if $i_1=\ldots=i_M$.) The CP decomposition of a third-order tensor is then simplified as $x_{ijk} \approx \sum_{r=1}^{R} z_r a_{ir} b_{jr} c_{kr}$, as illustrated in Figure 3. We use [**z**] to denote a superdiagonal tensor, where the operation [·] transforms a vector **z** to a superdiagonal tensor by setting tensor element $z_{k\ldots k}$=$\mathbf{z}_k$ and other elements as 0. Thus the CP decomposition of a three-way tensor can be written as $\mathcal{X} \approx [\mathbf{z}] \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}$.

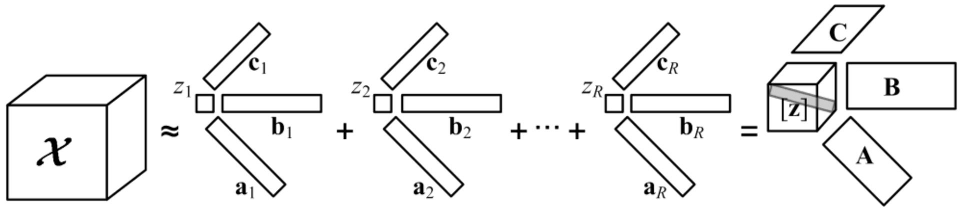

Figure 3: CP decomposition of a three-way tensor.

4. PROBLEM FORMULATION

This section defines the problem of discovering latent community structure that represents the context of user actions in social networks. We discuss the community discovery problem and the issues involved in section 4.1. In section 4.2, we propose metagraph – a representation of multi-relational social context. In section 4.3, we formally state the technical problems of community discovery from multi-relational social data represented by a metagraph.

4.1 Community Discovery

We formalize the community discovery problem as latent space extraction from multi-relational social data. Our goal is to discover latent community structures that represent the context of user actions in social media networks. We are interested in clusters of people who interact with each other in a coherent manner. Some of the interaction can be implicit, as when two users bookmark the same document, and the interactions can be further enhanced by other interactions. Hence we consider a community as a latent space of consistent interactions or relations among users and objects. In other words, interactions most likely occur when the involved users or objects belong to the same community.

As discussed in section 1.1, there are several issues involved in community discovery: (1) how to represent multi-relational social data, (2) how to reveal the latent communities consistently across multiple relations, and (3) how to track the communities over time. We address the first issue in the next subsection. The second and the third issues will be formalized as optimization problems (section 4.3) and the solutions are given in section 5 and 6, respectively.

4.2 Metagraph Representation

We introduce *metagraph*, a relational hypergraph for representing multi-relational and multi-dimensional social data. We use a metagraph to configure the relational context – this is the key that makes our community analysis adaptable to various social contexts, for example, an enterprise or a social media platform like Digg. We shall use an enterprise example to illustrate three concepts: *facet*, *relation*, and *relational hypergraph*.

As illustrated in Figure 4(a), assume we observe a set of users in an enterprise. These users might collaborate through different working projects, e.g. user u_1 and u_2 work for a project j_1, and user u_2 belongs to two projects, j_1 and j_2 at the same time. Collaboration can occur implicitly with the aid of social media. For example, these users can interact with each other through instant messenger or email, e.g. user u_3 frequently "IMs" with u_1 and u_2.

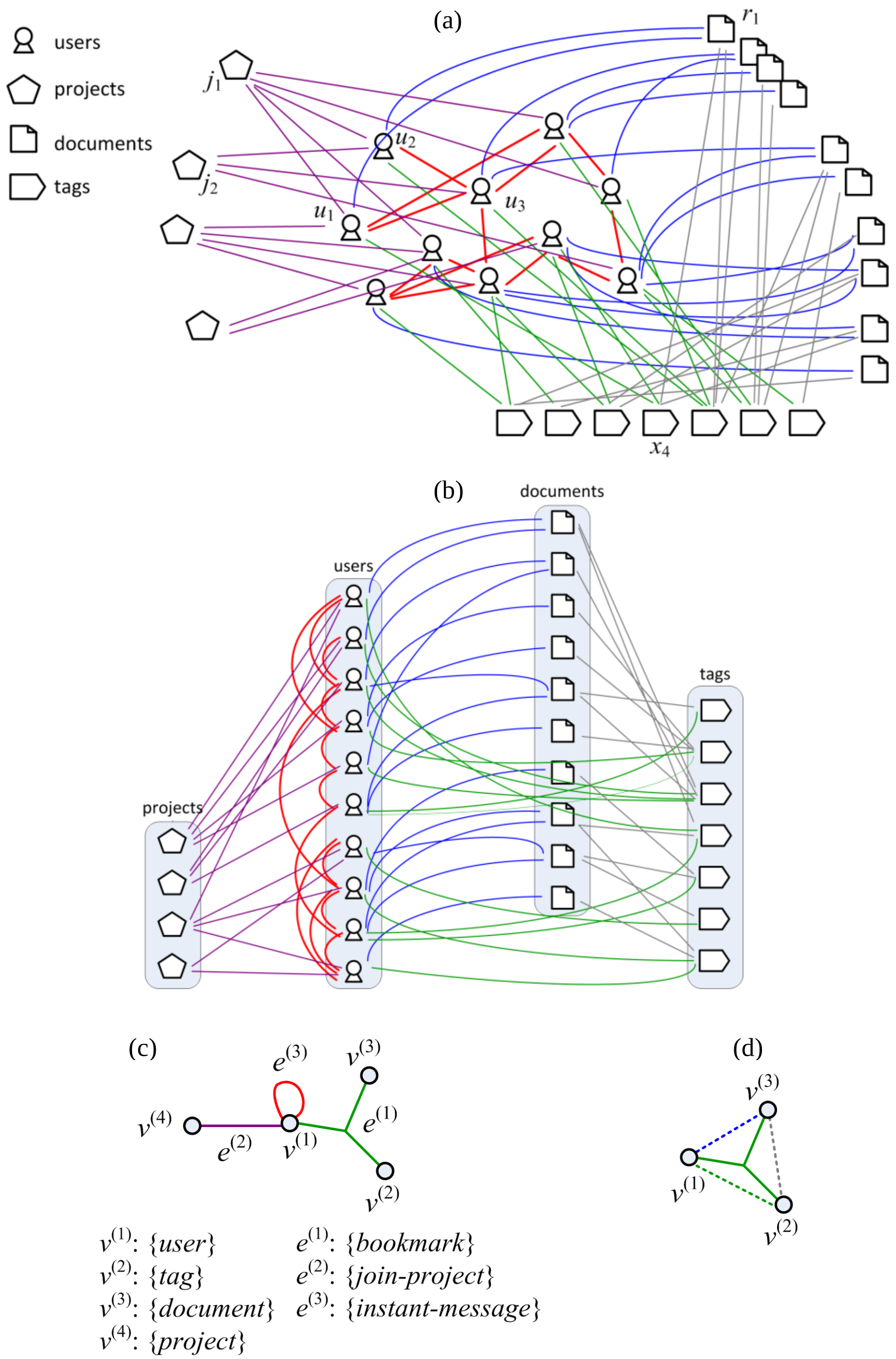

Figure 4: (a) A toy example that shows users and related objects in an enterprise. (b) Different types of objects interact with one another through different relationship. (c) A metagraph that represents the enterprise social context. (d) A hyperedge indicates that the incident facets are involved simultaneously.

Some of them might use a bookmarking system to share information (webpages or documents) that is relevant to their work. A typical bookmarking system allows a user to annotate shared documents with tags. A bookmarking activity involves a user, a document and one or more tags simultaneously, and it can be described as a tuple, e.g. $\langle u_1, r_1, x_4\rangle$ represents a user u_1 bookmarks a document r_1 with a tag x_4.

To represent such social context, let us assemble the same type of objects or entities, as in Figure 4 (b). We denote a set of objects or entities of the same type as a *facet*, e.g. a user facet is a set of users, a project facet is a set of projects. We denote the interactions among facets as a *relation*; a relation can involve two (i.e. binary relation) or more facets, e.g. the "join-project" relation involves two facets ⟨user, project⟩, and the "bookmark" relation involves three facets ⟨user, document, tag⟩. A facet may be removed from a metagraph if the facet entities do not interact with other facets, e.g. the set of bookmark entities might be omitted due to no interaction with other facets.

Formally, we denote the q-th facet as $v^{(q)}$ and the set of all facets as V. A set of instantiations of an M-way relation e on facets $v^{(1)}, v^{(2)},\ldots, v^{(M)}$ is a subset of the Cartesian product $v^{(1)}\times\ldots\times v^{(M)}$. We denote a particular relation by $e^{(r)}$ where r is the relation index. The observation of an M-way relation $e^{(r)}$ is represented as an M-way data tensor $\mathcal{X}^{(r)}$.

Now we introduce a *multi-relational hypergraph*, denoted as *metagraph*, as shown in Figure 4 (c), to describe the combination of relations and facets in a specific social context. A hypergraph is a graph where edges, called *hyperedges*, connect to any number of vertices. The idea is to draw a hypergraph such that each vertex represents a facet, and an M-way hyperedge represents the set of interactions of M facets. Such a hypergraph is a "graph about graphs" because a hyperedge on a metagraph represents many instance hyperedges and a vertex on a metagraph represents many data entities in the observed networks. By using a metagraph, we can represent a diverse set of relational contexts in social networks. Note that a metagraph defines a particular structure of interactions among facets, as opposed to specific interactions among facet elements.

Formally, for a set of facets $V=\{v^{(q)}\}$ and a set of relations $E=\{e^{(r)}\}$, we construct a metagraph $G=(V,E)$. To reduce notational complexity, V and E also represent the set of all vertex and edge indices respectively. A hyperedge/relation $e^{(r)}$ is a tuple consisting of vertices as its elements; $e^{(r)}$ is said to be *incident* to a facet/vertex $v^{(q)}$ if $v^{(q)}$ is an element in $e^{(r)}$, which is represented by $v^{(q)}\sim e^{(r)}$ or $e^{(r)}\sim v^{(q)}$. E.g., in Figure 4 (c), the vertex $v^{(1)}$ represents the user facet, and the hyperedge $e^{(1)}=\langle v^{(1)},v^{(2)},v^{(3)}\rangle$ represents the "bookmark" relation. Note that the three-way hyperedge $e^{(1)}$ implies interactions between any two of the three facets, as shown in Figure 4(d). However, the hyperedge representation is more informative than a complete subgraph (i.e. a clique), e.g. the triangle in Figure 4 (d), as it indicates all incident facets are involved in the corresponding relation *simultaneously*. We summarize our notations in Table 1.

Symbol	Description
$\mathbf{x}$	a vector (boldface lower-case letter)
$\mathbf{X}$	a matrix (boldface capital letter)
$\boldsymbol{\mathcal{X}}$	a tensor (boldface Euler script letter)
$I_1,\ldots,I_M$	the dimensionality of mode 1, …, M
$v^{(q)}$	a vertex $v^{(q)}\in V$ represents the facet $v^{(q)}$
$e^{(r)}$	a hyperedge $e^{(r)}\subseteq V$ represents the relation $e^{(r)}$
V	the set of all facets $V=\{v^{(q)}\}$, or the set of all vertex indices
E	the set of all relations $E=\{e^{(r)}\}$, or the set of all hyperedge indices
G	a metagraph $G=(V,E)$, where V is a set of facets and E is a set of relations
K, L	Constants

Table 1: Description of notations.

The concepts involved in a metagraph may also be related to the Entity-Relationship models (ERMs) [Chen 1976], which are abstract representations of data for database modeling. The commonality is that both metagraphs and ERMs are used to represent the mathematical concepts about M-way relations. Although we could consider a facet in a metagraph as an *entity set*, and a relation in a metagraph as a *relationship set* in an ERM, a range of notions used in the ERM modeling, such as attributes and key constraints, do not have corresponding semantics in our mathematical representation. To prevent misconceptions, we propose using a metagraph as a succinct representation of our model instead of deriving terminologies from the ERMs. Besides, the metagraph representation has a corresponding probabilistic interpretation about event spaces, which will be discussed in section 5.2.

4.3 Community Discovery using Metagraphs

We assume consistent interactions in a community (ref. section 4.1). Therefore, the interaction between any two entities (users or media objects) i and j in a community k, written as x_{ij}, can be viewed as a function of the relationships between community k with entity i, and k with entity j. The function can be considered to be stochastic. By letting $p_{k\rightarrow i}$ indicate how likely an interaction in the k-th community involves the i-th entity and p_k be the probability of an interaction in the k-th community, we can express x_{ij} by $x_{ij}\approx\sum_k p_k\cdot p_{k\rightarrow i}\cdot p_{k\rightarrow j}$. Figure 5(a) illustrates how the interactions in a user-document network are captured by two communities. In Figure 5 (a), the interaction between user u_7 and document r_6 is captured by community C1 and C2 in terms of $p_{C1} \cdot p_{C1\to u_7} \cdot p_{C1\to r_6} + p_{C2} \cdot p_{C2\to u_7} \cdot p_{C2\to r_6}$, where p_{C1} and p_{C2} represent the probability of an interaction in C1 or C2 (visually indicated by the ellipse sizes), and where $p_{C1\to u_7}$, $p_{C1\to r_6}$, $p_{C2\to u_7}$, $p_{C2\to r_6}$ represent how likely the two communities involves user u_7 and document r_6, respectively (visually indicated by the thickness of the links connected to communities C1 and C2).

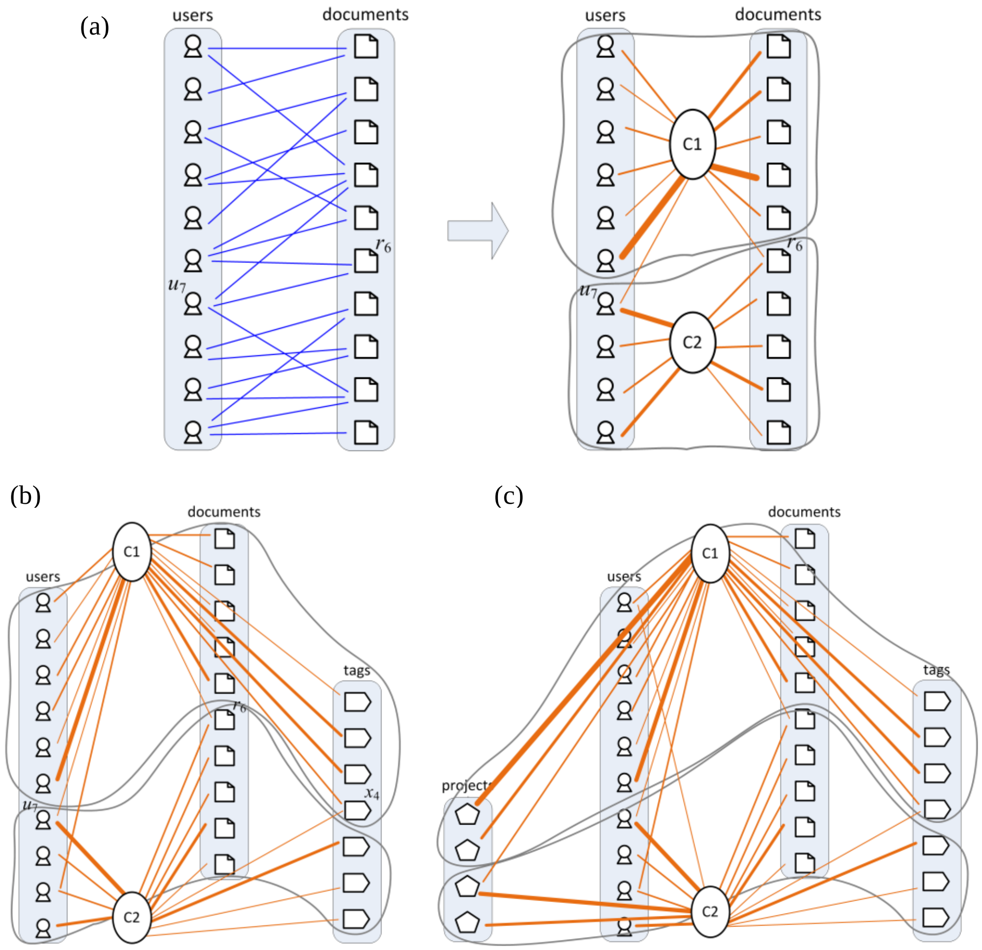

Figure 5: (a) An illustration of how two communities capture consistent interactions in a user-document network, e.g. the interaction between user u_7 and document r_6 are captured by community C1 and C2 in terms of their relationship with C1 and C2. (b) An illustration of how two communities capture the three-way interaction among users, documents and tags. (c) We seek to find communities that capture all the relations shown in Figure 4.

Likewise, $x_{i_1 i_2 i_3}$ is a three-way interaction among entities i_1, i_2 and i_3 and is factorized as follows: $x_{i_1 i_2 i_3} \approx \sum_k p_k \cdot p_{k\to i_1} \cdot p_{k\to i_2} \cdot p_{k\to i_3}$. Figure 5 (b) illustrates how the three-way interaction among users, documents and tags is captured by two communities, e.g. the interaction among user u_7, document r_6 and tag x_4 is captured by C1 and C2 in terms of $p_{\mathrm{C1}} \cdot p_{\mathrm{C1}\to u_7} \cdot p_{\mathrm{C1}\to r_6} \cdot p_{\mathrm{C1}\to x_4} + p_{\mathrm{C2}} \cdot p_{\mathrm{C2}\to u_7} \cdot p_{\mathrm{C2}\to r_6} \cdot p_{\mathrm{C2}\to x_4}$. A set of such interactions among entities in facet $v^{(1)}$, $v^{(2)}$ and $v^{(3)}$ can be written as follows:

$$\mathbf{X} \approx \sum_{k=1}^{K} p_k \mathbf{u}_k^{(1)} \circ \mathbf{u}_k^{(2)} \circ \mathbf{u}_k^{(3)} = [\mathbf{z}] \prod_{m=1}^{3} \times_m \mathbf{U}^{(m)}, \qquad \text{<1>}$$

where, $\mathbf{X} \in \Re_+^{I_1 \times I_2 \times I_3}$ is the data tensor representing the observed three-way interactions among facets $v^{(1)}$, $v^{(2)}$ and $v^{(3)}$, $[\mathbf{z}]$ is a nonnegative superdiagonal core tensor, and where $p_{k\to i_q}$ is written as an (i_q,k)-element of $\mathbf{U}^{(q)}$ for q=1,2,3. $\mathbf{U}^{(q)}$ is a $I_q \times K$ matrix, where I_q is the size of $v^{(q)}$. The probabilities of communities are elements of $\mathbf{z}$, i.e. $p_k = \mathbf{z}_k$. This is similar to the CP decomposition of a tensor (ref. section 3.2), except that the core tensor $[\mathbf{z}]$ and the factor matrices $\{\mathbf{U}^{(q)}\}$ are constrained to contain nonnegative probability values. Under the nonnegative constraints and each column of $\{\mathbf{U}^{(q)}\}$ must sum to one, the 3-way tensor factorization is equivalent to the three-way aspect model in a three-dimensional co-occurrence data [Popescul et al. 2001].

The nonnegative tensor decomposition can be viewed as community discovery in a single relation. The interactions in social media networks are more complex – usually involving multiple two- or multi-way relations. In the enterprise example, we seek to find communities that capture all the relations shown in Figure 4, as illustrated in Figure 5 (c). By using metagraphs, we represent a diverse set of relational contexts in the same form and define the community discovery problem on a metagraph, with the following two technical issues.

The first issue is how to extract community structure as coherent interaction latent spaces from observed social data defined on a metagraph, which is formally stated as follows.

Problem (Metagraph Factorization, or MF): given a metagraph G=(V,E) and a set of observed data tensors $\{\mathcal{X}^{(r)}\}_{r\in E}$ defined on G, find a nonnegative superdiagonal core tensor [$\mathbf{z}$] and factors $\{\mathbf{U}^{(q)}\}_{q\in V}$ for corresponding facets $V=\{v^{(q)}\}$ so as to explain the distribution of the observed data.[1]

The second issue concerns the dynamic nature of human activities – those interactions might be consistent during a short time period but are unlikely to be stable all the time. The problem, how to extract community structure as coherent interaction latent spaces from time evolving data given a metagraph, is defined below.

Problem (Metagraph Factorization for Time evolving data, or MFT): given a metagraph $G=(V,E)$ and a sequential set of observed data tensors $\{\boldsymbol{\mathcal{X}}_t^{(r)}\}_{r\in E}$ defined on G for time t=1,2,…, find a nonnegative core tensor [$\mathbf{z}_t$] and factors $\{\mathbf{U}_t^{(q)}\}_{q\in V}$ corresponding to facets $V=\{v^{(q)}\}$ for each time t so as to explain the distribution of the observed data.

We will present our method in two steps: (1) present a solution to MF (section 5.1); (2) extend the solution to solve MFT (section 5.2).

5. METAGRAPH FACTORIZATION

This section presents our solution to the metagraph factorization problem (MF). Our method relies on formulating MF as an optimization problem (section 5.1). We then provide an algorithm to find a solution to the optimization objective and discuss its computational complexity (section 5.2).

5.1 Optimization Objective

The MF problem can be stated in terms of optimization. Let us first consider a toy example. Assume we are given a metagraph $G=(V,E)$ with three vertices $V=\{v^{(1)}, v^{(2)}, v^{(3)}\}$ and two 2-way hyperedges $E=\{e^{(a)},e^{(b)}\}$ that describe the interactions among these three facets, as shown in Figure 6. The observed data corresponding to the hyperedges are two second-order data tensors (i.e. matrices) $\{\boldsymbol{\mathcal{X}}^{(a)},\boldsymbol{\mathcal{X}}^{(b)}\}$ with facets $\{v^{(1)}, v^{(2)}\}$ and $\{v^{(2)}, v^{(3)}\}$ respectively. The facet $v^{(2)}$ is shared by both tensors.

The goal is to extract community structure from data tensors, through finding a nonnegative core tensor [$\mathbf{z}$] and factors $\{\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(3)}\}$ corresponding to the three facets. The core tensor and factors need to consistently explain the data, i.e. we can approximately express the data by $\boldsymbol{\mathcal{X}}^{(a)}\approx[\mathbf{z}]\times_1\mathbf{U}^{(1)}\times_2\mathbf{U}^{(2)}$ and $\boldsymbol{\mathcal{X}}^{(b)}\approx[\mathbf{z}]\times_2\mathbf{U}^{(2)}\times_3\mathbf{U}^{(3)}$, as in eq<1>. The core tensor [$\mathbf{z}$] and facet $\mathbf{U}^{(2)}$ are shared by the two approximations, and the

[1] Since E also represents the set of all edge indices, the notations $r\in E$ and $e^{(r)}\in E$ are interchangeable. Likewise, $q\in V$ and $v^{(q)}\in V$ are interchangeable.

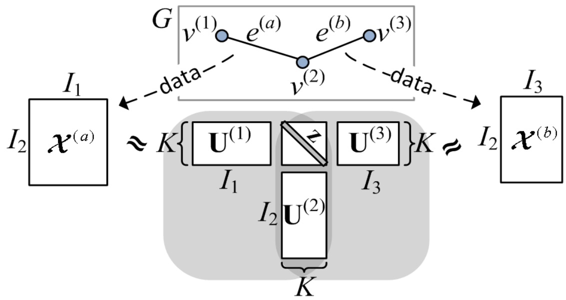

Figure 6: A toy example of the metagraph factorization (MF). Given observed data tensors $\{\mathcal{X}^{(a)},\mathcal{X}^{(b)}\}$ and a metagraph G that describes the interaction among facets $\{v^{(1)}, v^{(2)}, v^{(3)}\}$, find consistent community structure expressed by core tensor [z] and facet factors $\{\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(3)}\}$.

length of $\mathbf{z}$ is determined by the number of latent spaces (communities) to be extracted. Since both the left- and the right-hand side of the approximation are probability distributions, it is natural to use the KL-divergence (denoted as $D(\cdot\|\cdot)$) as a measure of approximation cost. To simultaneously reduce two approximation costs we can define a cost function as:

$$D(\mathbf{X}^{(a)} \| [\mathbf{z}] \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)}) + D(\mathbf{X}^{(b)} \| [\mathbf{z}] \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)}), \qquad \text{<2>}$$

where $D(\mathcal{A}\|\mathcal{B})=\sum_i (\mathbf{a}_i \log \mathbf{a}_i/\mathbf{b}_i - \mathbf{a}_i + \mathbf{b}_i)$ is the generalized KL-divergence (also called I-divergence) between tensor $\mathcal{A}$ and $\mathcal{B}$ and $\mathbf{a} = vec(\mathcal{A})$, $\mathbf{b} = vec(\mathcal{B})$, $\sum_i \mathbf{a}_i = \sum_i \mathbf{b}_i = 1$.

The solution to eq.<2> will be an MF solution for the metagraph in Figure 6. We observe three things in this example: In eq.<2>, each $D(\cdot\|\cdot)$ corresponds to a hyperedge, each tensor product operation corresponds to how facets are incident to a hyperedge, and the summation operation corresponds to all hyperedges on the graph. We then generalize eq.<2> to any metagraph G, as follows.

Given a metagraph $G=(V,E)$, the objective is to factorize all data tensors such that all tensors can be approximated by a common nonnegative superdiagonal core tensor [**z**] and a shared set of nonnegative factors $\{\mathbf{U}^{(q)}\}$, i.e., to minimize the following cost function:

$$\begin{aligned} & J(G) = \sum_{r \in E} D(\mathbf{X}^{(r)} \| [\mathbf{z}] \prod_{m: v^{(m)} \sim e^{(r)}} \times_m \mathbf{U}^{(m)}) \\ & s.t\ \mathbf{z} \in \Re_+^{1\times K}, \mathbf{U}^{(q)} \in \Re_+^{I_q \times K}\ \forall q, \sum_i \mathbf{U}_{ik}^{(q)} = 1\ \forall q \forall k \end{aligned} \qquad \text{<3>}$$

where K is the number of communities, and $D(\cdot\|\cdot)$ is the generalized KL-divergence as described above. The constraint that each column of $\{\mathbf{U}^{(q)}\}$ must sum to one is added due to the conditionally independent assumption, that is, the probability of an occurrence of a relation on an entity is independent of other entities in a community.

Without loss of generality, we have assumed the elements in each of the data tensors are nonnegative and sum to one such that each data tensor represents a distribution over all possible co-occurrences of elements in incident facets. With this normalization, we can balance different types of relations in the objective, since the amount of data in each relation can vary. Eq.<3> can be easily extended to incorporate weights on relations (to encode the importance of different relations).

5.2 Algorithm

We now present an algorithm for identifying K communities from data tensors by finding a solution to the objective function defined in eq.<3> above. From eq.<3>, it is difficult to guarantee a global minimum solution[2], as eq.<3> is not convex in all variables. In the following we derive a local minimum solution to eq.<3> by employing the concavity of the log function in the generalized KL-divergence.

Theorem 1. *The cost defined in eq.<3> is non-increasing under the following update rules and therefore converges to a (local) optimal solution to the MF problem*:

$$\mathbf{z}_k \leftarrow \frac{1}{L}\sum_{r\in E}\sum_{i_1\ldots i_{M_r}} \mathbf{X}^{(r)}_{i_1\ldots i_{M_r}} \mu^{(r)}_{i_1\ldots i_{M_r}k} \ , \qquad \text{<4>}$$

$$\mathbf{U}^{(q)}_{i_q k} \leftarrow \frac{1}{L_q}\sum_{l:e^{(l)}\sim v^{(q)}}\sum_{i_1\ldots i_{q-1}i_{q+1}\ldots i_{M_l}} \mathbf{X}^{(l)}_{i_1\ldots i_{M_l}} \mu^{(l)}_{i_1\ldots i_{M_l}k}, \qquad \text{<5>}$$

then normlize such that each column of $\mathbf{U}^{(q)}$ *sum to one,*

where $\mathbf{z}$ *is a length* K *vector,* $L=|E|$ *denotes the total number of hyperedges on* G, $L_q=|\{l{:}e^{(l)}\sim v^{(q)}\}|$ *denotes the number of hyperedges incident to* $v^{(q)}$, *and*

$$\mu^{(r)}_{i_1\cdots i_{M_r}k} \leftarrow \frac{\mathbf{z}_k \prod_{m:v^{(m)}\sim e^{(r)}} \mathbf{U}^{(m)}_{i_m k}}{([\mathbf{z}]\prod_{m:v^{(m)}\sim e^{(r)}} \times_m \mathbf{U}^{(m)})_{i_1\cdots i_{M_r}}} \ . \qquad \text{<6>}$$

The proof of Theorem 1 is provided in the appendix. Because of the column normalization step of $\mathbf{U}^{(q)}$, we can omit dividing by L_q in eq.<5>. The initial values of $\mathbf{z}$ and $\{\mathbf{U}^{(q)}\}$ can be drawn from a uniform distribution. This iterative update algorithm is a generalization of the algorithm proposed by Lee and Seung [2001] for solving the single nonnegative matrix factorization problem. In metagraph factorization, the update for the

[2] The NP-hardness results for nonnegative matrix factorization established by Vavasis [Vavasis 2007] suggest that solving the nonnegative tensor factorization to optimality may also be a difficult problem.

core tensor [**z**] depends on all hyperedges on the metagraph, and the update for each facet factor $\mathbf{U}^{(q)}$ depends on the hyperedges incident to the facet.

The computation in eq.<4>–<6> can be time-consuming due to the high dimensionalities of tensors. We now discuss an efficient implementation of the update rules. In eq.<4>–<6>, $\mu^{(r)}_{i_1\cdots i_{M_r} k}$ is an element of a $I_1\times\cdots\times I_{M_r}\times K$ tensor. Let $\mathcal{M}^{(r)}\in \Re^{\psi}_{+}$ denote this tensor, where ψ denotes the dimensionalities $I_1\times\cdots\times I_{M_r}\times K$ in short. Because $\mathcal{M}^{(r)}$ is expensive to compute and operate, we want to reduce computation that involves $\mathcal{M}^{(r)}$. By observing the shared part for updating the core tensor and all facet factors in eq.<4> and <5>, we can use the following strategy to achieve efficient computation: Instead of computing $\mathcal{M}^{(r)}$ explicitly, we compute an intermediate tensor $\mathcal{S}^{(r)}$ of the same dimensionalities as $\mathcal{M}^{(r)}$. $\mathcal{S}^{(r)}$ will save the repeating part of multiplication of $\mathcal{M}^{(r)}$ with $\{\mathbf{U}^{(q)}\}$ and **z** in eq.<4> and <5>. Thus, the above update rules can be rewritten as follows:

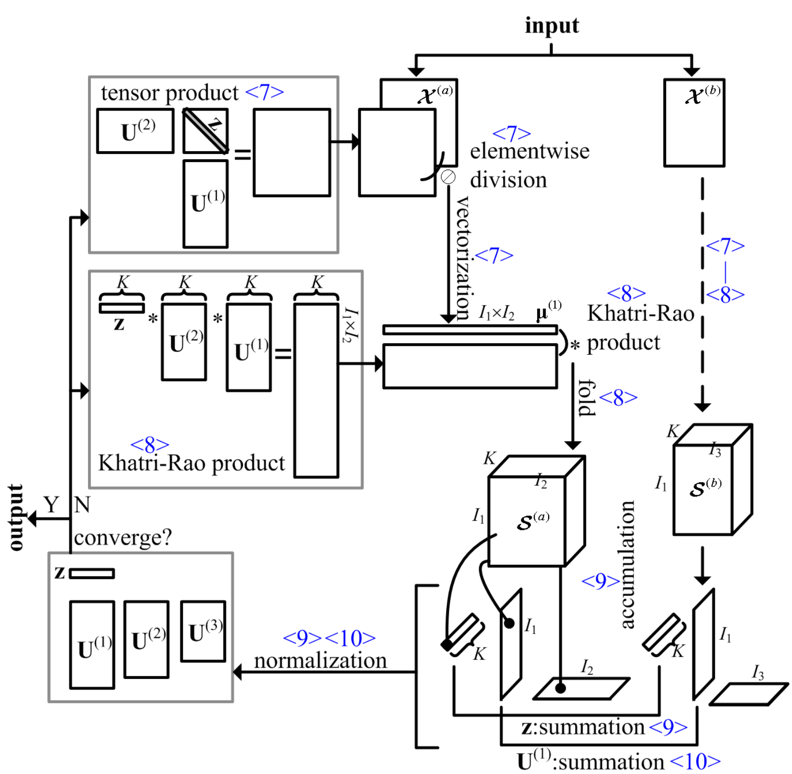

Figure 7: A pictorial view of MF algorithm for solving the problem shown in Figure 6.

First, for each $e^{(r)}$, compute a tensor $\mathcal{S}^{(r)} \in \Re^{\psi}_{+}$ by:

$$\boldsymbol{\mu}^{(r)} \leftarrow vec(\mathbf{X}^{(r)} \% ([\mathbf{z}] \prod\nolimits_{m: v^{(m)} \sim e^{(r)}} \times_m \mathbf{U}^{(m)}))$$ <7>

$$\mathbf{S}^{(r)} = fold(\boldsymbol{\mu}^{(r)} * (\mathbf{z} * \mathbf{U}^{(M_r)} * \cdots * \mathbf{U}^{(1)})^{\mathrm{T}})$$ <8>

where $\oslash$ denotes the element-wise division, and $*$ denotes the Khatri-Rao product.

The second step is to update $\mathbf{z}$ and $\{\mathbf{U}^{(q)}\}$ by:

$$\mathbf{z} \leftarrow \frac{1}{L} \sum_{r \in E} acc(\mathbf{S}^{(r)}, M_r + 1)$$ <9>

$$\mathbf{U}^{(q)} \leftarrow \sum_{l: e^{(l)} \sim v^{(q)}} acc(\mathbf{S}^{(l)}, (q, M_l + 1))$$ <10>

where M_r+1 is the last mode of $\mathcal{S}^{(r)}$. The multiplication of $\mathcal{M}^{(r)}$ and $\mathcal{X}^{(r)}$ in eq.<4> and <5> is now pre-computed in eq.<7> and <8> by utilizing the Khatri-Rao product. To obtain $\mathbf{z}$ and $\{\mathbf{U}^{(q)}\}$, we only need to accumulate $\mathcal{S}^{(r)}$ on the corresponding modes. $\{\mathbf{U}^{(q)}\}$ obtained from eq.<10> will be equivalent to those from eq.<5> after normalization. Eq.<7>–<10> yield exactly the same results as eq.<4>–<6>. The algorithm shares the same form of the expectation-maximization algorithm, where eq.<7> and <8> correspond to the E-step and eq.<9> and <10> correspond to the M-step. Note that the information contained in each data tensor with respect to a hyperedge is aggregated through the E-step and is shared by the core tensor and all facet factors in the M-step, thus the extracted communities will be coherent latent spaces. Table 2 summarizes the whole process to solve an MF problem. We illustrate the process in Figure 7 for solving the problem shown in Figure 6.

Algorithm: MF

Input: metagraph G = (V,E) and data tensors $\{\mathcal{X}^{(r)}\}$ on G
Output: $\mathbf{z}$ and $\{\mathbf{U}^{(q)}\}$

Method:
- Initialize $\mathbf{z}$, $\{\mathbf{U}^{(q)}\}$
- Repeat until *convergence*
  - For each $r \in E$, compute $\mathcal{S}^{(r)}$ by eq.<7> and <8>
  - update $\mathbf{z}$ by eq. <9>
  - For each $q \in V$, update $\mathbf{U}^{(q)}$ by eq. <10>

Table 2: The MF (metagraph factorization) algorithm.

Probability interpretation. The solution core tensor [$\mathbf{z}$] and facet matrices $\{\mathbf{U}^{(q)}\}$ uniquely define the clustering structure. We refer to $\langle \mathbf{z}, \{\mathbf{U}^{(q)}\} \rangle$ as a community model, from which we infer the probabilistic or *soft* membership of entities in each facet. The soft membership assumes that an entity (such as a user or a tag) can belong to multiple communities, with membership weights that sum to one, indicating how likely the entity belongs to those communities. As described in section 4.3, each element $\mathbf{z}_k$ of $\mathbf{z}$ is $p(k)$ (i.e. p_k, the probability of an interaction in community k), which can be considered as popularity of the k-th community, and each (i,k)-element of a facet matrix $\mathbf{U}$ is $p(i|k)$ (i.e. $p_{k\rightarrow i}$, how likely an interaction in the community k involves entity i), which can be considered as the contribution of entity i in the k-th community. We determine the the soft membership of entity i with respect to community k as $p(k|i)$, the conditional probability of a community given the entity i, which is computed by $p(i|k)p(k)/p(i)$, where $p(i)=\sum_{k'} p(i|k')p(k')$ is the marginal probability of an interaction involving entity i.

The proposed community model can be interpreted as a generalization of mixed event space models [Schein et al. 2001] which encode a generative process consisting of multiple types of events. An example of a three-space model based on Figure 4(c) is depicted in Figure 8(a) where $\mathcal{E}^{(1)}$, $\mathcal{E}^{(2)}$ and $\mathcal{E}^{(3)}$ denote the “bookmark”, “joint-project” and “instant-message” event space. A random variable R determines the type of event $r\in R$ that will be generated. In the generative process we pick r and the latent variable k independently. Each entity participating in a tuple is assumed independent of other entities given the knowledge of k. For example, in the “joint-project” $\mathcal{E}^{(2)}$, a user u chooses a latent variable k, which in turn determines the project j to join. Each event space $\mathcal{E}^{(r)}$ corresponds to a hyperedge $e^{(r)}$ in a metagraph. The generalized representation is shown in Figure 8(b). Once r is chosen, an entity in each of the incident facets of the hyperedge $e^{(r)}$ is chosen according to the distribution of the facet. In the example of Figure 8(a), if r=1, a user $u\in v^{(1)}$, a tag $x\in v^{(2)}$, and a document $d\in v^{(3)}$ are chosen based on $p(u|k)$, $p(x|k)$, and $p(d|k)$, respectively. If r=2, a user $u\in v^{(1)}$ and a project $j\in v^{(4)}$ are chosen based on $p(u|k)$ and $p(j|k)$ respectively. If r=3, two users are chosen to send each other messages both using $p(u|k)$. Also if r=1, the probability of choosing a project is zero. We consider that the type r event occurs with probability $\beta^{(r)}$, where $\beta^{(r)}\in[0,1]$ and $\sum_r\beta^{(r)}=1$. In our model, the mixing portion $\beta^{(r)}$ is assumed to be uniform, but it can be extended with non-uniform weights on relations. However, determining an optimal weighting function for different types of applications is an interesting question for future research.

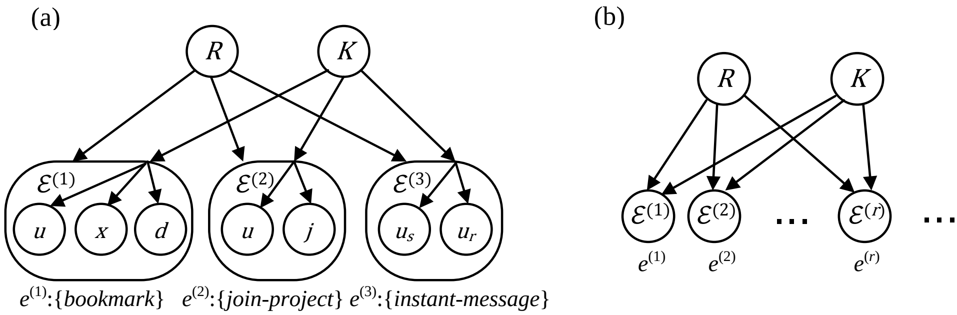

Figure 8: (a) The corresponding generative model for the metagraph shown in Figure 4. (b) The generative model for any given metagraph.

Note that in this example, we use a user facet to model the same set of users participating in the "instant-message" relation and the relation is encoded in a pairwise symmetric matrix $\mathbf{W}=\boldsymbol{\mathcal{X}}^{(2)}= \mathbf{W}^T$. We can obtain the solution[3] for the user facet factor $\mathbf{U}^{(1)}$ through the MF algorithm – the update of each facet factor is based on $\{\boldsymbol{\mathcal{S}}^{(r)}\}$, which in turn is based on the current solution of $\{\mathbf{U}^{(q)}\}$. The solution of $\mathbf{U}^{(1)}$ is interpreted as the distribution of a user participating in the relation without differentiating the message sender and receiver. It is trivial to use two facet factors to model different distributions of message senders and receivers with respect to an asymmetric sender-receiver matrix. The solution facet factors can be obtained in the same way by the MF algorithm. However, determining the community membership of each user in terms of his or her two different roles (as a message sender or as a receiver) may be cumbersome.

Computational Complexity. We now discuss the time complexity for the updates. The most time-consuming step in the algorithm is to compute $\boldsymbol{\mathcal{S}}^{(r)}$ for each hyperedge $e^{(r)}$. As can be seen in eq.<7>, we can take advantage of the sparseness of the data tensor $\boldsymbol{\mathcal{X}}^{(r)}$ and compute only the non-zero elements (total number of tuples) in $\boldsymbol{\mathcal{X}}^{(r)}$. Let n denote the maximum number of non-zero elements of the involved data tensors. This step has time complexity O($nKML$), where K is the number of clusters, M is the maximal number of incident facets of a relation, and L is the total number of input relations. Usually, K, M, L are much smaller than n. If we consider K, M and L are bounded by some constants, the

[3] When there is only a single relation existing between entities in the same facet, such as a user-user relation, the problem turns out to be symmetric NMF. Symmtric NMF has been studied in prior work [Catral et al. 2004; Ding et al. 2005] and similar algorithms can be derived for tensor version, but it is not the focus of this work.

time complexity per iteration is linear in O(n), the number of non-zero elements in all data tensors.

6. TIME EVOLVING EXTENSION

This section presents our solution to the problem of metagraph factorization with time evolving data (MFT). We formulate a new optimization objective for MFT (section 6.1) and provide an algorithm to find a solution to it (section 6.2).

6.1 Optimization Objective

In the MFT problem, the relational data is constantly changing as evolving tensor sequences. We propose an on-line version of MF to handle dynamic data. Since historic information is contained in the community model extracted based on previously observed data, the new community structure to be extracted should be consistent with previous community models and new observations. The idea is similar to the evolutionary clustering discussed by Lin et al. [2008]. To achieve this, we extend the objective in eq.<3> in this section.

A community model for a particular time t is defined uniquely by the factors $\{\mathbf{U}_t^{(q)}\}$ and core tensor $[\mathbf{z}_t]$. (To avoid notation clutter, we omit the time indices for t.) For each time t, the objective is to factorize the observed data into the nonnegative factors $\{\mathbf{U}^{(q)}\}$ and core tensor $[\mathbf{z}]$ which are close to the prior community model, $[\mathbf{z}_{t\text{-}1}]$ and $\{\mathbf{U}_{t\text{-}1}^{(q)}\}$. We introduce a cost l_{prior} to indicate how the new community structure deviates from the previous structure in terms of the KL-divergence. The new objective is defined as follows:

$$
\begin{aligned}
&J_2(G) = (1-\alpha)\sum_{r\in E} D(\mathbf{X}^{(r)} \,\|\, [\mathbf{z}] \prod_{m:v^{(m)}\sim e^{(r)}} \times_m \mathbf{U}^{(m)}) + \alpha l_{prior} \\
&l_{prior} = D(\mathbf{z}_{t-1} \,\|\, \mathbf{z}) + \sum_q D(\mathbf{U}_{t-1}^{(q)} \,\|\, \mathbf{U}^{(q)}) \\
&s.t\ \mathbf{z} \in \Re_+^{1\times K}, \mathbf{U}^{(q)} \in \Re_+^{I_q\times K}\ \ \forall q, \textstyle\sum_i \mathbf{U}_{ik}^{(q)} = 1\ \forall q \forall k
\end{aligned}
\tag{<11>}
$$

where α is a real positive number between 0 and 1 to specify how much the prior community model contributes to the new community structure. l_{prior} is a regularizer used to find similar pairs of core tensors and pairs of facet factors for consecutive times. The new community structure will be a solution incrementally updated based on a prior community model. We shall discuss the semantics of the regularization term in the next section.

6.2 Algorithm

We provide a solution to eq. <11> as follows.

Theorem 2. *The cost defined in eq.<11> is non-increasing under the following update rules and therefore converge to a (local) optimal solution to the MFT problem*:

$$\mathbf{z}_k \leftarrow (1-\alpha)\sum_{r\in E}\sum_{i_1\ldots i_{M_r}} \mathbf{X}^{(r)}_{i_1\ldots i_{M_r}} \mu^{(r)}_{i_1\ldots i_{M_r}k} + \alpha \mathbf{z}_{k;t-1}$$
$$\textit{then normlize such that } \sum_k \mathbf{z}_k = 1, \qquad \text{<12>}$$

$$\mathbf{U}^{(q)}_{i_q k} \leftarrow (1-\alpha) \sum_{l:e^{(l)}\sim v^{(q)}} \sum_{i_1\ldots i_{q-1} i_{q+1}\ldots i_{M_l}} \mathbf{X}^{(l)}_{i_1\ldots i_{M_l}} \mu^{(l)}_{i_1\ldots i_{M_l}k} + \alpha \mathbf{U}^{(q)}_{i_q k;t-1}$$
$$\textit{then normlize such that each column of } \mathbf{U}^{(q)} \textit{ sum to one,} \qquad \text{<13>}$$

where $\mu^{(r)}_{i_1\cdots i_{M_r}k}$ *is defined as in eq.<6>.*

Because of the normalization step, we have dropped the scaling constant for updating **z** and $\mathbf{U}^{(q)}$. The proof of Theorem 2 is similar to the proof of Theorem 1 and is omitted.

According to eq.<11>, α controls how much the current community structure $\langle \mathbf{z}, \{\mathbf{U}^{(q)}\}\rangle$ depends on the historic community structure $\langle \mathbf{z}_{t-1}, \{\mathbf{U}_{t-1}^{(q)}\}\rangle$, i.e. $\langle \mathbf{z}, \{\mathbf{U}^{(q)}\}\rangle$ partly depends on $\langle \mathbf{z}_{t-1}, \{\mathbf{U}_{t-1}^{(q)}\}\rangle$, which in turn partly depends on $\langle \mathbf{z}_{t-2}, \{\mathbf{U}_{t-2}^{(q)}\}\rangle$ and so on, and the dependency is controlled by α. Hence we can consider α as a parameter that controls how much historic information is considered in extracting the current community structure.

In eq.<11>, l_{prior} can be written as:

$$\begin{cases} \log(\mathbf{z}) = \sum_k \mathbf{z}_{k;t-1} \log \mathbf{z}_k + c_{\mathbf{z}} \\ \log(\mathbf{U}^{(q)}) = \sum_{ik} \mathbf{U}^{(q)}_{ik;t-1} \log \mathbf{U}^{(q)}_{ik} + c_q, \forall q \end{cases},$$

where $c_{\mathbf{z}}$ is a value irrelevant to **z**, and c_q is a value irrelevant to $\mathbf{U}^{(q)}$. The parameters in the previous model $\langle \mathbf{z}_{t-1}, \{\mathbf{U}_{t-1}^{(q)}\}\rangle$ act as the Dirichlet prior distribution for the parameters in the current model $\langle \mathbf{z}, \{\mathbf{U}^{(q)}\}\rangle$. So $\langle \mathbf{z}_{t-1}, \{\mathbf{U}_{t-1}^{(q)}\}\rangle$ can be considered as hyperparameters that act as pseudocounts to augment observed community membership. This Dirichlet prior model provides a way to add smoothing to the observed community membership. Such membership smoothing shares the same idea of the “Preserving Cluster Membership” (PCM) criterion as discussed by Chi et al. [2007], where they use a spectral clustering approach. One issue with their work is that the clusters at time t-1 are not explicitly mapped to those clusters at time t and therefore some partition-matching algorithms (e.g., [Lovasz and Plummer 1986]) must be used to obtain the optimal cluster mapping between those at time t-1 and those at time t, where the partition-matching is an NP-hard problem.

Chi et al. [2007] also discuss a "Preserving Cluster Quality" (PCQ) criterion, which aims to find a community structure that explains both historic data and current data well. In our case, we can formulate the PCQ as follows:

$l_{prior} = \sum_{r \in E} D(\mathbf{X}_{t-1}^{(r)} \| [\mathbf{z}] \prod_{m:v^{(m)} \sim e^{(r)}} \times_m \mathbf{U}^{(m)})$, which requires the current model $\langle \mathbf{z}, \{\mathbf{U}^{(q)}\} \rangle$ to also explain the data tensors observed at time t-1. However, large variation in membership is more likely in this formulation due to lack of membership regularization. In our work, we focus on the PCM criterion that defined in eq.<11>.

The update rules can be rewritten as the following operations with $\mathcal{S}^{(r)}$ pre-computed by eq.<7> and <8>:

$$\mathbf{z} \leftarrow (1-\alpha) \sum_{r \in E} acc(\mathbf{S}^{(r)}, M_r + 1) + \alpha \mathbf{z}_{t-1} \qquad <14>$$

$$\mathbf{U}^{(q)} \leftarrow (1-\alpha) \sum_{l:e^{(l)} \sim v^{(q)}} acc(\mathbf{S}^{(l)}, (M_l + 1, q)) + \alpha \mathbf{U}_{t-1}^{(q)} \qquad <15>$$

where M_r+1 is the last mode of $\mathcal{S}^{(r)}$. The whole process of finding solutions to the MFT problem is summarized in Table 3.

For time evolving social data, changes might happen in interactions among entities, or even in interactions among facets (e.g. due to the evolution of system features), which lead to changes in the metagraph. One advantage of our MFT algorithm is that it only requires new observed data defined on any given metagraph, so it is straightforward to incorporate the changes of a metagraph (the algorithm can take different input metagraph G_t).

Algorithm: MFT

Input: metagraph $G = (V,E)$,
the data tensors $\{\mathcal{X}^{(r)}\}$ on G observed at time t,
previous model $\mathbf{z}_{t-1}$, and $\{\mathbf{U}_{t-1}^{(q)}\}$
Output: new model $\mathbf{z}$ and $\{\mathbf{U}^{(q)}\}$

Method:
Initialize $\mathbf{z}, \{\mathbf{U}^{(q)}\}$
Repeat until *convergence*
For each $r \in E$, compute $\mathcal{S}^{(r)}$ by eq.<7> and <8>
update $\mathbf{z}$ by eq. <14>
For each $q \in V$, update $\mathbf{U}^{(q)}$ by eq. <15>

Table 3: The MFT algorithm.

Computational Complexity. For each time t, the time complexity of each iteration in the MFT algorithm is of the same magnitude as that in the MF algorithm, since both algorithms involve computing eq.<7> where the time complexity is O($nKML$). Recall that n is the number of non-zero elements in all data tensors, K is the number of clusters, M is the maximal number of incident facets of a relation, and L is the total number of input relations. Hence with bounded K, M, and L, the time complexity per iteration is linear in O(n).

7. EXPERIMENTS

This section reports our experimental study on two real-world social media datasets collected from an enterprise and the public Digg social network site. We first describe the datasets (section 7.1) and present the extracted communities (section 7.2). We evaluate our technique through prediction tasks (section 7.3). Finally, we evaluate the scalability of our factorization method on synthetic datasets (section 7.4).

7.1 Dataset Description

This section provides a brief description on two real-world data collections used in our experiments. The first, denoted as "ENTERPRISE", is an intranet dataset collected in a corporation, and the second is an internet dataset collected in Digg, a popular online social media website. The Digg dataset and code are available online[4].

7.1.1 *ENTERPRISE Dataset*

We have collected collaboration relationships from the employee profiles and rich-context social/communication media. Data from different sources represent different aspects of the relationships among users (i.e. employees). This collection allows us to build five relations among seven facets. The relations are summarized in Table 4, which correspond to the metagraph shown in Figure 9(a). Based on the availability of the timestamps, we consider four static relations and one dynamic relation.

Relation	**Tensor / incident facets**	**#Tuples**
(R1) bookmark	dynamic (user, tag, URL)	389,617
(R2) join-wiki	static (user, wiki)	1322
(R3) department	static (user, department)	2788
(R4) directory	static (user, directory)	2788
(R5) country	static (user, country)	2788

Table 4: Summary of the relations in ENTERPRISE dataset.

Static relations. We collect data from two sources: First, we collect data from the IBM Lotus® *communities* service which allows users to co-author Wiki-like web pages by registering as community members. A "community-wiki" (or "wiki" for short) is often

[4] http://www.public.asu.edu/~ylin56/kdd09sup.html

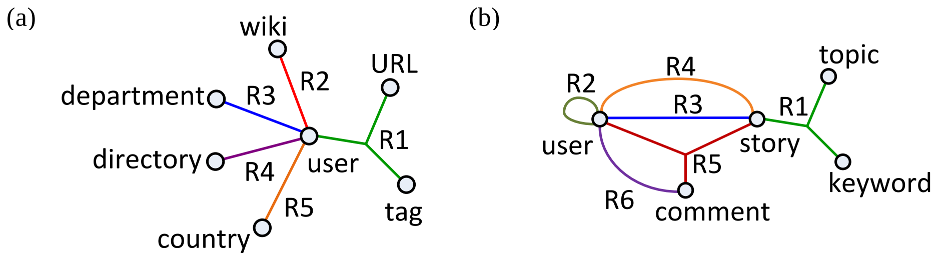

Figure 9: The metagraph representation for (a) ENTERPRISE dataset, and (b) Digg dataset. Nodes represent facets and edges represent relations.

organized based on users' common interests in work or life, e.g. "Lotus Sales Community", "Mac Fans", etc. Second, we extract the formal collaboration relationships from the corporate employee profiles, including their departments, directories, countries, etc. These relations represent the stable context of users.

Dynamic relation. We collect bookmark data from the *dogear* service [Millen et al. 2006], a social bookmarking system hosted on the corporate intranet which has been widely adopted across the enterprise to index and share internal documentation as well as public web resources. The collected dataset contains bookmarks with timestamps ranging from January 2006 to June 2008. From the collection we have extracted users who have more than 3 bookmarks. Based on the timestamps, we construct a sequence of three-way tensors for the facet users, tags and webpage URLs, and each tensor comprises the bookmark data generated in a month.

7.1.2 Digg Dataset

We have collected data from a large set of user actions from Digg. Digg is a popular social news aggregator that allows users to submit, vote (i.e. digg) and comment on news stories. It also allows users to create social networks by designating other users as friends and tracking friends' activities. The dataset used in our experiments include stories, users and their actions (submit, digg, comment and reply) with respect to the stories, as well as the explicit friendship (contact) relation among these users. To analyze users' topical interests, we also retrieve the topics of the stories and extract keywords from the stories' titles.

From this dataset, we select five facets (user, story, comment, keyword, and topic) and build six relations among them. The relations are summarized in Table 5, which correspond to the metagraph shown in Figure 9(b). Except for the contact relation, all relations have timestamps. We assume the contact relation is static and consider the other relations as dynamic. For dynamic relations, we extract tuples with timestamps ranging from August 1 to August 27, 2008. To study the data evolution, we segment the duration into 9 time slots (i.e. every three days), and construct a sequence of data tensors for each dynamic relation. In the following we shall use $t \in [1,9]$ to denote a time slot index. The total number of tuples in each tensor sequence per relation is listed in Table 5.

Relation	**Tensor / incident facets**	**#Tuples**
(R1) content	dynamic (story, keyword, topic)	151,779
(R2) contact	static (user, user)	56,440
(R3) submit	dynamic (user, story)	44,005
(R4) digg	dynamic (user, story)	1,157,529
(R5) comment	dynamic (user, story, comment)	241,800
(R6) reply	dynamic (user, comment)	94,551

Table 5: Summary of the relations in Digg dataset.

7.2 Community Case Studies

We present a case study of the community extraction results for both the ENTERPRISE and Digg datasets. The communities are extracted using the MFT algorithm with α=0.2; the parameter settings will be specified otherwise.

7.2.1 ENTERPRISE Communities

We use our algorithm to extract and track two latent communities from the ENTERPRISE collection, based on relations R1, R2, and R3 (ref. Table 4). The number of communities chosen here is solely for the ease of presentation and does not necessarily correspond to the *true* number of latent communities in the enterprise.

We summarize the extracted communities in Table 6. For each community, we are able to extract entities in the five facets (users, tags, resources, departments and wikis) that are mostly likely to be involved in the communities. The entities are extracted based on their $p(i|k)$ values (as interpreted in section 5.2) aggregated over all timesteps. We omit to show the department facet as the department titles (e.g. "SMP RL Rational Architect") may not be informative to the readers. For privacy reasons we have anonymized the user identities[5]; instead, we show the users' job description rather than their identifiable names or electronic ids/addresses. The users' job description will still allow an investigation of certain characteristics of the extracted communities.

As shown in Table 6, the two communities appear to have distinct profiles. In the first community (or C1 for short), the likely members (i.e. users with high probability to be involved in C1) come with engineering or services job titles, while in the second community (C2), many likely members come with sales or software integration/architect job titles. It is interesting to note that "engineer" and "sales" plays complementary roles in a technical company, and the two extracted communities appear to correspond to the two roles. The resource facet suggests the different information consumption behaviors of the two communities – C1 tends to bookmark the implementation aspect (how-to) of a technique, while C2 tends to bookmark the analytical or business aspects of a technique (e.g. technical news relevant to the corporation). Two communities are also distinguishable in their popular bookmarks of specific search entries, e.g. search entries of "research" and "travel" information in C1 and C2, respectively (ref. Table 6 "resources" facet).

facets	**Community 1** (engineering, services, etc.)	**Community 2** (sales, integrations, etc.)
users (job description)	WebSphere Community Engineer, Collaborative Application Development Research, services enabler / software engineer / abuser of systems, Tape Manufacturing Engineer, Software Engineer, …	Technical Sales - Lotus Software, Service Area Manager and Certified Consulting IT Specialist, Enterprise Integration Architecture, Technical Solution Architect, Business Transformation Consultant, …
tags	projectmanagement, ldap, plaxo, css, lotusphere, xss, analyst, rsa, cmu, pipes, ...	caching, worm, disaster-recovery, rss, technology_adoption, healthcare, journal, cio_agenda, magazine, dou, ...
resources	IBM Research Web 2.0 Unified Search (internal webpage), BASH Programming – Introduction HOW-TO (http://tldp.org/HOWTO/Bash-Prog-Intro-HOWTO.html), Top 10 Ways to Use del.icio.us (http://www.lifehack.org/articles/technology/top-10-ways-to-use-delicious.html), …	*** Evangelist Site – Innovation Through Collaboration (internal webpage), IBM Travel Map (internal webpage), I.B.M to Introduce Workers' Networking Software – New York Times (http://www.nytimes.com/2007/01/22/technology/22ibm.html), …
wikis	Lotus Social Software Community, WPLC Architecture, Mac Fans, BilWithOneL's Web 2.0 Hangout, Notes 8 experts, ...	Lotus Sales Community, Agile Development, Web 2.0 for Business (Web20forbiz), 5live Innovation Day Attendee, Mobile 2.0, ...

Table 6: Summary of two communities extracted from the ENTERPRISE dataset.

Community Evolution. It might not be easy to understand the semantics of the tags, so instead of comparing these tags, we compare the *evolution* of the likely tags within each community. We quantify the evolution of tags within each community based on the cosine similarity between consecutive tag distributions of a community. Specifically, from our algorithm we obtain a sequence of tag factor matrices $\{\mathbf{X}_t\}$ for t=1,2,… Each column, $\mathbf{x}_k$ of $\mathbf{X}_t$ represents the tag distribution of the k-th community at time t. Then we compute the temporal tag dissimilarity by $1\text{-}cos(\mathbf{x}_{k;t\text{-}1}, \mathbf{x}_{k;t})$ for time t to plot the concept evolution curve for the k-th community, as shown in Figure 10 (a). We annotate the peaks

[5] A more in-depth discussion about the results may require users' identifiable information or some cultural understanding about the company; hence we omit the detailed discussion.

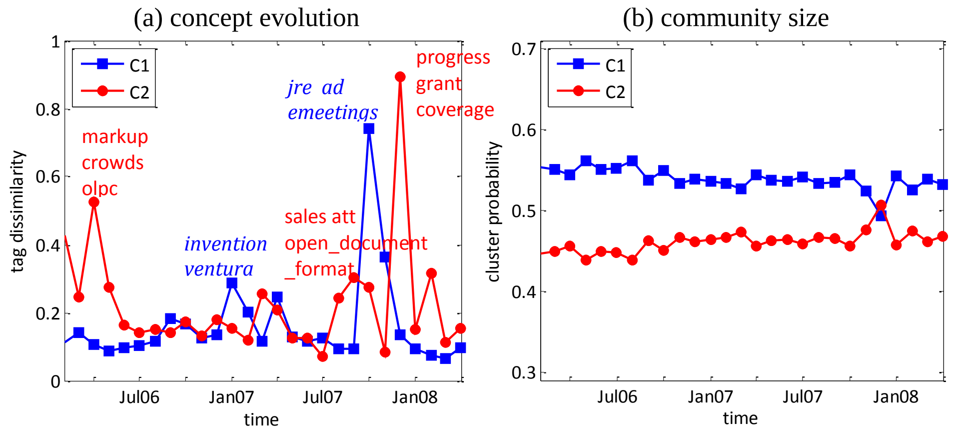

Figure 10: The community evolution characterized based on (a) change in tag distribution (concept evolution) in each community and (b) change in size of communities (cluster probability).

by the most likely tags at the peak times for each cluster. In this figure, we can see distinct patterns of tag evolution within the two extracted communities – both the peak terms and peak times are different. In our interviews with the employees in the company, some pointed out that the peak times at the end of 2007 correspond to some proposal deadlines where new concepts are likely to be introduced. The two peaks seem to suggest there are different deadlines for the sales people and the engineering people. Figure 10 (b) shows the changes in the community sizes over time. We can see the community structure extracted from the ENTERPRISE data is quite stable except at the end of year 2007, which may reflect a certain activity burst in the sales community (C2). This case study suggests that our algorithm is able to generate meaningful mining results from an enterprise data collection.

7.2.2 *Digg Communities*

We now present a detailed qualitative analysis of the communities extracted from the Digg dataset, which demonstrates an advantage of probabilistic interpretation given by our method. We first show all communities extracted for a particular time and then examine the community evolution within these communities.

To illustrate what kinds of stories are "dugg" by what kind of communities, we track the latent communities based on the digging activities that involve relation R1 and R4. Figure 14(a) and (e) shows the corresponding metagraph and the number of tuples in the two relations. In our factorization algorithm, we assume that the number of communities, *K*, is given beforehand. Here we show communities extracted given *K*=2, 4 and 12.

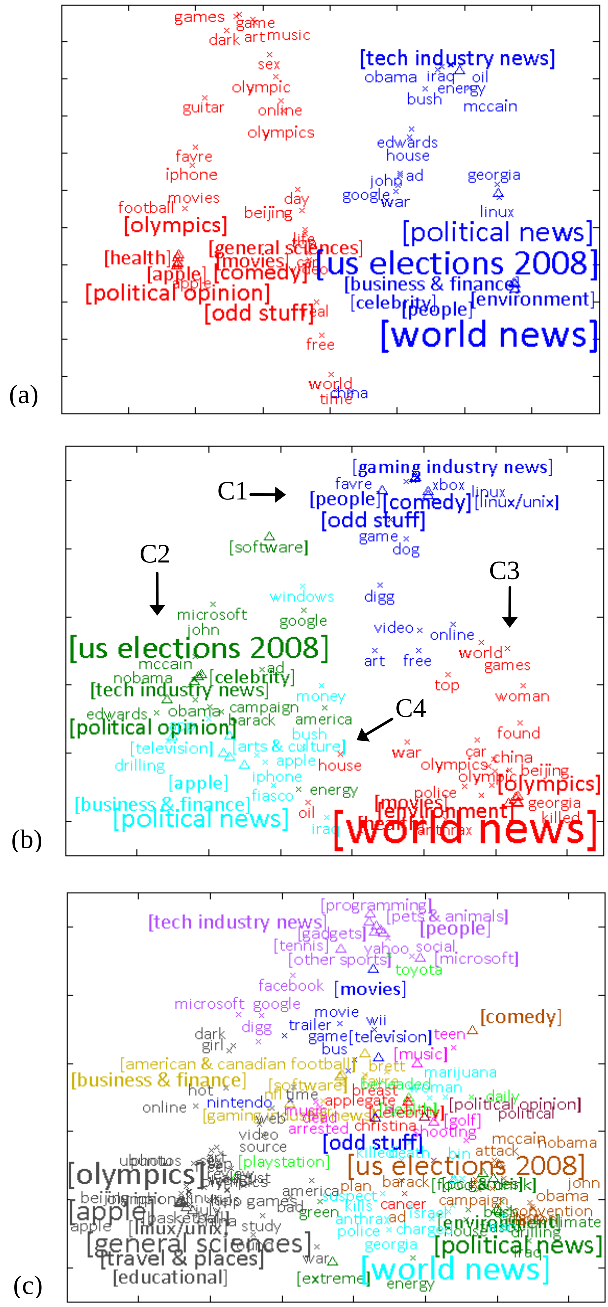

Figure 11: Community extracted based on the user digging activities, for time *t*=3 (August 6-9, 2008) and number of communities (a) *K*=2, (b) *K*=4 and (c) *K*=12. The most likely keyword and topic terms (shown within brackets) in each community are projected based on their soft membership. The size of each term indicates its probability and each term is colored based on its most likely community. The results show coherent topical preference in communities, as the terms with the same colors are located closely.

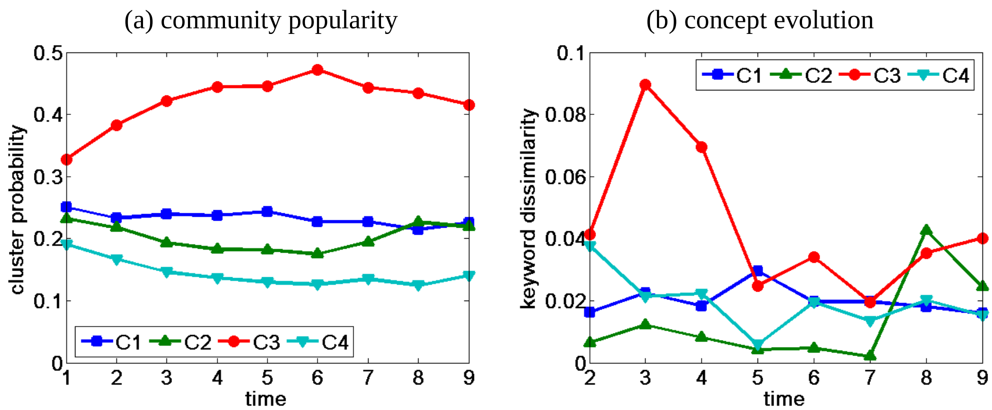

Figure 12: The community evolution characterized based on (a) change in size of communities, and (b) change in keyword distribution in each community.

Based on relation R1 and R4, four facets are involved: user, story, keyword and topic. We present the keyword and topic facets because they are more informative to the readers than other facets. Figure 11 shows the most likely keywords and topics in each community. We present the results of t=3 (August 6-9, 2008). We project those keyword and topic (shown within brackets) terms onto a 2D plane. The location of the i-th keyword or topic term indicates its relative proximity to other terms and is computed based on its soft membership $p(k|i)$. (The position is determined by standard multidimensional scaling (MDS) [Borg and Groenen 2005] with the soft membership as input.) The size of the i-th term indicates how likely the term appears in a story and is determined based on the probability $p(i)$. Each term is colored based on its most likely community, i.e. by choosing k with maximal $p(k|i)$. In the figure we can see the communities based on users' digging activities have coherent topical preference, as the terms with the same colors are located closely. The 2-, 4- and 12-community results show the communities at different scales. The 2-community result distinguishes political interests from the Olympics news (Figure 11(a)). The 4-community shows four topical interests in communities: C1: gaming industry news, C2: US election news, C3: world news, and C4: general political news (Figure 11 (b)). The two major topics ("olympics" and "georgia") in C3 are further split in the 12-community result (Figure 11 (c)).

Community Evolution. We select the 4-community result and examine its evolution. Figure 12(a) shows the probabilities of the four communities over time, and Figure 12(b) shows the keyword dissimilarity across time where the dissimilarity is computed based on the cosine similarity of keyword distribution in each community of consecutive timestamps. We observe two critical times in Figure 12 (b): for communities C2 and C3, the keywords distribution change drastically at *t*=3 (August 6-9) and *t*=8 (August 21-24). To examine the events occurring during these times, we look at the keyword distributions of the two communities. Table 7 lists the top 10 keywords that are most likely to appear in C3 and C2, at *t*=2,3 and *t*=7,8 respectively. At *t*=3, the new popped keywords “olympics” and “georgia” reflect users’ attention to two significant events: the 2008 Summer Olympics began on August 8 and the 2008 Russia-Georgia conflict started on August 7. At *t*=8, the new popped keywords “joe”, “biden”, “vp” correspond to the time when presidential candidate Barack Obama announced that Joe Biden would be his running mate (on August 22). Another critical time is captured by the change in community size. In Figure 12(a), we see the community C3 keeps growing until *t*=6, when the Russia-Georgia conflict ended with a ceasefire agreement signed on August 15 and 16.

C3				C2			
t=2		t=3		t=7		t=8	
olympic	0.007	olympics	0.01	mccain	0.048	mccain	0.048
china	0.006	olympic	0.01	obama	0.039	obama	0.043
beijing	0.006	beijing	0.009	john	0.009	biden	0.018
anthrax	0.005	china	0.007	google	0.006	john	0.01
police	0.004	top	0.006	barack	0.005	joe	0.008
olympics	0.004	georgia	0.005	campaign	0.005	vp	0.007
found	0.004	war	0.004	nobama	0.005	google	0.006
woman	0.004	world	0.004	war	0.004	barack	0.005
top	0.004	games	0.004	georgia	0.004	ad	0.005
world	0.004	killed	0.004	conventio	0.004	news	0.005

Table 7: The keyword distribution of community C2 and C3 during two critical times, *t*=3 and *t*=8.

The characterization of community evolution based on change in the probability of a cluster and on change in the distribution of entities such as keywords (Figure 12 and Table 7) demonstrates the advantage of our soft clustering method – the obtained probability values can be used not only to determine the community membership, but also to infer the importance or representativeness of entities in terms of their contribution to the community structure, as well as to capture the community evolution in various dimensions.. The presented case study suggests that our method is able to generate meaningful mining results from dynamic multi-relational social media data.

7.3 Evaluation via Prediction

We use prediction tasks to demonstrate the utility of our techniques. Due to the lack of ground truth in real world datasets, evaluating the community detection results is challenging. To address this, we design prediction tasks that allow evaluating how well the detected community structures capture the interaction probabilities among entities. The tasks are designed based on different scenarios for the two datasets. We study three aspects of our method through these prediction tasks:

(1) How does our community discovery framework help predict users' *future* interests?

(2) How much historic information do we need (i.e., the impact of α)?

(3) Which relation is relevant to the prediction?

7.3.1 Performance Metrics

We use two metrics adopted in Information Retrieval:

(1) **P@10** (the precision of the top 10 results): For each user we compute P@10 as the portion of the correctly predicted tags (or stories) in the first 10 retrieved tags (or stories) for the user. The overall P@10 for the set of users is computed by taking the mean of P@10 per user, per time slot.

(2) **NDCG** (Normalized Discount Cumulative Gain [Järvelin and Kekäläinen 2000]): One advantage of the measure is its sensitivity to the prediction order. The NDCG is given by:

$$NDCG \propto \sum_i \delta(i)/\log(1+i) \text{ ,} \qquad \text{<16>}$$

where i is the rank of predicted tags (or stories), $\delta(i)$=1 if the prediction of the rank-i tag is correct and 0 otherwise.

In general, the P@N metric assigns an equal weight for each of the top N predicted stories, so the results may be sensitive to the choice of N (N=10 is used in our experiment). The NDCG metric allows different levels of relevance and weighs the prediction according to their ranks in the ranked list. We use top-5 story prediction as an example to illustrate the difference: Assume our model gives an ordered list of stories, $\langle s_1, s_2, s_3, s_4, s_5\rangle$, for a given user. In case s_1 and s_2 are correctly predicted, the NDCG $\propto$ 1/log(1+1)+1/log(1+2)≈1.63; in the other case, if s_3 and s_5 are correctly predicted, the NDCG $\propto$ 1/log(1+3)+1/log(1+5) ≈0.89. The P@5 for both cases is the same (0.4). Hence we expect it is less sensitive to the cut-off of the ranking list used for prediction, and is more effective in differentiating prediction qualities in this task. We use P@N as a complementary metric in order to give an intuitive sense about the prediction quality.

7.3.2 ENTERPRISE Dataset

Prediction setting (tag prediction). For the ENTERPRISE dataset, we design a prediction task to illustrate how our community tracking algorithm can be utilized to predict users' future interests. Specifically, given data D_t at time t, we extract communities to predict users' future use of tags, and compare the prediction with the ground truth in data D_{t+1}. We consider D_t as training data and D_{t+1} as testing data. This is a constrained setting because there might be relevant tagging activities occurring before t and after t+1. In our prediction experiments we only consider two consecutive time slots[6] so to minimize the boundary effects at the beginning and the end of the dataset time span.

The task is designed to understand the meaningfulness of the extracted community structure in the absence of ground truth for the community memberships. The extracted community structures are considered to be meaningful if these structures correlate with external relevant data (in this case, the future individual actions) to a certain extent and hence enable the prediction.

Our prediction method. Our prediction is derived from the community tracking results. We determine if a user u_i will be interested in a tag x_j by the conditional probability:

$$p(x_j \mid u_i) \propto p(x_j, u_i) \approx \sum\nolimits_k \mathbf{z}_k \cdot \mathbf{U}_{ik}^{(1)} \cdot \mathbf{U}_{jk}^{(2)}, \qquad <17>$$

where $\mathbf{z}_k$ is the k-th diagonal element of the core tensor, $\mathbf{U}_{ik}^{(1)}$ is the (i,k)-element of the user factor matrix and $\mathbf{U}_{jk}^{(2)}$ is the (j,k)-element of the tag factor matrix. For each time t, we obtain the community model and derive the conditional probability to predict the user-tag association at time t+1.

We use the MFT algorithm to handle the temporal data and extract communities incrementally. As a comparison, we also report the results of the MF algorithm that extracts communities for each time slice (non-incrementally).

Baseline methods. We compare our method with two baseline methods:

(1) *recurring interests* – predicting future tags (at t+1) as the tags most frequently used by the user at t. This is a simple frequency based heuristic.

(2) *collective interests* (pLSA) – predicting future tags by using a well-known collective filtering method (probabilistic latent semantic analysis [Hofmann 1999] or pLSA) on the user-tag matrix.

Results and Discussion. In our experiment, the time interval is one month. The overall prediction performance is obtained by taking average prediction performance over

[6] The time slot duration is empirically chosen – we choose one month for the ENTERPRISE dataset and three days for the Digg dataset. Given the context of two social media environments, the time duration should be sufficient for the reasonable prediction. For example, in Digg, users are more likely to digg a news story submitted within few days.

10-month data. The best results for each method are reported in Table 8. The results indicate the prediction given by our community discovery framework outperforms the baseline methods by 43-81% (P@10) or 27-72% (NDCG) on the average, which suggest that our method can better capture the dynamics of users' interests.

method \ metric	ENTERPRISE tag prediction	
	P@10	NDCG
recurring	0.108±0.001	0.020±0.000
pLSA	0.155±0.001	0.026±0.000
MF	0.175±0.001	0.028±0.000
MFT	**0.195±0.001**	**0.035**±0.000

Table 8: The average prediction performance for ENTERPRISE tag prediction, evaluated by P@10 and NDCG metrics.

We explain the results in the following. If a user's interests remain similar from time *t*-1 to *t*, the simple frequency based heuristic (recurring interests) would be able to predict the user's interests. On the other hand, if the user's interests changes from *t*-1 to *t*, which cannot be captured by the recurring interests method, we may predict the user's interests based on other users' interests – we use pLSA to capture the collective interests. However, pLSA only considers users' contexts via a single relation (other users' interests on tags) and cannot handle richer aspects of users' contexts. Figure 13 (b) shows that by leveraging cooperate relations (R3 and R4), 19.5% of the users' future interests can be predicted by our method under the experiment setting. Note that even using a single relation (R1) as in pLSA, our method still performs better due to its ability to handle tensor data. It is also possible to use existing tensor analysis to incorporate all possible relations; however, in the next section we show that transforming the data into a single, high dimensional tensor does not necessarily yield good prediction results.

We further study how the prediction performance is affected by different historic information and relational context.

Effect of historic information. We vary the weight of the prior model in MFT by setting $\alpha \in [0,1]$, and report the average P@10 (averaging over time) against α, as shown in Figure 13(a). The results suggest that incorporating prior knowledge does work better than no prior (α=0). Note that the prediction performance drops when overlooking current observation (α>0.6). In practice, a good α value can be obtained through cross-validation.

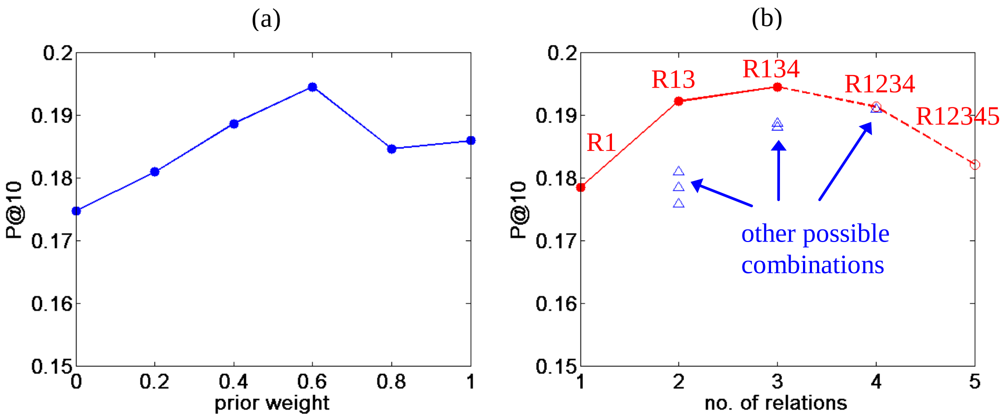

Figure 13: Effect of (a) prior community model (historic information) and (b) different input relations.

Effect of various relational context. We seek to determine the effective relations in terms of future tag prediction. We consider relations as a set of features and employ a forward-feature selection approach to select the best combination of relations. Starting from the first relation (R1, i.e. bookmark relation), we select one relation among the rest that best improves the prediction. The results are shown in Figure 13 (b). The results suggest that combining multiple relations can significantly improve the prediction performance. However, the performance gain does not always come with the increasing number of input relations. In our case, the effective relations for future tag prediction are relation R1, R3 and R4 (i.e. bookmark, department and directory), which interestingly suggest how an individual's use of tags is affected by her or his cooperate structures.

7.3.3 Digg Dataset

Prediction setting (voting and commenting prediction). Based on the Digg scenario, we design prediction tasks to predict users' future interests on digging (i.e. voting) and commenting on Digg stories. There are two tasks: (a) digg prediction – what stories a user will digg, and (b) comment prediction – what stories a user will comment on. Both tasks are evaluated on data from each time slot. We use stories that have digging or commenting events in time slot $t_s \in [2,9]$ as testing sets and the available relational data (ref. Table 5) in time slot t_s-1 as training sets. The prediction results are compared with the actual diggs and comments occurring in slot t_s. This is a constrained setting because there might be more digging or commenting activities occurring after t_s (also see footnote 5). In our prediction experiments we only consider diggs and comments in each single slot t_s as ground truth. The idea behind the design of both prediction tasks is similar to the ENTERPRISE tag prediction discussed in the previous section.

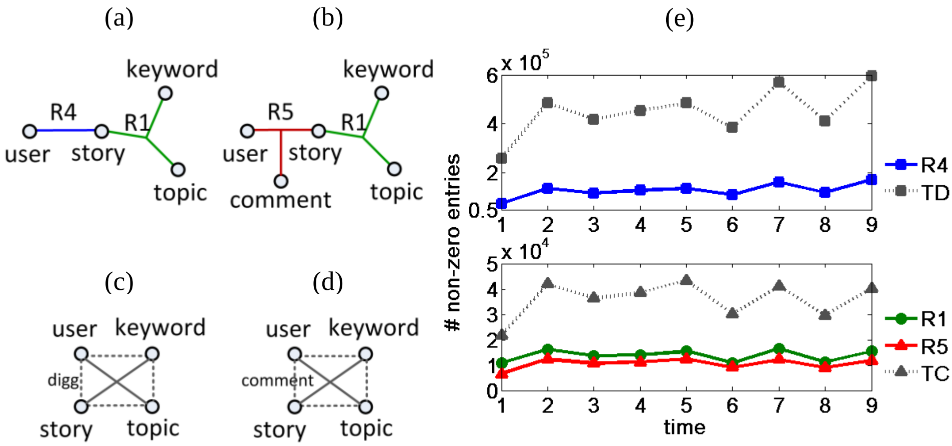

Figure 14: Relations used by different methods for digg and comment prediction: (a) R1 and R4 used in our method for digg prediction; (b) R1 and R5 used in our method for comment prediction; (c) TD tensors used in PARAFAC and MWA for digg prediction; (d) TC tensors used in PARAFAC and MWA for comment prediction; (e) no. of tuples in each relation over time.

Our prediction method. We generate predictions based on the community structure extracted by our method, denoted by MF and MFT. The MF algorithm outputs community structure from relational data of each time slot t_s-1. The MFT algorithm uses the same data as MF, with the aid of a community model extracted for time t_s-2 as an informative prior. Hence MFT gives results incrementally. From an extracted community model we obtain the probability of a community *k*, *p*(*k*), and the probability of a user *u*, a keyword *w* and a topic *j*, given community *k*, i.e. *p*(*u*|*k*), *p*(*w*|*k*) and *p*(*j*|*k*). To predict if a user *u* will digg or comment on a story *r*, we first use a folding-in technique [Popescul et al. 2001] to compute *p*(*r*|*k*), the probability of a story given each community *k*, based on the story topic and keywords. Then a prediction is made based on the condition probability $p(r|u) \propto p(u,r) \approx \Sigma_{k'} p(k)p(u|k')p(r|k')$.

Baseline methods. Three baseline methods are used[7]:

[7] The baseline methods chosen here are different from the baseline methods used in the ENTERPRISE prediction task due to the differences in the available facet information for making predictions. In ENTERPRISE prediction, the task is to predict future tag use based on a data matrix – a single relation involving the user and tag facets. Given the single matrix setting we have chosen pLSA that has been considered effective to handle (document-term) matrix data. In Digg voting prediction, the task is to predict future digg interests based on one or more high-dimensional data tensors – one relation involving the user and story facets and another relation involving the story, keyword and topic facets. (The comment prediction has a similar high-dimensional setting.) To handle this high-

(1) Frequency based heuristics (FREQ) – predicting stories based on the frequency of story topic and keywords at t_s-1.
(2) Standard tensor analysis (PARAFAC) – predicting stories by using the CP/PARAFAC tensor decomposition [Bader and Kolda 2006] for data in slot t_s-1. The stories to be predicted are first projected on the latent spaces, and the prediction is made based on the dot product of the user and story projected vectors.
(3) Multi-way aspect model (MWA) – predicting stories by using the multi-way aspect model [Popescul et al. 2001], a special case of our model (ref. section 4.3).

The ability to handle relational contexts is the key to our comparison. We choose specific relations to illustrate the utility of leveraging a specific context by a metagraph – relation R1 and R4 for digg prediction and R1 and R5 for comment prediction (ref. Figure 14(a) and (b)), and we shall evaluate the effect of other relations later in this section. Since PARAFAC and MWA only deal with a single high dimensional relation, we construct two 4-way tensors per time that contains digg actions and comment actions with respect to stories. The two tensors, denoted by TD and TC are shown in Figure 14 (c) and (d). Figure 14 (e) shows the number of non-zero entries (tuples) of these data tensors over time. The number of tuples in an R5 tensor corresponds to the number of stories per time.

Results and Discussion. The overall prediction performance is obtained by taking the average of prediction performance on data for each time slot (t=1…8 for training and t=2…9 for testing) over different K values. The results (mean and standard deviation) are given in Table 9. There are several observations. First, our method, MF and MFT, significantly outperforms all baseline methods. In digg prediction, our MF method outperforms the best baseline, PARAFAC by 43% (P@10), 45% (NDCG) on the average. In comment prediction, the MF method outperforms the best baseline MWA by 73% (P@10), 89% (NDCG). Second, the MFT performs the best. It slightly outperforms MF in digg prediction and improves MF by 15% in comment prediction. In Figure 15, we show the prediction results over time based on P@10. The results indicate that during the test period, MFT tend to have better prediction performance than MF. The performance gain of MF and MFT may be attributed to the ability to handle relational contexts. As shown in Figure 14 (e), transforming the data into 4-way tensors results in the increase of non-zero entries, while a larger amount of entries does not necessarily help predict users'

---

dimensional setting we have chosen MWA, a high-dimensional extension of pLSA, and

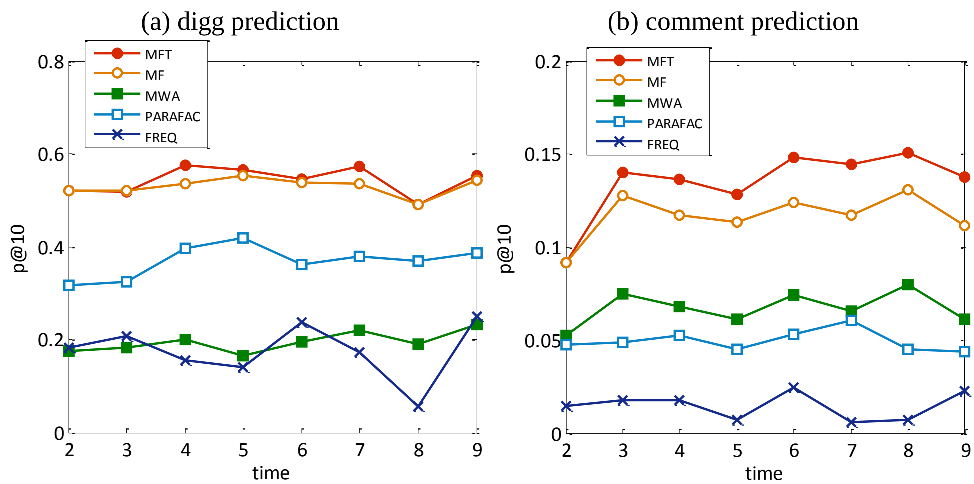

Figure 15: The prediction performance over time for (a) digg and (b) comment prediction, evaluated by P@10.

interests. In the 4-way tensor representation, users' preference for stories may be underrated due to the multiple counts of the facets (keywords or topics) of popular stories. Unlike PARAFAC and MWA, which only deal with a single high dimensional relation, MF and MFT are able to handle multiple relational tensors simultaneously, which balances the information given by user-story and story-keyword-topic relations.

metric / method	digg prediction		comment prediction	
	P@10	NDCG	P@10	NDCG
FREQ	0.175±0.061	0.035±0.016	0.015±0.007	0.004±0.002
PARAFAC	0.369±0.004	0.145±0.002	0.049±0.002	0.018±0.001
MWA	0.195±0.002	0.069±0.001	0.067±0.001	0.020±0.000
MF	0.529±0.008	0.212±0.002	0.117±0.001	0.038±0.000
MFT	**0.543±0.007**	**0.215±0.004**	**0.135±0.001**	**0.043±0.000**

Table 9: The average prediction performance for digg and comment prediction, evaluated by P@10 and NDCG metrics.

Next we examine two characteristics, to show how our prediction can be further improved by (a) incorporating a historic model and (b) leveraging other relations through a metagraph.

Effect of historic information. We vary the weight of the prior model in MFT by setting $\alpha \in [0,1]$ and report the average P@10 over α values (Figure 16 (a)). The results suggest that incorporating historic information as prior knowledge works better than no

PARAFAC which is a variant of higher-order SVD.

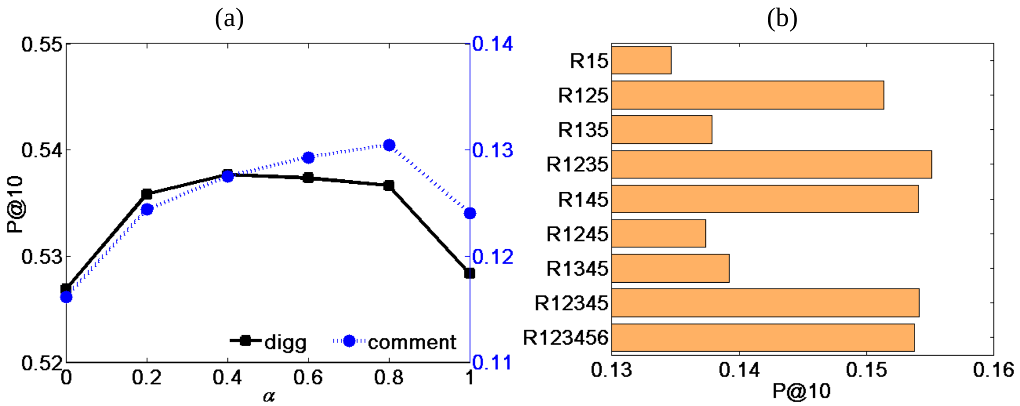

Figure 16: Effect of (a) prior community model (historic information) and (b) different input relations.

prior (α=0, i.e. MF). The effects of historic information are different for the two activities. For digg prediction, the prediction performance drops when α>0.4. For comment prediction, the performance drops increase α≤0.8. This suggests that the future comment activities are more consistent with the historic community structure than the digg activities, which also implies a longer lasting correlation of community structures on users' commenting behavior.

Effect of various relational contexts. For comment prediction, we evaluate the prediction performance over different relational contexts. Figure 16(b) shows the average prediction results. The label R* indicates which relations are used in the training set, e.g. R125 denotes relation R1, R2 and R5. We observe that different combinations of the relations affect the prediction performance. For example, incorporating the contact relation R2 with R1 and R5 significantly helps predict users' comment activities, which implies some correlation between the contact relation and the comment activities (e.g. users are likely to comment on stories on which their friends also give comments). This comparison shows the complexity of choosing the best context in prediction. A metagraph can leverage a mechanism similar to the feature selection scheme for comparing against a family of relational contexts.

Further discussion on prediction results. Although our method significantly outperforms baseline methods in the prediction tasks, predicting the items of users' future interests based on historic data is not easy in practice -- in the ENTERPRISE data (ref. Table 8), the best performance (by our method) indicates 19.5% of the users have at least one item of 10 predicted correctly (according to P@10), but the low NDCG value implies these items may not be predicted in the correct order. In the Digg dataset (ref. Table 9), the best performance (by our method) of the digg prediction indicates 54.3% of the users have at least one item of 10 predicted correctly, while the relatively low NDCG value (0.215) still reflects the inaccurate ranks of the predicted items. The best results of the comment prediction are similar to those in the ENTERPRISE dataset. These results suggest that leveraging multiple relations may be effective in increasing the coverage of predictable items, but may be limited in giving a correct ranks of these items.

7.4 Scalability Evaluation

We use synthetic datasets to illustrate the scalability of our algorithms. We study how the computational time of our algorithm increases with different types of data growth – (a) non-zero elements in a data tensor, (b) number of tensor modes (dimensions), (c) number of relations (tensors) on a given metagraph, as well as (d) the algorithm parameter, i.e. number of clusters. The four experiments are described below (ref. Figure 17).

(1) We randomly generate an M-way tensor of dimensionality $I_1, \ldots, I_M$. We increase the number of non-zero elements in a data tensor by setting M=3, I_1=I_2=I_3=1000. In Figure 17 (a), we show the average running time per iteration of our algorithm against the number of non-zero elements.

(2) With a fixed number of non-zero elements (10^5), we vary the number of tensor modes (i.e. the number of incident facets of a relation). The dimensionalities of all facets are fixed (we set I_q=1000 for all q's). Figure 17 (b) shows the average running time per iteration over the number of tensor modes.

(3) With a fixed number of non-zero elements (10^5), we vary the number of relations in a metagraph by connecting an existing vertex with a new vertex. The order (tensor modes) and dimensionalities of each relation are fixed (M=3, I_1=I_2=I_3=1000). Figure 17 (c) shows the average running time per iteration over the total number of relations.

(4) With a fixed number of non-zero elements (10^5) in the data tensors, we vary the input parameter, the number of clusters, K. Figure 17 (d) shows the running time per iteration over the number of clusters.

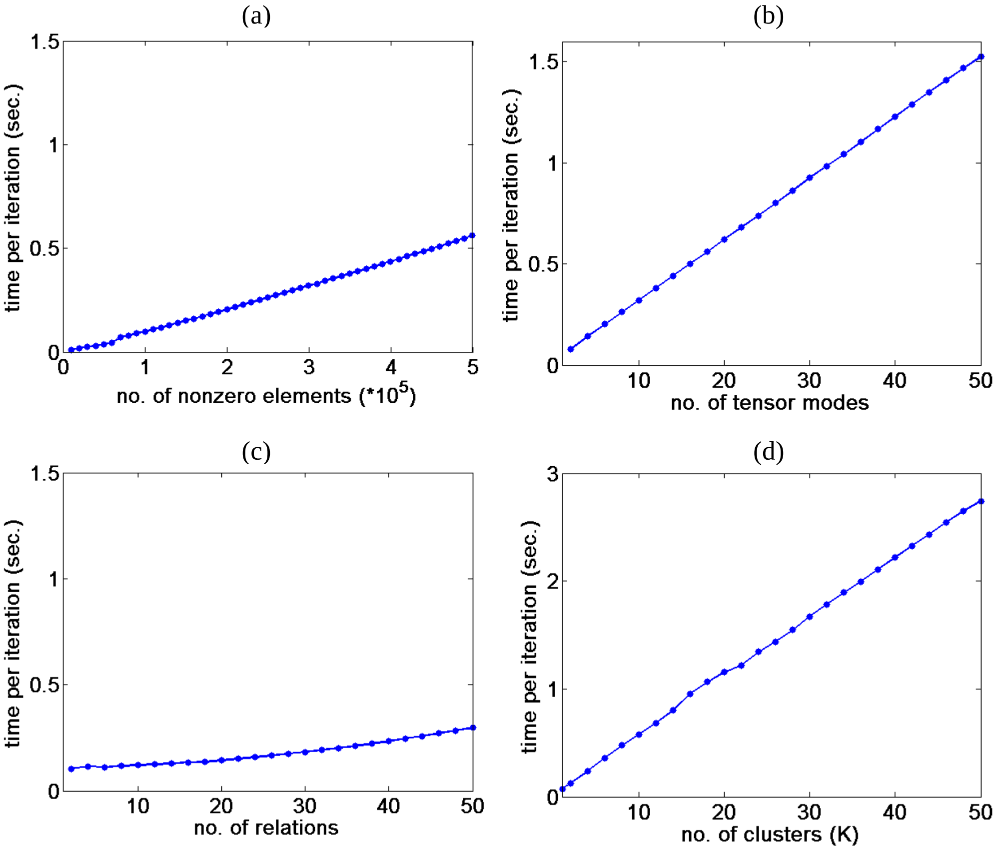

Figure 17: Running time per iteration (sec.) for different types of data growth (let *n* denote the value on the *x*-axis of each plot): (a) number of non-zero elements (one 3-way tensor with *n* non-zero elements), (b) number of tensor modes (one *n*-way tensor), (c) number of relations (*n* 3-way tensors) in a metagraph, and (d) for different algorithm parameter, the number of clusters (*K*).

The results empirically show the running time per iteration scales linearly with the data sizes, the number of tensor modes, the total number of relations, and the number of clusters. Note that the slope for increasing tensor modes is steeper than increasing relations. Empirically, the non-zero elements in a higher mode tensor are usually much more than lower mode tensors (as in Figure 14(e)). Therefore, by leveraging a metagraph, we can efficiently combine multiple low-dimensional relations instead of constructing a high-dimensional tensor.

The experimental results on the synthetic datasets correspond to our analysis in section 5.2 and suggest that our algorithm can efficiently deal with large sparse multi-relational data.

8. OPEN ISSUES

We discuss some open issues in the proposed framework:

(1) Evaluating the results of community detection is challenging due to the lack of ground truth in real world multi-relational datasets. In this work, we have tried to address this issue in two ways: (a) we present case studies in both ENTERPRISE and Digg datasets, and (b) we use prediction tasks to study the "potential links" derived from the detected communities. The case studies rely on human interpretation which is expensive and difficult for comparison, making it an unpromising method to scale to very large datasets. It is important to note that the prediction framework only provides an indirect assessment. Developing a direct and unambiguous quantitative assessment of community detection in multi-relational networks requires annotated benchmarks and would be critical in our future work.

(2) In our factorization method, we use the product of the facet matrices to fit each observed relation. A more natural extension is to use kernel representation for those factors to exploit their non-linear relationship.

(3) We have assumed the number of communities, K, is given beforehand. However, different K values affect the extracted community structure. Selecting the best K is highly dependent on the context of application, e.g. it is usually more useful to select a smaller K for community discovery, but a larger K might improve the prediction accuracy. Moreover, the community might have hierarchical structure such that reasonable mining results occur with many K's. There is work proposing the flat structure assumption in determining optimal number of communities from a single, static network. [Newman and Girvan 2004] proposes a modularity function Q to measure the strength of the discovered clustering structure, which quantifies the deviation between fraction of edges within communities and the expected fraction of such edges. They empirically demonstrate that Q is an effective measure for evaluating the community structure in large networks, where a maximal Q leads to good modular structure in the network. Their modularity function relies on a hard membership assumption – each entity can belong to only a single community. Lin et al. [2008] extend the idea to take soft membership into consideration. Hofman and Wiggins [2008] incorporate the idea of modularity into a Bayesian treatment on the stochastic block model. They develop a solution to this problem that relies on inferring distributions over the model parameters based on a set of hyperparameters which act as pseudocounts that augment observed edge counts and occupation

numbers. Extending their idea to the multi-relational network could be a fruitful research direction.

(4) There are different aspects of community evolution, e.g. (i) change in the community size, (ii) change in the number of communities and (iii) change in the community membership, content or features (what the community is about). To study the evolution within communities, our method has assumed the number of communities does not change across time (i.e. we do not consider the second aspect). Ahmed and Xing [2008] propose a temporal Dirichlet process mixture model (TDPM) for evolutionary clustering which allows the clusters to retain, die out or emerge over time, and the actual parameterization of each cluster can also evolve over time in a Markovian fashion. However, extending their framework to efficiently model various types of co-evolving objects is non-trivial. Moreover, comprehending several evolution aspects in a unified process is a challenging issue.

(5) Our proposed method is useful in quantifying how much the data changes over time in terms of the changes in overall community structure or the changes in specific facet distribution. Lahiri and Berger-Wolf [2008] propose a frequency-based approach to mine periodic or near periodic subgraphs in dynamic networks. You et al. [2009] extract (a) a set of “graph rewriting rules” to describe the changes in graph sequences and (b) a set of “transformation rules” to describe the structures of these changes based on compression-based metrics. These studies have a different focus on mining the *types* of network changes.

9. CONCLUSION

We proposed the MetaFac framework to extract community structures from various social contexts and interactions. There were three key ideas: (1) metagraph, a relational hypergraph for representing multi-relational social data; (2) MF algorithm, an efficient non-negative multi-tensor factorization method for community extraction on a given metagraph; (3) MFT, an on-line factorization method to handle time-varying relations. We conducted extensive experiments on synthetic and two real world datasets, the ENTERPRISE and Digg datasets. A qualitative analysis on communities extracted from both datasets suggested that our method is able to extract meaningful work communities. We evaluated our method by tag prediction (using ENTERPRISE data), as well as digg / comment story prediction (using Digg data). We generated the predictions based on the extracted community models and compared results with baselines. Our method outperformed baselines up to an order of magnitude. We showed our method can be further improved by (a) incorporating a historic model and (b) leveraging other relations through a metagraph. A further examination on the prediction metrics indicates predicting users' future interests based on historic data is non-trivial; nevertheless, the improvement over baseline methods suggest the utility of leveraging metagraphs to handle time-varying social relational contexts.

There are several future directions of this work. (1) As our algorithm does not tie to a specific data schema, it can be easily extended to deal with schema changes. (2) By combining various social relations of data and applying model selection approach, it can be used to identify effective social relations. As part of our future work, we are interested in efficiently determining a relational hypergraph that is effective for a given information task.

10. APPENDIX

Proof of Theorem 1. To prove the convergence of eq.<4>–<6>, we will make use of an *auxiliary function* similar to the NMF algorithms by Lee and Seung [2001] (which are based on the Expectation-Maximization algorithm [Dempster et al. 1977]).

First, we employ the concavity of log function. Because -log($\sum_k \mathbf{a}_k \mathbf{B}_{jk}$) is a convex function, the following equality holds for all j, and $\sum_k \mu_{jk} = 1$.

$$-\log\left(\sum_k \mathbf{a}_k \mathbf{B}_{jk}\right) \le -\left(\sum_k \mu_{jk} \log \frac{\mathbf{a}_k \mathbf{B}_{jk}}{\mu_{jk}}\right), \text{ where } \mu_{jk} = \frac{\mathbf{a}_k \mathbf{B}_{jk}}{\sum_k \mathbf{a}_k \mathbf{B}_{jk}}$$ <18>

Let $\mathbf{a}$=$\mathbf{z}$ and $\mathbf{B} = (\mathbf{U}^{(M_r)} * \ldots * \mathbf{U}^{(1)})^{\mathrm{T}}$ for each r, where $*$ denotes the Khatri-Rao product operation. Then we have:

$$\begin{aligned} J(G) &= \sum_{r\in E} \sum_{i_1 \ldots i_{M_r}} \left(-\mathbf{X}^{(r)}_{i_1 \ldots i_{M_r}} \log \sum_k \mathbf{z}_k \prod_{m:v^{(m)} \sim e^{(r)}} \mathbf{U}^{(m)}_{i_m k} + [\mathbf{z}] \prod_{m:v^{(m)} \sim e^{(r)}} \times_m \mathbf{U}^{(m)} \right) + const \\ &\le \sum_{r\in E} \sum_k \mathbf{z}_k - \sum_{r\in E} \sum_{i_1 \ldots i_{M_r} k} \mathbf{X}^{(r)}_{i_1 \ldots i_{M_r}} \mu^{(r)}_{i_1 \ldots i_{M_r} k} \log \left(\frac{\mathbf{z}_k \prod_{m:v^{(m)} \sim e^{(r)}} \mathbf{U}^{(m)}_{i_m k}}{\mu^{(r)}_{i_1 \ldots i_{M_r} k}} \right) + const \\ &\overset{def}{=} Q(\mathbf{z}, \{\mathbf{U}^{(q)}\}; \{\mu^{(r)}\}) \end{aligned}$$ <19>

where $\mu^{(r)}_{i_1 \ldots i_{M_r} k}$ is defined as in eq.<6>.

Update $\mathbf{z}$: We define $Q_{\mathbf{z}}(\mathbf{z};\mathbf{z}^t)$ as an auxiliary function [Lee and Seung 2001] for $J_{\mathbf{z}}(\mathbf{z})$ with respect to $\mathbf{z}$, where $\{\mathbf{U}^{(q)}\}$ are fixed and $\mathbf{z}^t$ represents the values at the t-th iteration in $\mu^{(r)}_{i_1 \ldots i_{M_r} k}$. The auxiliary function satisfies $Q_{\mathbf{z}}(\mathbf{z};\mathbf{z}^t) \ge J_{\mathbf{z}}(\mathbf{z})$ and $Q_{\mathbf{z}}(\mathbf{z};\mathbf{z}) = J_{\mathbf{z}}(\mathbf{z})$, such that $J_{\mathbf{z}}(\mathbf{z})$ is nonincreasing under the update $\mathbf{z}^{t+1}$=argmin$_{\mathbf{z}}$ $Q_{\mathbf{z}}(\mathbf{z};\mathbf{z}^t)$. Because $J_{\mathbf{z}}(\mathbf{z}^{t+1}) \le Q_{\mathbf{z}}(\mathbf{z}^{t+1};\mathbf{z}^t) \le Q_{\mathbf{z}}(\mathbf{z}^t;\mathbf{z}^t) = J_{\mathbf{z}}(\mathbf{z}^t)$, the sequence of iterative minimization of $Q_{\mathbf{z}}$ leads to a monotonic decrease in the objective function value $J_{\mathbf{z}}$ and ensures the convergence of the update relations.

Hence, with the constraints $\sum_{i_q} \mathbf{U}_{i_q k}^{(q)} = 1 \ \forall q \forall k$, the Lagrangian of $Q_{\mathbf{z}}$ is defined as:

$$L_{\mathbf{Z}} = Q_{\mathbf{Z}} + \sum_q \varepsilon_q (\sum_{i_q} \mathbf{U}_{i_q k}^{(q)} - 1) \qquad \text{<20>}$$

With $\{\mathbf{U}^{(q)}\}$ fixed, and by taking the derivative of L and setting its result to zero, we have:

$$\frac{\partial L_{\mathbf{z}}}{\partial \mathbf{z}_k} = \frac{\partial L}{\partial \mathbf{z}_k} = \sum_r \sum_{i_1 \dots i_{M_r} k} -\mathbf{X}_{i_1 \dots i_{M_r}}^{(r)} \mu_{i_1 \dots i_{M_r} k}^{(r)} \Big/ \mathbf{z}_k + const = 0 , \qquad \text{<21>}$$

where $L = Q + \sum_q \varepsilon_q (\sum_{i_q} \mathbf{U}_{i_q k}^{(q)} - 1)$ is the Lagrangian of Q. <22>

By solving this equation, we obtain the update rule for **z** (eq.<4>).

Update $\mathbf{U}^{(q)}$: Similarly, with **z** and $\{\mathbf{U}^{(m)}\}_{m \neq q}$ fixed, we have:

$$\frac{\partial L}{\partial \mathbf{U}_{i_q k}^{(q)}} = \sum_{l: e^{(l)} \sim v^{(q)}} \sum_{i_1 \dots i_{q-1} i_{q+1} \dots i_{M_l} k} -\mathbf{X}_{i_1 \dots i_{M_l} k}^{(l)} \mu_{i_1 \dots i_{M_l} k}^{(l)} \Big/ \mathbf{U}_{i_q k}^{(q)} + \varepsilon_q + const = 0 \qquad \text{<23>}$$

$$\frac{\partial L}{\partial \varepsilon_q} = \sum_k \mathbf{U}_{i_q k}^{(q)} - 1 = 0 \qquad \text{<24>}$$

By solving the equations, we obtain the update rule for $\mathbf{U}^{(q)}$ (eq.<5>). □

ACKNOWLEDGMENTS

This article is based upon work supported in part by IBM Ph.D. Fellowship and a Kauffman Entrepreneur Scholarship. We are pleased to acknowledge Yun Chi, Shenghuo Zhu, Kai Yu, Belle Tseng and Jun Tatemura, from NEC Labs America, for providing invaluable advices on evolutionary community discovery, which led to further study and resulted in this work.

REFERENCES

ADAMIC, L. AND ADAR, E. 2003. Friends and neighbors on the web. *Social Networks 25* (3): 211-230.

AGGARWAL, C. AND YU, P. 2005. Online analysis of community evolution in data streams. *SDM*, 2005, Society for Industrial Mathematics, 56.

AHMED, A. AND XING, E. 2008. Dynamic non-parametric mixture models and the recurrent chinese restaurant process: With applications to evolutionary clustering. *ICDM*, 2008.

AIROLDI, E., BLEI, D., FIENBERG, S. AND XING, E. 2008. Mixed membership stochastic blockmodels. *The Journal of Machine Learning Research 9*: 1981-2014.

ASUR, S., PARTHASARATHY, S. AND UCAR, D. 2007. An event-based framework for characterizing the evolutionary behavior of interaction graphs. *SIGKDD*, ACM.

BACKSTROM, L., HUTTENLOCHER, D., KLEINBERG, J. AND LAN, X. 2006. Group formation in large social networks: Membership, growth, and evolution. *SIGKDD*, 2006, ACM, 44-54.

BADER, B., HARSHMAN, R. AND KOLDA, T. 2006. *Temporal analysis of social networks using three-way dedicom*. Tech. Rep. SAND2006-2161, Sandia National Labs, Albuquerque, NM and Livermore, CA, Apr. 2006.
BADER, B. AND KOLDA, T. 2006. Algorithm 862: Matlab tensor classes for fast algorithm prototyping. *TOMS 32* (4): 635-653.
BANERJEE, A., BASU, S. AND MERUGU, S. 2007. Multi-way clustering on relation graphs. *SDM*, 2007, Society for Industrial Mathematics.
BARBER, M., FARIA, M., STREIT, L. AND STROGAN, O. 2008. Searching for communities in bipartite networks. American Institute of Physics, 2 Huntington Quadrangle, Suite 1 NO 1, Melville, NY, 11747-4502, USA, 171-182.
BASU, A. AND BLANNING, R. 2007. *Metagraphs and their applications*. Springer-Verlag New York Inc
BEKKERMAN, R., EL-YANIV, R. AND MCCALLUM, A. 2005. Multi-way distributional clustering via pairwise interactions. *ICML 2005*, ACM, 41-48.
BERGE, C. 1976. *Graphs and hypergraphs*. North-Holland Pub. Co.
BERGER-WOLF, T. AND SAIA, J. 2006. A framework for analysis of dynamic social networks. *SIGKDD*, 2006, ACM, 528.
BORG, I. AND GROENEN, P. 2005. *Modern multidimensional scaling: Theory and applications*. Springer
BORGATTI, S. AND CROSS, R. 2003. A relational view of information seeking and learning in social networks. *Management Science*: 432-445.
CARROLL, J. AND CHANG, J. 1970. Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition. *Psychometrika 35* (3): 283-319.
CATRAL, M., HAN, L., NEUMANN, M. AND PLEMMONS, R. 2004. On reduced rank nonnegative matrix factorization for symmetric nonnegative matrices. *Linear Algebra and its applications 393*: 107-126.
CHAKRABARTI, D., KUMAR, R. AND TOMKINS, A. 2006. Evolutionary clustering. *SIGKDD*, ACM, 554-560.
CHEN, P. 1976. The entity-relationship model—toward a unified view of data. *ACM Transactions on Database Systems (TODS) 1* (1): 36.
CHI, Y., SONG, X., ZHOU, D., HINO, K. AND TSENG, B. 2007. Evolutionary spectral clustering by incorporating temporal smoothness. *SIGKDD*, 2007, ACM.
CHI, Y., ZHU, S., GONG, Y. AND ZHANG, Y. 2008. Probabilistic polyadic factorization and its application to personalized recommendation. *CIKM 2008*, 2008, ACM.
CHUNG, F. 1997. *Spectral graph theory*. American Mathematical Society
DEMPSTER, A., LAIRD, N. AND RUBIN, D. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological) 39* (1): 1-38.
DING, C., HE, X. AND SIMON, H. 2005. On the equivalence of nonnegative matrix factorization and spectral clustering. *SDM*, Society for Industrial Mathematics, 606–610.
FALKOWSKI, T., BARTELHEIMER, J. AND SPILIOPOULOU, M. 2006. Mining and visualizing the evolution of subgroups in social networks. *WI 2006.*, 2006, IEEE, 52-58.
FORTUNATO, S. 2010. Community detection in graphs. *Physics Reports 486* (3-5): 75-174.
FRIEDMAN, N., GETOOR, L., KOLLER, D. AND PFEFFER, A. 1999. Learning probabilistic relational models. *IJCAI*, Stockholm, Sweden, 1999, 1300-1309.
GIRVAN, M. AND NEWMAN, M. 2002. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences 99* (12): 7821.
GRANOVETTER, M. 1985. Economic action and social structure: A theory of embeddedness. *American Journal of Sociology 91* (3): 481-510.
GRUJIC, J., MITROVIC, M. AND TADIC, B. 2009. Mixing patterns and communities on bipartite graphs on web-based social interactions. IEEE, 1-8.
HARADA, K., ISHIOKA, T., SUZUKI, I. AND FURUKAWA, M. 2007. A method for solving a bipartite-graph clustering problem with sequence optimization. IEEE, 915-920.
HARSHMAN, R. 1970. Foundations of the parafac procedure: Models and conditions for an" Explanatory" Multi-modal factor analysis. *UCLA working papers in phonetics 16* (1): 1.
HITCHCOCK, F. 1927. The expression of a tensor or a polyadic as a sum of products. *J. Math. Phys 6* (1): 164-189.
HOFMAN, J. AND WIGGINS, C. 2008. Bayesian approach to network modularity. *Physical review letters 100* (25): 258701.
HOFMANN, T. 1999. Probabilistic latent semantic indexing. *SIGIR 1999*, 1999, ACM.
HOLLAND, P. AND LEINHARDT, S. 1981. An exponential family of probability distributions for directed graphs. *Journal of the American Statistical Association 76* (373): 33-50.
JÄRVELIN, K. AND KEKÄLÄINEN, J. 2000. Ir evaluation methods for retrieving highly relevant documents. *SIGIR*, 2000, ACM, 41-48.
KEMP, C., GRIFFITHS, T. AND TENENBAUM, J. 2004. Discovering latent classes in relational data. *Massachusetts Institute of Technology Computer Science and Artificial Intelligence Laboratory*.

KEMP, C., TENENBAUM, J., GRIFFITHS, T., YAMADA, T. AND UEDA, N. 2006. Learning systems of concepts with an infinite relational model. *Proc. of the Natl. Conf. on AI*, AAAI, 381.
KUMAR, R., NOVAK, J. AND TOMKINS, A. 2006. Structure and evolution of online social networks. *SIGKDD*, Philadelphia, PA, USA, 2006, ACM, 611-617.
LAHIRI, M. AND BERGER-WOLF, T. 2008. Mining periodic behavior in dynamic social networks. *ICDM*, 2008, IEEE, 373-382.
LEE, D. AND SEUNG, H. 2001. Algorithms for non-negative matrix factorization. *NIPS*, 2001, 556–562.
LESKOVEC, J., BACKSTROM, L., KUMAR, R. AND TOMKINS, A. 2008. Microscopic evolution of social networks. *SIGKDD*, 2008, ACM.
LIN, Y.-R., CHI, Y., ZHU, S., SUNDARAM, H. AND TSENG, B.L. 2008. Facenet: A framework for analyzing communities and their evolutions in dynamics networks. *WWW 2008*, 2008, ACM.
LIN, Y.-R., CHI, Y., ZHU, S., SUNDARAM, H. AND TSENG, B.L. 2009a. Analyzing communities and their evolutions in dynamics networks. *Transactions on Knowledge Discovery from Data (TKDD) 3* (2).
LIN, Y.-R., SUN, J., CASTRO, P., KONURU, R., SUNDARAM, H. AND KELLIHER, A. 2009b. Metafac: Community discovery via relational hypergraph factorization. *SIGKDD*, 2009, ACM.
LONG, B., ZHANG, Z. AND YU, P. 2007. A probabilistic framework for relational clustering. *SIGKDD*, ACM, 470-479.
LOVASZ, L. AND PLUMMER, M. 1986. *Matching theory*. North Holland
MILLEN, D., FEINBERG, J. AND KERR, B. 2006. Dogear: Social bookmarking in the enterprise. *SIGCHI*, ACM, 111-120.
MONGE, P. AND CONTRACTOR, N. 2001. Emergence of communication networks. *The new handbook of organizational communication: Advances in theory, research, and methods*: 440-502.
NEWMAN, M. AND GIRVAN, M. 2004. Finding and evaluating community structure in networks. *Physical Review E 69* (2): 26113.
NEWMAN, M. AND LEICHT, E. 2007. Mixture models and exploratory analysis in networks. *Proceedings of the National Academy of Sciences 104* (23): 9564.
PALLA, G., BARABASI, A. AND VICSEK, T. 2007. Quantifying social group evolution. *Nature 446* (7136): 664-667.
POPESCUL, A., UNGAR, L.H., PENNOCK, D.M. AND LAWRENCE, S. 2001. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. *UAI 2001*, 437-444.
POWELL, W., KOPUT, K. AND SMITH-DOERR, L. 1996. Interorganizational collaboration and the locus of innovation: Networks of learning in biotechnology. *Administrative science quarterly 41* (1).
RUGG, R. 1984. Part 4: Mathematical, algorithmic and data structure issues: Building a hypergraph-based data structure the examples of census geography and the road system. *Cartographica: The International Journal for Geographic Information and Geovisualization 21* (2): 179-187.
SARKAR, P. AND MOORE, A. 2005. Dynamic social network analysis using latent space models. *SIGKDD Explorations Newsletter 7* (2): 31-40.
SCANDURA, T.A. AND WILLIAMS, E.A. 2000. Research methodology in management: Current practices, trends, and implications for future research. *Academy of Management Journal 43* (6): 1248-1264.
SCHEIN, A., POPESCUL, A., UNGAR, L. AND PENNOCK, D. 2001. Generative models for cold-start recommendations. *Proceedings of the 2001 SIGIR Workshop on Recommender Systems*, 2001.
SEIDMAN, S. 1981. Structures induced by collections of subsets: A hypergraph approach. *Mathematical Social Sciences 1* (4): 381-396.
SHI, J. AND MALIK, J. 2000. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22* (8): 888-905.
SIMSEK, Z. AND VEIGA, J.F. 2000. The electronic survey technique: An integration and assessment. *Organizational research methods 3* (1): 93.
SINGH, A. AND GORDON, G. 2008. Relational learning via collective matrix factorization. *SIGKDD*, ACM.
SPILIOPOULOU, M., NTOUTSI, I., THEODORIDIS, Y. AND SCHULT, R. 2006. Monic: Modeling and monitoring cluster transitions. *SIGKDD*, ACM, 706-711.
STANTON, J.M. AND ROGELBERG, S.G. 2001. Using internet/intranet web pages to collect organizational research data. *Organizational research methods 4* (3): 200.
SUN, J., FALOUTSOS, C., PAPADIMITRIOU, S. AND YU, P. 2007. Graphscope: Parameter-free mining of large time-evolving graphs. *SIGKDD*, 2007, ACM, 687-696.
TANG, L., LIU, H., ZHANG, J. AND NAZERI, Z. 2008. Community evolution in dynamic multi-mode networks. *SIGKDD*, 2008, ACM.
TANTIPATHANANANDH, C., BERGER-WOLF, T. AND KEMPE, D. 2007. A framework for community identification in dynamic social networks. *SIGKDD*, ACM, 717-726.
TASKAR, B., ABBEEL, P. AND KOLLER, D. 2002. Discriminative probabilistic models for relational data. *UAI*, 895–902.
VAVASIS, S. 2007. On the complexity of nonnegative matrix factorization. *Submitted for publication*.
WANG, X., SUN, J., CHEN, Z. AND ZHAI, C. 2006. Latent semantic analysis for multiple-type interrelated data objects. *SIGIR*, 2006, ACM, 236-243.

WASSERMAN, S. AND FAUST, K. 1994. *Social network analysis: Methods and applications*. Cambridge University Press
YANG, T., CHI, Y., ZHU, S., GONG, Y. AND JIN, R. 2009. A bayesian approach toward finding communities and their evolutions in dynamic social networks. *SDM*, 2009, Society for Industrial Mathematics.
YOU, C., HOLDER, L. AND COOK, D. 2009. Learning patterns in the dynamics of biological networks. *SIGKDD*, ACM, 977-986.
YU, K., YU, S. AND TRESP, V. 2005. Soft clustering on graphs. *NIPS 2005*.
ZHU, S., YU, K., CHI, Y. AND GONG, Y. 2007. Combining content and link for classification using matrix factorization. *SIGIR*, 2007, ACM, 487-494.