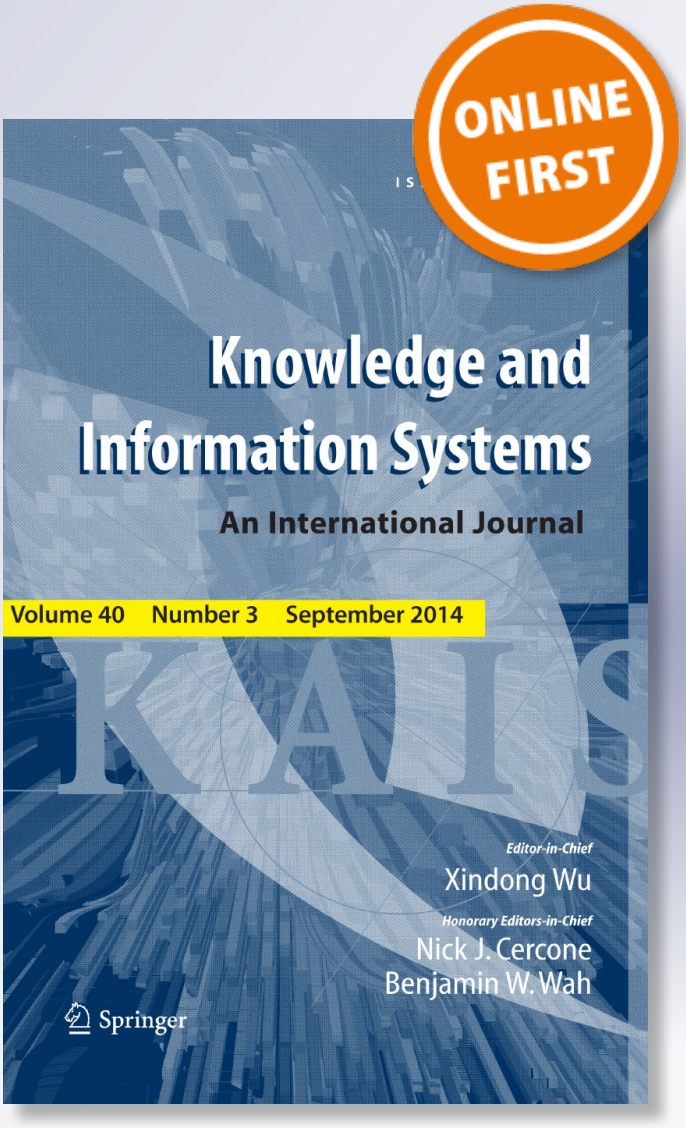
Constructing topical hierarchies in heterogeneous information networks

Chi Wang, Jialu Liu, Nihit Desai, Marina Danilevsky & Jiawei Han

Knowledge and Information Systems
An International Journal

ISSN 0219-1377

Knowl Inf Syst
DOI 10.1007/s10115-014-0777-4



Springer

Knowl Inf Syst
DOI 10.1007/s10115-014-0777-4

REGULAR PAPER

Constructing topical hierarchies in heterogeneous information networks

Chi Wang · Jialu Liu · Nihit Desai · Marina Danilevsky · Jiawei Han

Received: 21 December 2013 / Revised: 28 May 2014 / Accepted: 28 July 2014

Abstract Many digital documentary data collections (e.g., scientific publications, enterprise reports, news articles, and social media) can be modeled as a heterogeneous information network, linking text with multiple types of entities. Constructing high-quality hierarchies that can represent topics at multiple granularities benefits tasks such as search, information browsing, and pattern mining. In this work, we present an algorithm for recursively constructing multi-typed topical hierarchies. Contrary to traditional text-based topic modeling, our approach handles both textual phrases and multiple types of entities by a newly designed clustering and ranking algorithm for heterogeneous network data, as well as mining and ranking topical patterns of different types. Our experiments on datasets from two different domains demonstrate that our algorithm yields high-quality, multi-typed topical hierarchies.

Keywords Topic hierarchy · Information network · Link mining · Text mining · Topic modeling

1 Introduction

Digital documentary data collections, such as scientific publications and social media, often contain additional information beyond plain text. For example, a research paper is linked to

C. Wang (✉) · J. Liu · N. Desai · M. Danilevsky · J. Han
University of Illinois at Urbana-Champaign, Urbana, IL, USA
e-mail: chiwang1@illinois.edu

J. Liu
e-mail: jliu64@illinois.edu

N. Desai
e-mail: nhdesai2@illinois.edu

M. Danilevsky
e-mail: danilev1@illinois.edu

J. Han
e-mail: hanj@illinois.edu

Published online: 26 August 2014

its authors and the venue it was published. A tweet is linked to its twitter and the hashtags or urls mentioned in the tweet. The text linked with multi-typed objects (authors, venues, twitters, hashtags etc.) form *heterogeneous information networks*. In order to facilitate tasks such as efficient search, mining and summarization of these collections, it is valuable to discover and organize the topics present in a dataset into a multi-typed topical hierarchy. Such a construction allows a user to perform more meaningful analysis of the terminology, people, places, and other linked entities, which are organized into topics and subtopics at different levels of granularity.

A variety of existing work is devoted to constructing topical hierarchies from text data. However, few approaches utilize link information from typed entities that may be present in the data. Conversely, existing methods for heterogeneous network analysis and topic modeling have demonstrated that multiple types of linked entities improve the quality of topic discovery (e.g., NetClus [19]), but these methods are not designed for finding hierarchical structures (see Fig. 1a for an example output of NetClus). Therefore, there is no existing method that is able to construct a multi-typed topical hierarchy from a heterogeneous network.

In this study, we develop a method that makes use of both textual information and heterogeneous linked entities to automatically construct multi-typed topical hierarchies. The main contributions of this work are:

- We recursively construct a topical hierarchy where each topic is represented by ranked lists of phrases and entities of different types (e.g., authors, venues). Figure 1c shows an example. We go beyond the topical hierarchies that are constructed by analyzing textual information alone (e.g., Fig. 1b), and enrich the topic representation with ranked lists of entities, which provide additional informative context for each topic in the hierarchy.
- We propose a mutually enhancing clustering and ranking method for recursively generating subtopics from each topic in the hierarchy. Our approach retains the benefits of NetClus, a recently developed technique for analyzing heterogeneous networks, but is far more robust and well-suited to the task of topical hierarchy construction. Our unified general model is not confined to a particular network schema and incorporates an inference algorithm that is guaranteed to converge.
- We develop an extension to our method that is able to automatically determine the importance of different types of entity links at different topic granularity. For example, the links to venues are important for finding computer science areas in a rough granularity, but less important for fine granularity topics. Our learning method is theoretically justified and guaranteed to converge.

2 Related work

2.1 Topical hierarchy construction

Topical hierarchies, concept hierarchies, ontologies, etc., provide a hierarchical organization of data at different levels of granularity and have many important applications, such as in web search and browsing tasks [7]. Although there has been a substantial amount of research on ontology learning from text, it remains a challenging problem (see [22] for a recent survey). The learning techniques can be broadly categorized as statistics-based or linguistic-based. Many studies are devoted to mining subsumption ('is-a') relationships [11], either by using

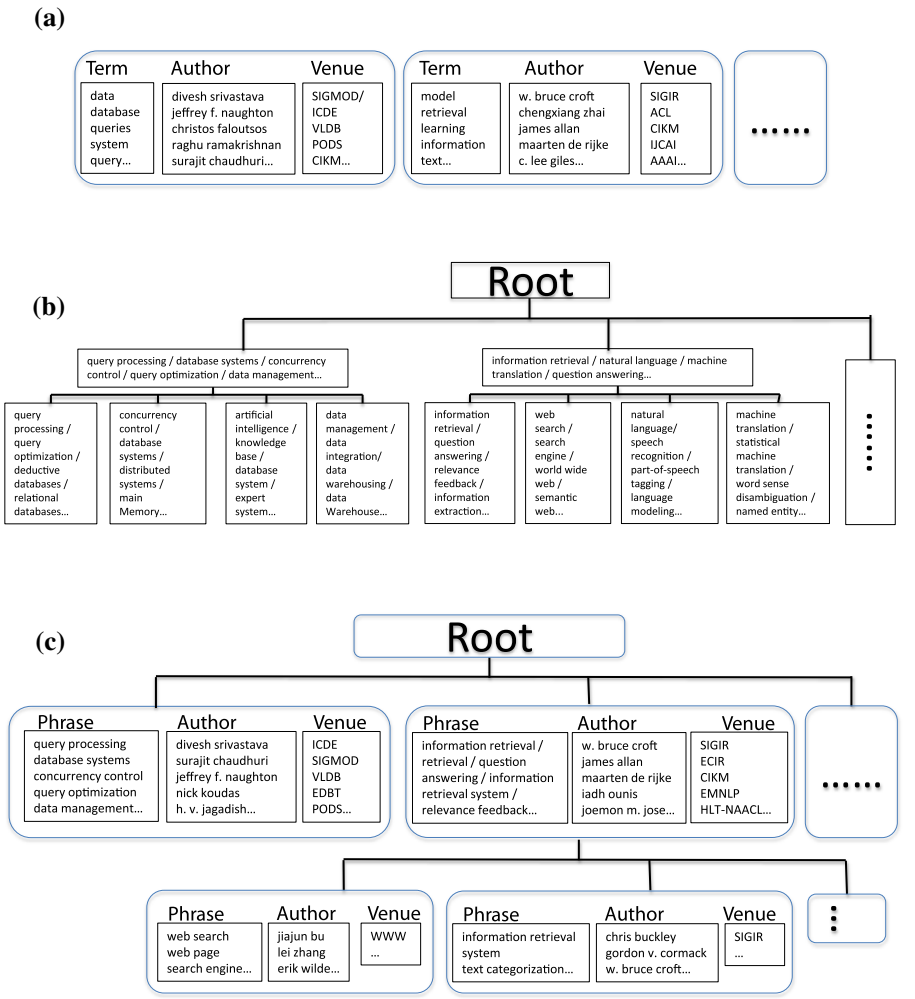


Fig. 1 Sample output from three methods run on a computer science publication collection with term, author, and venue information. **a** NetClus [19]—clusters of multi-typed entities. Each *rounded rectangle* represents one cluster, containing a ranked list of unigrams and two ranked lists of entities. **b** CATHY [21]—topical hierarchy of text only. Each node in the hierarchy contains a ranked list of phrases. **c** CATHYHIN—topical hierarchy of multi-typed entities. Each node has a ranked list of phrases and two ranked entity pattern lists

lexico-syntactic patterns(e.g., 'x is a y') [14,17] or statistics-based approaches [6,23]. [4] and [13] generate taxonomies of given keyword phrases by supplementing hierarchical clustering techniques with knowledge bases and search engine results.

With respect to input and output, our definition of the construction of topical hierarchy largely follows our previous work [21]. We proposed CATHY, a statistics-based technique that constructs a topical hierarchy without resorting to external knowledge resources such as WordNet or Wikipedia. However, CATHY hierarchy is constructed using only text information, while our CATHYHIN approach works with a heterogeneous network and discovers multi-typed topical entities.

2.2 Mining topics in heterogeneous networks

Basic topic modeling techniques such as PLSA (probabilistic latent semantic analysis) [9] and LDA (latent dirichlet allocation) [1] take documents as input and output word distributions for each topic. Recently, researchers have studied how to mine topics when documents have additional links to multiple typed entities [3,5,10,18–20]. These approaches make use of multiple typed links in different ways. iTopicModel [18] and TMBP–Regu [5] use links to regularize the topic distribution so that linked documents or entities have similar topic distributions. [3] and [10] extend LDA to use entities as additional sources of topic choices for each document. [20] argue that this kind of extension has a problem of 'competition for words' among different sources when the text is sparse. They propose to aggregate documents linked to a common entity as a pseudodocument and regularize the topic distributions inferred from different aggregation views to reach a consensus.

Nearly all of these studies still model topics as distributions over words, though they use linked entity information to help with topic inference in various ways. NetClus [19] takes a different approach by simultaneously clustering and ranking terms as well as linked entities in a heterogeneous network. It is therefore the only aforementioned approach, which may be used to recursively construct heterogeneous topical hierarchies (with some slight modification). We therefore examine adapting NetClus to this task and describe the limitations of this construction, which are overcome by our method.

2.3 NetClus

As illustrated in Fig. 2, the input to the NetClus algorithm is a heterogeneous network of star schema. The example network has one central object type—the star object—and four types of attribute objects (where the type of an object is denoted by its shape and color family). Only links between a star object and an attribute object are allowed. For example, a collection of

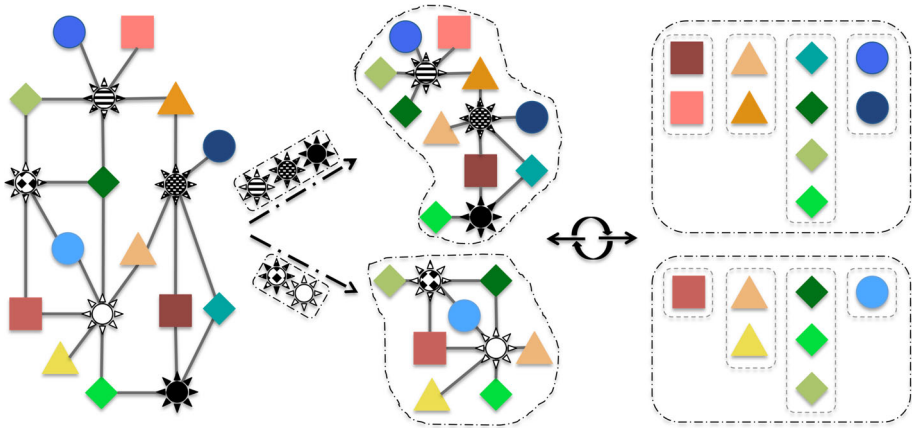


Fig. 2 An illustration of the NetClus framework. *Left* NetClus analyses a star schema network, where every link is between a central object and an attribute object of some type (*central objects* are denoted by *stars*; attribute objects of the same type are represented by the *same shape and color family*, with individual objects differentiated by hue). (*Middle*) The *star objects* are partitioned into clusters so that each *star* appears in exactly one cluster. (*Right*) Attribute objects (which may appear in multiple clusters) are ranked within each cluster, grouped by type. NetClus iterates over these clustering (*middle*) and ranking (*right*) steps, as denoted by the two-way *circular arrow symbol*

papers may be transformed into a star schema where each paper is a star object, and attributes such as authors, venues, and terms are attribute objects.

NetClus performs hard clustering on the star objects, and the induced network clusters consist of star objects and their linked attribute objects. Thus, an attribute object may belong to multiple clusters, but each star object is assigned to precisely one cluster. Next, the attribute objects within each subnetwork cluster are ranked via a PageRank-like algorithm, which is based on the structure of the cluster. A generative model then uses the ranking information to infer a cluster distribution for each star object. The cluster memberships of the star objects are then adjusted using a k-means algorithm, and the ranks of attribute objects are re-calculated. Thus, the NetClus algorithm iterates over clustering the star objects based on their inferred membership distribution (as calculated by a generative model based on the existing ranking information), and re-ranking the attribute objects within each newly defined network cluster. The heterogeneous nature of the attribute objects is respected during the ranking step, as only objects of the same type are ranked together, as shown in Fig. 2.

The iterative clustering and ranking steps of NetClus thus mutually enhance each other. The clustering step provides a context for the ranking calculations, since the ranks of the attribute objects should vary among different clusters (e.g., different areas of computer science). The ranking step in turn improves the quality of found clusters, since highly ranked objects should serve as stronger indicators of cluster membership for their linked star objects.

NetClus can be naturally extended for topical hierarchy construction: after each network is clustered, each of the induced subnetworks are then used as new input and may thus be recursively clustered and ranked. However, several properties of NetClus render it undesirable for the task of topical hierarchy construction:

1. Topics are represented by ranked lists of terms, and other individual attribute objects. For topics of fine granularity in the hierarchy, this representation may be hard to interpret because single terms and entities may be ambiguous.
2. NetClus assumes a star schema, which hinders its application to more general information networks.
3. NetClus hard clusters star objects, which are usually documents. However, a document is often related to a mixture of topics, especially in the lower levels of a hierarchy. The forced hard clustering can thus result in lost information, as relevant documents fail to appear in relevant subtopics, further decreasing the hierarchy's quality.
4. The iterative algorithm used by NetClus is not guaranteed to converge. The deeper into the hierarchy, the more severe this problem becomes because the output of one level will be input of the next level of the constructed hierarchy.

3 CATHYHIN framework

This section describes our framework CATHYHIN (shown in Fig. 3), which incorporates the two positive characteristics of NetClus: the utilizing of heterogeneous link types, and the mutually enhancing clustering and ranking steps, while overcoming the disadvantages discussed in Sect. 2.3.

Definition 1 (*Heterogeneous Topical Hierarchy*) A heterogeneous topical hierarchy is defined as a tree $\mathcal{T}$ in which each node is a topic. The root topic is denoted as o. Every non-root topic t with parent topic $Par(t)$ is represented by m ranked lists of patterns $L_1, \ldots, L_m$ where $L_x = \{P_i^{x,t}\}$ is the sequence of patterns for type x in topic t. The subtopics of every

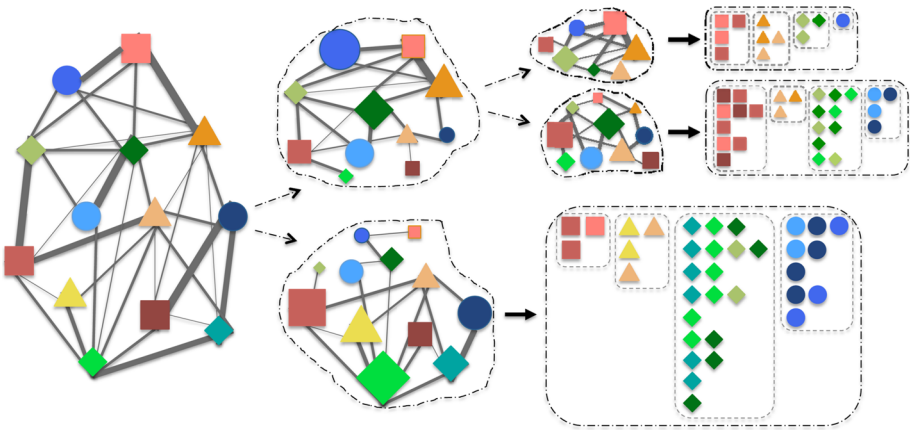


Fig. 3 An illustration of the CATHYHIN framework. (*L*) *Step 1:* CATHYHIN analyses a node-typed and edge-weighted network, with no central star objects. (*M*) *Step 2:* A unified generative model is used to partition the edge weights into clusters and rank single nodes in each cluster (here, node rank within each node type is represented by variations in node size). (*R bottom*) *Step 3:* Patterns of nodes are ranked within each cluster, grouped by type. (*R top*) *Step 4:* Each cluster is also an edge-weighted network and is therefore recursively analyzed. The final output is a hierarchy, where the patterns of nodes of each cluster have a ranking within that cluster, grouped by type

non-leaf topic t in the tree are its children $C^t = \{z \in \mathcal{T}, Par(z) = t\}$. A pattern can appear in multiple topics, though it will have a different ranking score in each topic.

This definition of the heterogeneous topical hierarchy addresses the first aforementioned criticism of NetClus by representing each topic as multiple lists of ranked patterns, where each list contains *patterns* of objects, rather than individual objects (e.g., phrases rather than unigrams). For instance, the topics in Fig. 1c each contain three lists of patterns.

Our approach does not restrict the network schema and does not perform hard clustering for any objects. We discover topics by hierarchically soft clustering the links, so that any node may be assigned to multiple topics and subtopics. This removes the restrictions outlined in criticisms 2 and 3 of NetClus. We only require a collection of some kind of information chunks, such as documents, so that each chunk contains multiple objects and we can mine frequent patterns from these chunks.

Formally, every topic node t in the topical hierarchy is associated with an edge-weighted network $G^t = (\{V_x^t\}, \{E_{x,y}^t\})$, where V_x^t is the set of type-x nodes in topic t, and $E_{x,y}^t$ is the set of link weights between type x and type y nodes (x and y may be identical) in topic t. $e_{i,j}^{x,y,t} \in E_{x,y}^t$ represents the weight of the link between node v_i^x of type x and node v_j^y of type y. The network G^t can contain m node types and l link types with the constraint $l \leq m^2$.

For every non-root node $t \neq o$, we construct a subnetwork G^t by clustering the network $G^{Par(t)}$ of its parent $Par(t)$. G^t inherits the nodes from $G^{Par(t)}$, but contains only the fraction of the original link weights that belongs to the particular subtopic t. Figure 3 visualizes the weight of each link in each network and subnetwork by line thickness (disconnected nodes and links with weight 0 are omitted).

If the original network naturally has a star schema, but the star type is not included in the final topic representation (e.g., the document), we can construct a 'collapsed' network by connecting every pair of attribute objects, which are linked to the same star object. In the

derived network, the link weight $e_{i,j}^{x,y,t}$ between two nodes v_i^x and v_j^y is therefore equal to the number of common neighbors they share in the original star schema network.

Example 1 We can construct a collapsed network from the research publications, with $m = 3$ types of nodes: term, author and venue; and $l = 5$ types of links: term–term, term–author, term–venue, author–author, author–venue. The link weight between every two nodes is equal to the number of papers where the two objects co-occur.

Our framework employs a unified generative model for recursive network clustering and subtopic discovery. The model seamlessly integrates mutually enhanced ranking and clustering while guaranteeing convergence for the inference algorithm, thus addressing the final critique of NetClus.

Our framework generates a heterogeneous topical hierarchy in a top-down, recursive way:

Step 1. Construct the edge-weighted network G^o. Set $t = o$.
Step 2. For a topic t, cluster the network G^t into subtopic subnetworks G^z, $z \in C^t$ using a generative model.
Step 3. For each subtopic $z \in C^t$, extract candidate topical patterns within each topic, and rank the patterns using a unified ranking function. Patterns of different lengths are directly compared, yielding an integrated ranking.
Step 4. Recursively apply Steps 2–3 to each subtopic $z \in C^t$ to construct the hierarchy in a top-down fashion.

We describe steps 2 and 3 in Sects. 3.1 and 3.2. Then in Sect. 3.3, we discuss how to customize the shape of the hierarchy, including the number of children of each topic, the depth of the hierarchy, and the balance of the size of subtopics.

3.1 Topic discovery in heterogeneous information networks

Given a topic t and the associated network G^t, we discover subtopics by performing clustering and ranking with the network. In Sect. 3.1.1, we describe our unified generative model and present an inference algorithm with a convergence guarantee. In Sect. 3.1.2, we further extend our approach to allow different link types to play different degrees of importance in the model (allowing the model to, for example, decide to rely more on term co-occurrence information than on co-author links). Table 1 summarizes the notations.

3.1.1 The generative model

We first introduce the basic generative model, which considers all link types to be equally important. For a given topic t, we assume C^t contains k child topics, denoted by $z = 1 \ldots k$. The value of k can be either specified by users or chosen using a model selection criterion. We discuss the choice of k in Sect. 3.3.

We assume every link has a direction. For undirected networks, we convert them to directed networks by duplicating each link between v_i^x and v_j^y in both directions $v_i^x \rightarrow v_j^y$ and $v_j^y \rightarrow v_i^x$. So our model can be applied to both undirected and directed networks.

Similar to NetClus, we assume each node type x has a multinomial distribution $\phi^{x,z}$ in each subtopic $z \in C^t$, such that $\phi_i^{x,z}$ is the importance of node v_i^x in topic z, subject to $\sum_i \phi_i^{x,z} = 1$. Each node type x also has a multinomial distribution $\phi^{x,0}$ for the background topic, as well as an overall distribution ϕ^x, where ϕ_i^x is proportional to the indegree of

Table 1 Notations used in our model

Symbol	Description
G^t	The HIN associated with topic t
V_x^t	The set of nodes of type x in topic t
$E_{x,y}^t$	The set of nonzero link weights of type (x, y) in topic t
$Par(t)$	The parent topic of topic t
C^t	The set of child topics of topic t
z	Child topic index of topic t
m	The number of node types
l	The number of link types
$n_{x,y}$	The total number of type-x and type-y node pairs that have links
v_i^x	The ith node of type x
$e_{i,j}^{x,y,z}$	The link weight between v_i^x and v_j^y in topic z
M_t	The sum of link weight in topic t: $\sum_{i,j,x,y} e_{i,j}^{x,y,t}$
$M_t^{x,y}$	The sum of type-(x, y) link weight in topic t: $\sum_{i,j} e_{i,j}^{x,y,t}$
$\phi^{x,z}$	The distribution over type-x nodes in topic z
ϕ^x	The overall distribution over type-x nodes
ρ	The distribution over subtopics
θ	The distribution over link types
$\alpha_{x,y}$	The importance of link type (x, y)

node v_i^x. In contrast to NetClus, we softly partition the link weights in G^t into subtopics. We model the generation of links so that we can simultaneously infer the partition of link weights (clustering) and the node distribution (ranking) for each topic.

Example 2 In a computer science publication network, each of the three node types term, author, and venue has a ranking distribution in each topic in the hierarchy. The distributions for a hypothetical topic about database may be: (i) term—{database: 0.01, system: 0.005, query: 0.004, …}; (ii) author—{Sujarit Chaudhuri: 0.03, Jeffery F. Naughton: 0.02, …} and (iii) venue—{SIGMOD: 0.2, VLDB: 0.25, …}

To derive our model, we first assume the links between any two nodes can be decomposed into one or multiple unit-weight links (e.g., a link with weight 2 can be seen as a summation of two unit-weight links). Later we will discuss the case where the link weight is not an integer. Each unit-weight link has a topic label, which is either a subtopic $z \in C^t$, or a dummy label 0, implying the link is generated by a background topic and should not be attributed to any topic in C^t.

The generative process for a link with unit weight is as follows:

1. Generate the topic z according to a multinomial distribution ρ.
2. Generate the link type (x, y) according to a multinomial distribution θ.
3. If $z \in C^t$,

   (a) Generate the first end node u_1 from the type-x ranking distribution $\phi^{x,z}$.
   (b) Generate the second end node u_2 from the type-y ranking distribution $\phi^{y,z}$.

   Else $(z = 0)$

(a) Generate the first end node u_1 from the type-x ranking distribution $\phi^{x,0}$.
(b) Generate the second end node u_2 from the type-y ranking distribution ϕ^{y}.

Example 3 A link between two terms *query* and *processing* in a topic $z = Database$ may be generated in the following order: (i) generate the topic $z = Database$ with probability $\rho_z = 0.2$; (ii) generate the link type $(x, y) = (term, term)$ according to $\theta_{x,y} = 0.15$; (iii) generate the first end node $u_1 = query$ from the term distribution $\phi_{u_1}^{x,z} = 0.004$; and (iv) generate the second end node $u_2 = processing$ from the term distribution $\phi_{u_2}^{y,z} = 0.001$.

This process is repeated for M^t times, to generate all the unit-weight links. Note that when generating a background topic link, the two nodes i and j are not symmetric. The first end node is a background node and can have a background topic link with any other nodes based simply on node indegree, irrespective of any topic. Highly ranked nodes in the background topic tend to have a link distribution over all nodes that is similar to their overall indegree distribution. This part can be altered if the network follows a different assumption about the background nodes (e.g., background nodes are linked with other nodes according to their outdegree). See Fig. 4a for a graphical representation of the model.

With these generative assumptions for each unit-weight link, we can derive the distribution of link weight for any two nodes (v_i^x, v_j^y). First, we notice that the total number of topic-z unit-weight links is expected to be $M^t \rho_z$. It implies that we are expected to repeat the generation of topic-z unit-weight links for $M^t \rho_z$ times. Second, we investigate how many topic-z unit-weight links will be generated between two certain nodes v_i^x and v_j^y, among all the topic-z links. Consider the event 'the two end nodes are v_i^x and v_j^y' for each generated topic-z link. It is a Bernoulli trial with success probability $\theta_{x,y}\phi_i^{x,z}\phi_j^{y,z}$ for $z \in C^t$. When $M^t \rho_z$ is large, the total number of successes $e_{i,j}^{x,y,z}$ asymptotically follows a Poisson distribution $Pois\left(M^t \rho_z \theta_{x,y}\phi_i^{x,z}\phi_j^{y,z}\right)$. Similarly, the total number of background topic links $e_{i,j}^{x,y,0}$ asymptotically follows a Poisson distribution $Pois\left(M^t \rho_0 \theta_{x,y}\phi_i^{x,0}\phi_j^{y}\right)$.

One important implication due to the *additive* property of Poisson distribution is:

$$e_{i,j}^{x,y,t} = \sum_{z=0}^{k} e_{i,j}^{x,y,z} \sim Poisson\left(M^t \theta_{x,y} s_{i,j}^{x,y,t}\right) \tag{1}$$

where $s_{i,j}^{x,y,t} = \sum_{z=1}^{k} \rho_z \phi_i^{x,z}\phi_j^{y,z} + \rho_0 \phi_i^{x,0}\phi_j^{y}$.

This leads to a 'collapsed' model as depicted in Fig. 4b. Though we have so far assumed the link weight to be an integer, this collapsed model remains valid with non-integer link weights (due to Lemma 1, discussed in Sect. 3.1.2).

Given the model parameters, the probability of all observed links is:

$$p\left(\{e_{i,j}^{x,y,t}\}|\theta, \rho, \phi\right) = \prod_{v_i^x, v_j^y} \frac{\left(M^t \theta_{x,y} s_{i,j}^{x,y,t}\right)^{e_{i,j}^{x,y,t}} \exp\left(-M^t \theta_{x,y} s_{i,j}^{x,y,t}\right)}{e_{i,j}^{x,y,t}!} \tag{2}$$

We learn the parameters by the *Maximum Likelihood* (ML) principle: find the parameter values that maximize the likelihood in Eq. (2). First, we take the logarithm and remove the part independent of the parameters:

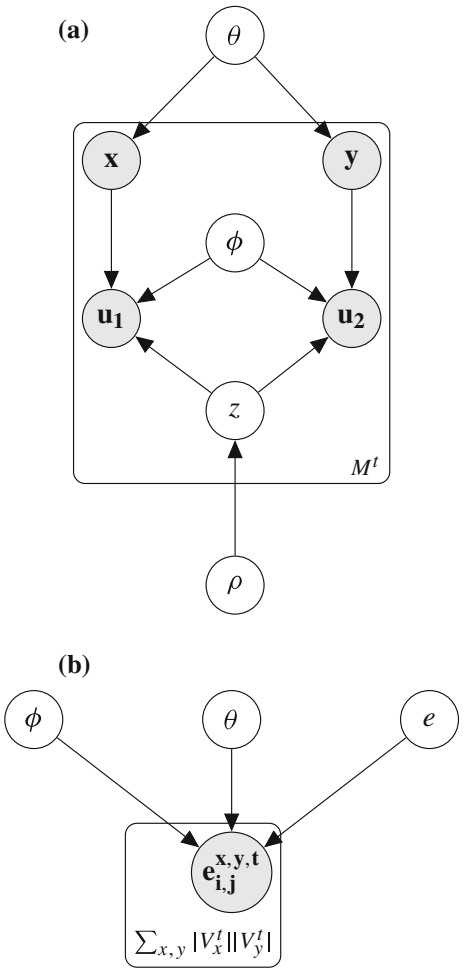


Fig. 4 Two graphical representations of our generative model for links in a topic t. The models are asymptotically equivalent. **a** The generative process of the 'unit-weight' links. **b** The 'collapsed' generative process of the link weights

$$
\begin{aligned}
L(\theta, \rho, \phi) &= \sum_{v_i^x, v_j^y} \left[e_{i,j}^{x,y,t} \log \left(\theta_{x,y} s_{i,j}^{x,y,t} \right) - M^t \theta_{x,y} s_{i,j}^{x,y,t} \right] \\
&= \sum_{v_i^x, v_j^y} e_{i,j}^{x,y,t} \left[\log \theta_{x,y} + \log \left(\sum_{z=1}^{k} \rho_z \phi_i^{x,z} \phi_j^{y,z} + \rho_0 \phi_i^{x,0} \phi_j^{y} \right) \right] \\
&\quad - M^t \sum_{v_i^x, v_j^y} \theta_{x,y} \left(\sum_{z=1}^{k} \rho_z \phi_i^{x,z} \phi_j^{y,z} + \rho_0 \phi_i^{x,0} \phi_j^{y} \right)
\end{aligned}
$$

The gradient method is hard to apply due to the logarithm of summation. We introduce an auxiliary distribution $q_{i,j}^{x,y}$ over subtopics for every link (v_i^x, v_j^y). They satisfy

$$
\sum_{z=0}^{k} q_{i,j}^{x,y,z} = 1, \forall v_i^x, v_j^y \tag{3}
$$

Due to Jensen's inequality, we have:

$$\begin{aligned}
&\log\left(\sum_{z=1}^{k}\rho_z\phi_i^{x,z}\phi_j^{y,z}+\rho_0\phi_i^{x,0}\phi_j^{y}\right)\\
&=\log\left(\sum_{z=1}^{k}q_{i,j}^{x,y,z}\frac{\rho_z\phi_i^{x,z}\phi_j^{y,z}}{q_{i,j}^{x,y,z}}+q_{i,j}^{x,y,0}\frac{\rho_0\phi_i^{x,0}\phi_j^{y}}{q_{i,j}^{x,y,0}}\right)\\
&\geq\sum_{z=1}^{k}q_{i,j}^{x,y,z}\log\frac{\rho_z\phi_i^{x,z}\phi_j^{y,z}}{q_{i,j}^{x,y,z}}+q_{i,j}^{x,y,0}\log\frac{\rho_0\phi_i^{x,0}\phi_j^{y}}{q_{i,j}^{x,y,0}}
\end{aligned}$$

where the equality holds if and only if:

$$\frac{\rho_z\phi_i^{x,z}\phi_j^{y,z}}{q_{i,j}^{x,y,z}}=\frac{\rho_0\phi_i^{x,0}\phi_j^{y}}{q_{i,j}^{x,y,0}},\forall z=1,\ldots,k \tag{4}$$

We define an auxiliary function $F(q,\theta,\rho,\phi)$:

$$\begin{aligned}
F(q,\theta,\rho,\phi)=&\sum_{v_i^x,v_j^y}e_{i,j}^{x,y,t}\left(\log\theta_{x,y}+\sum_{z=1}^{k}q_{i,j}^{x,y,z}\log\frac{\rho_z\phi_i^{x,z}\phi_j^{y,z}}{q_{i,j}^{x,y,z}}+q_{i,j}^{x,y,0}\log\frac{\rho_0\phi_i^{x,0}\phi_j^{y}}{q_{i,j}^{x,y,0}}\right)\\
&-M^t\sum_{v_i^x,v_j^y}\theta_{x,y}\left(\sum_{z=1}^{k}\rho_z\phi_i^{x,z}\phi_j^{y,z}+\rho_0\phi_i^{x,0}\phi_j^{y}\right)
\end{aligned}$$

F can be maximized by iteratively applying two alternating steps: (i) Fix θ,ρ,ϕ, choose q to optimize F; (ii) Fix q, choose θ,ρ,ϕ to optimize F. For i), Eqs. (4) and (3) together imply:

$$q_{i,j}^{x,y,z}(\theta,\rho,\phi)=\frac{\rho_z\phi_i^{x,z}\phi_j^{y,z}}{\sum_{c=1}^{k}\rho_c\phi_i^{x,c}\phi_j^{y,c}+\rho_0\phi_i^{x,0}\phi_j^{y}},z=1,\ldots,k \tag{5}$$

$$q_{i,j}^{x,y,0}(\theta,\rho,\phi)=\frac{\rho_0\phi_i^{x,0}\phi_j^{y}}{\sum_{c=1}^{k}\rho_c\phi_i^{x,c}\phi_j^{y,c}+\rho_0\phi_i^{x,0}\phi_j^{y}} \tag{6}$$

For ii), we use the Lagrange multiplier method to incorporate the probability constraints for θ, ρ and ϕ. The gradient method yields a closed-form solution:

$$\theta_{x,y}(q)=\frac{\sum_{v_i^x,v_j^y}e_{i,j}^{x,y,t}}{M^t} \tag{7}$$

$$\rho_z(q)=\frac{\sum_{v_i^x,v_j^y}e_{i,j}^{x,y,t}q_{i,j}^{x,y,z}}{M^t} \tag{8}$$

$$\phi_i^{x,z}(q)=\frac{\sum_{v_j^y}(e_{i,j}^{x,y,t}q_{i,j}^{x,y,z}+e_{j,i}^{y,x,t}q_{j,i}^{y,x,z})}{\sum_{v_u^x,v_j^y}(e_{u,j}^{x,y,t}q_{u,j}^{x,y,z}+e_{j,u}^{y,x,t}q_{j,u}^{y,x,z})},z=1,\ldots,k \tag{9}$$

$$\phi_i^{x,0}(q)=\frac{\sum_{v_j^y}e_{i,j}^{x,y,t}q_{i,j}^{x,y,0}}{\sum_{v_u^x,v_j^y}e_{u,j}^{x,y,t}q_{u,j}^{x,y,0}} \tag{10}$$

During the iterations, the value of function F keeps non-decreasing. Let $q^{(a)}, \theta^{(a)}, \rho^{(a)}, \phi^{(a)}$ denote the parameter values after the ath iteration. We then have: $L(\theta^{(a)}, \rho^{(a)}, \phi^{(a)}) = F(q^{(a+1)}, \theta^{(a)}, \rho^{(a)}, \phi^{(a)})$. Therefore, the value of $L(\theta^{(a)}, \rho^{(a)}, \phi^{(a)})$ also keeps non-decreasing during the iterations. Since the function L is upper bounded, $L(\theta^{(a)}, \rho^{(a)}, \phi^{(a)})$ eventually converges to a local maximum.

This solution is similar to an Expectation-Maximization (EM) algorithm that is used for ML inference for many statistical models. In fact, we have the following theorem.

Theorem 1 *The solution Eqs.* (5)–(10) *derived from the collapsed model (Fig.* 4*a) is equivalent to an EM solution derived from the unrolled model (Fig.* 4*b)*

Proof In the unrolled model, the likelihood of a unit-weight link can be written as:

$$p(x, y, u_1, u_2|\theta, \rho, \phi) = p(x, y|\theta) \sum_z p(z|\rho) p(u_1|x, z, \phi) p(u_2|y, z, \phi)$$

in which the topic z is a latent variable. The EM algorithm iteratively applies the following two steps.

Expectation step (E-step) Calculate the expected value of the log likelihood function with respect to the conditional distribution of latent variables given observed variables under the current estimate of the parameters $\theta^{(a)}, \rho^{(a)}, \phi^{(a)}$.

$$\begin{aligned}
&Q(\theta, \rho, \phi|\theta^{(a)}, \rho^{(a)}, \phi^{(a)}) \\
&= \sum_{x,y,u_1,u_2} \sum_z p(z|x, y, u_1, u_2, \theta^{(a)}, \rho^{(a)}, \phi^{(a)}) \log p(x, y, u_1, u_2|z, \theta, \rho, \phi) \\
&= \sum_{x,y,u_1,u_2} \sum_z p(z|x, y, u_1, u_2, \theta^{(a)}, \rho^{(a)}, \phi^{(a)}) \log p(x, y|\theta) p(u_1|z, \phi) p(u_2|z, \phi)
\end{aligned}$$

Applying Bayes' theorem, we have:

$$\begin{aligned}
p(z|x, y, u_1, u_2, \theta^{(a)}, \rho^{(a)}, \phi^{(a)}) &= \frac{p(z, x, y, u_1, u_2|\theta^{(a)}, \rho^{(a)}, \phi^{(a)})}{p(x, y, u_1, u_2|\theta^{(a)}, \rho^{(a)}, \phi^{(a)})} \\
&= \frac{p(x, y|\theta^{(a)} p(z|\rho^{(a)}) p(u_1|z, \phi^{(a)}) p(u_2|z, \phi^{(a)})}{\sum_c p(x, y|\theta^{(a)}) p(c|\rho^{(a)}) p(u_1|c, \phi^{(a)}) p(u_2|c, \phi^{(a)})} \\
&= \frac{p(z|\rho^{(a)}) p(u_1|z, \phi^{(a)}) p(u_2|z, \phi^{(a)})}{\sum_c p(c|\rho^{(a)}) p(u_1|c, \phi^{(a)}) p(u_2|c, \phi^{(a)})} \\
&= \begin{cases} \frac{\rho_z \phi_{u_1}^{x,z} \phi_{u_2}^{y,z}}{\sum_{c=1}^k \rho_c \phi_{u_1}^{x,c} \phi_{u_2}^{y,c} + \rho_0 \phi_{u_1}^{x,0} \phi_{u_2}^{y}} & z \in C^t \\ \frac{\rho_0 \phi_{u_1}^{x,0} \phi_{u_2}^{y}}{\sum_{c=1}^k \rho_c \phi_{u_1}^{x,c} \phi_{u_2}^{y,c} + \rho_0 \phi_{u_1}^{x,0} \phi_{u_2}^{y}} & z = 0 \end{cases}
\end{aligned} \tag{11}$$

We omit the superscript (a) in Eq. (11). Comparing Eq. (11) with Eqs. (5) and (6), we find that the auxiliary distribution q we introduced above is actually equal to the posterior distribution over the topics on each link. Now we can write Q as:

$$
\begin{aligned}
&Q(\theta, \rho, \phi|\theta^{(a)}, \rho^{(a)}, \phi^{(a)}) \\
&= \sum_{x,y,u_1,u_2} \sum_{z} p(z|x, y, u_1, u_2, \theta^{(a)}, \rho^{(a)}, \phi^{(a)}) \log p(x, y|\theta) p(u_1|z, \phi) p(u_2|z, \phi) \\
&= \sum_{x,y,e_{i,j}^{x,y,t} \in E_{x,y}^t} e_{i,j}^{x,y,t} \left(\sum_{z=1}^{k} q_{i,j}^{x,y,z(a)} \log \phi_i^{x,z} \phi_j^{y,z} + q_{i,j}^{x,y,0(a)} \log \phi_i^{x,0} \phi_j^{y} + \log \theta_{x,y} \right)
\end{aligned}
$$

Maximization step (M-step) Find the parameters that maximize Q:

$$
\theta^{(a+1)}, \rho^{(a+1)}, \phi^{(a+1)} = \arg\max_{\theta,\rho,\phi} Q(\theta, \rho, \phi|\theta^{(a)}, \rho^{(a)}, \phi^{(a)})
$$

Using the Lagrange multiplier method, we can obtain the solution:

$$
\theta_{x,y}^{(a+1)} = \frac{\sum_{e_{i,j}^{x,y,t} \in E_{x,y}^t} e_{i,j}^{x,y,t}}{M^t} \tag{12}
$$

$$
\rho_z^{(a+1)} = \frac{\sum_{e_{i,j}^{x,y,t} \in E_{x,y}^t} e_{i,j}^{x,y,t} q_{i,j}^{x,y,z(a)}}{M^t} \tag{13}
$$

$$
\phi_i^{x,z(a+1)} = \frac{\sum_{e_{i,j}^{x,y,t} \in E_{x,y}^t} e_{i,j}^{x,y,t} q_{i,j}^{x,y,z(a)} + \sum_{e_{j,i}^{y,x,t} \in E_{y,x}^t} e_{j,i}^{y,x,t} q_{j,i}^{y,x,z(a)}}{\sum_{e_{u,j}^{x,y,t} \in E_{x,y}^t} e_{u,j}^{x,y,t} q_{u,j}^{x,y,z(a)} + \sum_{e_{j,u}^{y,x,t} \in E_{y,x}^t} e_{j,u}^{y,x,t} q_{j,u}^{y,x,z(a)}} \tag{14}
$$

$$
\phi_i^{x,0(a+1)} = \frac{\sum_{e_{i,j}^{x,y,t} \in E_{x,y}^t} e_{i,j}^{x,y,t} q_{i,j}^{x,y,0(a)}}{\sum_{e_{u,j}^{x,y,t} \in E_{x,y}^t} e_{u,j}^{x,y,t} q_{u,j}^{x,y,0(a)}} \tag{15}
$$

It is easy to verify the equivalence of Eqs. (12)–(15) and Eqs. (7)–(10). □

This theorem shows that we can derive the same solution from both the unrolled generative model and the collapsed model. The unrolled model is natural and intuitive, but the collapsed model is easier for extension, as we will see in next subsection.

The theorem also incarnates q as a posterior distribution over the topics on each link. Based on it, we can calculate the expected number of topic-z links between every two nodes:

$$
\hat{e}_{i,j}^{x,y,z} = e_{i,j}^{x,y,t} q_{i,j}^{x,y,z} \tag{16}
$$

Then we have the update rules based on $\hat{e}_{i,j}^{x,y,z}$:
E-step:

$$
\hat{e}_{i,j}^{x,y,z} = \frac{e_{i,j}^{x,y,t} \rho_z \phi_i^{x,z} \phi_j^{y,z}}{\sum_{c=1}^{k} \rho_c \phi_i^{x,c} \phi_j^{y,c} + \rho_0 \phi_i^{x,0} \phi_j^{y}} \tag{17}
$$

$$
\hat{e}_{i,j}^{x,y,0} = \frac{e_{i,j}^{x,y,t} \rho_0 \phi_i^{x,0} \phi_j^{y}}{\sum_{c=1}^{k} \rho_c \phi_i^{x,c} \phi_j^{y,c} + \rho_0 \phi_i^{x,0} \phi_j^{y}} \tag{18}
$$

M-step:

$$\theta_{x,y} = \frac{\sum_{v_i^x, v_j^y} e_{i,j}^{x,y,t}}{M^t} \tag{19}$$

$$\rho_z = \sum_{v_i^x, v_j^y} \frac{\hat{e}_{i,j}^{x,y,z}}{M^t} \tag{20}$$

$$\phi_i^{x,z} = \frac{\sum_{v_j^y} (\hat{e}_{i,j}^{x,y,z} + \hat{e}_{j,i}^{y,x,z})}{\sum_{v_u^x, v_j^y} (\hat{e}_{u,j}^{x,y,z} + \hat{e}_{j,u}^{y,x,z})} \tag{21}$$

$$\phi_i^{x,0} = \frac{\sum_{v_j^y} \hat{e}_{i,j}^{x,y,0}}{\sum_{v_u^x, v_j^y} \hat{e}_{u,j}^{x,y,0}} \tag{22}$$

These equations are intuitive. In E-step, the expected link weight of each subtopic $\hat{e}$ is calculated from the posterior distribution q given the current parameter estimates. This can be viewed as soft clustering of links. In M-step, the parameters are re-estimated based on the link clustering: the link type weight $\theta_{x,y}$ is calculated as dividing the total link weight of type (x, y) by the total link weight; the topic distribution ρ is estimated by the expected number of links in each topic; and the ranking distribution over nodes in each topic z is estimated by the total number of topic-z links associated these nodes.

We update $\hat{e}$, ϕ, ρ in each iteration because $\theta_{x,y}$ is a constant. The EM algorithm can be run multiple times with random initializations, and the solution with the best likelihood will be chosen.

The subnetwork for topic z is naturally extracted from the estimated $\hat{e}$ (expected link weight attributed to each topic). For efficiency purposes, we remove links whose weight is less than 1, and then filter out all resulting isolated nodes. We can then recursively apply the same generative model to the constructed subnetworks until the desired hierarchy is constructed.

3.1.2 Learning link type weights

The generative model described above does not differentiate between the importance of different link types. However, we may wish to discover topics that are biased toward certain types of links, and the bias may vary at different levels of the hierarchy. For example, in the computer science domain, the links between venues and other entities may be more important indicators than other link types in the top level of the hierarchy; however, these same links may be less useful for discovering subareas in the lower levels (e.g., authors working in different subareas may publish in the same venue).

We therefore extend our model to capture the importance of different link types. We introduce a *link type weight* $\alpha_{x,y} > 0$ for each link type (x, y). We use these weights to scale a link's observed weight up or down, so that a unit-weight link of type (x, y) in the original network will have a *scaled* weight $\alpha_{x,y}$. Thus, a link of type (x, y) is valued more when $\alpha_{x,y} > 1$, less when $0 < \alpha_{x,y} < 1$, and becomes negligible as $\alpha_{x,y}$ approaches 0.

When the link type weights $\alpha_{x,y}$ are specified for our model, the EM inference algorithm is unchanged, with the exception that all the $e_{i,j}^{x,y,t}$ in the update equations should be replaced by $\alpha_{x,y} e_{i,j}^{x,y,t}$. When all $\alpha_{x,y}$'s are equal, the weight-learning model reduces to the basic model.

Most of the time, the weights of the link types will not be specified explicitly by users and must therefore be learned from the data.

We first note an important property of our model, justifying our previous claim that link weights need not be integers.

Lemma 1 [Scale invariant] *The EM solution is invariant to a constant scaleup of all the link weights. That is, if we replace all the* $e_{i,j}^{x,y,t}$ *with* $ce_{i,j}^{x,y,t}$, *the resulting* $q_{i,j}^{x,y,z}$, ρ_z, $\theta_{x,y}$ *and* $\phi_i^{x,z}$ *all remain unchanged for topic* t *and all descendant topics of* t.

The proof is straightforward by induction.

With the scale-invariant property on the link weights, we can prove the following theorem.

Theorem 2 *For a set of* l *positive numbers* $\alpha_{x,y} > 0$, *there exist another set of* l *positive numbers* $\beta_{x,y} > 0$, *such that the EM solution based on link weights* $\alpha_{x,y}$ *and* $\beta_{x,y}$ *are identical, and* $\prod_{e_{i,j}^{x,y,t}>0} e_{i,j}^{x,y,t} = \prod_{e_{i,j}^{x,y,t}>0} (\beta_{x,y} e_{i,j}^{x,y,t})$.

Proof Let $\pi = \dfrac{\prod_{e_{i,j}^{x,y,t}>0} e_{i,j}^{x,y,t}}{\prod_{e_{i,j}^{x,y,t}>0} (\alpha_{x,y} e_{i,j}^{x,y,t})}$, $N = \sum_{x,y} n_{x,y}$. We define:

$$\beta_{x,y} \equiv \pi^{\frac{1}{N}} \alpha_{x,y} \tag{23}$$

The scale-invariant property implies that the EM solution based on link weights $\alpha_{x,y}$ and $\beta_{x,y}$ are identical. So we have:

$$\begin{aligned} \prod_{e_{i,j}^{x,y,t}>0} (\beta_{x,y} e_{i,j}^{x,y,t}) &= \prod_{e_{i,j}^{x,y,t}>0} (\pi^{\frac{1}{N}} \alpha_{x,y} e_{i,j}^{x,y,t}) \\ &= \pi \prod_{e_{i,j}^{x,y,t}>0} (\alpha_{x,y} e_{i,j}^{x,y,t}) = \prod_{e_{i,j}^{x,y,t}>0} e_{i,j}^{x,y,t} \end{aligned} \tag{24}$$

□

With this theorem, we can assume that *w.l.o.g.*, the product of all the nonzero link weights remains invariant before and after scaling:

$$\prod_{e_{i,j}^{x,y,t}>0} e_{i,j}^{x,y,t} = \prod_{e_{i,j}^{x,y,t}>0} (\alpha_{x,y} e_{i,j}^{x,y,t}) \tag{25}$$

that reduces to $\prod_{x,y} \alpha_{x,y}^{n_{x,y}} = 1$, where $n_{x,y} = |E_{x,y}^t|$ is the number of nonzero links with type (x, y). With this constraint, we maximize the likelihood $p(\{e_{i,j}^{x,y,t}\}|\theta, \rho, \phi, \alpha)$:

$$\max \prod_{v_i^x, v_j^y} \frac{(\alpha_{x,y} M_{x,y}^t s_{i,j}^{x,y,t})^{\alpha_{x,y} e_{i,j}^{x,y,t}} \exp(-\alpha_{x,y} M_{x,y}^t s_{i,j}^{x,y,t})}{(\alpha_{x,y} e_{i,j}^{x,y,t})!} \tag{26}$$

$$s.t. \prod_{x,y} \alpha_{x,y}^{n_{x,y}} = 1, \alpha_{x,y} > 0 \tag{27}$$

where $M^t_{x,y} = \sum_{v_i^x, v_j^y} e^{x,y,t}_{i,j}$ is the total weight for type (x, y) links. With Stirling's approximation $n! \sim (\frac{n}{e})^n \sqrt{2\pi n}$, we rewrite the log likelihood:

$$\max \sum_{v_i^x, v_j^y} \Big(\alpha_{x,y} e^{x,y,t}_{i,j} \log(\alpha_{x,y} M^t_{x,y} s^{x,y,t}_{i,j}) - \alpha_{x,y} M^t_{x,y} s^{x,y,t}_{i,j} \tag{28}$$

$$-\alpha_{x,y} e^{x,y,t}_{i,j} [\log(\alpha_{x,y} e^{x,y,t}_{i,j}) - 1] - \frac{1}{2} \log(\alpha_{x,y} e^{x,y,t}_{i,j}) \Big)$$

$$s.t. \sum_{x,y} n_{x,y} \log \alpha_{x,y} = 0 \tag{29}$$

Using the Lagrange multiplier method, we can find the optimal value for α when the other parameters are fixed:

$$\alpha_{x,y} = \frac{\left[\prod_{x,y} \left(\frac{1}{n_{x,y}} \sum_{i,j} e^{x,y,t}_{i,j} \log \frac{e^{x,y,t}_{i,j}}{M^t_{x,y} s^{x,y,t}_{i,j}} \right)^{n_{x,y}} \right]^{\frac{1}{\sum_{x,y} n_{x,y}}}}{\frac{1}{n_{x,y}} \sum_{i,j} e^{x,y,t}_{i,j} \log \frac{e^{x,y,t}_{i,j}}{M^t_{x,y} s^{x,y,t}_{i,j}}} \tag{30}$$

With some transformation of the denominator:

$$\begin{aligned} \sigma_{x,y} &= \frac{1}{n_{x,y}} \sum_{i,j} e^{x,y,t}_{i,j} \log \frac{e^{x,y,t}_{i,j}}{M^t_{x,y} s^{x,y,t}_{i,j}} \\ &= \frac{M^t_{x,y}}{n_{x,y}} \sum_{v_i^x, v_j^y} \frac{e^{x,y,t}_{i,j}}{M^t_{x,y}} \log \frac{e^{x,y,t}_{i,j} / M^t_{x,y}}{s^{x,y,t}_{i,j}} \end{aligned} \tag{31}$$

we can see more clearly that the link type weight is negatively correlated with two factors: the average link weight $\frac{M^t_{x,y}}{n_{x,y}}$ and the KL-divergence of the expected link weight distribution to the observed link weight distribution $\sum_{v_i^x, v_j^y} \frac{e^{x,y,t}_{i,j}}{M^t_{x,y}} \log \frac{e^{x,y,t}_{i,j} / M^t_{x,y}}{s^{x,y,t}_{i,j}}$. The first factor is used to balance the scale of link weights of different types (e.g., a type-1 link always has X times greater weight than a type-2 link). The second factor measures the importance of a link type in the model. The more the prediction diverges from the observation, the worse the quality of a link type.

So we have the following iterative algorithm for optimizing the joint likelihood:

1. Initialize all the parameters.
2. Fixing α, update ρ, ϕ using EM equations:

$$\hat{e}^{x,y,z}_{i,j} = \frac{\alpha_{x,y} e^{x,y,t}_{i,j} \rho_z \phi^{x,z}_i \phi^{y,z}_j}{\sum_{c=1}^k \rho_c \phi^{x,c}_i \phi^{y,c}_j + \rho_0 \phi^{x,0}_i \phi^y_j} \tag{32}$$

$$\hat{e}^{x,y,0}_{i,j} = \frac{\alpha_{x,y} e^{x,y,t}_{i,j} \rho_0 \phi^{x,0}_i \phi^y_j}{\sum_{c=1}^k \rho_c \phi^{x,c}_i \phi^{y,c}_j + \rho_0 \phi^{x,0}_i \phi^y_j} \tag{33}$$

$$\rho_z = \frac{\sum_{v_i^x, v_j^y} \hat{e}^{x,y,z}_{i,j}}{\sum_{x,y} \alpha_{x,y} M^t_{x,y}} \tag{34}$$

$$\phi_i^{x,z} = \frac{\sum_{v_j^y}(\hat{e}_{i,j}^{x,y,z} + \hat{e}_{j,i}^{y,x,z})}{\sum_{v_u^x, v_j^y}(\hat{e}_{u,j}^{x,y,z} + \hat{e}_{j,u}^{y,x,z})} \tag{35}$$

$$\phi_i^{x,0} = \frac{\sum_{v_j^y} \hat{e}_{i,j}^{x,y,0}}{\sum_{v_u^x, v_j^y} \hat{e}_{u,j}^{x,y,0}} \tag{36}$$

3. Fixing ρ, ϕ, update α using Eq. (30).
4. Repeat steps 2) and 3) until the likelihood converges.

In each iteration, the time complexity is $O(\sum_{x,y} n_{x,y})$, i.e., linear to the total number of nonzero links. The likelihood is guaranteed to converge to a local optimum. Once again, a random initialization strategy can be employed to choose a solution with the best local optimum.

3.2 Topical pattern mining and ranking

Having discovered the topics using our generative model, we can now identify the most representative topical patterns for each topic. This is done in two stages: topical pattern mining and ranking the mined patterns.

Pattern mining in each topic A pattern P^x of type x is a set of type-x nodes: $P^x = \{v_i^x\}$. For example, a pattern of a 'term' type is a set of unigrams that make up a phrase, such as $\{support, vector, machine\}$ (or 'support vector machine' for simpler notation). A more general definition of a pattern can involve mixed node types within one pattern, but is beyond the scope of this paper.

A pattern P that is regarded to be representative for a topic t must first and foremost be frequent in the topic. The frequency of a pattern $f(P)$ is the number of documents (or other meaningful information chunks) that contain all the nodes in the pattern (or the number of star objects that are linked to all the nodes). The pattern must also have sufficiently high *topical frequency* in topic t.

Definition 2 (Topical Frequency) The topical frequency $f_t(P)$ of a pattern is the number of times the pattern is attributed to topic t. For the root node o, $f_o(P) = f(P)$. For each topic node with subtopics C^t, $f_t(P) = \sum_{z \in C^t} f_z(P)$ (i.e., topical frequency is the sum of sub-topical frequencies.)

Table 2 illustrates a hypothetical example of estimating topical frequency for patterns of various types (term, author, and venue) in a computer science topic that has 4 subtopics.

Table 2 A hypothetical example of estimating topical frequency

Pattern	ML	DB	DM	IR	Total
Support vector machines	85	0	0	0	85
Query processing	0	212	27	12	251
Hui Xiong	0	0	66	6	72
SIGIR conference	444	378	303	1,117	2,242

The topics are assumed to be inferred as machine learning, database, data mining, and information retrieval from the data

We estimate the topical frequency of a pattern based on two assumptions: (i) For a type-x topic-t pattern of length n, each of the n nodes is generated with the distribution $\phi^{x,t}$, and (ii) the total number of topic-t phrases of length n is proportional to ρ_t.

$$f_t(P^x) = f_{Par(t)}(P^x) \frac{\rho_t \prod_{v_i^x \in P^x} \phi_i^{x,t}}{\sum_{z \in C^{Par(t)}} \rho_z \prod_{v_i^x \in P^x} \phi_i^{x,z}} \tag{37}$$

Both ϕ and ρ are learned from the generative model as described in Sect. 3.1.

To extract topical frequent patterns, all frequent patterns can first be mined using a pattern mining algorithm such as FP growth [8], and then filtered given some minimal topical frequency threshold $minsup$.

Pattern ranking in each topic There are four criteria for judging the quality of a pattern (similar criteria are proposed for ranking phrases by [21], and we adopt them for other types of patterns).

- *Frequency*—A representative pattern for a topic should have sufficiently high topical frequency.
- *Exclusiveness*—A pattern is exclusive to a topic if it is only frequent in this topic and not frequent in other topics. *Example: 'query processing' is more exclusive than 'query' in the Databases topic.*
- *Cohesiveness*—A group of entities should be combined together as a pattern if they co-occur significantly more often than the expected co-occurrence frequency given the chances of occurring independently. *Example: 'active learning' is a more cohesive pattern than 'learning classification' in the Machine Learning topic.*
- *Completeness*—A pattern is not complete if it rarely occurs without the presence of a longer pattern. *Example: 'support vector machines' is a complete pattern, whereas 'vector machines' is not because 'vector machines' is almost always accompanied by 'support' in occurrence.*

The pattern ranking function should take these criteria into consideration. The ranking function must also be able to directly compare patterns of mixed lengths, such as 'classification,' 'decision trees,' and 'support vector machines.'

Let N_t be the number of documents that contain at least one frequent topic-t pattern, T a subset of $C^{Par(t)}$ that contains t, and N_T the number of documents that contain at least one frequent topic-z pattern for some topic $z \in T$. We use the following ranking function that satisfies all these requirements [21]:

$$r^t(P) = \begin{cases} 0, & \text{if } \exists P' \supsetneq P,\ f_t(P') \geq \gamma f_t(P) \\ p(P|t)\left(\log \frac{p(P|t)}{\max_T p(P|T)} + \omega \log \frac{p(P|t)}{p_{indep}(P|t)}\right) & \text{o.w.} \end{cases} \tag{38}$$

where $p(P|t) = \frac{f_t(P)}{N_t}$ is the occurrence probability of a pattern P, measuring frequency; $p_{indep}(P|t) = \prod_{v \in P} \frac{f_t(v)}{N_t}$ is the probability of independently seeing every node in pattern P, measuring exclusiveness; and $p(P|T) = \frac{\sum_{t \in T} f_t(P)}{N_T}$ is the probability of phrase P conditioned on a mixture T of t and other sibling topics, measuring cohesiveness. Incomplete patterns are filtered if there exists a superpattern P' that has sufficiently high topical frequency compared to P. $\gamma \in [0, 1]$ is a parameter that controls the strictness of the completeness criterion, where a larger value of γ deems more phrases to be complete. Complete phrases are ranked according to a combination of the other three criteria. Frequency plays the most important role. The weight between exclusiveness and cohesiveness is controlled by a parameter $\omega \in [0, +\infty)$, with larger values of ω biasing the ranking more heavily toward

cohesiveness. Due to space limitation, we refer to [21] for more detailed discussion of this ranking function.

3.3 Shape of hierarchy

In this section, we discuss the following issues that affect the shape of the constructed hierarchy.

Number of children for each topic Since our framework is recursive, the shape of the tree is essentially determined by how many children each node has, i.e., how many subtopics each topic has. For every topic, our model can work with arbitrary number of subtopics that is larger than 1. However, it may be more reasonable to have certain number of subtopics than others. In general, we prefer each topic to have a small number of subtopics, e.g., between 2 and 10, in order to make it easy for browsing. For example, if the root has 5 subtopics and each of them has 4 subtopics, the three-level hierarchy is in general easier to browse than directly showing all 20 topics under the root.

Given a range of subtopic number, such as [2, 10], we would like to choose a reasonable number of children k for each topic. It is a model selection problem. Among various model selection strategies in the literature, we select two of them and introduce how they can be adapted for our model.

The first strategy was proposed in [16] to adopt cross validation to choose the parameter K. In our setting, we can first fit the generative model to a sampled subnetwork H^t of the given network G^t. Then, we evaluate the likelihood of the model on the rest part of the network $G^t - H^t$, which is called the held-out network. By checking the averaged held-out likelihood with varying number of sub-clusters, the parameter with the maximum value will be chosen as the best candidate.

The second strategy is based on the Bayesian information criterion (BIC). A similar criterion is Akaike information criterion (AIC). Both BIC and AIC resolve the overfitting problem. When we increase the number of topics k, it is possible to increase the likelihood, but may result in overfitting because the model will have a larger number of parameters. BIC and AIC introduce a penalty term for the number of parameters in the model, and the penalty term is larger in BIC than in AIC. Using BIC, the measure for our model is defined as:

$$BIC = -2 \log p\left(\{e_{i,j}^{x,y,t}\} | \theta, \rho, \phi\right) + |\theta, \rho, \phi| \cdot \log |E^t|$$

where $|\theta, \rho, \phi|$ is the number of free parameters in the model and $|E^t|$ refers to the size of observed links. As we only care about the topic number k, $|\theta, \rho, \phi|$ can be reduced to $|V^t|k$ plus a constant independent of k, where $|V^t|$ is the number of nodes. We can then select k with the largest BIC score.

BIC is derived under the assumptions that the data distribution is in the exponential family. Cross validation only assumes that the sampled network and the held-out network are generated from the same model. Comparing these two criteria, we generally recommend cross validation over BIC when there are sufficient data. However, when the network is small, cross validation is prone to high variation and BIC can be used as an alternative.

Depth of the hierarchy A simple and intuitive strategy to decide the depth of the hierarchy is to rely on the selected number of children mentioned in above. For example, if the best number of topics is $k = 1$, it implies we should stop expanding the current topic node. In practice, we can set a threshold on the largest depth of the tree, as well as the size of the network. Once the tree has reached the maximal depth or the size of the network in current

Table 3 # Links in our datasets

DBLP (# Nodes)	Term (6,998)	Author (12,886)	Venue (20)
Term	693,132	900,201	104,577
Author	–	156,255	99,249
NEWS (# Nodes)	Term (13,129)	Person (4,555)	Location (3,845)
Term	686,007	386,565	506,526
Person	–	53,094	129,945
Location	–	–	85,047

topic is too small, we can cease the recursion. A general implication is that the more children each node has, the less deep the final hierarchy will be.

Balance of subtree size The distribution of ρ_z's determines the size of subtrees. The more evenly distributed are ρ_z's, the more balanced are the subtrees. Generally we would like to generate a balanced tree because it is efficient for browsing. If this is the case, we should randomly initialize the topic of each link from a uniform distribution. In case one would like to generate a skewed hierarchy, the random initialization of each link's topic distribution should follow a non-uniform multinomial, which can be generated from a Dirichlet prior. Our model can be extended into a Bayesian framework, which can incorporate conjugate prior for all the parameters. The shape of the hierarchy can then be controlled by the hyperparameters of prior. We leave it as future work.

4 Experiments

Lack of gold standard is a known issue for unsupervised topic modeling methods. As such, people have proposed evaluation metrics without relying on labels. Our task of constructing multi-typed topic hierarchy is new and there is neither gold standard for it. We leverage the existing evaluation metrics, pointwise mutual information [15] and intrusion detection [2] that are proved to be effective in text-based topic modeling, and modify them to evaluate our multi-typed topic hierarchy. The metrics can be used to compare different methods in arbitrary datasets.

We evaluate the performance of our proposed method on two datasets (see Table 3 for summary statistics of the constructed networks):

- *DBLP* Following [21], we collected 33,313 recently published computer science papers from DBLP.[1] We constructed a heterogeneous network with three node types: term (from paper title), author and venue, and 5 link types: term–term, term–author, term–venue, author–author and author–venue.[2]
- *NEWS* We crawled 43,168 news articles on 16 top stories from Google News,[3] extracted text content from html pages by heuristic rules, and ran an information extraction algo-

[1] We chose papers published in 20 conferences related to the areas of Artificial Intelligence, Databases, Data Mining, Information Retrieval, Machine Learning, and Natural Language Processing from http://www.dblp.org/.

[2] As a paper is always published in exactly one venue, there can naturally be no venue–venue links.

[3] The 16 topics chosen were: Bill Clinton, Boston Marathon, Earthquake, Egypt, Gaza, Iran, Israel, Joe Biden, Microsoft, Mitt Romney, Nuclear power, Steve Jobs, Sudan, Syria, Unemployment, US Crime.

rithm [12] to extract entities. We constructed a heterogeneous network with three node types: term (from article title), person and location, and 6 link types: term–term, term–person, term–location, person–person, person–location and location–location.

Our datasets will be online available at http://illimine.cs.illinois.edu.

Our recursive framework relies on two key steps: subtopic discovery and topical pattern mining. The major contribution of this paper is the subtopic discovery step. Hence, our evaluation is twofold: (i) we evaluate the efficacy of subtopic discovery given a topic and its associated heterogeneous network; and (ii) we perform several 'intruder detection' tasks to evaluate the quality of the constructed hierarchy based on human judgment.

4.1 Efficacy of subtopic discovery

We first present a set of experiments designed to evaluate just the subtopic discovery step (Step 2 in Sect. 3).

Evaluation measure We extend the pointwise mutual information (PMI) metric in order to measure the quality of our multi-typed topics. The metric of pointwise mutual information PMI has been proposed by [15] as a way of measuring the semantic coherence of topics. It is generally preferred over other quantitative metrics such as perplexity or the likelihood of held-out data [20]. In order to measure the quality of our multi-typed topics, we extend the definition of PMI as follows:

For each topic, PMI calculates the average relatedness of each pair of the words ranked at top-K:

$$\mathrm{PMI}(\mathbf{w}, \mathbf{w}) = \frac{2}{K(K-1)} \sum_{1 \le i < j \le K} \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)} \tag{39}$$

where PMI $\in [-\infty, \infty]$, and $\mathbf{w}$ are the top K most probable words of the topic. PMI $= 0$ implies that these words are independent; PMI > 0 (< 0) implies they are overall positively (negatively) correlated.

However, our multi-typed topic contains not only words, but also other types of entities. So we define *heterogeneous* pointwise mutual information as:

$$\mathrm{HPMI}(\mathbf{v}^x, \mathbf{v}^y) = \begin{cases} \frac{2}{K(K-1)} \sum_{1 \le i < j \le K} \log \frac{p(v_i^x, v_j^y)}{p(v_i^x)p(v_j^y)} & x = y \\ \frac{1}{K^2} \sum_{1 \le i,j \le K} \log \frac{p(v_i^x, v_j^y)}{p(v_i^x)p(v_j^y)} & x \ne y \end{cases} \tag{40}$$

where $\mathbf{v}^x$ are the top K most probable type-x nodes in the given topic. When $x = y$, HPMI reduces to PMI. The HPMI score for every link type (x, y) is calculated and averaged to obtain an overall score. We set $K = 20$ for all node types.[4]

Methods for comparison:

- *CATHYHIN (equal weight)*—The weight for every link type is set to be 1.
- *CATHYHIN (learn weight)*—The weight of each link type is learned, as described in Sect. 3.1. No parameters need hand tuning.
- *CATHYHIN (norm weight)*—The weight of each link type is explicitly set as: $\alpha_{x,y} = \frac{1}{\sum_{i,j} e_{i,j}^{x,y}}$. This is a heuristic normalization that forces the total weight of the links for each link type to be equal.

[4] The one exception is venues, as there are only 20 venues in the DBLP dataset, so we set $K = 3$ in this case.

- *NetClus*—The current state-of-the-art clustering and ranking method for heterogeneous networks. We use the implementation by [5]. The smoothing parameter λ_S is tuned by a grid search in [0, 1]. The optimal value for these two domains is 0.3 and 0.5, respectively. Note that the link type weight-learning method for CATHYHIN does not apply to NetClus because NetClus does not have a single objective function to optimize.
- *TopK*—Select the top K nodes from each type according to their frequency to form a pseudo topic. This method serves as a baseline value for the proposed HPMI metric.

Experiment setup We discover the subtopics of four datasets:

- DBLP (20 conferences)—Aforementioned DBLP dataset. This dataset is used for evaluating the performance when constructing the first level of the hierarchy.
- DBLP (database area)—A subset of the DBLP dataset consisting only of papers published in 5 Database conferences. By using this dataset, which roughly represents a subtopic of the full DBLP dataset, we analyze the quality of discovered subtopics in a lower level of the hierarchy.
- NEWS (16 topics)—Aforementioned NEWS dataset.
- NEWS (4 topic subset)—A subset of the NEWS dataset limited to 4 topics, which center around different types of entities: Bill Clinton, Boston Marathon, Earthquake, Egypt.

We use the BIC model selection criterion described in Sect. 3.3 to select k. It aligns with our prior knowledge. For example, on DBLP (20 conferences), $k = 6$ and there are 6 actual areas in the data.

Experiment results All the methods finish in 1.5 h for these datasets, on a Windows server running MATLAB R2011a with Intel Xeon X5650 2.67 GHz and 48 GB RAM.

We show the heterogeneous pointwise mutual information averaged over the learned topics in Tables 4 and 5, our generative model consistently posts a higher HPMI score than NetClus (and TopK) across all links types in every dataset. Although NetClus HPMI values are better than the TopK baseline, the improvement of our best performing method—CATHYHIN (learn weight)—over the TopK baseline are better than the improvement posted by NetClus by factors ranging from 2 to 5.8. Even the improvement over the TopK baseline of CATHYHIN (equal weight), which considers uniform link type weights, is better than the improvement posted by NetClus by factors ranging from 1.6 to 4.6.

CATHYHIN with learned link type weights consistently yields the highest overall HPMI scores, although CATHYHIN with normalized link type weights sometimes shows a slightly higher score for particular link types (e.g., author–author for both DBLP datasets, and person–person for both NEWS datasets). CATHYHIN (norm weight) assigns a high weight to a link type whose total link weights were low in the originally constructed network, pushing the discovered subtopics to be more dependent on that link type. Normalizing the link type weights does improve CATHYHIN performance in many cases, as compared to using uniform link type weights. However, this heuristic determines the link type weight based solely on their link density. It can severely deteriorate the coherence of dense but valuable link types, such as term–term in both DBLP datasets, and rely too heavily on sparse but uninformative entities, such as Venues in the Database subtopic of the DBLP dataset.

Figure 5 demonstrates the learned link weights by CATHYHIN (learn weight) on DBLP datasets. At the first level, the term–venue and author–venue link types are assigned high weight, because the venue is a most important discriminator for general areas (Artificial Intelligence, Databases, Data Mining, Information Retrieval, Machine Learning, and Natural Language Processing). At the second level, the venue links are much less useful for discovering subtopics in each area.

Table 4 Heterogeneous pointwise mutual information in DBLP (20 conferences and database area)

	Term–term	Term–author	Author–author	Term–venue	Author–venue	Overall
DBLP (satabase area)						
TopK	−0.5228	−0.1069	0.4545	0.0348	−0.3650	−0.0761
NetClus	−0.3962	0.0479	0.4337	0.0368	−0.2857	0.0260
CATHYHIN (equal weight)	0.0561	0.4799	0.6496	0.0722	−0.0033	0.3994
CATHYHIN (norm weight)	−0.1514	0.3816	**0.6971**	0.0408	**0.2464**	0.3196
CATHYHIN (learn weight)	**0.3027**	**0.6435**	0.5574	**0.1165**	0.1805	**0.5205**
DBLP (20 conferences)						
TopK	−0.4825	−0.0204	0.5466	−1.0051	−0.4208	−0.0903
NetClus	−0.1995	0.5186	0.5404	0.2851	1.2659	0.4045
CATHYHIN (equal weight)	0.2936	0.8812	0.6595	0.5191	1.0466	0.6949
CATHYHIN (norm weight)	0.1825	0.8674	**0.9476**	0.7472	1.3307	0.7601
CATHYHIN (learn weight)	**0.4964**	**1.0618**	0.7161	**1.1283**	**1.7511**	**0.9168**

The bold implies the highest value in each column

Table 5 Heterogeneous pointwise mutual information in NEWS (16 topics collection and 4 topics subset)

	Term–term	Term–person	Person–person	Term–location	Person–location	Location–location	Overall
NEWS (4 topics subset)							
TopK	−0.2479	0.1671	0.0716	0.0787	0.2483	0.3632	0.1317
NetClus	0.1279	0.3835	0.2909	0.3240	0.4728	0.4271	0.3575
CATHYHIN (equal weight)	**1.0471**	0.7917	0.4902	0.8506	0.6821	0.6586	0.7610
CATHYHIN (norm weight)	0.7975	0.8825	**0.5553**	0.8682	**0.8077**	0.7346	0.8023
CATHYHIN (learn weight)	0.9935	**0.9354**	0.5142	**0.9784**	0.7389	**0.7645**	**0.8434**
NEWS (16 topics)							
TopK	−1.7060	−0.8663	−0.8462	−1.0238	−0.5665	−0.4578	−0.8783
NetClus	−0.3847	0.0943	0.0313	−0.1114	0.1291	0.1376	−0.0274
CATHYHIN (equal weight)	0.7804	1.0170	0.8393	0.8354	0.9467	0.6382	0.8749
CATHYHIN (norm weight)	0.8579	**1.1143**	**0.9086**	0.8530	0.9624	**0.7143**	0.9284
CATHYHIN (learn weight)	**0.9234**	1.1109	0.7966	**0.9731**	**0.9718**	0.6965	**0.9500**

The bold implies the highest value in each column

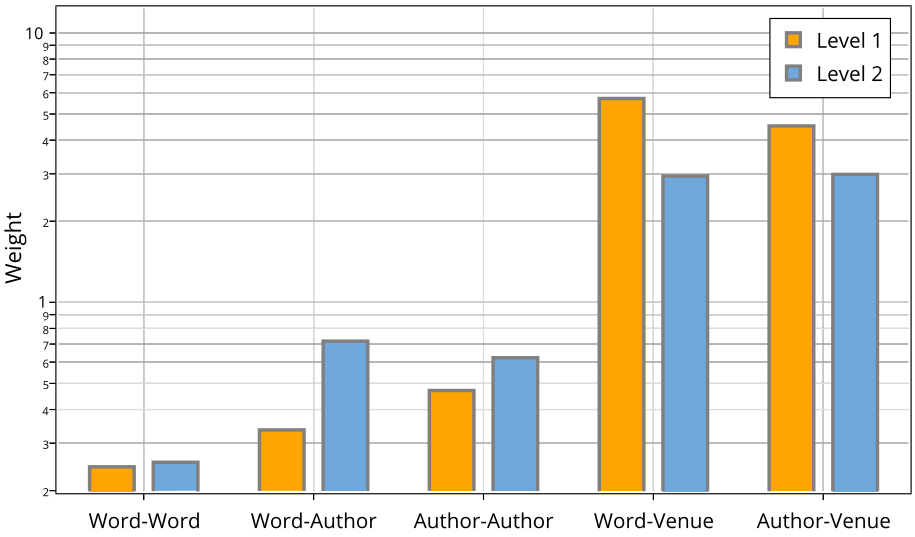


Fig. 5 Learned link weights in DBLP

We may conclude from these experiments that CATHYHIN's unified generative model consistently outperforms the state-of-the-art heterogeneous network analysis technique NetClus. In order to generate coherent, multi-typed topics at each level of a topical hierarchy, it is important to learn the optimal weights of different entity types, which depends on the link type density, the granularity of the topic to be partitioned, and the specific domain.

4.2 Topical hierarchy quality

Our second set of evaluations assesses the ability of our method to construct a hierarchy of multi-typed topics that human judgement deems to be high quality. We generate and analyze multi-typed topical hierarchies using the DBLP dataset (20 conferences) and the NEWS dataset (16 topics collection).

Experiment setup We adapt two tasks from [2], who were the first to explore human evaluation of topic models. Each task involves a set of questions asking humans to discover the 'intruder' object from several options. Three annotators manually completed each task, and their evaluations scores were pooled.

The first task is Phrase Intrusion, which evaluates how well the hierarchies are able to separate phrases in different topics. Each question consists of X ($X = 5$ in our experiments) phrases; $X - 1$ of them are randomly chosen from the top phrases of the same topic and the remaining phrase is randomly chosen from a sibling topic. The second task is Entity Intrusion, a variation that evaluates how well the hierarchies are able to separate entities present in the dataset in different topics. For each entity type, each question consists of X entity patterns; $X - 1$ of them are randomly chosen from the top patterns of the same topic and the remaining entity pattern is randomly chosen from a sibling topic. This task is constructed for each entity type in each dataset (Author and Venue in DBLP; Person and Location in NEWS). The third task is Topic Intrusion, which tests the quality of the parent–child relationships in the generated hierarchies. Each question consists of a parent topic t and X candidate child topics. $X - 1$ of the child topics are actual children of t in the generated hierarchy, and the remaining

Phrase Intrusion

Question 1/210	data mining	association rules	logic programs	data streams
Question 2/210	natural language	query optimization	data management	database systems

Venue Intrusion

Question 1/210	KDD	SDM	ICDE	ICDM
Question 2/210	EMNLP	ACL	AAAI	HLT-NAACL

Question 1/60 **Topic Intrusion**

Parent topic	**Child topic 1**	**Child topic 2**	**Child topic 3**	**Child topic 4**
database systems	web search	data management	query processing	database system
data management	search engine	data integration	query optimization	database design
query processing	semantic web	data sources	query databases	expert system
management system	search results	data warehousing	relational databases	management system
data system	web pages	data applications	query data	design system

Fig. 6 Examples of intruder detection questions

child topic is not. Each topic is represented by its top 5 ranked patterns of each type—e.g., for the NEWS dataset, the top 5 phrases, people, and locations are shown for each topic.

For DBLP, we generate 210 phrase intrusion questions, 210 entity intrusion questions for both author and venue type, and 60 topic intrusion questions. For NEWS, we generate 280 phrase intrusion questions, 280 entity instruction questions for both person and location type, and 100 topic intrusion questions. Figure 6 shows examples of generated questions in DBLP. For each question, 3 human annotators with background knowledge of computer science and news select the intruder phrase, entity, or subtopic. If they are unable to make a choice, or choose incorrectly or inconsistently, the question is marked as a failure.

Methods for comparison:

- *CATHYHIN*—As defined in Sect. 3
- *CATHYHIN*$_1$—The pattern length of text and every entity type is restricted to 1.
- *CATHY*—As proposed by [21], the hierarchy is constructed only from textual information.
- *CATHY*$_1$—The phrase length is restricted to 1.
- *CATHY*$_{heuristic\_HIN}$—Since neither CATHY nor CATHY$_1$ provides topical ranks for entities, we construct this method to have a comparison for the Entity Intrusion task. We use a heuristic entity ranking method based on the textual hierarchy generated by CATHY, and the original links in the network. An entity's rank for a given topic is a function of its frequency in the topic (estimated as the number of documents in that topic which are linked to the entity in the original network), and its exclusivity.
- *NetClus*$_{pattern}$—NetClus is used for subtopic discovery, followed by the topical mining and ranking method of CATHYHIN, as described in Sect. 3.2 (this can also be thought of CATHYHIN, where Step 2 is replaced by NetClus).
- *NetClus*$_{pattern\_1}$—Equivalent to NetClus$_{\text{pattern}}$ with the pattern length of text and every entity type restricted to 1.
- *NetClus*—As defined in [19].

Table 6 Results of intruder detection tasks (% correct intruders identified)

	DBLP				NEWS			
	Phrase	Venue	Author	Topic	Phrase	Location	Person	Topic
CATHYHIN	**0.83**	**0.83**	**1.0**	**1.0**	**0.65**	**0.70**	**0.80**	**0.90**
CATHYHIN_1	0.64	–	–	0.92	0.40	0.55	0.50	0.70
CATHY	0.72	–	–	0.92	0.58	–	–	0.65
CATHY_1	0.61	–	–	0.92	0.23	–	–	0.50
$\text{CATHY}_{\text{heur\_HIN}}$	–	0.78	0.94	0.92	–	0.65	0.45	0.70
$\text{NetClus}_{\text{pattern}}$	0.33	0.78	0.89	0.58	0.23	0.20	0.55	0.45
$\text{NetClus}_{\text{pattern\_1}}$	0.53	–	–	0.58	0.20	0.45	0.30	0.40
NetClus	0.19	0.78	0.83	0.83	0.15	0.35	0.25	0.45

The bold implies the highest value in each column

The pattern mining and ranking parameters for both CATHY and CATHYHIN are set to be $minsup = 5, \omega = \gamma = 0.5$ according to [21]. The optimal smoothing parameter for NetClus is $\lambda_S = 0.3$ and 0.7 in DBLP and NEWS respectively.

Table 6 displays the results of the intruder detection tasks. For the Entity Intrusion task on the DBLP dataset, we restricted the entity pattern length to 1 in order to generate meaningful questions. This renders the methods CATHYHIN_1 and $\text{NetClus}_{pattern\_1}$ equivalent to CATHYHIN and $\text{NetClus}_{pattern}$, respectively, so we omit the former methods from reporting.

Experiment results The Phrase Intrusion task performs much better when phrases are used rather than unigrams, for both CATHYHIN and CATHY, on both datasets. The NEWS dataset exhibits a stronger preference for phrases, as opposed to the DBLP dataset, which may be due to the fact that the terms in the NEWS dataset are more likely to be noisy and uninformative outside of their context, whereas the DBLP terms are more technical and therefore easier to interpret. This characteristic may also help explain why the performance of every method on DBLP data is consistently higher than on NEWS data. However, neither phrase mining and ranking nor unigram ranking can make up for poor performance during the topic discovery step, as seen in the three NetClus variations. Therefore, both phrase representation and high-quality topics are necessary for good topic interpretability.

For the Entity Intrusion task, all of the relevant methods show comparable performance in identifying Author and Venue intruders in the DBLP dataset (though CATHYHIN is still consistently the highest). Since the DBLP dataset is well structured, and the entity links are highly trustworthy, identifying entities by topic is likely easier. However, the entities in the NEWS dataset were automatically discovered from the data, and the link data are therefore noisy and imperfect. CATHYHIN is the most effective in identifying both Location and Person intruders. Once again, both better topic discovery and improved pattern representations are responsible for CATHYHIN's good results and simply enhancing the pattern representations, whether for CATHY or NetClus, cannot achieve competitive performance.

CATHYHIN performs very well in the Topic Intrusion task on both datasets. Similar to the Phrase Intrusion task, both CATHYHIN and CATHY yield equally good or better result when phrases and entity patterns are mined, rather than just terms and single entities. The fact that CATHYHIN always outperforms CATHY demonstrates that utilizing entity link information is indeed helpful for improving topical hierarchy quality. In all three intruder detection tasks on both datasets, CATHYHIN consistently outperforms all other methods, showing that an

Table 7 The 'information retrieval' topic, as generated by three methods

CATHYHIN	$\text{CATHY}_{\text{heuristic\_HIN}}$	$\text{NetClus}_{\text{pattern}}$
{information retrieval; web search; retrieval}/{W. Bruce Croft; Iadh Ounis; James Allen}/{SIGIR; WWW; ECIR}	{information retrieval; search engine; web search}/{Ryen W. White; C. Lee Giles; Mounia Lalmas}/{SIGIR; WWW; ECIR}	{information retrieval; statistical machine translation; conditional random fields}/{W. Bruce Croft; Zheng Chen; Chengxiang Zhai}/{ACL; SIGIR; HLT-NAACL}

Table 8 The 'Egypt' topic and the least sensible subtopic, as generated by three methods (only phrases and locations are shown)

CATHYHIN	$\text{CATHY}_{\text{heuristic\_HIN}}$	$\text{NetClus}_{\text{pattern}}$
{egypt; egypts; death toll; morsi}/{Egypt; Egypt Cairo; Egypt Israel; Egypt Gaza}	{egypt; egypts morsi; egypt imf loan; egypts president}/{Egypt; Cairo; Tahrir Square; Port Said}	{bill clinton; power nuclear; rate unemployment; south sudan}/{Egypt Cairo; Egypt Coptic; Israel Jerusalem; Libya Egypt}
↓	↓	↓
{death toll; egyptian; sexual harassment; egypt soccer}/{Egypt Cairo; Egypt Gaza; Egypt Israel}	{supreme leader; army general sex; court; supreme court}/{US; Sudan; Iran; Washington}	{egypts coptic pope; egypts christians; obama romney; romney campaign}/{Egypt Cairo; Egypt Coptic; Israel Jerusalem; Egypt}

integrated heterogeneous model consistently produces a more robust hierarchy, which is more easily interpreted by human judgment.

4.3 Case study

In Sect. 4.2, we analyzed the two main reasons for the performance gain of CATHYHIN: improved pattern representations and better topic discovery. In Fig. 1, we have shown real examples of the constructed hierarchy in the DBLP data. It is clear that the multi-typed entities enrich the context of each topic and improve the representations of text-only topics. Here, we use one simple example to illustrate the topic discovery performance, using the same topic representations.

Table 7 illustrates three representations of the topic 'information retrieval' (one of the 6 areas in DBLP dataset). Overall, CATHYHIN finds more 'pure' information retrieval entities. $\text{CATHY}_{heuristic\_HIN}$ generates very similar top-ranked phrases and venues, but different authors. While the top authors found by $\text{CATHY}_{heuristic\_HIN}$ indeed work on information retrieval, they spread their interest in other fields too, such as data mining and human computer interaction. This is because $\text{CATHY}_{heuristic\_HIN}$ only uses the links between text and entities to rank entities posterior to the text-based topic discovery, while CATHYHIN can further use the author–author and author–venue links to refine the topics and find the more accurate position for each entity. $\text{NetClus}_{pattern}$ also utilizes the multiple types of links, but it mixes different topics, such as information retrieval and natural language processing, perhaps due to the hard partitioning of papers and heuristic combination of ranking and clustering.

As a worst-case study, Table 8 illustrates three representations of the topic 'Egypt' (one of the 16 top stories in NEWS dataset), each with its least comprehensible subtopic. The loca-

tions found within the CATHYHIN subtopic are sensible. However, $\text{CATHY}_{heuristic\_HIN}$ first constructs phrase-represented topics from text and then uses entity link information to rank entities in each topic. Thus the entities are not assured to fit well into the constructed topic, and indeed, the $\text{CATHY}_{heuristic\_HIN}$ subtopic's locations are not reasonable given the parent topic. For example, CATHY discovers *'supreme leader/army general sex/court/supreme court/egypts prosecutor general'* to be a subtopic of *'egypt/egypts morsi/egypt imf loan/egypts president/muslim brotherhood,'*. Resorting to the original network links to discover each topic's entity rankings results in the claim that the locations *'US/Sudan/Iran/Washington'* represent a subtopic of locations *'Egypt/Cairo/Tahrir Square/Port Said,'* which is not easily interpretable. Finally, $\text{NetClus}_{pattern}$ conflates 'Egypt' with several other topics, and the pattern representations can do little to improve the topic interpretability.

5 Conclusion

In this work, we address the problem of constructing a multi-typed topical hierarchy from heterogeneous information networks. We develop a novel clustering and ranking method which can be recursively applied to hierarchically discover multi-typed subtopics from heterogeneous network data. Our approach mines each discovered topic for topical patterns, yielding a comprehensive representation of each topic comprising lists of ranked patterns with different types (phrases, authors, etc.). Our experiments on the science and news domains demonstrate the significant advantage of our unified generative model for the task of hierarchical topic discovery, as compared to the state-of-the-art heterogeneous network analysis technique. We also show our constructed topical hierarchies have high quality based on human judgement.

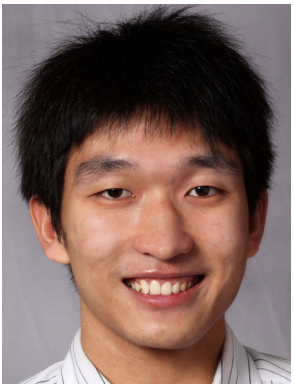
We hope to further improve our multi-typed topical hierarchy construction method to be able to accommodate user preference for the particular hierarchical organization of a dataset. We are also interested in constructing evolving topical hierarchies that would be able to work with the constantly changing information found in data streams.

Acknowledgments Research was sponsored in part by the Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), the Army Research Office under Cooperative Agreement No. W911NF-13-1-0193, National Science Foundation IIS-1017362, IIS-1320617, and IIS-1354329, DTRA, and MIAS, a DHS-IDS Center for Multimodal Information Access and Synthesis at UIUC. Chi Wang was supported by a Microsoft Research PhD Fellowship. Marina Danilevsky was supported by a National Science Foundation Graduate Research Fellowship Grant NSF DGE 07-15088.

References

1. Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res 3:993–1022
2. Chang J, Boyd-Graber J, Wang C, Gerrish S, Blei DM (2009) Reading tea leaves: how humans interpret topic models. NIPS
3. Chen X, Zhou M, Carin L (2012) The contextual focused topic model. In: KDD
4. Chuang SL, Chien LF (2004) A practical web-based approach to generating topic hierarchy for text segments. In: CIKM
5. Deng H, Han J, Zhao B, Yu Y, Lin CX (2011) Probabilistic topic models with biased propagation on heterogeneous information networks. In: KDD
6. Di Caro L, Candan KS, Sapino ML (2008) Using tagflake for condensing navigable tag hierarchies from tag clouds. In: KDD
7. Gauch S, Chaffee J, Pretschner A (2003) Ontology-based personalized search and browsing. Web Intell Agent Syst 1(3/4):219–234
8. Han J, Pei J, Yin Y, Mao R (2004) Mining frequent patterns without candidate generation: a frequent-pattern tree approach. Data Min Knowl Discov 8(1):53–87

9. Hofmann T (2001) Unsupervised learning by probabilistic latent semantic analysis. Mach Learn 42(1–2):177–196
10. Kim H, Sun Y, Hockenmaier J, Han J (2012) Etm: Entity topic models for mining documents associated with entities. In: ICDM
11. Lawrie D, Croft WB (2000) Discovering and comparing topic hierarchies. In: Proceedings of RIAO
12. Li Q, Ji H, Huang L (2013) Joint event extraction via structured prediction with global features. In: ACL
13. Liu X, Song Y, Liu S, Wang H (2012) Automatic taxonomy construction from keywords. In: KDD
14. Navigli R, Velardi P, Faralli S (2011) A graph-based algorithm for inducing lexical taxonomies from scratch. In: IJCAI
15. Newman D, Lau JH, Grieser K, Baldwin T (2010) Automatic evaluation of topic coherence. In: NAACL-HLT
16. Smyth P (2000) Model selection for probabilistic clustering using cross-validated likelihood. Stat Comput 10(1):63–72
17. Snow R, Jurafsky D, Ng AY (2004) Learning syntactic patterns for automatic hypernym discovery. NIPS
18. Sun Y, Han J, Gao J, Yu Y (2009a) itopicmodel: information network-integrated topic modeling. In: ICDM
19. Sun Y, Yu Y, Han J (2009b) Ranking-based clustering of heterogeneous information networks with star network schema. In: KDD
20. Tang J, Zhang M, Mei Q (2013) One theme in all views: modeling consensus topics in multiple contexts. In: KDD
21. Wang C, Danilevsky M, Desai N, Zhang Y, Nguyen P, Taula T, Han J (2013) A phrase mining framework for recursive construction of a topical hierarchy. In: KDD
22. Wong W, Liu W, Bennamoun M (2012) Ontology learning from text: a look back and into the future. ACM Comput Surv (CSUR) 44(4):20
23. Zavitsanos E, Paliouras G, Vouros GA, Petridis S (2007) Discovering subsumption hierarchies of ontology concepts from text corpora. In: Proceedings of IEEE/WIC/ACM international conference on web intelligence



Chi Wang is a researcher in Microsoft Research, Redmond, Washington. He received the Ph.D. degree in computer science from Univ. of Illinois at Urbana-Champaign in 2014. He graduated from Tsinghua Univ., China, in 2009. His research has been focused on data mining, information network analysis and text mining. He is the first winner of the prestige Microsoft Research Graduate Research Fellowship in the history of CS, UIUC.

Jialu Liu is a fourth-year Ph.D. student in the Department of Computer Science at UIUC, supervised by Prof. Jiawei Han. Before he joined UIUC, he received the B.S. degree from Department of Computer Science, Zhejiang University, China, in 2011. Currently his research includes data mining and machine learning, especially in unsupervised network/graph analysis and text mining.

Nihit Desai graduated from University of Illinois at Urbana-Champaign in 2013 with a B.S. in Computer Science. At UIUC, his undergraduate research activities focused on mining of Information Networks. Since 2013, Nihit has been working as a Software Engineer at LinkedIn as part of the Search Quality team.

Marina Danilevsky is a research scientist at IBM Almaden Research Center, San Jose, California. She received her Ph.D. degree in 2014, and M.S. degree in 2011, both from the Department of Computer Science, University of Illinois Urbana-Champaign. She received her B.S. degree in mathematics in 2007 from the University of Chicago. Her research interests include data mining, text mining, and information network analysis.

Jiawei Han is Abel Bliss Professor in the Department of Computer Science at the University of Illinois. He has been researching into data mining, information network analysis, and database systems, with over 600 publications. He served as the founding Editor-in-Chief of ACM Transactions on Knowledge Discovery from Data (TKDD). Jiawei has received ACM SIGKDD Innovation Award (2004), IEEE Computer Society Technical Achievement Award (2005), IEEE Computer Society W. Wallace McDowell Award (2009), and Daniel C. Drucker Eminent Faculty Award at UIUC (2011). He is a Fellow of ACM and a Fellow of IEEE. He is currently the Director of Information Network Academic Research Center (INARC) supported by the Network Science-Collaborative Technology Alliance (NS-CTA) program of U.S. Army Research Lab. His co-authored textbook "Data Mining: Concepts and Techniques" (Morgan Kaufmann) has been adopted worldwide.