LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-CONF-422685

HCDF: A Hybrid Community Discovery Framework

K. Henderson, T. Eliassi-Rad, S. Papadimitriou, C. Faloutsos

January 21, 2010

2010 SIAM Conference on Data Mining (SDM10)
Columbus, OH, United States
April 29, 2010 through May 1, 2010

Disclaimer

HCDF: A Hybrid Community Discovery Framework

Keith Henderson* Tina Eliassi-Rad* Spiros Papadimitriou† Christos Faloutsos‡
*Lawrence Livermore National Laboratory †IBM T.J. Watson ‡Carnegie Mellon University
*{keith, eliassi}@llnl.gov †spapadim@us.ibm.com ‡christos@cs.cmu.edu

Abstract

We introduce a novel Bayesian framework for hybrid community discovery in graphs. Our framework, $\mathcal{HCDF}$ (short for *Hybrid Community Discovery Framework*), can effectively incorporate hints from a number of other community detection algorithms and produce results that outperform the constituent parts. We describe two $\mathcal{HCDF}$-based approaches which are: (1) effective, in terms of link prediction performance and robustness to small perturbations in network structure; (2) consistent, in terms of effectiveness across various application domains; (3) scalable to very large graphs; and (4) nonparametric. Our extensive evaluation on a collection of diverse and large real-world graphs, with millions of links, show that our $\mathcal{HCDF}$-based approaches (a) achieve up to 0.22 improvement in link prediction performance as measured by area under ROC curve (AUC), (b) never have an AUC that drops below 0.91 in the worst case, and (c) find communities that are robust to small perturbations of the network structure as defined by Variation of Information (an entropy-based distance metric).

1 Introduction

Graph analysis methods have recently attracted significant interest. In this paper, we focus on the problem of discovering community structures in large, real-world graphs. In this context, three desirable properties are:

(P1) *Effective:* The global connectivity patterns of the network are successfully factored into communities, which are highly predictive of individual links and robust to small perturbations in network structure.
(P2) *Scalable:* Time and space complexity are strictly sub-quadratic w.r.t. the number of nodes, scaling up to graphs with millions of links.
(P3) *Nonparametric:* The number of communities need not be specified *a priori*, instead it is determined from the data itself.

Graphs have attracted interest because they arise naturally in many different settings, and across a very diverse set of domains, including bibliographic analysis, social networks, the World-Wide Web, computer networks, product recommendations, and so on. Consequently, a fourth property is also desirable:

(P4) *Consistent:* The effectiveness of the discovered community structure, as defined in (P1), is consistently high, across a wide range of data sets.

Despite the recent interest, approaches that possess **all** of the above properties, (P1) through (P4), are rare. Here, we introduce two hybrid approaches that exhibit all of them. Our approaches are based on a novel Bayesian framework, $\mathcal{HCDF}$ (for *Hybrid Community Discovery Framework*), which can successfully incorporate communities discovered via other, non-Bayesian approaches as *hints* that lead to improved effectiveness of results and consistency across various domains.

Why is a hybrid community discovery a good idea? Intuitively, soft clustering approaches with mixed-membership (most of them Bayesian) have more leeway to explain a node's links, compared to hard clustering approaches (most of them non-Bayesian). On one hand, if a node's links cannot be explained by a single membership, soft clustering has an advantage. On the other hand, if a node's links can be explained almost equally well by a number of single and mixed memberships, hard clustering may make a simpler assignment. Therefore, combining these two clustering approaches can potentially lead to improved community factorization.

However, designing a framework that can successfully combine different approaches is not a trivial task. We extensively studied three orthogonal aspects, which we summarize next.

Core Bayesian model. This serves as the foundation, upon which $\mathcal{HCDF}$ is built. We chose *Latent Dirichlet Allocation on Graphs* (*LDA-G*) [12] as the core Bayesian method for community detection, because it is simple and satisfies properties (P1) to (P3).

Hint sources. Among known methods, we examined two that also satisfy properties (P1) to (P3). The first is *Fast Modularity* (*FM*) [5, 18], which is a spectral partitioning method. The second is *Cross-Associations* (*XA*) [3], which is an MDL-based approach. Both methods produce hard clusters (also referred to as

communities or groups).

Coalescing strategies. We explored three options for incorporating hints into the core Bayesian model: *seed* which uses hints only as an initial configuration for the inference procedure; *prior* which propagates hints from one configuration to the next; and *attribute* which incorporates hints as additional link-attributes.

To quantitatively measure the effectiveness of the discovered community structure, we use link prediction performance and robustness to small perturbations in the network structure. *Why is link prediction a good measure?* A factorization of the graph's connectivity structure into a number of communities is good, if it can be used to predict the presence or absence of links between a pair of nodes based on their respective communities. Therefore, we evaluate effectiveness by randomly holding out a number of links, building the model under evaluation, and then trying to predict the held-out links. We use area under the ROC curve (AUC), as an accurate measure of performance.[1] *Why is robustness to small perturbations in the network structure a good measure?* As Karrer *et al.* argue, community structures that are "significant / believable" should be able to withstand small perturbations in the network structure [13]. We utilize their quantification of robustness, measured by Variation of Information Δ: an entropy-based distance metric that measures the distance between two clusterings – one on the original network and one on the perturbed network.

We describe an extensive evaluation on several real data sets from a diverse range of domains (see Section 4). Our methods, *HCD-X* (which utilizes hints from Cross-Associations as link-attributes) and *HCD-M* (which utilizes hints from Fast Modularity as link-attributes), achieve significant improvements in effectiveness across the board, while maintaining scalability. We observe up to 0.22 improvement in AUC scores for link prediction. The average AUC scores for *HCD-M* and *HCD-X* (across nine data sets and five trial runs) are 0.95 and 0.96, respectively. Furthermore, the AUC of our hybrid methods never drops below 0.91 in the worst case.

With respect to robustness to small perturbations in the network structure, both of our hybrid methods maintain low values for Variation of Information, $\Delta \in [0, 1]$. For example, with 10% of the links randomly rewired (while maintaining the same degree distribution), our hybrid methods have on average Δ of 0.19 (indicating that "significant / believable" communities were found).

Summarizing, the main contributions of the paper are:

- We propose a generic Bayesian framework, $\mathcal{HCDF}$, for hybrid community discovery; and present three ways for coalescing hints from various community discovery algorithms.
- We propose two novel solutions, *HCD-X* and *HCD-M*, for the task of community discovery that are (1) effective in terms of link prediction and robustness to small perturbations in network structure, (2) scalable, (3) nonparametric, and (4) consistent across various application domains.
- We present results from an extensive evaluation of *HCD-X* and *HCD-M* on several large real-world graphs, with millions of links. Our results demonstrate that *HCD-X* and *HCD-M* are highly effective and consistent across different application domains

The rest of the paper is organized as follows. Next, we review the related work. We present our proposed method in Section 3, followed by experimental results and discussion. We conclude the paper in Section 5.

2 Related work

Bayesian Approaches to Community Discovery in Graphs. There are only a handful of scalable Bayesian approaches to community discovery in graphs [26, 23, 22, 15, 12], most extend *Latent Dirichlet Allocation* (LDA) [2]. LDA is a latent variable model for topic modeling. *SSN-LDA* [23] and LDA-G [12] are the simplest adaptions of LDA for community discovery in graphs. *GWN-LDA* [22] introduces a Gaussian distribution with inverse-Wishart prior on a LDA-based model to find communities in social networks with weighted links. LDA-based models have also been used to find communities in textual attributes and relations [24, 15]. For simplicity's sake, we chose *LDA-G* for the Bayesian constituent of *HCD-X* and *HCD-M*. We could have easily chosen one of the other scalable Bayesian approaches. The effectiveness and consistency of these non-hybrid Bayesian models to community discovery in a diverse collection of real-world graphs is unknown.

In [15], the authors propose a multi-step approach involving LDA to cluster large document collections containing both text and relations. In [16, 10], the authors propose a two-step approach to find clusters in data containing attributes and relations (namely, citation graphs and a gene-interaction network). The strategies used to incorporate information from one step to the next is different in $\mathcal{HCDF}$ than in these methods.

[1] AUC is equal to the probability of a model ranking a randomly chosen positive instance higher than a randomly chosen negative instance. The default value for AUC is 0.5. In our case, a positive instance is a nonzero entry in the adjacency matrix versus a negative instance is a zero entry.

Also, it is not clear how these methods would perform across a diverse collection of real-world graphs (besides citation networks and biological networks).

Non-Bayesian Approaches to Community Discovery in Graphs. There are numerous non-Bayesian methods for community detection, including the popular METIS [14], spectral methods [4] maximum flow methods [9], co-clustering [7], multi-level clustering (Graclus) [6] and others [25]. All of the above methods do not satisfy the nonparametric property of the problem statement because they require the user to specify the number of clusters k, or some other, related threshold.

Cross-Associations (*XA*) [3] is one of the few methods that automatically determines the number of clusters, as the one that minimizes the description length of the whole graph. Graphically, *XA* rearranges the rows and columns of the adjacency matrix such that each row-group/column-group intersection is as dense or as sparse as possible. The runtime complexity of *XA* for a graph $G = [V, E]$ is $O(|E|)$, which is usually much smaller than $O(|V|^2)$.

Extensions of *XA* that detect communities in time-evolving graphs are: *timeFalls* [8] and *GraphScope* [20]. They determine communities, as well as merge and split of them, as time grows. Although scalable and effective, *XA* can not handle hints, neither in its base form, nor in its extensions above.

Another popular method, that also does not require the number of communities *a priori*, is based on the concept of "modularity" [18], a spectral partitioning method. However, its original version has prohibitive complexity. Its fast version (*FM*) [5] addresses the runtime complexity, with $O(|V| \cdot \log^2(|V|))$, though it only operates on the largest connected component of a graph and not the entire graph. None of the modularity approaches are able to handle hints.

Different Notions of a Good Community. *FM*'s notion of a good community is one in which the density of intra-community linkage is more than inter-community linkage as compared to the expected number. *XA*'s notion of a good community is based on minimizing the total encoding cost, where community members tend to have the same set of neighboring nodes and may not be linked to each other at all. *LDA-G*'s notion of a good community is similar to *XA*'s except that community structure is found by maximizing likelihood and community members tend to have the same set of neighboring nodes in similar proportions (i.e. it produces soft memberships in communities).

3 Proposed Method

Table 1 lists the notations used in this paper. We first briefly describe *LDA-G*, which is core Bayesian model in our proposed $\mathcal{HCDF}$, as well as an extension to *LDA-G*, which handles hints as additional observed attributes. Then, we provide details on $\mathcal{HCDF}$.

3.1 Preliminaries *LDA-G* [12] is an extension of Latent Dirichlet Allocation (LDA) [2] for use in graphs rather than text corpora. *LDA-G* models each source node in the graph as a multinomial distribution over some set of communities Z. The cardinality of Z is unknown *a priori* and is learned during inference (by adopting a Dirichlet prior). In *LDA-G*, each source node generates a series of communities from its multinomial; and each community is a multinomial distribution over target nodes. Any time a community is generated by a source node, that community generates a target node from its distribution. The distributions over source-node to community and community to target-node are learned using MCMC techniques (most commonly Gibbs sampling) [11]. To simplify inference, it is assumed that the roles of a node as a source-node and as a target-node are probabilistically independent. The generative model for *LDA-G* is as follows:

$$v_i | z_i, \phi^{(z_i)} \sim Discrete(\phi^{(z_i)}) \tag{3.1}$$

$$\phi \sim Dirichlet(\beta) \tag{3.2}$$

$$z_i | \theta^{u_i} \sim Discrete(\theta^{u_i}) \tag{3.3}$$

$$\theta \sim Dirichlet(\alpha) \tag{3.4}$$

Then, *LDA-G*'s update equation is:

$$p(z_i = k | \mathbf{z_{-i}}, \mathbf{v}) \propto \frac{n_u^k + \alpha}{n_u + \alpha K} \cdot \frac{n_k^v + \beta}{n_k + \beta N} \tag{3.5}$$

Unlike most approaches to community discovery, *LDA-G* only requires present links (i.e., non-zero entries in the adjacency matrix). This property helps its runtime and space complexities. Its runtime complexity is $O(NKM)$; its space complexity is $O(N(K + M))$. Since $K \ll N$ and $M \sim \log(N)$, LDA-G runs in $O(N \cdot \log(N)) \approx O(|E|)$.

3.2 Extending *LDA-G* to Handle Node-Attributes For use in $\mathcal{HCDF}$, we extend *LDA-G* to graphs with attributes by augmenting communities with a new collection of multinomial distributions, one per link-attribute. Each node-attribute generates two link-attributes since source and target nodes are treated

V	set of nodes	u_i	i^{th} source node (analogous to a document in LDA)
E	set of links	v_i	i^{th} target node (analogous to a word in LDA)
N	number of nodes $= \|V\|$	z_i	i^{th} community (analogous to a topic in LDA)
M	average node degree	ζ_i	community for graph element i
Z	set of communities	ζ_i^r	row-group for graph element i
K	number of communities $= \|Z\|$	ζ_i^c	column-group for graph element i
R	set of attributes to be added to the graph as hints	a_{ij}	j^{th} attribute value on link i
A	number of attributes $= \|R\|$	A_i	maximum number of possible values for attribute i
n_u^k	count of community k in source-node u	θ	Dirichlet prior on source-to-community distributions
n_u	$\sum_{i=1}^{K} n_u^k$	α	hyperparameter for Dirichlet on source-to-community
n_k^v	count of target-node v in community k	ϕ	Dirichlet prior on community-to-target distributions
n_k	$\sum_{i=1}^{N} n_k^v$	β	hyperparameter for Dirichlet on community-to-target
$n_k^{a_i}$	count of attribute a_i in community k	ρ	Dirichlet prior on attribute distributions
		γ	hyperparameter for Dirichlet on attributes

Table 1: Notations used in the paper.

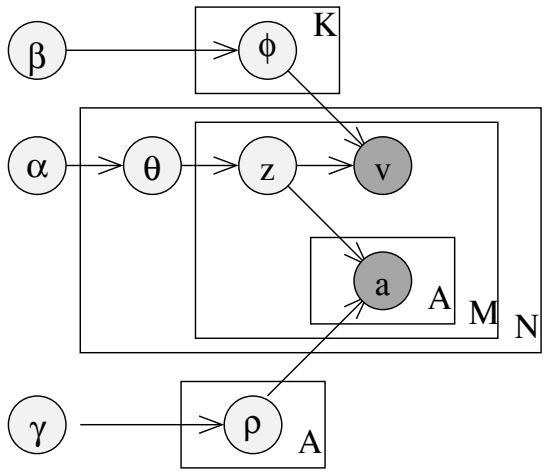

Figure 1: Graphical model for LDA-G with Attributes. The observables are the target nodes v and the attributes a. The latent variable are the communities z.

separately and independently. The generative model for *LDA-G* with link-attributes adds the following generators to the aforementioned *LDA-G* model:

$$a_{ij}|z_i, \rho_j^{(z_i)} \sim Discrete(\rho_j^{(z_i)}) \tag{3.6}$$

$$\rho \sim Dirichlet(\gamma) \tag{3.7}$$

Figure 1 depicts the graphical model for *LDA-G* with attributes. This model is appropriate for categorical attributes. However, attributes can be binary-valued or real-valued attributes, so long as they are given the appropriate priors. We use Gibbs sampling [11] for the inference procedure. The runtime complexity of *LDA-G* on graphs with attributes is $O(NKMA) \approx O(E)$. The update equation for *LDA-G* on graphs with attributes is:

$$p(z_i = k|\mathbf{z_{-i}}, \mathbf{u}, \mathbf{v}, \mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_A}) \propto$$

$$\frac{n_u^k + \alpha}{n_u + \alpha K} \cdot \frac{n_k^v + \beta}{n_k + \beta N} \prod_{i=1}^{A} \frac{n_k^{a_i} + \gamma}{n_k + A_i\gamma} \tag{3.8}$$

3.3 Hybrid Community Detection Framework ($\mathcal{HCDF}$)

$\mathcal{HCDF}$ takes any pair of ⟨non-Bayesian, Bayesian⟩ community discovery algorithms and a coalescing strategy, and produces an algorithm for finding community structure in graphs.

3.3.1 $\mathcal{HCDF}$($algorithm_1$, ***LDA-G***, **ATTR**)

$\mathcal{HCDF}$'s coalescing strategy, **ATTRIBUTE** or **ATTR** for short, incorporates the communities found by $algorithm_1$ with *LDA-G* by treating them as attributes. In other words, the group assignments of $algorithm_1$ became attributes of the input graph. Then, *LDA-G* learns a model of the community structure given both the network structure and these attributes.

To make our discussion more concrete here, assume we have chosen *XA* for $algorithm_1$. Algorithm 1

presents our *HCD-X* algorithm (an instantiation of $\mathcal{HCDF}$). Given a graph $G = [V, E]$, *HCD-X* runs *XA* and treats its row- and column-groups as link-attributes. Then, *HCD-X* runs *LDA-G* on the attributed graph (i.e., G with these link-attributes).

HCD-X treats each node's row-group (i.e. cluster label) as one categorical attribute and its column-group as another categorical attribute (rather than treating each row-group/column-group intersection as a single attribute). In graphs with non-symmetric adjacency matrices, these row- and column-groups are often different.

In Algorithm 1, *XA* can be replaced with any other scalable community detection algorithm. Specifically, when we replace *XA* with *FM*, we refer to it as *HCD-M*.

Algorithm 1 $\mathcal{HCDF}$(*XA*, *LDA-G*, **ATTR**) = *HCD-X*

Require: Graph: $G = [V, E]$
  /* Run Cross-Association on G */
  $[rowGroups, colGroups] = XA(G)$
  $R = [rowGroups, colGroups]$
Ensure: $\forall i \in [1, A]$ and $\forall < u, v > \in E$: $R \equiv \{a_i^{<u,v>}\}$
  /* Run *LDA-G* on graph G with attributes R and hyperparameters $\gamma \approx 10$ and $\alpha = \beta \approx 1$ */
  /* Set the prior */
  $a_{i1} \leftarrow a_i^{<u,\cdot>}$; $a_{i2} \leftarrow a_i^{<\cdot,v>}$
  Initialize all count variables n_u, n_u^k, n_k, n_k^v, $n_k^{a_{i1}}$, $n_k^{a_{i2}}$ to 0
  for each link $< u, v > \in E$ **do**
    Sample community $z_i = k$ using Equation 3.8
    Increment count variables n_u, n_u^k, n_k, n_k^v, $n_k^{a_{i1}}$, $n_k^{a_{i2}}$ by 1
  end for
  /* Run Gibbs sampling */
  for each edge $< u, v > \in E$ with in community k **do**
    Decrement count variables n_u, n_u^k, n_k, n_k^v, $n_k^{a_{i1}}$, $n_k^{a_{i2}}$ by 1
    Sample community $z_i = k$ using Equation 3.8
    Increment count variables n_u, n_u^k, n_k, n_k^v, $n_k^{a_{i1}}$, $n_k^{a_{i2}}$ by 1
  end for

3.3.2 $\mathcal{HCDF}$(*algorithm*$_1$, ***LDA-G***, **SEED**)

$\mathcal{HCDF}$'s coalescing strategy, **SEED**, is the simplest way of fusing the output of a given community discovery algorithm, *algorithm*$_1$ with *LDA-G*. $\mathcal{HCDF}$(*algorithm*$_1$, *LDA-G*, **SEED**) merely uses the communities from *algorithm*$_1$ to set the initial configuration for *LDA-G*, then discards this information (i.e., forgets the hints) in subsequent sampling. Algorithm 2 outlines $\mathcal{HCDF}$(*algorithm*$_1$, *LDA-G*, **SEED**), where Gibbs sampling is used to conduct inference in *LDA-G* and *algorithm*$_1$'s communities are used to initialize the count variables.

Algorithm 2 $\mathcal{HCDF}$(*algorithm*$_1$, *LDA-G*, **SEED**)

Require: Graph: $G = (V, E)$
  /* Run community discovery algorithm *algorithm*$_1$ on G */
  $hint = algorithm_1(G)$
Ensure: $hint \equiv [s_u, s_u^k, s_k, s_k^v]$
  /* Run *LDA-G* on graph G */
  /* Set the prior */
  /* Initialize all count variables */
  $n_u \leftarrow s_u$; $n_u^k \leftarrow s_u^k$; $n_k \leftarrow s_k$; $n_k^v \leftarrow s_k^v$
  /* Select initial configuration */
  for each edge $< u, v > \in E$ **do**
    Sample community $z_i = k$ using Equation 3.5
    Increment count variables n_u, n_u^k, n_k, n_k^v by 1
  end for
  /* Forget the hint */
  $n_u \leftarrow n_u - s_u$; $n_u^k \leftarrow n_u^k - s_u^k$; $n_k \leftarrow n_k - s_k$; $n_k^v \leftarrow n_k^v - s_k^v$
  /* Run Gibbs sampling */
  for each edge $< u, v > \in E$ with in community k **do**
    Decrement count variables n_u, n_u^k, n_k, n_k^v by 1
    Sample community $z_i = k$ using Equation 3.5
    Increment count variables n_u, n_u^k, n_k, n_k^v by 1
  end for

3.3.3 $\mathcal{HCDF}$(*algorithm*$_1$, ***LDA-G***, **PRIOR**)

$\mathcal{HCDF}$'s coalescing strategy, **PRIOR**, is similar to **SEED**, but it does not discard *algorithm*$_1$'s hints (i.e., community information) after choosing an initial configuration. Instead, the multinomial distributions for each source-node and community are comprised of (1) the empirical prior imparted by *algorithm*$_1$'s results and (2) the learned distribution from *LDA-G*'s inference. There is no speed-penalty for retaining the empirical prior because the Gibbs sampler requires only the n_u^k and n_k^v counts (which refer to the count of community k in source-node u and the count of target-node v in community k, respectively). Algorithm 3 presents $\mathcal{HCDF}$(*algorithm*$_1$, *LDA-G*, **PRIOR**).

4 Experiments

This section presents an empirical comparison of link-prediction performance and community robustness to small perturbation in network structure. Link prediction demonstrates a community structure's ability to model the underlying connectivity of the graph. Community robustness to small perturbations in the network structure measures the "significant / believability" of the discovered community structure. Specifically, if a

Algorithm 3 $\mathcal{HCDF}(algorithm_1$, *LDA-G*, **PRIOR**)

Require: Graph: $G = (V, E)$
/* Run community discovery algorithm $algorithm_1$ on G */
$hint = algorithm_1(G)$
Ensure: $hint \equiv [s_u, s_u^k, s_k, s_k^v]$
/* Run *LDA-G* on graph G */
/* Set the prior */
/* Initialize all count variables */
$n_u \leftarrow s_u$; $n_u^k \leftarrow s_u^k$; $n_k \leftarrow s_k$; $n_k^v \leftarrow s_k^v$
/* Select initial configuration */
for each edge $< u, v > \in E$ **do**
  Sample community $z_i = k$ using Equation 3.5
  Increment count variables n_u, n_u^k, n_k, n_k^v by 1
end for
/* Run Gibbs sampling */
for each edge $< u, v > \in E$ with in community k **do**
  Decrement count variables n_u, n_u^k, n_k, n_k^v by 1
  Sample community $z_i = k$ using Equation 3.5
  Increment count variables n_u, n_u^k, n_k, n_k^v by 1
end for

community structure is significant, then it should be able to withstand small perturbations to network structure [13].

4.1 Experimental Setup

4.1.1 Data Sets and Algorithms Tables 2 and 3 summarize the graphs used in our experiments. **Internet Topology Data:** Autonomous Systems (*AS*) Graph[2] is an AS-level connectivity graph collected on May 26, 2001. It includes Oregon route-views, Looking glass data, and Routing registry data. **IP Traffic Data:** (*IP1* through *IP5*) are composed of IP traffic[3] collected at the perimeter of an enterprise network over five days in 2007. **PubMed Data:** AxK and AxA are collections of data from the *PubMed* database.[4] AxK is a bipartite *Author* × *Knowledge* graph, where author nodes connect to knowledge-theme[5] nodes. A link exists from an author u to a knowledge-theme k for every article in which u is a coauthor and k is a theme appearing in the abstract. AxA is a coauthorship graph. It is composed of 4555 connected components. The largest connected component has 8763 authors (approximately 24% of the entire graph). **WWW Data:** This graph was originally created to measure the diameter of the Web [1]. It was collected by a Web crawler that started from a *nd.edu* site.[6]

Table 4 outlines the different algorithms used in our experiments. Details of each method were provided in Sections 2 and 3.

4.1.2 Experimental Methodology for Link Prediction Our experimental methodology for measuring link-prediction performance on a single graph is a three-step process. First, we randomly select a subset of links from a graph. When selecting links, we pick both present and absent links. A present link has a non-zero entry in the graph's adjacency matrix. An absent link has a zero entry in the graph's adjacency matrix. The selected links comprise the held-out test set. Second, each algorithm is given the remaining graph in which to discover communities.[7] Third, each algorithm uses its discovered communities to estimate the probability that a link in the held-out test set was a present link. These estimates are then used to calculate the area under the ROC curve (AUC).

For each data set, we ran five trials. In each trial, the held-out test-set contains 1000 randomly chosen links, with 500 present links and 500 are absent links. Across the different methods (*XA*, *LDA-G*, *HCD-X*, etc), the held-out test sets are the same for each graph. The reported AUC-scores are averages of the AUC-scores over the five trials.

Using Communities to Predict Links To predict a link between source-node u and target-node v in *LDA-G*, *HCD-X*-**SEED**, and *HCD–X*-**PRIOR**, we use the following formula:

$$score_{LDA-G}(< u, v >) = \sum_{k \in K} p(\zeta_{<u,v>} = k | Z, E) \tag{4.9}$$

Recall that K is the number of communities discovered, Z is the discovered communities and E is the graph's links. $p(\zeta_{<u,v>} = k|Z, E)$ is the same as *LDA-G*'s update equation (see Equation 3.5).

To predict a link between source-node u and target-node v in *HCD-X* and *HCD-M*, we use the following formula:

[2]Available at http://topology.eecs.umich.edu/data.html.

[3]This data is proprietary.

[4]PubMed is a repository containing millions of citations from biomedical articles (see http://www.pubmedcentral.nih.gov/).

[5]Knowledge themes were extracted based on term frequency in PubMed abstracts.

[6]Available at http://www.nd.edu/ networks/resources.htm.

[7]*LDA-G* handles the held-out links as "unknown" observations. However, *XA* and *FM* treat them as absent links (with zero entries in the adjacency matrix).

Real-World Graphs	Acronym	$\mid V \mid$	$\mid E \mid$	# Components	% of LCC in V
Autonomous Systems	AS	11,461	32,730	1	1
Day 1: IP × IP	IP1	34,449	303,175	4	99.98%
Day 2: IP × IP	IP2	33,732	320,754	8	99.96%
Day 3: IP × IP	IP3	34,661	428,596	2	99.99%
Day 4: IP × IP	IP4	34,730	425,368	2	99.99%
Day 5: IP × IP	IP5	33,981	112,271	13	99.92%
PubMed Author × Knowledge	AxK	37,346 (A) & 117 (K)	119,443	1	1
PubMed Coauthorship	AxA	37,225	143,364	4,556	23.54%
WWW Graph	WWW	325,729	1,497,135	1	1

Table 2: Summary of real-world graphs used in experiments. LCC refers to the Largest Connected Component.

Data Graph	Maximum Edge Weight	Average Degree	Clustering Coefficient	Average Path	Diameter	# of Articulation Points	% of Articulation Points in V
AS	2,432	2.86	0.258	3.67	11	828	7.2%
IP1	19,623	8.80	0.198	3.23	7	1,258	3.7%
IP2	23,130	9.51	0.18	3.22	8	1,208	3.6%
IP3	23,428	12.37	0.198	3.04	6	920	2.7%
IP4	22,454	12.25	0.216	3.07	7	841	2.4%
IP5	12,923	3.30	0.058	3.54	7	1,524	4.5%
AxK	5,366	3.19	0	1.00	1	54	0.1%
AxA	178	3.85	0.49	8.85	23	1,467	3.9%
WWW	10,721	4.60	0.28	11.38	58	21,780	6.7%

Table 3: Some characteristics of real-world graphs used in experiments. An articulation point is a node such that its removal increases the number of connected components.

Algorithms Tested	Acronym
Cross-Association	*XA*
Fast Modularity	*FM*
Latent Dirichlet Allocation on Graphs	*LDA-G*
$\mathcal{HCDF}$(*XA*, *LDA-G*, **ATTRIBUTE**)	*HCD-X*
$\mathcal{HCDF}$(*FM*, *LDA-G*, **ATTRIBUTE**)	*HCD-M*
$\mathcal{HCDF}$(*XA*, *LDA-G*, **SEED**)	*HCD-X*-**SEED**
$\mathcal{HCDF}$(*XA*, *LDA-G*, **PRIOR**)	*HCD-X*-**PRIOR**

Table 4: List of methods used in experiments.

(4.10)
$$score_{HCD}(< u, v >) = \sum_{k \in K} p(\zeta_{<u,v>} = k | Z, E, R)$$

Recall that R is the attribute-set. $p(\zeta_{<u,v>} = k|Z, E, R)$ is the same as the update equation for *LDA-G* on graphs with attributes (see Equation 3.8).

For *XA* and *FM*, we use density to predict links. Specifically, we use the following equations to predict a link between source-node u and target-node v:

(4.11)
$$score_{FM}(u, v) = \frac{\mid \{\forall < i, j > \in E : (\zeta_i \equiv \zeta_u) \& (\zeta_j \equiv \zeta_v)\} \mid}{\mid \zeta_u \mid \cdot \mid \zeta_v \mid}$$

where ζ_i returns the *FM*-assigned community for node i.

(4.12)
$$score_{XA}(u, v) = \frac{\mid \{\forall < i, j > \in E : (\zeta_i^r \equiv \zeta_u^r) \& (\zeta_j^c \equiv \zeta_v^c)\} \mid}{\mid \zeta_u^r \mid \cdot \mid \zeta_v^c \mid}$$

where ζ_i^r is the row-group assigned to node i and ζ_i^c is the column-group assigned to node i.

4.1.3 Experimental Methodology for Community Robustness We adapt the experimental methodology described in [13] to measure community robustness. Specifically, we perturb a data graph by randomly

reassigning a number of its links. The rewiring parameter $c \in [0, 1]$ determines the fraction of links rewired. Links are reassigned in a way that preserves the expected degree of each node in the graph. Then, a community detection algorithm is applied to the original and perturbed graphs, and the *Variation of Information* [13], Δ, is calculated between the two community assignments C (with $c = 0$) and C' (with $c \in [0, 1]$):

$$\Delta(C, C') = H(C|C') + H(C'|C) \tag{4.13}$$

H is the entropy function; thus, $H(C'|C)$ measures the information needed to describe C' given C). Δ treats each assignment as a message; it is a symmetric entropy-based measure of the distance between these messages.

Δ is bounded between 0 and $\log(N)$ for hard clustering assignments, but in the case of mixed-membership models the measure can grow without bound. For consistency across data sets, we generate hard clusters for all algorithms and normalize by reporting $\Delta/\log(N)$. For *LDA-G*, *HCD-X*, and *HCD-M*, we assign each node to the group that contains a plurality of its outgoing links in the Maximum Likelihood (ML) configuration. Ties are broken by selecting the largest group (i.e., the group that appears most often in the ML configuration). For *XA*, we only consider the row assignments and discard column assignments.

Since *FM* only operates on the largest connected component (LCC) of a graph, we extract the LCC of the original graph before perturbing the network for *FM* experiments. Thus, only the LCC graph, G', is perturbed, and Δ values are calculated over it. For *HCD-M*, we remove G' from the original graph and replace it with the perturbed versions of G'. Except for the Coauthorship graph, the remaining graphs used in our experiments have LCCs that contain over 99.99% of the nodes in the graph. The Coauthorship graph's LCC covers only 24% of the nodes in the graph.

We apply this method to each data set and each community discovery algorithm, considering various values of the rewiring probability c.

4.2 Results on Link Prediction Figure 2 shows performance of the methods listed in Table 4 on the IPxIP graphs. *HCD-X* (black line) and *HCD-M* (red line) are the only two methods that consistently produce high AUC results (> 0.93) on link prediction. *HCD-X* performs better than *HCD-M* on these data sets since *XA* produces better hints than *FM* here. We have included error bars on the *LDA-G* link-prediction curve, which shows that we would not reach *HCD-X*'s high results even if we ran *LDA-G* multiple times.

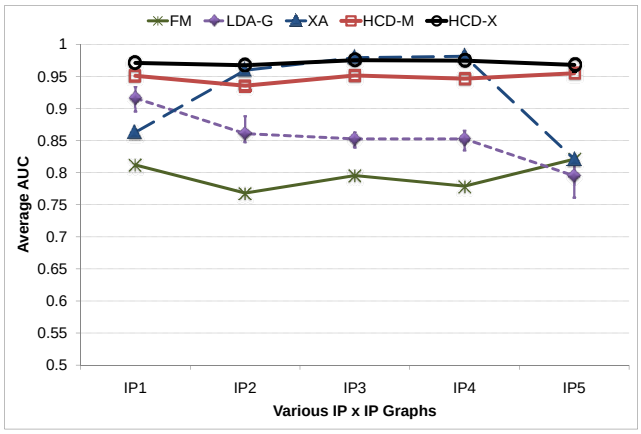

Figure 2: Link prediction results on various IPxIP graphs

Figure 3 presents performance of methods listed in Table 4 on the PubMed graphs, the Web graph, and the Internet AS graph. Again, *HCD-X* and *HCD-M* are the only two robust methods across these diverse data sets. They produce community structures that consistently predict links with high AUC results (> 0.91). The same cannot be said for the other methods.

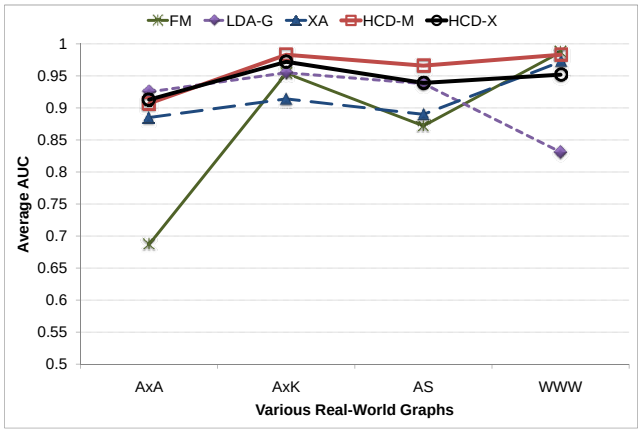

Figure 3: Link prediction results on various graphs from PubMed, WWW, and Internet.

Figure 4 depicts the effectiveness of different coalescing strategies in *HCD-X*. As expected, direct incorporation of *XA*-provided hints as link-attributes provides the best link-prediction performance. The next best coalescing strategy is **PRIOR** since it propagates the hints from one configuration to the next. **SEED** is the simplest and weakest coalescing strategy since it forgets the hints after setting the initial configuration.

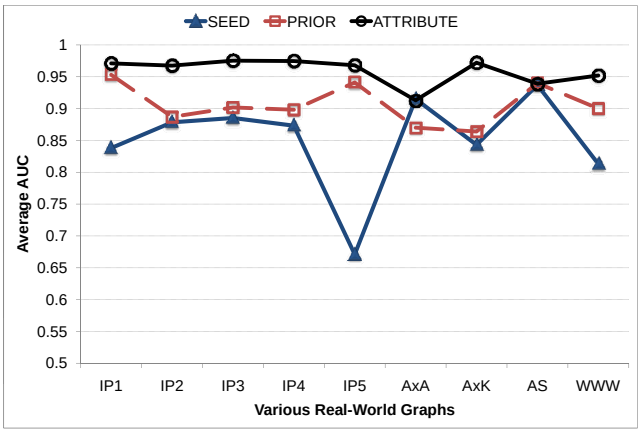

Figure 4: Effectiveness of different coalescing strategies. Results show average AUC on link prediction with $\mathcal{HCDF}(XA, LDA\text{-}G, strategy)$, where strategy is one of **SEED**, **PRIOR**, and **ATTRIBUTE**.

Figure 5 shows improvements in the average AUC scores (w.r.t. link prediction) when comparing *HCD-X* to *LDA-G*, *HCD-X* to *XA*, *HCD-M* to *LDA-G*, and *HCD-M* to *FM* across our diverse collection of real graphs. On average, the hybrid methods offer significant improvements, an increase of up to 0.22 in average AUC. There are rare cases in which *HCD-X* and *HCD-M* do not perform as well as their constituents. These cases arise when the constituent providing the hints generate communities that are really bad at predicting link structure (e.g., *FM*'s average AUC on PubMed AxA is 0.63).

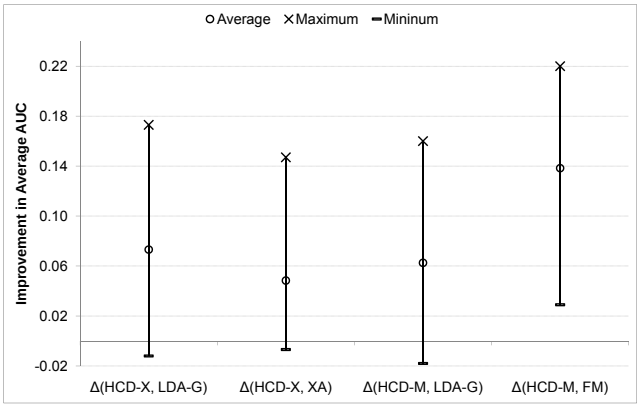

Figure 5: Typical link prediction performance improvement for *HCD-X* and *HCD-M* over their constituents. $\Delta(M_1, M_2) = AverageAUC(M_1) - AverageAUC(M_2))$.

Figure 6 depicts the worst-case average AUC across the methods listed in Table 4. The hybrid methods (*HCD-X* and *HCD-M*) have worst-case average AUC-scores above 0.9. The non-hybrid methods (*FM*, *LDA-G*, and *XA*) do not show such consistent effectiveness.

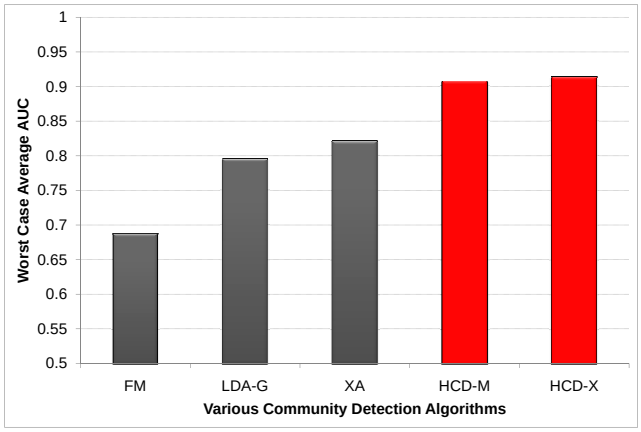

Figure 6: Worse-case performance across all graphs.

4.3 Results on Community Robustness Figure 7 depicts the Variation of Information, Δ, for the PubMed Author×Knowledge graph as we vary the rewiring probability. A lower value for Δ indicates a more robust community structure. *HCD-X* has a lower Δ than its constituent parts: *LDA-G* and *XA*. *HCD-M* has a lower Δ than *LDA-G* and a comparable one to *FM*.

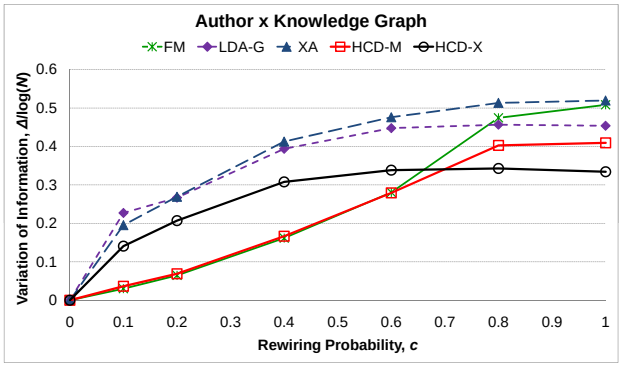

Figure 7: Robustness across various algorithms and rewiring probabilities for the PubMed Author×Knowledge graph.

A community discovery algorithm can artificially produce a low Δ by finding a trivial community structure (e.g., by placing all nodes in one community). To guard against this, we also need to look at other prop-

erties of the community structure such as the number of communities and the largest community fraction. In the Author×Knowledge graph, all five methods find on average tens of communities and the average largest community fraction is less than 31%. So, the lower Δ results here are not artificial.

Figure 8 shows Δ at rewiring probability $c = 0.2$ for the IP graphs and for the communities discovered on them by *LDA-G*, *FM*, and *HCD-M*. Points to note here are (1) *HCD-M* has much lower Δ values than *LDA-G*; (2) *HCD-M* has comparable Δ values to *FM*, but it has a higher AUC (on average +0.15) and more consistent link prediction performance than *FM* (as previously shown in Figure 2). As we describe in the next section, *XA* produces artificially low Δ values; therefore, we exclude *XA* and *HCD-X* from Figure 8.

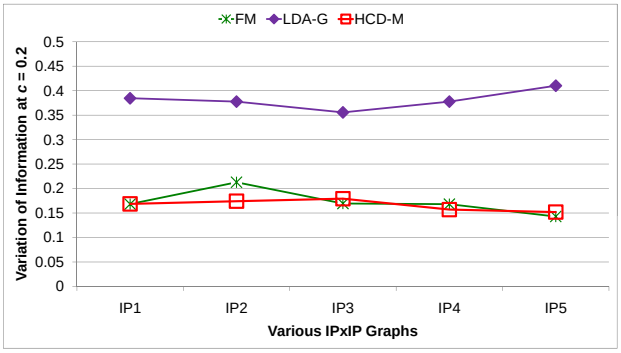

Figure 8: Variation of information for the IP×IP graphs at rewiring probability $c = 0.2$.

For brevity, we omit showing robustness results for the other data graphs in our experiments. Their results are generally similar to Figures 7 and 8. The next section presents a summary discussion of our experiments on other graphs.

4.4 Discussion We first present some general observations from the link prediction results. **(1)** *HCD-X* and *HCD-M* are consistent w.r.t. the link prediction performance of their discovered community structures on a diverse set of real-world graph data. **(2)** *HCD-X* and *HCD-M* always improve on one of their constituents and usually improve on both. **(3)** Restarting *LDA-G* multiple times (with different random seeds to discover a good configuration) does not improve link-prediction performance sufficiently (as shown in Figure 2). This suggests that available computation time is better spent by combining methods than by applying a single method repeatedly. **(4)** On our sparsest IP graph, IP5 with roughly 34K nodes and 112K links, all non-hybrid methods perform only fairly ($\sim$ 0.8 average AUC). However, *HCD-X* and *HCD-M* perform extremely well ($\sim$ 0.96 average AUC).

There are at least two explanations for the superior performance of hybrid methods over non-hybrid methods. First, we explain the improvement over *LDA-G*. Note that *XA* and *FM* are optimizing for different functions than *LDA-G*. Thus, we expect them to compensate for each others' shortcomings when applied together as in *HCD-X*. The hints offered by *XA*, for example, can be used by *HCD-X* to make decisions about community structures that *LDA-G* is ambiguous about. Second, we explain the improved performance of the hybrid methods over the non-Bayesian methods. Neither *XA* nor *FM* can generate mixed-membership models for communities (i.e., soft clustering). In these models, each node belongs to a single community and this community must explain all of the node's links. Because LDA-G is a mixed-membership model, it can loosen this restriction and learn models that explain *all* of a node's links, rather than trying to explain *most* of them.

Of the non-Bayesian methods studied (*XA* and *FM*), *XA* typically produces more predictive community structures. This is not unexpected. As shown by Stone [19], Akaike's criterion (MDL) and cross-validation (MLE, short for Maximum Likelihood Estimate) essentially pick the same model. Also note that, a MDL-based approach can be represented in a Bayesian framework with Universal (Solomonoff) prior probability function and a uniform likelihood function [21]. We empirically observed the relationship between compression and good prediction in our experiments, where methods with higher average AUC produced community-sorted adjacency matrices that have more whitespace and less "snow." Figure 9 shows this phenomenon for the IP5 graph, where the original IP5 adjacency matrix and its corresponding community-sorted matrices produced by *LDA-G*, *FM*, *XA*, *HCD-M*, and *HCD-X* are depicted. The matrices with more whitespace (i.e., less "snow") have higher average AUC on link prediction. The community-sorted matrices were produced as follows. For *FM*, the rows and columns of the adjacency matrix are sorted by their *FM*-assigned communities, so that rows/columns that belong to the same community are next to each other. For *XA*, rows are sorted by their *XA*-assigned row-groups; and similarly, columns are sorted by their *XA*-assigned column-groups. For *LDA-G*, *HCD-M*, and *HCD-X*, rows are sorted first by their *primary community*. A primary community for a row i is the community with the highest percentage of row i's entries. Then, within each primary community, the rows are sorted by their degrees of membership in that community (from highest to lowest). Columns are sorted in the same way.

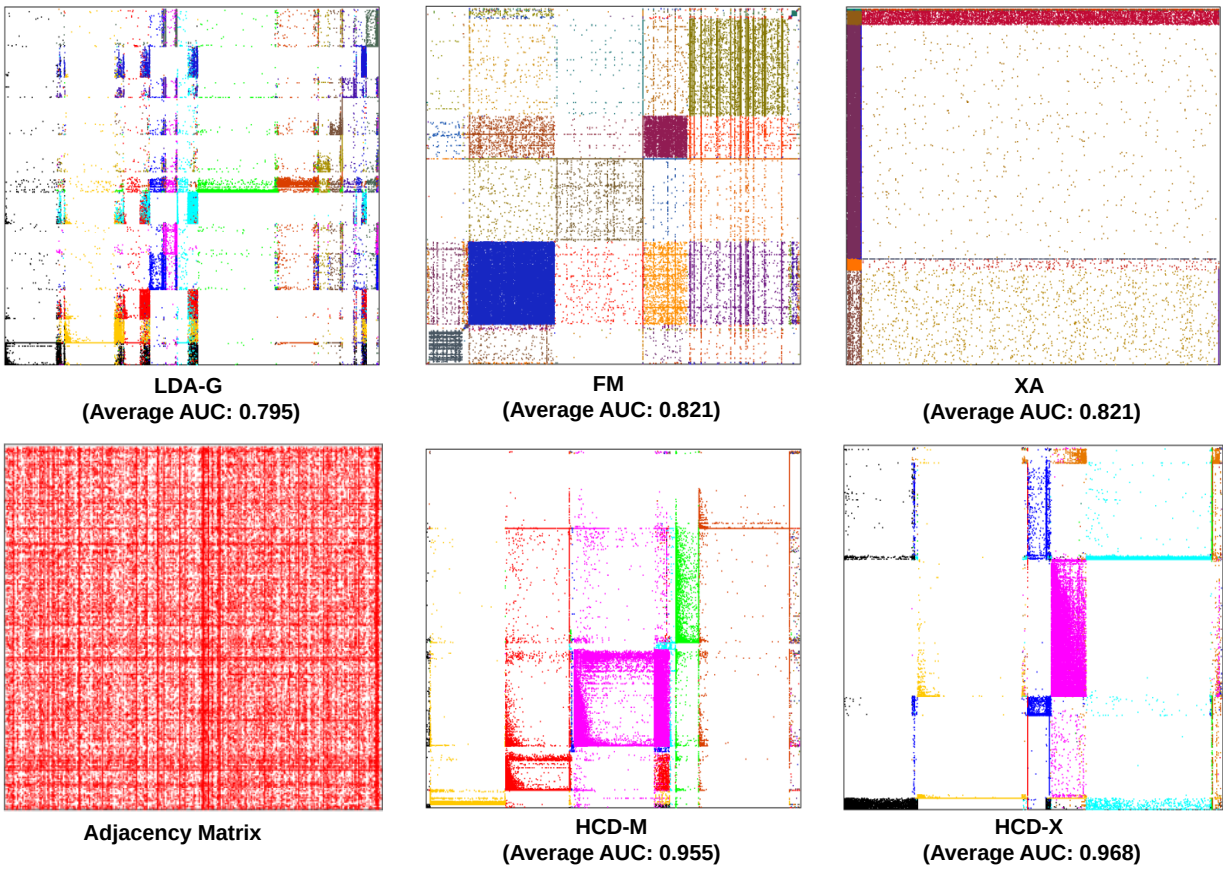

Figure 9: Plots showing the adjacency matrix for the IP5 graph and its corresponding community-sorted matrices produced by existing approaches (*FM*, *XA*, and *LDA-G*) and proposed methods (*HCD-X* and *HCD-M*) on that graph. The matrices associated with the proposed methods have more whitespace (i.e., less "snow"), achieve better clustering, and have higher average AUC on link prediction.

W.r.t. robustness to small perturbations in network structure, Δ generally increases with rewiring probability c, and is maximized for a given graph and algorithm when $c = 1$. This corresponds to a perturbed graph where all of the links are randomly reassigned (i.e., the only remaining information from the original graph is the expected degree of each node). For each IP graph, *XA* produces almost identical clusters at all values of c. We conclude that in these cases, *XA* is determining the groups almost entirely based on degree (which remains the same as links get rewired).

LDA-G has high Δ values even at $c = 0.2$, but the clusters do not deteriorate significantly beyond that point. This suggests that it suffers significantly from small changes in network topology. The hybrid methods have much less variation than *LDA-G*, are comparable to *FM*, and have slightly higher variation than *XA*. However, as discussed above in most cases *XA*'s Δ is artificially low with one community containing a substantial fraction ($> 67\%$) of the nodes. Lastly, we observed that algorithms with higher AUC on link prediction tend to have lower Δ values (assuming the Δ values are not artificially low).

Lastly, *HCD-X* can also be applied to time-evolving graphs by using the appropriate constituents. For example, *LDA-G* can be applied to time-evolving graph by simply using the ML configuration from time $t-1$ as the prior for time t. GraphScope [20] details extensions of *XA* to time-evolving graphs. through the use of appropriate constituents. For instance, Newman, *et al.* [17] describe distributed inference for LDA.

5 Conclusions

We present a novel Bayesian framework, $\mathcal{HCDF}$, for community discovery in large graphs. $\mathcal{HCDF}$ produces algorithms that are (1) effective, in terms of link predic-

tion performance and community structure robustness; (2) consistent, in terms of effectiveness across various application domains; (3) scalable to very large graphs; and (4) nonparametric. A key aspect of $\mathcal{HCDF}$ is its effectiveness in incorporating hints from a number of other community detection algorithms. $\mathcal{HCDF}$ produces results that outperform each of the constituents. On link prediction across a collection of diverse real-world graphs, $\mathcal{HCDF}$ methods achieve (1) improvements of up to 0.22 in average AUC and (2) an average AUC of 0.96. $\mathcal{HCDF}$ methods also find communities that are robust to small perturbations of the network structure as defined by Variation of Information.

6 Acknowledgements

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344.

References

[1] R. Albert, H. Jeong, and A.-L. Barabási. Diameter of the world wide web nature. *Nature*, 140, 1999.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[3] D. Chakrabarti, S. Papadimitriou, D. Modha, and C. Faloutsos. Fully automatic cross-associations. In *KDD*, 2004.

[4] F. R. K. Chung. *Spectral Graph Theory*. AMS Bookstore, 1997.

[5] A. Clauset, M. Newman, and C. Moore. Finding community structure in very large networks. *Phys. Rev. E*, 70:066111, 2004.

[6] I. Dhillon, Y. Guan, and B. Kullis. Weighted Graph Cuts without EigenVectors : A Multilevel Approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1944–1957, November 2007.

[7] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *KDD*, 2003.

[8] J. Ferlez, C. Faloutsos, J. Leskovec, D. Mladenic, and M. Grobelnik. Monitoring network evolution using MDL. In *ICDE*, 2008.

[9] G. W. Flake, S. Lawrence, and C. L. Giles. Efficient identification of web communities. In *KDD*, 2000.

[10] R. Ge, M. Ester, B. J. Gao, Z. Hu, B. Bhattacharya, and B. Ben-Moshe. Joint cluster analysis of attribute data and relationship data: The connected k-center problem, algorithms and applications. *ACM TKDD*, 2(2):1–35, 2008.

[11] T. Griffiths. Gibbs sampling in the generative model of latent Dirichlet allocation. Technical report, Stanford University, 2002. Available at http://citeseer.ist.psu.edu/613963.html.

[12] K. Henderson and T. Eliassi-Rad. Applying latent Dirichlet allocation to group discovery in large graphs. In *ACM SAC*, 2009.

[13] B. Karrer, E. Levina, and M. E. J. Newman. Robustness of community structure in networks. *Phys. Rev. E*, 77:046119, 2008.

[14] G. Karypis and V. Kumar. Parallel multilevel k-way partitioning for irregular graphs. *SIAM Review*, 41(2):278–300, 1999.

[15] H. Li, Z. Nie, W.-C. Lee, C. L. Giles, and J.-R. Wen. Scalable community discovery on textual data with relations. In *CIKM*, 2008.

[16] F. Moser, R. Ge, and M. Ester. Joint cluster analysis of attribute and relationship data without a-priori specification of the number of clusters. In *KDD*, 2007.

[17] D. Newman, A. Asuncion, P. Smyth, and M. Welling. Distributed inference for latent Dirichlet allocation. *NIPS*, 20:1081–1088, 2007.

[18] M. E. J. Newman. Modularity and community structure in networks. *PNAS USA*, 103:8577, 2006.

[19] M. Stone. An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion. *J. Roy. Statist. Soc. Ser. B*, 39:44–47, 1977.

[20] J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu. Graphscope: Parameter-free mining of large time-evolving graphs. In *KDD*, 2007.

[21] P. M. B. Vitányi and M. Li. Minimum description length induction, Bayesianism, and Kolmogorov complexity. *IEEE Transactions on Information Theory*, 46(2):446–464, 2000.

[22] H. Zhang, C. L. Giles, H. C. Foley, and J. Yen. Probabilistic community discovery using hierarchical latent gaussian mixture model. In *AAAI*, 2007.

[23] H. Zhang, B. Qiu, C. L. Giles, H. C. Foley, and J. Yen. An lda-based community structure discovery approach for large-scale social networks. In *IEEE ISI*, 2007.

[24] D. Zhou, Eren, Manavoglu, J. Li, C. L. Giles, and H. Zha. Probabilistic models for discovering e-communities. In *WWW*, 2006.

[25] H. Zhou and D. Woodruff. Clustering via matrix powering. In *PODS*, 2004.

[26] L. Zhou, Y. Liu, J. Wang, and Y. Shi. Hsn-pam: Finding hierarchical probabilistic groups from large-scale networks. In *ICDM Workshops*, 2007.