2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)

Multi-relational Affinity Propagation

Hossam Sharara
Computer Science and Engineering Department
The American University in Cairo
Cairo, Egypt
hossam.sharara@aucegypt.edu

Lise Getoor
Computer Science Department
University of California, Santa Cruz
Santa Cruz, CA
getoor@soe.ucsc.edu

***Abstract*—There is a growing need for clustering algorithms which can operate in complex settings where there are multiple entity types with potential dependencies captured in different kinds of links. In this work, we present a novel approach for multi-relational clustering based on both the similarity of the entities' features, along with the multi-relational structure of the network among the entities. Our approach extends the affinity propagation clustering algorithm to multi-relational domains and encodes a variety of relational constraints to capture the dependencies across different node types in the underlying network. In contrast to the original formulation of affinity propagation that relies on enforcing hard constraints on the output clusters, we model the relational dependencies as soft constraints, allowing control over how they influence the final clustering of the nodes. This formulation allows us to balance between the homogeneity of the entities within the resulting clusters and their connections to clusters of nodes of the same and differing types. This in turn facilitates the exploration of the middle ground between feature-based similarity clustering, community detection, and block modeling in multi-relational networks. We present results on clustering a sample from *Digg.com*, a richly structured online social news website. We show that our proposed algorithm outperforms other clustering approaches on a variety of evaluation measures. We also analyze the impact of different parameter settings on the clustering output, in terms of both the homogeneity and the connectedness of the resulting clusters.**

I. Introduction

Cluster analysis is one of the foundational components of unsupervised learning and exploratory data analysis. It has long attracted the attention of researchers from multiple disciplines. The classical clustering approaches focus on feature similarity for finding latent groupings in the data. However, with the emergence of data that is naturally described in more complex ways, particularly in the form of heterogeneous graphs or networks, these classical approaches are no longer sufficient.

In order to address the challenges in this structured data, a number of graph clustering and community detection approaches have been proposed [1]–[3]. The methods find groups of nodes that are tightly connected to each other, and loosely connected to nodes in different clusters. Similar ideas for graph partitioning are also used for feature-based clustering, by constructing a network among the data points based on their attribute similarity rather than intrinsic structure (e.g., [4]).

IEEE/ACM ASONAM 2022, November 10-13, 2022
978-1-6654-5661-6/22/$31.00 © 2022 IEEE

In addition, block modeling approaches have also been proposed for grouping nodes that link to similar collections of other nodes [5]–[7].

However, many real-world problems include rich, structured relationships that include multiple dependencies among different entity types. Clustering heterogeneous (multiple node types), multi-relational (multiple edge types) networks poses a number of challenges that the proposed algorithms should be able to address. First, the proposed algorithm should be able to account for the structural dependencies between nodes of the same type, as well as the information contained in their descriptive features. Second, the algorithm should be able to model the relationships across different entity types and incorporate them in the clustering process.

To motivate the multi-relational clustering problem, consider the task of customer segmentation for marketing purposes. Using only the customer demographics as features, the best achievable segmentation is a one based on age, gender, etc. Although this demographic profiling might help determine suitable products or appropriate marketing design strategy, it does not provide insight into the social structure, which may be important for predicting product adoption or collections of customers to target. By incorporating the social network structure, a relational clustering algorithm can produce segments that are based on both the demographics and the connectivity of the users in their corresponding social communities. This is likely to help in determining the projected adoption and gives some insight into the social influence. In addition, by considering information about additional relationship types, such as affiliations and memberships between people and other organizations or entities in the network, we may be able to develop a more nuanced picture. For example, a multi-relational clustering algorithm can account for customers' affiliations to different industrial segments and their organizational roles. This may lead to a better quality segmentation, that is more helpful for influence estimation or targeted advertising.

To address these challenges, a number of multi-relational clustering approaches have been introduced in the literature [8]–[12]. While each of these methods has their advantages and disadvantages, the majority of them either make certain distributional assumptions about the underlying data or require certain characteristics in the feature set. In addition, many of these approaches rely on expensive inference methods, such as Gibbs sampling or other MCMC approaches. In this work, we present a novel, general clustering approach that utilizes both feature similarity and relational dependencies across multiple relationship and entity types to produce a clustering that balances between the homogeneity of the data points and their relational structure. The main advantages of our approach are that it is simple, elegant, scalable, and does not make any distributional assumptions about the underlying data. Our work extends the affinity propagation (AP) clustering algorithm [13] to complex networks domains, by leveraging the relational dependencies in the underlying network data through the introduction of structural constraints in the AP model. These constraints bias the optimization problem to favor clusterings which conserve both the homogeneity of the data points as well as their connectedness.

The proposed multi-relational affinity propagation framework uses signals from the links among both similar and different node types to augment the information gained through features similarity, while allowing the user to control the extent of this effect. This facilitates the exploration of clusterings that account for both feature and structural similarities. We show the advantages of our framework over previous approaches through a set of experiments on a sample network from the social news website, *Digg.com*.

II. Related Work

Early work in relational clustering was first done in the ILP community, in which objects of each type are clustered based on the objects of other types linked with them (e.g., [14]). In addition to the logical-based approaches, there has been also a body of literature on probabilistic approaches. Taskar et al. [8] proposed a relational clustering algorithm based on probabilistic relational models that used both feature and link information in uncovering the latent group structure. However, one of the drawbacks of the algorithm is the acyclicity constraint which is hard to satisfy in general network data. Neville et al. [9] proposed a hybrid approach for graph partitioning that relies on both link and feature information.

Although a number of clustering methods have been proposed to combine both feature and structural information, most of them have focused on clustering a single node type, with the link structure serving as an additional factor in determining similarities or enforcing constraints on the clustering problem. Recently, the problem of clustering general heterogeneous data in multi-relational settings has started to attract the attention of more researchers, especially with the increased complexity of the existing data and the associated analysis tasks. An early example is the framework proposed by Zeng et al. [15] for clustering heterogeneous web objects, through an iterative reinforcement clustering process. A different approach was proposed by Xu et al. [6] by introducing an infinite dimensional latent variable for each entity in the network, as part of a Dirichlet process mixture model. As the inference in this approach mainly relies on the Chinese Restaurant Process, the method's performance might not scale favorably for large networks. More recently, a probabilistic framework approach was proposed by Long et al. [16] for clustering different types of entities, taking into consideration the multiple types of relationships among them. However, one of the limitations of this work is that it assumes the underlying statistical distribution of the data belongs to the exponential family.

Bekkerman et al. [10] proposed a framework that simultaneously clusters variables of different types based on their pairwise interactions. More recently, Banerjee et al. [12] proposed a multi-way clustering approach for relational data, that relies on simultaneously clustering multiple entity types represented as a multi-modal tensor. One of the limitations of this approach is that it is only applicable to Bregman loss functions. However, formulating the problem as tensor clustering is an active area of research that has been recently attracting the attention of multiple researchers (e.g., [17], [18]). Other related work includes the framework proposed by Plangprasopchok et al. [19], which extends affinity propagation to account for structural constraints in inferring consistent taxonomies from shallow personal hierarchies on the web. In addition to the previous approaches, a recent logic-based approach was proposed by Kok et al. [20] for discovering new concepts in ILP settings, using a second-order Markov logic framework. The proposed model forms multiple relational clusterings, while iteratively refining them based on the underlying data.

Affinity propagation [13] has also been extended in recent work to social network settings where structural features are considered as an aspect for identifying affinities. Tshimula et al. [21] recently proposed a new approach based on Markov Chain Models to model the affinity in online forums, which is capable of tracking the evolution of affinity over time and predicting affinity relationships arising from the influence of certain community members.

III. Method

We represent the underlying multi-modal network structure as a complex graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{T})$, where $\mathcal{T}$ denotes the different node types in the network, $\mathcal{V} = \{V^t : t \in \mathcal{T}\}$ represents the set of node sets of each type in $\mathcal{G}$, and $\mathcal{E} = \{E^{t_1 \to t_2} : t_1, t_2 \in \mathcal{T}\}$ represents the set of edge sets in $\mathcal{G}$. We distinguish between two types of edges in $\mathcal{E}$: *homogeneous* and *heterogeneous* edges. Homogeneous edges are edges among the same node type (e.g., friendship links among people in a social network), and take the form $E^{t \to t}$. To simplify the notation, we represent homogeneous edges between nodes of type t as E^t. Heterogeneous edges link entities of different types (e.g., affiliation links between people and organizations), and are denoted as $E^{t_1 \to t_2}$ where $t_1 \neq t_2$.

Feature-based clustering approaches focus on clustering data points using similarity measures defined over their features. One simple framework that has been recently proposed is affinity propagation (AP) [13]. AP is an exemplar-based clustering that relies on a message passing algorithm. Given the similarities among the underlying data points, it finds a clustering by identifying a set of exemplars, and finds an optimal assignment of the rest of the data points to these exemplars.

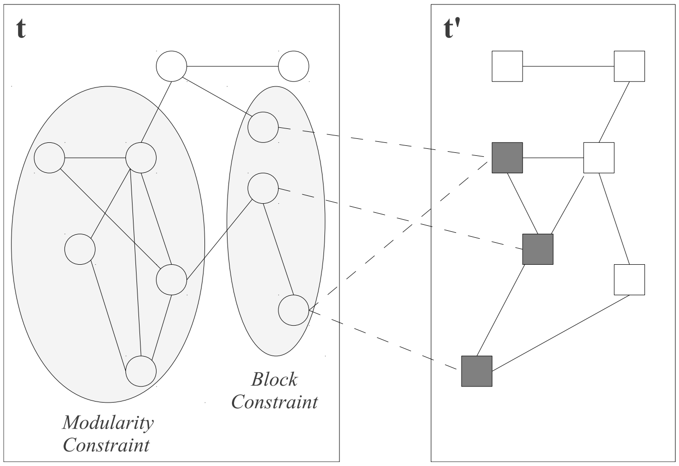


Fig. 1: Sample bimodal network

One of the appealing aspects of affinity propagation is its formulation as a max-sum algorithm on a binary factor graph model [22]. This formulation facilitates the incorporation of new constraints via functional nodes in the underlying factor graph. The similarity values among all pairs of data points, along with the 1-of-*N* constraint which enforces that each node is assigned to a single exemplar, and the exemplar consistency constraint that asserts that exemplar nodes should only choose themselves as exemplars, constitute the core of the affinity propagation algorithm that we use as a base for our approach.

In addition to the feature similarities among the nodes, the edges in in $\mathcal{G}$ also encode a set of relational dependencies across the corresponding entities the nodes represent. These dependencies should be made use of in the proposed multi-relational clustering algorithm. Our proposed algorithm takes these dependencies into consideration along with the feature-based similarity during the clustering process. Thus, we require our clustering algorithm to satisfy these additional conditions:

1) Nodes that are connected by homogeneous links should be in the same cluster (*modularity constraint*).
2) Nodes that are connected by heterogeneous links to nodes of a different type residing in the same cluster should be clustered together (*block constraint*).

The first condition, referred to as a *modularity* constraint, favors clusterings that ensure a high degree of connectivity among the members of the same cluster. This is a common assumption made in a variety of community detection algorithms. The second condition ensures that nodes of one type that are connected to the same cluster of nodes of a different type, are also clustered together. This is a common assumption made in a variety of block-modeling algorithms, and we refer to this constraint as a *block* constraint. Our goal is to encode these conditions in a flexible clustering framework that allows users to vary the importance of each.

A. Model Formulation

Starting from the binary factor graph model introduced by Givoni and Frey [22], we augment it with the additional information needed in our multi-relational setting. Each possible assignment of node i of type t to an exemplar j is modeled as a binary variable c_{ij}^t, such that $(c_{ij}^t = 1)$ iff node i is assigned to the cluster represented by exemplar j. To simplify the discussion, we consider the bimodal network illustrated in Figure 1. The example network $\mathcal{G}$ contains only two different node types $\left(\mathcal{V} = \left\{V^t, V^{t'}\right\}; |V^t| = N, |V^{t'}| = M\right)$, one homogeneous link type among each node type, and one heterogeneous link type across them $\left(\mathcal{E} = \left\{E^t, E^{t'}, E^{t\to t'}\right\}\right)$. Note that the same analysis can be easily extended to settings where there are more than two node and edge types.

Given the above network, the possible assignments of nodes from both types to their corresponding exemplars can be described by the two sets: $\left\{c_{ij}^t\right\}; i,j \in \{1,2,\dots,N\}$, and $\left\{c_{i'j'}^{t'}\right\}; i',j' \in \{1,2,\dots,M\}$. Accordingly, we extend all the constraints defined in the original AP model to our multi-modal settings by replicating the factor nodes for each node type in the model as shown in Figure 2.

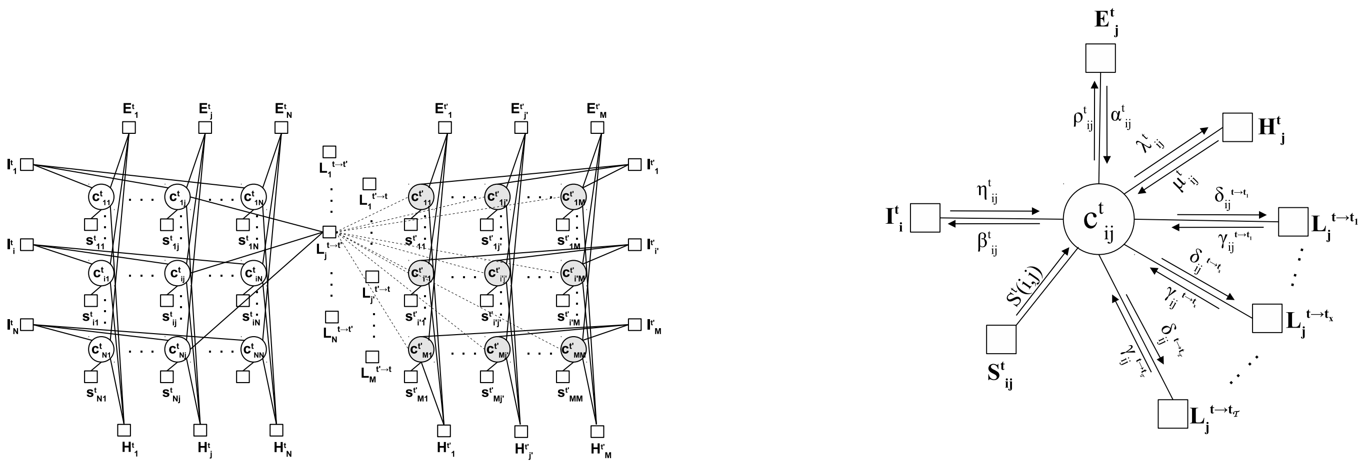
Next, we introduce two additional factors for each type to enforce our proposed modularity and block constraints. We formulate our structural constraints as soft constraints, parametrized by different costs for violating them. As opposed to the formulation of the 1-of-N and exemplar consistency constraints in the original AP model as hard constraints, our soft structural constraints allow the user to control the level of impact of the relational dependencies on the clustering output and at the same time increases the search space by permitting the model to violate some of the constraints to reach a better solution in the optimization process.

The modularity constraint is represented by the factor H^t, which is defined over each node type t as follows:

$$H_j^t(c_{1j}^t, ..., c_{Nj}^t) = \begin{cases} -\theta_i^t & e^t(i,k) \in E^t : c_{ij}^t = 0, c_{kj}^t = 1 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The factor H_j^t is defined over nodes of the same type, and it penalizes clusterings that assign different exemplars for nodes that are directly linked by an edge. For each potential exemplar j for node k, if there is an edge that links i to k, where j is not the current exemplar for node i, a cost θ_i^t is included in the objective function to reduce the likelihood of the corresponding clustering. This cost value can either be constant or variable depending on some structural properties of the terminal nodes (such as clustering coefficient, degree, etc.).

For the second type of constraint capturing the clustering across edges among different node types, we introduce the block constraint factor $L^{t\to t'}$, defined for each pair of node



(a) Binary variable diagram with the structural dependencies factors. To simplify the diagram, we are including only one instance of the factor L

(b) Factor graph representation

Fig. 2: Multi-relational Affinity Propagation Model

types t and t' as follows:

$$L_j^{t\to t'}(c^t_{1j},...,c^t_{Nj}) = \begin{cases} -\omega_i^{t\to t'} & e(i,i'),e(k,k')\in E^{t\to t'}, c^{t'}_{i'j'}=c^{t'}_{k'j'}=1 \\ & : c^t_{ij}=0, c^t_{kj}=1 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The factor $L^{t\to t'}$ is defined over heterogeneous edges, connecting nodes of type t to all the potential exemplars of the nodes of type t' that they are linked to. It penalizes clusterings that assign different exemplars for nodes of type t which are linked to nodes of type t' residing in the same cluster by introducing a cost $\omega^{t\to t'}$ for such configurations. This guides the clustering process to favor clusterings that account for the structural dependencies across node types.

The global objective function of our proposed multi-relational affinity propagation on a network $\mathcal{G}$ can then be expressed as follows:

$$\begin{aligned} S(c^1_{11},\ldots,c^{\mathcal{T}}_{N^{\mathcal{T}}N^{\mathcal{T}}}) &= \sum_{\substack{t\in\mathcal{T}\\ i,j\in V^t}} S^t_{ij}(c^t_{ij}) \\ &+ \sum_{\substack{t\in\mathcal{T}\\ i\in V^t}} I^t_i(c^t_{i1},\ldots,c^t_{iN^t}) + \sum_{\substack{t\in\mathcal{T}\\ j\in V^t}} E^t_j(c^t_{1j},\ldots,c^t_{N^tj}) \\ &+ \sum_{\substack{t\in\mathcal{T}\\ j\in V^t}} H^t_j(c^t_{1j},\ldots,c^t_{N^tj}) + \sum_{\substack{t,t'\in\mathcal{T}\\ j\in V^t}} L^{t\to t'}_j(c^t_{1j},\ldots,c^t_{N^tj}) \end{aligned} \tag{3}$$

B. Message Derivation

Following the derivation of the original AP model, we use the max-sum algorithm to optimize the objective function in Equation 3, by deriving the scalar message updates in the factor graph model shown at Figure 2(b). The max-sum message update rules from a variable node to a factor node can be simply defined as follows [23]:

$$\beta^t_{ij} = \alpha^t_{ij} + \mu^t_{ij} + \sum_{\substack{t'\in\mathcal{T}\\ t'\neq t}} \delta^{t\to t'}_{ij} + S_{ij} \tag{4}$$

$$\rho^t_{ij} = \eta^t_{ij} + \mu^t_{ij} + \sum_{\substack{t'\in\mathcal{T}\\ t'\neq t}} \delta^{t\to t'}_{ij} + S^t_{ij} \tag{5}$$

$$\lambda^t_{ij} = \alpha^t_{ij} + \eta^t_{ij} + \sum_{\substack{t'\in\mathcal{T}\\ t'\neq t}} \delta^{t\to t'}_{ij} + S^t_{ij} \tag{6}$$

$$\gamma^{t\to t'}_{ij} = \alpha^t_{ij} + \mu^t_{ij} + \eta^t_{ij} + \sum_{\substack{t''\in\mathcal{T}\\ t''\neq t,t'}} \delta^{t\to t''}_{ij} + S^t_{ij} \tag{7}$$

where the value S_{ij} corresponds to the feature-similarity between node i and its potential exemplar j, and the message pairs $(\alpha^t_{ij},\rho^t_{ij})$ and $(\eta^t_{ij},\beta^t_{ij})$ are the ones defined for factors E^t_j and I^t_i in the original affinity propagation model for encoding the 1-of-*N* and the exemplar consistency constraints. The message pair $(\mu^t_{ij},\lambda^t_{ij})$ is the one associated with the introduced modularity constraint factor H^t_j, and the messages $(\gamma^{t\to t'}_{ij},\delta^{t\to t'}_{ij})$ are the ones associated with the block constraint factor $L^{t\to t'}_{ij}$.

We now move to the derivation of the message updates from the introduced factors to the corresponding variable nodes. We start with the modularity constraint factor H^t_j defined over nodes of similar types. To simplify the notation, we remove the type qualifiers, as the message derivation for the factor H_j is independent of the node type. To derive the message μ associated with factor H_j, we have to consider the two possible settings for each variable node c_{ij}. First, when $c_{ij}=1$, we get:

$$\mu_{ij}(1) = \max_{c_{kj},k\neq i}\left(H_j(c_{1j},...,c_{ij}=1,...,c_{iN}) + \sum_{l\neq i}\lambda_{lj}(c_{lj})\right) \tag{8}$$

For the cases where a node l is not connected to i, the value of the function H_j reduces to zero. However, for the set of neighboring nodes $D(i) = \{k : \exists e(i,k) \in E\}$ that are homogeneously linked with i, there are two different cases: first, if k is in the same cluster j as i (i.e., $c_{kj} = 1$), then the function H_j reduces to zero. However, in the second case where ($c_{kj} = 0$), the function H_j evaluates to the corresponding cost $-\theta_k$. By taking both cases into consideration, Equation 8 can then be re-written as follows

$$\mu_{ij}(1) = \sum_{k\in D(i)} \max(\lambda_{kj}(1), \lambda_{kj}(0) - \theta_k) + \sum_{l\notin D(i)} \max_{c_{lj}} \lambda_{lj}(c_{lj}) \quad (9)$$

Next, we consider the case when $c_{ij} = 0$:

$$\mu_{ij}(0) = \max_{c_{kj},k\neq i} (H_j(c_{1j}, ..., c_{ij} = 0, ..., c_{iN}) + \sum_{l\neq i} \lambda_{lj}(c_{lj}))$$

Similarly, the assignment of the nodes that are not directly linked to i is unconstrained. However, for the nodes $k \in D(i)$, H_j evaluates to zero for the nodes that are not assigned to exemplar j ($c_{kj} = 0$), and to the cost value $-\theta_i$ of the node i for the ones associated with the value $c_{kj} = 1$. Therefore, the previous equation reduces to

$$\mu_{ij}(0) = \sum_{k\in D(i)} \max(\lambda_{kj}(0), \lambda_{kj}(1) - \theta_i) + \sum_{l\notin D(i)} \max_{c_{lj}} \lambda_{lj}(c_{lj}) \quad (10)$$

By taking the difference between Equations 9 and 10, we get

$$\begin{aligned}\mu_{ij} &= \sum_{k\in D(i)} \max(\min(\lambda_{kj}, \theta_i), \min(-\theta_k, \theta_i - \theta_k - \lambda_k j)) \\ &= \sum_{k\in D(i)} \max(\min(\lambda_{kj}, \theta_i), \min(\lambda_{kj}, \theta_i) - \theta_k - \lambda_{kj}) \\ &= \sum_{k\in D(i)} (\min(\lambda_{kj}, \theta_i) + \max(0, -\lambda_{kj} - \theta_k)) \\ &= \sum_{k\in D(i)} (\min(\lambda_{kj}, \theta_i) - \min(0, \lambda_{kj} + \theta_k)) \quad (11)\end{aligned}$$

It is worth noting that in the final message value for μ_{ij}, if we replace the costs θ_i and θ_k with infinite value, turning the modularity constraint H_J into a hard constraint, the message value reduces to ($\mu_{ij} = \sum_{k\in D(i)} \lambda_{kj}$), which corresponds to the summation of all the incoming messages to i from its similar-type neighbors. However, if we replaced the cost values by zero instead, the value of μ_{ij} reduces to zero, effectively removing the effect of the corresponding constraint.

For deriving the update messages for the second factor type $L_j^{t\to t'}$, we first generalize our definition of the typed neighbor set $D^{t'}(i) = \left\{i' \in V^{t'} : \exists e(i,i') \in E^{t\to t'}\right\}$ as the set of neighboring nodes of type t' that are directly linked to a given node i of type t. We start by considering the case where $c_{ij}^t = 1$:

$$\delta_{ij}^{t\to t'}(1) = \max_{c_{kj},k\neq i} \Bigg(L_j^{t\to t'}(c_{1j}, \ldots, c_{ij} = 1, \ldots, c_{Nj}) + \sum_{l\neq i} \gamma_{lj}^{t\to t'}(c_{lj}) + \sum_{i',j'\in V^{t'}} \gamma_{l'j'}^{t\to t'}(c'_{l'j'}) \Bigg) \quad (12)$$

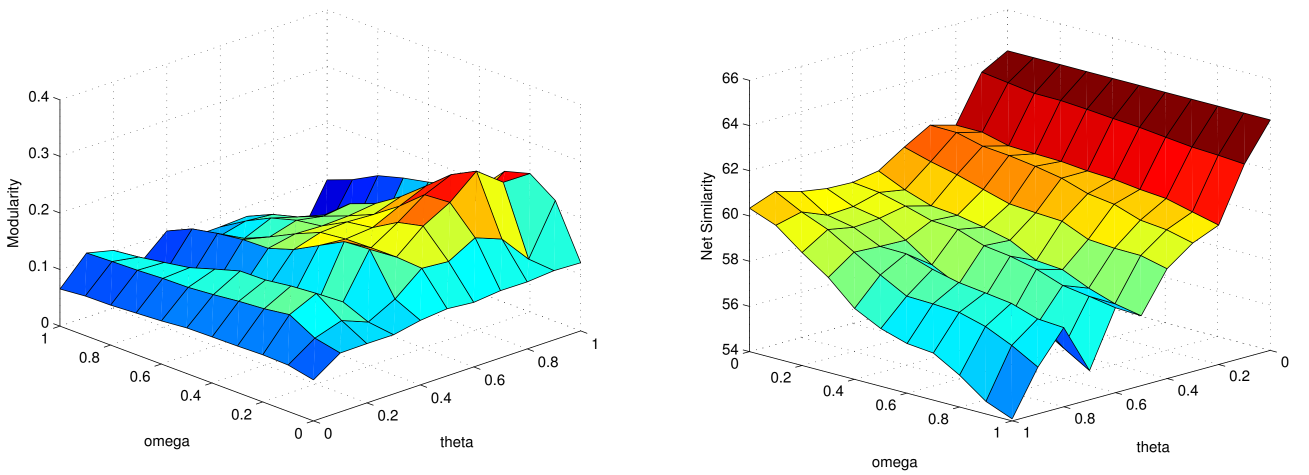
To evaluate the function $L_j^{t\to t'}$ in the previous equation, we need to consider all the potential settings of the other variables c_{kj}^t of type t. For each node $c_{kj}^t = 0$, the function evaluates to the cost value $-\omega_k^{t\to t'}$ for all the nodes k' of type t' that are in the neighbor set $D^{t'}(k)$, and are associated with with a value of $c'_{k'j'} = 1$. The function evaluates to zero in all other cases. To simplify the notation, considering a network with two different entity types, we refer to variables of type t with c_{ij} and the ones of the opposite type t' with c'_{ij}. During the derivation, we also remove the type qualification from the γ, δ, and ω values as we are focusing on deriving the messages of type t, depending on one alternate type t' at a time. So, all the values mentioned in the equations afterwards are presumably qualified with $t \to t'$ type dependency. Thus, the previous equation can be written as follows:

$$\begin{aligned}\delta_{ij}(1) = \sum_{j'\in V^{t'}} \Bigg[&\sum_{l'\notin\{\cup_x D'(x)\}} \max_{c_{l'j'}} \gamma_{l'j'}(c'_{l'j'}) \\ &+ \sum_{i'\in D'(i)} \max\Big[\gamma_{i'j'}(0) + \sum_{k\neq i} \sum_{k'\in D'(k)} \max(\gamma_{kj}(c_{kj}) + \gamma_{k'j'}(c'_{k'j'})), \\ &\gamma_{i'j'}(1) + \sum_{k\neq i} \max\Big[\gamma_{kj}(1) + \sum_{k'\in D'(k)} \max_{c'_{k'j'}} \gamma_{k'j'}(c'_{k'j'}), \\ &\gamma_{kj}(0) + \sum_{k'\in D'(k)} \max(\gamma_{k'j'}(0), \gamma_{k'j'}(1) - \omega_k)\Big]\Big]\Bigg] \end{aligned} \quad (13)$$

The previous equation consists of two main parts: First, all nodes of type t' that are not connected to any node of type t have unconstrained assignment to the any exemplar j'. The second part is a summation over the neighbor nodes of type t' that are connected to the current node i; the function $L_j^{t\to t'}$ evaluates to zero for the nodes in $D'(i)$ that are not assigned to the current exemplar j', and thus all other similar and opposite type nodes are now unconstrained. However, for the nodes $i' \in D'(i)$ that are associated with the value ($c'_{i'j'} = 1$), we need to consider all the other nodes k that are of the same type t as node i; if ($c_{kj} = 1$) then the assignment of the nodes in $D'(k)$ is unconstrained, while if ($c_{kj} = 0$) the function $Lt \to t'_j$ reduces to the cost value ω_k for all the nodes in $D'(k)$ that are assigned to exemplar j'.

Similarly, for $c_{ij} = 0$, the same derivation applies, and the value for $\delta_{ij}(0)$ can be represented as follows:

$$\begin{aligned}\delta_{ij}(0) = \sum_{j'\in V^{t'}} \Bigg[&\sum_{l'\notin\{\cup_x D'(x)\}} \max_{c_{l'j'}} \gamma_{l'j'}(c'_{l'j'}) \\ &+ \sum_{i'\in D'(i)} \max\Big[\gamma_{i'j'}(0) + \sum_{k\neq i k'\in D'(k)} \max(\gamma_{kj}(c_{kj}) + \gamma_{k'j'}(c'_{k'j'})), \\ &\gamma_{i'j'}(1) + \sum_{k\neq i} \max\Big[\gamma_{kj}(0) + \sum_{k'\in D'(k)} \max_{c'_{k'j'}} \gamma_{k'j'}(c'_{k'j'}), \\ &\gamma_{kj}(1) + \sum_{k'\in D'(k)} \max(\gamma_{k'j'}(0), \gamma_{k'j'}(1) - \omega_i)\Big]\Big]\Bigg] \end{aligned} \quad (14)$$



(a) Modularity

(b) Net Similarity

Fig. 3: The effect of varying the cost parameters (θ, ω) on the net similarity and the modularity of the output clustering

By taking the difference between Equations 13 and 14, we get

$$\delta_{ij} = \sum_{j' \in V^{t'}} \sum_{i' \in D'(i)} \Bigg(\min\Big(\gamma_{i'j'} + \sum_{k \neq i} \min(0, \gamma_{kj}), \sum_{k \neq i} \min(A_{i,k}, \gamma_{kj})\Big) - \min\Big(\gamma_{i'j'} + \sum_{k \neq i} \min(0, \gamma_{kj}), \sum_{k \neq i} \min(0, \gamma_{kj} - B_k)\Big) \Bigg) \tag{15}$$

where the variables $A_{i,k}$ and B_k are defined as follows

$$A_{i,k} = \sum_{k' \in D'(k)} \big(\min(\omega_i, \gamma_{k'j'}) - \min(0, \gamma_{k'j'}) \big)$$

$$B_k = \sum_{k' \in D'(k)} \big(\max(0, \gamma_{k'j'} - \omega_k) - \max(0, \gamma_{k'j'}) \big)$$

IV. Experimental Evaluation

To evaluate our proposed multi-relational affinity propagation approach, we use a dataset from *Digg.com*, a popular social news website, where users can post stories on different topics, and then vote on them in a process referred to as "*digging*" to determine the story's ranking on the front page. Digg users form a social network by "*following*" other users, which in turn results in the target user posts showing on their homepages. We constructed a sample from the Digg network which includes 104,478 "*following*" links among 3750 users and their 438,379 "*digging*" links to 3305 stories. We use the "**title**" and "**description**" of the stories to construct a normalized tf-idf word vector for each post, which is then used to measure cosine similarities between different stories. Similarly for the users, we used the "**about**" field that the user provides upon registration to compute the cosine similarities between users over the corresponding tf-idf vectors.

The AP algorithm relies on setting a preference value for each node in the network that reflects the likelihood of this point being an exemplar, which then affects the number of clusters in the output. In our experiments, we follow the approach that was proposed in the original AP model where there is no prior bias for certain nodes to be exemplars, and thus we set the preference value to the median of the corresponding input similarities. For evaluation, we use both the modularity of the resulting community structure and the total similarity, referred to as net similarity, of the exemplars to their assigned nodes in the output clustering to show the impact of different cost settings. As for the comparison with other baselines, we use both Davies-Bouldin [24] and Dunn [25] indices for internal clustering validation, as well as normalized mutual information (NMI) [26] for external validation with the ground truth when available.

First, we show the performance of our algorithm over a range of cost values for both the homogeneous and heterogeneous structural constraints. Figure 3(a) shows the performance of our proposed algorithm for both the modularity and the net similarity measures of the user nodes clustering in the Digg dataset. We note that for lower values of ω, the modularity of the output clustering increases with increasing the value of θ, which corresponds to having higher costs on violating the homogeneous communities constraint. However, by increasing the value of the heterogeneous link cost ω for a given value of θ, the modularity of the output clustering increases initially, and then starts to decreases on higher values of ω. This can be attributed to the fact that increasing the cost of violating the block constraint initially provides additional evidence for the clustering of the alternate node type, but after a given point it starts fragmenting the clustering output, resulting in an increased number of clusters which decreases the overall modularity of the output clustering. On the other hand, Figure 3(b) shows the trade-off in terms of the decrease in the net similarity of the clustering output with increasing the cost values. However, we can note from the

	Users			Stories		
	Modularity	**DB Index**	**Dunn Index**	**NMI**	**DB Index**	**Dunn Index**
MMRC	0.005	2.23	0.63	0.106	2.09	0.81
Modularity Maximization	**0.458**	2.4	0.57	N/A	N/A	N/A
Affinity Propagation	0.072	**1.504**	0.67	0.209	1.86	0.86
Multi-relational AP	0.13 (0.28)	1.52 (1.54)	**0.76** (**0.78**)	**0.287** (**0.34**)	**1.852** (**1.859**)	**0.868** (**0.87**)

TABLE I: Comparison of MMRC, Modularity Maximization, AP, and multi-relational AP on different clustering evaluation measures. For multi-relational AP, the reported values are the average over all settings of the cost parameters, while the ones in parentheses are obtained from the optimal parameter settings, identified through an exhaustive search. The entries in bold face correspond to the best performance for the corresponding measure.

figure that the average decrease in the net similarity across different cost settings is much lower than the increase gained in terms of the modularity of the output clustering. Due to the lack of edges among story nodes, we are unable to show the effect of changing the value of θ on the clustering of stories, or to compute the modularity of the output clustering. However, by varying the cost of the block constraint, we get similar trends in the net similarity, which we are omitting due to space constraints. Next, we compare our approach to the MMRC relational clustering algorithm proposed in Long et al. [16], as well as the original affinity propagation model and the modularity maximization algorithm [3]. We use different clustering evaluation measures for analyzing the performance of the algorithms as shown in Table I. The results show that our proposed multi-relational affinity propagation approach results have a superior performance compared to the MMRC relational clustering algorithm on all evaluation measures. By analyzing the clustering of the "users" type, we find that while the modularity maximization algorithm achieves the best modularity score for its output, it performs poorly on similarity-based measures. On the contrary, the original affinity propagation model shows better performance on the similarity based measures than the modularity score. However, our proposed multi-relational AP algorithm shows good performance on both structure-based and similarity-based evaluation measures, illustrating the balance that it is capable of achieving between both paradigms.

Finally, we move to the evaluation of the clustering for the "stories" node type. Due to the fact that there are no links among stories in Digg, we are unable to compute the modularity measure. However, the stories on Digg are manually assigned to a specific topic when posted. Therefore, we are able to use the story topic as an evaluation measure for clustering this node type, which enables us to compute the normalized mutual information (NMI) quality measure of the output clustering. As we can see in Table I, our proposed algorithm achieves the best performance on all evaluation measures, including the NMI measure with the ground truth. This shows the value of the signal from block constraint and the favorable effect of coupling the clustering process across different node types. Another important advantage of our multi-relational affinity propagation algorithm over the MMRC algorithm is computational efficiency, as our multi-relational affinity propagation algorithm is orders of magnitude faster than the MMRC algorithm.

V. Conclusion

In this work, we presented a novel, multi-relational clustering framework for identifying latent groupings in complex network domains. Our proposed approach provides a simple and elegant way of extending the affinity propagation framework to multi-relational network domains by incorporating different soft constraints for capturing the structural dependencies among different types of nodes in the network. We showed how our proposed multi-relational affinity propagation algorithm can be used to output different clustering with varying degrees of dependence on both the feature similarity of the nodes as well as their relational structure. We conducted a set of experiments on a sample network from an online social news website, and showed that our proposed approach outperformed previous approaches for multi-relational clustering.

References

[1] M. Everett and S. Borgatti, "Analyzing clique overlap connections," *Connections*, vol. 21, no. 1, pp. 49–61, 1998.

[2] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *PNAS*, vol. 99, no. 12, pp. 7821–7826, 2002.

[3] M. E. J. Newman, "Modularity and community structure in networks," *PNAS*, vol. 103, no. 23, pp. 8577–8696, 2006.

[4] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE PAMI Trans.*, vol. 22, no. 8, pp. 888–905, 2000.

[5] C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda, "Learning systems of concepts with an infinite relational model," in *AAAI*, 2006.

[6] Z. Xu, V. Tresp, K. Yu, , and H. Kriegel, "Infinite hidden relational models," in *UAI*, 2006.

[7] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, "Mixed membership stochastic blockmodels," *JMLR*, vol. 9, pp. 1981–2014, 2008.

[8] B. Taskar, E. Segal, and D. Koller, "Probabilistic classification and clustering in relational data," in *IJCAI*, 2001.

[9] J. Neville, M. Adler, and D. Jensen, "Clustering relational data using attribute and link information," in *Text Mining and Link Analysis Workshop*, 2003.

[10] R. Bekkerman, R. El-Yaniv, and A. McCallum, "Multi-way distributional clustering via pairwise interactions," in *ICML*, 2005.

[11] B. Long, Z. Zhang, X. Wú, and P. S. Yu, "Spectral clustering for multi-type relational data," in *ICML*, 2006.

[12] A. Banerjee, S. Basu, and S. Merugu, "Multi-way clustering on relation graphs," in *SDM*, 2007.

[13] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, pp. 972–976, 2007.

[14] M. Kirsten and S. Wrobel, "Relational distance-based clustering," in *ILP*, 1998.

[15] H. J. Zeng, Z. Chen, and W. Y. Ma, "A unified framework for clustering heterogeneous web objects," in *WISE*, 2002.

[16] B. Long, Z. Zhang, and P. S. Yu, "A probabilistic framework for relational clustering," in *KDD*, 2007.

[17] S. Jegelka, S. Sra, and A. Banerjee, "Approximation algorithms for tensor clustering," in *ALT*, 2009.

[18] I. Sutskever, R. Salakhutdinov, and J. Tenenbaum, “Modelling relational data using Bayesian clustered tensor factorization,” in *NIPS*, 2009.

[19] A. Plangprasopchok, K. Lerman, and L. Getoor, “A probabilistic approach for learning folksonomies from structured data,” in *WSDM*, 2011.

[20] S. Kok and P. Domingos, “Statistical predicate invention,” in *ICML*, 2007.

[21] J. Tshimula, B. Chikhaoui, and S. Wang, “A new approach for affinity relationship discovery in online forums,” *Social Network Analysis and Mining*, vol. 10, 12 2020.

[22] I. E. Givoni and B. J. Frey, “A binary variable model for affinity propagation,” *Neural Computation*, vol. 21, no. 6, pp. 1589–1600, 2009.

[23] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[24] D. L. Davies and D. W. Bouldin, “A clustering separation measure,” *IEEE PAMI Trans.*, vol. 1, no. 2, pp. 224–227, 1979.

[25] J. C. Dunn, “A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters,” *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, 1973.

[26] A. Strehl and J. Ghosh, “Cluster ensembles - a knowledge reuse framework for combining partitionings,” in *AAAI*, 2002.