Towards Description of Block Model on Graph

Zilong Bai[1(✉)], S. S. Ravi[2], and Ian Davidson[1]

[1] University of California, Davis, USA
zlbai@ucdavis.edu, davidson@cs.ucdavis.edu
[2] University of Virginia, Charlottesville, USA
ssravi0@gmail.com

Abstract. Existing block modeling methods can detect communities as blocks. However it remains a challenge to easily explain to a human why nodes belong to the same block. Such a description is very useful for answering *why people in the same community tend to interact cohesively*. In this paper we explore a novel problem: Given a block model already found, describe the blocks using an auxiliary set of information. We formulate a combinatorial optimization problem which finds a *unique* disjunction of the auxiliary information shared by the nodes either in the same block or between a pair of different blocks. The former terms intra-block description, the latter inter-block description. Given an undirected graph and its $k-$block model, our method generates $k + \frac{k(k-1)}{2}$ different descriptions. If the tags are descriptors of events occurring at the vertices, our descriptions can be interpreted as common events occurring within blocks and between blocks. We show that this problem is intractable even for simple cases, e.g., when the underlying graph is a tree with just two blocks. However, simple and efficient ILP formulations and algorithms exist for its relaxation and yield insights different from a state-of-the-art related work in unsupervised description. We empirically show the power of our work on multiple real-world large datasets.

Keywords: Explainable artificial intelligence · Unsupervised graph analysis · Block model

1 Introduction

Block modeling is an effective community detection method (cf. [1,30]). Conventionally, communities can be defined as subgraphs that are densely connected within but weakly connected to each other (cf. [24,43]). However, block models find communities based on different definitions of similarity/equivalence in terms of how they connect to the rest of the graph, such as structural equivalence (i.e., second-order proximity [38,43]) and stochastic equivalence (e.g., stochastic block model [1]). Equation 1 exemplifies a basic formulation of block modeling (cf. [15]),

F. Hutter et al. (Eds.): ECML PKDD 2020, LNAI 12459, pp. 37–53, 2021.
https://doi.org/10.1007/978-3-030-67664-3_3

which simultaneously discovers k blocks stacked column-wise in an $n \times k$ block allocation matrix $\mathbf{F}$ and the block-level connectivity in a $k \times k$ image matrix $\mathbf{M}$.[1]

$$\underset{\mathbf{F}\geq\mathbf{0},\mathbf{M}\geq\mathbf{0}}{Minimize} \|\mathbf{G} - \mathbf{F}\mathbf{M}\mathbf{F}^T\|_F \ s.t., \mathbf{F}^T\mathbf{F} = \mathbf{D} \tag{1}$$

Block modeling produces a useful abstract view of the graph it simplifies by finding, for instance, structural equivalence [43]. However, existing block models are ineffective in explaining to a human domain expert *why* the members in the same block are equivalent which limits their use in some domains. Consider the sociological definition of a "community" which requires several properties[2] including i) interacting people, and ii) members who share common values, beliefs, or behaviors. A block model finds the former but not the latter. Despite existing work on posterior descriptions of detected communities (e.g., [29]) and efforts on descriptive community detection (e.g., [5,6,21]), it remains understudied in finding a complete explanation for a given block model. A complete explanation would describe both communities themselves *and* differences between the communities. Explaining differences between communities is important as in block models, in contrast to conventionally defined communities, strong inter-block connectivity can exist (e.g., [20,22]). This is particularly important if the blocks are discovered as *roles* [34], as each role is not defined individually but based on its interactions with others (e.g., [3,27]).

In this paper we consider the setting where we are given a block model $\mathcal{B} = \{\mathbf{F}^{n\times k}, \mathbf{M}^{k\times k}\}$ discovered from an adjacency matrix $\mathbf{A}$ of graph $G(V, E)$. We aim to explain the block model $\mathcal{B}$ using auxiliary information which are human-interpretable tags. We explore a novel problem of finding a description as a *disjunction* comprising the most frequently used tags for the edges within each block and between each pair of different blocks. The former is denoted as intra-block description, the latter inter-block description. If G is an undirected simple graph, our method generates k intra-block descriptions and $\frac{k(k-1)}{2}$ inter-block descriptions. We introduce two different disjointness constraints between the descriptions of different edge sets to address *how each edge set is unique or special to the other edge sets*. We illustrate our work with a toy example in Fig. 1 and a running example in Table 1 by comparing against DTDM [12]. Albeit closely related to our work in terms of Integer Linear Programming (ILP) formulations, DTDM considers *item* cluster description whereas ours focuses on *edge* subsets induced by the block model on the graph.

The major contributions of this work are as follows: 1. We propose a novel combinatorial optimization formulation, Valid Tag Assignment to Edges (VTAE), for describing the result of an existing block model based on edge covering (Sect. 3). 2. We show VTAE is intractable (Sect. 3), hence no exact efficient algorithm is possible. 3. We propose ILP formulations for VTAE and a relaxation with efficient algorithm that guarantees feasibility for this relaxation (Sect. 4). 4. We demonstrate our method on three real-world datasets (Sect. 5).

[1] Here $\mathbf{D}$ denotes a diagonal matrix. We use $\| \bullet \|_F$ to denote the Frobenius norm.
[2] https://www.oxfordbibliographies.com/view/document/obo-9780199756384/obo-9780199756384-0080.xml.

Table 1. Partial experimental results of descriptions learnt by our VTAE with *global* disjointness (see Sect. 3) and DTDM [12] applied to a Twitter dataset of 880 nodes given its precomputed 4-block block model.

Method	Block 1 "Trump supporters"	Block 2 "Hillary supporters"	Between block 1 and 2
Our VTAE-g	`Hillary`, `TrumpRally`, `CNN`, `2A`, `Ohio`, `USA`, `AmericaFirst`, `trump2016`, `TeamTrump`, `TRUMP2016`, `VoteTrump2016`	`BernieSanders`, `Bernie`, `Hillary2016`, `p2`, `PAprimary`, `UniteBlue`, `MDPrimary`, `BernieOrBust`, `CT`, `ArizonaPrimary`	`NewYorkValues`, `ISIS`, `AZPrimary`, `MSM`, `NewYorkPrimary`, `pjnet`, `WI`, `trumptrain`, `1A`, `maga`, `CCOT`, `CRUZ`, `SECPrimary`
DTDM [12]	`GOPdebate`, `DNCinPHL`, `Clinton`, `GOPDEBATE`, `BlackLivesMatter`, `MAGA`, `Sanders`, `CrookedHillary`, `Hannity`, `UniteBlue`, `CCOT`	`NeverTrump`, `DemsInPhilly`, `GOP`, `1`, `SCPrimary`, `FeelTheBern`, `PrimaryDay`, `DemTownHall`, `FITN`, `NewYork`, `DumpTrump`	**N/A**

Our VTAE demonstrates advantages over DTDM [12] in: i) representing the similarity/difference between different block models (Fig. 6), ii) being robust against small perturbations in a given block allocation (Fig. 7).

2 Related Work

There has been a growing need for explainable machine learning and data mining approaches [16]. However many approaches focus on supervised learning [36], such as explaining per-instance prediction/classification result (e.g., [7,14,33]) of a *black-box* deep learning model [2]. Some other efforts are about describing unsupervised learning results. In line with the focus of this work, we briefly review former efforts on describing unsupervised learning (on graph).

Conceptual Clustering [18] attempts to use the *same features* for finding the clustering and explaining it. Extensions to find clusterings in one view and a description in another exist [10] but is limited to a single view of categorical variables. To find a clustering and explanation simultaneously in multiple views, recent work has explored conceptual clustering extensions with constraint programming [23], SAT [25] or ILP [10,28] but their scalability is limited due to the Pareto optimization formulation. Our work is different from this setting as we explicitly consider the edges in a graph.

Clustering on Vertex-Labeled Graphs explores finding vertex clusterings based on graph topology and vertex labels. Recent advances on vertex/edge-attributed graphs include novel extensions of community detection [4,17] or SBM [1]. Representative work of the latter includes SBM on graphs where the nodes are associated to binary attributes (e.g., [42]) or continuous values (e.g., [37]). A general underlying premise of such work is that the graph topology and tag/label information compensate for each other [13]. Although it seems relevant, this line of research does not create succinct descriptions for the *subgraph*s induced from a graph by a vertex set partitioning, hence is different from ours.

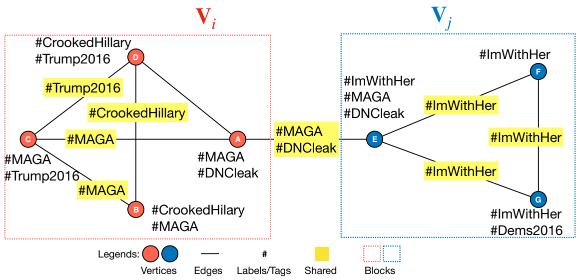
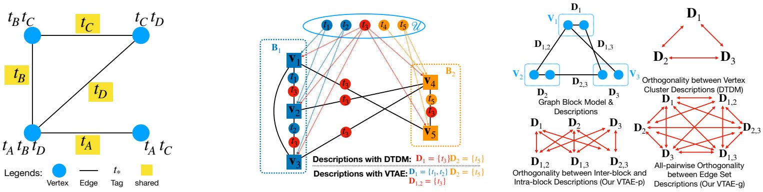


Fig. 1. An example using hashtags to describe two blocks/communities V_i, V_j, red and blue, on a Twitter network. The edge between vertices A and E is the only edge between V_i and V_j covered by {`DNCleak`}. The edges within V_i are covered by {`CrookedHillary`, `MAGA`, `Trump2016`}, V_j by {`ImWithHer`}. Previous work [12] finds overly simple distinct descriptions {`MAGA`, `Trump2016`} and {`ImWithHer`} to cover V_i and V_j.

Descriptive Community Detection simultaneously *discovers* communities and their descriptions on vertex/edge-labeled graphs. Representative approaches include seminal work [6] and its later extension [5] that develop a search-based algorithm where each node on the search tree is a candidate community that can be described by a conjunction of labels. Another approach discusses three different `0−1` predicates to describe the edges within each community - i.e., conjunction, majority, and disjunction, and explores the first two in their algorithm development and experiments [21]. A framework alternates between two phases: i) the community detection phase and ii) inducing a disjunction of possible negated conjunctions to match the community and reshape the community if no perfect match is found [32]. Different from descriptive community detection approaches, our work aims to faithfully describe the effect of a given block model on a graph instead of discovering or modifying the community structure. We are post-processing results hence agnostic to how a block model is generated. Moreover, existing work is primarily concerned with finding a description for *each* community with assumption of community structure in a conventional sense - i.e., the nodes are well-connected to each other within the same community but are less connected to the outside nodes (cf. [19,31]). Whereas our work aims to find a succinct description for each set of intra-block or inter-block edges induced from a given block model, where significant inter-block/community edges can exist and need to be described.

Post-processing Output of Existing Algorithms (i) *Given* a community structure, [29] explores topic modeling; it *separately* builds topic models with Latent Dirichlet Allocation (LDA) (cf. [39,40]) and discovers communities based on the Girvan-Newman community detection algorithm (cf. [24]), and then combines the two by computing the probabilistic distribution over the topics for each community. However, such topic model only applies to each community without modeling the interactions between different communities. Our work considers a



(a) Our VTAE vs DTDM [12] on a single community.

(b) Advantage of our VTAE over DTDM [12] in a two-block example.

(c) Comparison of disjointness constraints.

Fig. 2. Comparing our VTAE against DTDM [12].

fundamentally different type of description as a disjunction of tags and simultaneously discovers descriptions for both the inter-block edges and the intra-block edges induced by a block model on a graph. **(ii)** Most similar to ours, [12] formulates the novel cluster description problem as a combinatorial optimization problem to find concise but different descriptors of each cluster. However, that work is not specifically for graphs and ignores any graph structure. A technical limitation of [12] is that the description it learns for one community can actually cover a lot of nodes in other communities. Our method alleviates this issue as we simultaneously discover descriptions with disjointness constraints for both intra-block and inter-block edge sets, which can reduce the odds of using the attributes shared by nodes from two different blocks in describing the intra-block edges. See Fig. 2 for more illustrative examples. In this paper, we do not compare with [35] which proposes efficient algorithms based on random rounding as a follow-up work on [12], since we focus on formulation-wise contributions.

3 Problem Definition and Complexity Results

In this section we define the description problem for block models and show that the problem is intractable even for relaxed variations. We first introduce some necessary notation to formalize the problem. Each undirected graph $G(V, E)$ considered here has the following properties.

1. Each vertex $v \in V$ has an associated set $T(v)$ of tags. Some times we will refer to all tags for all instances as a matrix L.
2. For each edge $\{x, y\} \in E$, the associated tag sets $T(x)$ and $T(y)$ have *at least one common tag*; that is, $T(x) \cap T(y) \neq \emptyset$.[3]

Input. Given an undirected graph $G(V, E)$ and a subset $V_i \subseteq V$ of vertices found by the block model, we use $G_i(V_i, E_i)$ to denote the subgraph of G *induced* on the vertex set V_i. (Thus, E_i contains all and only those edges $\{x, y\}$ such that

[3] When this condition does not hold, our method can still be used but just with these edges removed as they are unexplainable.

both x and y are in V_i and $\{x, y\} \in E$.) We refer to each set E_i as an **intra-block edge set**. For each pair of subsets V_i and V_j, with $i \neq j$, we use E_{ij} to denote the set of edges that join a vertex $x \in V_i$ to a vertex $y \in V_j$. We refer to each such set as an **inter-block edge set**. Each partition of V into k subsets gives rise to k sets of intra-block edges and $k(k-1)/2$ sets of inter-block edges. We denote the collection of these edge sets as $\mathcal{E}(G)$. Some of these edge sets may be empty and can be ignored if their corresponding image matrix entries are zero.

Output. Let $\Gamma = \cup_{v \in V} T(v)$ denote the collection/union of all the tags assigned to the vertices which is the universe of all tags. The goal of the problem considered below is to find a description for each edge set (both intra-block and inter-block). To do this it is required to first find a function $\tau : E \longrightarrow \Gamma$ that assigns a tag $\tau(e) \in \Gamma$ to each edge $e \in E$. Given a tag assignment function τ and an edge set $X \subseteq E$, we use $\tau(X)$ to denote the set of all the tags assigned to the edges in X. A tag assignment function τ is **valid** if and only if it satisfies *both* of the following conditions.

(a) [**Compatibility:**] For each edge $e = \{x, y\} \in E$, $\tau(e) \in T(x) \cap T(y)$; that is, the tag $\tau(e)$ assigned to e appears in the tag sets of both the end points of e.

(b) [**Disjointness:**] For a pair of *different* edge sets X and Y in $\mathcal{E}(G)$, $\tau(X) \cap \tau(Y) = \emptyset$. That is, the sets of tags assigned to X and Y must be *disjoint*.

In our work and experiments we explored two types of disjointness: i) Partial Disjointness which requires X to be an intra-block edge set and Y to be an inter-block edge set and ii) Global Disjointness which requires disjointness between any two different edge sets from $\mathcal{E}(G)$.

We consider the following decision problem and show it is intractable. The optimization version of the problem adds the requirement of finding the most compact description (see Eq. 2).

Valid Tag Assignment to Edges (VTAE)

Instance: An undirected graph $G(V, E)$, for each vertex $v \in V$ a nonempty set $T(v)$ of tags so that for any edge $\{x, y\} \in E$, $T(x) \cap T(y) \neq \emptyset$, a partition of V into k subsets $V_1, V_2, \ldots, V_k$.

Question: Is there a valid tag assignment function $\tau : E \longrightarrow \Gamma$, where Γ is the union of all the tag sets of vertices?

The Complexity of VTAE

We first observe that if the partition of the vertex set V consists of just one block (namely, the vertex set V), the VTAE problem is trivial: for each edge $\{x, y\}$, one can choose any tag from the nonempty set $T(x) \cap T(y)$, since there is no inter-block edge set in this case. The following theorem shows that the VTAE problem becomes computationally intractable even when the underlying graph is a tree and the given partition of V contains just two blocks/subsets.

Theorem 1. *The VTAE problem with either* ***partial*** *or* ***global*** *disjointness is* ***NP****-complete even when the graph $G(V, E)$ is a tree and the vertex set is partitioned into just two subsets.*

Proof: See supplementary material[4].

4 ILP Formulations and Algorithms

In this section we first formulate a basic instantiation of the VTAE problem and an extension with enhanced statistical significance as ILP formulations (Eq. 2 and 3). We then relax VTAE to ensure that feasible solutions exist and facilitate scalability. Particularly, our **divide and conquer** Algorithm 1 can describe the block models on a graph of $2 \times 10^6 s$ edges over 10^4 vertices with marginally ignored edges in about 2 min on a machine with 4-core Intel Xeon CPU (Fig. 5).

Notations. We are given an unweighted & undirected graph $G(V, E)$ over n vertices and a k-way vertex set partitioning $\{V_1, \ldots, V_k\}$ of a block model. Each vertex is associated with a subset of the $|\Gamma|$ tags. These tags are in an $n \times |\Gamma|$ matrix L. The block model induces k **intra-**block and $\frac{k(k-1)}{2}$ **inter-**block edge sets on the graph. The collection of these edge sets is $\mathcal{E}(G)$. For each edge set $Z \in \mathcal{E}(G)$ we solve for a $|\Gamma|$-dimensional binary vector D_Z where $D_Z(t) = 1$ `iff` tag t is chosen to describe this edge set. We define other frequently used notations in Table 2. Note they are precomputed to facilitate ILP formulations.

Minimal VTAE and an Extension - ILP Formulations. Here we present the ILP formulation for VTAE with an objective to find the most compact collection of descriptions in Eq. 2. To enhance the statistical significance of description, we propose an extension to VTAE which aims to find the most frequently used tags to build each description, additionally constrained by compactness (e.g., an integer upper bound c_Z) in Eq. 3.

Table 2. Frequently used notations.

Notation	Definition and computation
L_Z	**Tag allocation matrix** of edge set Z. To assist formulating **compatibility** constraint as a set coverage requirement. $\lvert Z\rvert \times \lvert\Gamma\rvert$ binary matrix. $L_Z(e, t) = 1$ `iff` $e \in Z, \{x, y\} = e, t \in T(x) \cap T(y)$
U_Z	**Tag universe** indicator where the most concise explanation for edge set Z draws tags from. $\lvert\Gamma\rvert$-d binary vector. $U_Z(t) = 1$ `iff` $L_Z(e, t) = 1, \exists e \in Z$
w_Z	**Edge coverage** of tags in Z. $\lvert\Gamma\rvert$-d integer vector. $w_Z(t) = \sum_{e \in Z} L_Z(e, t)$

[4] Proofs for our theorems, our codes and other supplementary materials are available on https://github.com/ZilongBai/ECMLPKDD2020TDBMG. .

$$\begin{aligned}
&\underset{D_Z(t)\in\{0,1\}}{Minimize} \ \sum_Z\sum_t \ D_Z(t)\\
s.t.&\sum_t D_Z(t) \times L_Z(e,t) \geq 1, \forall e \in Z, \forall Z \in \mathcal{E}(G) \ (\textbf{cover})\\
&D_X(t) + D_Y(t) \leq 1, \forall t, \forall X \neq Y \in \mathcal{E}(G) \ (\textbf{disjointness})
\end{aligned} \tag{2}$$

$$\begin{aligned}
&\underset{D_Z(t)\in\{0,1\}}{Maximize} \ \sum_Z\sum_t \ D_Z(t) \times w_Z(t)\\
s.t.&\sum_t D_Z(t) \times L_Z(e,t) \geq 1, \forall e \in Z, \forall Z \in \mathcal{E}(G) \ (\textbf{cover})\\
&D_X(t) + D_Y(t) \leq 1, \forall t, \forall X \neq Y \in \mathcal{E}(G) \ (\textbf{disjointness})\\
&\sum_t \ D_Z(t) \leq c_Z, \forall Z \in \mathcal{E}(G)
\end{aligned} \tag{3}$$

Relaxations to Ensure Feasibility and Enhance Efficiency. Due to the rigorous constraints **(cover)** and **(disjointness)**, the solver on real-world datasets commonly finds Eq. 2 and 3 to be *infeasible*, and the problem for whether VTAE is feasible is computationally *intractable* (see Sect. 3).

Inspired by [12] we could have two intuitive solutions to side-step the infeasibility issue: **(1)** allow limited overlaps between descriptions and **(2)** "cover or forget" edges to guarantee disjointness. Note that neither of these changes makes the VTAE problem (with either disjointness) tractable. For example for limited overlapping descriptions Sect. 2 of the supplementary material (see Footnote 4) shows that problem is also intractable (Theorem 2). We discuss ILP formulations for these two relaxations in the supplementary material (see Footnote 4) due to their limited value of practical use. In particular, **minimizing overlap** yields significant overlap between the descriptions in our empirical study and **cover or forget** demands an auxiliary binary variable to identify whether *each* edge needs to be forgotten, rendering a prohibitively large search space for our edge covering.

We propose a **divide and conquer** Algorithm 1 to explore the idea of selectively removing tags from the tag universes such that the tag universes are *disjoint* and each edge set covering subproblem can be solved *independently*. Firstly, we use Eq. 4 and 5 to find binary vector $R_Z(t), \forall Z \in \mathcal{E}(G)$, where $R_Z(t) = 1$ `iff` t needs to be removed from U_Z to make the tag universes disjoint. Let u^* be the minimal feasible function value of Eq. 4, we set $u = u^*$ in Eq. 5 to find tags to exclude with small impact on the coverage of edges. Then we set the t-th *column* in matrix L_Z to all-zero if $R_Z(t) = 1$. The edges whose *rows* in L_Z turn all-zero are removed from the following covering problem. Figure 3 shows how to preprocess the example in Fig. 1. Finally, on each preprocessed edge set Z we learn description using the most frequently used tags with Eq. 7 constrained by the optimal succinctness discovered by Eq. 6. Our algorithm effectively reduces the search space and scales well in practice. The independency between subproblems can facilitate parallel implementation.

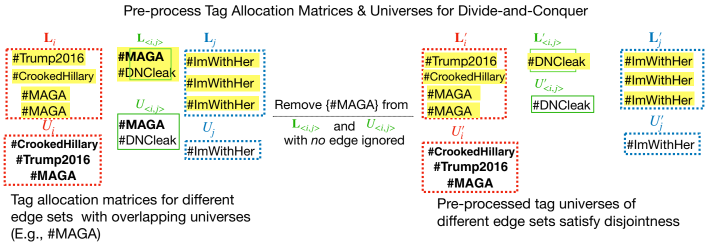


Fig. 3. Preprocess tag allocation matrices of the example in Fig. 1.

Algorithm 1: Solve VTAE with Divide-and-Conquer.

Input: Unweighted simple graph $G(V, E)$, vertex tags L, block structure $\mathcal{V} = \{V_1, V_2, \ldots, V_k\}$

Precompute tag allocation matrices $L_Z, \forall Z \in \mathcal{E}(G)$ based on G and $\mathcal{V}$;
Solve Eq. 4 for the minimal upper-bound of tags to exclude u^* to achieve tag universe disjointness ;
Solve Eq. 5 with u^* in **(upper-bound)** for the tags to exclude: $R^*_Z(t)$;
Exclude $R^*_Z(t)$ from each row of L_Z respectively ;
Remove empty rows from $L_Z, \forall Z \in \mathcal{E}(G)$;
while $Z \in \mathcal{E}(G)$ **do**
    Solve Eq. 6 for D_Z, set $d \leftarrow \sum_t D_Z(t)$;
    Solve Eq. 7 for D^*_Z ;
end
return $D^*_Z, \forall Z \in \mathcal{E}(G)$

$$
\begin{aligned}
&\underset{R_Z(t)\in\{0,1\}, Z\in\mathcal{E}(G)}{Minimize} \quad u \\
&s.t. \sum_t R_Z(t) \le u, \forall Z \in \mathcal{E}(G) \ (\mathbf{upper-bound}) \\
&\quad U_X(t) \times (1 - R_X(t)) + U_Y(t) \times (1 - R_Y(t)) \le 1, \\
&\quad\quad \forall t, \forall X \ne Y \in \mathcal{E}(G) \ (\mathbf{disjointness})
\end{aligned}
\tag{4}
$$

$$
\begin{aligned}
&\underset{R_Z(t)\in\{0,1\}, Z\in\mathcal{E}(G)}{Minimize} \quad \sum_Z \sum_t w_Z(t) \times R_Z(t) \\
&s.t. \sum_t R_Z(t) \le u, \forall Z \in \mathcal{E}(G) \ (\mathbf{upper-bound}) \\
&\quad U_X(t) \times (1 - R_X(t)) + U_Y(t) \times (1 - R_Y(t)) \le 1, \\
&\quad\quad \forall t, \forall X \ne Y \in \mathcal{E}(G) \ (\mathbf{disjointness})
\end{aligned}
\tag{5}
$$

$$
\underset{D_Z(t)\in\{0,1\}}{Minimize} \ \sum_t D_Z(t) \quad s.t. \quad \sum_t D_Z(t) \times L_Z(e,t) \ge 1, \quad \forall e \in Z \quad (\mathbf{cover}) \tag{6}
$$

$$\underset{D_Z(t)\in\{0,1\}}{Maximize} \ \sum_t \ D_Z(t) \times w_Z(t)$$
$$s.t. \sum_t D_Z(t) \times L_Z(e,t) \geq 1, \ \ \forall e \in Z \ \ (\mathbf{cover}); \sum_t D_Z(t) \leq d \ \ (\mathbf{upper-bound}) \tag{7}$$

5 Experiments

We evaluate our methods, VTAE when the disjointness is **global** (VTAE-g) or **partial** (VTAE-p), on three real-world datasets from different application domains: i) Twitter data with *two* graphs, ii) the BlogCatalog dataset[5] [26,41], and iii) graphs constructed from brain imaging data (fMRI of BOLD) from ADNI[6]. The details for the construction of the Twitter graphs and the brain imaging graphs are in the supplementary material (see Footnote 4). We summarize their basic statistics in Table 3. On these real-world large datasets, we aim to address the following questions:

Table 3. Statistics for the datasets.

Dataset	Twitter (smaller)	Twitter (larger)	BlogCatalog	Brain imaging graphs
# of Nodes	880	10,000	5,196	1,730
# of Edges	73,136	2,301,732	171,743	575,666.05 (mean)
# of Tags	136	136	8,189	1,730

Q1. *Novelty in description:* Can our descriptions offer novel insights different from existing methods for cluster description (i.e., [12])?
Q2. *Scalability:* How well does our method scale to large datasets?
Q3. *Usefulness for model selection:* Can our descriptions better represent the similarity/difference between different block models on the same graph?
Q4. *Robustness to perturbations in block allocation:* Is our description sensitive to small perturbations in block allocation of nodes?
Q5. *Description for historically/scientifically known model:* Is our method useful in situations where the block structure is not a result of an algorithm but historically/scientifically known?
Q6. *Stability/diversity w.r.t. graph topology:* When the block structure is fixed but graph topology changes, can our descriptions demonstrate *stability/diversity* consistent with domain knowledge?

[5] http://people.tamu.edu/~xhuang/BlogCatalog.mat.zip.
[6] http://adni.loni.usc.edu/study-design/collaborative-studies/dod-adni.

We address **Q1** on the smaller Twitter graph, **Q2** and **Q3** on the larger Twitter graph. We explore **Q4** on the larger Twitter graph and the BlogCatalog dataset. To demonstrate the versatility of our method, we explore **Q5** and **Q6** on the brain imaging graphs where the labels are not explicitly given. The experiments are conducted on a machine with 4-core Intel Xeon CPU. The ILP formulations are solved by `Gurobi` (https://www.gurobi.com/) in `Python`.

Precompute Block Model with NMtF. For Twitter and BlogCatalog data we use Nonnegative Matrix tri-Factorization (NMtF) which is not jointly convex with multiplicative update rules in seminal work [15] to generate block models (block allocation $\mathbf{F}$, image matrix $\mathbf{M}$) on the adjacency matrix of a graph $\mathbf{G}$.

Scientifically Known Block Structure. For the fMRI data we use the well-known default mode network (DMN) that divides the brain into multiple interacting regions. We consider two regions - i.e., foreground and background as explored in [8,9]. See supplementary material (see Footnote 4) for visualization of DMN and the two-region partitioning.

Baseline. We choose the **cover or forget** relaxation of DTDM [12] (DTDM-`cof`) to find a succinct and distinct description D_i, a $|\Gamma|$-dimensional binary vector for vertex cluster $V_i,\ i = 1, \ldots, k$. Follow [12] we define and precompute binary matrices L_i for vertex cluster V_i where $L_i(v,t) = 1$ `iff` vertex v is assigned to cluster $V_i, 1 \leq i \leq k$ and is associated with tag t. We use $|V|$ binary variables f_v, each marks whether a vertex v needs to be forgotten or not. The number of forgotten vertices (i.e., $\sum\limits_{v \in V} f_v$) is part of the minimization objective (Eq. 8).

$$\begin{aligned} &\underset{D_i(t)\in\{0,1\}, f_v\in\{0,1\}}{Minimize} \sum_i \sum_t D_i(t) + \sum_{v\in V} f_v \\ &s.t.\ f_v + \sum_t D_i(t) \times L_i(v,t) \geq 1, \forall v \in V_i, \forall i;\ D_i(t) + D_j(t) \leq 1, \forall t, \forall i \neq j \end{aligned} \tag{8}$$

Answering Q1. We simplified the smaller Twitter graph (Table 3) into 4 blocks. Upon examination of the members in each block (see supplementary material (see Footnote 4)) we observe a distinct political polarization as expected [11]. We then describe which hashtags are commonly used over the edges within a block $E_1, \ldots, E_4$ and between blocks $E_{1,2}, E_{2,3}, \ldots, E_{3,4}$ (see Fig. 4). Compared to Table 5 the edges demonstrate higher complexity than the vertices in the sense of associated content (i.e., can demand more tags in a description upon minimal length). Our VTAE-`g` generates description for each **intra-**block edge set that drastically differs from its corresponding vertex block description of baseline method DTDM [12]. See VTAE-`p` results in the supplementary material (see Footnote 4).

Table 4. The statistics of interacting blocks found using block modeling [15] for us to describe. See supplementary material (see Footnote 4) for full member list in each block.

Block	Prominent members	Size	Intra. edges	Ext. edges
1: Trump supporters	Trump Train, Italians4Trump	184	7,332	20,646
2: Clinton and supporters	Hillary Clinton, WeNeedHillary	227	5,288	21,566
3: Other-candidates	Ted Cruz, John Kasich	309	10,756	23,307
4: Trump inner circle	DonaldTrump, DonaldTrumpJr	160	5,880	20,931

Table 5. Vertex cluster descriptions of DTDM-`cof` [12] (Eq. 8) for Table 4.

Block	Vertex cluster description
$\mathbf{V}_1$	`GOPdebate`, `DNCinPHL`, `Clinton`, `GOPDEBATE`, `BlackLivesMatter`, `MAGA`, `Sanders`, `CrookedHillary`, `Hannity`, `UniteBlue`, `CCOT`
$\mathbf{V}_2$	`NeverTrump`, `DemsInPhilly`, `GOP`, `1`, `SCPrimary`, `FeelTheBern`, `PrimaryDay`, `DemTownHall`, `FITN`, `NewYork`, `DumpTrump`
$\mathbf{V}_3$	`Trump`, `RNCinCLE`, `SuperTuesday`, `IowaCaucus`, `NYPrimary`, `DonaldTrump`, `Election2016`, `MakeAmericaGreatAgain`, `TrumpRally`, `NewYorkValues`, `DNCleak`, `USA`, `trump2016`
$\mathbf{V}_4$	`DemDebate`, `Trump2016`, `NHPrimary`, `trump`, `ImWithHer`, `iacaucus`, `BernieSanders`, `TRUMP`, `CruzSexScandal`

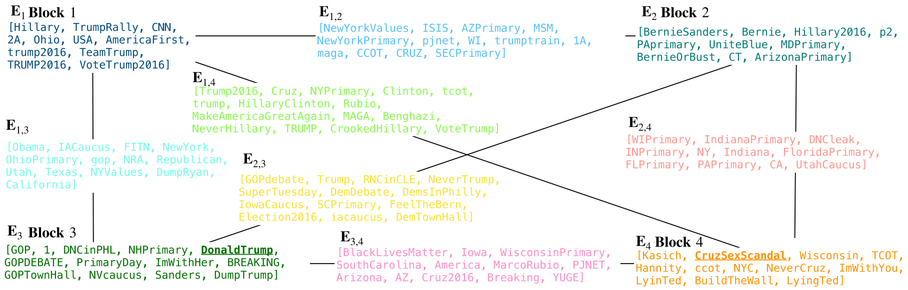
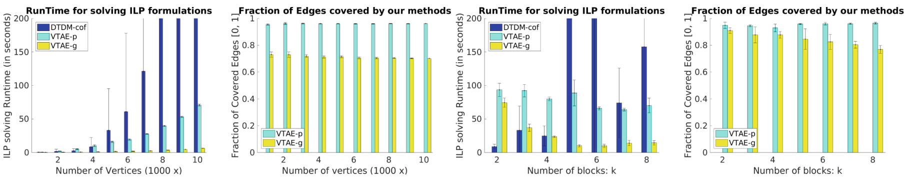


Fig. 4. Edge set explanations discovered by VTAE-`g`. Underlined are tags used by both our VTAE-`g` and DTDM-`cof` [12] for the same block (edges within and the vertices) in Table 5. Our result and the baseline use very few tags in common.



(a) Runtime vs n. (b) Fraction of covered edges vs n. (c) Runtime vs k. (d) Fraction of covered edges vs k.

Fig. 5. Runtime of our methods (and DTDM-`cof` [12]) on Twitter (larger).

(a) DTDM-`cof` [12] (b) VTAE-`g` (c) VTAE-`p` (d) **F**

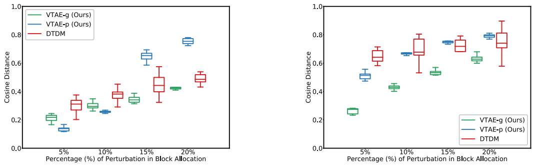
Fig. 6. Similarity matrices at $k = 7$. Corresponds to $k = 7$ in Table 6. Visually our method VTAE-`p` better represents the similarity/difference between **F**'s than DTDM [12].

Answering Q2. We address this problem on the larger Twitter graph (Table 3). We explore both the effects of the number of sampled vertices given a fixed 5-way block model (Fig. 5a, 5b) and k the number of blocks (Fig. 5c, 5d). For the former we randomly sample 10 times from the entire 10^4 individuals at each given number of vertices. For the latter we generate 10 block models ($\{\mathbf{F}, \mathbf{M}\}$'s) with random initializations at each k with the NMtF formulation [15]. We show that the combined runtime of ILP formulations in our Algorithm 1 are comparable to earlier work [12] despite finding $k + \frac{k(k-1)}{2}$ descriptions for *edge* sets versus k descriptions for *vertex* clusters. Such runtime efficiency of our methods is achieved by marginally ignoring edges (Fig. 5d, 5b). Particularly, our method can explain the block modeling results on this large Twitter graph within 2 min, ignoring $\leq 5\%$ edges for VTAE-`p` given an 8-block model.

Answering Q3. To be useful for model selection, the descriptions should represent the similarity/difference between different block models for the same graph. We investigate the descriptions generated in answering **Q2** on the large Twitter graph. For the 10 block models precomputed at each k, we measure the cosine similarity between the descriptions of each pair of block models generated by our VTAE-`p`, VTAE-`g`, and the baseline method DTDM [12], respectively. Each yields a 10×10 similarity matrix $\mathbf{S}^k_{[method]}$. We compute $\mathbf{S}^k_g$ for **F**'s as ground-truth. Since a block allocation is unique up to permutation, all the similarities are measured after aligning the descriptions and **F**'s according to their corresponding image matrices **M**'s. We visualize the results at $k = 7$ in Fig. 6 and present $\|\mathbf{S}^k_{[method]} - \mathbf{S}^k_{\mathbf{F}}\|_F$ in Table 6 for each method at $k = 2, \ldots, 8$. We do not use NMI or ACC to measure the similarity between **F**'s as neither provides useful information to align the **inter-**block descriptions for a fair comparison.

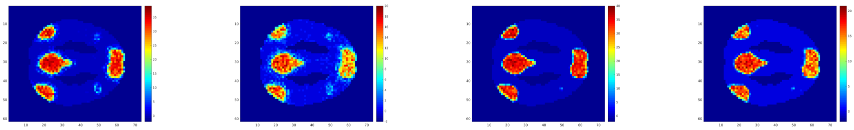
Table 6. Difference between similarity matrices of each method and **F**'s $\|\mathbf{S}^k_{[method]} - \mathbf{S}^k_{\mathbf{F}}\|_F$. Our VTAE-`p` outperforms DTDM [12] at $k = 5, 6, 7$.

k	2	3	4	5	6	7	8
DTDM [12]	**0.7045**	**1.2834**	**1.6660**	2.3027	2.1325	2.5608	**1.9610**
Our VTAE-`g`	1.1270	2.0602	2.8341	3.6892	3.3882	4.0660	3.4347
Our VTAE-`p`	1.2043	1.8109	1.6930	**1.2069**	**1.9389**	**1.0781**	2.5427



(a) Twitter (larger) (b) BlogCatalog

Fig. 7. Changes in descriptions due to perturbations in block allocation.



(a) VTAE-p (**Y**) 40 (b) VTAE-p (**D**) 21 (c) VTAE-g (**Y**) 40 (d) VTAE-g (**D**) 21

Fig. 8. Heat maps for subjects using voxels in their descriptions.

Answering Q4. We select one block model generated at $k = 4$ for the larger Twitter data and generate block model at $k = 6$ (i.e., the number of ground-truth classes) for the BlogCatalog data. We artificially introduce different levels of perturbations in its $\mathbf{F}$: We change the block membership of randomly selected $p = 5\%, 10\%, 15\%, 20\%$ of nodes. We measure the *cosine distance* between the descriptions learnt for perturbed block models and the original block model (Fig. 7). Our method demonstrates stronger robustness than [12] against small perturbations in most cases.

Answering Q5 and Q6. We generate vertex tags with the 10 nearest neighbors of these vertices on the correlation graphs, each graph measures how much the BOLD signals of each two voxels correlate. Our method then explains the *strong* interactions of the default mode cognitive network in terms of the nearest neighbors to it on the correlation graph. We summarize the descriptions at the cohort level - i.e., control cohort ($\mathbf{Y}$) and Alzheimer's affected cohort **(D)**, as *heat maps* in Fig. 8. The color of each pixel indicates the number of subjects using it in their descriptions in a cohort. Red means many subjects using this subject. Figure 8a and Fig. 8c show that our descriptions for the control cohort subjects are confined within the DMN region. Figure 8b and Fig. 8d show our descriptions for the Alzheimer's affected group with less well-defined regions. Importantly, our method generates *stable* descriptions for the control cohort despite having 40 subjects, while our descriptions for the demented cohort with 21 are highly *diverse*. Note our method does not explicitly distinguish subjects from different cohorts. This *stability/diversity* result is consistent with the neuroscience domain knowledge that similar intra-DMN interactions exist amongst the young and healthy subjects from the control cohort whilst the Alzheimer's affected subjects can have demented DMN interactions for *varied* causes.

6 Conclusion

In this paper we have researched a novel problem of describing a given block modeling result on a graph by using labels or tags associated to the nodes. These nodal tags are not utilized in the block modeling process. We have formally defined the problem as Valid Tag Assignment to Edges (VTAE) to find descriptions for both the intra-block edge sets and the inter-block edge sets. The descriptions of different edge sets are built with disjoint subsets of tags. We show that the decision version of VTAE is intractable but relaxations exist and can be efficiently solved by existing ILP solver `Gurobi`. Experiments on Twitter data show that even for the same block modeling result on the same graph, our formulations can generate descriptions significantly different from the baseline approach for cluster description [12]. Our numerical evaluations on Twitter and BlogCatalog datasets demonstrate advantages of our method over the baseline method [12]. In some settings, our descriptions can i) better represent the similarity/difference between different block models on the same graph and ii) be more robust against small perturbations in block allocation. On graphs built with brain imaging data, we show the versatility of our method as it can be useful even when the tags are not explicitly given but generated with nearest neighbors on a weighted graph. Experimental results on this dataset also show that the distribution of our descriptions is consistent with domain knowledge at the cohort-level. In the future, we aim to derive extensions of existing models to enhance the block modeling performance in more challenging settings, such as large noisy and incomplete real-world graphs.

Acknowledgments. This research is supported by ONR Grant N000141812485 and NSF Grants IIS-1910306, IIS-1908530, OAC-1916805, ACI-1443054 (DIBBS), IIS-1633028 (BIG DATA) and CMMI-1745207 (EAGER).

References

1. Abbe, E.: Community detection and stochastic block models: recent developments. J. Mach. Learn. Res. **18**(1), 6446–6531 (2017)
2. Adadi, A., Berrada, M.: Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE Access **6**, 52138–52160 (2018)
3. Akar, E., Mardikyan, S.: User roles and contribution patterns in online communities: a managerial perspective. Sage Open **8**(3), 2158244018794773 (2018)
4. Atzmueller, M.: Descriptive community detection. In: Missaoui, R., Kuznetsov, S.O., Obiedkov, S. (eds.) Formal Concept Analysis of Social Networks. LNSN, pp. 41–58. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-64167-6_3
5. Atzmueller, M., Doerfel, S., Mitzlaff, F.: Description-oriented community detection using exhaustive subgroup discovery. Inf. Sci. **329**, 965–984 (2016)
6. Atzmueller, M., Mitzlaff, F.: Efficient descriptive community mining. In: FLAIRS (2011)
7. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLoS One **10**(7), e0130140 (2015)

8. Bai, Z., Qian, B., Davidson, I.: Discovering models from structural and behavioral brain imaging data. In: SIGKDD, pp. 1128–1137 (2018)
9. Bai, Z., Walker, P., Tschiffely, A., Wang, F., Davidson, I.: Unsupervised network discovery for brain imaging data. In: SIGKDD, pp. 55–64 (2017)
10. Chabert, M., Solnon, C.: Constraint programming for multi-criteria conceptual clustering. In: Beck, J.C. (ed.) CP 2017. LNCS, vol. 10416, pp. 460–476. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66158-2_30
11. Conover, M.D., Ratkiewicz, J., Francisco, M., Gonçalves, B., Menczer, F., Flammini, A.: Political polarization on Twitter. In: ICWSM (2011)
12. Davidson, I., Gourru, A., Ravi, S.: The cluster description problem-complexity results, formulations and approximations. In: NIPS, pp. 6190–6200 (2018)
13. Deshpande, Y., Sen, S., Montanari, A., Mossel, E.: Contextual stochastic block models. In: NIPS, pp. 8581–8593 (2018)
14. Dhurandhar, A., et al.: Explanations based on the missing: towards contrastive explanations with pertinent negatives. In: NIPS, pp. 592–603 (2018)
15. Ding, C., Li, T., Peng, W., Park, H.: Orthogonal nonnegative matrix t-factorizations for clustering. In: SIGKDD, pp. 126–135 (2006)
16. Došilović, F.K., Brčić, M., Hlupić, N.: Explainable artificial intelligence: a survey. In: 2018 41st MIPRO, pp. 0210–0215. IEEE (2018)
17. Falih, I., Grozavu, N., Kanawati, R., Bennani, Y.: Community detection in attributed network. In: WWW, pp. 1299–1306 (2018)
18. Fisher, D.H.: Knowledge acquisition via incremental conceptual clustering. Mach. Learn. **2**(2), 139–172 (1987). https://doi.org/10.1007/BF00114265
19. Fortunato, S.: Community detection in graphs. Phy. Rep. **486**(3–5), 75–174 (2010)
20. Funke, T., Becker, T.: Stochastic block models: a comparison of variants and inference methods. PLoS One **14**(4), e0215296 (2019)
21. Galbrun, E., Gionis, A., Tatti, N.: Overlapping community detection in labeled graphs. Data Min. Knowl. Disc. **28**(5), 1586–1610 (2014). https://doi.org/10.1007/s10618-014-0373-y
22. Ganji, M., et al.: Image constrained blockmodelling: a constraint programming approach. In: SDM, pp. 19–27 (2018)
23. Garey, M.R., Johnson, D.S.: Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York (1990)
24. Girvan, M., Newman, M.E.: Community structure in social and biological networks. PNAS **99**(12), 7821–7826 (2002)
25. Guns, T., Nijssen, S., De Raedt, L.: k-Pattern set mining under constraints. IEEE TKDE **25**(2), 402–418 (2011)
26. Huang, X., Li, J., Hu, X.: Label informed attributed network embedding. In: WSDM, pp. 731–739 (2017)
27. Kao, H.T., Yan, S., Huang, D., Bartley, N., Hosseinmardi, H., Ferrara, E.: Understanding cyberbullying on Instagram and Ask.fm via social role detection. In: WWW, pp. 183–188 (2019)
28. Kotthoff, L., O'Sullivan, B., Ravi, S., Davidson, I.: Complex clustering using constraint programming: Modelling electoral map (2015)
29. Li, D., et al.: Community-based topic modeling for social tagging. In: CIKM (2010)
30. Müller, B., Reinhardt, J., Strickland, M.T.: Neural Networks: An Introduction. Springer Science & Business Media, Heidelberg (2012)
31. Newman, M.E.: Modularity and community structure in networks. PNAS **103**(23), 8577–8582 (2006)
32. Pool, S., Bonchi, F., Leeuwen, M.V.: Description-driven community detection. TIST **5**(2), 1–28 (2014)

33. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why should I trust you?" explaining the predictions of any classifier. In: SIGKDD, pp. 1135–1144 (2016)
34. Rossi, R.A., Ahmed, N.K.: Role discovery in networks. IEEE TKDE **27**(4), 1112–1131 (2014)
35. Sambaturu, P., Gupta, A., Davidson, I., Ravi, S., Vullikanti, A., Warren, A.: Efficient algorithms for generating provably near-optimal cluster descriptors for explainability. In: AAAI, pp. 1636–1643 (2020)
36. Samek, W., Montavon, G., Vedaldi, A., Hansen, L.K., Müller, K.-R. (eds.): Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. LNCS (LNAI), vol. 11700. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-28954-6
37. Stanley, N., Bonacci, T., Kwitt, R., Niethammer, M., Mucha, P.J.: Stochastic block models with multiple continuous attributes. Appl. Netw. Sci. **4**(1), 1–22 (2019)
38. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: large-scale information network embedding. In: WWW, pp. 1067–1077 (2015)
39. Tang, J., Jin, R., Zhang, J.: A topic modeling approach and its integration into the random walk framework for academic search. In: IEEE ICDM (2008)
40. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: Arnetminer: extraction and mining of academic social networks. In: SIGKDD, pp. 990–998 (2008)
41. Tang, L., Liu, H.: Relational learning via latent social dimensions. In: SIGKDD, pp. 817–826 (2009)
42. Yang, J., McAuley, J., Leskovec, J.: Community detection in networks with node attributes. In: IEEE ICDM, pp. 1151–1156 (2013)
43. Zhang, D., Yin, J., Zhu, X., Zhang, C.: Network representation learning: a survey. IEEE Trans. Big Data **6**, 3–28 (2018)