Towards Description of Block Model on Graph

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

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

Abstract. This is the supplementary material for paper entitled "Towards Description of Block Model on Graph". In this material, section 1 presents the proof of intractability for the problem of VTAE with partial disjointness (VTAE-p). Section 2 shows that a relaxation of VTAE-p with limited overlapping descriptions is also intractable. [1] Section 3 presents intuitive relaxations to side-step the *infeasibility* issue and experiments to demonstrate practical issue. Section 4 presents more details for the experimental section.

1 Proof for Complexity Results

Theorem 1. *The VTAE problem is **NP**-complete even the graph $G(V, E)$ is a tree and the node set is partitioned into just two subsets.*

Proof: It is easy to see that VTAE is in **NP**, since one can guess a tag assignment function τ and efficiently verify that it satisfies both the compatibility and disjointness properties.

To prove **NP**-hardness, we use a reduction from 3SAT which is known to be **NP**-complete [4]. Let x_1, x_2, ..., x_n denote the n variables and Y_1, Y_2, ..., Y_m denote the m clauses of the 3SAT instance. The reduction to the VTAE problem is as follows.

1. For each variable x_i, we create two tags denoted by α_i and β_i, $1 \leq i \leq n$. (The tags α_i and β_i correspond to the literals x_i and $\overline{x_i}$ respectively.) We also create an additional tag denoted by λ. The set of all tags Γ is given by $\Gamma = \{\alpha_1, \beta_1, \ldots, \alpha_n, \beta_n, \lambda\}$. (Thus, $|\Gamma| = 2n + 1$.)
2. For each variable x_i, we also create two nodes, denoted by a_i and b_i, and the edge $\{a_i, b_i\}$, $1 \leq i \leq n$. The tag sets $T(a_i)$ and $T(b_i)$ are chosen as follows: $T(a_i) = \{\alpha_i, \beta_i\}$ and $T(b_i) = \{\alpha_i, \beta_i, \lambda\}$, $1 \leq i \leq n$.

[1] VTAE with global disjointness is intractable as the global disjointness essentially adds additional constraints to the partial disjointness, and we prove here that VTAE with partial disjointness to be intractable even for simple cases where G is a tree or $k = 2$. Therefore, we focus on proving VTAE-p to be intractable in this document and refer to VTAE-p as VTAE in sections 1 and 2.

3. For each clause Y_j, we create two nodes, denoted by u_j and v_j, and the edge $\{u_j, v_j\}$, $1 \leq j \leq m$. The tag sets $T(u_j)$ and $T(v_j)$ are chosen as follows. Consider clause Y_j and suppose the literals appearing in Y_j correspond to variables x_p, x_q and x_r. For each such variable x_i ($i \in \{p, q, r\}$), if x_i appears as an unnegated literal, then tag α_i appears in both $T(u_i)$ and $T(v_i)$; if x_i appears as a negated literal, then tag β_i appears in both $T(u_i)$ and $T(v_i)$. In addition, tag λ is added to each of the tag sets $T(u_j)$, $1 \leq j \leq m$. (Thus, for $1 \leq j \leq m$, $|T(u_j)| = 4$ and $|T(v_j)| = 3$.)
4. We create an additional node denoted by d. The tag sets for d is given by $T(d) = \{\lambda\}$. (The purpose of adding node d is to ensure that the underlying graph is a tree.)
5. We add the following edges: (i) for $1 \leq i \leq n$, the edge $\{d, b_i\}$ and (ii) for $1 \leq j \leq m$, the edge $\{d, u_j\}$.
6. The graph $G(V, E)$ consists of all the nodes and edges created in the above steps. The node set V is partitioned into V_1 and V_2, where

$$V_1 = \{a_1, b_1, a_2, b_2, \ldots, a_n, b_n, d, u_1, u_2, \ldots, u_m\} \quad \text{and} \quad V_2 = \{v_1, v_2, \ldots, v_m\}.$$

It can be verified that the above construction can be carried out in polynomial time and that the resulting graph $G(V, E)$ is a tree. Further, since V is partitioned into just two sets, the construction produces one intra-block edge set E_1 and one inter-block edge set $E_{1,2}$. (The intra-block edge set E_2 is empty; there is no edge between any pair of nodes in V_2.)

An example of the graph and tag sets produced by this construction is shown in Figure 1. We now show that there is a solution to the resulting VTAE instance iff there is a solution to the 3SAT instance.

Suppose there is a solution to the 3SAT instance. We assign a tag to each edge as follows.

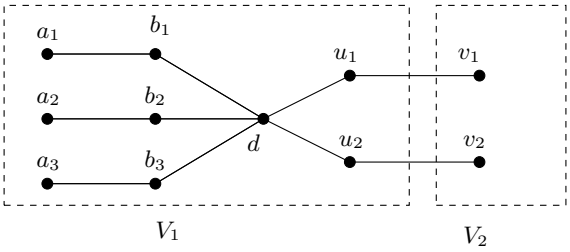
(a) Consider each edge $\{a_i, b_i\}$, $1 \leq i \leq n$. If x_i is assigned the value `True`, then the tag for edge $\{a_i, b_i\}$ is β_i; otherwise, the tag for the edge is α_i.
(b) For each edge $\{b_i, d\}$, the tag is λ, $1 \leq i \leq n$.
(c) For each edge $\{d, u_j\}$, the tag is λ, $1 \leq j \leq m$.
(e) Consider each edge $\{u_j, v_j\}$, $1 \leq j \leq m$. The given satisfying assignment causes some literal, say of variable x_p, of the corresponding clause Y_j to be `True`. If x_p occurs unnegated in Y_j, we choose the tag of $\{u_j, v_j\}$ to be α_p; otherwise, we choose the tag of that edge to be β_p.

It can be verified that the resulting tag assignment satisfies both the compatibility and disjointness properties; that is, the tag assignment is valid. Thus, there is a solution to the VTAE instance.

For the converse, assume that there is a solution to the VTAE instance. Let E' denote the edge set $\{\{a_i, b_i\} : 1 \leq i \leq n\}$ and E'' denote the edge set $\{\{u_j, v_j\} : 1 \leq j \leq m\}$. We have the following claim.

Claim 1: For each i, $1 \leq i \leq n$, the set of tags assigned to the edges in E'' *cannot* contain both α_i and β_i.

Example: The 3SAT instance consists of three Boolean variables x_1, x_2 and x_3 and two clauses $Y_1 = (\overline{x_1} \vee x_2 \vee \overline{x_3})$ and $Y_2 = (x_1 \vee \overline{x_2} \vee x_3)$. The node set V of the graph $G(V, E)$ resulting from the reduction is partitioned into two subsets V_1 and V_2. The resulting graph and the tag sets of nodes are shown below.



Node	**Tag set**
a_1	$\{\alpha_1, \beta_1\}$
a_2	$\{\alpha_2, \beta_2\}$
a_3	$\{\alpha_3, \beta_3\}$
b_1	$\{\alpha_1, \beta_1, \lambda\}$
b_2	$\{\alpha_2, \beta_2, \lambda\}$
b_3	$\{\alpha_3, \beta_3, \lambda\}$
d	$\{\lambda\}$
u_1	$\{\beta_1, \alpha_2, \beta_3, \lambda\}$
u_2	$\{\alpha_1, \beta_2, \alpha_3, \lambda\}$
v_1	$\{\beta_1, \alpha_2, \beta_3\}$
v_2	$\{\alpha_1, \beta_2, \alpha_3\}$

Fig. 1: An Example to Illustrate the Reduction Used to Prove Theorem 1.

Proof of Claim 1: The proof is by contradiction. Suppose the set of tags used for E'' contains both α_i and β_i for some i. Because of the chosen partition of the nodes, the set of tags used for the edges in E'' must be disjoint from those used for the edges in E'. Thus, neither α_i nor β_i can be used as a tag for any edge in E'. Consider the edge $\{a_i, b_i\} \in E'$. Since $T(a_i) = \{\alpha_i, \beta_i\}$ and $T(b_i) = \{\alpha_i, \beta_i, \lambda\}$, the tag assigned to $\{a_i, b_i\}$ must be either α_i or β_i. This contradicts the previous conclusion that neither α_i nor β_i can be used as a tag for any edge in E'. This completes our proof of Claim 1.

We construct a truth assignment to the Boolean variables x_1, x_2, ..., x_n as follows. Consider variable x_i. If any of the edges in E' is assigned the tag α_i, we set x_i to `True`; otherwise, we set x_i to `False`. This method assigns different truth values to the literals x_i and $\overline{x_i}$, $1 \leq i \leq n$ since by Claim 1 at most one of α_i and β_i appears as a tag for the edges in E''. We now show that this is a satisfying assignment. To see this, consider any clause Y_j and suppose it contains unnegated or negated literals corresponding to the variables x_p, x_q and x_r. The edge corresponding to Y_j is $\{u_j, v_j\}$. If x_p appears unnegated in Y_j, tag sets $T(u_j)$ and $T(v_j)$ both include α_p; otherwise both include β_p. A similar comment

holds for the other two variables x_q and x_r. In other words, the tag assigned to the edge $\{u_j, v_j\}$ must be one of α_p, β_p, α_q, β_q, α_r, β_r. Suppose the tag assigned to $\{u_j, v_j\}$ is α_p. By our construction, x_p appears unnegated in Y_j. Further, since we set x_p to `True`, this satisfies Y_j. A similar argument holds for the five other possible tags that may be assigned to the edge $\{u_j, v_j\}$. This completes our proof of Theorem 1. □

2 A Complexity Result for a Variant of VTAE

We now consider a variant of VTAE obtained by relaxing the disjointness requirement. Given a nonnegative integer δ, we say that a tag assignment function τ is **weakly δ-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) **[Limited overlap]** For any pair of edge sets X and Y, where X is an intra-block edge set and Y is an inter-block edge set, $|\tau(X) \cap \tau(Y)| \leq \delta$; that is, for intra-block edge set X and any inter-block edge set Y, the sets of tags assigned to X and Y have at most δ tags in common. (As before, there is no restriction on the overlap of the tag sets associated with two intra-block edge sets or two inter-block edge sets.)

We note that this is a relaxed form of validity; when $\delta = 0$, we have the notion of validity considered in the previous section. The decision problem for this relaxed form of validity is as follows.

Weakly δ-Valid Tag Assignment to Edges (WVTAE)

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

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

By modifying the proof of Theorem 1, we now show that the WVTAE problem is also **NP**-complete.

Theorem 2. *For any fixed $\delta \geq 0$, the WVTAE problem is **NP**-complete even the graph $G(V, E)$ is a tree and the node set is partitioned into just two subsets.*

Proof: For $\delta = 0$, we already know from Theorem 1 that the problem is **NP**-complete. So, we will assume that $\delta \geq 1$.

It is easy to see that WVTAE is in **NP**. To prove **NP**-hardness, we use the reduction from 3SAT discussed in the proof of Theorem 1 with the following modifications.

1. We add δ additional pairs of nodes a_{n+i}, b_{n+i} to the subset V_1 and the edges $\{a_{n+i}, b_{n+i}\}$, $1 \leq i \leq \delta$. The tag set for $T(a_{n+i}) = \{\rho_i\}$ and that for $T(b_{n+i}) = \{\rho_i, \lambda\}$, $1 \leq i \leq \delta$. (Here, $\rho_1, \ldots, \rho_\delta$ are new tags.)

2. We add δ additional pairs of nodes u_{m+i}, v_{m+i} and the edges $\{u_{m+i}, v_{m+i}\}$, $1 \leq i \leq \delta$. The tag set for $T(u_{n+i}) = \{\rho_i, \lambda\}$ and that for $T(v_{n+i}) = \{\rho_i\}$, $1 \leq i \leq \delta$.
3. We also add the following edges: (i) $\{b_{n+i}, d\}$, $1 \leq i \leq \delta$ and (ii) $\{d, u_{m+i}\}$, $1 \leq i \leq \delta$.
4. The node sets V_1 and V_2 are as follows.

$$V_1 = \{a_1, b_1, a_2, b_2, \ldots, a_{n+\delta}, b_{n+\delta}, d, u_1, u_2, \ldots, u_{m+\delta}\}$$
$$\text{and} \quad V_2 = \{v_1, v_2, \ldots, v_{m+\delta}\}.$$

It is easy to see that the graph remains a tree. Note that for any δ-weakly valid assignment of a label to each edge, the edge $\{a_{n+i}, b_{n+i}\}$ must have the tag ρ_i, since that is the only tag that appears in both $T(a_{n+i})$ and $T(b_{n+i})$, $1 \leq i \leq \delta$. Likewise, the edge $\{u_{m+i}, v_{m+i}\}$ must have the tag ρ_i, since that is the only tag that appears in both $T(u_{m+i})$ and $T(v_{m+i})$, $1 \leq i \leq \delta$. Therefore, the δ tags in $\{\rho_1, \rho_2, \ldots, \rho_\delta\}$ must appear in both the edge sets E_1 and $E_{1,2}$. Since the maximum allowed overlap between the tag sets used for E_1 and $E_{1,2}$ is δ, it follows that the tag set used for the edge set $E' = \{\{a_i, b_i\} \ : \ 1 \leq i \leq n\}$ and that for edge set $E'' = \{\{u_j, v_j\} \ : \ 1 \leq j \leq m\}$ cannot have any tags in common. The rest of the proof is similar to that presented in the proof of Theorem 1. □

3 Intuitive Relaxations of VTAE to Side-step Infeasibility Issue in Practice

We present here the ILP formulations of the two intuitive relaxations of VTAE to side-step its feasibility issue. We show in figure 2 the overlap between different descriptions by different relaxations of VTAE. We summarize the frequently used notations in table 1.

Table 1: Frequently used notations.

	Definition and Computation
$\mathcal{E}(G)$	*Edge set collection* of the $k + \frac{k(k-1)}{2}$ edge sets induced from graph $G(V, E)$ given a $k-$block block model.
Z, X, Y	*An edge set* in $\mathcal{E}(G)$. Partial Disjointness which requires X to be an intra-block edge set and Y to be an inter-block edge set. Global Disjointness which requires disjointness between all combinations of intra-block and inter-block edge sets, i.e., $\forall X \neq Y \in \mathcal{E}(G)$.
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` $e \in Z, \{x, y\} = e, t \in T(x) \cap T(y)$

1. **Overlapping Descriptions** Here we allow each tag to be used at most r times across all descriptions by replacing the disjointness constraints with the following constraint. We could even make r part of the objective function to minimize the disjointness.

$$\sum_{X \neq Y} D_X(t) + D_Y(t) \leq r, \forall t \quad \textbf{(relaxed-disjointness)} \tag{1}$$

2. **"Cover or Forget" Edges** Another relaxation of the problem is to allow some edges not to be covered but minimize such occurrences. This can be achieved by simply having for each instance in the cover constraint another variable $\lambda_e \in \{0, 1\}$ which if set to 1 means the edge e is ignored. The new relaxed formulation is:

$$\begin{aligned} &\underset{D_Z(t)\in\{0,1\}}{Minimize} \sum_{\forall Z \in \mathcal{E}(G), \forall t} D_Z(t) + \sum_e \lambda_e \\ &s.t. \sum_t \lambda_e + D_Z(t) L_Z(e,t) \geq 1 \\ &\quad \forall e \in Z, \forall Z \in \mathcal{E}(G) \ \textbf{(Cover-or-forget)} \\ &D_X(t) + D_Y(t) \leq 1 \quad \forall X, Y \in \mathcal{E}(G) \ \textbf{(disjointness)} \end{aligned} \tag{2}$$

3.1 Significant overlaps between descriptions induced by minimizing overlap (mo)

Here we demonstrate the issue with relaxation that allows **overlapping descriptions**. We replace the disjointness constraint from Minimal VTAE (equation 4.2 in the paper) with equation 1 and add r to the objective function. The results of this relaxation is denoted as VTAE-mo-**p** and VTAE-mo-g for **partial** or **global** disjointness respectively. Figure 2 shows that minimizing overlap between descriptions regardless of the disjointness constraint can induce significant overlap, which undermines the usefulness for this method in practical use.

Metric. Given an description result of our method in $D_Z, Z \in \mathcal{E}(G)$, we measure the overlap between descriptions of different edge sets with equation 3. Note that we measure overlap between descriptions of intra-block and inter-block edge sets as Overlap-p using X to denote intra-block edge sets and Y to denote inter-block edge sets. We measure the overlap between any different edge sets as Overlap-g with X and Y denoting any two different edge sets from $\mathcal{E}(G)$. Note this metric is *second-order* with respect to $D_Z, Z \in \mathcal{E}(G)$, and it is not directly optimized with our ILP formulation. Therefore, the overlap between inter-block and intra-block descriptions Overlap-p computed for the descriptions generated by VTAE-mo-**p** can be *higher* than VTAE-mo-g.

Experimental Setting. We evaluate the overlap generated by different relaxations (i.e., VTAE-p, VTAE-g solved by our algorithm 1 and VTAE-mo-p,

VTAE-mo-g) on the smaller Twitter graph over 880 individuals. We generate 10 block models with random initializations at each k with the NMtF formulation (which is not jointly convex) with seminal work [3].

$$Overlap(D_Z, Z \in \mathcal{E}(G)) = \sum_{X \neq Y \in \mathcal{E}(G)} \sum_{t} D_X(t) D_Y(t) \tag{3}$$

4 Details for the Experiments Section

4.1 Datasets

Twitter dataset. Our US dataset contains a subset of all the tweets published during the USA primary election period, between the 12/30/2015 and the 08/18/2016. We used the names of politicians[2] involved in the election at this time as query for the Twitter Stream API. We extract the retweet network as it has been shown to grasp political polarization [1] and the hashtags usage by user. We create an undirected edge if two individuals share a follower/followee relation. The dataset, has $3,448,096$ individuals, $2,515,421$ hashtags but many of those individuals rarely post so we limit ourselves to the 880 most influential posters in terms of followers for an illustrative experiment (i.e., the smaller Twitter graph) and a data set of $10,000$ individuals for scalability experiments (i.e., the larger Twitter graph). We consider 136 hashtags.

In this supplementary material folder, we enclose the smaller graph over 880 nodes and a 4-block model to demonstrate our program implementation of our VTAE-p and VTAE-g. The large datasets will be released upon the acceptance of the paper due to limit of submission file size.

BlogCatalog dataset. We use a publicly available BlogCatalog dataset [3] [5] [6] which provides a graph with $171,743$ edges over $5,196$ nodes and $8,189$ nodal attributes as tags. The nodes are users, edges are friendship relation between users, and tags are the keywords in the blog descriptions.

fMRI dataset. We apply our method to graphs constructed with fMRI scans from ADNI[4]. Each fMRI scan is a fourth order tensor in $x \times y \times z \times t$ with BOLD measure at time t for each voxel in the spatial domain at (x,y,z). BOLD measurement effectively measures the amount of activity at that part of the brain at a particular time. We construct a fully connected undirected graph from a scan by measuring the absolute Pearson Correlation between the temporal activations of each pair of voxels on the 36-th slice. On this slice there are $1,730$ voxels within the brain activity region. We keep edges with weights above 0.5 to make graphs unweighted.

[2] bush,carson,christie,cruz,fiorina,gilmore,graham,huckabee, kasich,pataki,paul,rubio,santorum,trump

[3] http://people.tamu.edu/~xhuang/BlogCatalog.mat.zip.

[4] http://adni.loni.usc.edu/study-design/collaborative-studies/dod-adni

4.2 Files related to experiments on the Twitter dataset

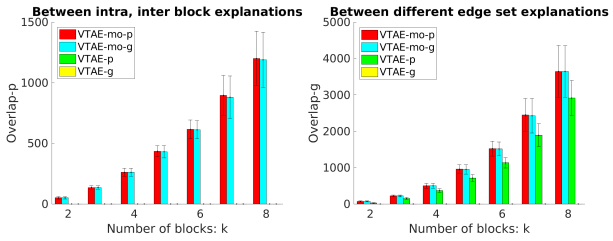
Member lists for the blocks on the smaller Twitter graph. We list the block members of each block in Table 2 of our paper in `.csv` files in the zipped folder of supplementary material.

- Block 1. Trump Supporters: `Trump_Supporters.csv`
- Block 2. Clinton and Supporters: `Clinton_and_Supporters.csv`
- Block 3. Other-Candidates: `Other_Candidates.csv`
- Block 4. Trump Inner Circle: `Trump_Inner_Circle.csv`

The overall set of hashtags. The 136 hashtags we consider in our experiments on the Twitter dataset are indexed in `nodesHashtagUSA.csv`.

4.3 Experimental Result of VTAE-p.

Figure 3 shows the descriptions of the edge sets with our Algorithm 1 for VTAE-p. The edge sets are induced from the block structure in Table 2 on the Twitter `follower/followee` graph over 880 individuals.



(a) Overlap-p (b) Overlap-g

Fig. 2: Overlap computed with equation 3 for the descriptions generated by different feasible relaxations of VTAE.

4.4 Default Mode Network and its generated Graph Cut

We visualize the well known Default Mode Network on the $36-th$ slice and the two-way graph cut it induces in Figure 4.

References

1. Conover, M.D., Ratkiewicz, J., Francisco, M., Gonçalves, B., Menczer, F., Flammini, A.: Political polarization on twitter. In: ICWSM (2011)

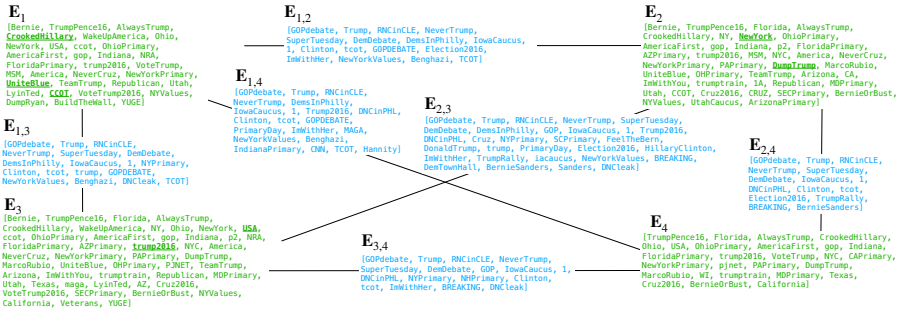
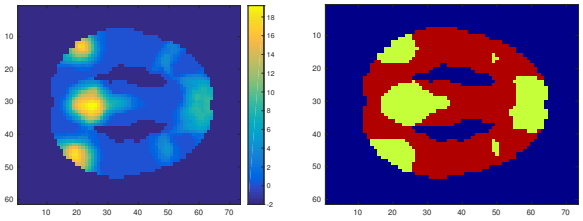


Fig. 3: Edge set descriptions discovered by VTAE-p. We underline the descriptors discovered by our method for **intra-**block edge sets that are *also* chosen by DTDM-cof of [2] to describe their corresponding vertex clusters in table 1 in the paper. Few or none tags are common in the result of our method and the baseline.



(a) (b)

Fig. 4: Default Mode Network (DMN) endorsed by domain experts of neuroscience (Left) and the two-way graph cut (yellow as foreground and red as background) generated from it by thresholding at 20% of maximum volume (Right).

2. Davidson, I., Gourru, A., Ravi, S.: The cluster description problem-complexity results, formulations and approximations. In: NIPS. pp. 6190–6200 (2018)
3. Ding, C., Li, T., Peng, W., Park, H.: Orthogonal nonnegative matrix t-factorizations for clustering. In: SIGKDD. pp. 126–135 (2006)
4. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-completeness. W. H. Freeman & Co., San Francisco (1979)
5. Huang, X., Li, J., Hu, X.: Label informed attributed network embedding. In: WSDM. pp. 731–739 (2017)
6. Tang, L., Liu, H.: Relational learning via latent social dimensions. In: SIGKDD. pp. 817–826 (2009)