Significant Edge Detection in Target Network by Exploring Multiple Auxiliary Networks

Nan Du, Jing Gao, Liang Ge, Vishrawas Gopalakrishnan, Xiaowei Jia, Kang Li, and Aidong Zhang
Department of Computer Science and Engineering
The State University of New York at Buffalo
Buffalo, NY 14260-2500
Email: {nandu, jing, liangge, vishrawa, xiaowei, kli22, azhang}@buffalo.edu

Abstract—Despite the ability to model many real world settings as a network, one major challenge in analyzing network data is that important and reliable links between objects are usually obscured by noisy information and hence not readily discernible. In this paper, we propose to detect these important and reliable links - significant edges, from a target network by using multiple auxiliary networks and a limited amount of labelled information. In this process, we first abstract the community knowledge learnt across target and auxiliary networks to detect significant patterns. The mined community knowledge captures the key profile of network relationships and thus can be used to determine whether an existing edge indicates a true or false relationship. Experiments on real world network data show that our two staged solution – a joint matrix factorisation procedure followed by edge significance score ranking, accurately predicts significant edges in target network by jointly exploring the underlying knowledge embedded in both target and auxiliary networks.

Keywords—Auxiliary Networks, Significant Edge Detection, Social Network, Link Prediction.

I. INTRODUCTION

The growth and development of graph techniques has promulgated the development of various social and biological networks. While this has led to better analysis of the data, errors in generating the data has led researchers to think and reflect about the accuracy of the edges between the nodes - not all the edges truly reflect relationships amongst nodes. There are three main sources for this problem. First, reliable edges can be camouflaged by unreliable and noisy edges. For example, the Protein-Protein Interaction (PPI) networks generated by some high-throughput techniques usually contain lots of noise arising from sample contamination, experimental design or measurement errors. It is estimated that the false-negative rate in directed two-hybrid screens range from 43\% to 71\% [7]. In such cases, the real edges representing true protein-protein interactions are concealed beneath the noise. Second, important edges may be hidden by a mass of trivial and non-essential edges. This is not only true in extracted network such as PPI but also in naturally occurring networks such as social graphs. For instance, according to the study of Hampton et al. [11], an average Facebook user in United States has 245 “friends”. However, Brashears [4] pointed out that an average American has only two close friends making the remaining 243 “friends” very likely to be at best only acquaintances. Third, both reliable and important edges are likely to be missing in the target network. For example, even though Alice and Mary are close friends they might not add each other as friends in Facebook. In summary, we can see that a large number of trivial and unreliable edges can obscure the discovery of pertinent and useful relationship. Our goal is to identify these significant edges from the other edges in the network.

Relying only on domain information to determine the significance of an edge is insufficient or can be misleading. One way to address this concern is to combine useful information from multiple sources to assist in extracting key reliable knowledge. The rationale behind this is that, if two nodes in a network share a “significant” relationship, then they are likely to also exhibit some level of interaction in other networks in which they are participating. The keyword here is “participating” - it is essential that the nodes in the target network also appear in the auxiliary network for us to reliably infer the edge characteristics. However, there are some challenges in using the auxiliary network information, namely: (1) Different sources may present conflicting information making it essential for our system to have a built-in ambiguity resolver. (2) Closely related to the earlier issue is the concern about the trustworthiness of these auxiliary networks. To tackle challenge (1) we can abstract the knowledge in auxiliary networks at community level and correlate it with the target network - viz., community membership and community relationship. To handle the issue raised in (2), we assign weights to the auxiliary network which can be based on the limited labels that we have or on the structural similarity between the networks.

The abstracted knowledge across auxiliary sources allows us to focus only on patterns consistent across sources - thus avoiding inferences from noisy and contradicting information present in these sources itself. This line of thinking stems from the fact that although objects may act differently in different sources, they tend to be grouped into similar communities where the objects share similar patterns in the relative fields. We believe that the patterns or knowledge shared by multiple sources usually encode reliable information. At the same time, different from community membership information that predominantly is consistent across domains of participating vertices, inter-community interactions do depend on the situation and source setting. Hence we propose to discover the underlying knowledge between target network and multiple auxiliary networks and use the shared community memberships as a bridge to unify knowledge across networks. Thus, using the community knowledge that we have learnt from both target and the auxiliary networks, we can estimate each edge’s significance score based on the rationale that: (1) users belonging to the same community or similar communities
are very likely to be in close social relationship, and (2) nodes that are across different communities but contact frequently should also have a close relationship.

Figure 1 shows an holistic work-flow of our approach. The first panel in this figure shows the target network and a series of related auxiliary networks. The middle part shows the community knowledge that we have learnt from the target network and the auxiliary networks. Finally, the right part represents the target network with significance scores - the bold lines represent the edges estimated with high significance scores and the dashed lines represent the edges estimated with low significance scores (only a subset of edges is shown here).

The rest of the paper is organized as follows. Section II introduces notations used and formally defines the problem statement. Section III presents the proposed joint matrix factorization approach to solve the problem. Its efficacy is demonstrated through extensive analysis and discussions in Section IV. We conclude the paper in Section VI after briefly recapitulating some of previous related works in Section V.

II. PROBLEM FORMULATION

We begin by introducing the notations that will be used in our discussions. Let a matrix be represented with an uppercase letter, e.g. $D, d_i$ denotes the $ij$-th entry in $D$, $d_{ij}$ denote vectors of $i$-th row and $j$-th column of $D$, respectively. Table I shows the important notations that we use in this paper.

**Target Network:** We have a target network $G = \{V, E, W\}$, where $V$ is a set of nodes and $E$ is a set of undirected and unweighted edges between the nodes. $W \in \mathbb{R}^{N \times N}$ represents the adjacency matrix of the target network, where $N$ is the number of nodes in the target network. For $v_i, v_j \in V$, $w_{ij} = 1$ if an edge exists between the nodes $i$ and $j$; $w_{ij} = 0$ otherwise.

**Auxiliary Networks:** We also have $M$ auxiliary networks $G^m = \{V, E^m, W^m\} \ (1 \leq m \leq M)$, where each auxiliary network’s adjacency matrix is defined as $W^m$, where $w^m_{ij} = 1$ if an edge exists between node $i$ and node $j$ in the $m$-th auxiliary network. Note that, we assume each auxiliary network has the same nodes as the target network.

**Community Membership Matrix:** We denote the community membership $S$ as an $N \times K$ matrix, where $K$ is the number of communities. This community membership matrix $S$ is shared between the target network and all auxiliary networks.

**Community Relationship Matrices:** The community relationship matrix $H \in \mathbb{R}^{K \times K}$ represents the relationships amongst the $K$ communities from the view of the target network. Similarly, the community relationship $H^m$ in the $m$-th auxiliary network is denoted as a $K \times K$ matrix.

**Label Information Matrix:** An $N \times N$ matrix $F$, represents any prior knowledge we may have on the significance of particular edges in the target network. If we know an edge between $i$ and $j$ to be significant then $f_{ij} = 1$ (otherwise 0).

Thus, based on the above notations, our goal is to use the adjacency matrix $W$ from the target network, adjacency matrices $W^m \ (1 \leq m \leq M)$ from the auxiliary networks and the label information matrix $F$ for the target network, to derive an $N^2 \times 1$ vector $P$, where each entry $p_{ij}$ denotes our confidence on the edge connecting node $i$ and node $j$.

III. METHODOLOGY

In this section, we present our approach in detecting significant edges using auxiliary networks and label information. Section III-A formulates the task as an optimization problem for which we present an efficient iterative algorithm in Section III-B. Finally, we analyze the computational complexity of the proposed method in Section III-C. We refer to our approach as **SEDTN** - Significant Edge Detection in Target Network.

A. Integrated Framework

As mentioned earlier, we aim at discovering consistent patterns at community level and hence decompose each source into two parts - the community membership and the community relationship. This is done by using both target network and multiple auxiliary networks. We perform a joint matrix factorization so that such information across sources are easily aligned and transferred, and then estimate each edge’s significance score. We formulate the significant edge detection problem as the following objective function:

$$\min_{S,H,H^m} J = \min_{S,H,H^m} \|W - SHST\|_F^2 + \sum_{m=1}^M \alpha_m \|W^m - SH^mST\|_F^2 + \beta \left\| (SHST - F)^T \circ F \circ (SHST - F) \right\|_F$$ (1)

s.t. $s_{ik} \geq 0, h_{kl} \geq 0, h^m_{kl} \geq 0, \forall i = 1...N, \forall k = 1...K, \forall l = 1...K, \forall m = 1...M$,

where $\circ$ is Hadamard product (element-wise product) and $\| \cdot \|_F$ is the Frobenius norm of a matrix.
In Eq. 1, the first term $\|W - SHST\|^2_F$ seeks a community membership matrix $S$ and a community relationship matrix $H$ so that their product can well present the target network $W$. The second term $\sum_{m=1}^{M} \alpha_m \|W^m - SH^m ST\|^2_F$ ensures that a specific auxiliary network $W^m (1 \leq m \leq M)$ can be represented by the community membership matrix $S$ and a community relationship matrix $H^m$. Note that $\alpha_m$ is a predefined parameter that denotes the confidence over the $m$-th auxiliary network, and we define the sum of this series of parameters according to auxiliary networks with a constant $\alpha$, i.e., $\sum_{m=1}^{M} \alpha_m = \alpha$. To weight each auxiliary network, strategies like exploiting the label information, calculating similarity to the target network, etc. can be used. For the sake of simplicity, we weight them equally if no prior knowledge is available (i.e., $\alpha_m = \frac{1}{M} (1 \leq m \leq M)$).

These two terms reflect our idea that the community membership information $S$ is consistent across sources, and the community relationship information (i.e. $H$ and $H^m$) represents some special knowledge unique in a specific domain. The last term $\beta \| (SHST - F)^T \circ F \circ (SHST - F) \|^2_F$ puts the constraint that the estimation should not deviate too much from the corresponding edge ground-truth, where $\beta$ is a parameter that expresses our confidence on the ground truth. To be specific, this regularizer can also be represented as $\beta \sum_{i=1}^{N} \sum_{j=1}^{N} f_{ij} \left( (SHST)_{ij} - f_{ij} \right)^2$, which enforces that the estimation should be close to the prior knowledge and the penalties are only imposed on the labeled edges.

![Fig. 2: Learning Community Knowledge across Sources.](image)

The process of learning $S$, $H$ and $H^m (1 \leq m \leq M)$ is summarized in Figure 2. As we can see, the community membership is used as a public base. All sources not only update but also learn from it. Through this process, the public community membership knowledge $S$ improves every source’s effectiveness by avoiding its local noise. Although the sources reconcile their differences on the community membership $S$, they still maintain individual observations for the corresponding specific field in $H$ (or $H^m$). The label information $F$ also contributes to the learning of $S$ and $H$ in the target network. After we learnt the optimal $S$ and $H$, their product $P = SHST$ is the edge significance matrix, where $p_{ij}$ represents the significance score for the edge connecting node $i$ and node $j$. It is worth noting that the proposed method can detect the significant edges from not only the existing edges but also the missing edges in the target network.

**B. Derivation of Update Rules**

Now, we present the method to solve the proposed optimization problem. Our method iteratively updates each variable while fixing the others. Eq. 1 can be rewritten as:

$$
J = Tr(W^TW) + Tr(SHST^2SHST^T) - Tr(2W^TSHST) + \sum_{m=1}^{M} \alpha_m [Tr(W^mTW^m) + Tr(SH^mST^mSH^mST^T) - Tr(2W^mTSH^mST^T)] + Tr(\beta (SHST - F)^T \circ F \circ (SHST - F)).
$$

Update rules for $S$: Fixing $H$ and $H^m (1 \leq m \leq M)$, we can obtain the minimum of $S$ by setting its partial derivative to 0:

$$
\frac{\partial J}{\partial S} = Y + \sum_{m=1}^{M} \alpha_m (B^m - Q^m) + \beta (R - D) = 0, \quad (3)
$$

where $P = 2SHST^2SH + 2SHSTSH^T$, $Y = 2W^TSH + 2WSH^T$, $B^m = 2SH^mST^mSH + 2SH^mST^mSH^mST^m$, $Q^m = 2W^mST^mSH^m + 2W^mTSH^mST^m$, $R = (F \circ SHST^T)S + (F \circ SHST^T)SH + (F \circ F)SH^T$, $D = (F \circ F)SH + (F \circ F)SH^T$. Solving this equation gives us the update rule of $S$ as:

$$
s_{ik} = s_{ik} \left[ \frac{Y + \sum_{m=1}^{M} \alpha_m Q^m + \beta D}{P + \sum_{m=1}^{M} \alpha_m B^m + \beta R} \right]_{(ik)}. \quad (4)
$$

Update rules for $H$: Fixing $S$ and $H^m (1 \leq m \leq M)$, we can obtain the minimum of $H$ by setting its partial derivative to 0:

$$
\frac{\partial J}{\partial H} = 2S^TSHST^2S - 2S^TWS + S^T(F \circ SHST^T)S - S^T(F \circ F^T)S = 0. \quad (5)
$$

Solving this equation gives us the update rule of $H$ as:

$$
h_{kl} = h_{kl} \sqrt{\frac{[2S^TWS + S^T(F \circ F^T)S]_{(kl)}}{[2S^TSHST^2S + S^T(F \circ SHST^T)S]_{(kl)}}}. \quad (6)
$$

Update rules for $H^m (1 \leq m \leq M)$: Similarly, the update rule for $H^m$ is:

$$
h_{kl}^m = h_{kl}^m \sqrt{\frac{[S^T W^m S]_{(kl)}}{[S^T S^m ST^m S]_{(kl)}}}. \quad (7)
$$

It is easy to prove that the update rules for the objective function satisfy the Karush-Kuhn-Tucker (KKT) conditions [3]. The pseudo-code of iteratively solving Eq. 1 is shown in Algorithm 1 - line 4 to line 12. After obtaining the optimal $S$ and $H$, we obtain the edge significance score matrix $P$, which is symmetric and non-negative.
Algorithm 1 Edge Significance Score Computation

Input: Adjacent matrix of target network \( W \), adjacent matrices of a series of auxiliary networks \( W^m \) (1 \( \leq m \leq M \)), edge label information \( F \), number of communities \( K \) and parameters \( \alpha_m, \beta_m \) (1 \( \leq m \leq M \))

Output: Ranking list of edge significance scores \( \tilde{P} \);

1: Initialize \( S \), \( H \) and \( H^m \) (1 \( \leq m \leq M \));
2: begin
3: repeat
4:   for \( i \leftarrow 1 \) to \( N \), \( k \leftarrow 1 \) to \( K \) do
5:     Update \( s_{ik} = s_{ik} \sqrt{Y + \sum_{m=1}^{M} \alpha_m Q^m + \beta D}_{(ik)} \)
6:   end for
7:   for \( k \leftarrow 1 \) to \( K \), \( l \leftarrow 1 \) to \( K \) do
8:     Update \( h_{kl} = h_{kl} \sqrt{[2 S^T W S + \beta S^T (P_0 F^T) S]_{(kl)}} \)
9:   end for
10:   for \( m \leftarrow 1 \) to \( M \), \( k \leftarrow 1 \) to \( K \), \( l \leftarrow 1 \) to \( K \) do
11:     Update \( h_{kl} = h_{kl} \sqrt{[S^T W^m S]_{(kl)}} \)
12: end for
13: until Convergence criterion is satisfied
14: \( P = SHS^T \);
15: \( \tilde{P} \leftarrow \) Sort the pairs of nodes using significance score in descending order.
16: end

C. Time Complexity

Recall that \( N \) is the number of nodes, \( K \) is the number of communities and \( M \) is the number of auxiliary sources. In our proposed method, the time for conducting joint matrix factorization is \( O(N^2 K + NK^2 + MNK^2 + MN^2 K + N^3) \) and the time for ranking \( F \) is \( O(N \log N) \). Note that, \( O(N^3) \) comes from the Hadamard product with the label information matrix \( F \). Since \( F \) is very sparse (we usually know very limited about the labeled edges) and Coopersmith-Winograd algorithm \[15\] is used, the time spent on Hadamard product can be reduced to \( O(N^2) \). Furthermore, we usually have much fewer auxiliary networks than nodes (i.e., \( M \ll N \)) and fewer communities than nodes (i.e., \( K \ll N \)). Thus, the overall time complexity is \( O(N^2 K) \). Suppose the number of iterations is \( T \), the time complexity of entire algorithm is \( O(TN^2 K) \). In experiments, we observe that \( T \) is usually between 8 and 60. Considering the output of the proposed method is an \( N^2 \times 1 \) vector, this running time is acceptable.

IV. Experiments

Having described our methodology, we now present a detailed evaluation and reasoning of results along with parameter sensitivity analysis.

A. Dataset

We use two datasets - friend social dataset and biological dataset, to demonstrate the proposed method’s performance.

Friend Social Dataset: This dataset was collected by MIT human dynamics lab \[16\], which closely tracked the daily life of students living in a dormitory. It contains friendship associations and levels and the participants measured at regular time interval.

To construct our target network, we transform all the edges whose weights (average friendship score) are over 0.5 into an unweighted (0 or 1) network. Only 30\% of total edges are used as “label” in our method (i.e. matrix \( F \)), and the noisy edges with the same quantity is added. In other words, the noisy rate on the target network is 50\%. By doing this, we only know who and who claim to know each other but do not know the exact relationship between them, just like a Facebook friend network - really close friendships have been covered by a mass of trivial and inessential relationships.

Furthermore, we also have three auxiliary networks corresponding to three different contact methods - phone calls, messages and music shares. Note that, since all the edges in the auxiliary networks are unweighted, it makes the task challenging. To summarize, our task is to distinguish close friendships from the rest.

Biological Dataset: This dataset is from Kashima et al. \[13\]. This dataset consists of three different species - C.elegans, S.cerevisiae and H.pylori. Each species has a metabolic network, where each node is an enzyme. There is a link between two nodes if a pair of enzymes associated to the nodes that catalyze successive reactions in a known metabolic pathway. Although the nodes (i.e., enzymes) for each species are different, we can easily map the label information from auxiliary species to the target species network based on the cross-species similarity matrices. Specifically, if there is an edge between molecules \( a \) and \( b \) in one particular species, their corresponding molecule \( a^* \) and \( b^* \) (the molecule with the highest similarity according to the cross-species similarity matrices) in the target species also have an edge. Since this metabolic information comes from precise biological experiments, we can use it as the ground truth. Moreover, this dataset has the cross-species similarity matrices that are computed by the normalized Smith-Waterman score \[19\] of pairwise molecule.

Since the biological dataset has three different species, we have three detection tasks: When one is used as the target network, the remaining two are used as auxiliary networks. Due to the fact that all edges in the network now are reliable, we need to add some noise to the target network in order to simulate the real biological network. Therefore, we randomly add \( \lambda \times V \) edges to the target network to make it noisy, where \( V \) is the number of labeled edges we have used (less or equal to the sum of labeled edges) and \( \mu \) is a parameter used to control the noisy edge generation.

B. Experiment Setup

In this part, we present the performance measure and discuss the baseline methods.

1) Performance Measure: Because the ground truths that reflect the friendship levels in the friend social dataset are continuous values, whereas those in the biological dataset are binary values, we need to adopt different performance evaluation methods for the two datasets.

Evaluation on friend social dataset: To circumvent the problem of continuous values, we use the top \( \lambda \% \) edges
from both sorted ground truth and estimated significance score list \( \hat{P} \) and calculate the accuracy of the method as:
\[
\text{Accuracy}_{\text{Social}} = \frac{L_e \cap L_g}{\text{ceil}(N_{e} \times N_{g} \times \lambda_{g})},
\]
where \( \text{ceil}(\cdot) \) rounds a real number towards positive infinity, \( L_e \) and \( L_g \) are the top \( \lambda_{g} \) elements (i.e., edges) in \( \hat{P} \) and ground truth list, respectively.

**Evaluation on biological dataset:** In the case of biological dataset, since the ground truths are binary and we know the exact number of significant edges (assume it is \( X \)), we can directly compare the top \( X \) elements. Therefore, the accuracy in the biological dataset can be defined as:
\[
\text{Accuracy}_{\text{Bio}} = \frac{L_e \cap L_g}{X},
\]
where the \( L_e \) is the top \( X \) elements in \( \hat{P} \) and \( L_g \) is the \( X \) labeled edges from the ground truths.

2) Baseline Methods: We compare our approach with the following baseline methods, which cover a wide variety of ways to recommend edges. These approaches can be partitioned into the following five categories.

Local similarity indices: We use Common Neighbors (CN) [17] and Adamic-Adar Index (AA) [1] as the two baselines in this scheme. The basic idea is to use node specific information to calculate similarity and more the two nodes are similar, more likely is the edge connecting them.

Global similarity indices: As against the local similarity indices, techniques like Katz Index [14] and Average Commute Time (ACT) [8] use the entire network topology, for instance in this cases “walk” on the nodes”, to ascertain the closeness between the two nodes and consequently the presence of edge.

It is worth noting that the above four baseline methods are originally designed to work on only one particular network. For the sake of fairness and comprehensiveness, we enrich these methods through the information from other sources. We first generate a consensus network constructed from all the sources (target network and auxiliary networks) by majority voting, and then apply these four methods on it. Note that the methods use this strategy are marked with star mark (*).

Link Propagation (LP): This method is based on Label Propagation wherein the similarity between two edges is determined by their state [13].

Matrix Factorization (MF): Tang et al. [23] proposed a method to recommend edges based on matrix factorization. Besides the target network on which predictions are made, their method also needs a homophily coefficient matrix to calculate the similarity between nodes, which we compute according to the nodes topological information in the auxiliary networks.

Different Penalty Regularizer (DPR): Given the labeled edges in a specific network (i.e., matrix \( F \)), some studies like [23] and [24] prefer to use a different penalty regularizer \( Tr(SFS^T) \). The idea of this penalty regularizer is that, if we know there is an edge between node \( a \) and node \( b \), then they should belong to the same community. Since we observe frequent inter-community interactions, we think this penalty regularizer may be too strong in our problem. So, we adapt this as \( Tr\left( (SHS^T - F)^T \circ F \circ (SHS^T - F) \right) \), so as to only prevent the predictions on certain edges that deviate too much on the label information. Comparison with DPR will help us understand which penalty regularizer is better in the significant edge detection problem.

<table>
<thead>
<tr>
<th>Method</th>
<th>10%</th>
<th>15%</th>
<th>20%</th>
<th>25%</th>
<th>30%</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEDTN</td>
<td>0.525</td>
<td>0.662</td>
<td>0.734</td>
<td>0.808</td>
<td>0.872</td>
</tr>
<tr>
<td>CN</td>
<td>0.417</td>
<td>0.473</td>
<td>0.553</td>
<td>0.620</td>
<td>0.662</td>
</tr>
<tr>
<td>CN*</td>
<td>0.423</td>
<td>0.486</td>
<td>0.550</td>
<td>0.613</td>
<td>0.655</td>
</tr>
<tr>
<td>Adar</td>
<td>0.414</td>
<td>0.475</td>
<td>0.552</td>
<td>0.630</td>
<td>0.661</td>
</tr>
<tr>
<td>Adar*</td>
<td>0.429</td>
<td>0.483</td>
<td>0.557</td>
<td>0.621</td>
<td>0.657</td>
</tr>
<tr>
<td>Katz</td>
<td>0.433</td>
<td>0.496</td>
<td>0.610</td>
<td>0.686</td>
<td>0.767</td>
</tr>
<tr>
<td>Katz*</td>
<td>0.521</td>
<td>0.609</td>
<td>0.665</td>
<td>0.721</td>
<td>0.766</td>
</tr>
<tr>
<td>ACT</td>
<td>0.378</td>
<td>0.480</td>
<td>0.549</td>
<td>0.614</td>
<td>0.671</td>
</tr>
<tr>
<td>ACT*</td>
<td>0.394</td>
<td>0.475</td>
<td>0.532</td>
<td>0.582</td>
<td>0.632</td>
</tr>
<tr>
<td>LP</td>
<td>0.443</td>
<td>0.545</td>
<td>0.592</td>
<td>0.652</td>
<td>0.722</td>
</tr>
<tr>
<td>MF</td>
<td>0.516</td>
<td>0.569</td>
<td>0.627</td>
<td>0.689</td>
<td>0.743</td>
</tr>
<tr>
<td>DPR</td>
<td>0.520</td>
<td>0.557</td>
<td>0.605</td>
<td>0.644</td>
<td>0.689</td>
</tr>
</tbody>
</table>

### C. Experimental Results

To show the effectiveness of the proposed framework, we perform three sets of experiments. The first set of experiments compares the results of the proposed algorithm with the baseline algorithms. The second set of experiments demonstrates the sensitivity with respect to the number of potential communities (i.e., \( K \)). We demonstrate that the proposed algorithm is not sensitive to \( K \). The third set of experiments describes the effect of varying number of auxiliary sources on our proposed method. We demonstrate that increasing the number of auxiliary sources improves the performance.

In our experiments, we set \( \alpha = 1 \) and \( \beta = 5 \). The rationale is that \( \alpha \) reflects our confidence on the knowledge transferred from auxiliary networks that may contain noise or conflicts, thus assigning a relative low value to them is suitable. On the other hand \( \beta \) shows our confidence on the labeled edges that come from reliable ground truth, and hence can be assigned a relative larger value. Also, we fix \( \mu = 2 \) to decide the number of noisy edges added to the target biological network, which means the noisy edges are two times of the labeled edges at any cases. By doing this, it is easy to see that identifying significant edges from a target species network is going to be a very challenging task. Finally, for each setting, we perform 10 experiments and report the mean values.

1) Performance Comparison: For both friend social and biological datasets, we vary the parameter \( \lambda \) which is the percentage of elements to be evaluated as 10%, 15%, 20%, 25% and 30%. By comparing the performance under different values of \( \lambda \), we can better demonstrate the effectiveness of identifying significant edges from different levels. For the friend social dataset the results of the proposed method and the baselines are shown in Table II. For biological dataset, the results of distinguishing the reliable interactions (e.g., edges) between enzymes from *C.elegans*, *S.cerevisiae* and *H.pylori* are reported in Table III.

From the performances on these two datasets, it can be found that SEDTN outperforms the other methods in identifying the significant edges under different labeled percentage. Also, as expected, an increase in labeled examples percentage leads to increase in performance. Moreover, as the \( \lambda \) increases, the gaps of performance between SEDTN and the other baselines become more and more obvious. There are mainly two reasons for that: First, our constrained regularizer makes sure that the label information is used accurately. Second,
although this label information is not directly used by the auxiliary networks, it indirectly propagates to them by the transfer of the shared knowledge $S$.

From the results, we can also see that the performances of the local and global similarity-based indices have not been improved when the auxiliary sources are applied. The reason behind this is that these methods cannot distinguish the real and useful information from the noisy information. This also demonstrates that the underlying community membership which we have learnt across the sources is a mutually shared knowledge that provides significant guidance on the significant edge detection. Furthermore, it is worth noticing that when different penalty regularizer is applied in our objective function (i.e., $DPR$), the performances of $SEDTN$ decrease. This may be due to its strong assumption that if there is an edge between node $a$ and node $b$, then they should belong to the same community. In our case, frequent interactions are usually found across the communities.

2) Varying the Number of Communities: One key parameter in the proposed method is the number of the communities $K$. In most cases, we do not know the exact number of communities hidden in the network. Therefore, we investigate the impact of $K$ on the performance of the proposed approach in terms of $\lambda$. We increase $K$ from 9 to 45 with a step of 3 and show the performance in friend social dataset in Figure 6 with various $\lambda$. Similarly, we test the performance of $SEDTN$ on the biological dataset in terms of $K$. Figure 3(a), Figure 3(b) and Figure 3(c) show the performance on detecting significant edges in terms of $K$ according to 10%, 20% and 30% ratio of labeled data, respectively.

In general, when we increase $K$, the performance in both datasets shows similar patterns: The performance is slightly low when $K$ is extremely small, and the performance begins to increase. After reaching the peak, the performance decreases slightly. Although the performance changes with a certain pattern according to $K$, the changes are actually very slight in an overall view. Thus, $SEDTN$ is not very sensitive to the choice of the number of communities, i.e., the accuracy does not dramatically change as the number of communities increases. Furthermore, we find that the tendency of the performance is very similar with different ratio of labeled data. Using different ratio of labeled data, for each species, the performance tends to reach the peak at the relatively the same value of $K$ (i.e., around 24 for C.elegans, around 30 for S.cerevisiae and around 36 for H.pylori). These patterns can be used to determine the optimal value of $K$ for the $SEDTN$ in practice.

3) Varying the Number of Sources: In this part, we demonstrate the relationship between $SEDTN$’s performance and the number of auxiliary sources that we use to help detecting significant edges. First, there are three auxiliary networks in the friend social dataset, and then we use different number of these auxiliary networks to detect the significant edges from a target network. Thus, we have different combinations of the sources. For example, when we want to use two auxiliary networks, there are three possible combinations including - phone calls with sms, phone calls with music shares and sms with music shares. In such a case, we only report the best result of these combinations. We show $SEDTN$’s performance on friend social network with different number of auxiliary networks in Figure 5, while varying the parameter $\lambda$.

For the biological dataset, we iteratively select one species as the target network (when each species has been selected as target network, noise is added to it with the strategy we mentioned in Section IV-B), and then use different number of sources to determine the significant edges. To be more specific, we compare the situations where we have zero, one and two auxiliary sources. We also choose seven different ratio $10\%$, $15\%$, $20\%$, $25\%$, $30\%$, $35\%$ and $40\%$ of the proven edges in the target network as label information. The results of distinguishing significant edges form $C.elegans$, $S.cerevisiae$ and $H.pylori$ are shown in Figure 4(a), Figure 4(b) and Figure 4(c), respectively. As the figures reveal, the accuracy is improved with the increasing of the number of sources. The results on both datasets show that each auxiliary network aids at finding the reliable edges from a target network, and more the auxiliary networks we use, better the performance we get.

4) Parameter Sensitivity: There are namely 2 parameters in our objective function Eq. 1: $\alpha_m$ ($1 \leq m \leq M$) and $\beta$. As mentioned above, we weight $\alpha_m$ uniformly over parameter $\alpha$ if no prior knowledge is available (i.e., $\alpha_m = \frac{1}{M}$ ($1 \leq m \leq M$)). We conducted the sensitivity experiments on the friend social dataset, refer Figure 7, by varying one parameter when other is fixed as the default setting (i.e., $\alpha = 1$ and $\beta = 5$).

Note that, $\alpha$ represents the confidence of our belief over information propagation from auxiliary networks, which may contain conflict or unreliable information. Therefore, smaller $\alpha$ usually yields better performance. $\beta$ represents our confidence on the prior knowledge on whether some particular edges in the target are significant. This prior knowledge, which is obtained from the surveys, is deemed to be reliable and thus a large $\beta$ is usually better. The result in Figure 7 confirms our observation.

V. RELATED WORK

Broadly speaking, our work is related to link prediction with various strategies such as local similarity indices [17], [25], global similarity indices [8], [22], and probabilistic models [27]. Most of these studies aim to recommend some possible missing edges for the current network or recommend some edges to link in the future based on one particular source of data. Furthermore, these methods tend to believe the network they analyze is reliable. Some methods [5], [6] only focus on detecting significant edges from either existing edges or missing edges. In our setting, the significant edges can be hidden in both cases.

Another relevant topic is trustworthiness analysis [2], [26], [28], which tries to find the truth given multiple conflicting sources. Usually the truth is a weighted combination of multiple pieces of information where weights are derived for multiple sources based on their reliability. Different from these truth discovery approaches, we target at identifying the truth of edges based on community level knowledge uncovered from different sources. Furthermore, most of these previous truth discovery studies are unsupervised which is different from us.

Some studies formulate the similar problem as a supervised learning task where labeled edges are required for model training [12]. However, these methods can only provide a binary prediction result for a specific test edge - existence or not. Our work is different in that we estimate the significance score of each edge by considering multiple sources in a supervised
TABLE III: Comparison with Baselines Using Different Labeled Percentage on the Biological Dataset

<table>
<thead>
<tr>
<th>Method</th>
<th>C.elegans</th>
<th>S.cerevisiae</th>
<th>H.pylori</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10%</td>
<td>15%</td>
<td>20%</td>
</tr>
<tr>
<td>SEDTN</td>
<td>0.343</td>
<td>0.441</td>
<td>0.501</td>
</tr>
<tr>
<td>CN*</td>
<td>0.025</td>
<td>0.051</td>
<td>0.102</td>
</tr>
<tr>
<td>CN+</td>
<td>0.039</td>
<td>0.043</td>
<td>0.048</td>
</tr>
<tr>
<td>Adar</td>
<td>0.043</td>
<td>0.061</td>
<td>0.138</td>
</tr>
<tr>
<td>Adar+</td>
<td>0.039</td>
<td>0.043</td>
<td>0.048</td>
</tr>
<tr>
<td>Katz</td>
<td>0.121</td>
<td>0.146</td>
<td>0.161</td>
</tr>
<tr>
<td>Katz+</td>
<td>0.080</td>
<td>0.087</td>
<td>0.077</td>
</tr>
<tr>
<td>AC+</td>
<td>0.062</td>
<td>0.064</td>
<td>0.068</td>
</tr>
<tr>
<td>LP</td>
<td>0.194</td>
<td>0.248</td>
<td>0.322</td>
</tr>
<tr>
<td>MF</td>
<td>0.085</td>
<td>0.135</td>
<td>0.192</td>
</tr>
<tr>
<td>DFP</td>
<td>0.175</td>
<td>0.260</td>
<td>0.350</td>
</tr>
</tbody>
</table>

Fig. 3: Effect of Varying Number of Communities on Biological Dataset: (a) C.elegans, (b) S.cerevisiae and (c) H.pylori.

Fig. 4: Effect of Varying Number of Sources on Biological Dataset: (a) C.elegans, (b) S.cerevisiae and (c) H.pylori.

Fig. 5: Effect of Varying Number of Sources on Friend Social Dataset.

Fig. 6: Effect of Varying Number of Communities on Friend Social Dataset.

manner. Moreover, Tang et al. proposed a trust prediction [23] based on the matrix factorization, which is similar to our method. However, their method is an unsupervised framework without considering label information. In addition, our
work can be a useful pre-processing tool for recommendation systems where many efforts have been devoted to developing collaborative filtering [18], [20], [21] algorithms. Many of such algorithms involve matrix factorization; however, they are mainly focusing on one particular matrix. Different from these studies, we adopt a joint matrix factorization procedure to project multiple rating matrices into a common subspace so that group behavior is comparable across sources and discrepancy can be found. Also, the proposed framework shares similar principles with multiple source learning [9], [10] where hidden knowledge has to be discovered by learning from multiple sources simultaneously, but the perspective of utilizing multiple sources in significant edge detection is novel.

VI. Conclusion

In this paper, we proposed to tackle the problem of detecting significant edges from a target network by extracting underlying community knowledge shared across multiple sources. To our best knowledge this is the first work that aims to identify the importance of edges using multiple sources. Our proposed two-staged model SEDTN first captures the underlying community knowledge from both target and auxiliary networks using joint matrix factorization. We developed an effective optimization method to solve this joint matrix factorization problem. After obtaining the community knowledge, we then estimate the significance scores for all edges based on the shared community membership information extracted from multiple networks. Experiments on real world datasets demonstrate the superior performance of the proposed method in detecting significant edges from the target network as compared to other baseline methods. We contend that the proposed approach is an effective analytical tool that can find abundant applications in numerous noisy networks to filter noisy edges and extract only pertinent relationships for further analysis and exploration.

References