MϵEPs: Soft Roles in Large Networks

No Author Given

No Institute Given

Abstract. In social network analysis, the notion of *roles* corresponds to actors having similar structural signatures. Actors performing the same role have similar behavioural and functional characteristics. Few examples of structural roles are bridge nodes, clique members and star centers. Role discovery involves partitioning the nodes in a network based on their structural characteristics. The notion of roles is complementary to the notion of community detection, which involves partitioning the network into cohesive subgroups. In this paper we address the problem of automatically discovering the roles performed by the nodes in a network. We propose *Multiple ϵ-Equitable Partitions* (MϵEPs) based approach for finding soft role memberships of nodes. We show that MϵEPs helps in diverse graph mining tasks: soft role discovery, studying of node & link evolution in temporal networks. Results on real world multi-role ground-truth networks show that MϵEPs outperforms the recent approaches to role discovery in social networks.

Keywords: Role analysis, structural equivalence, soft roles, graph mining, graph partitioning

1 Introduction

Structural role discovery in networks is an emerging research area in the data mining community [5, 12]. Historically, the *dual* notions of *social role* and *social position* have been long used by sociologists to draw equivalence classes from network of social relationships [15, 1]. Role discovery involves partitioning the nodes in a network based on their structural features. This involves identifying *social position* as collection of actors who are similar in their ties with other actors in the network. Nodes which perform similar *social roles*, have similar structural interaction patterns to other nodes or positions in the network. Such nodes share similar traits, such as betweenness centrality, degree centrality and the number of triangles to which they belong. Few example *roles* are broker nodes, peripheral nodes and near clique members, the corresponding *positions* are node clusters having actors with high betweenness centrality, low degree centrality and higher clustering co-efficient respectively. The importance of role discovery methods has been established in a variety of graph mining tasks: anomaly detection [11], transfer learning [5] and network sampling [10] to name a few. As an example, head coaches in different football teams occupy the *position manager* by the virtue of the similar kind of relationship with players, assistant coaches, medical staff

and the team management. It might happen that an individual coach at the position manager may or may not have interaction with other coaches at the same position. Further, the actors at the position *manager* can be in a *role* of "Coach" to actors at the position *player* or a "Colleague" to the actors at the position *assistant coach*. Similarly, the actors at position *medical staff* can be in a role of "Physiotherapist" or a "Doctor" to actors at the position *player*.

The notion of structural roles and positions in networks, is complementary to the popular notion of community detection. Nodes which perform same structural role have similar node characteristics such as, in-degree, out-degree, average weight of edges entering and leaving the node egonet etc. to name a few; and nodes which belong to the same community are characterized by properties such as, modularity, density and cohesion.

Classical methods of role discovery find the structural correspondence among nodes by partitioning the network into *disjoint subsets* using an equivalence criterion. These notions of role discovery like regular equivalence [16], automorphisms [2] and equitable partition [9] often lead to trivial partitioning of the nodes in the network. In case of structural equivalence, automorphic equivalence and equitable partition, the trivial partitioning is primarily due to the strictness in the number or type of connections a node has to other roles in the network. This results in largely singleton roles. On the other hand, unrestricted leniency, as in the case of regular equivalence, results in a giant equivalence class.

An ϵ-equitable partition (ϵEP) [6] is a notion of equivalence, which has many advantages over the classical role extraction methods. ϵEP allows a leeway of ϵ in the number of connections the node performing same role can have with the nodes of another role. The notion of ϵEP is similar in spirit to the notion of stochastic blockmodels (SBM) [14], in that both approaches permit a bounded deviation from perfect equivalence among nodes. In the Indian movies dataset from IMDb, authors in [6] have shown that actors who fall in the same cell of the partition, tend to have acted in similar kinds of movies. Further, the authors also show that people who belong to a same role of an ϵEP tend to evolve similarly.

Classical methods [16, 2, 9] along with ϵEP [6] and SBM [14], extract roles using the *complete structural view of a graph*. They extract roles by taking into account the connectivity patterns of each actor, to all the other actors/roles in the network. We refer to these methods as *global graph characteristics* based approaches. Recent methods of role discovery [5, 3] on the other hand perform role extraction using local node features, we refer to these methods as based on *local node characteristics*. These methods create a *feature vector* for each of the node in the graph using structural/network specific attribute features, such as, degree, weighted degree, egonet degree and etc. Role memberships are then extracted using this node-feature matrix by performing Non-negative Matrix Factorization (NMF); NMF gives two factors: a node-role matrix and a role-feature matrix. Hence these methods assign *soft role memberships* to the actors. It is worth mentioning here that, though these methods identify *multiple roles* for a node, they do not categorize nodes based on the set of roles they play, *i.e.*,

they identify soft roles, but *do not* cluster them into *positions*. While the methods based on *local node characteristics* have good scaling behaviour, they require that we choose *a priori*, the set of local node features used to characterize roles. With *global graph characteristics* based approaches, one can discover structural signatures corresponding to roles, but *i.*) these methods do not scale to large graphs, and *ii.*) by partitioning the actors into *disjoint subsets*, the role assignment is *hard membership*; actors in the real world play multiple roles. In this paper, we propose Multiple ϵ-Equitable Partitions (MϵEPs), which addresses the problems associated with the *global graph characteristics* and *local node characteristics* based role assignment methods, specifically:

i MϵEPs computes roles using the complete structural view of the graph and is scalable.
ii MϵEPs assigns soft roles to the actors in a network.
iii Given these soft roles, MϵEPs categorizes the nodes based on the set of roles they play.

The rest of the paper is organized as follows, Section 2 gives an overview of ϵ-equitable partition. In Section 3 we discuss our proposed method of MϵEPs. In Section 4 we present the evaluation methodology, datasets and the results. Section 5 concludes the paper with future research directions.

2 Overview of ϵ-Equitable Partition

Definition 2.1. (*equitable partition*) A partition $\pi = \{c_1, c_2, ..., c_k\}$ on the vertex set V of graph G is said to be *equitable* [9] if,

$$\text{for all } 1 \leq i, j \leq k, deg(u, c_j) = deg(v, c_j) \text{ for all } u, v \in c_i \tag{1}$$

where,

$$deg(v_i, c_j) = sizeof\{v_k \mid (v_i, v_k) \in E \text{ and } v_k \in c_j\} \tag{2}$$

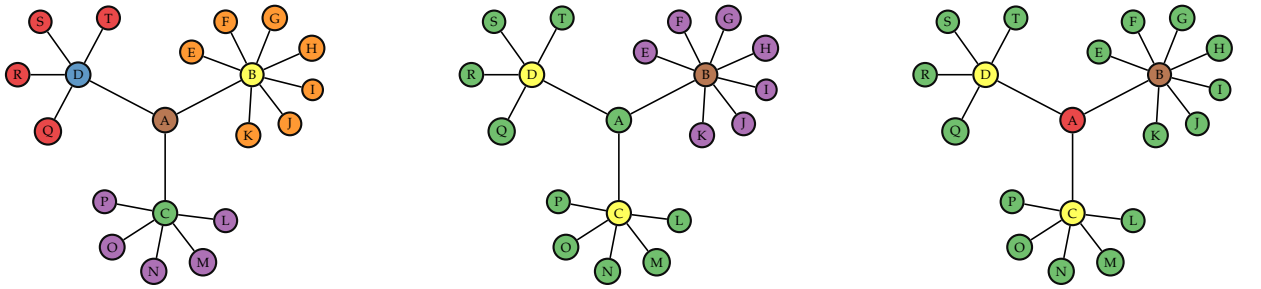
An equitable partition can be used to define positions in a network; each cell c_j corresponds to a position and $deg(v_i, c_j)$ corresponds to the number of connections the actor v_i has to the position c_j.

The *degree vector* of a node u is a vector of size k (the total number of cells in partition). Each component of the vector is the number of neighbours u has in each of the member cells of the partition, as given by Equation 2.

Definition 2.2. (ϵ-*equitable partition*) A partition $\pi = \{c_1, c_2, ..., c_k\}$ of the vertex set $\{v_1, v_2, ..., v_n\}$, is defined as ϵ-*equitable partition* [6] if:

$$\text{for all } 1 \leq i, j \leq k, |deg(u, c_j) - deg(v, c_j)| \leq \epsilon, \text{ for all } u, v \in c_i \tag{3}$$

The above definition proposes a relaxation to the strict partitioning condition of equitable partition (Equation 1), now equivalent actors can have a difference of ϵ in the number of connections to other cells/positions in the partition.



(a) Equitable Partition. (b) First ϵ-Equitable Partition (ϵEP1), $\epsilon = 1$. (c) Second ϵ-Equitable Partition (ϵEP2), $\epsilon = 1$.

Fig. 1: Example Network. (a) Equitable Partition is [{A},{B},{C},{D},{E-K},{L-P},{Q-T}]. (b) First ϵ-Equitable Partition with $\epsilon = 1$ is [{A,L-P,Q-T},{B},{C,D},{E-K}]. (c) Second ϵ-Equitable Partition with $\epsilon = 1$ is [{A},{B},{C,D},{E-T}]. Nodes at same positions are depicted with same colours. The ϵ-equitable partition of a graph is not unique.

Illustrative Example Figure 1 depicts an example toy network under equitable partition (EP) and ϵ-equitable partition (ϵEP). The equitable partition of Figure 1(a) has *seven positions*. For the same graph, the ϵEP with $\epsilon = 1$ has *four positions* as shown in Figure 1(b). The *degree vectors* for the nodes A, L and T under the EP of Figure 1(a) and ϵEP ($\epsilon = 1$) of Figure 1(b) are shown in Table 1.

EP	{A}	{B}	{C}	{D}	{E,..,K}	{L,..,P}	{Q,..,T}
$\overrightarrow{deg}(A)$	0	1	1	1	0	0	0
$\overrightarrow{deg}(L)$	0	0	1	0	0	0	0
$\overrightarrow{deg}(T)$	0	0	0	1	0	0	0

ϵEP	{A, L..T}	{B}	{C,D}	{E,..,K}
$\overrightarrow{deg}(A)$	0	1	2	0
$\overrightarrow{deg}(L)$	0	0	1	0
$\overrightarrow{deg}(T)$	0	0	1	0

Table 1: The degree vectors of nodes A, L and T for the equitable partition of Figure 1(a) and the ϵ-equitable partition of Figure 1(b).

The nodes A, L, T of the graph, occupy three different positions under the EP, but they belong to the same position under ϵEP, with $\epsilon = 1$ of Figure 1(b). As evident from the degree vectors of node A, L, T from Table 1, their values for the components of positions $\{B\}$, $\{C\}$ and $\{D\}$ differ under the EP. Therefore they violate the definition of EP (Equation 1) of a graph. On the other hand, amongst each other, the degree vectors of A, L, T differ by at most one degree for the same set components under the ϵEP ($\epsilon = 1$). Hence they have common cell membership under the ϵEP of the graph. This relaxation in the definition of the ϵEP of the graph (Equation 3) leads to non trivial partitioning of the network.

Computing ϵ-Equitable Partition A scalable algorithm to compute the ϵEP of a graph is proposed by the authors in [4]. We discuss this algorithm briefly. The ϵ-equitable partition takes as input an *unit* partition of the graph G. An *active* list is used to hold the indices of all the unprocessed cells from π, and is updated in every iteration of the refinement procedure. c_a is the set of vertices from the *current active cell* of the partition π. The initial active cell c_a is therefore the entire vertex set V. Additionally, a function f, which maps every vertex $u \in V$ to its

degree to c_a is used. Mathematically, $f : V \rightarrow \mathcal{N}$ is defined as follows:

$$f(u) = deg(u, c_a), \ \forall u \in V \tag{4}$$

The procedure then sorts the vertices in each cell of the partition using the value of the function f as a key for comparison. The procedure then *splits* the contents of each cell wherever the keys are more than ϵ apart, thereby creating new cells. The partition π is updated accordingly, and the indices corresponding to any new cells formed after the split are added to the *active* list. The procedure exits when either the *active* list is empty or the partition becomes discrete. The resulting partition is the *ϵ-equitable partition* of the graph G. The authors show that the algorithm is highly scalable for sparse graphs.

3 Proposed Method

Given a network, the goal of MϵEPs is to derive soft role memberships of actors and categorize these actors into positions. MϵEPs consists of two components: *i*.) *Soft Role Assignment*: generating *multiple* ϵ-equitable partitions of a network, each ϵEP of the network corresponds to a single role assignment and *ii*.) *Positional Equivalence*: categorizing actors based on the set of roles they play. We discuss role assignment in §§3.1 and positional equivalence in §§3.2.

3.1 Soft Role Assignment

Non-Uniqueness of ϵ-Equitable Partition: An observation we made from ϵEP method is that the initial order in which the algorithm groups actors may affect the final position of an actor. A different starting order for the algorithm may therefore allow us to analyze *multiple roles* in a better way. For example, given the degree vectors (DVs) of the equitable partition of the network of Figure 1(a), we can generate two different ϵ-equitable partitions. We start with the DVs of the equitable partition, and continue merging two adjacent cells at a time which are within ϵ of each other. This merging of cells is subject to the constraint that, merging two cells doesn't violate the ϵ criterion for the other cells of the partition. For generating the first ϵ-equitable partition ϵEP1, with $\epsilon = 1$, we start with the *shuffled cell order* $[\{L-P\}, \{Q-T\}, \{A\}, \{C\}, \{B\}, \{D\}, \{E-K\}]$. The final 1-equitable partition is $[\{A, L-P, Q-T\}, \{B\}, \{C, D\}, \{E-K\}]$ as depicted in Figure 1(b). The *shuffled cell order* for generating the second ϵ-equitable partition is $[\{E-K\}, \{L-P\}, \{Q-T\}, \{C\}, \{D\}, \{B\}, \{A\}]$. The partition after the final merge is $[\{A\}, \{B\}, \{C, D\}, \{E-T\}]$, as depicted by ϵEP2 in the Figure 1(c).

Multiple ϵ-Equitable Partitions The implementation of our scalable *multiple ϵ-equitable partitions* algorithm is based on the modification of Scalable ϵEP algorithm proposed by the authors in [4]. The key in our proposed method is to generate *multiple initial split cell sequences* which are *split* within ϵ from the initial active cell using f (Eq. 4) as the comparison key. Once these multiple sequences are generated, we can refine them in *parallel* using the Scalable ϵEP method [4] to generate *multiple* ϵEPs of a graph.

Algorithm 1 MϵEPs: Algorithm to find multiple ϵ-equitable partitions

Input: Graph G, epsilon ϵ
Output: multiple ϵ-equitable partitions: $[\pi_1, \pi_2, ..., \pi_r]$

1: *spliCellSequences* = []
2: c_a = unit partition of G ▷ Initialize active cell as the vertex set of G
3: SORT(c_a) using function f as the comparison key ▷ Equation 4
4: Initialize a *sliding window* starting from the reverse of the sorted active cell. The initial window has the vertices with the largest value of the comparison key f.
5: Split the cells within ϵ using f as the comparison key, with the order captured by the position of the sliding window. Split in both the left and right directions of the window. This generates one instance of the split cell sequence, append it to *splitCellSequences*
6: Move the sliding window in the left direction to the set of vertices having the next largest value of the comparison key f.
7: Repeat Steps 5 and 6 until all starting points for the sliding window have exhausted or the required number of multiple cell orderings have been generated or no new sequence is added.
8: **for each** *cellSequence* in *splitCellSequences* **do**
9: Compute the corresponding ϵ-equitable partition using the Scalable ϵEP Algorithm from [4].
10: **end for**

(a)

Active Cell c_a	1	2	3	4	5	6	7	8	9	10
Function f	1	1	3	3	5	5	0	0	2	2

(b)

Active Cell c_a	7	8	1	2	9	10	3	4	5	6
Function f	0	0	1	1	2	2	3	3	5	5

(c)

Active Cell c_a	7	8	1	2	9	10	3	4	5	6
Function f	0	0	1	1	2	2	3	3	5	5

(d)

Active Cell c_a	7	8	1	2	9	10	3	4	5	6
Function f	0	0	1	1	2	2	3	3	5	5

Table 2: Example of Multiple Split Cell Sequences. (a) The initial active cell of a graph having 10 nodes and the corresponding example values of the function f (Equation 4). (b) The initial active cell sorted by using f as the comparison key. (c) First split cell sequence for an $\epsilon = 2$. (d) Second split cell sequence for an $\epsilon = 2$. The **dark line** under the cells in red colour depicts the starting position of the sliding window.

The algorithm to find Multiple ϵ-Equitable Partitions is given in Algorithm 1. The algorithm starts with generating *multiple cell sequences*. The process is illustrated with an example in Table 2. The example network has a vertex set V with 10 elements. Therefore, the initial active cell c_a is the unit partition of the vertex set and is equal to $[1, 2, 3, ..., 10]$. The corresponding dummy values of the function f (Equation 4) are depicted in Table 2(a). Table 2(b) shows the *sorted active cell* using f as the comparison key. Based on Algorithm 1 *line 4*, we initialize the *sliding window* with the vertices having largest value of the sorting key. These vertices are depicted with a **dark line** under the red cells in Table 2(c). Algorithm 1 *line 5* generates one instance of the split cell sequence. We move the sliding window in the left direction to the vertex set having the next largest value of the comparison key f (Algorithm 1 *line 6*). The new contents of the sliding window are depicted in Table 2(d). The Steps 5 and 6 of the Algorithm 1 repeat until the starting points for the sliding window have exhausted or the required number of multiple cell orderings have been generated or no new sequence is added. The cell colours in Table 2(c) and 2(d) depict the final split cell sequences for the example network using $\epsilon = 2$. Each sequence becomes an input to the second iteration of the Scalable ϵEP Algorithm from [4]. The Scalable ϵEP Algorithm generates an ϵEP for each of the input sequences (Algorithm 1, *lines 8-10*). The proposed method is *scalable*, since each of these ϵEPs can be computed in parallel and independent of each other in a distributed setting. The scalability results by the authors [4] on random power-law graphs ($2 \leq \gamma \leq 3$) show that practical running time complexity of the ϵEP algorithm is $O(n)$ for large sparse graphs.

For MϵEPs, generating r multiple ϵEPs will lead to a practical running time complexity of $O(r.n)$. Given that $r \ll n$, $O(r.n) \sim O(n)$. This implies that MϵEPs is highly scalable for large sparse graphs.

3.2 Positional Equivalence

Given the soft roles, MϵEPs categorizes the nodes based on the set of roles they play. We define the notion of positional equivalence in MϵEPs in this section.
Actor-Actor Similarity Score: We define the notion of *actor-actor similarity score*, which forms the basis for the definition of a "**Position in MϵEPs**". Given, nodes $v_i, v_j \in G$ and n multiple ϵ-equitable partitions $\{\pi_1, \pi_2, ..., \pi_n\}$ of G. The actor-actor similarity is defined as follows:

$$sim(v_i, v_j) = \frac{\sum_{k=1}^{n} cellMembers_{\pi_k}(v_i, v_j)}{n} \tag{5}$$

The function $cellMembers_{\pi}$ returns a value of 1 (`true`), if the nodes v_i and v_j both belong to the same *cell* of the partition π. Thus, the *similarity* score of two actors v_i and v_i is defined as the number of ϵEPs in which both v_i and v_j occupy a same position, divided by the total number of multiple ϵ-equitable partitions generated for the graph G.
Positional Equivalence in Multiple ϵ-Equitable Partitions Hierarchical Agglomerative Clustering (HAC) [7] performed using the *similarity* measure computed using pairwise actor-actor similarity scores (Equation 5) signifies **positional equivalence** in MϵEPs.

The actor *dendrogram* created from the hierarchical clustering based on the actor-actor pairwise similarity scores is a rich structure depicting the actor position merges at each level of the hierarchy. The strategy to obtain the most relevant partition, *i.e.*, to the *best level* for cutting the dendrogram tree is discussed next. Prior to that, we define the notion of mean-of-mean cell distance of a partition as follows. Given, the distance matrix *dist* (*distance* = 1.0 − *similarity*) and a clustering $\pi = \{c_1, c_2, ...c_m\}$. We compute the *mean-of-mean cell/block distance* of π as follows:

$$\mu(\pi) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{\binom{|c_i|}{2}} \sum_{v_j, v_k \in c_i} dist(v_j, v_k) \tag{6}$$

For each unique pair of vertices in a cell, we compute the mean of distances between these pairs and then we compute the mean of these means.
Best Level to Cut the Dendrogram Tree Given, the HAC linkage and threshold range $[\tau_1, \tau_2]$. We iterate over each level of the HAC linkage to evaluate the following. At each level we take the partition π corresponding to the clustering at that level of HAC. We compute the *mean-of-mean block distance* of the partition π using Equation 6. We then determine whether the rate-of-change of *mean-of-mean cell distance of* π, with respect to the *cluster combination distance* is (*i*) *greater* than the previously computed *slope and* (*ii*) is within a *threshold range* $[\tau_1, \tau_2]$. We update the *maximum slope* and the *best clustering level* seen so far accordingly.

We return the clustering/partition π at the level corresponding to the *best level*. The actors at each of the positions of this partition π correspond to the **best set of equivalent actors in MϵEPs**.

Significance of the Cutting Criteria Our measure to find best clustering is purely based on the distance between the data points. The goal of a clustering algorithm is to find clusters which have high similarity among cluster members, at the same time, clusters which are well separated. To address these, first, we restrict the *threshold range* for the *mean-of-mean cell distance of partition* as $[\tau_1 = 0.15, \tau_2 = 0.35]$. These thresholds correspond to a bound on mean-of-mean cell *similarity* of a partition in the range $[0.65, 0.85]$. Second, we compare the current iteration *slope*, with the *slope* computed from the previous level of the clustering hierarchy. Iteratively, we capture the drastic changes in the *slope* as *maximum slope*. Therefore, the *slope* and *threshold range* correspond to an objective function which finds a level in the tree having the following significance: *i*.) The mean of "the intra-position actor-actor similarity mean" for the all the positions in the partition is *at least* 65% and *ii*.) cut the dendrogram tree at a level prior to the level which has the **maximum** slope. This helps us to cut at the level after which there was a drastic change in the *slope*.

4 Experimental Evaluation

In this section we discuss the datasets, evaluation methodology and the results of our proposed approach.

4.1 Evaluation on Multi-Role Ground Truth Networks

Datasets

1. Internet Movie Database Co-Cast Network The Internet Movie Database (IMDb) is an online database of movies, television shows and video games. We create a movie Co-Cast network as follows: The cast members from *English Action* movies from the year 2013 form the nodes of the Co-Cast network. Using the complete IMDb database, two nodes i and j are linked with a weight w_{ij}. We discard links whose weight is less than 10, corresponding isolate nodes are also discarded. We use weighted network for *RolX* [5] to improve its node features and simple graphs for other approaches. IMDb dataset has 10 defined roles: *Actor, Director, Producer* etc. to name a few, these correspond to the *ground-truth roles*. The IMDb Co-Cast network consists of 7874 vertices and 11393 edges.

2. Summer School Network The Summer School Network (SSN) is a network of summer school attendees [8], the network has a link from vertex A to vertex B, if person A knew person B. For our evaluation, we assume the links to be symmetric and hence we convert the network to an undirected graph. We use directed graph for *RolX* to improve its node features. This graph has 73 vertices and 1138 edges. Further, the dataset has corresponding "roles" performed by each actor. Few roles defined for an actor are: *Senior Organizer, Local Organizer, Visitor, Speaker* etc., these correspond to the *ground-truth roles*.

Evaluation Methodology In multi-role networks, the actors can perform more than one *role*. To evaluate the positions performing *multiple roles*, we extend

IMDb	h	Pr	Re	N_p	N_a
MϵEPs-CL ($\epsilon = 5$)	**0.60**	**19.58**	84.90	**28**	**60**
MϵEPs-CL ($\epsilon = 10$)	0.74	18.22	95.26	22	48
MϵEPs-SL ($\epsilon = 5$)	0.76	16.85	95.22	16	37
MϵEPs-SL ($\epsilon = 10$)	0.75	16.12	93.21	12	28
ϵEP ($\epsilon = 5$)	0.71	19.28	95.56	22	51
ϵEP ($\epsilon = 10$)	0.78	17.85	**98.71**	16	37
Equitable	**0.30**	**25.22**	66.44	**27**	**58**
RolX	0.73	13.06	86.39	0	0
SBM	0.81	15.94	**100.0**	0	0

Summer School	h	Pr	Re
MϵEPs-CL ($\epsilon = 4$)	0.21	**50.66**	97.26
MϵEPs-CL ($\epsilon = 8$)	0.26	44.59	**100.0**
MϵEPs-SL ($\epsilon = 4$)	**0.19**	40.39	80.82
MϵEPs-SL ($\epsilon = 8$)	0.24	46.10	98.63
ϵEP ($\epsilon = 4$)	**0.19**	47.26	90.41
ϵEP ($\epsilon = 8$)	0.22	**48.26**	97.26
Equitable	–	–	–
RolX	0.28	41.23	**100.0**
SBM	0.22	47.40	**100.0**

Table 3: Ground-truth evaluation results on IMDb Co-Cast Network and Summer School Network. h denotes the *hamming loss* (lower the better), Pr and Re are *precision* and *recall* respectively. N_p denotes the number of positions correctly identified performing **exact multiple roles** and N_a denotes the number of actors at these positions.

the evaluation metrics used for studying *multi-label classification*. In multi-label classification [13], each example $x_i \in X$ is associated with a set of ground-truth labels $Y \subseteq L$, with $|L| > 2$. The *multi-label classifier* H predicts a set of labels $Z_i = H(x_i)$, for each of the examples $x_i \in X$.

Notion of Predicted Labels for Evaluating MϵEPs Since multi-label classification is a supervised learning approach and MϵEPs is an unsupervised approach, we extend the evaluation metrics as follows. The *predicted label set* Z of a vertex u, corresponds to the union of all the ground-truth labels of the vertices which belong to the position occupied by u. This can be mathematically defined as follows. Given, a vertex $u \in c_j$, where c_j is the cell of the partition π given by our method. The predicted label Z is defined as:

$$Z = \bigcup_{v \in c_j} \{r | r \in \boldsymbol{r_v} \text{ and } r \neq 0\} \tag{7}$$

where, $\boldsymbol{r_v}$ is the set of roles played by an actor v.

We **discard** the *singletons* from the study while computing each of these evaluation metrics. We do this to avoid an evaluation bias, since the predicted label and the actual label for singletons would always be the same. The metrics that we use are *hamming loss (h)*, *precision (Pr)* and *recall (Re)*.

Results on Ground Truth Networks We compare the results of positions given by MϵEPs with four other positional analysis approaches, namely, *equitable partition*, *RolX* [5], *ϵ-equitable partition* [6] and *stochastic blockmodel* (SBM) [8].

We use *complete-link* (CL) and *single-link* (SL) distance measures as *linkage criterion* for performing HAC in MϵEPs. The results for the IMDb network are depicted in Table 3. N_a and N_p correspond to number of actors and number of positions correctly identified *only* playing *exact multiple roles*. For N_a and N_p, both MϵEPs with *CL* $\epsilon = 5$ clustering and equitable partition perform better than other approaches. On the contrast, both SBM & *RolX* perform badly and fail to predict actors and exact positions performing multiple roles. The *hamming loss (h)* measure is best in case of EP and the performance of other methods is nowhere close to EP. This implies that most of the positions in EP performing

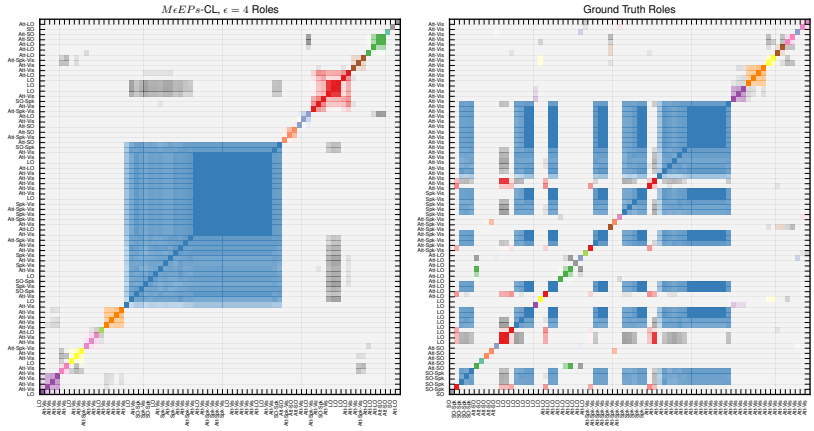


Fig. 2: Heatmap of actor-actor distance matrix of Summer School Network from MϵEPs-CL with $\epsilon = 4$. (a) The 73 nodes of the distance matrix are rearranged using the HAC with complete-link combination distance given by MϵEPs-CL. (b) The nodes rearranged by sorting *w.r.t. the ground-truth role lables*, the ground role labels could be zoomed-in for better viewing.

multiple roles, have fewer incorrect roles labeled. This is primarily attributed due to the fact that the non-singleton positions in EP have fewer number of actors as compared to other methods, hence the predicted labels assigned to a position are also smaller as compared to other methods. The *precision (Pr)* measure is best for EP, followed by MϵEPs with CL linkage, $\epsilon = 5$. The ϵEP with $\epsilon = 5$ from [6] is not far behind in performance. SBM is the best performer when it comes to the *recall (Re)* measure. The high recall in is attributed due to the fewer number of final positions: 40 in SBM. The final number of roles in *RolX* is only 3 (much less than the 10 ground-truth roles). This implies that both these methods are not suitable to perform the role analysis of larger networks, since they create a generalization around the actors in the network. From these evaluation metrics, equitable partition is a clear winner, followed by MϵEPs-CL with $\epsilon = 5$. A point worth mentioning here is that the EP of the Co-Cast Network of 7874 nodes has 1971 singleton positions, compared to 874 and 293 singletons in MϵEPs with complete-link and single-link clustering respectively for $\epsilon = 5$. Since the singleton positions don't provide any useful insights, ϵ-equitable partition based approaches are *tunable* according to the level of abstraction required in the study.

The *hamming loss (h)* measure is best for MϵEPs-SL and ϵEP for $\epsilon = 4$. This is primarily due to the more number of positions is both these partitions, which reduces the number of incorrect role annotations to other actors at same positions. For the *precision* metric, MϵEPs-CL for $\epsilon = 4$ is a clear winner, followed closely by ϵEP for $\epsilon = 4$ and SBM. The *recall* is high for SBM and *RolX*, since both of them have fewer positions as compared to the others. Overall, MϵEPs-CL with $\epsilon = 4$ is a clear winner, followed closely by ϵEP with $\epsilon = 8$, SBM and *RolX*.

Qualitative Analysis of Positions found using MϵEPs for the Summer School Network We present a qualitative analysis of the positions found with MϵEPs complete-link linkage for $\epsilon = 4$ on the SSN. Figure 2 shows the actor-actor distance matrix of summer school network from MϵEPs-CL with $\epsilon = 4$. Figure 2(a) depicts the 73 nodes of the distance matrix rearranged using the HAC complete-link linkage combination distance. Each colour represents a position assigned by MϵEPs, darker shades of the colour represent high similarity. Figure 2(b) depicts these nodes rearranged by sorting them *w.r.t. the ground-truth role labels*, the cluster colours and similarity values represent the MϵEPs positions; this makes it easy to study the of deviation in the multi-role positions found by MϵEPs with reference to the ground role labels. The ground-truth role labels can be zoomed-in on the axes. As evident from the diagram, majority of the multi-role positions identified by MϵEPs agree with the ground-truth roles; either the positions have fewer errors, or the bigger positions have been fragmented to smaller size positions. The qualitative evaluation along with the multi-class label metrics based in the previous subsection implies that the positions found by MϵEPs are both qualitatively and quantitatively superior than recent role discovery methods.

Results on Evolution of Ties/Links of Actor-Pairs We are primarily interested in finding out how the ties of actor pairs, which have common membership to a position in the network at time t, evolve over time with respect to their ties to other positions. For each pair of nodes that share a position in the time t network, we create a *difference degree vector* of their number of connections to all the other positions. For example, if node a has two connections to position p_1 and three connections to position p_2, and node b has one connection to position p_1 and four connections to position p_2, their *difference degree vector* $\boldsymbol{a} - \boldsymbol{b}$ would be $[1, -1]$. For same pair of actors, we compute their *difference degree vector* for the time $t + \delta t$ network. A natural way to compare the degree vectors of the t and $t + \delta t$ networks is using *cosine similarity*. We pad the smaller of the two vectors with *zeros* to make its dimension equal to the other one. We use the JMLR Co-Authorship network for the study. We generate 2 time evolving snapshots for the Co-Authorship network for years 2010 and 2011. Further, edges with number of co-authored or co-cited papers together with edge-weights less than 4 were pruned in the study. The year 2011 network has 1356 vertices and 2363 edges.

As evident from Figure 3, close to 40% actor-pairs follow their interaction patterns with same set of positions in the evolved network in MϵEPs. This signifies that a good number of actor-pairs, who add new collaboration links with time, mostly add them to similar social positions. The co-evolving actor-actor pairs found using ϵ-equitable partition based approaches show tie evolution properties. On the other hand, the co-evolving actors found using *RolX* and SBM have tie evolution difference degree vectors either independent or dissimilar to each other as evident from the plot. To conclude, MϵEPs performed better than all the other methods and successfully captured the co-evolution of node ties/links to other positions. The significance of tie evolution characteristics of our method can be easily used in link recommendation systems.

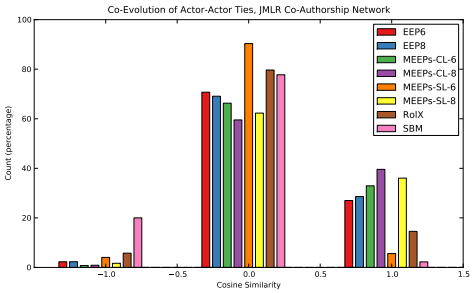


Fig. 3: Co-Evolution of Actor-Actor Ties for JMLR Co-Authorship Network from year 2010 to 2011. Cosine similarity value close to 1 signifies high degree of similarity in the evolution of links from year 2010 network to 2011 network. MϵEPs performs significantly better than other approaches.

5 Conclusions

In this paper we proposed a soft role discovery approach for networks. The primary contributions of our work are: *i*.) MϵEPs takes into account the *complete structural view of the graph* for computing the roles. *ii*.) MϵEPs is *scalable* to large sparse graphs. *iii*.) given the soft roles memberships of actors, MϵEPs categorizes the actors into equivalence classes or positions. *iv*.) to the best of our knowledge, this is the first attempt at evaluating roles and positions with real-world ground-truth datasets and using a methodology from multi-label classification for evaluation metrics. The future directions involve transforming the node memberships given by MϵEPs to numeric features and perform *hybrid* role discovery using global graph characteristics and local node features.

References

1. Borgatti, S., Everett, M.: Notions of Position in Social Network Analysis. Sociological Methodology 22(1), 1–35 (1992)
2. Everett, M.G.: Role Similarity and Complexity in Social Networks. Social Networks 7(4), 353–359 (1985)
3. Gilpin, S., Eliassi-Rad, T., Davidson, I.: Guided learning for role discovery (glrd): framework, algorithms, and applications. In: Proceedings of the 19th ACM SIGKDD. pp. 113–121. ACM (2013)
4. Gupte, P.V., Ravindran, B.: Scalable positional analysis for studying evolution of nodes in networks. SIAM Data Mining MNG Workshop (2014)
5. Henderson, K., Gallagher, B., Eliassi-Rad, T., Tong, H., Basu, S., Akoglu, L., Koutra, D., Faloutsos, C., Li, L.: RolX: structural role extraction & mining in large graphs. In: Proceedings of the 18th ACM SIGKDD. pp. 1231–1239. ACM (2012)
6. Kate, K., Ravindran, B.: Epsilon Equitable Partition: A Positional Analysis method for Large Social Networks. In: Proceedings of 15th COMAD (2009)
7. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval, vol. 1. Cambridge University Press (2008)
8. McDaid, A.F., Murphy, T.B., Friel, N., Hurley, N.J.: Improved Bayesian Inference for the Stochastic Block Model with application to Large Networks. Computational Statistics & Data Analysis 60, 12–31 (2013)
9. McKay, B.D.: Practical Graph Isomorphism. Congressus Numerantium 30 (1981)
10. N. K. Ahmed, J.N., Kompella, R.: Network sampling: From static to streaming graphs. TKDD pp. 1–45 (2013)
11. Rossi, R.A., Gallagher, B., Neville, J., Henderson, K.: Modeling dynamic behavior in large evolving graphs. In: Proceedings of the sixth ACM WSDM. pp. 667–676. ACM (2013)
12. Ruan, Y., Parthasarathy, S.: Simultaneous detection of communities and roles from large networks. In: Proceedings of the second edition of the ACM COSN. pp. 203–214. ACM (2014)
13. Tsoumakas, G., Katakis, I.: Multi-label Classification: An Overview. International Journal of Data Warehousing and Mining (IJDWM) 3(3), 1–13 (2007)
14. Wasserman, S., Anderson, C.: Stochastic a posteriori Blockmodels: Construction and Assessment. Social Networks 9(1), 1–36 (1987)
15. Wasserman, S., Faust, K.: Social Network Analysis: Methods and Applications. Cambridge University Press (1994)
16. White, D.R., Reitz, K.: Graph and Semigroup Homomorphisms on Semigroups of Relations. Social Networks 5, 193–234 (1983)