RolX: Structural Role Extraction & Mining in Large Graphs

Keith Henderson and Brian Gallagher
Lawrence Livermore National Laboratory
{keith, bgallagher}@llnl.gov

Tina Eliassi-Rad
Rutgers University
tina@eliassi.org

Hanghang Tong
IBM Watson
htong@us.ibm.com

Sugato Basu
Google Research
sugato@google.com

Leman Akoglu, Danai Koutra, Christos Faloutsos, and Lei Li
Carnegie Mellon University
{lakoglu, danai, christos, leili}@cs.cmu.edu

ABSTRACT

Given a network, intuitively two nodes belong to the same role if they have similar structural behavior. Roles should be automatically determined from the data, and could be, for example, "clique-members," "periphery-nodes," etc. Roles enable numerous novel and useful network-mining tasks, such as sense-making, searching for similar nodes, and node classification. This paper addresses the question: Given a graph, how can we automatically discover *roles* for nodes? We propose *RolX* (*Role eXtraction*), a scalable (linear in the number of edges), unsupervised learning approach for automatically extracting structural roles from general network data. We demonstrate the effectiveness of *RolX* on several network-mining tasks: from exploratory data analysis to network transfer learning. Moreover, we compare network role discovery with network community discovery. We highlight fundamental differences between the two (e.g., roles generalize across disconnected networks, communities do not); and show that the two approaches are complimentary in nature.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data mining; E.1 [**Data Structures**]: Graphs and networks

General Terms

Algorithms, Design, Performance, Experimentation.

Keywords

Graph mining, structural role discovery, network classification, similarity search, sense-making

1. INTRODUCTION

Given a network, we want to automatically capture the structural behavior (or function) of nodes via *roles*. Examples of possible roles include: centers of stars, members of cliques, peripheral nodes, etc. To this end, we propose a novel approach, called *RolX* (*Role eXtraction*), which automatically and effectively summarizes the behavior of nodes in large graphs. More precisely, *RolX* achieves the following two objectives. First, with *no* prior knowledge of the kinds of roles that may exist, it automatically determines the underlying roles in a network. Second, it appropriately assigns a mixed-membership of these roles to each node in the network. *RolX* is important because its roles form a basis for a number of novel and interesting network data analysis tasks such as network transfer learning, measuring structural similarity, sense-making (i.e., understanding the underlying behavior in a network), and network visualization. For instance, given two IP communication graphs from enterprise networks, we can use the extracted roles in one graph to build a relational classifier that can be used for a classification task in the other graph, effectively performing across-network classification or transfer learning on graphs.

RolX is an unsupervised learning approach for automatically extracting structural roles from general network data sets. Through extensive experiments, we demonstrate that the derived roles are effective in exploratory data analysis tasks (such as sense-making and node-similarity) and in prediction tasks such as across-network transfer learning. In the latter setting, we use the ability of roles to generalize behavior across networks as a way to perform network learning without relying on homophily, or on the availability of class labels in the target graph.

In our framework, roles are derived from structural features. In the absence of any other information, the problem of extracting roles from a network dataset is ill-defined, since there can be an infinite number of structural features that can be derived from the data. Once we choose a set of such features, the problem of role extraction can be well-formulated. Given a set of structural features that are extracted for a dataset, we define the role extraction problem as (1) finding the basis vectors in this structural feature space (where the number of basis vectors is determined by model selection), and (2) determining how much each network node belongs to each role. The structural features that are used for role extraction are domain-dependent. For example, for social networks, we propose a set of structural features that agree with the domain knowledge of sociologists.

The contributions of our work are as follows:

- *Effectiveness*: *RolX* enables numerous graph mining tasks, including:
  - *Role Generalization/Transfer Learning*: The structural roles of *RolX* generalize across disjoint networks.
  - *Structural Similarity*: *RolX* provides a natural

KDD'12, August 12–16, 2012, Beijing, China.

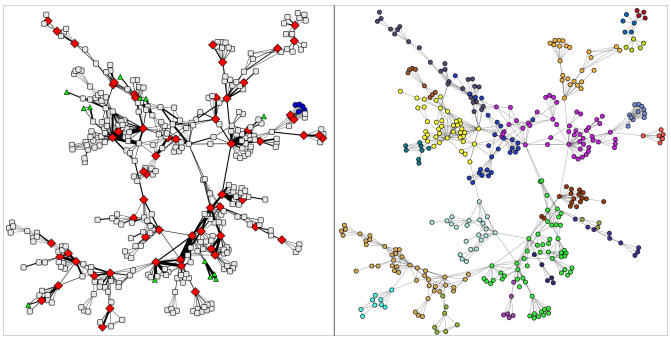


Figure 1: Role discovery and community discovery are complementary approaches to network analysis. Left: The 4 roles that *RolX* discovers on the largest connected component of the Network Science Co-authorship Graph: "bridge" nodes (as red diamonds), "main-stream" nodes (gray squares), etc - see text. Right: The 22 communities that Fast Modularity [6] finds on the same co-authorship graph. Roles capture node-level behaviors and generalize across networks whilst communities cannot.

way to determine similarity between nodes by comparing their role distributions.[1]

  - *Sense-making*: The structural roles of *RolX* can be understood intuitively by summarizing their characteristics (*NodeSense*) and their neighbors (*NeighborSense*).

- *Automation*: *RolX* is carefully designed to be fully automatic, without requiring user-specified parameters.
- *Scalability*: The runtime complexity of *RolX* is linear on the number of edges.

We want to emphasize that *RolX* as a role discovery approach is fundamentally different from (and complementary to) community detection: the former groups nodes of similar *behavior*; the latter groups nodes that are well-connected to each other.

Figure 1 depicts the difference between role discovery and community discovery for the largest connected component of a weighted co-authorship network [25]. *RolX* automatically discovers 4 roles vs. the 22 communities that the popular *Fast Modularity* [6] community discovery algorithm finds. *RolX* is a mixed-membership approach, which assigns each node a distribution over the set of discovered, structural roles. The node colors for *RolX* correspond to the node's primary role, and for Fast Modularity correspond to the node's community. Our four discovered roles represent these behaviors: "bridge" nodes (red diamonds) representing central and prolific authors, "main-stream" nodes (gray squares) representing neighborhoods of bridge nodes, "pathy" nodes (green triangles) representing peripheral authors with high edge-weight, and "tight-knit" nodes (blue circles) representing authors with many coauthors and homophilic neighborhoods.

The rest of the paper is organized as follows: proposed method, experimental results for the mining tasks outlined above, related work, and conclusions.

[1] *RolX* is a mixed-membership approach, which assigns each node a distribution over the set of discovered roles.

2. PROPOSED METHOD

Given a network, the goal of *RolX* is to automatically discover a set of underlying (latent) roles, which summarize the structural behavior of nodes in the network. *RolX* consists of three components: feature extraction, feature grouping, and model selection.

2.1 Feature Extraction

In its first step, *RolX* describes each node as a feature vector. Examples of node features are the number of neighbors a node has, the number of triangles a node participates in, etc. *RolX* can use any set of features deemed important. Among the numerous choices for feature extraction from graphs, we choose the structural feature discovery algorithm described in [15] since it is scalable and has shown good performance for a number of tasks. For a given node v, it extracts local and egonet features based on counts (weighted and unweighted) of links adjacent to v and within and adjacent to the egonet of v. It also aggregates egonet-based features in a recursive fashion until no informative feature can be added. Examples of these recursive features include degree and number of within-egonet edges, as well as aggregates such as "average neighbor degree" and "maximum neighbor degree." Again, *RolX* is flexible in terms of a feature discovery algorithm, so *RolX*'s main results would hold for other structural feature extraction techniques as well.

2.2 Feature Grouping

After feature extraction, we have n vectors (one per node) of f numerical entries each. How should we create groups of nodes with similar structural behavior/features? How can we make it fully automatic, requiring *no* input from the user?

We propose to use soft clustering in the structural feature space (where each node has a mixed-membership across various discovered roles); and specifically, an automatic version of matrix factorization.

Given a node-feature matrix $V_{n\times f}$, the next step of the *RolX* algorithm is to generate a rank r approximation $GF \approx V$ where each row of $G_{n\times r}$ represents a node's membership in each role and each column of $F_{r\times f}$ specifies how membership in a specific role contributes to estimated feature values. There are many methods to generate such an approximation (e.g., SVD, spectral decomposition) and *RolX* is not tied to any particular approach. For this study, we chose Non-negative Matrix Factorization because it is computationally efficient and non-negative factors simplify the interpretation of roles and memberships.

Formally, we seek two non-negative low rank matrices G and F to satisfy: $\text{argmin}_{G,F}\|V-GF\|_{fro}$, s.t. $G \geq 0,\ F \geq 0$, where $||\cdot||_{fro}$ is the Frobenius norm. The non-negativity constraint generally leads to a sparse, part-based representation of the original data set, which is often semantically more meaningful than other factorization methods. While it is difficult to find the optimal factorization of a matrix because of the non-convexity of the objective function, several efficient approximation algorithms exist (e.g., multiplicative update [18] and projective gradient decent [20]). *RolX* uses multiplicative update because of its simplicity. It is worth pointing out that *RolX* can naturally incorporate other variants of matrix factorization such as imposing sparseness constraint on F and/or G by incorporating some regularization terms in the objective function [10]). *RolX* can also use a general Bregman divergence [8] to measure approximation

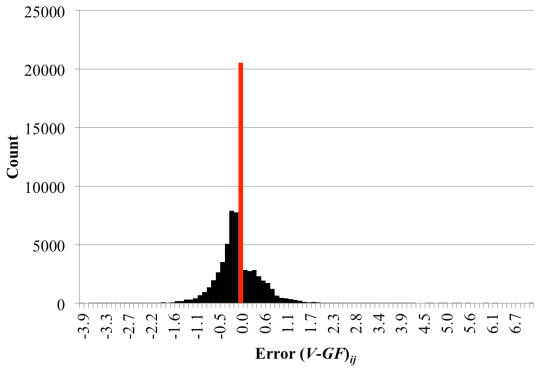


Figure 2: Errors do not appear to be normally distributed when approximating V in the Network Sciences Co-authorship graph. The spike at zero suggests that the Frobenius norm is not a good measure of model likelihood. *RolX* uses KL divergence to compute error description costs.

accuracy instead of using the Frobenius norm – this enables *RolX* to use other divergence functions (e.g., KL divergence) that can be more appropriate in some cases, as explained in Section 2.3.

One practical issue with feature grouping algorithms is that the model size (i.e., the number of roles) must be prespecified. In general, it is unrealistic to expect a practitioner to manually select an appropriate value for this parameter. Therefore, we explore several approaches for automatic model selection.

2.3 Automating the Method: Model Selection

Since roles summarize behavior, they can be used to compress the feature matrix V. We propose to use the Minimum Description Length criterion [27], to select the model size r that results in best compression. For a given model, we can compute the resulting description length in two parts: (1) the number of bits required to describe the model itself, and (2) the cost of describing the reconstruction errors in $V-GF$ (to achieve lossless compression). The selected model is the one that minimizes the description length $\mathcal{L}$, which is the sum of model description cost $\mathcal{M}$ and the coding cost (or equivalently, the cost of correcting the errors of our model) $\mathcal{E}$. That is, $\mathcal{L} = \mathcal{M} + \mathcal{E}$.

Assuming G and F are not sparse, the cost of describing the model using b bits per value is $\mathcal{M} = br(n+f)$.

How should we determine the representation cost of correcting errors in the reconstruction? That is, how do we compute the log-likelihood of a given model? Figure 2 suggests that the errors in $V-GF$ are not distributed normally, making the standard Frobenius norm a poor choice. Instead, we use KL divergence to compute the error description cost: $\mathcal{E} = \sum_{i,j}(V_{i,j} \log \frac{V_{i,j}}{(GF)_{i,j}} - V_{i,j} + (GF)_{i,j})$.

Because the model can contain high-precision floating point values, we combine Lloyd-Max quantization [22, 21] with Huffman codes [17] to increase compression. Note that with Huffman codes, the cost of encoding the model changes to $\bar{b}r(n+f)$, where $\bar{b}$ is the mean number of bits required per value. As a default, we choose $\log_2(n)$ quantization bins. However, the number of bins can be selected through parameter selection by choosing the number of bins and roles that minimizes description length. Figure 3 shows that for the Network Science Co-authorship graph, description lengths are minimized when we use $2^4 = 16$ quantization bins and 3 or 4 roles.

Log2(# Quantization Bins)

# Roles	1	2	3	4	5	6	7	8	9	10
1	37420	33855	31069	31347	31155	33015	34569	36579	39189	42098
2	35516	31433	27106	27572	27786	30492	33271	36562	40326	45127
3	32878	27769	23376	22149	24704	28723	32546	36486	41416	47242
4	37231	26295	23875	22198	25518	30485	34719	39282	44911	51674
5	46674	31456	25722	22245	25899	31398	35883	40947	46960	53941
6	46028	29327	27692	23849	29383	34753	39707	45016	51090	58691
7	40685	33922	29555	25561	30837	37667	42761	49399	56660	64456
8	42065	30322	28649	27526	34360	40804	47369	53842	61541	70812
9	39426	29422	29508	32557	38143	45765	52931	61155	69426	79495
10	45052	37736	31395	32844	41555	48523	56535	65402	74460	86220

Figure 3: Description length (in bits) is minimized when using 2^4 bins and 3 or 4 roles in the Network Science Co-authorship graph. Too many bins or roles increases the model description cost, while too few increases the cost of describing errors.

2.4 Computational Complexity

Let n be the number of nodes, m be number of edges, f = number of features, and r = number of roles.

LEMMA 1. *The running time complexity of* RolX *is linear on the number of edges, and specifically is* $O(mf + nfr)$.

PROOF. We give the complexity of each of the three steps of *RolX*.

Feature Extraction. This is $O(f(m+nf))$ [15].

Model Selection. The error computation takes $O(nrf)$ to multiply an $n \times r$ matrix by an $r \times f$ matrix. Quantization has a complexity of $O(nfilog(K))$, where K is the number of quantization bins and i is the number of iterations that we run the quantizer. Default for K is $log(n)$, so this becomes $O(nfilog(log(n)))$. Notice that the term of $O(log(log(n)))$ is a very small number. For example, for a graph with 1 billion nodes, this term is just about 3.1. So, the complexity for the quantization is roughly $O(nfi)$. There is also a term for the Huffman coding which is $O(nf + Klog(K))$. The $O(Klog(K))$ term is to build the tree that holds the codes. But this number is very small and therefore can be neglected.

Feature Grouping. We use the multiplicative update method for *RolX*, which has worst case complexity $O(nfr + nr^2 + fr^2) = O(nfr)$. □

2.5 Remarks

We experimented with several other options for clustering, model sizes criterion, and compression. For example, the information criterion proposed by Akaike (AIC) [1] can be used in place of compression. *RolX* can use as a drop-in replacement any other matrix factorization method, either in the usual form or the sparse counterpart (e.g., [10]).

Similarly, there are several representational choices for compressing floating points. We experimented with numerous of these choices, but we omit the details for brevity. Thus, unless otherwise stated *RolX* employs the feature extraction algorithm described in [15], non-negative matrix factorization for clustering, MDL for model selection, and KL divergence to measure likelihood.

	IP-A1	IP-A2	IP-A3	IP-A4	IP-B
# Nodes	81,450	57,415	154,103	206,704	181,267
(# labeled)	29,868	16,112	30,955	67,944	27,649
# Links	968,138	432,797	1,266,341	1,756,082	1,945,215
(# unique)	206,112	137822	358,851	465,869	397,925
% Web	32%	38%	38%	18%	42%
% DNS	36%	49%	39%	20%	42%
% P2P	32%	12%	23%	62%	16%

Table 1: Extracted real-world network trace data

		Test Graph			
		IP-A2	IP-A3	IP-A4	IP-B
Train Graph	IP-A1	✓	✓	✓	✓
	IP-A2		✓	✓	✓
	IP-A3			✓	✓
	IP-A4				✓
	IP-A1 to IP-A4				✓

Table 2: Across-network experiments performed on the network trace data

3. ROLE GENERALIZATION / TRANSFER LEARNING

In this section, we present experiments on role effectiveness for the across-network classification task (i.e., network transfer learning).

Data. We conduct experiments on two real-world data sets: IP communication networks and bluetooth proximity networks.

IP data: IP-A and IP-B are real network-trace data sets collected roughly one year apart on separate enterprise networks. The nodes are IP addresses and the links are communications between the IPs. The IP-A trace begins at midnight on day 1 and continues up to 12pm on day 5. The IP-B trace begins at midnight on day 1 and continues up to ≈5pm on day 6. For days 1-4 of the IP-A dataset (IP-A1 to IP-A4), we extract flows in the period from 12pm-1pm. We exclude day 5 because the trace ended at 12pm. For IP-B, we extract flows from 12pm-1pm for day 3 only. We then label all flows using a payload signature-based classification tool. Once network flows are labeled, we transfer labels to hosts by selecting the most frequent class-labels from among the host's flows. The payload classifier can distinguish between over 15 classes of traffic (e.g., Web, DNS, SMTP, P2P). However, since we found that 3 classes (namely, Web, DNS, and P2P) made up the dominant traffic type for over 90% of the labeled hosts, we remove all other labels and focus on the 3-class classification problem. Table 1 summarizes the data that we extracted. Notice the differences in the size and class distribution across these networks.

Reality Mining Device data: This dataset is constructed based on the data provided by the Reality Mining project [9]. That study was conducted in July 2004-June 2005 at the MIT Media Laboratory, with the participation of 94 human subjects using mobile phones pre-installed with several pieces of software, which recorded and sent to the research center various data including information about Bluetooth devices in proximity of approximately 5 meters. Subjects were tracked over 12 months and included students and faculty from the MIT Media Lab and the Sloan Business School. Within that period, about two million device scans were reported.

Classifiers. To test the predictive ability of the roles discovered by *RolX*, we use logistic regression with the *RolX* role memberships as features. In order to compare the predictive power of the *RolX* roles to the raw features that *RolX* uses as input, we also compare to a classifier that uses these raw features as input. The classifiers we compare are:

- *RolX*: a logistic regression model, which uses *RolX* role memberships as features
- *Feat*: a logForest model, which uses the same raw structural features as *RolX*, but does not decompose them into roles. The logForest is a bagged model, composed of a set of logistic regression classifiers, where each is given a subset of $\log(f) + 1$ of the f total features [13]. Note that for these raw features, logForest achieves vastly superior classification accuracy than logistic regression. Therefore, we omit logistic regression results for this classifier. See [15] for details.

The standard relational classifiers are not applicable for these transfer learning tasks since these methods rely on the availability of some known class labels in the test graph to seed the inference process. The test graphs here are completely unlabeled.

Methodology. For each experiment, the training graph has all known labels available. The test graph is *completely* unlabeled. Each classifier is evaluated on all known ground truth labels in the test graph. We use an identical set of roles for all data sets, which comes from running *RolX* on the IP-A1 data set. Table 2 summarizes the across-network experiments.

Results. We discuss predictive performance of roles next.

IP data: We ran *RolX* on a series of across-network transfer learning tasks (see Table 2). We train on one network where all known labels are available, and test on a separate network that is completely unlabeled. These tasks are difficult, given the (sometime extreme) differences in class distributions between data sets (see Table 1). The performance of the *Default* classifier is a good indicator of the difficulty of each task, since this model makes predictions based solely on the most frequent class from the training set. *Feat* uses the full set of 373 structural features extracted from IP-A1. *RolX* summarizes these features into 9 roles.

We see from Figure 4 that the roles produced by *RolX* effectively summarize the behavior of hosts in an IP network. In particular, *RolX* is able to generalize more effectively than *Feat* from network A to network B (*RolX* =85%, *Feat*=71% accuracy, p-value=0.01[2]). The results for transfer learning tasks across different days of network A are omitted since we cannot differentiate the performance of *RolX* and *Feat* (p-value=0.25).

Figures 5 and 6 demonstrate that the model selection criterion used by *RolX* is quite effective for our IP network classification task. *RolX* chooses a model size of 9 roles, which is right in the middle of the peak accuracy range shown in Figure 5. Figure 6a shows that the model selection criterion used by *RolX* is highly correlated with classification accuracy (Pearson correlation is -0.91). Figure 6b shows the default *RolX* model selection criterion decomposed into its constituent parts. *Model cost* is the cost associated with representing the model itself while *error cost* is the cost of

[2]The p-values are obtained from a one-tailed paired t-test.

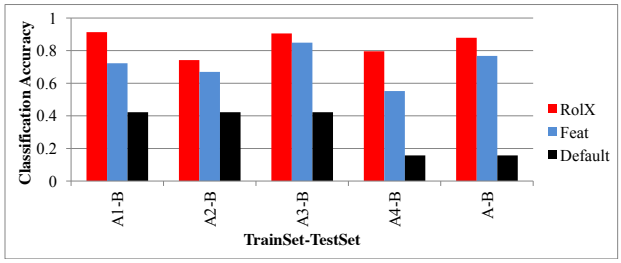


Figure 4: *RolX* provides better generalization performance between enterprise IP networks A and B (mean accuracy of *RolX* =85%, *Feat*=71%, p-value=0.01).

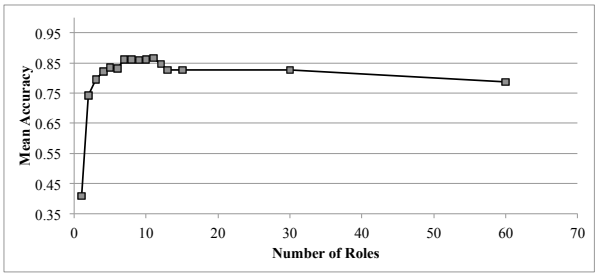
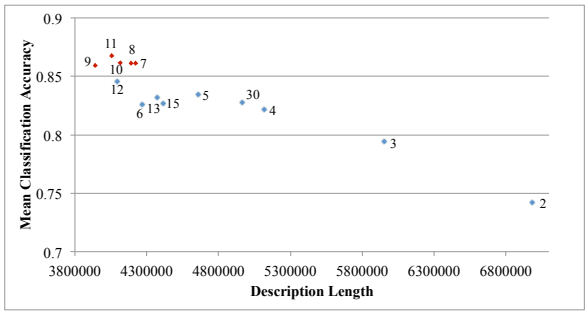
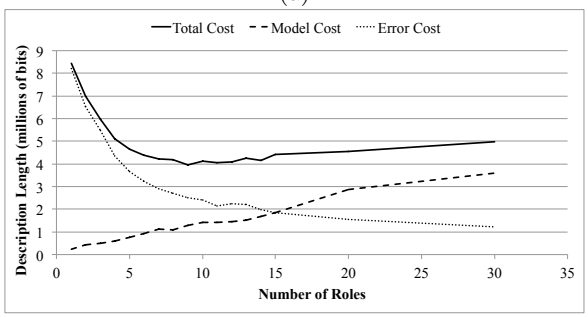


Figure 5: *RolX* chooses a high accuracy model size of 9 roles, in the middle of the peak accuracy range of 7-11 roles. The Y-axis depicts the mean classification accuracy using *RolX* (over all 4 test sets) by model size.



(a)



(b)

Figure 6: *RolX*'s model selection is effective: (a) Classification accuracy is highest when *RolX* selection criterion is minimized. Red markers indicate the peak performing model sizes of 7-11 roles (b) *RolX*'s model selection criterion balances model size and reconstruction accuracy.

representing the differences between the original feature values and the estimated feature values reconstructed using the model. As expected, we see a consistent increase in model cost and a consistent decrease in error cost as the number of roles increases.

Figure 7 shows that IP traffic classes are well-separated in the *RolX* "role space", with as few as 3 roles (extracted from the original 373 structural features). Note that we achieve even better separation with the automatically selected model size of 9 roles (see Figure 4), but we can only clearly visualize up to 3.

We omit for brevity the "sense-making" table for the 3-role IP experiment. It shows that Roles 1 and 2 are lower-volume IPs while Role 3 is high-volume servers or P2P nodes. Role 3 contains nodes of all three types (Web, DNS, P2P). This Role is overloaded since the model size of 3 is not as predictive as larger model sizes (see Figure 5).

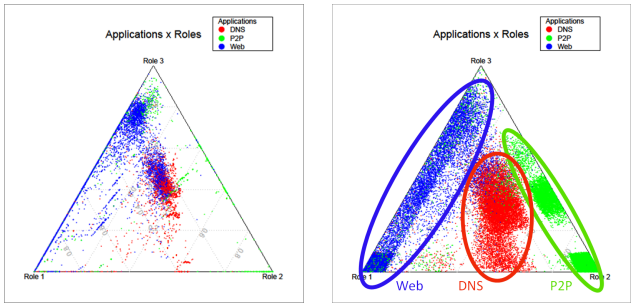
Reality Mining Device data: We conducted two sets of transfer learning experiments on this data. The first set of experiments involves a binary classification task where we try to predict whether a given subject is a business school student or not. The second set is similar, where we try to predict whether a subject is a graduate student in the Media Lab or not. As train and test sets, we used each pair of consecutive months in our dataset. In Figure 8, we show the accuracy of *RolX*. The Baseline is a classifier that learns to always predict the majority class of the training set on the test set. The time labels denote the month for the train data, and the month following that is used as the test data. We also use all the data in 2004 and 2005 as train and test data, respectively. Notice that *RolX* outperforms the baseline classifier most of the time with an average of 83% and 76% accuracy for the two experiment sets, respectively. We notice that *RolX*'s accuracy drops when September and May data is used as training, possibly because these months correspond to the start and end of the school semesters; the behavior of the subjects would be generally different than usual in these months, thus providing not as much predictive information as the other months would.

4. STRUCTURAL SIMILARITY

Here we describe experiments in which *RolX* is used for its most basic task: grouping nodes based on their structural similarity.

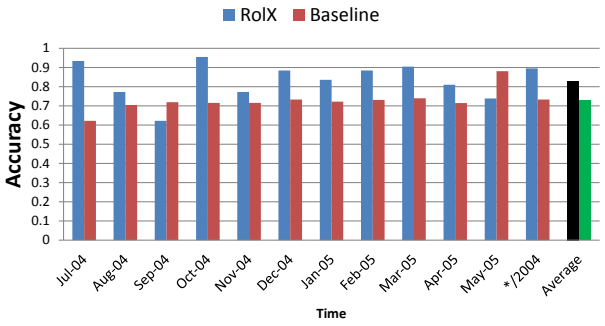
Network Science Coauthorship Data. Our first data set is a weighted co-authorship graph with 1589 authors (from the network science community) and 2743 weighted edges [25]. Figure 9 shows (a) the role-colored graph (where each node is colored by the primary role that *RolX* finds) and (b) the role affinity heat map. *RolX* finds four roles. Based on further analysis of the network and its roles, we discovered that the roles correspond to the following: (Disclaimer: The relative position in the graph does not reflect the total magnitude of contributions of the individual researchers. It is just a snapshot of networks-science-related data, and specifically in 2006.)

- *blue circle*: *tightly knit*, nodes that participate in tightly-

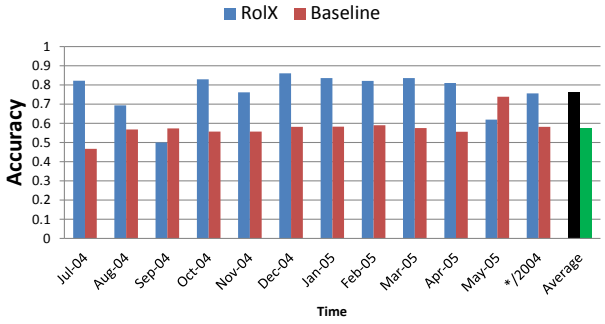


(a) (b)

Figure 7: IP traffic classes are well-separated in the *RolX* "role space" with as few as 3 roles. (a) Ternary plot showing the degree of membership of each DNS, P2P, and Web host in each of three roles. (b) Pseudo-density plot obtained by adding uniform noise to (a) to reveal overlapping points.



(a) Business Student vs. Rest

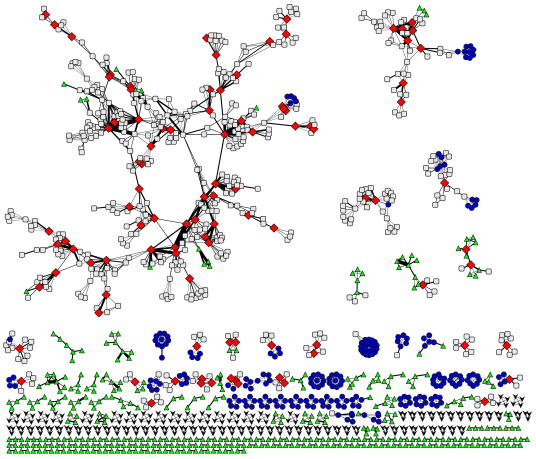


(b) Graduate Student vs. Rest

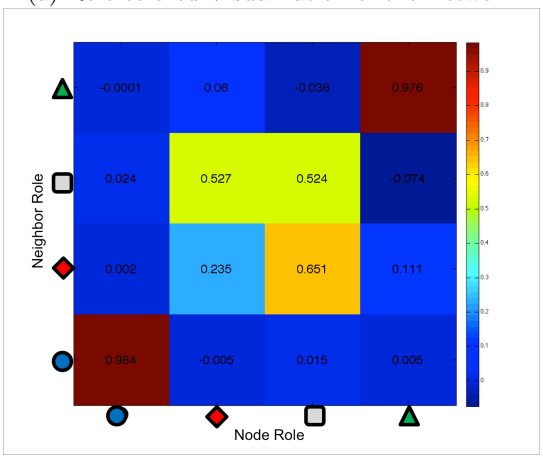
Figure 8: *RolX* (in blue) effectively generalizes behavior across time (higher is better). Figure shows results of across-network transfer learning on the Reality Mining Device dataset with *RolX*. Notice that *RolX* almost always performs well on the two different learning tasks with an average accuracy of 83% and 76%, respectively.

coupled groups. Examples are Andrei Broder and Christos Faloutsos.

- *red diamond*: *bridge* nodes, that connect groups of (typically, 'main-stream') nodes. Examples of bridges are Albert-Laszlo Barabasi and Mark Newman.



(a) Role-colored Visualization of the Network



(b) Role Affinity Heat Map

Figure 9: *RolX* effectively discovers roles in the Network Science Co-authorship Graph. (a) Author network *RolX* discovered four roles, like the heterophilous bridges (*red diamond*), as well as the homophilous "pathy" nodes (*green triangle*) (b) Affinity matrix (red is high score, blue is low) - strong homophily for roles #1 and #4.

- *gray rectangle*: *main-stream*, the vast majority of nodes - neither on a clique, nor a chain. Examples are John Hopcroft and Jon Kleinberg.
- *green triangle*: *pathy*, nodes that belong to elongated clusters. For example, Lada Adamic and Bernardo Huberman.

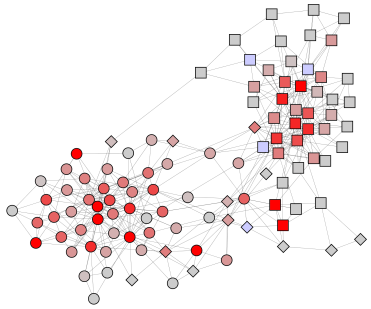
RolX's roles allow us to find similar nodes by comparing their role distributions. Figure 10 depicts node similarity for three (target) authors for the Network Science Co-authorship Graph: Mark Newman, F. Robert, and J. Rinzel. The primary roles for these three authors are different. Mark Newman's primary role is a broker (a prolific author); F. Robert's primary role places him in a tight-knit group (an author with homophilous neighborhood), and J. Rinzel's primary role places him in the periphery (an author with homophilous but "pathy" neighborhood). In each node-similarity picture, the target author is colored in yel-

low. The more similar nodes are red and less similar nodes are blue. Due to the generality of roles, *RolX* is able to find similar nodes across the entire graph even though it has many disconnected components.

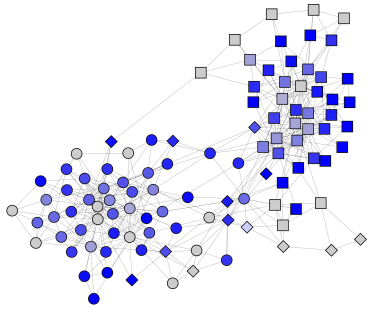
Political Books Co-purchase Data. We gave the 2000 Amazon Political Books Co-purchasing Network[3] to *RolX*. This graph has 105 nodes (representing books) and 441 edges (representing frequent co-purchasing of the books by the same buyers). *RolX* is able to effectively capture the purchasing behavior of customers by separating the "locally central" books from the "peripheral" books. Figure 11 depicts the "local central-ness" and "peripheral-ness" of the books.[4]

For the readers' convenience, we also present the human-provided labels for the nodes: conservative books (in circles), liberal books (in squares), and neutral books (in diamonds). Examples of highly central books are "Off With Their Heads" (conservative) and "Bushwhacked" (liberal). Highly peripheral books include "The Right Man" (conservative) and "Shrub" (liberal).

Node positions in Figure 11 are determined by a standard force-directed layout. Communities (conservative vs. liberal) are separated while nodes within a community are organized by role (central vs. peripheral) *RolX* can be used in conjunction with community discovery algorithms to find similar nodes in disparate communities or networks.



(a) Bright red nodes are locally central nodes.



(b) Bright blue nodes are peripheral nodes.

Figure 11: *RolX* effectively discovers roles in the 2000 Amazon Political Books Co-purchasing Network – illustrating how *RolX*'s roles can be used to find similar nodes in disparate communities. *RolX* finds two roles: locally central nodes and peripheral nodes. The redness of a node corresponds to its percentage membership in Role 1 (its "local central-ness"). Similarly, the blueness of a node corresponds to its percentage membership in Role 2 (its "peripheral-ness"). The node shapes corresponds to the human-provided labels of conservative (circle), liberal (square), and neutral (diamond).

5. SENSE-MAKING

To make "sense" of roles, we introduce two methods. One based on node measurements (*NodeSense*) and another based on neighbor measurements (*NeighborSense*).

NodeSense takes as input *RolX*'s node-by-role matrix, G, and a matrix of node measurements, M. For our experiments, we use the following node measurements: a node's degree, weighted degree, clustering coefficient, eccentricity (i.e., the longest geodesic from a node), PageRank, gatekeeper (i.e., whether a node is an articulation point for some pairs of nodes), local gatekeeper, pivot (i.e., a node with high betweenness), and structural hole (i.e., to what extent are a node's links redundant). *NodeSense* then computes a nonnegative matrix E such that $G \cdot E \approx M$. The matrix E represents the role contribution to node measurements. A default matrix E' is also computed by using $G' = ones(n, 1)$, where the n nodes belong to one role. Then, for each role r and for each measurement s, *NodeSense* computes $\frac{E(r,s)}{E'(r,s)}$. This ratio provides the role-contribution to node-measurements compared to the default contribution.

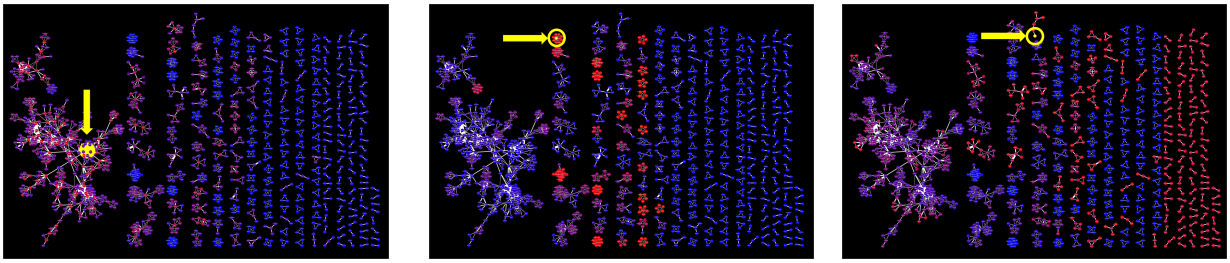
NeighborSense is similar to *NodeSense*–except instead of the matrix M, we use a neighbor matrix N, where the rows represent nodes and columns represent roles. $N(i, j)$ is the fraction of node i's neighborhood that is in role j. Then *NeighborSense* computes a nonnegative matrix Q such that $G \cdot Q \approx N$. The matrix Q represents the role affinities. A default matrix Q' is also computed using $G' = ones(n, 1)$, where the n nodes belong to one role. Then, for each pair of roles r_1 and r_2, *NeighborSense* computes $\frac{Q(r_1,r_2)}{Q'(r_1,r_2)}$. This ratio is the role-affinities compared to the default affinities.

Figure 12 depicts the results of *NodeSense* and *NeighborSense* on the Network Science Co-Authorship Graph. Recall that *RolX* found four roles on this graph. **Role 1** nodes (*blue circles* in Figure 9a) are authors with many coauthors and homophilic neighborhoods. They have high degree, high clustering coefficient, and high homophily. They are never gatekeepers (i.e. articulation points for some pairs of nodes) or pivotal nodes (i.e., with high betweenness). **Role 2** nodes (*red diamonds* in Figure 9a) represent authors who are central and prolific with high total weight. They have low clustering coefficient but high degree, high PageRank, and high affinity for Role 3 nodes (i.e., *gray rectangles*). Removal of Role 2 nodes severely interrupts graph connectivity because they are often gatekeepers and pivotal nodes. **Role 3** nodes (*gray squares* in Figure 9a) are peripheral authors. They have low degree and high eccentricity (i.e., nodes in the network periphery). They are almost never gatekeepers, but can be pivotal (i.e. they do not disconnect the graph, but they often increase geodesic lengths when removed). **Role 4** nodes (*green triangles* in Figure 9a) are isolated authors with high average edge weight and homophilic neighborhoods. Their links are not redundant (w.r.t. structural holes), but have low scores for most other measures (except homophily).

This sense-making analysis can be extremely useful for large networks, which cannot be easily visualized. By using

[3]Valdis Krebs, http://www.orgnet.com/.

[4]A third "super-peripheral" role is omitted for brevity. These nodes are gray in both (a) and (b).



(a) Node similarity for Mark Newman (b) Node similarity for F. Robert (c) Node similarity for J. Rinzel

Figure 10: *RolX* produces meaningful node-similarity across (disconnected) components of a graph. For each chosen node (in yellow circle), we use 'red' for the nodes with high structural similarity, and blue for the rest. (a) For Mark Newman (who is a "broker"), the red nodes are fairly central nodes, bridging others. (b) For F. Robert (who is from a tight-knit and homophilous neighborhood): the reds are clique members (of disconnected cliques) (c) For J. Rinzel (who is from a "pathy" but homophilous neighborhood): red nodes form small chains.

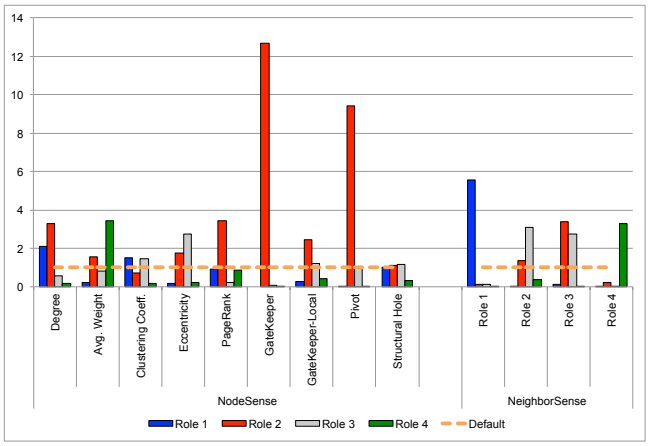


Figure 12: *NodeSense* and *NeighborSense* explain roles in the network science co-authorship graph. The *blue circle* authors (Role 1) are members of tight-knit communities and are role-homophilous. The *red diamond* authors (Role 2) are least homophilous. They are mostly brokers/bridges and are neighbors of *gray rectangles* (Role 3). The *gray rectangle* nodes (Role 3) are peripheral authors (i.e., have unusually high eccentricity). The *green triangle* nodes (Role 4) are isolated authors with disconnected coauthors.

role-homophily and topological measures, we can interpret roles regardless of network size.

6. RELATED WORK

Related research can be categorized into three parts: (1) graph features, (2) role discovery, and (3) transfer learning.

Graph Features. Features have been extracted from graphs for several data mining tasks. In [19], the authors propose extracting topological features for pairs of nodes for link prediction. In [14], the authors develop a multi-level framework to detect anomalies in time-varying graphs based on graph, sub-graph, and node level features. Their approach relies on extracting graph-level (global) features and tracking these metrics over time. In [2], the authors propose to detect abnormal nodes from weighted graphs based on features and patterns from egonets. There has also been work on using local and global structural features to improve the performance of network classifiers [12]. Some methods for feature extraction explicitly preserve the multi-cluster structure in the data [4]. Our work builds upon the approach of *recursive feature extraction* [15].

Role discovery. We propose using recursive graph features for role discovery of nodes. The task of role discovery has been studied in different types of graphs (e.g., social networks [23]). Different approaches have been used for role discovery, including Bayesian frameworks using MCMC sampling algorithm for learning multiple roles of data points [28], semi-supervised semantic role labeling [11], etc. These approaches do not scale up to handle large graphs.

There is another related body of work in role mining, a nice overview of which is given by Molloy et al. [24]. Role mining is somewhat different from the role discovery problem we discuss in this paper. It addresses the access permissions for different users in different roles in an access-controlled system. However, role mining algorithms use techniques similar to role discovery for inferring roles for nodes in a graph (e.g., hierarchical clustering).

In addition to using the inferred roles for exploratory graph mining (such as structural similarity and sense-making tasks), we use the inferred roles for improving classification. Previous approaches include using cluster structure for predicting class labels on graphs [16], and using cluster kernels for semi-supervised classification [30]. *RolX* is scalable.

RolX is different than generalized blockmodeling [26], which is commonly used in social network analysis. In particular, (1) *RolX* roles generalize across networks; (2) RolX is scalable; and (3) RolX incorporates local structure in addition to regular equivalence, and natively supports node attributes when they are available.

Transfer learning. Another aspect of our work is effective transfer learning in graphs. In the context of graph data, nonparametric models have been proposed for transfer learning for the task of collective link prediction [5]. The general framework of EigenTransfer constructs a graph to represent the source-target transfer learning task [7], while similarity matrix approximations of a hybrid graph have been explored

for transfer learning [29]. Relevant supervision has also been transferred across domains to improve the performance of clustering algorithms [3]. In all existing work, the features are *given* as the input and the goal is to leverage the *given* features to boost performance in the target domain.

7. CONCLUSIONS

Our main contribution is the careful design and extensive experimental validation of *RolX*, a novel role discovery approach with the following properties:

- *Effectiveness*: *RolX* enables numerous mining tasks, including: *Role Generalization/Transfer Learning*, (§ 3), *Structural Similarity* (§ 4), and *Sense-making* (§ 5).
- *Automation*: *RolX* is fully automatic, requiring no user-specified parameters.
- *Scalability*: Runtime is linear on the number of edges.

8. ACKNOWLEDGEMENTS

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract No. DE-AC52-07NA27344, and supported in part by IARPA via AFRL Contract No. FA8650-10-C-7061 and in part by DAPRA under SMISC Program Agreement No. W911NF-12-C-0028 and ARL No. W911NF-09-2-0053. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright annotation thereon. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of IARPA, AFRL, DARPA, or the U.S. Government.

9. REFERENCES

[1] H. Akaike. Information theory and an extension of the maximum likelihood principle. In *Proc. of the 2nd Int'l Symp. on Information Theory*, pages 267–281, 1972.

[2] L. Akoglu, M. McGlohon, and C. Faloutsos. Oddball: Spotting anomalies in weighted graphs. In *PAKDD*, pages 410–421, 2010.

[3] I. Bhattacharya, S. Godbole, S. Joshi, and A. Verma. Cross-guided clustering: Transfer of relevant supervision across domains for improved clustering. In *ICDM*, pages 41–50, 2009.

[4] D. Cai, C. Zhang, and X. He. Unsupervised feature selection for multi-cluster data. In *KDD*, pages 333–342, 2010.

[5] B. Cao, N. Liu, and Q. Yang. Transfer learning for collective link prediction in multiple heterogenous domains. In *ICML*, pages 159–166, 2010.

[6] A. Clauset, M. Newman, and C. Moore. Finding community structure in very large networks. *Phys. Rev. E*, 70:066111, 2004.

[7] W. Dai, O. Jin, G.-R. Xue, Q. Yang, and Y. Yu. Eigentransfer: a unified framework for transfer learning. In *ICML*, pages 193–200, 2009.

[8] I. S. Dhillon and S. Sra. Generalized nonnegative matrix approximations with Bregman divergences. In *NIPS*, pages 283–290, 2005.

[9] N. Eagle, A. Pentland, and D. Lazer. Inferring social network structure using mobile phone data. *PNAS*, 106(36):15274–15278, 2009.

[10] J. Eggert and E. Korner. Sparse coding and NMF. In *IJCNN*, pages 2529–2533, 2004.

[11] H. Fürstenau and M. Lapata. Semi-supervised semantic role labeling. In *EACL*, pages 220–228, 2009.

[12] B. Gallagher and T. Eliassi-Rad. Leveraging label-independent features for classification in sparsely labeled networks: An empirical study. *Lecture Notes in Computer Science*, 5498:1–19, 2010.

[13] B. Gallagher, H. Tong, T. Eliassi-Rad, and C. Faloutsos. Using ghost edges for classification in sparsely labeled networks. In *KDD*, pages 256–264, 2008.

[14] K. Henderson, T. Eliassi-Rad, C. Faloutsos, L. Akoglu, L. Li, K. Maruhashi, B. A. Prakash, and H. Tong. Metric forensics: A multi-level approach for mining volatile graphs. In *KDD*, pages 163–172, 2010.

[15] K. Henderson, B. Gallagher, L. Li, L. Akoglu, T. Eliassi-Rad, H. Tong, and C. Faloutsos. It's who you know: Graph mining using recursive structural features. In *KDD*, pages 663–671, 2011.

[16] M. Herbster. Exploiting cluster-structure to predict the labeling of a graph. In *ALT*, pages 54–69, 2008.

[17] D. Huffman. A method for the construction of minimum-redundancy codes. In *Proc. of the IRE*, pages 1098–1101, 1952.

[18] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[19] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla. New perspectives and methods in link prediction. In *KDD*, pages 243–252, 2010.

[20] C.-J. Lin. Projected Gradient Methods for Nonnegative Matrix Factorization. *Neural Computation*, 19(10):2756–2779, 2007.

[21] S. P. Lloyd. Least squares quantization in PCM. *IEEE Trans. on Info. Theory*, IT-28(2):129–137, 1982.

[22] J. Max. Quantizing for minimum distortion. *IRE Trans. on Information Theory*, IT-6:7–12, 1960.

[23] A. Mccallum, X. Wang, and A. Corrada-Emmanuel. Topic and role discovery in social networks with experiments on enron and academic email. *JAIR*, 30(1):249–272, 2007.

[24] I. Molloy, N. Li, T. Li, Z. Mao, Q. Wang, and J. Lobo. Evaluating role mining algorithms. In *SACMAT*, pages 95–104, 2009.

[25] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E.*, 74:036104, 2006.

[26] A. F. Patrick Doreian, Vladimir Batagelj. *Generalized Blockmodeling*. Cambridge University Press, 2005.

[27] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[28] M. Somaiya, C. Jermaine, and S. Ranka. Mixture models for learning low-dimensional roles in high-dimensional data. In *KDD*, pages 909–918, 2010.

[29] Z. Wang, Y. Song, and C. Zhang. Knowledge transfer on hybrid graph. In *IJCAI*, pages 1291–1296, 2009.

[30] J. Weston, C. Leslie, E. Ie, D. Zhou, A. Elisseeff, and W. S. Noble. Semi-supervised protein classification using cluster kernels. *Bioinformatics*, 21:3241–3247, 2005.