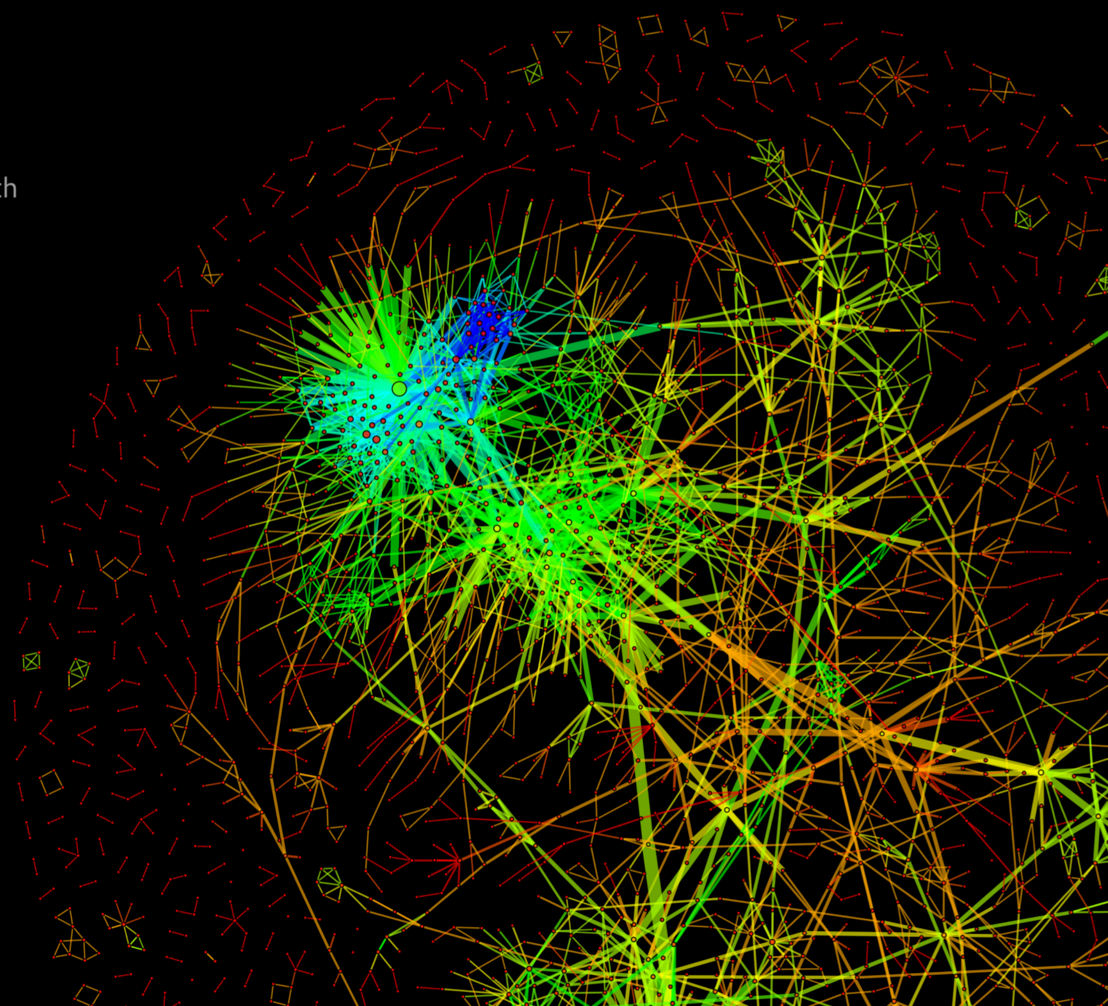
Relational Similarity Machines (RSM):
A Similarity-based Learning Framework for Graphs

Ryan A. Rossi
Adobe Research

Adobe Research

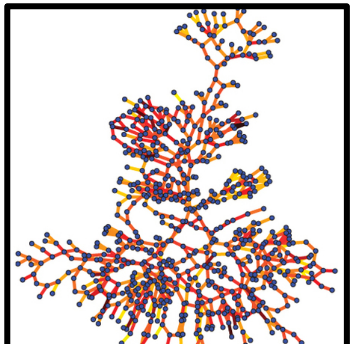


Joint work with:

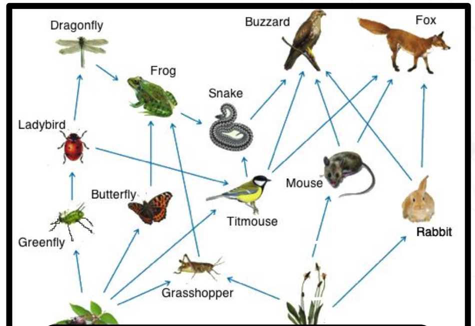
Rong Zhou (Google)
Nesreen K. Ahmed (Intel Labs)
Hoda Eldardiry (Xerox PARC)

Graphs – rich and powerful data representation

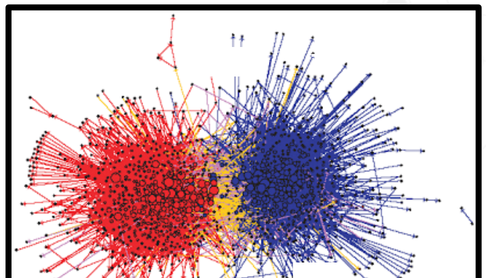
Social network



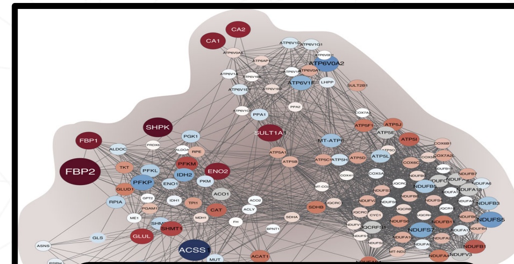
Power grid



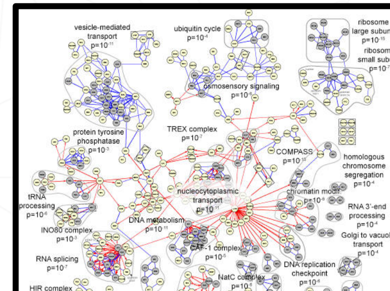
Food Web [2007]



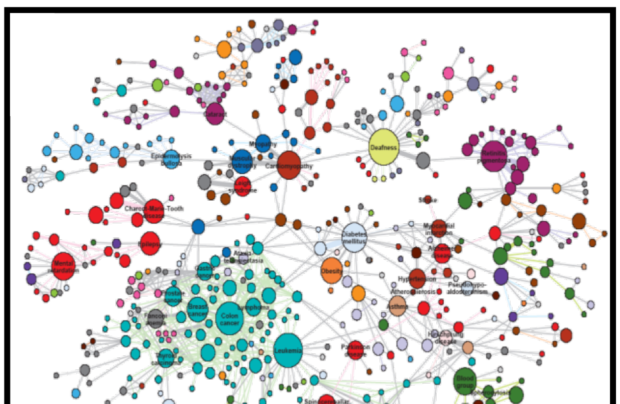
Political blogs



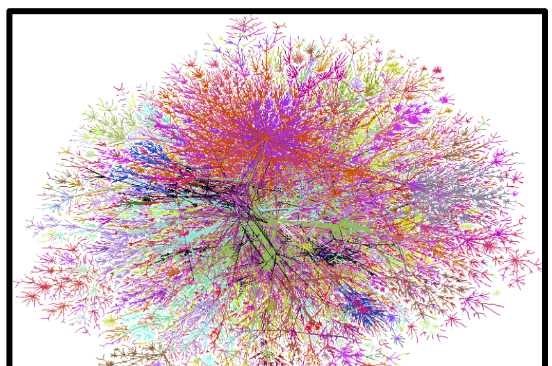
Protein Interactions [breast cancer]



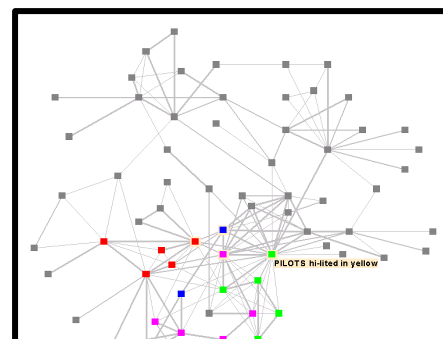
Gene Regulatory Network [Decourty 2008]



Human Disease Network [Barabasi 2007]



Internet (AS) [2005]



Terrorist Network [Krebs 2002]

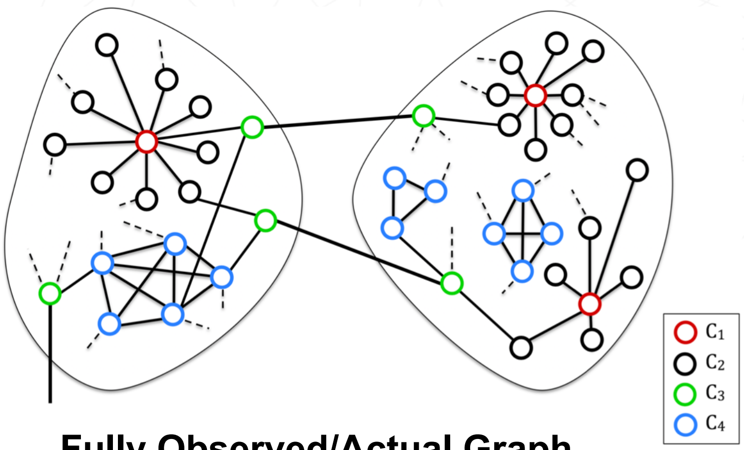
Problem & Motivation

- **Problem:** Given a graph with only a few labeled nodes, the goal is to predict the class labels of the unlabeled nodes
- **Issue:** Existing work assumes "smoothness" of labels over the graph/strong homophily ("birds of a feather flock together")
  - Works well when the neighboring labels of a node are **highly correlated** (i.e., the neighbors of a node have the same class label), and perform poorly otherwise
- Reverse is often true in actual data

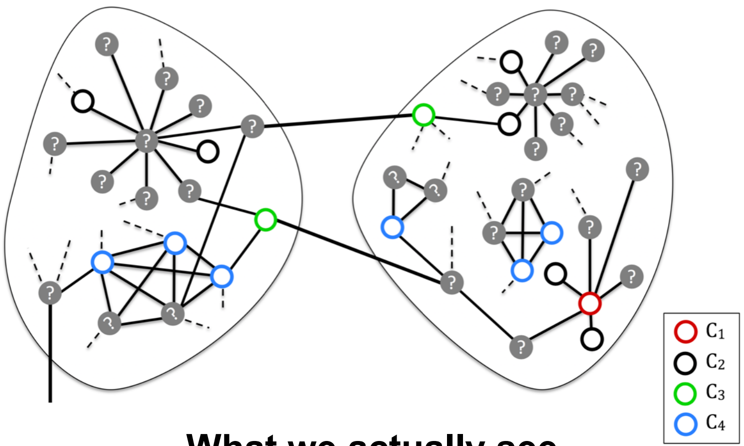
Heterophily Example

Class labels may not be correlated with neighbors (e.g., protein & molecular graphs).

- RSM generalizes across the spectrum of relational data characteristics, and avoids the issues, assumptions, and limitations of existing methods.
- Class of a node may be correlated with nodes that have ”similar” structural properties and more generally features (e.g., non-relational attributes, …).
- Existing methods based on diffusion/label propagation would fail (perform poorly)



Fully Observed/Actual Graph



What we actually see

This work

- Addresses limitation of previous work that assumes strong homophily between neighboring labels of a node
- Proposed general framework for graphs with arbitrary relational autocorrelation (**homophily** or **heterophily**)

Other Contributions:

- *Principled similarity-based relational learning framework* based on the notion of *maximizing similarity*.
- Fast, parallel, space-efficient, and flexible with many interchangeable components
- Designed for large-scale supervised *and* semi-supervised learning in *noisy* multi-dimensional networks

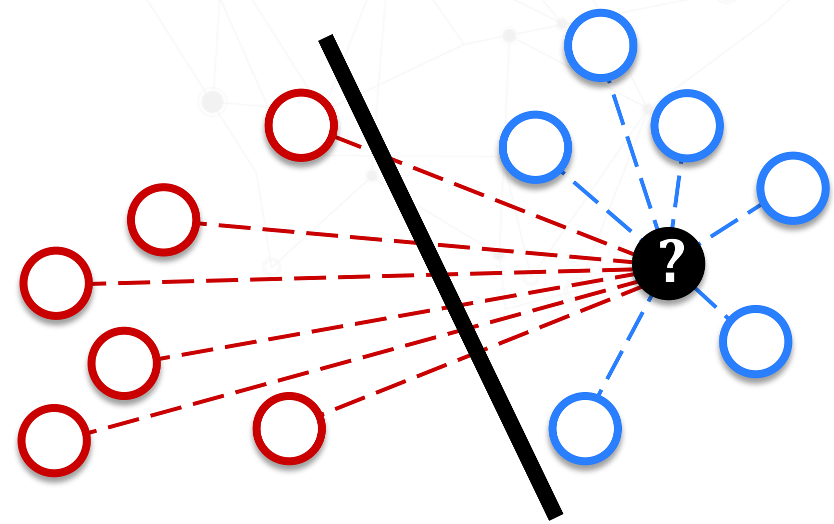
Non-relational example (RSM-IID)

The similarity of the labeled nodes in **X** of class k with respect to the test node feature vector $\mathbf{z}_i$ is formalized as,

$$w_k = \sum_{\mathbf{x}_j \in \mathcal{X}_k} \Phi \langle \mathbf{z}_i, \mathbf{x}_j \rangle$$

$$\Phi(\mathbf{z}_i, \mathbf{x}_j) = \exp\left(- \frac{\|\mathbf{z}_i - \mathbf{x}_j\|^2}{2\sigma^2} \right)$$

Repeated for each class k to obtain **w**



After computing **w**, then $\mathbf{z}_i$ is assigned to the class with maximum similarity

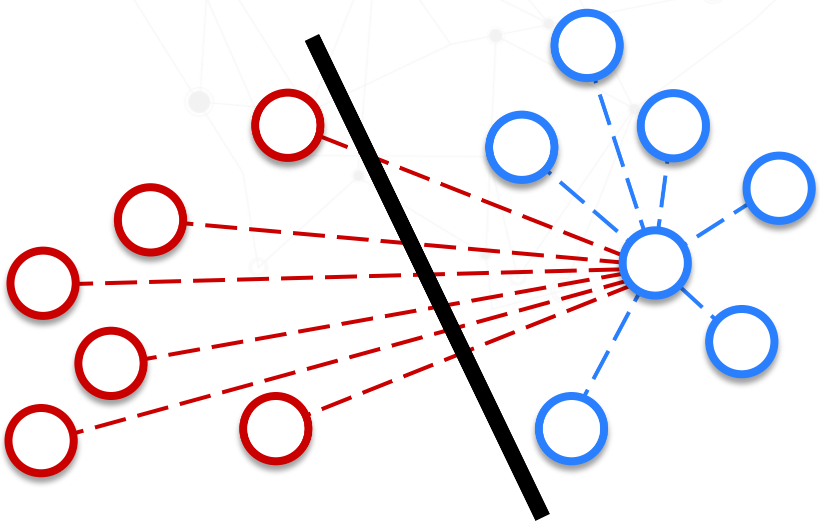
$$\xi(\mathbf{z}_i) = \underset{k \in \mathcal{C}}{\arg\max} \;\; w_k$$

Non-relational example (RSM-IID)

The similarity of the labeled nodes in **X** of class k with respect to the test node feature vector $\mathbf{z}_i$ is formalized as,

$$w_k = \sum_{\mathbf{x}_j \in \mathcal{X}_k} \Phi \langle \mathbf{z}_i, \mathbf{x}_j \rangle$$

$$\Phi(\mathbf{z}_i, \mathbf{x}_j) = \exp\left(- \frac{\|\mathbf{z}_i - \mathbf{x}_j\|^2}{2\sigma^2} \right)$$



Repeated for each class k to obtain **w**

After computing **w**, then $\mathbf{z}_i$ is assigned to the class with maximum similarity

$$\xi(\mathbf{z}_i) = \arg\max_{k \in \mathcal{C}} \; w_k$$

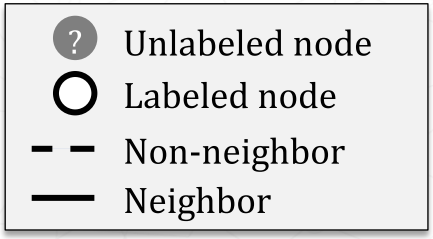
Supervised & Semi-Supervised Learning Components

- Supervised component:
  1. Use the nodes with known class labels to predict the class labels of the unknown nodes

- Semi-supervised components:
  1. Iteratively estimate the nodes with unknown class labels, and at each iteration we label a fixed percent of nodes with highest confidence
  2. Sample a small number of unlabeled nodes NOT connected to the node currently being estimated, and weight the similarity by the current class probabilities

Intuition

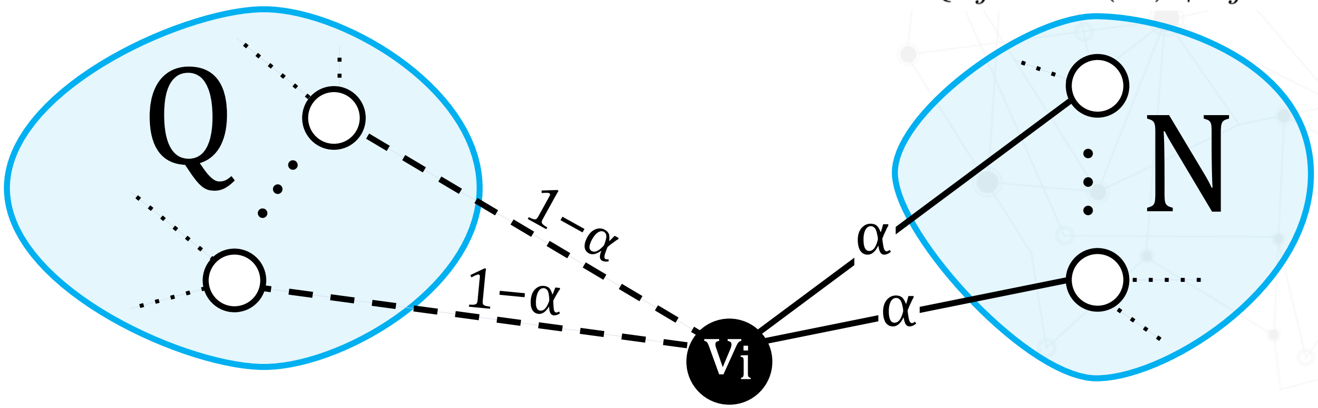
V^ℓ = nodes with **known labels**
V^u = nodes with **unknown labels**



N and Q = **labeled** neighbors *and* non-neighbors

$Q = \{v_j \notin \Gamma_h(v_i) \mid v_j \in V^\ell\}$

$N = \{v_j \in \Gamma_h(v_i) \mid v_j \in V^\ell\}$



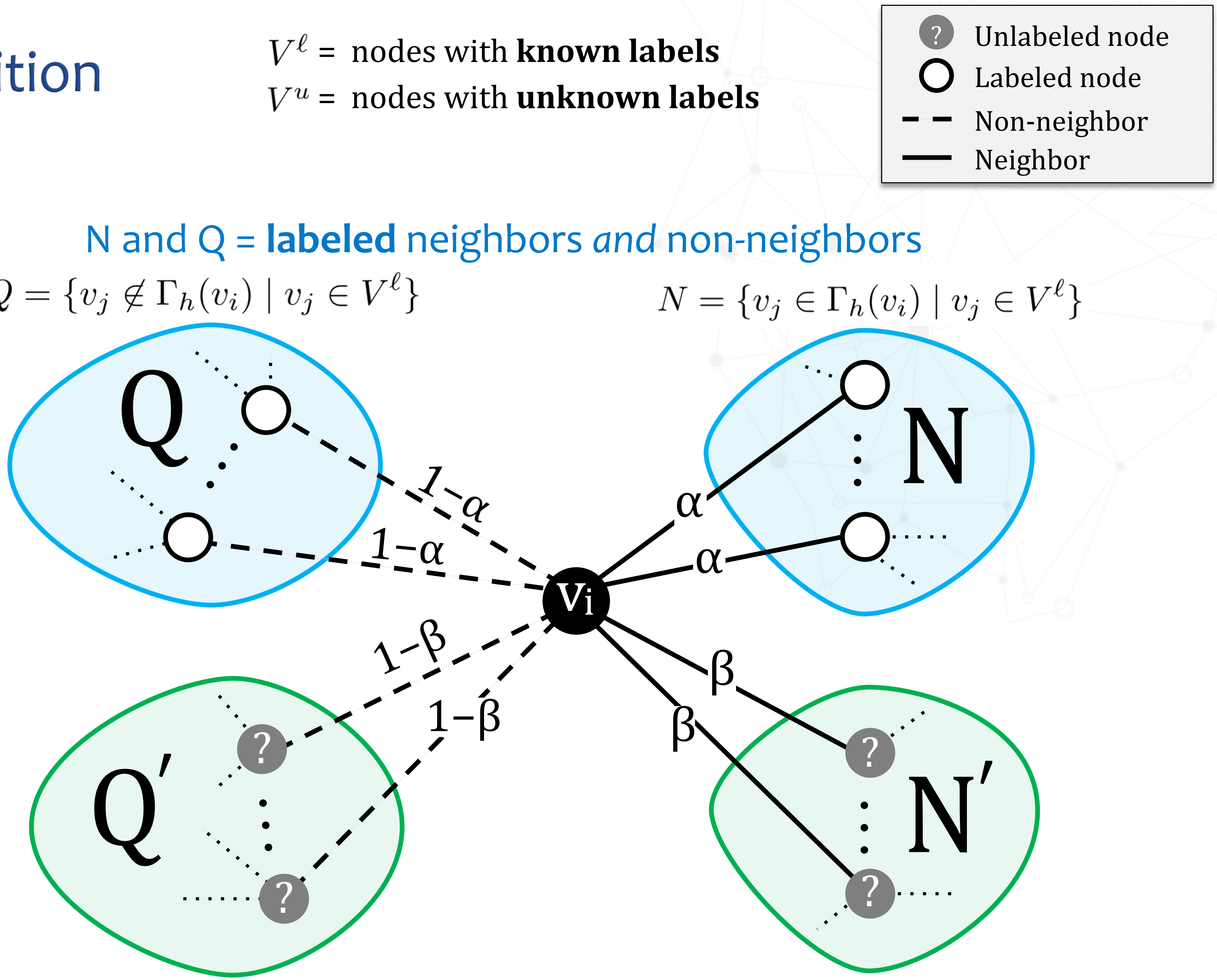
Intuition

V^ℓ = nodes with **known labels**
V^u = nodes with **unknown labels**

N and Q = **labeled** neighbors *and* non-neighbors

$Q = \{v_j \notin \Gamma_h(v_i) \mid v_j \in V^\ell\}$

$N = \{v_j \in \Gamma_h(v_i) \mid v_j \in V^\ell\}$



$Q' = \{v_j \notin \Gamma_h(v_i) \mid v_j \in V^u\}$

$N' = \{v_j \in \Gamma_h(v_i) \mid v_j \in V^u\}$

N’ and Q’ = **unlabeled** neighbors *and* non-neighbors

Leveraging Nodes with **Known** Labels

The similarity of the **labeled** nodes in **X** of class k with respect to the test node feature vector $\mathbf{z}_i$ is:

$$w_{ik}^{R} = \sum_{\mathbf{x}_j \in \mathcal{X}_k \wedge v_j \in N} p_{ik} \Phi \langle \mathbf{z}_i, \mathbf{x}_j \rangle$$

$$w_{ik}^{I} = \sum_{\mathbf{x}_j \in \mathcal{X}_k \wedge v_j \in Q} p_{ik} \Phi \langle \mathbf{z}_i, \mathbf{x}_j \rangle$$

Repeated for each class k

Obtain a set J by sampling $(V^{\ell} \cup U)$ via an arbitrary (weighted/uniform) distribution F (if needed, otherwise $J = V^{\ell} \cup U$)

parallel for each $v_j \in J$ ▷ Supervised component

    Set s_{ij} to be $\Phi\langle \mathbf{z}_i, \mathbf{x}_j \rangle$

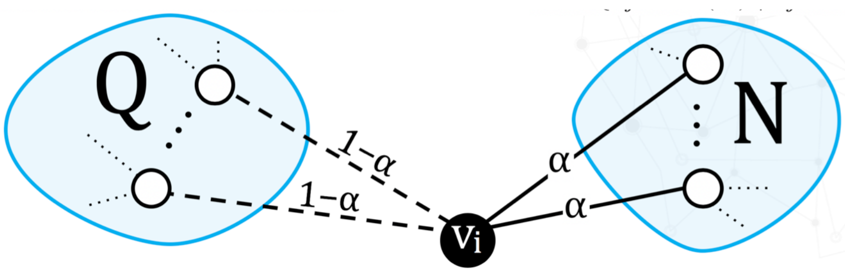
    Let $k \in \{1, \ldots, |\mathcal{C}|\}$ be the class label of $v_j \in V^{\ell}$

    if $v_j \in \Gamma_h(v_i)$ **then**

        Update $w_{ik}^{R} \leftarrow w_{ik}^{R} + p_{ik} \cdot s_{ij}$ ▷ labeled *neighbor*

    else Update $w_{ik}^{I} \leftarrow w_{ik}^{I} + p_{ik} \cdot s_{ij}$ ▷ labeled *non-neigh.*

end parallel



Leveraging Nodes with **Unknown** Labels

- How to make use of unlabeled nodes?

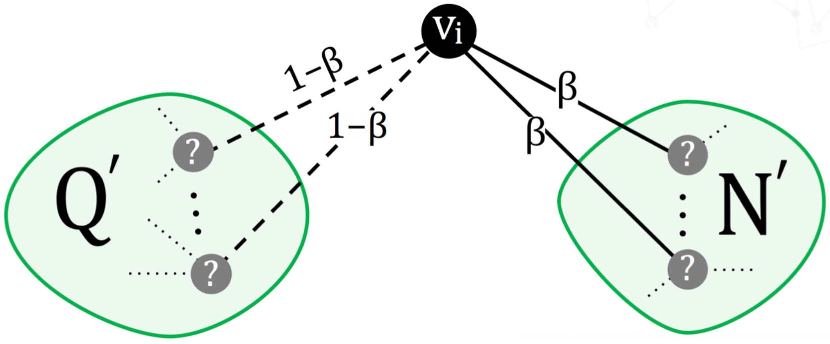
The similarity of the **unlabeled** nodes in **Z** with respect to the test node feature vector $\mathbf{z}_i$ is:

$$\mathbf{m}_i^R = \sum_{v_j \in N'} (\mathbf{p}_i \odot \mathbf{p}_j) \cdot \Phi\langle \mathbf{z}_i, \mathbf{z}_j \rangle$$

$$\mathbf{m}_i^I = \sum_{v_j \in Q'} (\mathbf{p}_i \odot \mathbf{p}_j) \cdot \Phi\langle \mathbf{z}_i, \mathbf{z}_j \rangle$$

Obtain a set J by sampling $(V^u \setminus U)$ via an arbitrary (weighted/uniform) distribution F (if needed, otherwise $J = V^u \setminus U$)
parallel for each $v_j \in J$ ▷ Semi-supervised component
  Set s_{ij} to be $\Phi\langle \mathbf{z}_i, \mathbf{z}_j \rangle$
  for each class $k \in \mathcal{C}$ **do**
    if $v_j \in \Gamma_h(v_i)$ **then**
      $m_{ik}^R \leftarrow m_{ik}^R + s_{ij}(p_{ik}p_{jk})$ ▷ unlabeled *neighbor*
    else $m_{ik}^I \leftarrow m_{ik}^I + s_{ij}(p_{ik}p_{jk})$ ▷ unlabel. non-neigh.
end parallel

Weight similarity by the probability that node i and j belong to the same class k



Maximizing Relational Similarity

$$\mathbf{w}_i = \overbrace{\underbrace{\alpha \mathbf{w}_i^R}_{\text{Relational}} + \underbrace{(1-\alpha)\mathbf{w}_i^I}_{\text{IID}}}^{\text{Supervised}}$$

$$\mathbf{m}_i = \overbrace{\underbrace{\beta \mathbf{m}_i^R}_{\text{Relational}} + \underbrace{(1-\beta)\mathbf{m}_i^I}_{\text{IID}}}^{\text{Semi-Supervised}}$$

$$\mathbf{p}_i^{(\tau+1)} = \overbrace{\left(\underbrace{\mathbf{w}_i}_{\text{Supervised}} + \underbrace{\mathbf{m}_i}_{\text{SSL}} \right)}^{\text{current estimate}} + \overbrace{\mathbf{p}_i^{(\tau)}}^{\text{previous}}$$

After computing $\mathbf{p}_i$, then $\mathbf{z}_i$ is assigned to the class with maximum similarity.

$$\xi(\mathbf{z}_i) = \arg\max_{k \in \mathcal{C}} \; P_{ik}$$

Similarity Functions

- Radial Basis Function (RBF):

$$\Phi(\mathbf{z}_i, \mathbf{x}_j) = \exp\left(- \frac{\|\mathbf{z}_i - \mathbf{x}_j\|^2}{2\sigma^2} \right)$$

- Polynomial functions:

$$\Phi(\mathbf{z}_i, \mathbf{x}_j) = \left(\langle \mathbf{z}_i, \mathbf{x}_j \rangle + c \right)^q$$

- Sigmoid kernel:

$$\Phi(\mathbf{z}_i, \mathbf{x}_j) = \tanh\left(a \langle \mathbf{z}_i, \mathbf{x}_j \rangle + c \right)$$

…among many other possibilities…

Complexity Analysis

n = # of nodes
d = # of features
k = # of class labels

- **Computational Complexity:**

Reduced by sampling from the different types of nodes

- Single test instance: $\mathcal{O}(|J| \cdot d)$ where $|J| \ll n$

- **Space Complexity:**
  - Given a test node to predict, RSM takes $\mathcal{O}(k)$ space
  - If collective inference is used, then RSM takes $\mathcal{O}(nk)$

In addition to graph and features

Experiments

Experimental setup

- Model selected via grid search

$$\alpha, \beta \in \{0, 0.01, 0.1, 0.25, 0.5, 0.75, 0.99, 1\}$$
$$\sigma \in \{0.001, 0.01, 0.05, 0.1\}$$

- 10% of the labeled nodes as training
- Results are averaged over 20 trials
- Attributes = degree, 3- and 4-node graphlets

Heterophily Experiments

- RSM outperforms WVRN, SVM-G, and RPT across all graphs with a mean gain of **24%, 52%, and 17%**, respectively
- RSM is effective for graphs with heterophily

HETEROPHILY CLASSIFICATION RESULTS

			F1 score				
Graph	$\lvert\mathcal{C}\rvert$	$\mathbb{L}$	RSM	RSM-IID	WVRN	SVM-G	RPT
DD645	20	2.0%	**0.769**	0.701	0.607	0.634	0.680
DD411	20	12.5%	**0.712**	0.695	0.605	0.239	0.615
DD5	19	6.7%	**0.561**	0.510	0.303	0.426	0.465
DD244	20	7.1%	**0.742**	0.708	0.558	0.175	0.646
DD159	20	2.70%	**0.734**	0.702	0.605	0.280	0.652
DD185	18	12.5%	**0.738**	0.644	0.585	0.262	0.489

homophily measure

$$\mathbb{L}(G) = {}^{1}/_{|E|} \sum_{(v_i,v_j)\in E} \mathbb{L}(v_i, v_j)$$

$\mathbb{L}(v_i, v_j) = 1$ if class labels of v_i and v_j match, otherwise 0

Homophily Experiments

- RSM significantly outperforms the other methods
- Effective for classification on graphs with homophily
- Flexible for prediction in graphs with arbitrary relational autocorrelation (homophily or heterophily)

CLASSIFICATION RESULTS FOR GRAPHS WITH HOMOPHILY

			Accuracy				
Graph	$\|\mathcal{C}\|$	$\mathbb{L}$	RSM	RSM-IID	WVRN	SVM-G	RPT
aff–polbooks	3	66.6%	**89.25**	71.16	74.41	69.89	70.97
bio–Gene	2	79.7%	**85.27**	64.88	72.41	61.85	60.24
Enzymes349	2	50.0%	**77.81**	67.93	62.50	55.36	66.07
musicGenre	8	84.9%	**82.35**	51.57	60.59	48.63	55.56

homophily measure

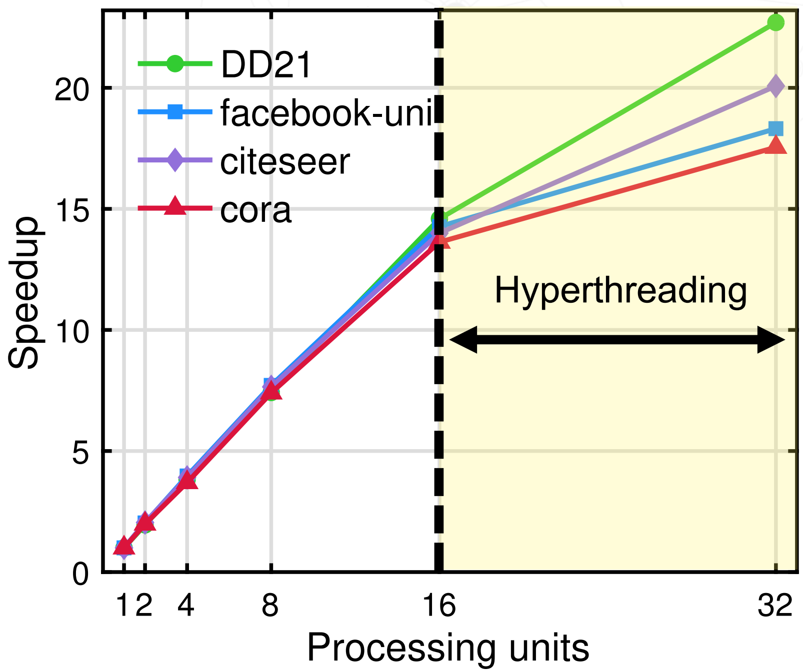
$$\mathbb{L}(G) = {}^{1}/_{|E|} \sum_{(v_i,v_j)\in E} \mathbb{L}(v_i, v_j)$$

$\mathbb{L}(v_i, v_j) = 1$ if class labels of v_i and v_j match, otherwise 0

Parallel Scaling

$$S_p = \frac{T_{\text{seq}}}{T_p}$$

- Strong scaling results
- RSM scales nearly linearly as the number of processors increases



Note a machine with two Intel Xeon E5-2687 CPUs @3.10GHz were used with 8 cores each.

Runtime Performance

- RSM is fast with real-time response times
  - in the range of a few ms or less
- Able to support real-time interactive learning & inference

AVERAGE TIME PER TEST INSTANCE

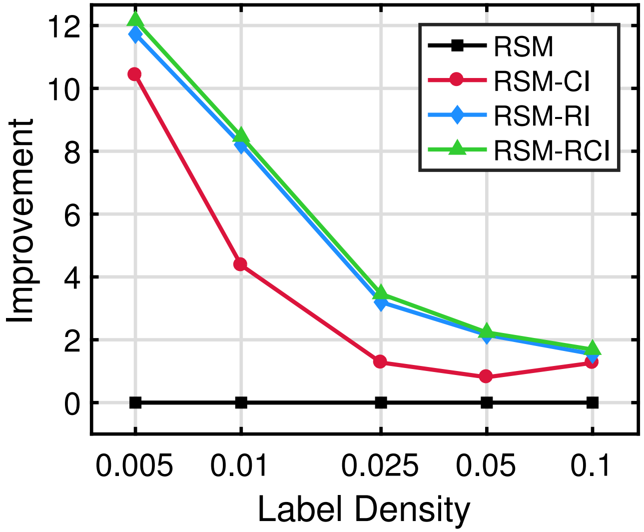
Graph	$\lvert V^{\ell} \rvert$	$\lvert \mathcal{C} \rvert$	**avg. time**
soc–TerroristRel	90	2	0.18 ms
aff–polbooks	12	3	0.12 ms
cora	272	7	0.97 ms
DD6	417	20	3.10 ms
political-retweet	1848	2	0.15 ms

Varying RSM Inference Types

1. **Collective Inference (CI):** Uses neighbor labels

2. **Relational Inference (RI):** Uses neighbor attributes

3. **Relational-Collective Inference (RCI):** Uses both neighbor labels and neighbor attributes

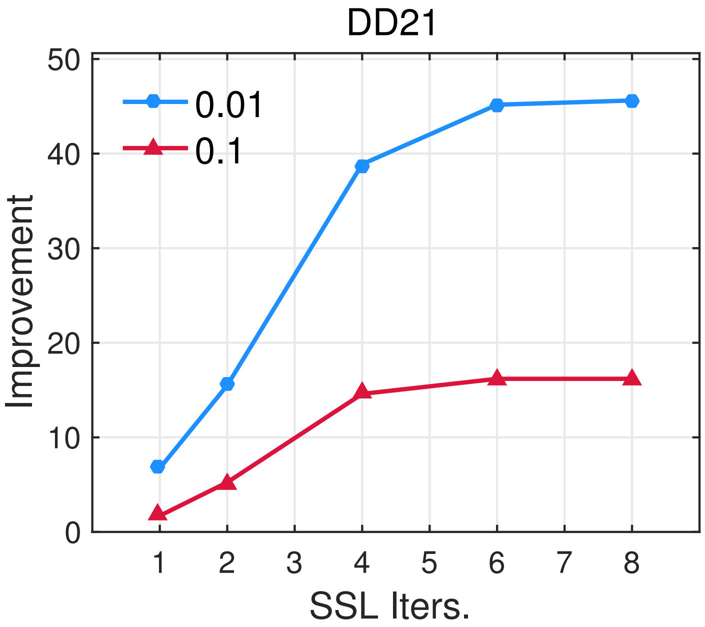


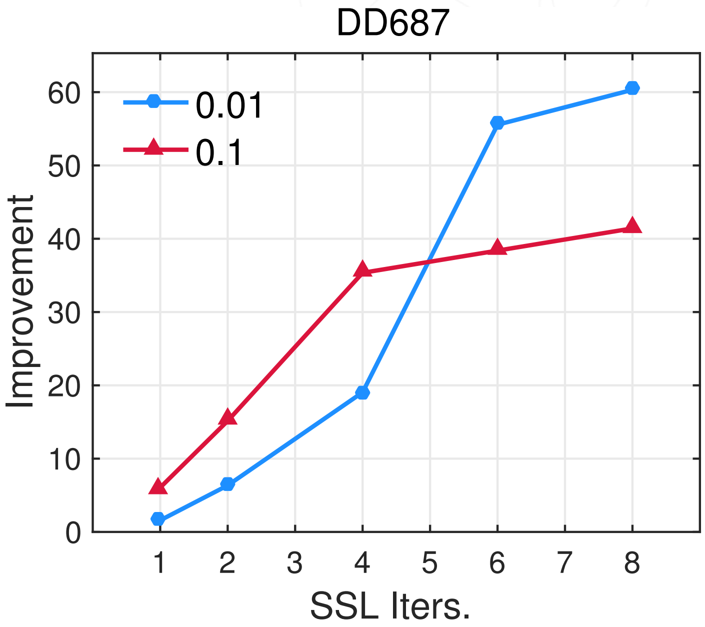
Impact of different classes of features/inference on classification (soc–TerrorRel.)

Accuracy (percent) improvement of RSM-CI, RSM-RI, and RSM-RCI over a basic variant of RSM.

Varying SSL iterations & known labels

- Improvement over label propagation as the # of SSL iterations increases
- More iterations appears to perform better when very few known labels





Effectiveness for Interactive RML

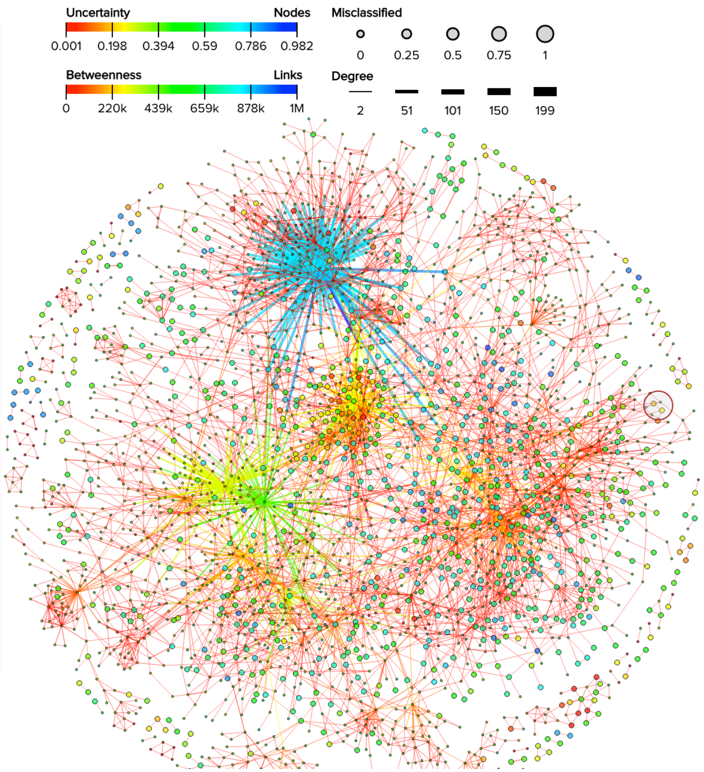
Methods often fail in practice due to low relational autocorrelation, noisy links, sparsely labeled graphs, and data representation

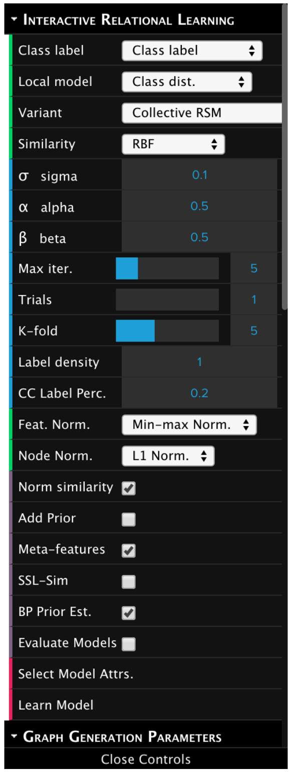
Class of RSM models are designed to be fast for interactive RML

cora

|V|	2708
|E|	5429
density	0.001
max degree	169
avg degree	4.01
|T|	5496
avg triangles	2.0
max triangles	170
local clustering	0.24
global clustering	0.10
assortativity	-0.0656
max k-core	5
max triangle-core	4
chromatic num.	7
max indep. set	1244
num communities	362
num roles	8
max betweenness	0.9M
diameter	19
mean distance	5.87
max pagerank	0.0121
wedges	49K
4-clique	245
4-chordal-cycle	2.7K
4-tailed-tri	57K
4-cycle	1.6K
3-star	1.1M
4-path	0.2M
4-triangle	4.6M
4-2star	0.1B
4-2edge	14M
4-1edge	20B
4-indep.	2.2T
label density	0.80
num classes	7
accuracy	85.01%

|V|	840
|E|	881
density	0.003
max degree	42
avg degree	2.10
|T|	577
avg triangles	0.7
max triangles	22
local clustering	0.16
global clustering	0.14
assortativity	-0.0658
max k-core	3
max triangle-core	0
chromatic num.	5
max indep. set	505
num communities	140
num roles	6
max betweenness	32K
diameter	19
mean distance	3.79
max pagerank	0.0142
wedges	3.5K
4-clique	16
4-chordal-cycle	184
4-cycle	90
4-path	9K
3-star	18K
4-tailed-tri	2.2K
4-1edge	0.3B
4-2edge	0.4M
4-2star	2.9M
4-triangle	0.1M
4-indep.	20B
label_density	1
num classes	7
accuracy	55.48





Users can **specify** the **RSM model** in real-time using a visual interface as well as perform evaluation, analyze errors, and make adjustments and refinements in a closed-loop.

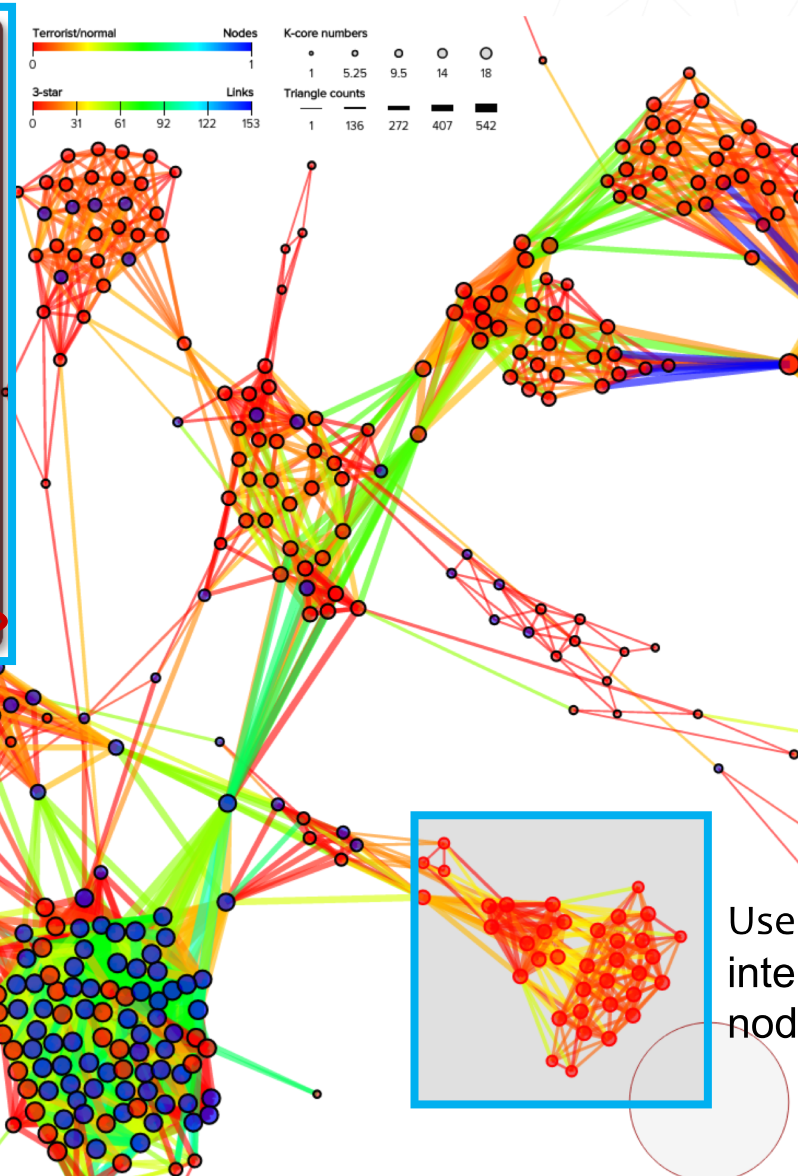
Users can search the space of RSM models in an interactive and visual fashion in real-time

Effectiveness for Interactive RML

soc-terror

|V|	593
|E|	4142
density	0.02
max degree	35
avg degree	13.97
|T|	38205
avg triangles	64.4
max triangles	300
local clustering	0.61
global clustering	0.57
assortativity	0.6904
max k-core	18
max triangle-core	17
chromatic num.	19
max indep. set	92
num communities	33
num roles	6
max betweenness	56K
diameter	15
mean distance	5.36
max pagerank	0.0026
wedges	29K
4-clique	29K
4-chordal-cycle	19K
4-tailed-tri	0.2M
4-cycle	3.9K
3-star	0
4-path	0.2M
4-triangle	7.1M
4-2star	16M
4-2edge	7.9M
4-1edge	0.7B
4-indep.	4.4B
label density	0.1
num classes	2
accuracy	85.34%

|V|	37
|E|	211
density	0.32
max degree	16
avg degree	11.41
|T|	1.4K
avg triangles	37.5
max triangles	66
local clustering	0.62
global clustering	0.57
assortativity	0.2190
max k-core	10
max triangle-core	5
chromatic num.	11
max indep. set	6
num communities	1
num roles	3
max betweenness	43
diameter	3
mean distance	1.75
max pagerank	0.0340
wedges	1K
4-clique	575
4-chordal-cycle	787
4-cycle	158
4-path	4.2K
3-star	0
4-tailed-tri	5.1K
4-1edge	22K
4-2edge	6.9K
4-2star	14K
4-triangle	6.7K
4-indep.	5.2K
label_density	0
[illegible]	2
accuracy	78.13



A local RSM model is learned in real-time from the selected subgraph(s); and accuracy and stats. are reported in the right side panel

Users can **select** subgraph(s) of interest by visually selecting nodes and edges.

Summary and Conclusion

- Introduced a general principled similarity-based relational learning framework
- Accurate for graphs with **arbitrary relational autocorrelation** (homophily *and* heterophily)
  - Previous methods perform well only when there is strong homophily
- Fast and scalable
  - Supports large-scale graphs, and applications requiring real-time performance such as interactive relational learning
- Flexible with many interchangeable components

Thanks!

Questions?

Data accessible online:

http://networkrepository.com

Adobe Research

Google

intel LABS

parc Palo Alto Research Center

xerox