Machine Learning with Graphs (MLG)

High-order Link Prediction

Can we predict combos of links?

Cheng-Te Li (李政德)

Institute of Data Science

National Cheng Kung University

chengte@mail.ncku.edu.tw

Canonical networks are everywhere slide, but with a hidden purpose!

Communications
nodes are people/accounts
edges show info. exchange

Drug compounds
nodes are substances
edge between substances that appear in the same drug

Collaboration
nodes are people/groups
edges link entities working together

Physical proximity
nodes are people/animals
edges link those that interact in close proximity

Real-world systems are composed of "**higher-order**" interactions that we often reduce to pairwise ones

Communications
nodes are people/accounts
emails often have several recipients, not just one

Drug compounds
nodes are substances
drugs are made up of several substances

Collaboration
nodes are people/groups
teams are made up of small groups

Physical proximity
nodes are people/animals
people often gather in small groups

High-order Link Prediction

time

Data.

$t_1 : \{1, 2, 3, 4\}$

$t_2 : \{1, 3, 5\}$

$t_3 : \{1, 6\}$

$t_4 : \{2, 6\}$

$t_5 : \{1, 7, 8\}$

$t_6 : \{3, 9\}$

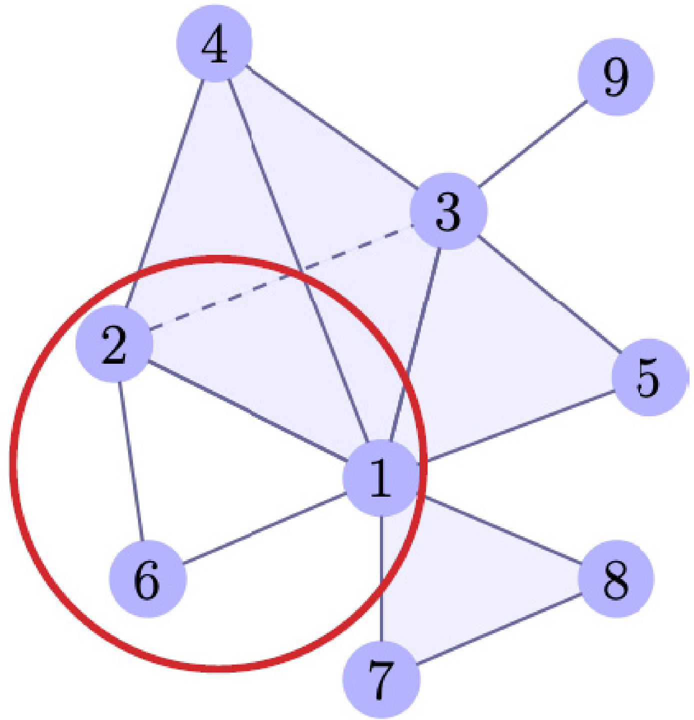
t

$t_7 : \{5, 8\}$

$t_8 : \{1, 2, 6\}$

- Observe **simplices** up to some time t
- Using this data, we want to predict **what groups of > 2 nodes will appear in a simplex** in the future

Such structure prediction cannot be considered in classical link prediction

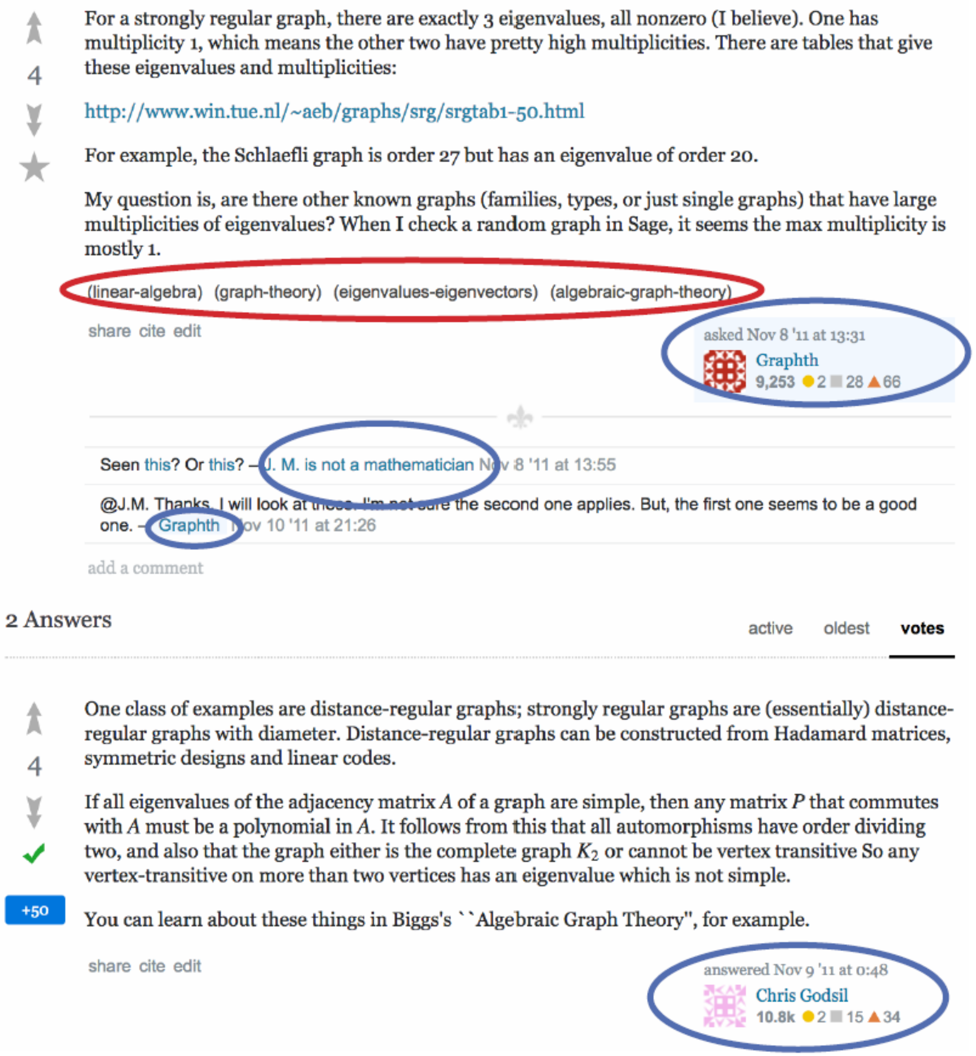


Potential applications

- Novel combinations of drugs for treatments.
- Group recommendation in social networks.
- Team formation

Datasets on High-order Interactions

1) Coauthorship in different domains
2) Emails with multiple recipients
3) Tags on Q&A forums
4) Threads on Q&A forums
5) Contact/proximity measurements
6) Musical artist collaboration
7) Substance makeup and classification codes applied to drugs the FDA examines
8) U.S. Congress committee memberships and bill sponsorship
9) Combinations of drugs seen in patients in ER visits



https://math.stackexchange.com/q/80181

How big are these datasets?

- Not large in terms of #(bytes) to store
- Large in terms of set structure complexity
- 1 size-5 simplex induces 31 sub-simplices

Dataset	# nodes	# timestamped simplices
coauth-DBLP	1.92M	3.70M
coauth-MAG-Geology	1.26M	1.59M
coauth-MAG-History	1.01M	1.81M
music-rap-genius	56.8K	225K
tags-stack-overflow	50.0K	14.5M
tags-ask-ubuntu	3.00K	271K
tags-math-sx	1.63K	822K
threads-stack-overflow	2.7M	11.3M
threads-math-sx	176K	720K
threads-ask-ubuntu	126K	193K
NDC-substances	5.31K	112K
NDC-classes	1.16K	49.7K
DAWN	2.56K	2.27M
congress-bills	1.72K	261K
congress-committees	863	679
email-Eu	998	235K
email-Enron	143	10.9K
contact-high-school	327	172K
contact-primary-school	242	107K

High-order data → Weighted Projected Graph

- Thinking of higher-order data as a **weighted projected graph** with “filled-in” structures

Data.

$t_1 : \{1, 2, 3, 4\}$

$t_2 : \{1, 3, 5\}$

$t_3 : \{1, 6\}$

$t_4 : \{2, 6\}$

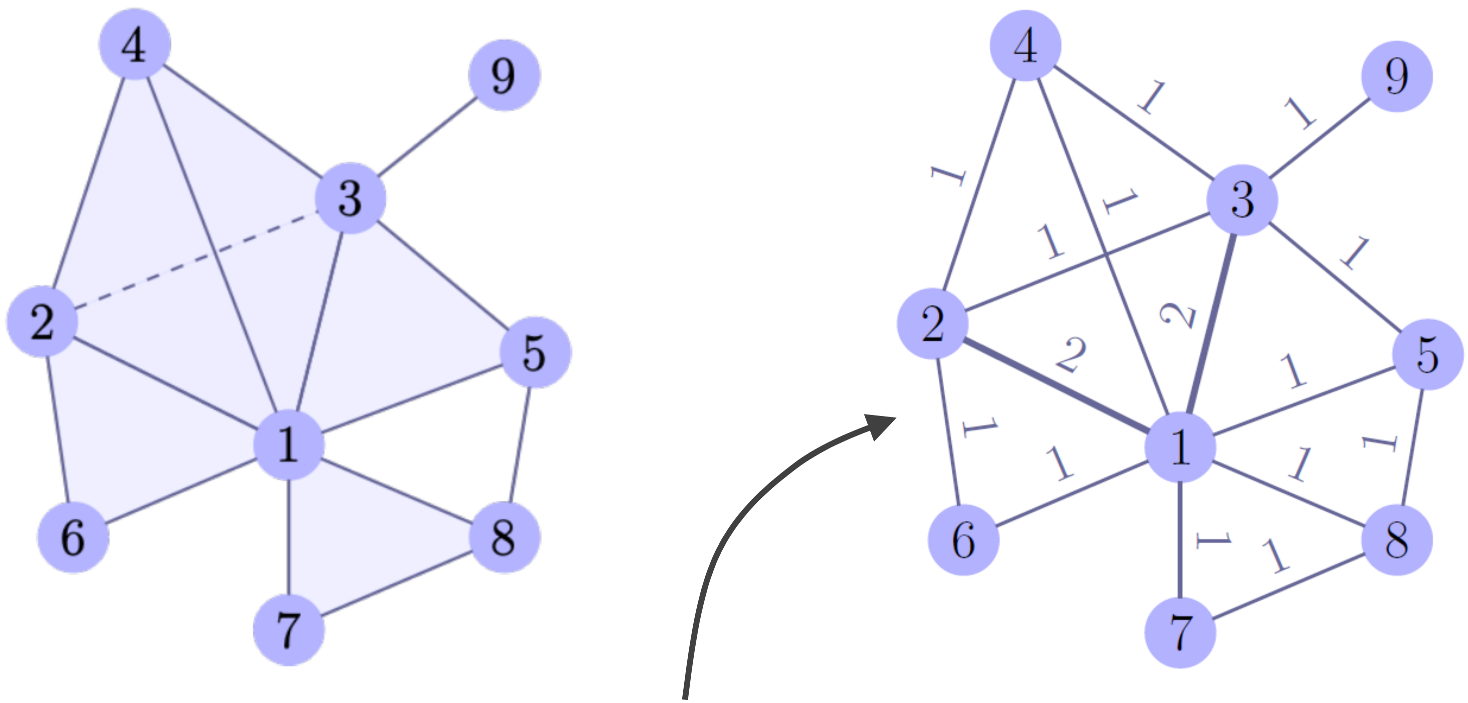
$t_5 : \{1, 7, 8\}$

$t_6 : \{3, 9\}$

$t_7 : \{5, 8\}$

$t_8 : \{1, 2, 6\}$

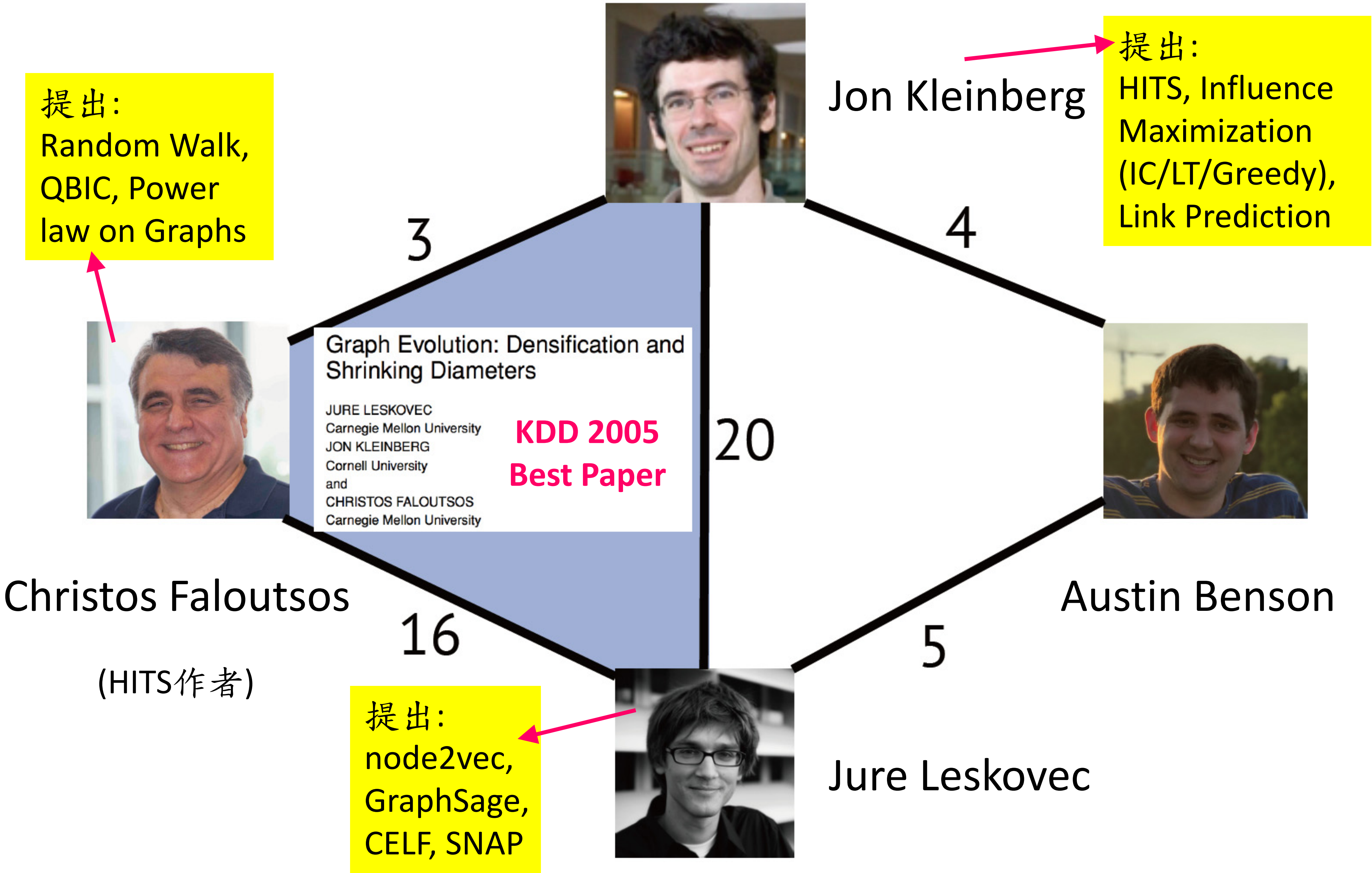
Pictures to have in mind.



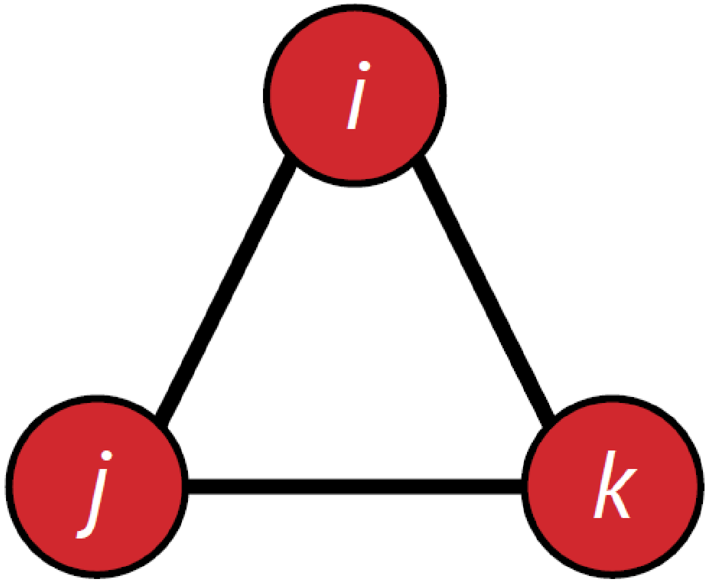
Projected graph $\boldsymbol{W}$

$\boldsymbol{W}_{ij}$: # of simplices containing nodes i and j

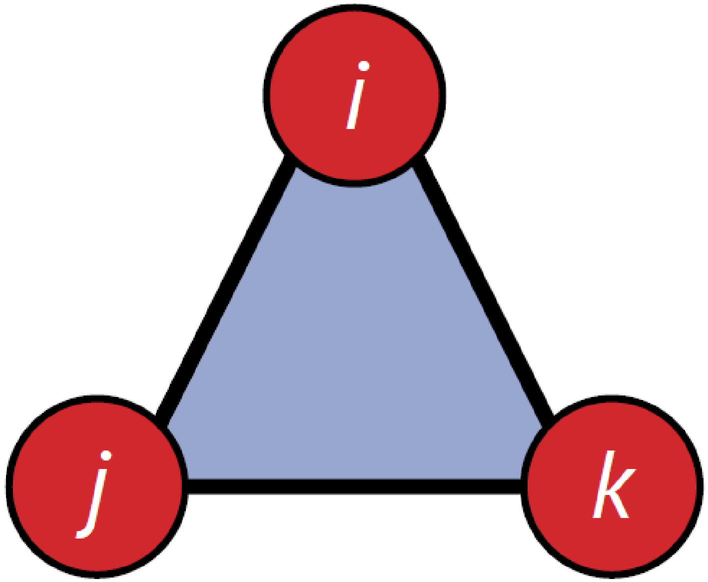
An Example on Projected Graph



“Open Triangle” vs. “Closed Triangle”



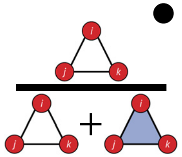
“Open Triangle”
each pair has been in a simplex together but **all 3 nodes have never been in the same simplex**

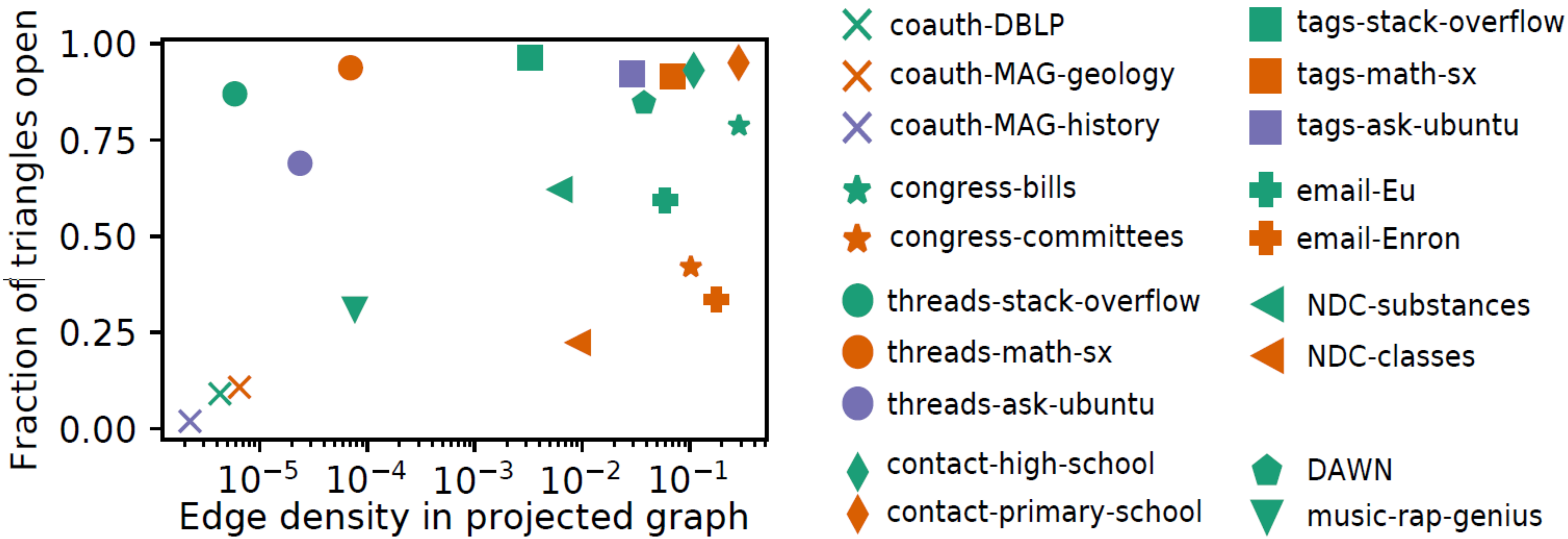


“Closed Triangle”
there is **some simplex that contains all 3 nodes**

Fraction of Open Triangles

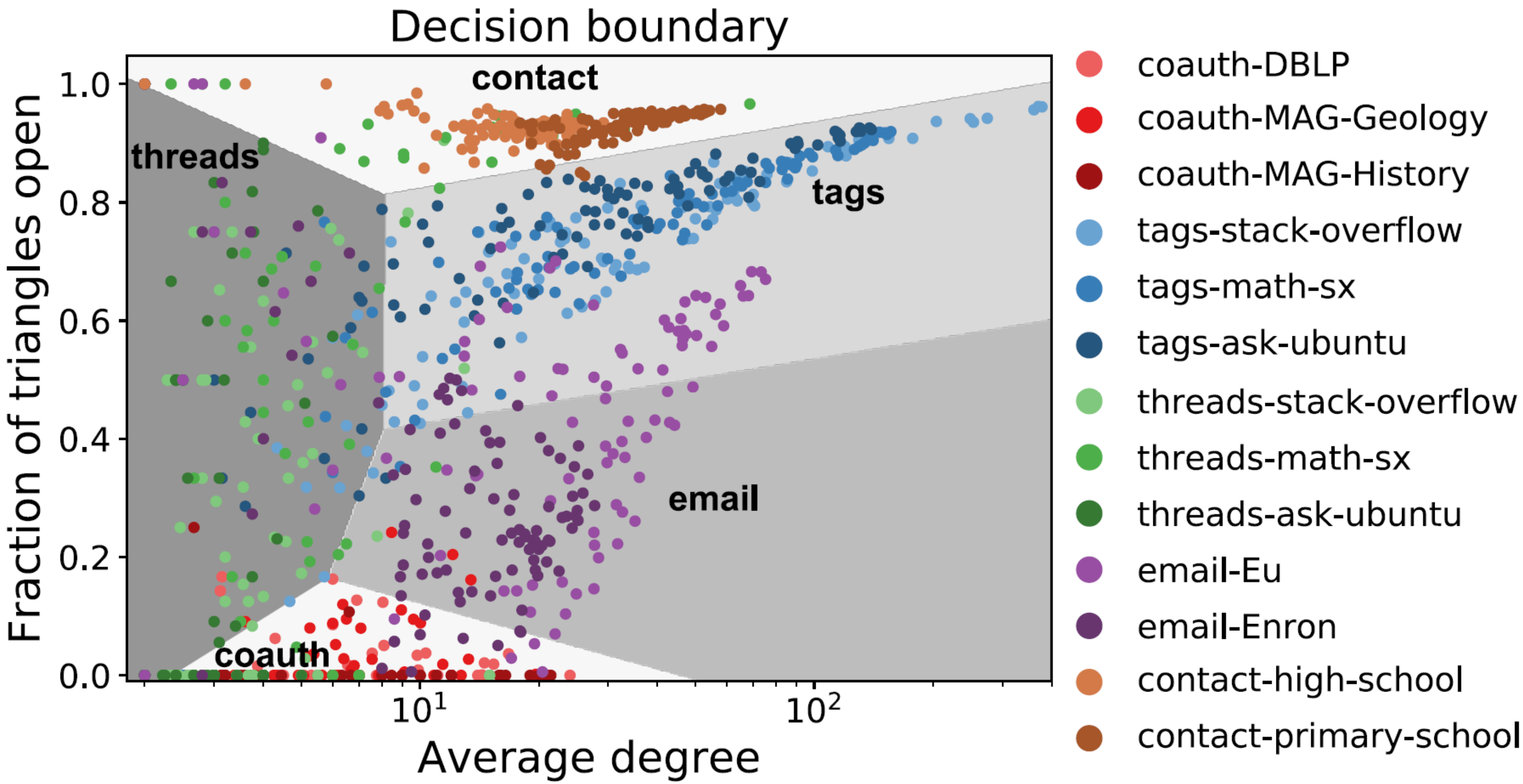
- Lots of variation in the
  fraction of open triangles vs. edge density
- Datasets from the same domain are similar





Edge density $=$ #edges / C_2^n

Domain separation also occurs at the **local level**



- Randomly sample 100 nodes per dataset and measure log of average degree and fraction of open triangles
- Logistic regression model to predict domain (co-authorship, tags, threads, email, contact)
- 75% model accuracy vs. 21% with random guessing

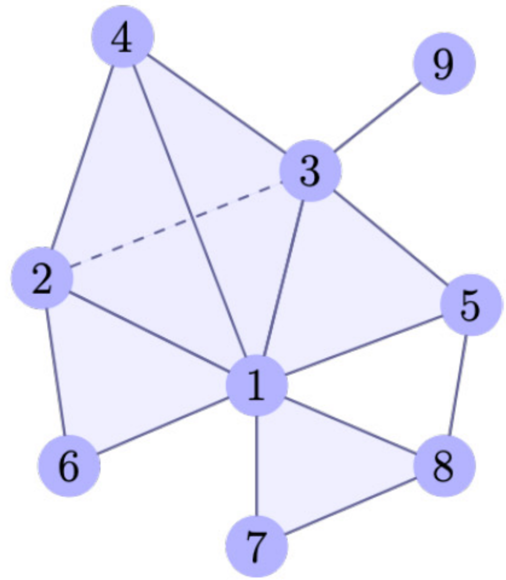
Simplical Closure

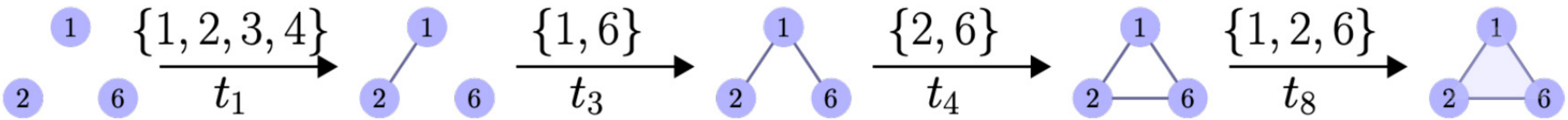
- Q1 What are the common ways in which new simplices appear?
- Q2 How do new closed triangles appear?

Ways that New Simplices Appear

Groups of nodes go through trajectories until finally reaching a "simplicial closure event"

$t_1 : \{1, 2, 3, 4\}$
$t_2 : \{1, 3, 5\}$
$t_3 : \{1, 6\}$
$t_4 : \{2, 6\}$
$t_5 : \{1, 7, 8\}$
$t_6 : \{3, 9\}$
$t_7 : \{5, 8\}$
$t_8 : \{1, 2, 6\}$

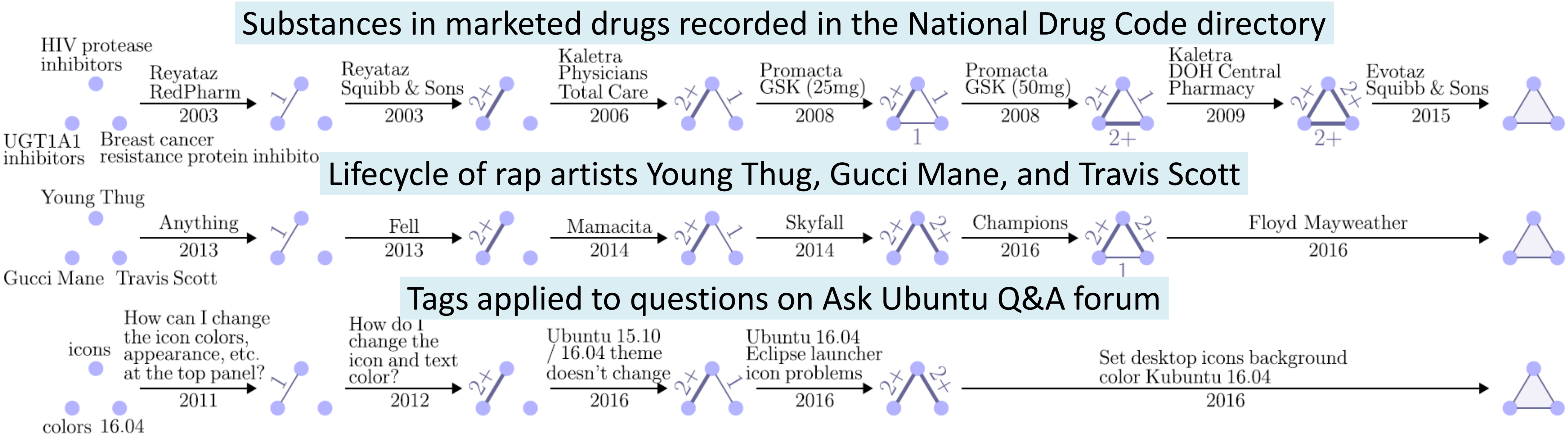




We focus on **simplicial closure on 3 nodes**

Ways that New Simplices Appear

Groups of nodes go through trajectories until finally reaching a "simplicial closure event"



Bin weighted edges into "weak" and "strong ties" in projected graph

$\boldsymbol{W_{ij}}$: # of simplices containing nodes i and j

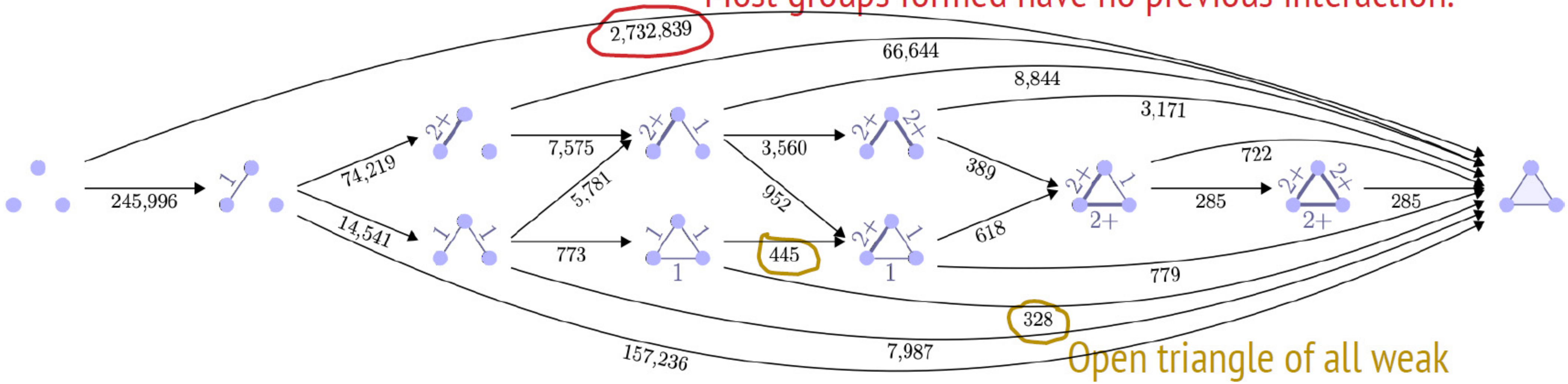
- **Weak ties**: $W_{ij} = 1$ (one simplex contains i and j)
- **Strong ties**: $W_{ij} \geq 2$ (at least two simplices contain i and j)

How do new closed triangles appear?

Analyzing the temporal dynamics in aggregate!

Coauthorship data of scholars publishing in history

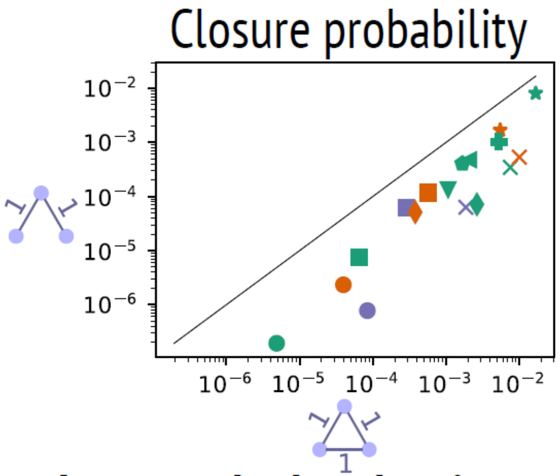
W_{ij} = # of simplices containing nodes i and j



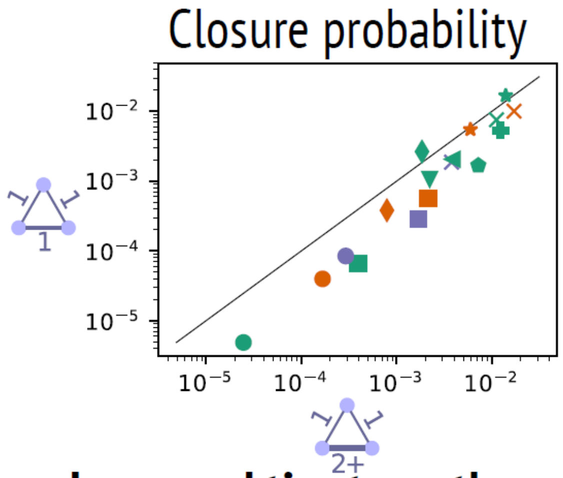
Open triangle of all weak ties more likely to form a strong tie before closing.

Simplicial closure depends on structure in projected graph

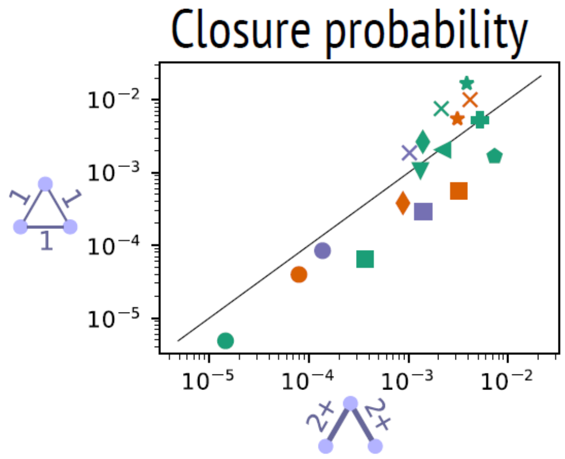
- First 80% of the data (in time) → record configurations of triplets not in closed triangle
- Remainder of data → find fraction that are now closed triangles



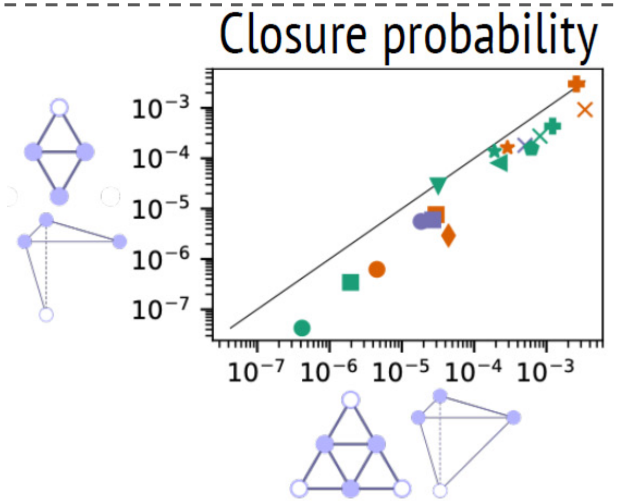
Increased edge density increases closure probability.



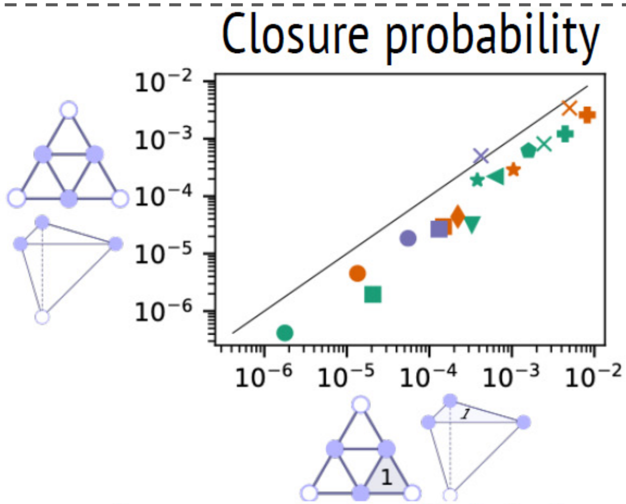
Increased tie strength increases closure probability.



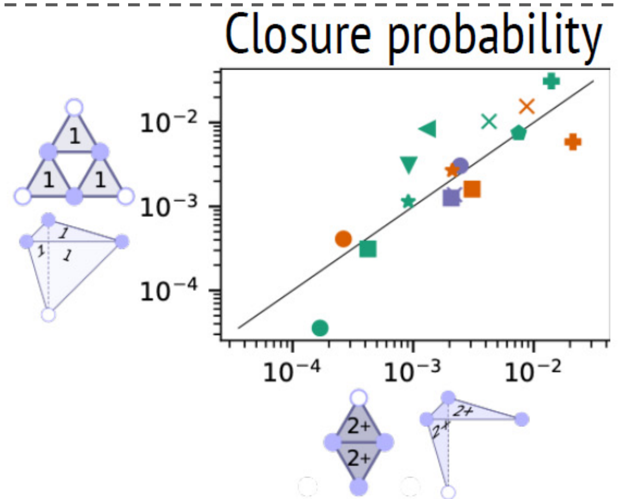
Tension between edge density and tie strength.



Increased edge density increases closure probability.



Increased simplicial tie strength increases closure probability.



Tension b/w edge density simplicial tie strength.

High-order Link Prediction

- Observe **simplices** up to some time *t*
- Using this data, the goal is to predict **what groups of > 2 nodes will appear in a simplex** in the future

time

Data.

$t_1 : \{1, 2, 3, 4\}$

$t_2 : \{1, 3, 5\}$

$t_3 : \{1, 6\}$

$t_4 : \{2, 6\}$

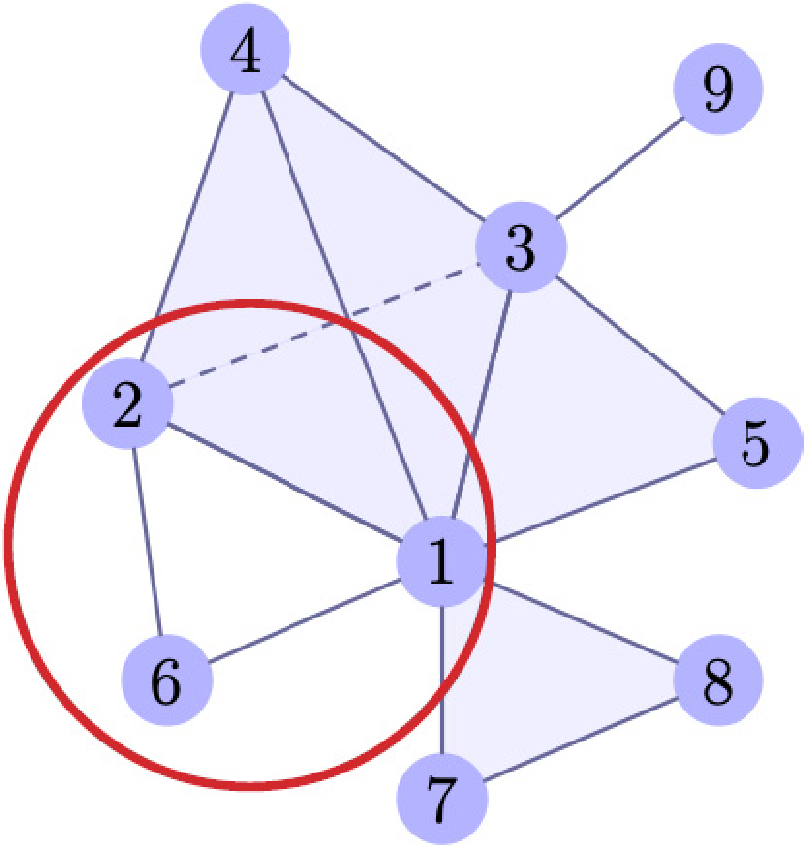
$t_5 : \{1, 7, 8\}$

$t_6 : \{3, 9\}$

---- t

$t_7 : \{5, 8\}$

$t_8 : \{1, 2, 6\}$

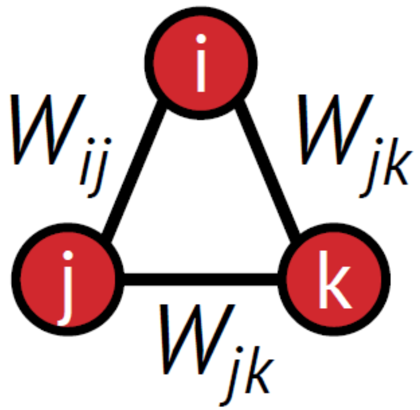


Insights from Structural Analysis

- Structural analysis tells us
  what we should be looking at for prediction

1) Edge density matters!

→ Focus our attention on predicting which open triangles become closed triangles (intelligently reduce search space)



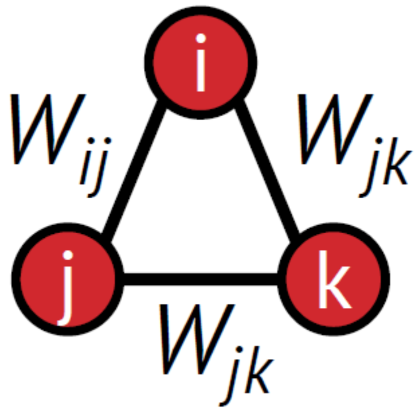
2) Tie strength matters!

→ Various ways of incorporating this information

Feature Engineering for **Open Triangles**

- Extracting features on first 80% of data
- Four classes of score functions $s(i, j, k)$

1) Functions of W_{ij}, W_{jk}, W_{ki}
   - arithmetic mean, geometric mean, etc.

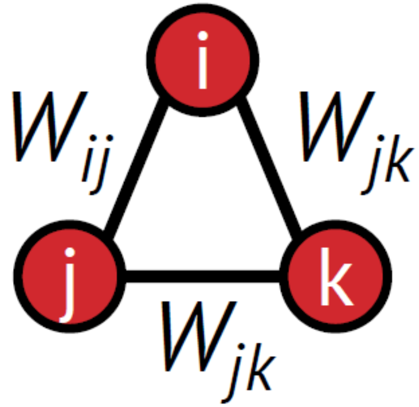


2) Look at common neighbors of the three nodes
   - Generalized Jaccard, Adamic-Adar, etc.
3) Use "whole-network" similarity scores on projected graph
   - Sum of PageRank or Katz scores amongst edges
4) Learn from data
   - Train a **logistic regression** model with features

- After computing scores, predict that open triangles with highest scores will be closed triangles in final 20% of data

(1) Functions of W_{ij}, W_{jk}, W_{ki}

1. Arithmetic mean
   $s(i,j,k) = (W_{ij} + W_{ik} + W_{jk})/3$



W_{ij} = # of simplices containing nodes i and j

2. Geometric mean
   $s(i,j,k) = (W_{ij} W_{ik} W_{jk})^{1/3}$

3. Harmonic mean
   $s(i,j,k) = 3/(W_{ij}^{-1} + W_{ik}^{-1} + W_{jk}^{-1})$

4. Generalized mean
   $s(i,j,k) = m_p(W_{ij}, W_{jk}, W_{ik}) = (W_{ij}^p + W_{jk}^p + W_{ik}^p)^{1/p}$

(2) $s(i, j, k)$ is a function neighbors

1. Number of common neighbors of all 3 nodes

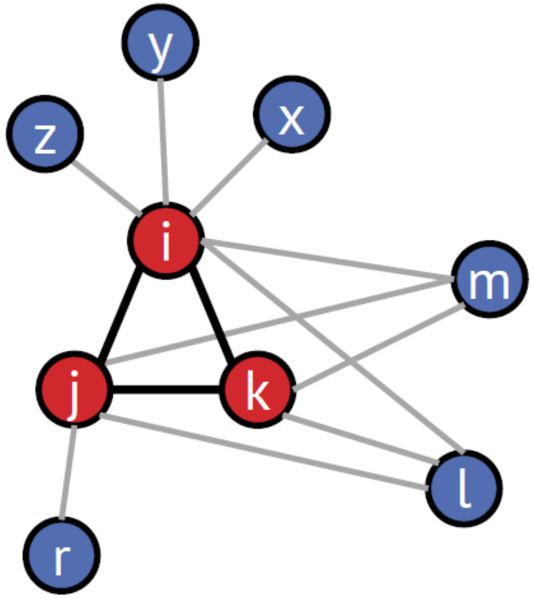
$$s(i,j,k) = |N(i) \cap N(j) \cap N(k)|$$

2. Generalized Jaccard coefficient

$$s(i,j,k) = \frac{|N(i) \cap N(j) \cap N(k)|}{|N(i) \cup N(j) \cup N(k)|}$$

3. Preferential attachment

$$s(i,j,k) = |N(i)| \cdot |N(j)| \cdot |N(k)|$$



$$N(i) = \{j, k, l, m, x, y, z\}$$

$$N(j) = \{i, k, l, m, r\}$$

$$N(k) = \{i, j, l, m\}$$

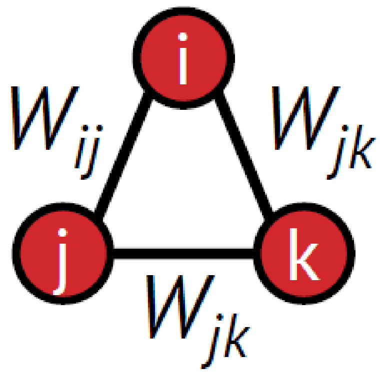
(3) $s(i,j,k)$ is built from "whole-network" similarity scores on edges

$$s(i,j,k) = S_{ij} + S_{ji} + S_{jk} + S_{kj} + S_{ik} + S_{ji}$$

1. PageRank (unweighted or weighted)

$$\boldsymbol{S} = (\boldsymbol{I} - \alpha \boldsymbol{W}\boldsymbol{D}_W^{-1})^{-1}$$

$$\boldsymbol{S} = (\boldsymbol{I} - \alpha \boldsymbol{A}\boldsymbol{D}_A^{-1})^{-1}$$



2. Katz (unweighted or weighted)

$$\boldsymbol{S} = (\boldsymbol{I} - \beta \boldsymbol{W})^{-1} - \boldsymbol{I}$$

$$\boldsymbol{S} = (\boldsymbol{I} - \beta \boldsymbol{A})^{-1} - \boldsymbol{I}$$

$\boldsymbol{A} = \min(\boldsymbol{W}, 1)$

$\boldsymbol{D}_W = \mathrm{diag}(\boldsymbol{W}\boldsymbol{1})$

$\boldsymbol{D}_A = \mathrm{diag}(\boldsymbol{A}\boldsymbol{1})$

(4) $s(i, j, k)$ is learned from data

1) Split data into training and validation sets
2) Compute features of (i, j, k) from previous ideas using training data
3) Throw features + validation labels into machine learning blender → learn model
4) Re-compute features on combined training + validation → apply model on the data

Performance Comparison

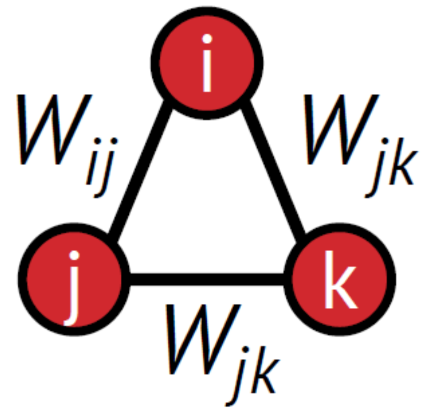
Table 3. Open triangle closure prediction performance based on eight models: harmonic, geometric, and arithmetic means of the three edge weights; three-way Adamic–Adar coefficient (A-A); preferential attachment (PA); Katz similarity; personalized PageRank similarity (PPR); and a feature-based supervised logistic regression model (Log. reg.)

random	Dataset	Harmonic mean	Geometric mean	Arithmetic mean	A-A	PA	Katz	PPR	Log. reg.
1.68e-03	coauth-DBLP	1.49	1.59	1.50	1.60	0.74	1.51	1.83	**3.37**
7.16e-04	coauth-MAG-history	1.69	2.72	3.20	5.82	2.49	3.40	1.88	**6.75**
3.35e-03	coauth-MAG-geology	2.01	1.97	1.69	2.71	0.97	1.74	1.26	**4.74**
6.82e-04	music-rap-genius	5.44	**6.92**	1.98	2.10	2.15	2.00	2.09	2.67
1.84e-04	tags-stack-overflow	**13.08**	10.42	3.97	6.63	2.74	3.60	1.85	3.37
1.08e-03	tags-math-sx	9.08	8.67	2.88	6.34	2.81	2.71	1.55	**13.99**
1.08e-03	tags-ask-ubuntu	12.29	**12.64**	4.24	7.51	5.63	4.15	2.54	7.48
1.14e-05	threads-stack-overflow	23.85	**31.12**	12.97	3.19	3.89	11.54	4.06	1.53
5.63e-05	threads-math-sx	20.86	16.01	5.03	23.32	7.46	4.86	1.18	**47.18**
1.31e-04	threads-ask-ubuntu	78.12	**80.94**	29.00	30.82	6.62	32.31	1.51	9.82
1.17e-03	NDC-substances	4.90	5.27	2.90	5.97	4.46	2.93	1.83	**8.17**
6.72e-03	NDC-classes	**4.43**	3.38	1.82	0.99	2.14	1.34	0.91	0.62
8.47e-03	DAWN	4.43	3.86	2.13	**4.77**	1.45	2.04	1.37	2.86
6.99e-04	congress-committees	3.59	3.28	2.48	5.04	1.31	2.59	3.89	**7.67**
1.71e-04	congress-bills	0.93	0.90	0.88	0.66	0.55	0.78	1.07	**107.19**
1.40e-02	email-Enron	1.78	1.62	1.33	0.87	0.83	1.28	**3.16**	0.72
5.34e-03	email-Eu	1.98	2.15	1.78	1.37	1.55	1.79	1.75	**3.47**
2.47e-03	contact-high-school	3.86	**4.16**	2.54	2.00	1.13	2.53	2.41	2.86
2.59e-03	contact-primary-school	5.63	6.40	3.96	3.21	0.94	4.02	4.31	**6.91**

Performance is AUC-PR relative to the random baseline, i.e., relative to the fraction of open triangles that close. The top performance number for each dataset is in boldface type.

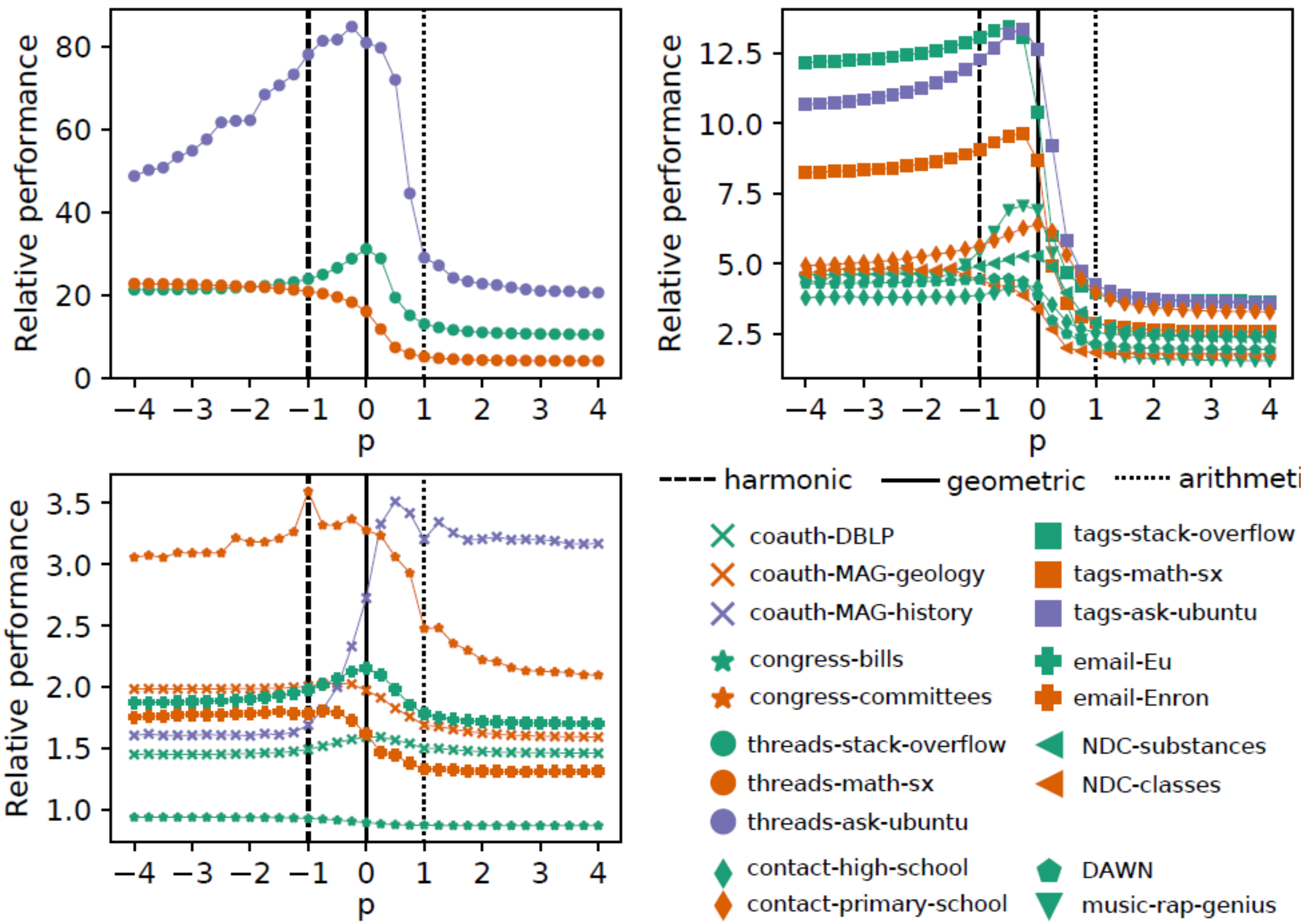
Lessons Learned

- Predicting pretty well on most datasets
  - 4x to 107x better than random in MAP
  - Note: only predicting on open triangles
- Thread co-participation and co-tagging on stack exchange are consistently easy to predict
- Simply averaging W_{ij}, W_{jk}, W_{ki} consistently performs well

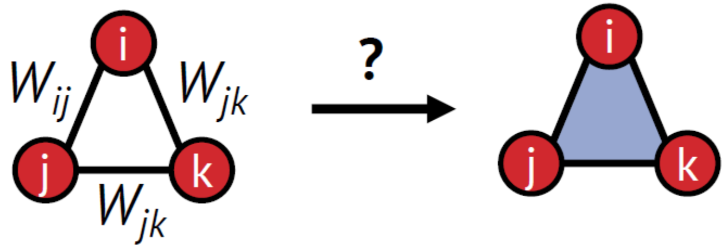


How about Generalized Mean?

- Generalized means of edges weights are often good predictors of new 3-node simplices appearing



$$\text{score}_p(i, j, k) = (W_{ij}^p + W_{jk}^p + W_{ik}^p)^{1/p}$$



Opportunities

[for your final project]

https://github.com/arbenson/ScHoLP-Tutorial

1) Higher-order data is pervasive!
   We have ways to represent data, and higher-order link prediction is a general framework for comparing models and methods
2) Develop fancy features/embeddings to out-perform our baselines
3) Why does generalized mean between harm. and geom. work well?
4) Computation is more challenging and complex for 4-node patterns