SCS CMU

Large Graph Mining: Patterns, Tools and Case Studies

Christos Faloutsos
Hanghang Tong
CMU

SCS CMU

Outline

- Part 1: Patterns
- Part 2: Matrix and Tensor Tools
- ➡ Part 3: Proximity
- Part 4: Case Studies

SCS CMU

Part 3: Proximity on Graphs

-Definitions, Fast Solutions, and Applications

SCS CMU

Joint work with

Christos Faloutsos (CMU)

Jia-Yu Pan (Google)

Yehuda Koren (AT&T Labs)

Spiros Papadimitriou

Philip S. Yu

Huiming Qu

Hani Jamjoom

IBM

Tina Eliassi-Rad (LLNL)

Brian Gallagher (LLNL)

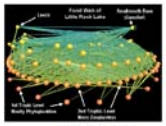
Kensuke Oonuma (Sony Corp.)

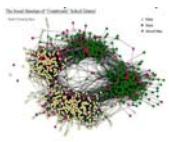
Yasushi Sakurai (NTT Labs)

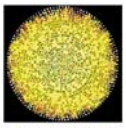
SCS CMU

Recap: Graphs are everywhere!

Internet Map [lumeta.com]

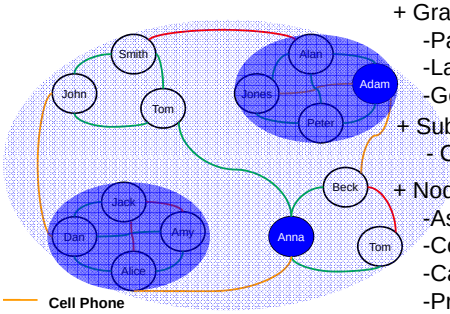

Food Web [Martinez '91]


Friendship Network [Moody '01]


Protein Interactions [genomebiology.com]

SCS CMU

Graph Mining: Big Picture

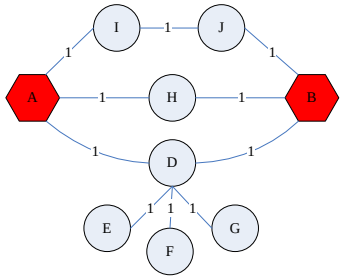


+ Graph Level
  -Patterns
  -Laws
  -Generators
+ Subgraph Level
  - Community
+ Node Level
  -Association
  -Correlation
  -Causality
  -Proximity

We are here!

SCS CMU

Proximity on Graph: What?



a.k.a Relevance, Closeness, 'Similarity'…

3-7

SCS CMU

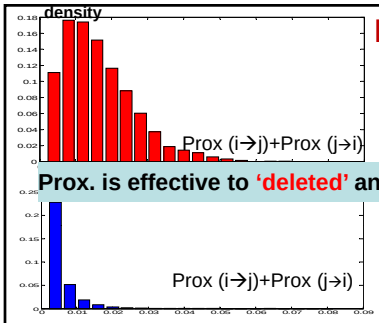
Proximity on Graphs: Why?

- Link prediction [Liben-Nowell+], [Tong+]
- Ranking [Haveliwala], [Chakrabarti+]
- Email Management [Minkov+]
- Image caption [Pan+]
- Neighborhooh Formulation [Sun+]
- Conn. subgraph [Faloutsos+], [Tong+], [Koren+]
- Pattern match [Tong+]
- Collaborative Filtering [Fouss+]
- Many more…

Will return to this later

 Copyright: Faloutsos, Tong (2008)

Link Prediction



Prox. Hist. for a set of deleted links

Prox. is effective to 'deleted' and absent edges!

Prox. Hist. for a set of absent links

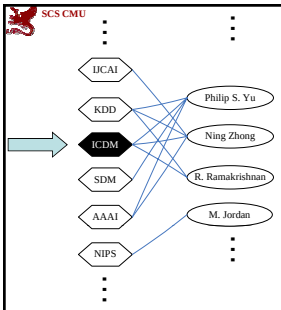
Q: How to predict the existence of the link?

A: Proximity! [Liben-Nowell + 2003]

 Copyright: Faloutsos, Tong (2008)

SCS CMU

Neighborhood Search on graphs



Conference Author

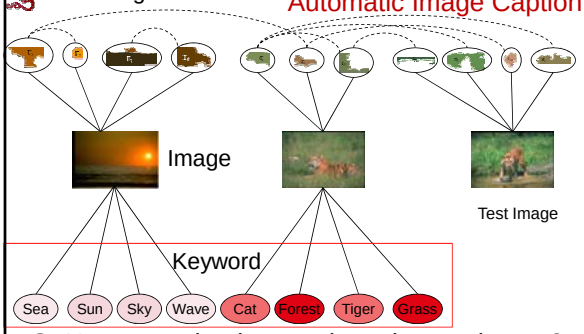
Q: what is most related conference to ICDM?

A: Proximity! [Sun+ ICDM2005]

 Copyright: Faloutsos, Tong (2008)

SCS CMU

Automatic Image Caption



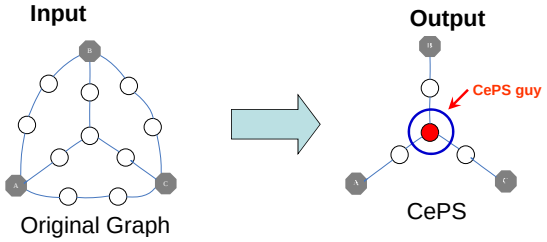
Q: How to assign keywords to the test image?

A: Proximity! [Pan+ 2004]

 Copyright: Faloutsos, Tong (2008)

SCS CMU

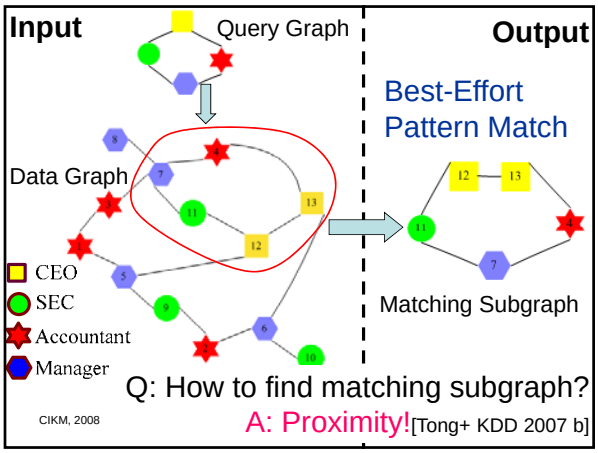
Center-Piece Subgraph(CePS)



Q: How to find hub for the black nodes?

A: Proximity! [Tong+ KDD 2006]

 Copyright: Faloutsos, Tong (2008)



SCS CMU

Outline: Part 3

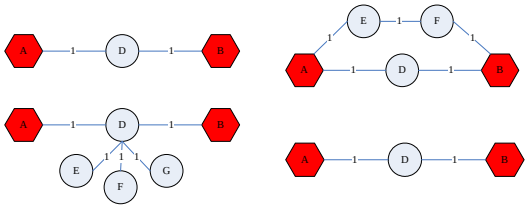
- Motivation
- Definitions
- Fast Solutions
- Applications
- Conclusion

- Basic: RWR
- Variants
- Properties
- Generalizations

SCS CMU

Some ``bad'' proximities

Why not shortest path?



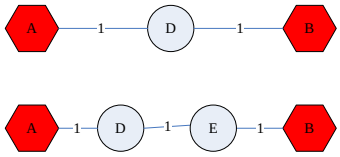
'pizza delivery guy' problem

'multi-facet' relationship

SCS CMU

Some ``bad'' proximities

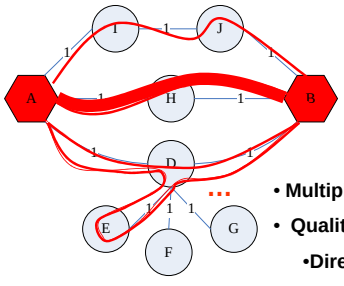
Why not max. netflow?



No punishment on long paths

SCS CMU

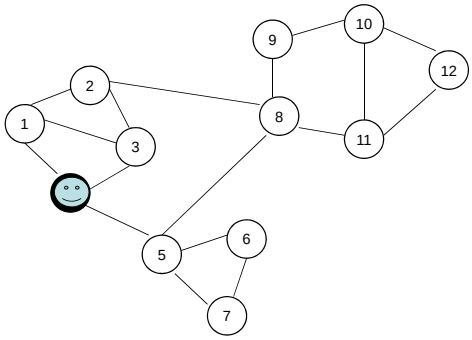
What is a ``good'' Proximity?



- **Multiple Connections**
- **Quality of connection**
  - **Direct & In-directed Conns**
  - **Length, Degree, Weight…**

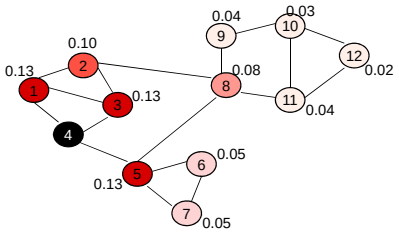
SCS CMU

Random walk with restart



SCS CMU

Random walk with restart



	Node 4
Node 1	0.13
Node 2	0.10
Node 3	0.13
Node 4	0.22
Node 5	0.13
Node 6	0.05
Node 7	0.05
Node 8	0.08
Node 9	0.04
Node 10	0.03
Node 11	0.04
Node 12	0.02

Nearby nodes, higher scores
More red, more relevant

Ranking vector $\vec{r}_4$

SCS CMU

Why RWR is a good score?

$$Q = (I - c\tilde{W})^{-1} = \; Q(i,j) \propto r_{i,j}$$

$\tilde{W}$: adjacency matrix.
c: damping factor

$$Q = c\,\tilde{W} + c^2\,\tilde{W}^2 + c^3\,\tilde{W}^3 + \cdots$$

all paths from i to j with length 1 | all paths from i to j with length 2 | all paths from i to j with length 3

SCS CMU

Outline: Part 3

- Motivation
- Definitions
- Fast Solutions
- Applications
- Conclusion

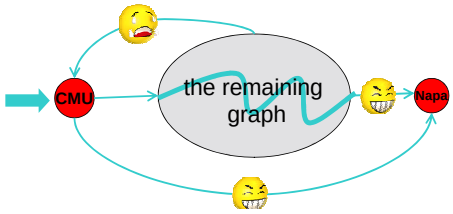
→ Variants

- Basic: RWR
- Properties
- Generalizations

SCS CMU

Variant: escape probability

- Define Random Walk (RW) on the graph
- Esc_Prob(CMU→Napa)
  - Prob (starting at CMU, reaches Napa before returning to CMU)



Esc_Prob = Pr (smile before cry)

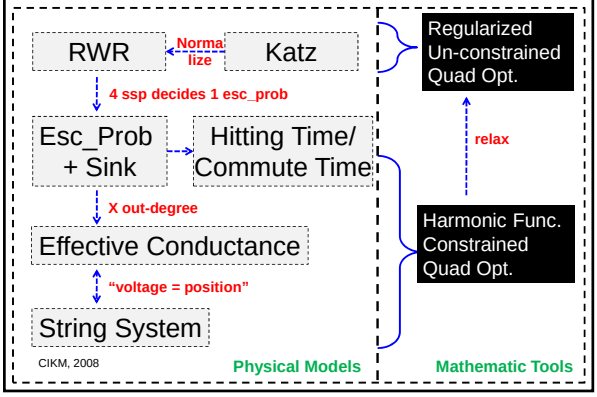
SCS CMU

Other Variants

- Other measure by RWs
  - Community Time/Hitting Time [Fouss+]
  - SimRank [Jeh+]
  - **All are "related to" or "similar to" random walk with restart!**
  - String Systems
- Katz [Katz], [Huang+], [Scholkopf+]
- Matrix-Forest-based Alg [Chobotarev+]

SCS CMU

Chaptering different measurements



SCS CMU

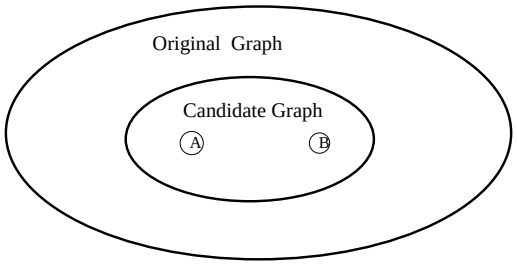
Outline: Part 3

- Motivation
- Definitions
- Fast Solutions
- Applications
- Conclusion

- Basic: RWR
- Variants
- Properties
- Generalizations

SCS CMU

Property: Monotonicity

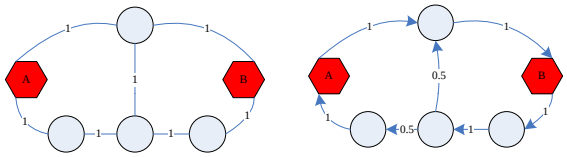


We want: $\text{Prox}^{candi}(a \to b) \leq \text{Prox}^{origianl}(a \to b)$

A: degree preserving! [Koren+ KDD06][Tong+ KDD07a][Tong+ SDM08]

SCS CMU

Property: Asymmetry [Tong+ KDD07 a]



What is Prox between A and B?

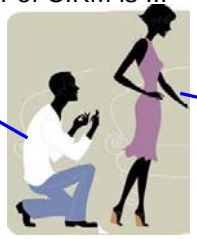
What is Prox from A to B?
What is Prox from B to A?

SCS CMU

Asymmetry in un-directed graphs

- Hanghang's # 1 Conf. is CIKM
- The #1 author of CIKM is ...

Hanghang



CIKM

SCS CMU

Outline: Part 3

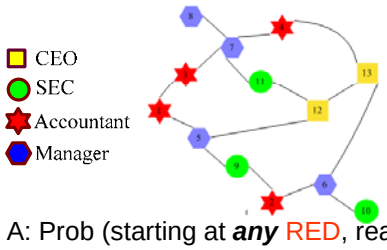
- Motivation
- Definitions
- Fast Solutions
- Applications
- Conclusion

- Basic: RWR
- Variants
- Properties
- Generalizations

SCS CMU

Group Proximity [Tong+ KDD07 a]

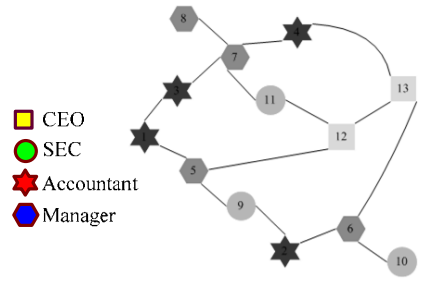
- Q: *How close are Accountants to SECs?*



- A: Prob (starting at ***any*** RED, reaches ***any*** GREEN before touching ***any*** RED again)

SCS CMU

Proximity on Attributed Graphs [Tong+ KDD07 b]

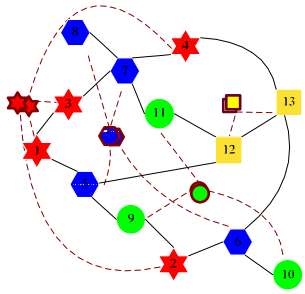


What is the proximity from node 7 to 10?

If we know that…

SCS CMU

A: Augmented graphs [Tong+ KDD07 b]



SCS CMU

More on Generalizations

- Attributed on edges [Chakrabarti+ KDD 06]
- Proximity w/ Time
  - [Minkov+], [Tong+ SDM 2008], [Tong+ CIKM 2008]
- Proximity w/ Side Information [Tong+ 2008]
- …

SCS CMU

Summary of Proximity Definitions

- Goal: Summarize multiple … relationship
- Solutions
  - Basic: Random Walk with Restart
    - [Pan+ 2004][Sun+ 2006][Tong+ 2006]
  - Properties: Asymmetry, monotonicity
    - [Koren+ 2006][Tong+ 2007] [Tong+ 2008]
  - Variants: Esc_Prob and many others.
    - [Faloutsos+ 2004] [Koren+ 2006][Tong+ 2007]
  - Generalizations: Group Prox, w/ Attr., w/ Time, w/ Side Information
    - [Charkrabarti+ 2006][Tong+ 2007] [Tong+ 2008]

SCS CMU

Outline: Part 3

- Motivation
- Definitions
- Fast Solutions
- Applications
- Conclusion

SCS CMU

Outline: Part 3

- Motivation
- Definitions
- Fast Solutions
  - B_Lin: RWR
  - BB_Lin: Skewed BGs
  - FastUpdate: Time-Evolving
- Applications
- Conclusion

SCS CMU

Preliminary: Sherman–Morrison Lemma

If:

$$\tilde{A} = A + u \times v^T$$

Then:

$$\tilde{A}^{-1} = (A + u \cdot v^T)^{-1} = A^{-1} - \frac{A^{-1} \cdot u \cdot v^T A^{-1}}{1 + v^T \cdot A^{-1} \cdot u}$$

SCS CMU

SM: The block-form

A	B
C	D

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} + A^{-1}B(D-CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D-CA^{-1}B)^{-1} \\ -(D-CA^{-1}B)^{-1}CA^{-1} & (D-CA^{-1}B)^{-1} \end{bmatrix}$$

Or…

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} (A-BD^{-1}C)^{-1} & -A^{-1}B(D-CA^{-1}B)^{-1} \\ -D^{-1}C(A-BD^{-1}C)^{-1} & (D-CA^{-1}B)^{-1} \end{bmatrix}$$

And many other variants…
Also known as Woodburg Identity

SCS CMU

SM Lemma: Applications

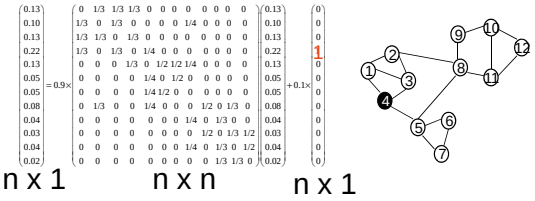
- RLS (Recursive least square)
  - and almost any algorithm in time series!
- Leave-one-out cross validation for LSR
- Kalman filtering
- Incremental matrix decomposition
- …
- … and all the fast sol.s we will introduce!

SCS CMU

Computing RWR

$$\vec{r}_i = c\tilde{W}\vec{r}_i + (1-c)\vec{e}_i$$

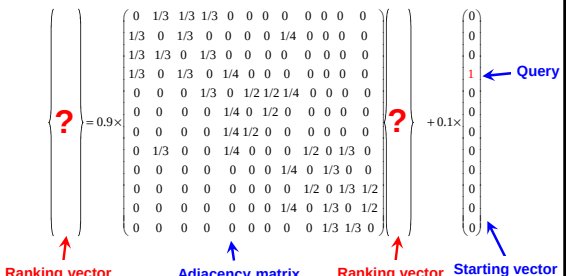
Ranking vector — Adjacency matrix — Restart p — Starting vector



n x 1 n x n n x 1

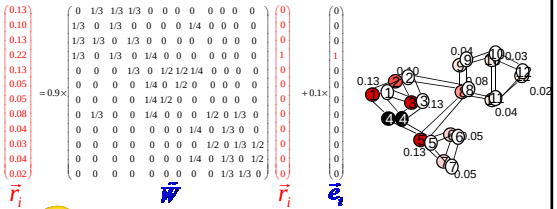
SCS CMU

Q: Given query *i*, how to solve it?



SCS CMU

OntheFly: $\vec{r}_i[t+1] = c\tilde{W}\vec{r}_i[t] + (1-c)\vec{e}_i$



No pre-computation/ light storage

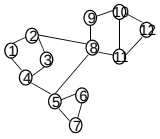
Slow on-line response O(mE)

SCS CMU

PreCompute

R:

	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}	r_{11}	r_{12}
	0.20	0.13	0.14	0.13	0.68	0.56	0.56	0.63	0.44	0.35	0.39	0.34
	0.28	0.20	0.13	0.96	0.64	0.53	0.53	0.85	0.60	0.48	0.53	0.45
	0.14	0.13	0.20	1.29	0.68	0.56	0.56	0.63	0.44	0.35	0.39	0.33
	0.13	0.10	0.13	2.06	0.95	0.78	0.78	0.61	0.43	0.34	0.38	0.32
	0.09	0.09	0.09	1.27	2.41	1.97	1.97	1.05	0.73	0.58	0.66	0.56
	0.03	0.04	0.04	0.52	0.98	2.06	1.37	0.43	0.30	0.24	0.27	0.22
	0.03	0.04	0.04	0.52	0.98	1.37	2.06	0.43	0.30	0.24	0.27	0.22
	0.08	0.11	0.04	0.82	1.05	0.86	0.86	2.13	1.49	1.19	1.33	1.13
	0.03	0.04	0.03	0.28	0.36	0.30	0.30	0.74	1.78	1.00	0.76	0.79
	0.04	0.04	0.04	0.34	0.44	0.36	0.36	0.89	1.50	2.45	1.54	1.80
	0.04	0.05	0.04	0.38	0.49	0.40	0.40	1.00	1.14	1.54	2.28	1.72
	0.02	0.03	0.02	0.21	0.28	0.22	0.22	0.56	0.79	1.20	1.14	2.05



$R = c \times Q$

$Q = (I - c\tilde{W})^{-1}$ [Haveliwala+ 2002]

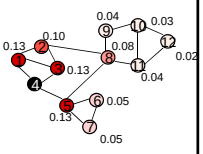
 Copyright: Faloutsos, Tong (2008)

SCS CMU

PreCompute: $Q = (I - c\tilde{W})^{-1}$

$Q = (I - c\tilde{W})^{-1}$

2.20	1.28	1.43	1.29	0.68	0.56	0.56	0.63	0.44	0.35	0.39	0.34
1.28	2.02	1.28	0.96	0.64	0.53	0.53	0.85	0.60	0.48	0.53	0.45
1.43	1.28	2.20	1.29	0.68	0.56	0.56	0.63	0.44	0.35	0.39	0.33
1.29	0.96	1.29	2.06	0.95	0.78	0.78	0.61	0.43	0.34	0.38	0.32
0.91	0.86	0.91	1.27	2.41	1.97	1.97	1.05	0.73	0.58	0.66	0.56
0.37	0.35	0.37	0.52	0.98	2.06	1.37	0.43	0.30	0.24	0.27	0.22
0.37	0.35	0.37	0.52	0.98	1.37	2.06	0.43	0.30	0.24	0.27	0.22
0.84	1.14	0.84	0.82	1.05	0.86	0.86	2.13	1.49	1.19	1.33	1.13
0.29	0.40	0.29	0.28	0.36	0.30	0.30	0.74	1.78	1.00	0.76	0.79
0.35	0.48	0.35	0.34	0.44	0.36	0.36	0.89	1.50	2.45	1.54	1.80
0.39	0.53	0.39	0.38	0.49	0.40	0.40	1.00	1.14	1.54	2.28	1.72
0.22	0.30	0.22	0.21	0.28	0.22	0.22	0.56	0.79	1.20	1.14	2.05



Fast on-line response

Heavy pre-computation/storage cost

$O(n^3)$ $O(n^2)$

SCS CMU

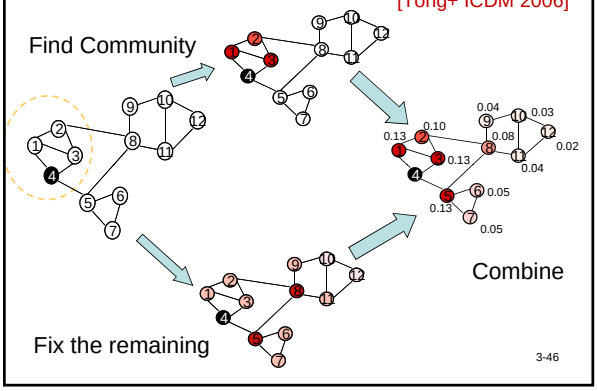
Q: How to Balance?

 Copyright: Faloutsos, Tong (2008)

SCS CMU

B_Lin: Basic Idea

[Tong+ ICDM 2006]



SCS CMU

B_Lin: details

details

$\tilde{W} \simeq$ + U S V

$\tilde{W}_1$: within community

Cross-community

 Copyright: Faloutsos, Tong (2008)

SCS CMU

B_Lin: details

details

$$\left(I - c\tilde{W}\right)^{-1} \approx \left[I - c\tilde{W}_1 - U\,S\,V\right]^{-1}$$

Easy to be inverted

LRA difference

SM Lemma!

 Copyright: Faloutsos, Tong (2008)

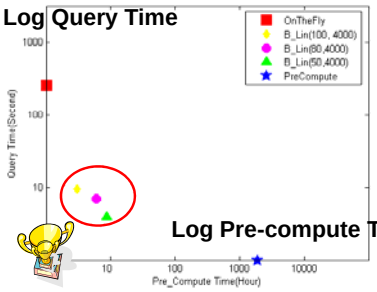
SCS CMU

B_Lin: summary

- Pre-Computational Stage
  - Q: Efficiently compute and store Q
  - A: A few small, instead of ONE BIG, matrices inversions
- On-Line Stage
  - Q: Efficiently recover one column of Q
  - A: A few, instead of MANY, matrix-vector multiplications

SCS CMU

Query Time vs. Pre-Compute Time



- **Quality: 90%+**
- **On-line:**
  - **Up to 150x speedup**
- **Pre-computation:**
  - **Two orders saving**

SCS CMU

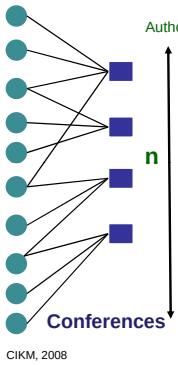
Outline: Part 3

- Motivation
- Definitions
- Fast Solutions
- Applications
- Conclusion

- B_Lin: RWR
- BB_Lin: Skewed BGs
- FastUpdate: Time-Evolving

SCS CMU

authors RWR on Bipartite Graph



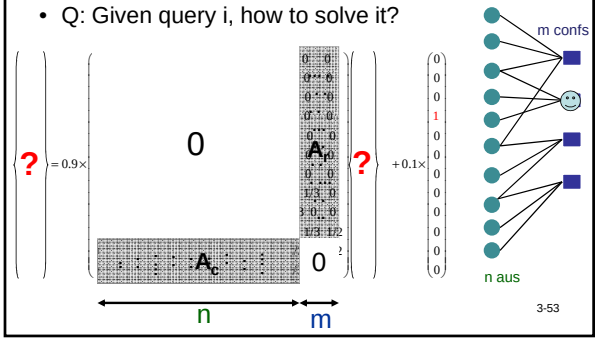
Observation: n >> m!

Examples:
1. DBLP: 400k aus, 3.5k confs
2. NetFlix: 2.7M usrs,18k mvs

SCS CMU

RWR on Skewed bipartite graphs

- Q: Given query i, how to solve it?



SCS CMU

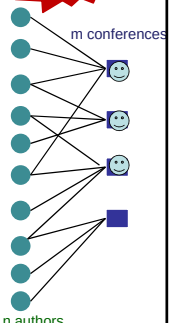
BB_Lin: Pre-Computation

2-step RWR for Conferences

[Tong+ ICDM 06]

details

- Step 1: $M = A_c \times A_r$



SCS CMU

BB_Lin: Pre-Computation [Tong+ ICDM 06]

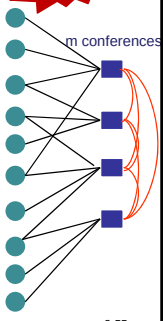
2-step RWR for Conferences

details

- Step 1: M = A_c x A_r

All Conf-Conf Prox. Scores

- Step 2: $\Lambda = (I - 0.9 \times M)^{-1}$



 Copyright: Faloutsos, Tong (2008)

SCS CMU

BB_Lin: Pre-Computation [Tong+ ICDM 06]

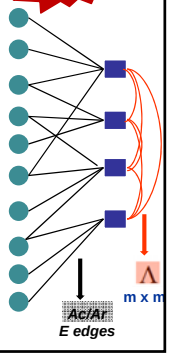
2-step RWR for Conferences

details

- Step 1: M = A_c x A_r

All Conf-Conf Prox. Scores

- Step 2: $\Lambda = (I - 0.9 \times M)^{-1}$
- Cost: $O(m^3 + m \cdot E)$
- Examples
  - NetFlix: 1.5hr for pre-computation;
  - DBLP: 1 few minutes



SCS CMU

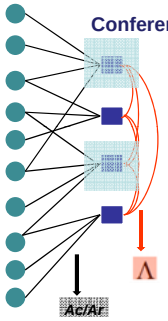
BB_Lin: On-Line Stage

authors

Conferences

(Base) Case 1:
- Conf - Conf

Read out !



SCS CMU

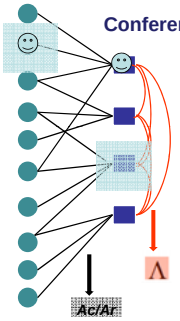
BB_Lin: On-Line Stage

authors

Conferences

Case 2:
- Au - Conf

1 matrix-vec!



SCS CMU

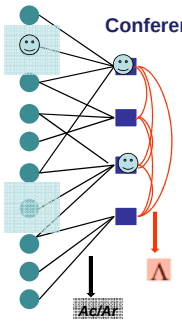
BB_Lin: On-Line Stage

authors

Conferences

Case 3:
- Au - Au

2 matrix-vec!



SCS CMU

BB_Lin: Examples

Dataset	Off-Line Cost	On-Line Cost
DBLP	a few minutes	frac. of sec.
NetFlix	1.5 hours	<0.01 sec.

400k authors x 3.5k conf.s

2.7m user x 18k movies

 Copyright: Faloutsos, Tong (2008)

SCS CMU

Outline: Part 3

- Motivation
- Definitions
- Fast Solutions
  - B_Lin: RWR
  - BB_Lin: Skewed BGs
  - FastUpdate: Time-Evolving
- Applications
- Conclusion

SCS CMU

Challenges

- BB_Lin is good for skewed bipartite graphs
  - for NetFlix (2.7M nodes and 100M edges)
  - **On-line cost** for query: fraction of seconds
    - w/ 1.5 hr pre-computation for m x m core matrix
- But…what if the graph is ***evolving*** over time
  - New edges/nodes arrive; edge weights increase…
  - **On-line cost:** 1.5hr itself becomes a part of this!

SCS CMU

Q: How to update the core matrix?

t=0 $\Lambda = (I - 0.9 \times M)^{-1}$ $O(m^3 + m \cdot E)$

t=1 $\tilde{\Lambda} = (I - 0.9 \times \tilde{M})^{-1}$ $O(m^3 + m \cdot E)$?

SCS CMU

details

Update the core matrix Λ

- Step 1:

$\tilde{M}$ = A_c x A_r = M + x

Rank 2 update

- Step 2:

$\tilde{\Lambda} = (I - 0.9 \times \tilde{M})^{-1}$

= Λ + x

$O(m^2)$ $O(m^3 + m \cdot E)$

SCS CMU

details

Update Λ: General Case

$\tilde{M}$ = A_c x A_r

n authors

m Conferences

- E' edges changed
- Involves n' authors, m' confs.

SCS CMU

details

Update Λ: General Case

- Observation: $\min(n', m') \ll E'$
  - the rank of update is small!
  - Real Example (DBLP Post)
    - 1258 time steps
    - E' up to ~20,000!
    - $\min(n', m')$ <=132
- Our Algorithm

$O(\min(n', m')m^2 + E')$

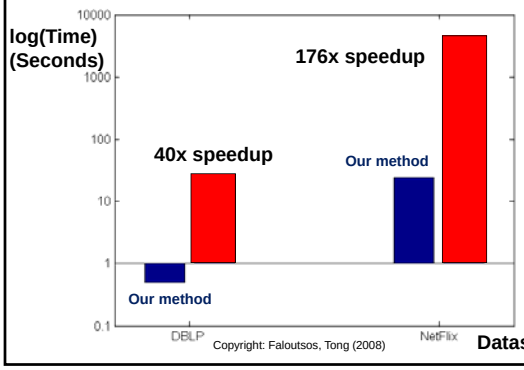
$O(m^3 + m \cdot E)$

n authors

m Conferences

SCS CMU

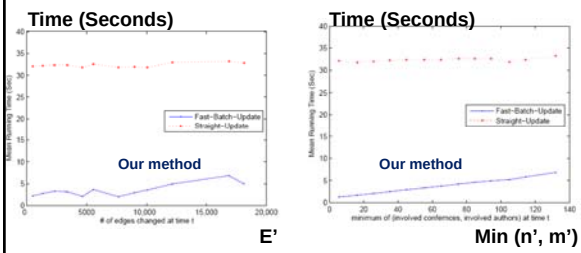
Fast-Single-Update



Copyright: Faloutsos, Tong (2008)

SCS CMU

Fast-Batch-Update



15x speed-up on average!

 Copyright: Faloutsos, Tong (2008)

SCS CMU

More on "Fast Solutions"

- FastAllDAP
  - Simultaneously solve multiple linear systems
  - [Tong+ KDD 2007 a]
- MT3
  - Multiple-Resolution Analysis on Time [Tong+ CIKM 2008]
- Fast-ProSIN
  - On-Line response for users' feedback [Tong+ ICDM 2008]

 Copyright: Faloutsos, Tong (2008)

SCS CMU

Summary of Fast Solutions

- Goal: Efficiently Solve Linear System(s)
- Sols.
  - B_Lin: one large linear system [Tong+ ICDM06]
  - BB_Lin: the intrinsic complexity is small [Tong+ ICDM06]
  - FastUpdate: dynamic linear system [Tong+ SDM08]
  - FastAllDAP: multiple linear systems [Tong+ KDD07 a]
  - MT3: [Tong+ CIKM 2008]
  - Fast-ProSIN: [Tong+ 2008]

 Copyright: Faloutsos, Tong (2008)

SCS CMU

Outline: Part 3

- Motivation
- Definitions
- Fast Solutions
- Applications
- Conclusion

 Copyright: Faloutsos, Tong (2008)

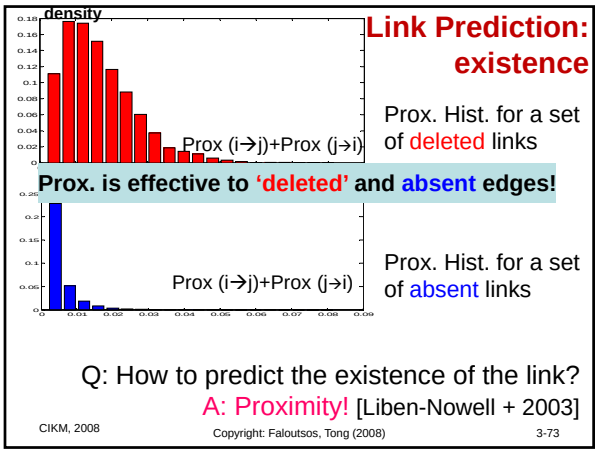
SCS CMU

Outline: Part 3

- Motivation
- Definitions
- Fast Solutions
- Applications
- Conclusion

→ Link Prediction & +
- Ranking Related Tasks
- User Specific Patterns
- Time Related Tasks

 Copyright: Faloutsos, Tong (2008)



Link Prediction: existence

Prox. Hist. for a set of deleted links

Prox. is effective to 'deleted' and absent edges!

Prox. Hist. for a set of absent links

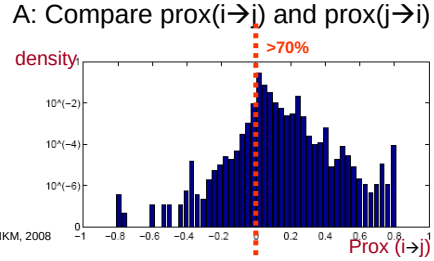
Q: How to predict the existence of the link?

A: Proximity! [Liben-Nowell + 2003]

SCS CMU

Link Prediction: direction [Tong+ KDD 07 a]

- Q: Given the existence of the link, what is the direction of the link?
- A: Compare prox(i→j) and prox(j→i)



SCS CMU

Beyond Link Prediction

- Collaborative Filtering [Fouss+]
- Name Disambiguation
  - [Minkov+ SIGIR 06]
- Anomaly Nodes/Edges
  - 'a' is abnormal if the neighborhood of 'a' is so different
  - [Sun+ ICDM 2005]

SCS CMU

Outline: Part 3

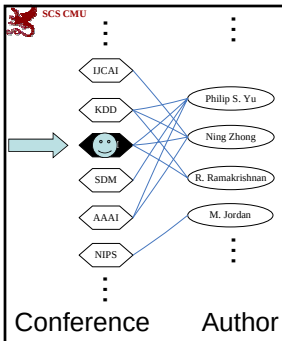
- Motivation
- Definitions
- Fast Solutions
- Applications
- Conclusion

→ Ranking Related Tasks

- Link Prediction & +
- User Specific Patterns
- Time Related Tasks

SCS CMU

Neighborhood Search on graphs



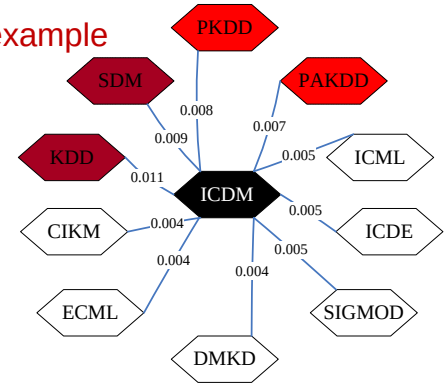
Conference Author

Q: what is most related conference to ICDM?

A: Proximity! [Sun+ ICDM2005]

SCS CMU

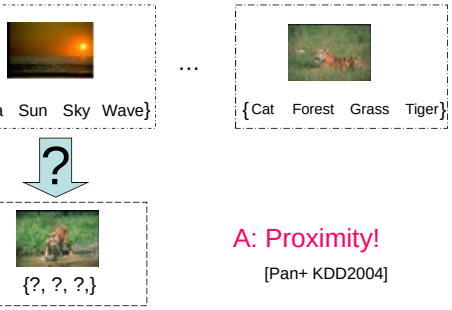
NF: example



SCS CMU

gCaP: Automatic Image Caption

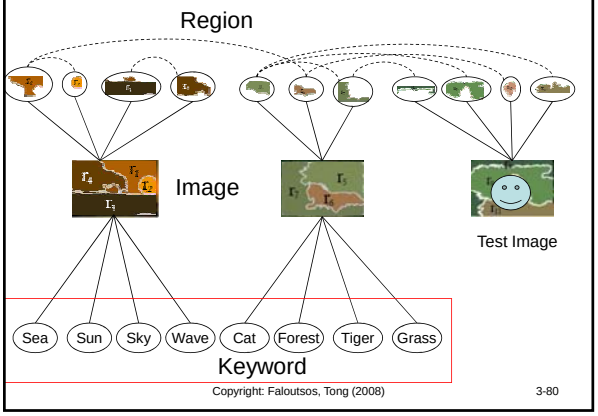
- Q



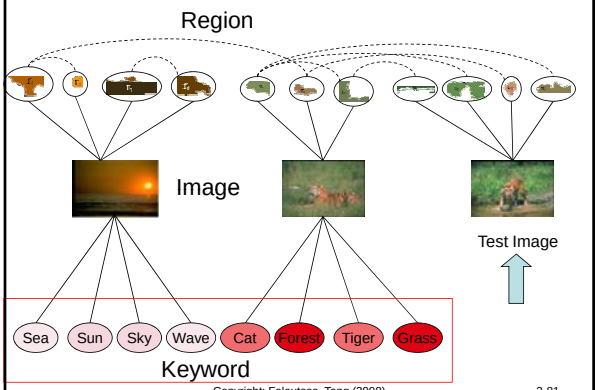
A: Proximity!

[Pan+ KDD2004]

SCS CMU

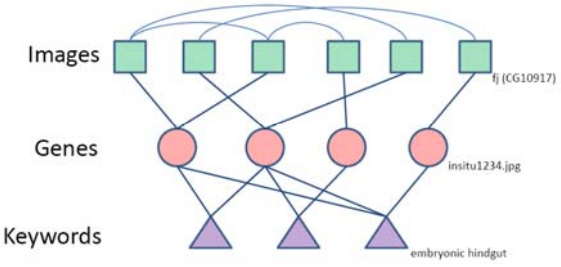


SCS CMU

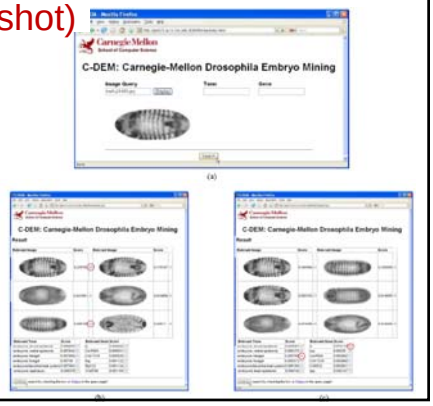


SCS CMU

C-DEM: Multi-Modal Query System for Drosophila Embryo Databases [Fan+ VLDB 2008]



C-DEM (Screen-shot)



SCS CMU

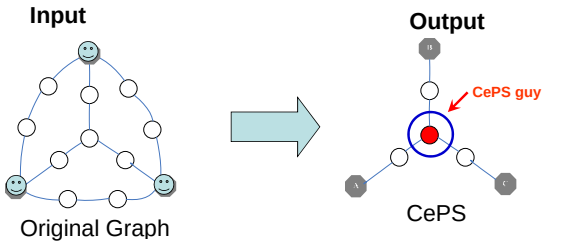
Outline: Part 3

- Motivation
- Definitions
- Fast Solutions
- Applications
- Conclusion

- Link Prediction & +
- Ranking Related Tasks
- User Specific Patterns
- Time Related Tasks

SCS CMU

Center-Piece Subgraph(CePS)



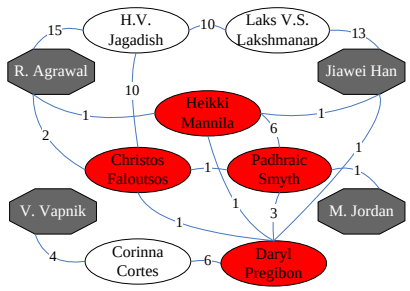
Q: How to find hub for the black nodes?

A: Proximity! [Tong+ KDD 2006]

Red: **Max** (Prox(A, Red) x Prox(B, Red) x Prox(C, Red))

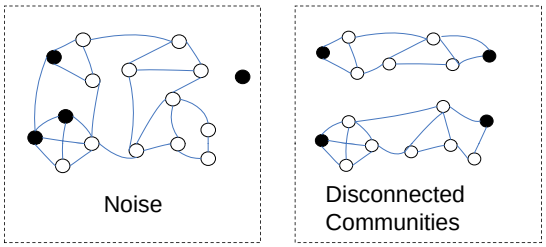
SCS CMU

CePS: Example



SCS CMU

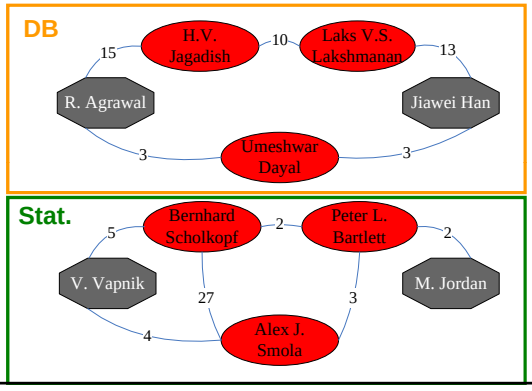
K_SoftAnd: Relaxation of AND



Asking AND query? ➔ No Answer!

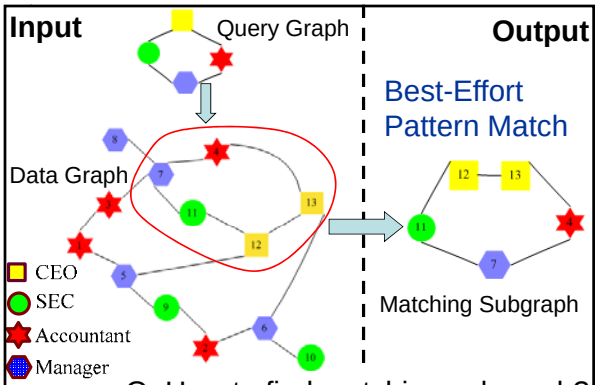
SCS CMU

CePS: 2 SoftAND



SCS CMU

Input / Output



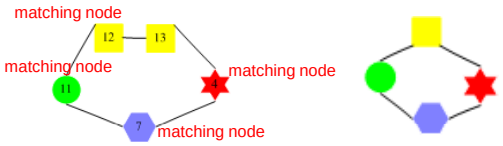
Q: How to find matching subgraph?

A: Proximity! [Tong+ KDD 2007 b]

SCS CMU

G-Ray: How to?

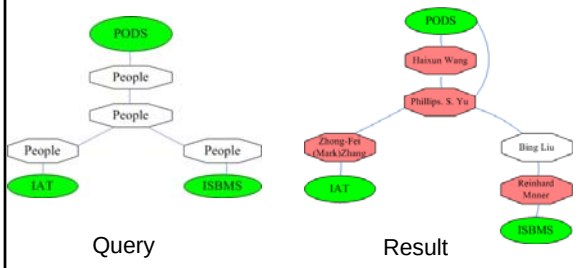
details



Goodness = Prox (12, 4) x Prox (4, 12) x
Prox (7, 4) x Prox (4, 7) x
Prox (11, 7) x Prox (7, 11) x
Prox (12, 11) x Prox (11, 12)

SCS CMU

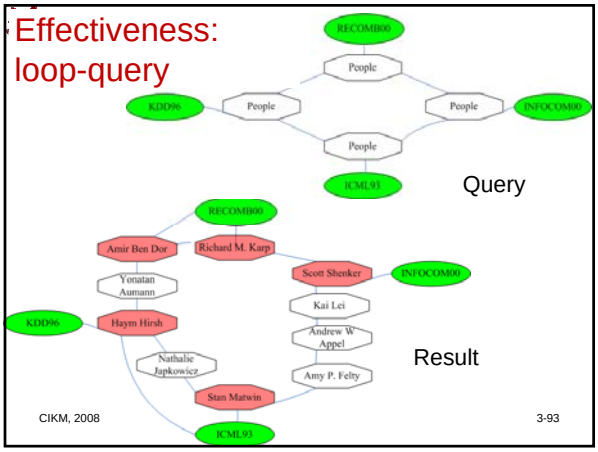
Effectiveness: star-query



SCS CMU

Effectiveness: line-query

Effectiveness: loop-query



SCS CMU

Outline: Part 3

- Motivation
- Definitions
- Fast Solutions
- Applications
- Conclusion

- Link Prediction & +
- Ranking Related Tasks
- User Specific Patterns
- Time Related Tasks

SCS CMU

Challenge

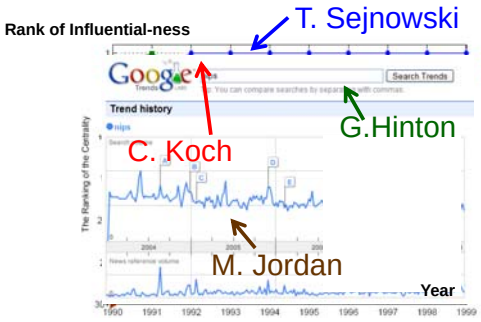
- Graphs are evolving over time!
  - –New nodes/edges show up;
  - –Existing nodes/edges die out;
  - –Edge weights change…

Q: How to generalize everything?

A: Track Proximity! [Tong+ SDM 2008]

SCS CMU

pTrack/cTrack: Trend analysis on graph level



SCS CMU

pTrack: Problem Definition

- **[Given]**
  - (1) a large, skewed time-evolving bipartite graphs,
  - (2) the query nodes of interest
- **[Track]**
  - (1) top-k most related nodes for each query node at each time step t;
  - (2) the proximity score (or rank of proximity) between any two query nodes at each time step t

SCS CMU

pTrack: Philip S. Yu's Top-5 conferences up to each year

ICDE ICDCS SIGMETRICS PDIS VLDB	CIKM ICDCS ICDE SIGMETRICS ICMCS	KDD SIGMOD ICDM CIKM ICDCS	ICDM KDD ICDE SDM VLDB
1992	1997	2002	2007

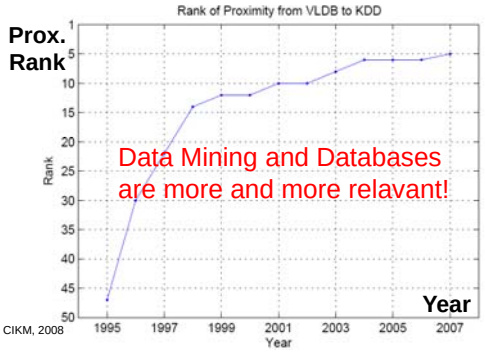
Databases
Performance
Distributed Sys.

DBLP: (Au. x Conf.)
- 400k aus,
- 3.5k confs
- 20 yrs

Databases
Data Mining

SCS CMU

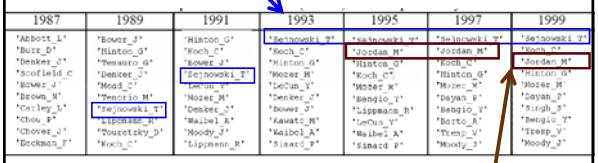
KDD's Rank wrt. VLDB over years



SCS CMU

cTrack:10 most influential authors in NIPS community up to each year

T. Sejnowski



M. Jordan

Author-paper bipartite graph from NIPS 1987-1999.
3k. 1740 papers, 2037 authors,
spreading over 13 years

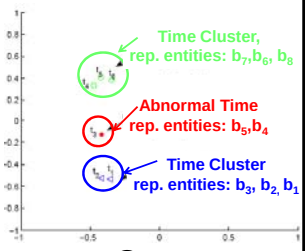
SCS CMU

T3: Understand Time in Complex Context

[Tong+ CIKM 2008]

Time	Event	Entity
t_1	e_1	b_1, b_2
	e_2	b_2, b_3
t_2	e_3	b_2, b_3
	e_4	b_3, b_4
t_3	e_5	b_4, b_5
t_4	e_6	b_5, b_6
	e_7	b_6, b_7
t_5	e_8	b_6, b_7
t_6	e_9	b_7, b_8

Input

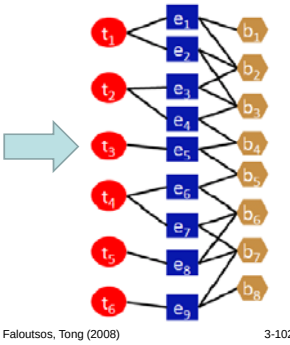


Output

SCS CMU

T3: Time-to-Time Proximity Matrix

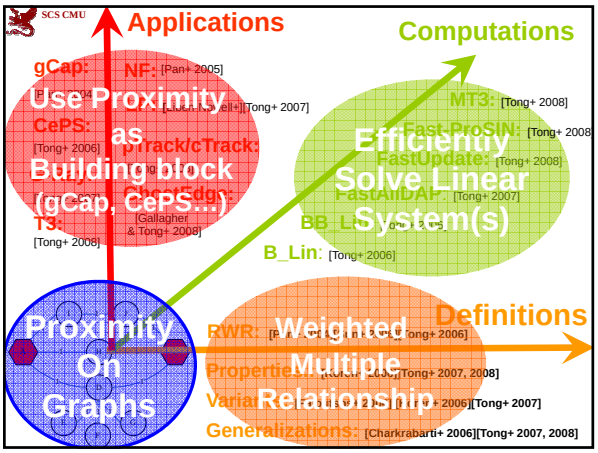
Time	Event	Entity
t_1	e_1	b_1, b_2
	e_2	b_2, b_3
t_2	e_3	b_2, b_3
	e_4	b_3, b_4
t_3	e_5	b_4, b_5
t_4	e_6	b_5, b_6
	e_7	b_6, b_7
t_5	e_8	b_6, b_7
t_6	e_9	b_7, b_8



SCS CMU

More Applications

- Clustering
  - Proximity as input [Ding+ KDD 2007]
- Email management [Minkov+ CEAS 06].
- Business Process Management [Qu+ 2008]
- ProSIN
  - Listen to clients' comments [Tong+ 2008]
- Ghost Edge
  - Within Network Classification [Gallagher & Tong+ KDD08 b]
- …



SCS CMU

Take-home messages

- Proximity Definitions
  - RWR $\vec{r}_i = c\tilde{W}\vec{r}_i + (1-c)\vec{e}_i$
  - and a lot of variants
- Computations
  - Sherman–Morrison Lemma
  - Fast Incremental Computation
- Applications
  - Proximity as a building block

SCS CMU

References

- L. Page, S. Brin, R. Motwani, & T. Winograd. (1998), The PageRank Citation Ranking: Bringing Order to the Web, Technical report, Stanford Library.
- T.H. Haveliwala. (2002) Topic-Sensitive PageRank. In WWW, 517-526, 2002
- J.Y. Pan, H.J. Yang, C. Faloutsos & P. Duygulu. (2004) Automatic multimedia cross-modal correlation discovery. In KDD, 653-658, 2004.
- C. Faloutsos, K. S. McCurley & A. Tomkins. (2002) Fast discovery of connection subgraphs. In KDD, 118-127, 2004.
- J. Sun, H. Qu, D. Chakrabarti & C. Faloutsos. (2005) Neighborhood Formation and Anomaly Detection in Bipartite Graphs. In ICDM, 418-425, 2005.
- W. Cohen. (2007) Graph Walks and Graphical Models. Draft.

SCS CMU

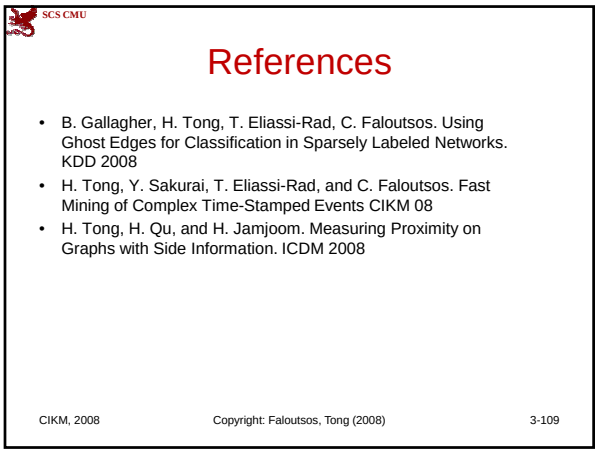
References

- P. Doyle & J. Snell. (1984) Random walks and electric networks, volume 22. Mathematical Association America, New York.
- Y. Koren, S. C. North, and C. Volinsky. (2006) Measuring and extracting proximity in networks. In KDD, 245–255, 2006.
- A. Agarwal, S. Chakrabarti & S. Aggarwal. (2006) Learning to rank networked entities. In KDD, 14-23, 2006.
- S. Chakrabarti. (2007) Dynamic personalized pagerank in entity-relation graphs. In WWW, 571-580, 2007.
- F. Fouss, A. Pirotte, J.-M. Renders, & M. Saerens. (2007) Random-Walk Computation of Similarities between Nodes of a Graph with Application to Collaborative Recommendation. IEEE Trans. Knowl. Data Eng. 19(3), 355-369 2007.

SCS CMU

References

- H. Tong & C. Faloutsos. (2006) Center-piece subgraphs: problem definition and fast solutions. In KDD, 404-413, 2006.
- H. Tong, C. Faloutsos, & J.Y. Pan. (2006) Fast Random Walk with Restart and Its Applications. In ICDM, 613-622, 2006.
- H. Tong, Y. Koren, & C. Faloutsos. (2007) Fast direction-aware proximity for graph mining. In KDD, 747-756, 2007.
- H. Tong, B. Gallagher, C. Faloutsos, & T. Eliassi-Rad. (2007) Fast best-effort pattern matching in large attributed graphs. In KDD, 737-746, 2007.
- H. Tong, S. Papadimitriou, P.S. Yu & C. Faloutsos. (2008) Proximity Tracking on Time-Evolving Bipartite Graphs. to appear in SDM 2008.


SCS CMU
References
• B. Gallagher, H. Tong, T. Eliassi-Rad, C. Faloutsos. Using Ghost Edges for Classification in Sparsely Labeled Networks. KDD 2008
• H. Tong, Y. Sakurai, T. Eliassi-Rad, and C. Faloutsos. Fast Mining of Complex Time-Stamped Events CIKM 08
• H. Tong, H. Qu, and H. Jamjoom. Measuring Proximity on Graphs with Side Information. ICDM 2008
CIKM, 2008
Copyright: Faloutsos, Tong (2008)
3-109