Large Graph Mining: Power Tools and a Practitioner's guide

Task 3: Recommendations & proximity

Faloutsos, Miller & Tsourakakis

CMU

Outline

- Introduction – Motivation
- Task 1: Node importance
- Task 2: Community detection
- **Task 3: Recommendations**
- Task 4: Connection sub-graphs
- Task 5: Mining graphs over time
- Task 6: Virus/influence propagation
- Task 7: Spectral graph theory
- Task 8: Tera/peta graph mining: hadoop
- Observations – patterns of real graphs
- Conclusions

Acknowledgement:

Most of the foils in ‘Task 3’ are by

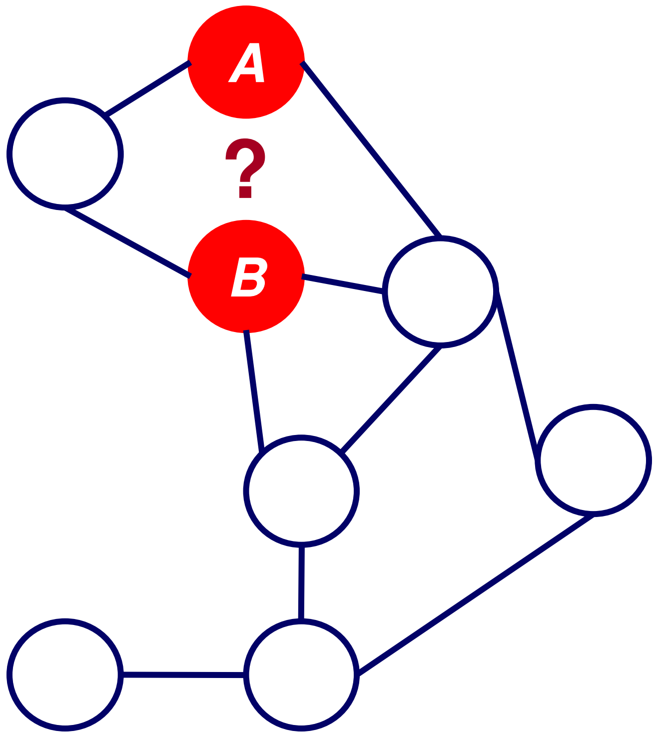
Hanghang TONG

www.cs.cmu.edu/~htong

Detailed outline

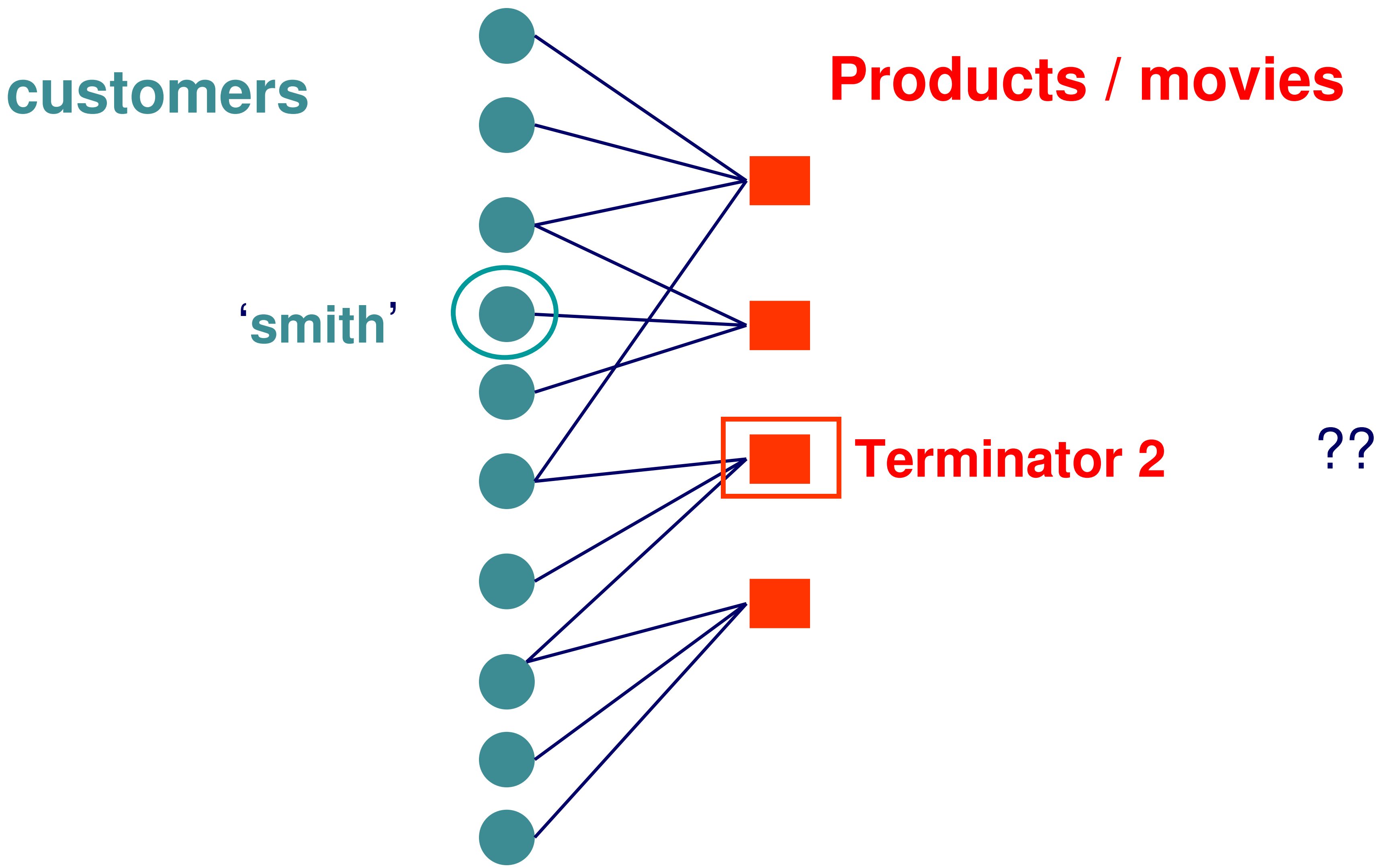
- Problem dfn and motivation
- Solution: Random walk with restarts
- Efficient computation
- Case study: image auto-captioning
- Extensions: bi-partite graphs; tracking
- Conclusions

Motivation: Link Prediction



Should we introduce
Mr. A to Mr. B?

Motivation - recommendations



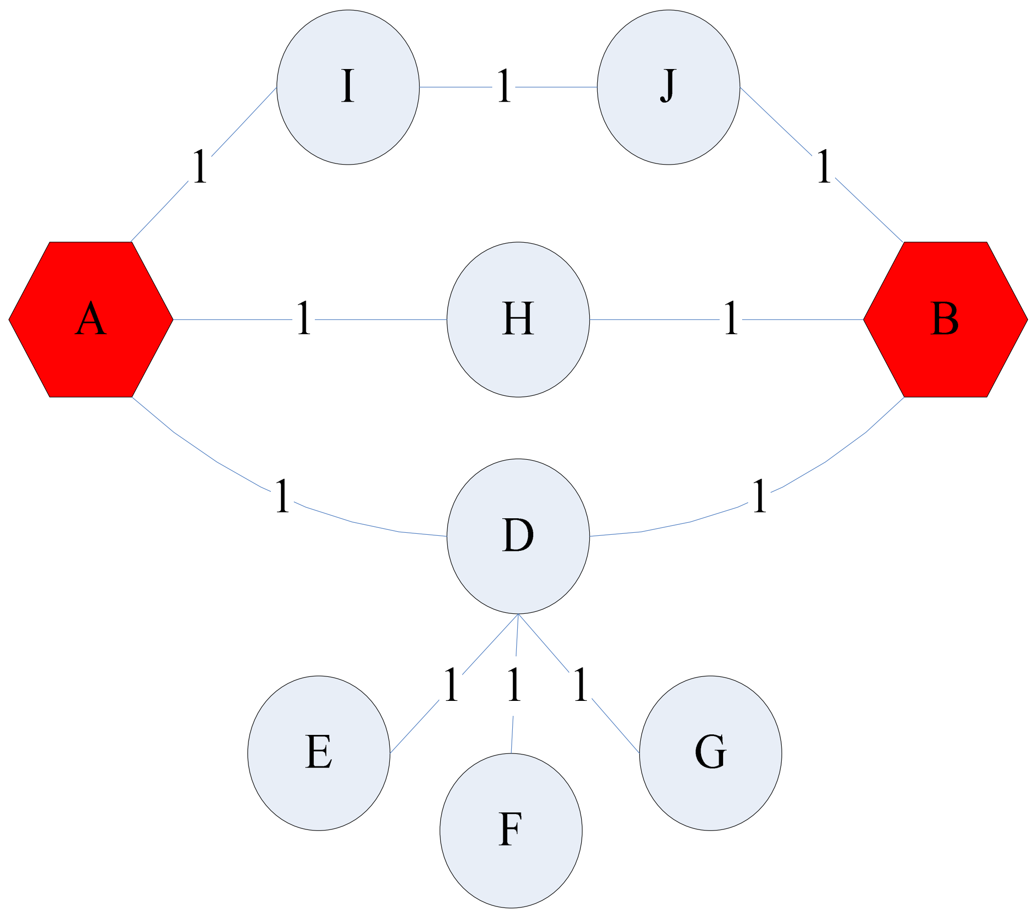
Answer: proximity

- ‘yes’, if ‘A’ and ‘B’ are ‘close’
- ‘yes’, if ‘smith’ and ‘terminator 2’ are ‘close’

QUESTIONS in this part:

- How to measure ‘closeness’/proximity?
- How to do it quickly?
- What else can we do, given proximity scores?

How close is 'A' to 'B'?



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

Why is it useful?

- Recommendation

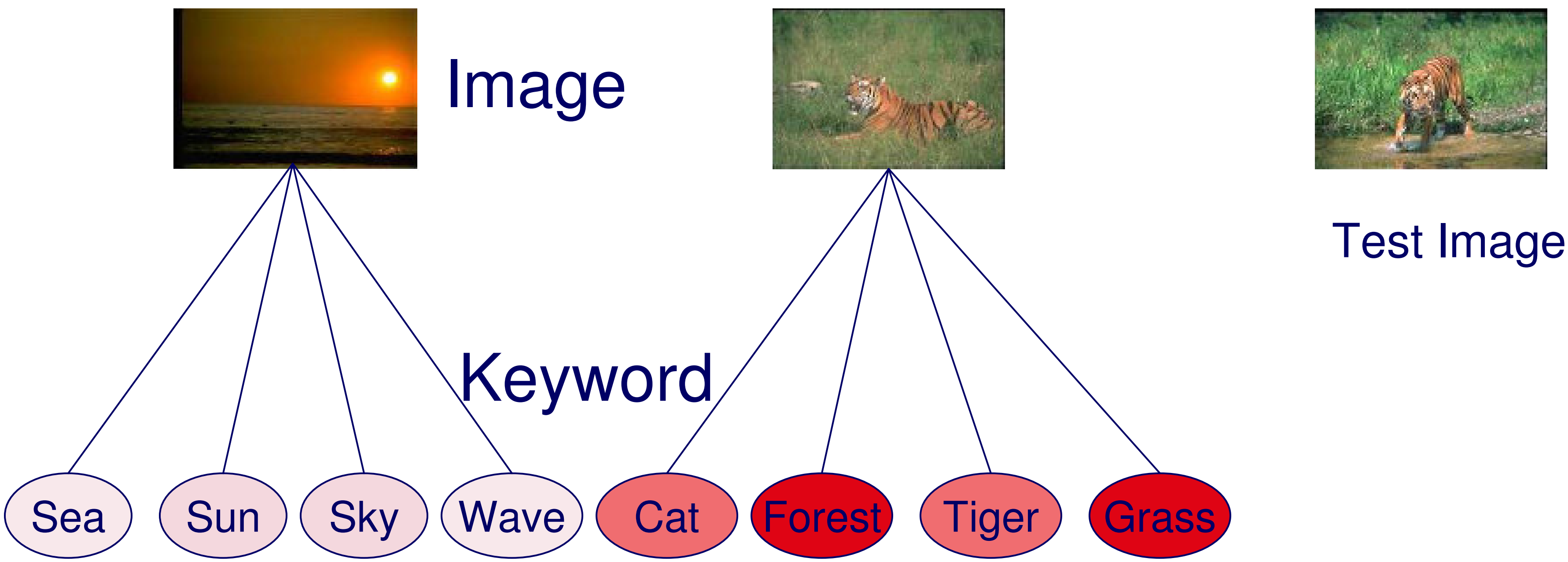
And many more

- **Image captioning** [Pan+]
- **Conn. / CenterPiece subgraphs** [Faloutsos+], [Tong+], [Koren+]

and

- Link prediction [Liben-Nowell+], [Tong+]
- Ranking [Haveliwala], [Chakrabarti+]
- Email Management [Minkov+]
- Neighborhood Formulation [Sun+]
- Pattern matching [Tong+]
- Collaborative Filtering [Fouss+]
- …

Automatic Image Captioning

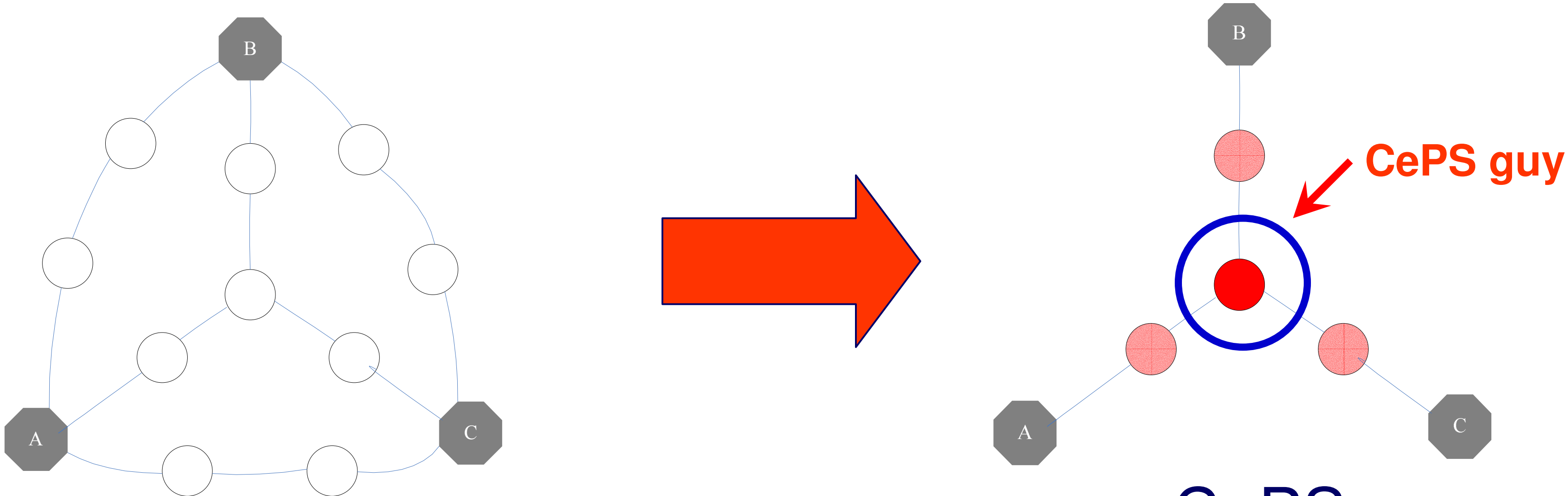


Q: How to assign keywords to the test image?

A: Proximity! [Pan+ 2004]

Center-Piece Subgraph(CePS)

Input **Output**



Original Graph CePS

Q: How to find hub for the black nodes?

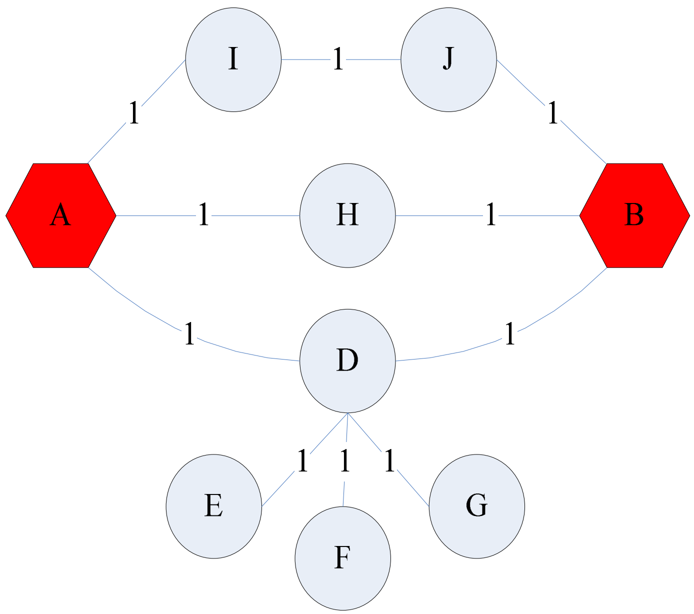
A: Proximity! [Tong+ KDD 2006]

Detailed outline

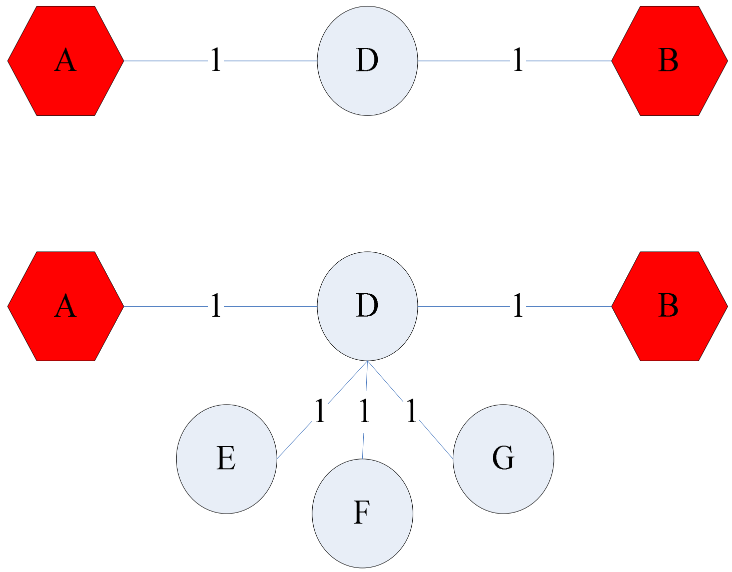
- Problem dfn and motivation
- Solution: Random walk with restarts
- Efficient computation
- Case study: image auto-captioning
- Extensions: bi-partite graphs; tracking
- Conclusions

How close is 'A' to 'B'?

Should be close, if they have
- many,
- short
- 'heavy' paths

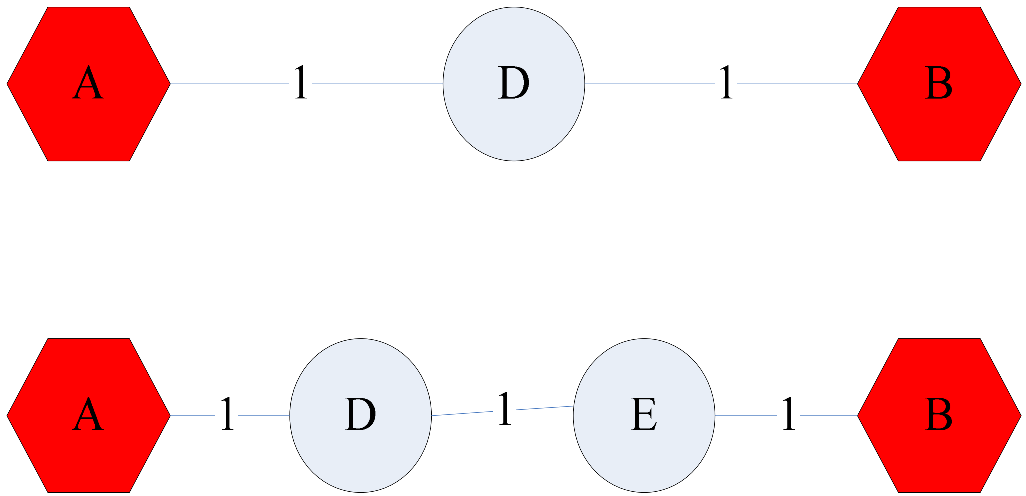


Why not shortest path?



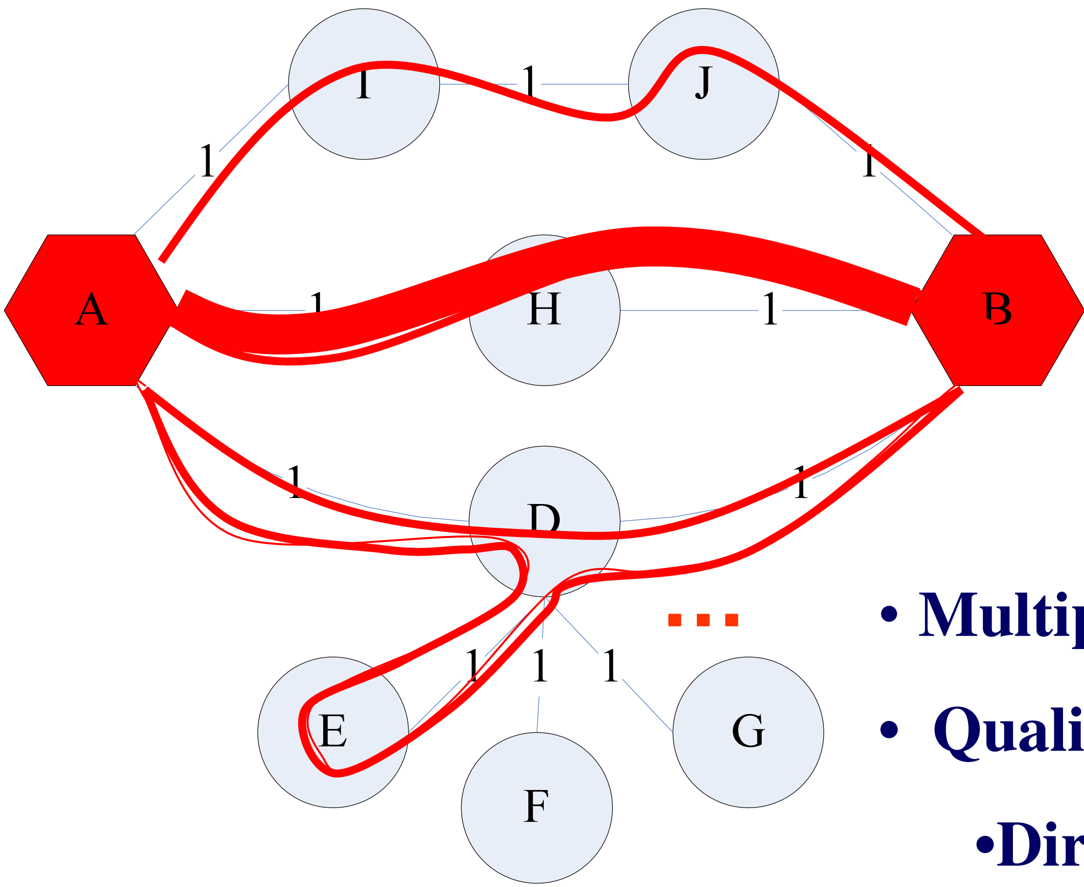
A: 'pizza delivery guy' problem

Why not max. netflow?



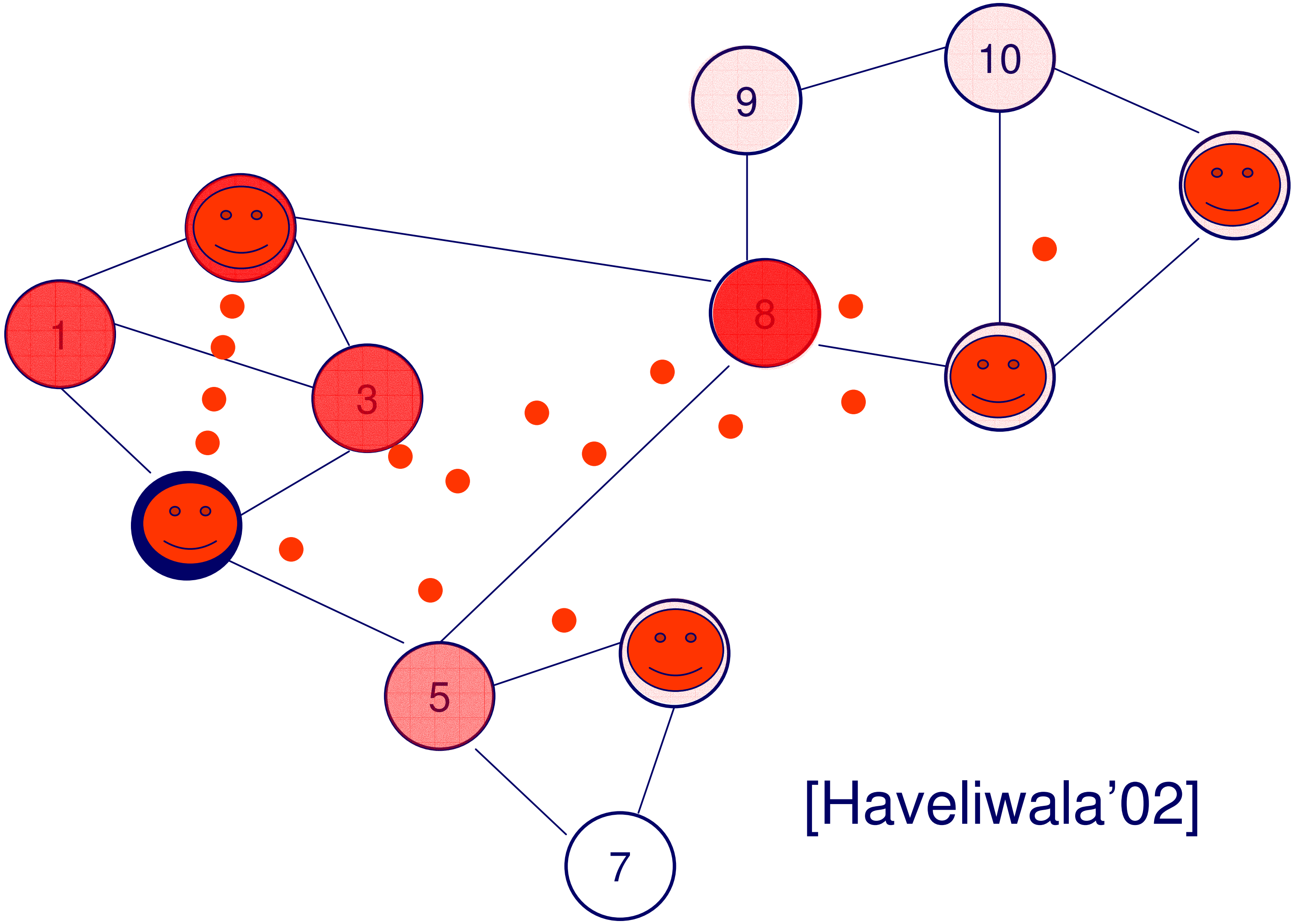
A: No penalty for long paths

What is a ``good'' Proximity?



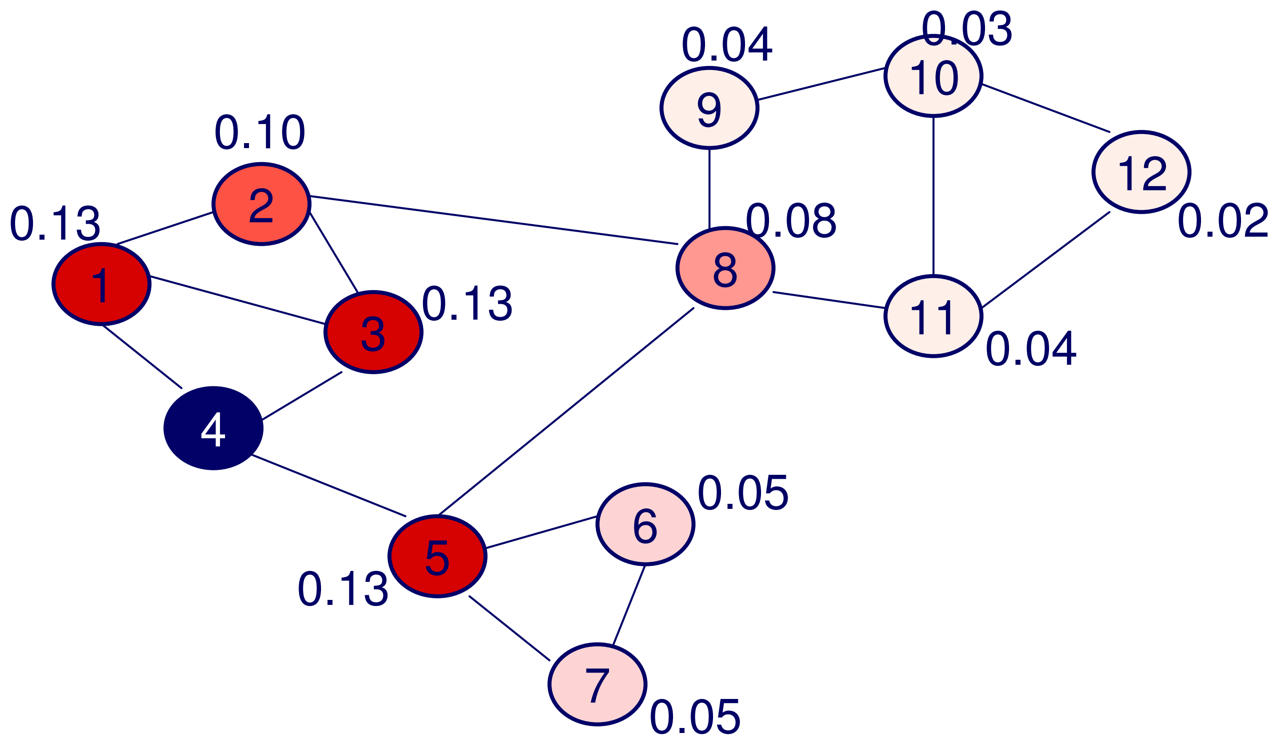
- **Multiple Connections**
- **Quality of connection**
  - **Direct & In-directed Conns**
  - **Length, Degree, Weight…**

Random walk with restart



[Haveliwala'02]

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

$\vec{r}_4$

Why RWR is a good score?

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

j

$Q(i,j) \propto r_{i,j}$

i

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

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

all paths from *i* to *j* with length **1**

all paths from *i* to *j* with length 2

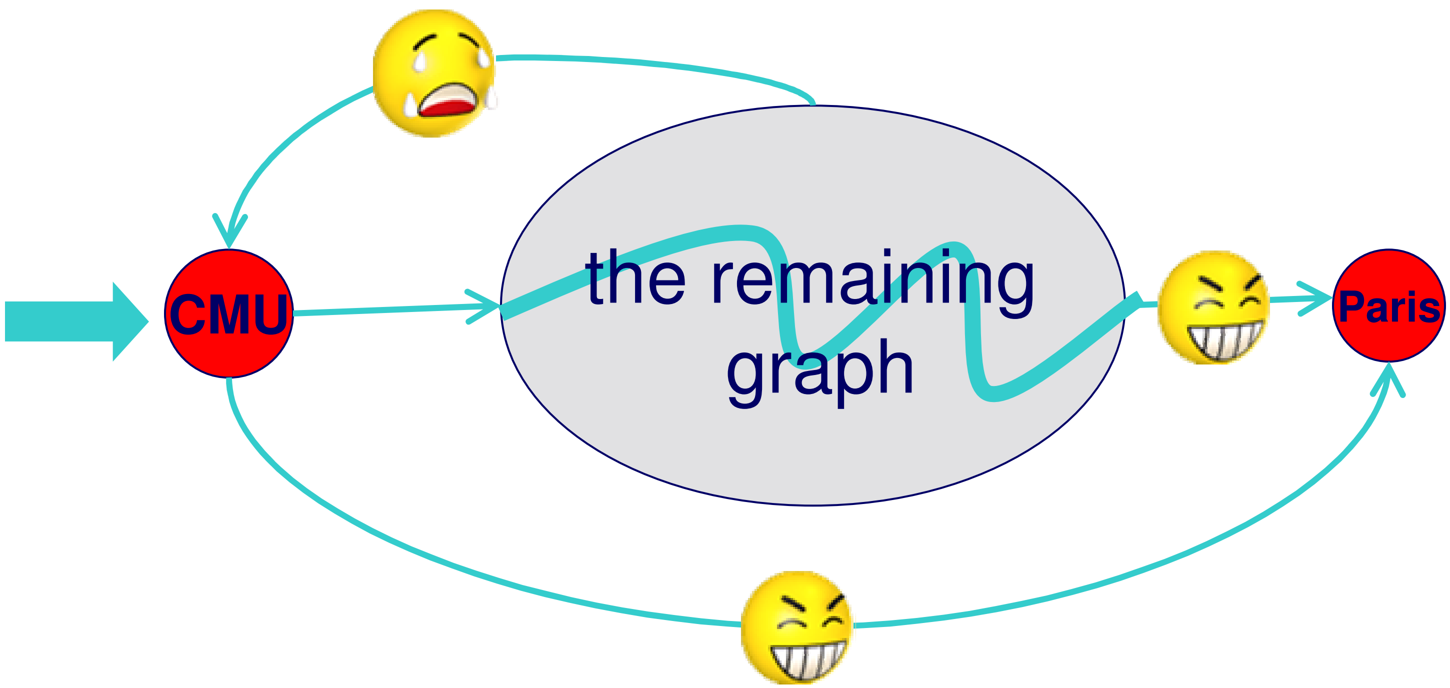
all paths from *i* to *j* with length 3

Detailed outline

- Problem dfn and motivation
- Solution: Random walk with restarts
  - variants
- Efficient computation
- Case study: image auto-captioning
- Extensions: bi-partite graphs; tracking
- Conclusions

Variant: escape probability

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



Esc_Prob = Pr (smile before cry)

details

Other Variants

- Other measure by RWs
  - Community Time/Hitting Time [Fouss+]
  - SimRank [Jeh+]
- Equivalence of Random Walks
  - Electric Networks:
    - EC [Doyle+]; SAEC[Faloutsos+]; CFEC[Koren+]
  - Spring Systems
- Katz [Katz], [Huang+], [Scholkopf+]
- Matrix-Forest-based Alg [Chobotarev+]

details

Other Variants

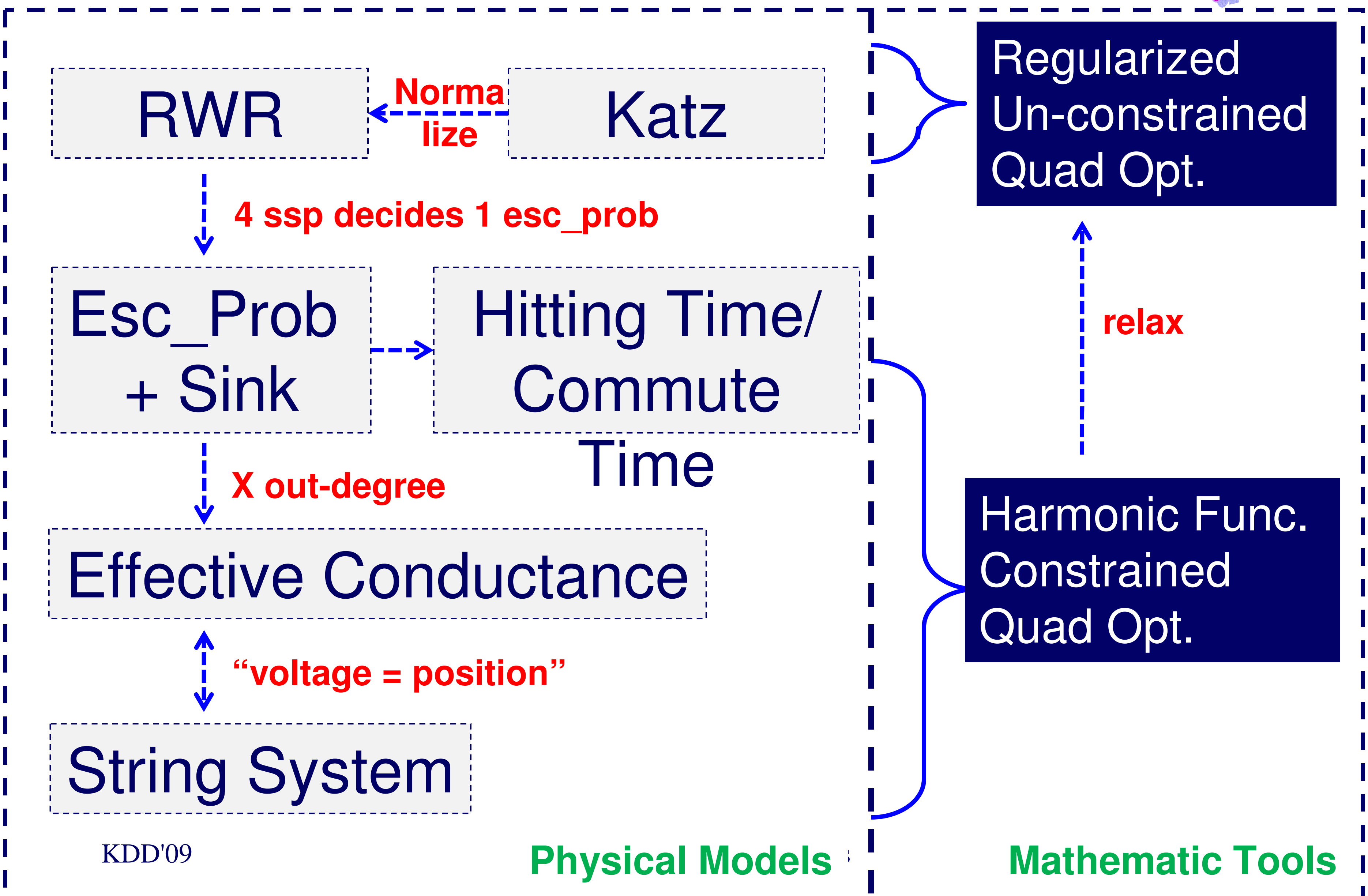
- Other measure by RWs
  - Community Time/Hitting Time [Fouss+]
  - SimRank [Jeh+]

All are "related to" or "similar to" random walk with restart!

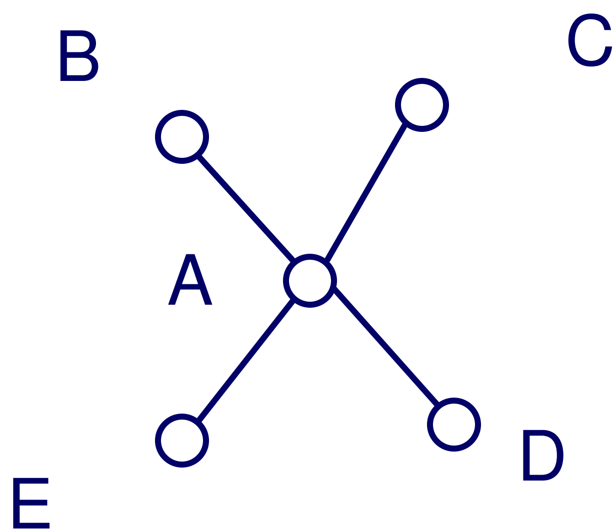
  - Spring Systems
- Katz [Katz], [Huang+], [Scholkopf+]
- Matrix-Forest-based Alg [Chobotarev+]

Map of proximity measurements

details



Notice: Asymmetry (even in undirected graphs)



C-> A : high
A-> C: low

Summary of Proximity Definitions

- Goal: Summarize multiple relationships
- Solutions
  - Basic: Random Walk with Restarts
    - [Haweliwala'02] [Pan+ 2004][Sun+ 2006][Tong+ 2006]
  - Properties: Asymmetry
    - [Koren+ 2006][Tong+ 2007] [Tong+ 2008]
  - Variants: Esc_Prob and many others.
    - [Faloutsos+ 2004] [Koren+ 2006][Tong+ 2007]

Detailed outline

- Problem dfn and motivation
- Solution: Random walk with restarts
- **Efficient computation**
- Case study: image auto-captioning
- Extensions: bi-partite graphs; tracking
- Conclusions

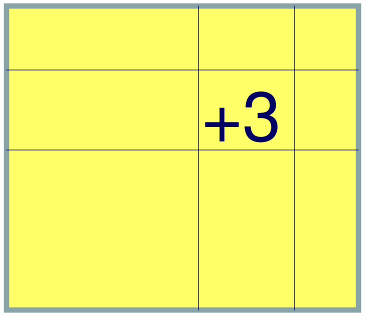
details

Preliminary: Sherman–Morrison Lemma

If:

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

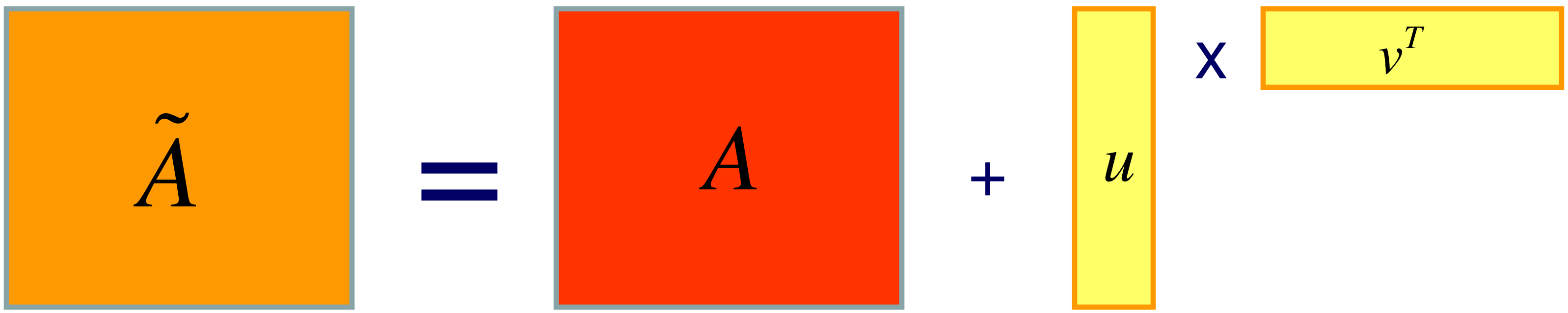
$$\tilde{A} = A +$$



Preliminary: Sherman–Morrison Lemma

details

If:



Then:

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

Sherman – Morrison Lemma – intuition:

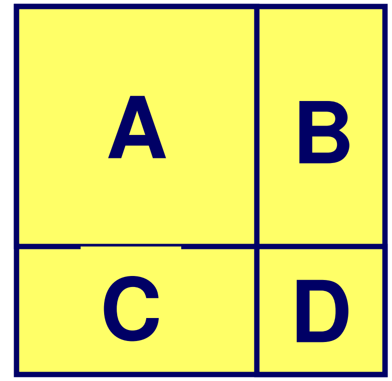
- Given a small perturbation on a matrix

  A -> A'

- We can quickly update its inverse

details

SM: The block-form



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

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 Woodbury Identity

details

SM Lemma: Applications

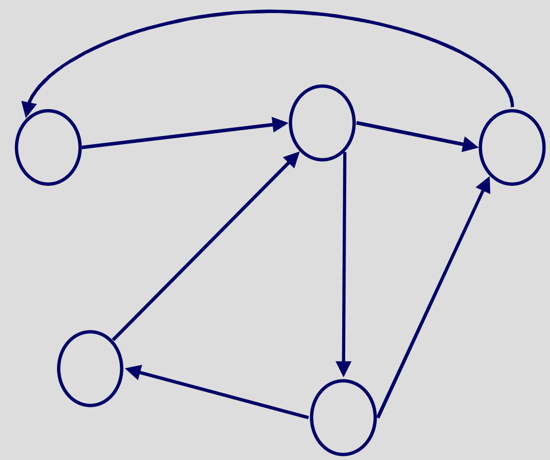
- RLS (Recursive least squares)
  - and almost any algorithm in time series!
- Kalman filtering
- Incremental matrix decomposition
- …
- … and all the fast solutions we will introduce!

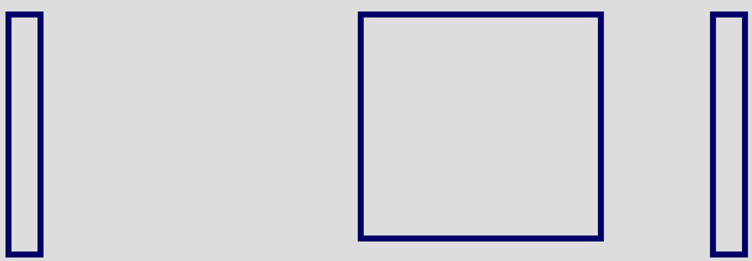
Reminder: PageRank

- With probability *1-c*, fly-out to a random node
- Then, we have

$$\mathbf{p} = c\,\mathbf{B}\,\mathbf{p} + (1\text{-}c)/n\ \mathbf{1} \Rightarrow$$

$$\mathbf{p} = (1\text{-}c)/n\ \ [\mathbf{I} - c\,\mathbf{B}]^{-1}\ \ \mathbf{1}$$





$$\mathbf{p} = c\,\mathbf{B}\,\mathbf{p} + (1\text{-}c)/n\,\mathbf{1}$$

The only difference

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

Ranking vector

Adjacency matrix

Restart p

Starting vector

Computing RWR

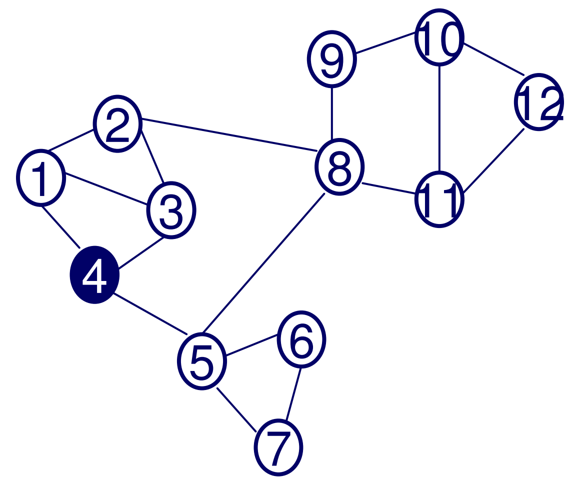
p = c **B** **p** + (1-c)/n **1**

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

Ranking vector — Adjacency matrix — Restart p — Starting vector

$$\begin{pmatrix}0.13\\0.10\\0.13\\0.22\\0.13\\0.05\\0.05\\0.08\\0.04\\0.03\\0.04\\0.02\end{pmatrix} = 0.9 \times \begin{pmatrix}0 & 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\1/3 & 0 & 1/3 & 0 & 0 & 0 & 0 & 1/4 & 0 & 0 & 0 & 0\\1/3 & 1/3 & 0 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\1/3 & 0 & 1/3 & 0 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\0 & 0 & 0 & 1/3 & 0 & 1/2 & 1/2 & 1/4 & 0 & 0 & 0 & 0\\0 & 0 & 0 & 0 & 1/4 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0\\0 & 0 & 0 & 0 & 1/4 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0\\0 & 1/3 & 0 & 0 & 1/4 & 0 & 0 & 0 & 1/2 & 0 & 1/3 & 0\\0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 1/3 & 0 & 0\\0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1/3 & 1/2\\0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 1/3 & 0 & 1/2\\0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 1/3 & 0\end{pmatrix} \begin{pmatrix}0.13\\0.10\\0.13\\0.22\\0.13\\0.05\\0.05\\0.08\\0.04\\0.03\\0.04\\0.02\end{pmatrix} + 0.1 \times \begin{pmatrix}0\\0\\0\\1\\0\\0\\0\\0\\0\\0\\0\\0\end{pmatrix}$$

n x 1 — n x n — n x 1



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

$$
\left\{ \; ? \; \right\} = 0.9 \times \begin{pmatrix}
0 & 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/3 & 0 & 1/3 & 0 & 0 & 0 & 0 & 1/4 & 0 & 0 & 0 & 0 \\
1/3 & 1/3 & 0 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/3 & 0 & 1/3 & 0 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1/3 & 0 & 1/2 & 1/2 & 1/4 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1/4 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1/4 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1/3 & 0 & 0 & 1/4 & 0 & 0 & 0 & 1/2 & 0 & 1/3 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 1/3 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1/3 & 1/2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 1/3 & 0 & 1/2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 1/3 & 0
\end{pmatrix} \left\{ \; ? \; \right\} + 0.1 \times \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}
$$

Query

Ranking vector

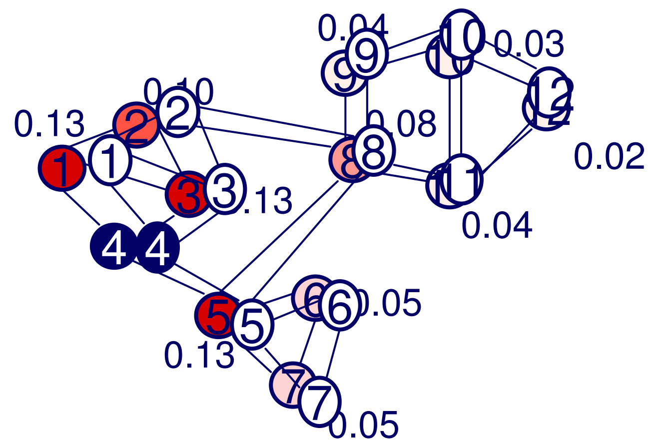
Adjacency matrix

Ranking vector

Starting vector

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

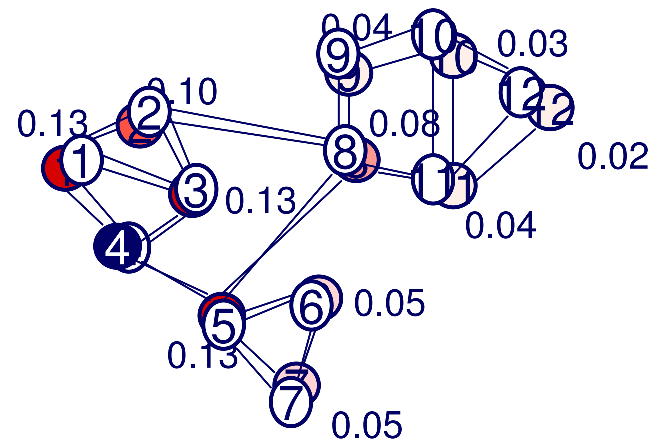
$$\underbrace{\begin{pmatrix} 0.13 \\ 0.10 \\ 0.13 \\ 0.22 \\ 0.13 \\ 0.05 \\ 0.05 \\ 0.08 \\ 0.04 \\ 0.03 \\ 0.04 \\ 0.02 \end{pmatrix}}_{\vec{r}_i} = 0.9 \times \underbrace{\begin{pmatrix} 0 & 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 1/3 & 0 & 0 & 0 & 0 & 1/4 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 1/3 & 0 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/3 & 0 & 1/2 & 1/2 & 1/4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/4 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/4 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/3 & 0 & 0 & 1/4 & 0 & 0 & 0 & 1/2 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1/3 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 1/3 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 1/3 & 0 \end{pmatrix}}_{\tilde{W}} \underbrace{\begin{pmatrix} 0.13 \\ 0.10 \\ 0.13 \\ 0.22 \\ 0.13 \\ 0.05 \\ 0.05 \\ 0.08 \\ 0.04 \\ 0.03 \\ 0.04 \\ 0.02 \end{pmatrix}}_{\vec{r}_i} + 0.1 \times \underbrace{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}}_{\vec{e}_i}$$



- No pre-computation/ light storage
- Slow on-line response O(mE)

PreCompute

R:



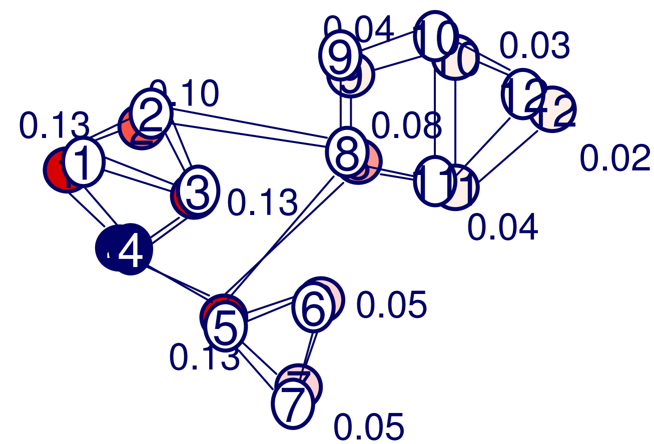
$$R = c \times Q$$

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

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

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

$$\begin{pmatrix} 0.13 \\ 0.10 \\ 0.13 \\ 0.22 \\ 0.13 \\ 0.05 \\ 0.05 \\ 0.08 \\ 0.04 \\ 0.03 \\ 0.04 \\ 0.02 \end{pmatrix} \leftarrow 0.1\times \quad \begin{matrix} 1.29 \\ 0.96 \\ 1.29 \\ 2.06 \\ 1.27 \\ 0.52 \\ 0.52 \\ 0.82 \\ 0.28 \\ 0.34 \\ 0.38 \\ 0.21 \end{matrix}$$

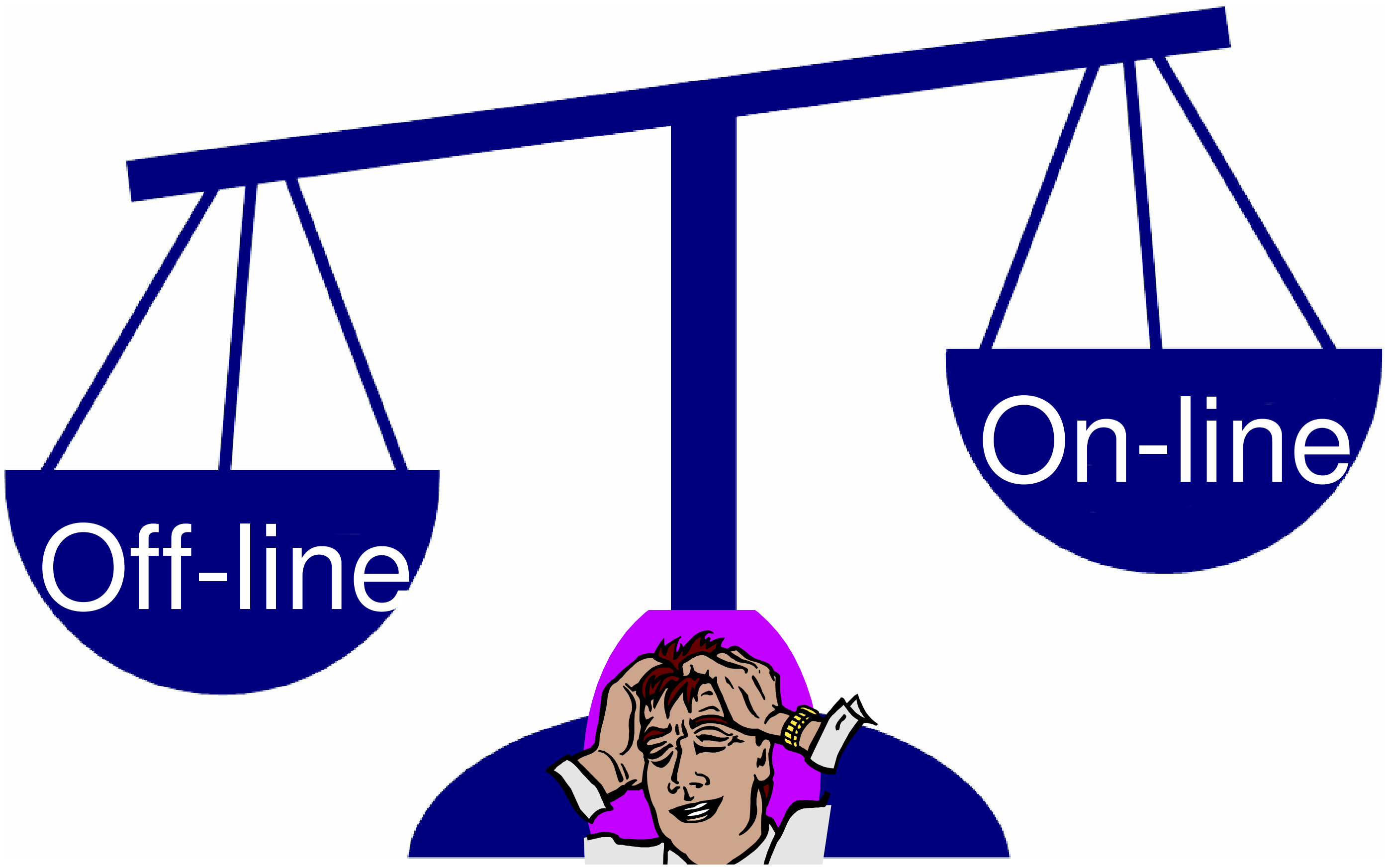


Fast on-line response

Heavy pre-computation/storage cost

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

Q: How to Balance?



How to balance?

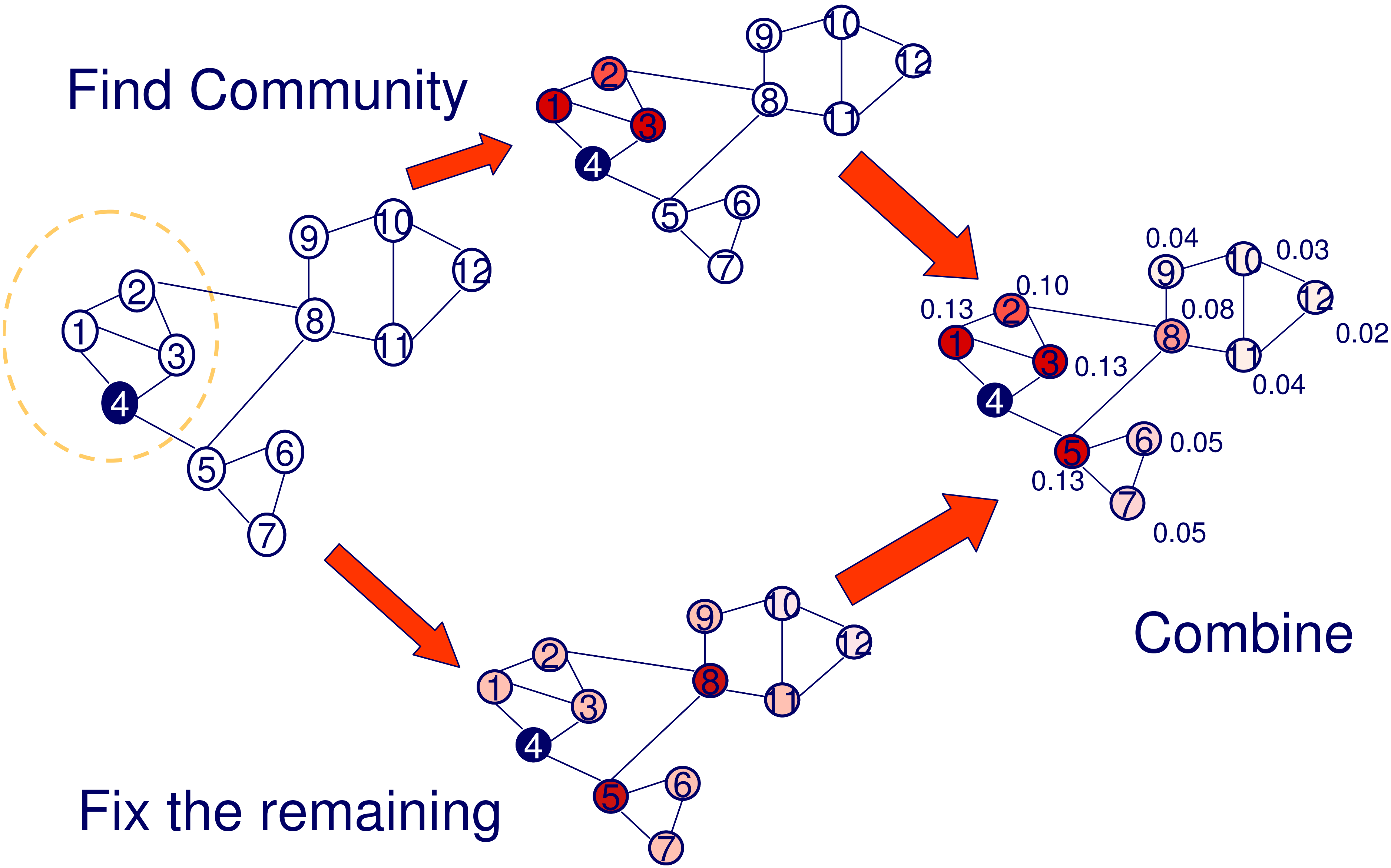
Idea ('B-Lin')

- Break into communities
- Pre-compute all, within a community
- Adjust (with S.M.) for 'bridge edges'

H. Tong, C. Faloutsos, & J.Y. Pan. *Fast Random Walk with Restart and Its Applications*. ICDM, 613-622, 2006.

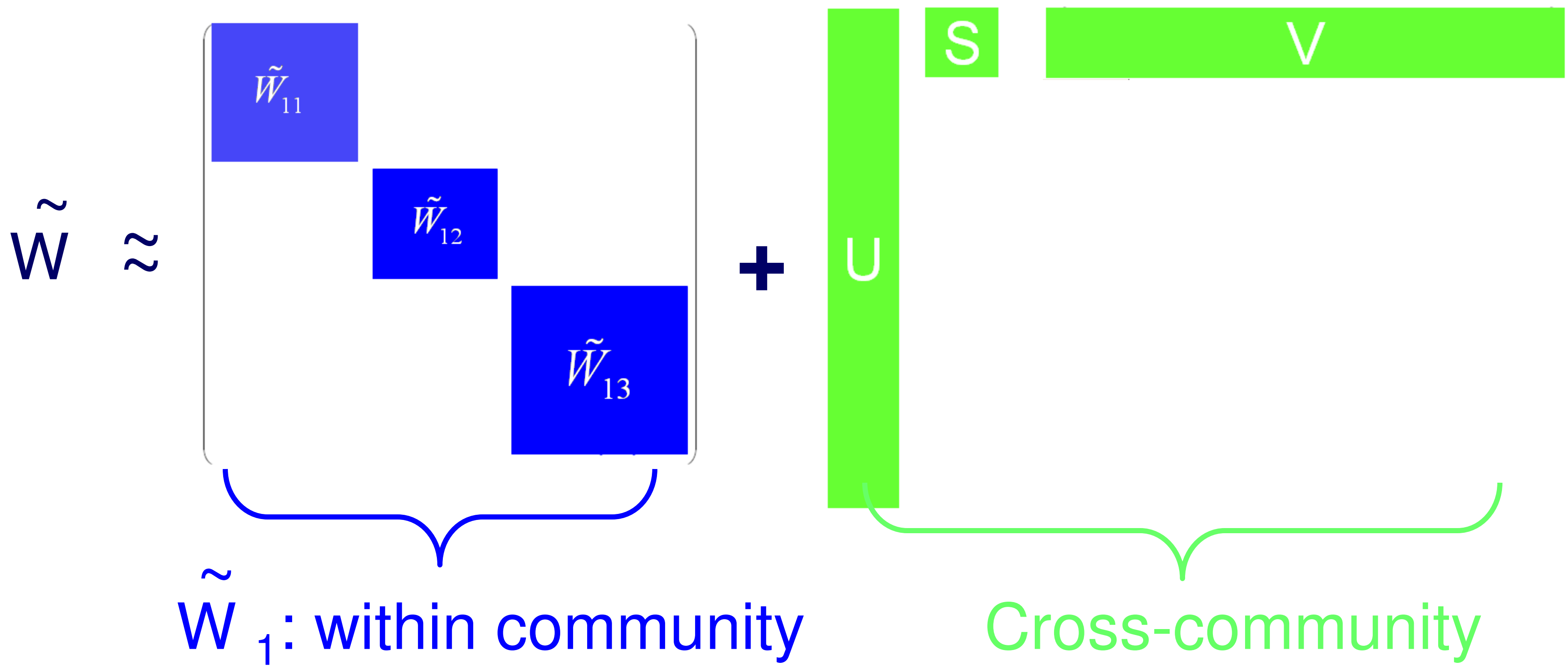
B_Lin: Basic Idea

[Tong+ ICDM 2006]



details

B_Lin: details



details

B_Lin: details

$$\left(I - c\tilde{W}\right)^{-1} \approx \left(\boxed{I - c\tilde{W}_1} - \boxed{cUSV}\right)^{-1}$$

Easy to be inverted

LRA difference

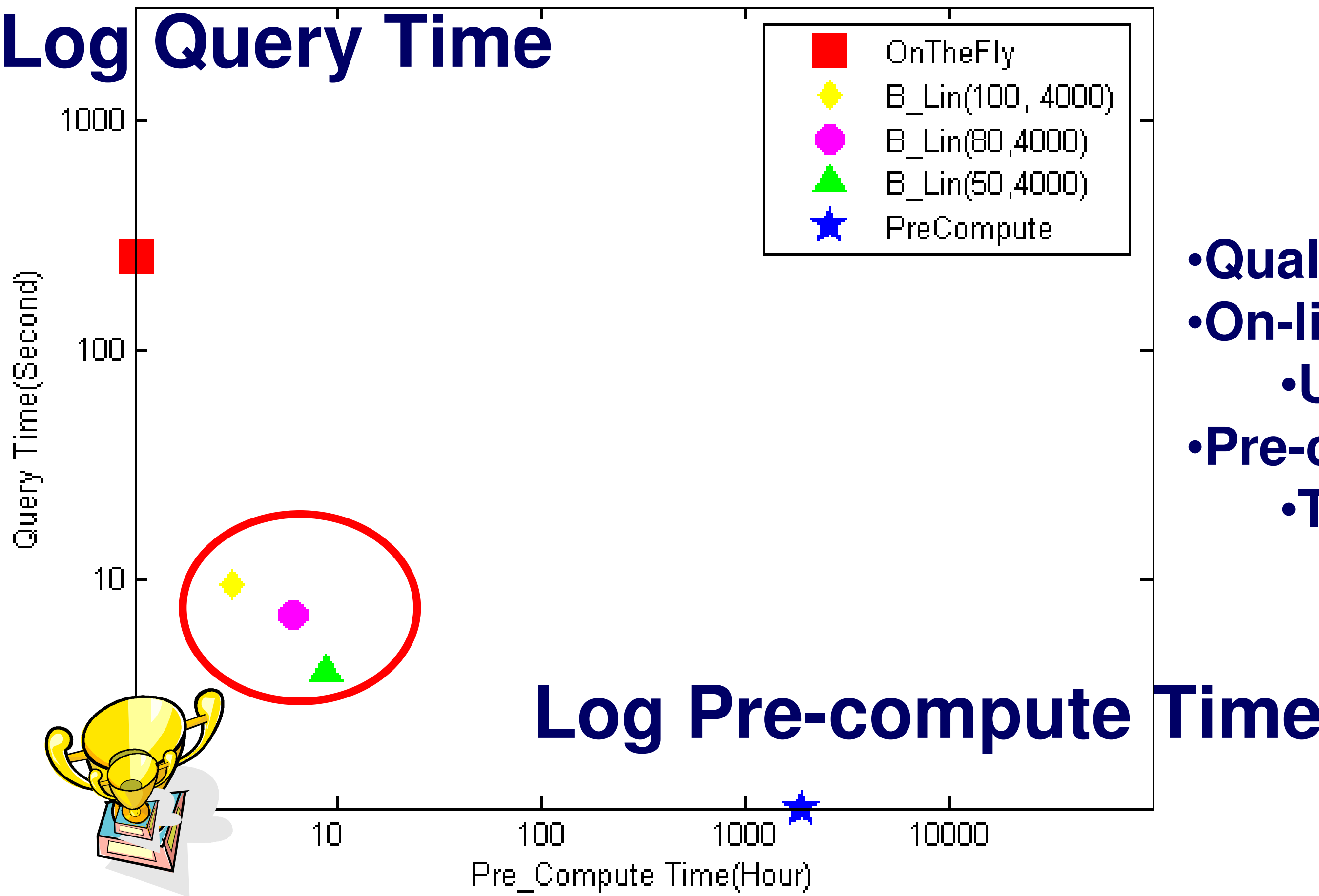
SM Lemma!

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

Query Time vs. Pre-Compute Time



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

Detailed outline

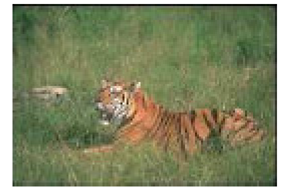
- Problem dfn and motivation
- Solution: Random walk with restarts
- Efficient computation
- Case study: image auto-captioning
- Extensions: bi-partite graphs; tracking
- Conclusions

gCaP: Automatic Image Caption

- Q

{Sea Sun Sky Wave}

...

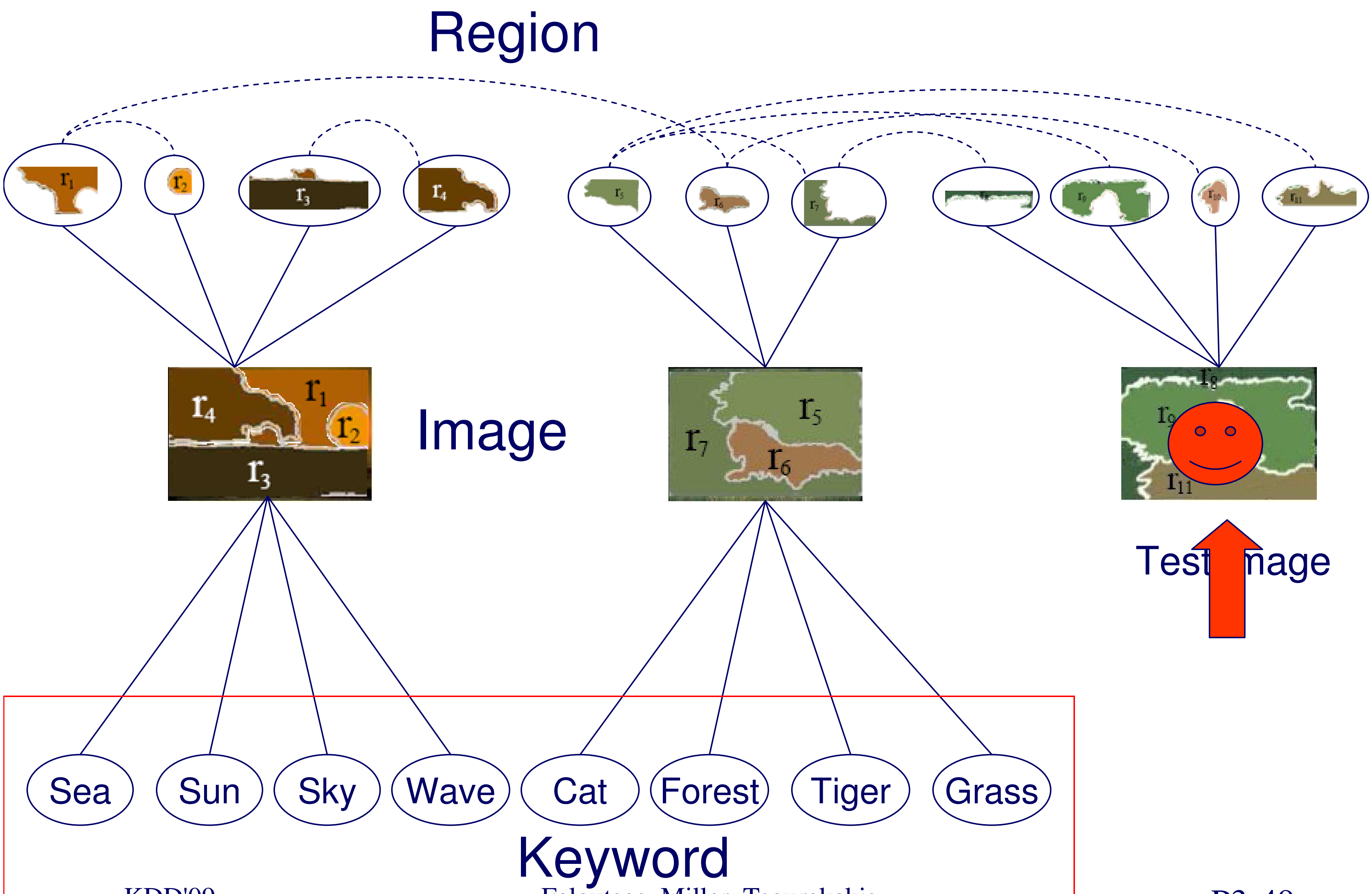


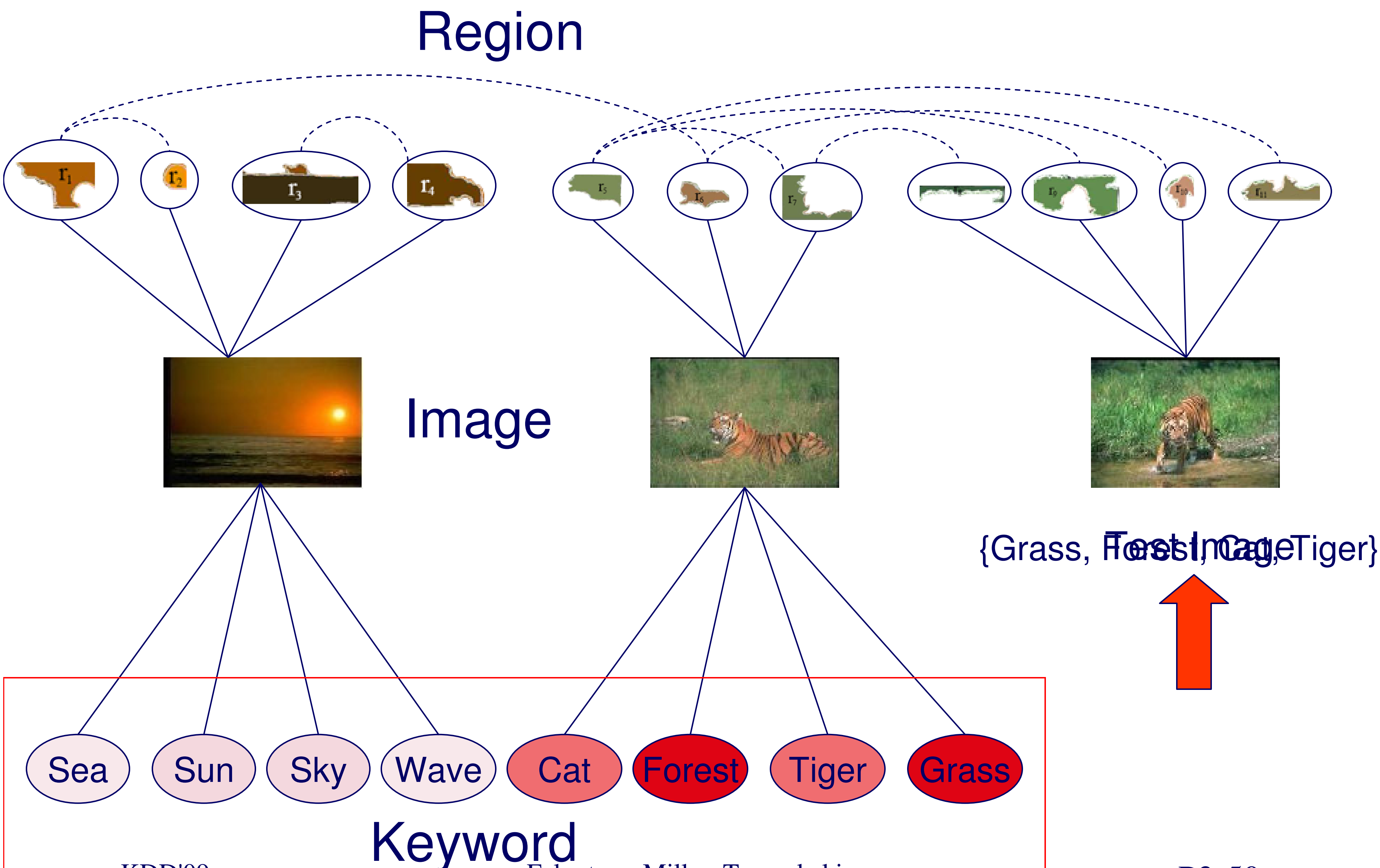
{Cat Forest Grass Tiger}

{?, ?, ?,}

A: Proximity!

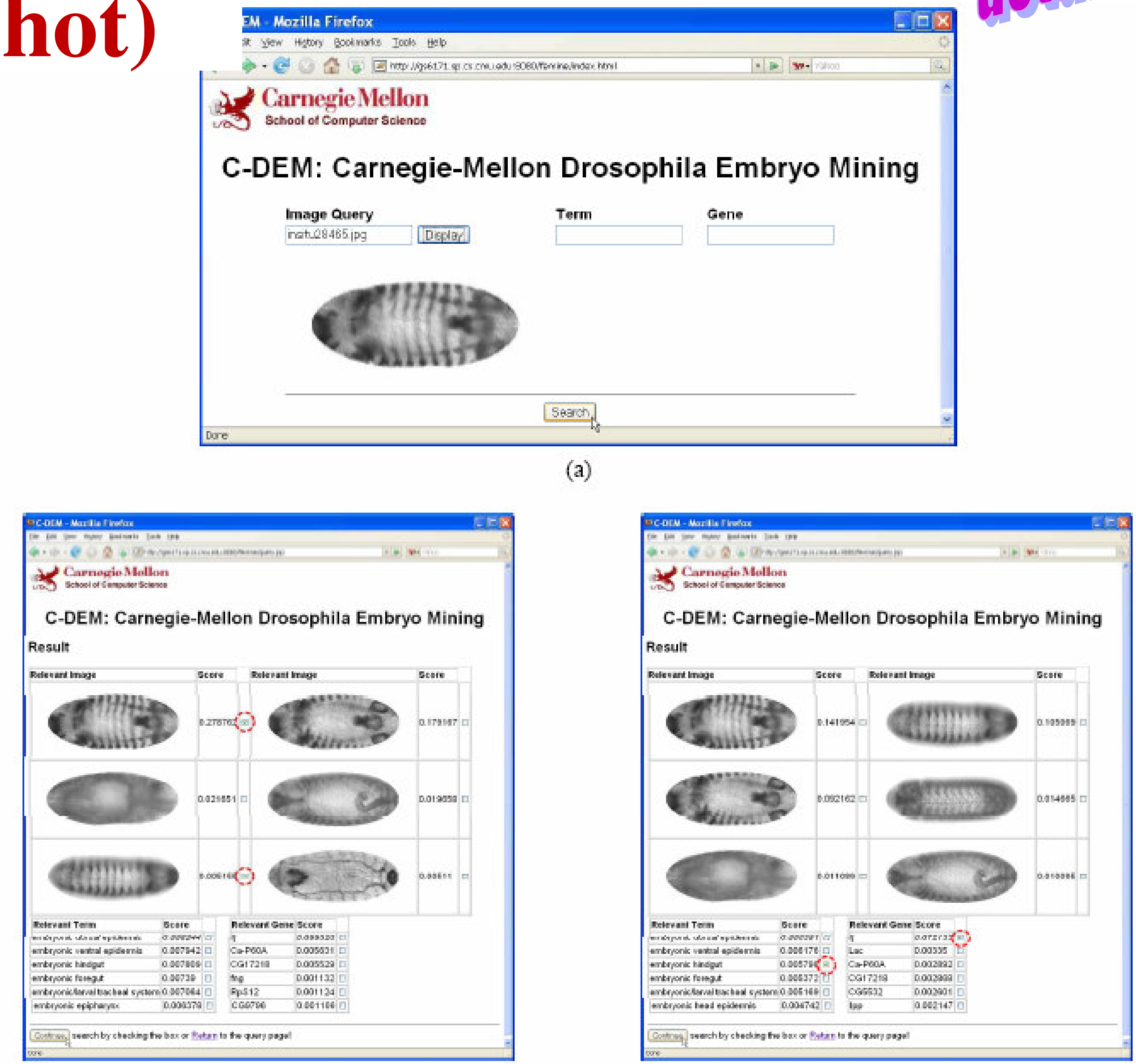
[Pan+ KDD2004]


Region
Image
Test Image
Sea
Sun
Sky
Wave
Cat
Forest
Tiger
Grass
Keyword


Region
r1
r2
r3
r4
Image
Test Image
{Grass, Forest, Cat, Tiger}
Sea
Sun
Sky
Wave
Cat
Forest
Tiger
Grass
Keyword

C-DEM (Screen-shot)

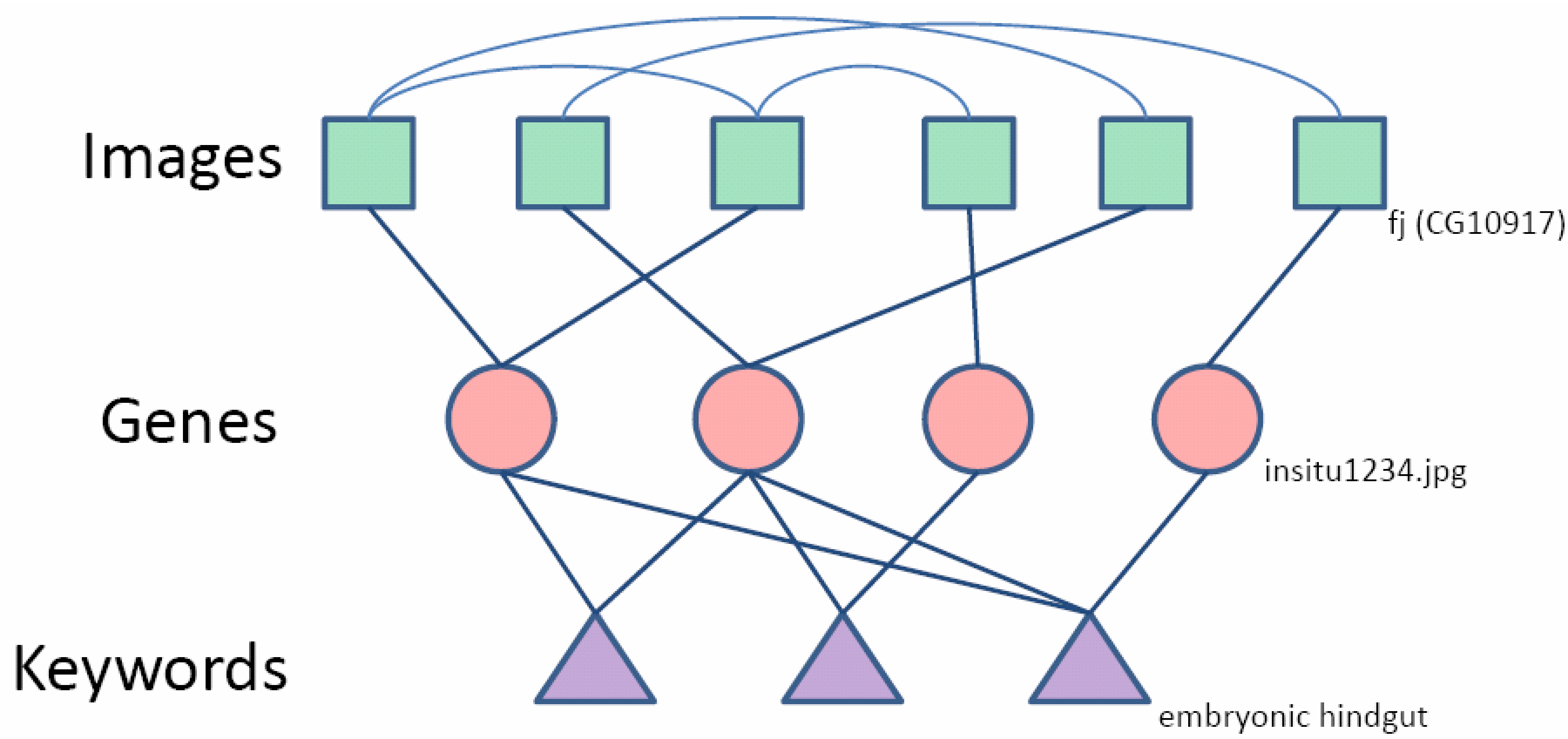
details



(a) (b) (c)

details

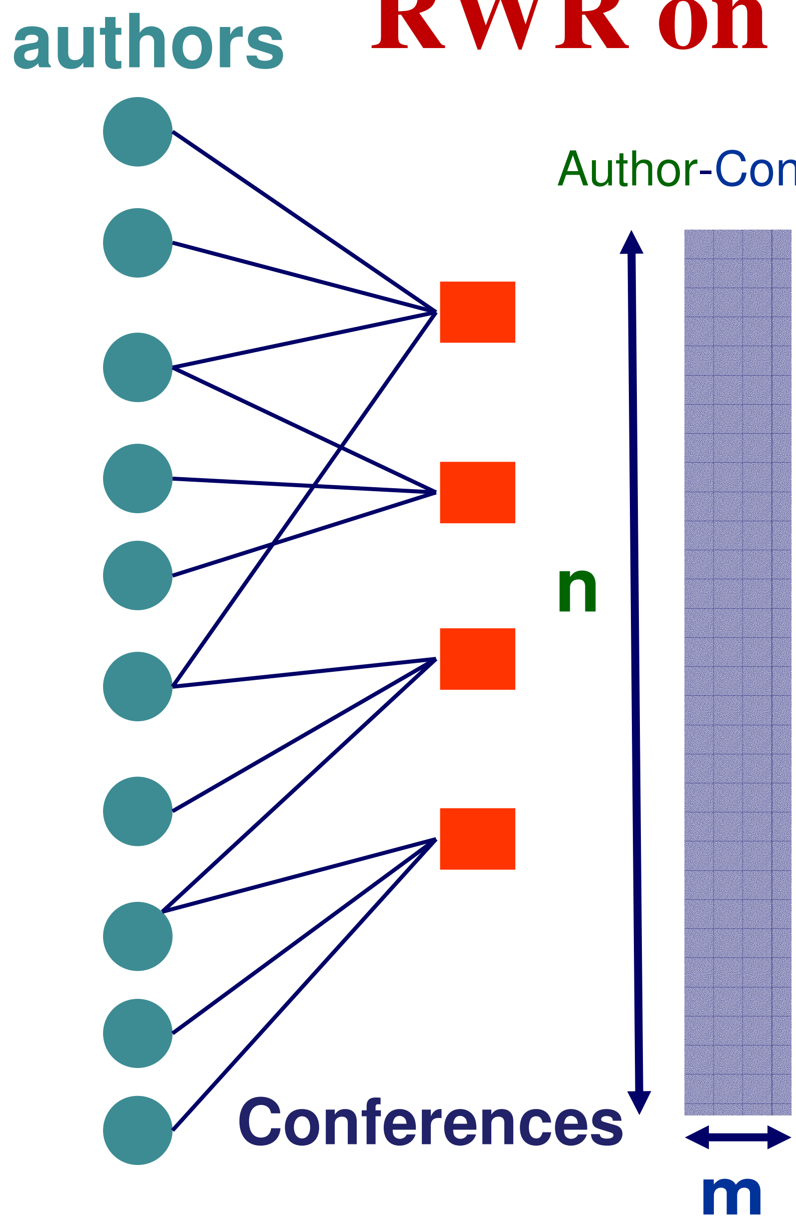
C-DEM: Multi-Modal Query System for Drosophila Embryo Databases [Fan+ VLDB 2008]



Detailed outline

- Problem dfn and motivation
- Solution: Random walk with restarts
- Efficient computation
- Case study: image auto-captioning
- Extensions: **bi-partite graphs**; tracking
- Conclusions

RWR on Bipartite Graph



Observation: n >> m!

Examples:

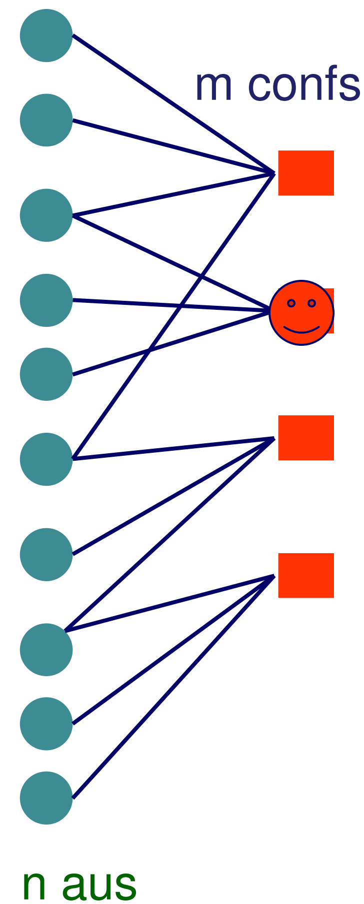
1. DBLP: 400k aus, 3.5k confs
2. NetFlix: 2.7M usrs,18k mvs

RWR on Skewed bipartite graphs

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

$$\left\{ \textbf{?} \right\} = 0.9 \times \begin{pmatrix} 0 & A_r \\ A_c & 0 \end{pmatrix} \left\{ \textbf{?} \right\} + 0.1 \times \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$\underbrace{\qquad\qquad}_{n}$ $\underbrace{\quad}_{m}$



Idea:

- Pre-compute the smallest, m x m matrix
- Use it to compute the rest proximities, on the fly

H. Tong, S. Papadimitriou, P.S. Yu & C. Faloutsos. *Proximity Tracking on Time-Evolving Bipartite Graphs*. SDM 2008.

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

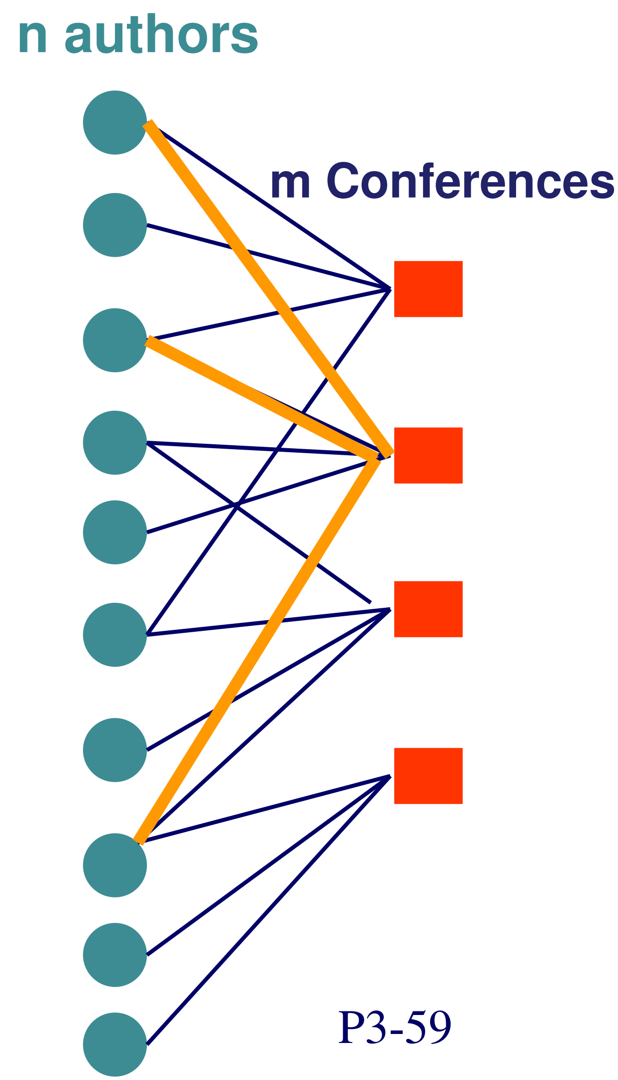
Detailed outline

- Problem dfn and motivation
- Solution: Random walk with restarts
- Efficient computation
- Case study: image auto-captioning
- Extensions: bi-partite graphs; **tracking**
- Conclusions

Problem: update

E’ edges changed

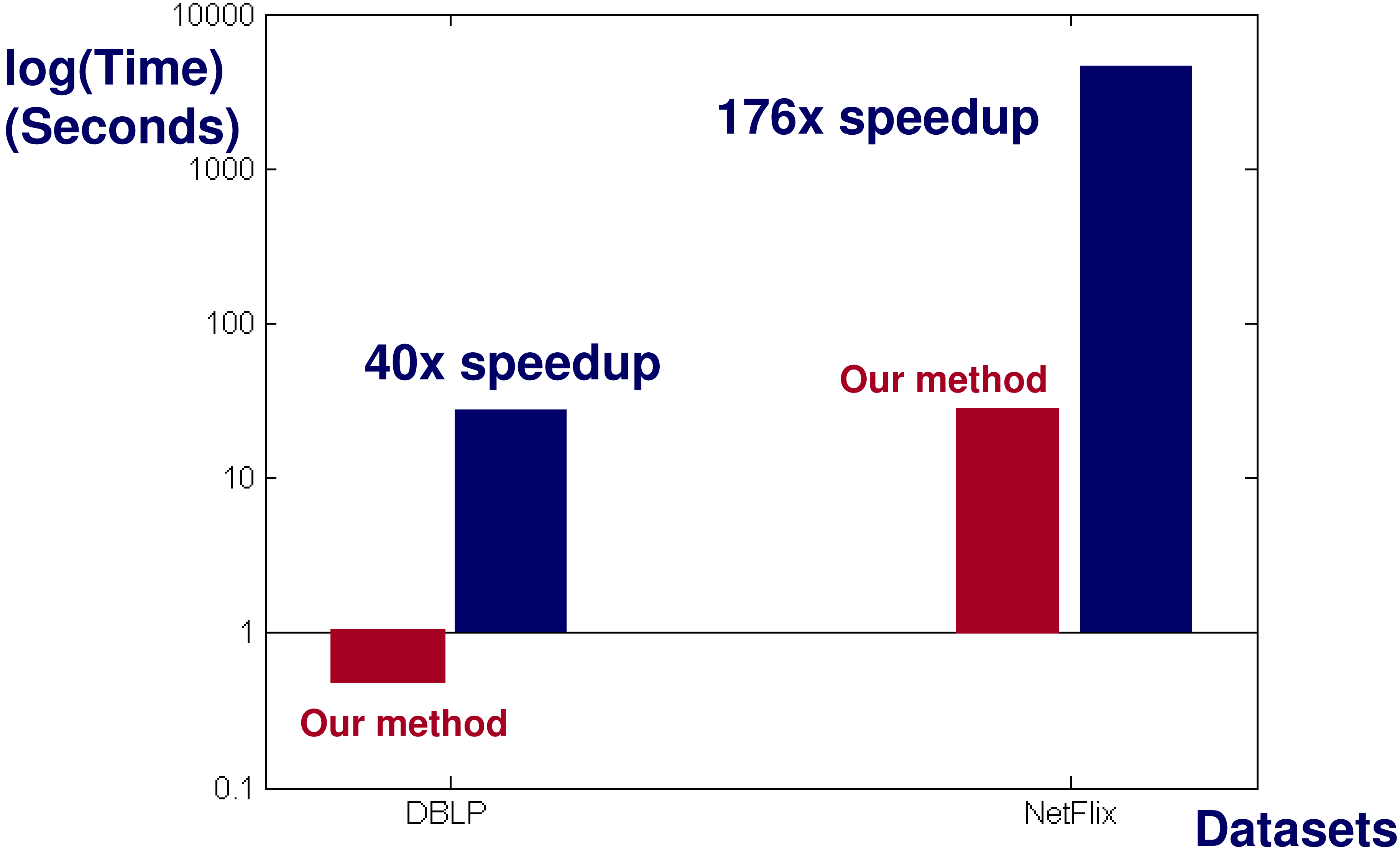
Involves n’ authors, m’ confs.



Solution:

- Use Sherman-Morrison to quickly update the inverse matrix

Fast-Single-Update



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

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

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

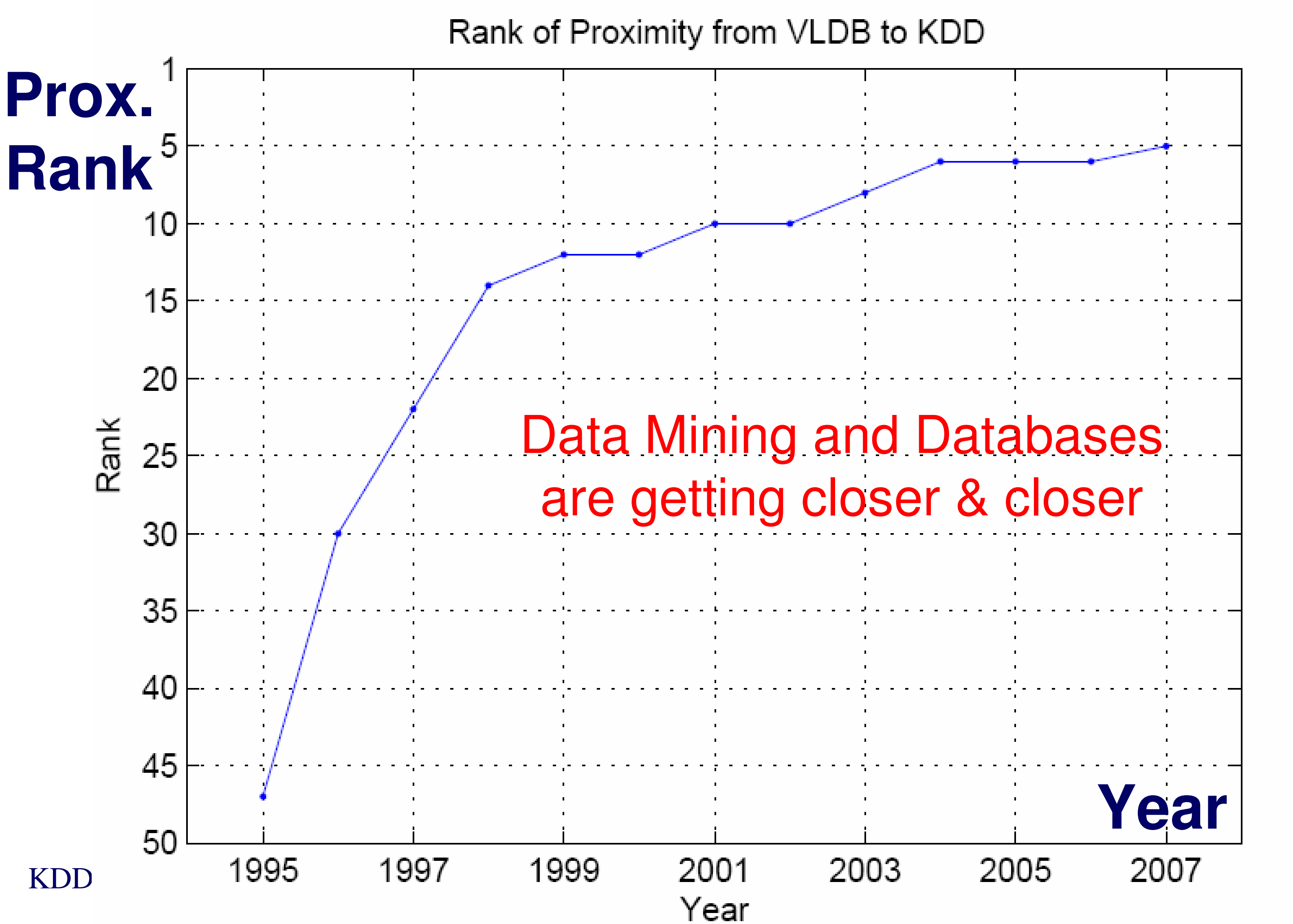
ICDE	CIKM	KDD	ICDM
ICDCS	ICDCS	SIGMOD	KDD
SIGMETRICS	ICDE	ICDM	ICDE
PDIS	SIGMETRICS	CIKM	SDM
VLDB	ICMCS	ICDCS	VLDB
1992	1997	2002	2007

Databases
Performance
Distributed Sys.

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

Databases
Data Mining

KDD's Rank wrt. VLDB over years



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

T. Sejnowski

1987	1989	1991	1993	1995	1997	1999
'Abbott_L'	'Bower_J'	'Hinton_G'	'Sejnowski_T'	'Sejnowski_T'	'Sejnowski_T'	'Sejnowski_T'
'Burr_D'	'Hinton_G'	'Koch_C'	'Koch_C'	'Jordan_M'	'Jordan_M'	'Koch_C'
'Denker_J'	'Tesauro_G'	'Bower_J'	'Hinton_G'	'Hinton_G'	'Koch_C'	'Jordan_M'
'Scofield_C	'Denker_J'	'Sejnowski_T'	'Mozer_M'	'Koch_C'	'Hinton_G'	'Hinton_G'
'Bower_J'	'Mead_C'	'LeCun_Y'	'LeCun_Y'	'Mozer_M'	'Mozer_M'	'Mozer_M'
'Brown_N'	'Tenorio_M'	'Mozer_M'	'Denker_J'	'Bengio_Y'	'Dayan_P'	'Dayan_P'
'Carley_L'	'Sejnowski_T'	'Denker_J'	'Bower_J'	'Lippmann_R'	'Bengio_Y'	'Singh_S'
'Chou_P'	'Lippmann_R'	'Waibel_A'	'Kawato_M'	'LeCun_Y'	'Barto_A'	'Bengio_Y'
'Chover_J'	'Touretzky_D'	'Moody_J'	'Waibel_A'	'Waibel_A'	'Tresp_V'	'Tresp_V'
'Eeckman_F'	'Koch_C'	'Lippmann_R'	'Simard_P'	'Simard_P'	'Moody_J'	'Moody_J'

M. Jordan

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

Conclusions - 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
- **Computation**
  - Sherman–Morrison Lemma
  - Fast Incremental Computation
- **Applications**
  - Recommendations; auto-captioning; tracking
  - Center-piece Subgraphs (next)
  - E-mail management; anomaly detection, …

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.

References

- 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.

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.

References

- 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.

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.

References

- 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. SDM 2008.

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

Resources

- www.cs.cmu.edu/~htong/soft.htm

  For software, papers, and ppt of presentations

- www.cs.cmu.edu/~htong/tut/cikm2008/cikm_tutorial.html

  For the CIKM'08 tutorial on graphs and proximity

Again, thanks to **Hanghang TONG** for permission to use his foils in this part