

Carnegie Mellon

Mining graphs and time series: patterns, anomalies, and fraud detection


Part 1: Graphs

Node importance & community detection

Christos Faloutsos

CMU SCS


<https://www.cs.cmu.edu/~christos/TALKS/19-Gol>



1

Carnegie Mellon

Roadmap



- Introduction
- Part#1: Graphs
- Part#2: Time series
- Part#3: extras (visualization, etc)
- Conclusions


Gov. of India Copyright (C) 2019 C. Faloutsos

2



2

Carnegie Mellon

Roadmap



- Introduction – Motivation
- Part#1: Graphs
 - P1.1: properties/patterns in graphs
 - ➔ – P1.2: node importance
 - P1.3: community detection
 - P1.4: fraud/anomaly detection
 - P1.5: belief propagation




Gov. of India Copyright (C) 2019 C. Faloutsos

3


3


Carnegie Mellon

Roadmap



- Introduction – Motivation
- Part#1: Graphs
 - P1.1: properties/patterns in graphs
 - ➔ – P1.2: node importance
 - PageRank and Personalized PR
 - HITS
 - (SVD)
 - SALSA





Gov. of India

Copyright (C) 2019 C. Faloutsos


4


4


Carnegie Mellon


'Recipe' Structure:

- Problem definition
- Short answer/solution
- LONG answer – details
- Conclusion/short-answer









Gov. of India

Copyright (C) 2019 C. Faloutsos


5

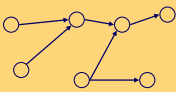
5

Carnegie Mellon

Node importance - Motivation:

- Given a graph (eg., web pages containing the desirable query word)
- Q1: Which node is the most important?
- Q2: How close is node 'A' to node 'B'?





Gov. of India

Copyright (C) 2019 C. Faloutsos

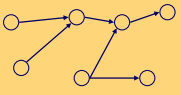
6

6

Carnegie Mellon

Node importance - Motivation:

- Given a graph (eg., web pages containing the desirable query word)
- Q1: Which node is the most important?
 - PageRank ($PR = RWR$), HITS, SALSA
- Q2: How close is node 'A' to node 'B'?
 - Personalized P.R. (/SALSA)



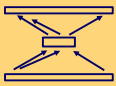
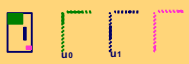
Gov. of India Copyright (C) 2019 C. Faloutsos 7

7

Carnegie Mellon

SVD properties

- ✓ Hidden/latent variable detection
- ✓ Compute node importance (HITS)
- ✓ Block detection
- ✓ Dimensionality reduction
- ✓ Embedding (linear)
 - SVD is a special case of 'deep neural net'



Gov. of India Copyright (C) 2019 C. Faloutsos 8

8

Carnegie Mellon

Roadmap

- Introduction – Motivation
- Part#1: Graphs
 - P1.1: properties/patterns in graphs
 - P1.2: node importance
 - PageRank and Personalized PR
 - HITS
 - SALSA

Gov. of India Copyright (C) 2019 C. Faloutsos 9

9

Carnegie Mellon

PageRank (google)



Larry Page Sergey Brin

- Brin, Sergey and Lawrence Page (1998). *Anatomy of a Large-Scale Hypertextual Web Search Engine*. 7th Intl World Wide Web Conf.
- Page, Brin, Motwani, and Winograd (1999). *The PageRank citation ranking: Bringing order to the web*. Technical Report

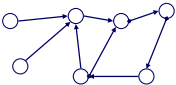
Gov. of India Copyright (C) 2019 C. Faloutsos 10

10

Carnegie Mellon

Problem: PageRank

Given a directed graph, find its most interesting/central node



A node is important, if its parents are important (recursive, but OK!)


Gov. of India Copyright (C) 2019 C. Faloutsos 11

11


Carnegie Mellon

Problem: PageRank - solution

Given a directed graph, find its most interesting/central node



Proposed solution: Random walk; spot most 'popular' node (-> steady state prob. (ssp))



A node **high ssp**, if its parents have **high ssp** (recursive, but OK!)

Gov. of India Copyright (C) 2019 C. Faloutsos 12

12

Carnegie Mellon

(Simplified) PageRank algorithm **DETAILS**

- Let A be the adjacency matrix;
- let B be the transition matrix: transpose, column-normalized - then

From To

$$B = \begin{bmatrix} & & 1 & & \\ 1 & & & 1 & \\ & 1/2 & & & 1/2 \\ & & & & 1/2 \\ & 1/2 & & & \end{bmatrix} \begin{bmatrix} p1 \\ p2 \\ p3 \\ p4 \\ p5 \end{bmatrix} = \begin{bmatrix} p1 \\ p2 \\ p3 \\ p4 \\ p5 \end{bmatrix}$$

Gov. of India Copyright (C) 2019 C. Faloutsos 13

13

Carnegie Mellon

(Simplified) PageRank algorithm **DETAILS**

- $B p = p$

B $p = p$

$$B = \begin{bmatrix} & & 1 & & \\ 1 & & & 1 & \\ & 1/2 & & & 1/2 \\ & & & & 1/2 \\ & 1/2 & & & \end{bmatrix} \begin{bmatrix} p1 \\ p2 \\ p3 \\ p4 \\ p5 \end{bmatrix} = \begin{bmatrix} p1 \\ p2 \\ p3 \\ p4 \\ p5 \end{bmatrix}$$

Gov. of India Copyright (C) 2019 C. Faloutsos 14

14

Carnegie Mellon

Definitions **DETAILS**

- A Adjacency matrix (from-to)
- D Degree matrix = $(\text{diag} (d_1, d_2, \dots, d_n))$
- B Transition matrix: to-from, column normalized
 $B = A^T D^{-1}$

Gov. of India Copyright (C) 2019 C. Faloutsos 15

15

Carnegie Mellon

DETAILS

(Simplified) PageRank algorithm

- $\mathbf{B} \mathbf{p} = \mathbf{1} * \mathbf{p}$
- thus, \mathbf{p} is the **eigenvector** that corresponds to the highest eigenvalue ($=1$, since the matrix is column-normalized)
- Why does such a \mathbf{p} exist?
 - \mathbf{p} exists if \mathbf{B} is $n \times n$, nonnegative, irreducible [Perron–Frobenius theorem]

Gov. of India


Copyright (C) 2019 C. Faloutsos


16


16

Carnegie Mellon

(Simplified) PageRank algorithm

- In short: imagine a particle randomly moving along the edges 
- compute its steady-state probabilities (ssp)

Full version of algo: with occasional random jumps 

Why? To make the matrix irreducible 

Gov. of India


Copyright (C) 2019 C. Faloutsos


17


17

Carnegie Mellon

(Simplified) PageRank algorithm

- In short: imagine a particle randomly moving along the edges 
- compute its steady-state probabilities (ssp)

Full version of algo: with occasional random jumps 

Why? To make the matrix irreducible 

Gov. of India

Copyright (C) 2019 C. Faloutsos


18

18

Carnegie Mellon


(Simplified) PageRank algorithm

- In short: imagine a particle randomly moving along the edges
- compute its steady-state probabilities (ssp)



Full version of algo: with occasional random jumps

Why? To make the matrix irreducible



Gov. of India

Copyright (C) 2019 C. Faloutsos


19

19

Carnegie Mellon


(Simplified) PageRank algorithm

- In short: imagine a particle randomly moving along the edges
- compute its steady-state probabilities (ssp)



Full version of algo: with occasional random jumps

Why? To make the matrix irreducible



Gov. of India

Copyright (C) 2019 C. Faloutsos


20

20

Carnegie Mellon


(Simplified) PageRank algorithm

- In short: imagine a particle randomly moving along the edges
- compute its steady-state probabilities (ssp)



Full version of algo: with occasional random jumps

Why? To make the matrix irreducible



Gov. of India

Copyright (C) 2019 C. Faloutsos

21


21

Carnegie Mellon

(Simplified) PageRank algorithm

- In short: imagine a particle randomly moving along the edges
- compute its steady-state probabilities (ssp)

PageRank = PR
= Random Walk with Restarts = RWR
= Random surfer



Gov. of India

Copyright (C) 2019 C. Faloutsos

22

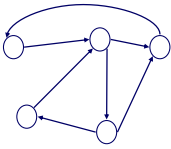
22

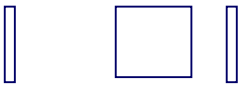
Carnegie Mellon

Full Algorithm

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

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

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




Gov. of India

Copyright (C) 2019 C. Faloutsos

23


23

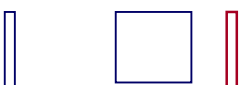

Carnegie Mellon

Full Algorithm

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

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

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


Gov. of India

Copyright (C) 2019 C. Faloutsos


24

24

Carnegie Mellon

Notice:

- $\text{pageRank} \sim \text{in-degree}$
- (and HITS, also: $\sim \text{in-degree}$)



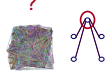


Gov. of India Copyright (C) 2019 C. Faloutsos 25

25

Carnegie Mellon

Roadmap


- Introduction – Motivation
- Part#1: Graphs
 - P1.1: properties/patterns in graphs
 - P1.2: node importance
 - PageRank and **Personalized PR**
 - HITS

Gov. of India Copyright (C) 2019 C. Faloutsos 26

26

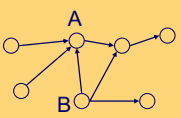
Carnegie Mellon

Node importance - Motivation:



- Given a graph (eg., web pages containing the desirable query word)
- Q1: Which node is the most important?

➔ • Q2: How close is node 'A' to node 'B'?



Gov. of India Copyright (C) 2019 C. Faloutsos 27

27

Carnegie Mellon

Personalized P.R.

Taher H. Haveliwala. 2002. *Topic-sensitive PageRank*. (WWW '02). 517-526.
<http://dx.doi.org/10.1145/511446.511513>

Page L., Brin S., Motwani R., and Winograd T. (1999). *The PageRank citation ranking: Bringing order to the web*. Technical Report

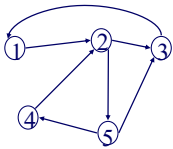
Gov. of India Copyright (C) 2019 C. Faloutsos 29

29

Carnegie Mellon

Extension: Personalized P.R.

- How close is '4' to '2'?
- (or: if I like page/node '2', what else would you **recommend**?)



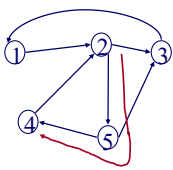
Gov. of India Copyright (C) 2019 C. Faloutsos 30

30

Carnegie Mellon

Extension: Personalized P.R.

- How close is '4' to '2'?
- (or: if I like page/node '2', what else would you **recommend**?)



Gov. of India Copyright (C) 2019 C. Faloutsos 31

31

Carnegie Mellon

Extension: Personalized P.R.

- How close is '4' to '2'?
- (or: if I like page/node '2', what else would you **recommend**?)

Gov. of India Copyright (C) 2019 C. Faloutsos 32

32

Carnegie Mellon

Extension: Personalized P.R.

- How close is '4' to '2'?
- (or: if I like page/node '2', what else would you **recommend**?)

High score (A -> B) if

- Many
- Short
- Heavy paths A->B

Gov. of India Copyright (C) 2019 C. Faloutsos 33

33

Carnegie Mellon

Extension: Personalized P.R.

- With probability $1-c$, fly-out to a ~~random~~ **your favorite** node(s)
- Then, we have

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

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

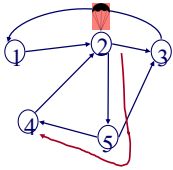
Gov. of India Copyright (C) 2019 C. Faloutsos 34

34

Carnegie Mellon

Extension: Personalized P.R.

- How close is '4' to '2'?
- A: compute Personalized P.R. of '4', restarting from '2'




Gov. of India Copyright (C) 2019 C. Faloutsos 35

35

Carnegie Mellon

Extension: Personalized P.R.

- How close is '4' to '2'?
- A: compute Personalized P.R. of '4', restarting from '2' – Related to
 - 'escape' probability
 - 'round trip' probability
 - ...



Gov. of India Copyright (C) 2019 C. Faloutsos 36

36

Carnegie Mellon

Roadmap

- Introduction – Motivation
- Part#1: Graphs
 - P1.1: properties/patterns in graphs
 - P1.2: node importance
 - PageRank and Personalized PR
 - Fast computation - 'Pixie'
 - HITS

37

Carnegie Mellon

Extension: Personalized P.R. DETAILS

- Q: Faster computation than:

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

Gov. of India Copyright (C) 2019 C. Faloutsos 38

38

Carnegie Mellon

Pixie algorithm

Chantat Eksombatchai, Pranav Jindal, Jerry Zitao Liu, Yuchen Liu, Rahul Sharma, Charles Sugnet, Mark Ulrich, Jure Leskovec:
Pixie: A System for Recommending 3+ Billion Items to 200+ Million Users in Real-Time.
 WWW 2018: 1775-1784
<https://dl.acm.org/citation.cfm?doid=3178876.3186183>

Gov. of India Copyright (C) 2019 C. Faloutsos 39

39

Carnegie Mellon

Pixie algorithm

- Q: Faster computation than:

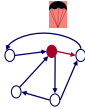


$$\mathbf{p} = (1-c)/n \ [\mathbf{I} - c \mathbf{B}]^{-1} \mathbf{1}$$
- A: **simulate** a few R.W.
 - keep visit counts C_i
 - fast and nimble

Gov. of India Copyright (C) 2019 C. Faloutsos 40

40

Carnegie Mellon

Personalized PageRank algorithm






Gov. of India Copyright (C) 2019 C. Faloutsos 41

41

Carnegie Mellon

Personalized PageRank algorithm

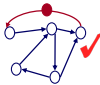




Gov. of India Copyright (C) 2019 C. Faloutsos 42

42

Carnegie Mellon

Personalized PageRank algorithm






Gov. of India Copyright (C) 2019 C. Faloutsos 43

43

Carnegie Mellon

Personalized PageRank algorithm





Gov. of India



Copyright (C) 2019 C. Faloutsos



44

44

Carnegie Mellon

Personalized PageRank algorithm





Gov. of India



Copyright (C) 2019 C. Faloutsos

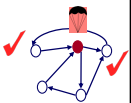
45

45

Carnegie Mellon

Personalized PageRank algorithm





Gov. of India



Copyright (C) 2019 C. Faloutsos

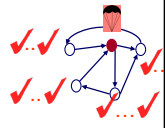
46

46

Carnegie Mellon

Personalized PageRank algorithm



Gov. of India


Copyright (C) 2019 C. Faloutsos

47

47

Carnegie Mellon

Roadmap



- Introduction – Motivation
- Part#1: Graphs
 - P1.1: properties/patterns in graphs
 - P1.2: node importance
 - PageRank and **Personalized PR**
 - Fast computation - 'Pixie'
 - Other applications
 - HITS




Gov. of India


Copyright (C) 2019 C. Faloutsos

48

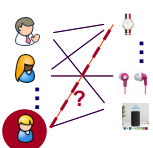
48

Carnegie Mellon

Applications of node proximity



- ➔ • Recommendation
- Link prediction
- 'Center Piece Subgraphs'
- ...



Gov. of India

Copyright (C) 2019 C. Faloutsos

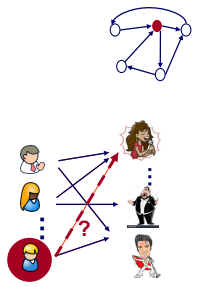
49

49

Carnegie Mellon

Applications of node proximity

- Recommendation
- ➔ • Link prediction
- ‘Center Piece Subgraphs’
- ...



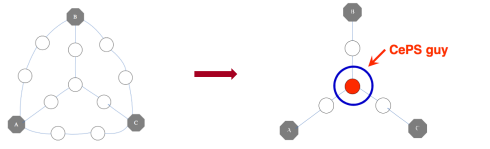
Gov. of India Copyright (C) 2019 C. Faloutsos 50

50

Carnegie Mellon

Applications of node proximity

- Recommendation
- Link prediction
- ➔ • ‘Center Piece Subgraphs’
- ...



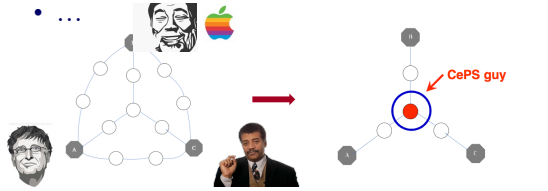
Gov. of India Copyright (C) 2019 C. Faloutsos 51

51

Carnegie Mellon

Applications of node proximity

- Recommendation
- Link prediction
- ➔ • ‘Center Piece Subgraphs’
- ...



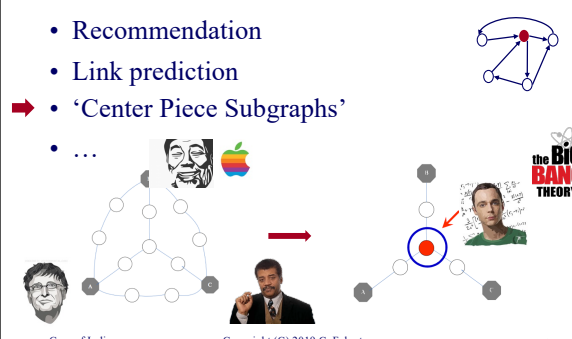
Gov. of India Copyright (C) 2019 C. Faloutsos 52

52

Carnegie Mellon

Applications of node proximity

- Recommendation
- Link prediction
- ➔ • ‘Center Piece Subgraphs’
- ...



Gov. of India Copyright (C) 2019 C. Faloutsos

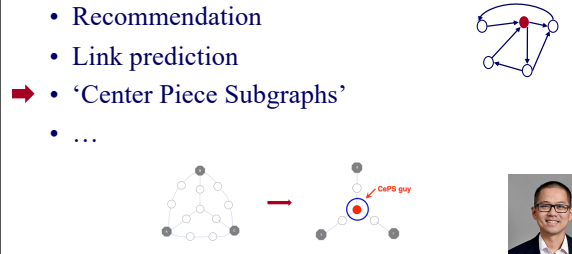
53

53

Carnegie Mellon

Applications of node proximity

- Recommendation
- Link prediction
- ➔ • ‘Center Piece Subgraphs’
- ...



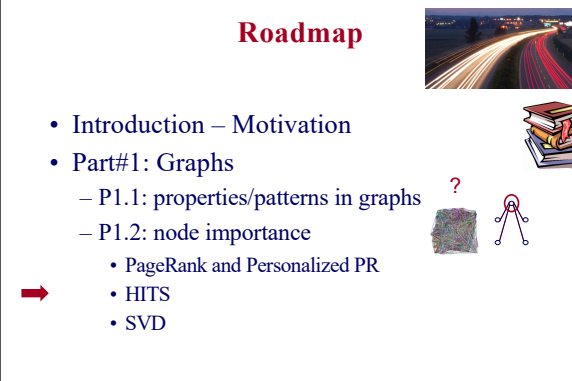
Fast Algorithms for Querying and Mining Large Graphs
Hanghang Tong, PhD dissertation, CMU, 2009. TR: CMU-ML-09-112.

54

Carnegie Mellon

Roadmap

- Introduction – Motivation
- Part#1: Graphs
 - P1.1: properties/patterns in graphs
 - P1.2: node importance
 - PageRank and Personalized PR
 - HITS
 - SVD



Gov. of India Copyright (C) 2019 C. Faloutsos

55

55

Carnegie Mellon

Kleinberg's algo (HITS)



Kleinberg, Jon (1998).
*Authoritative sources in a
hyperlinked environment.*
Proc. 9th ACM-SIAM
Symposium on Discrete
Algorithms.


Gov. of India
Copyright (C) 2019 C. Faloutsos
56

56

Carnegie Mellon

Recall: problem dfn

- Given a graph (eg., web pages containing the desirable query word)
- Q1: Which node is the most important?




Gov. of India
Copyright (C) 2019 C. Faloutsos
57

57

Carnegie Mellon

Why not just PageRank?

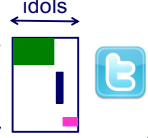
- HITS (and its derivative, SALSA), differentiate between “hubs” and “authorities”
- HITS can help to find the largest community
- (SVD: powerful tool)



idols

↑ ↓

fans



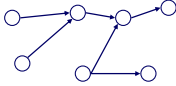
Gov. of India
Copyright (C) 2019 C. Faloutsos
58

58

Carnegie Mellon

Kleinberg's algorithm

- Problem defn: given the web and a query
- find the most 'authoritative' web pages for this query



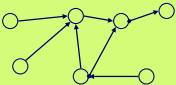
Gov. of India Copyright (C) 2019 C. Faloutsos 59

59

Carnegie Mellon

Problem: PageRank

Given a directed graph, find its most interesting/central node



A node is important, if its parents are important (recursive, but OK!)

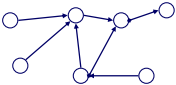
Gov. of India Copyright (C) 2019 C. Faloutsos 60

60

Carnegie Mellon

Problem: ~~PageRank~~ ^{HITS}

Given a directed graph, find its most interesting/central node



A node is important, if its parents are important (recursive, but OK!) ^{``wise``}

AND: A node is ``wise`` if its children are important

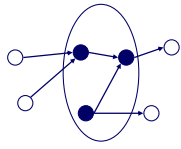
Gov. of India

61

Carnegie Mellon

Kleinberg's algorithm

- Step 0: find nodes with query word(s)
- Step 1: expand by one move forward and backward



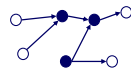
Gov. of India Copyright (C) 2019 C. Faloutsos 62

62

Carnegie Mellon

Kleinberg's algorithm

- on the resulting graph, give high score (= 'authorities') to nodes that many 'wise' nodes point to
- give high wisdom score ('hubs') to nodes that point to good 'authorities'



Gov. of India Copyright (C) 2019 C. Faloutsos 63

63

Carnegie Mellon

Kleinberg's algorithm

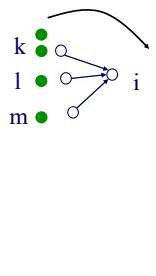
Then:

$$a_i = h_k + h_l + h_m$$

that is

$$a_i = \text{Sum}(h_j) \text{ over all } j \text{ that } (j,i) \text{ edge exists}$$

or

$$\mathbf{a} = \mathbf{A}^T \mathbf{h}$$


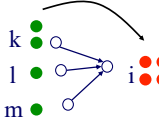
Gov. of India Copyright (C) 2019 C. Faloutsos 64

64

Carnegie Mellon

Kleinberg's algorithm

Then:




$$a_i = h_k + h_l + h_m$$

that is

$$a_i = \text{Sum}(h_j) \text{ over all } j \text{ that } (j,i) \text{ edge exists}$$

or

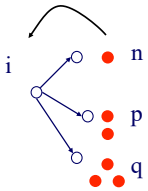
$$\mathbf{a} = \mathbf{A}^T \mathbf{h}$$


Gov. of India Copyright (C) 2019 C. Faloutsos 65

65

Carnegie Mellon

Kleinberg's algorithm




symmetrically, for the 'hubness':

$$h_i = a_n + a_p + a_q$$

that is

$$h_i = \text{Sum}(q_j) \text{ over all } j \text{ that } (i,j) \text{ edge exists}$$

or

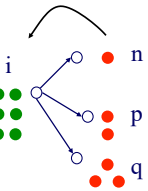
$$\mathbf{h} = \mathbf{A} \mathbf{a}$$


Gov. of India Copyright (C) 2019 C. Faloutsos 66

66

Carnegie Mellon

Kleinberg's algorithm




symmetrically, for the 'hubness':

$$h_i = a_n + a_p + a_q$$

that is

$$h_i = \text{Sum}(q_j) \text{ over all } j \text{ that } (i,j) \text{ edge exists}$$

or

$$\mathbf{h} = \mathbf{A} \mathbf{a}$$


Gov. of India Copyright (C) 2019 C. Faloutsos 67

67

Carnegie Mellon

Kleinberg's algorithm

In conclusion, we want vectors **h** and **a** such that:

$$\mathbf{h} = \mathbf{A} \mathbf{a}$$

$$\mathbf{a} = \mathbf{A}^T \mathbf{h}$$

$$\|\mathbf{h}\| = \|\mathbf{a}\|$$

Gov. of India

Copyright (C) 2019 C. Faloutsos

68

68

Carnegie Mellon

Kleinberg's algorithm

In conclusion, we want vectors **h** and **a** such that:

$$\mathbf{h} = \mathbf{A} \mathbf{a}$$

$$\mathbf{a} = \mathbf{A}^T \mathbf{h}$$

$$\|\mathbf{h}\| = \|\mathbf{a}\|$$

Gov. of India

Copyright (C) 2019 C. Faloutsos

69

69

Carnegie Mellon

Kleinberg's algorithm

In conclusion, we want vectors **h** and **a** such that:

$$\mathbf{h} = \mathbf{A} \mathbf{a}$$

$$\mathbf{a} = \mathbf{A}^T \mathbf{h}$$

$$\|\mathbf{h}\| = \|\mathbf{a}\|$$

Gov. of India

Copyright (C) 2019 C. Faloutsos

70

70

Carnegie Mellon

Kleinberg's algorithm

In conclusion, we want vectors **h** and **a** such that:

$\mathbf{h} = \mathbf{A} \mathbf{a}$
 $\mathbf{a} = \mathbf{A}^T \mathbf{h}$

$\|\cdot\| = \|\cdot\|$

Gov. of India Copyright (C) 2019 C. Faloutsos 71

71

Carnegie Mellon

Kleinberg's algorithm

In short, the solutions to

$$\mathbf{h} = \mathbf{A} \mathbf{a}$$

$$\mathbf{a} = \mathbf{A}^T \mathbf{h}$$

are the left- and right- singular-vectors of the adjacency matrix **A**.

Starting from random **a'** and iterating, we'll eventually converge ... to the vector of strongest singular value.

Dfn: in +2

Gov. of India Copyright (C) 2019 C. Faloutsos 72

72

Carnegie Mellon

Kleinberg's algorithm - results

Eg., for the query 'java':

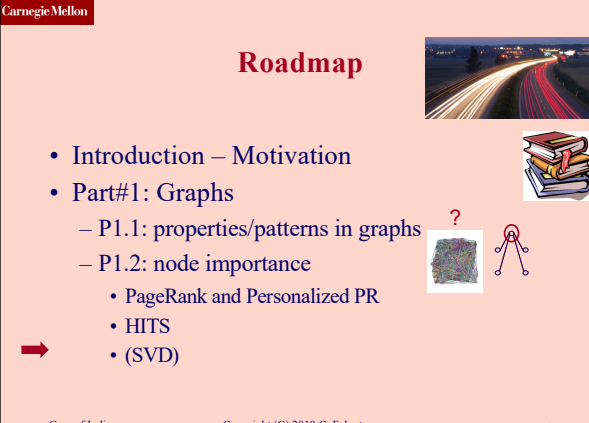
- 0.328 www.gamelan.com
- 0.251 java.sun.com
- 0.190 www.digitalfocus.com ("the java developer")

Gov. of India Copyright (C) 2019 C. Faloutsos 73

73

Carnegie Mellon

Roadmap



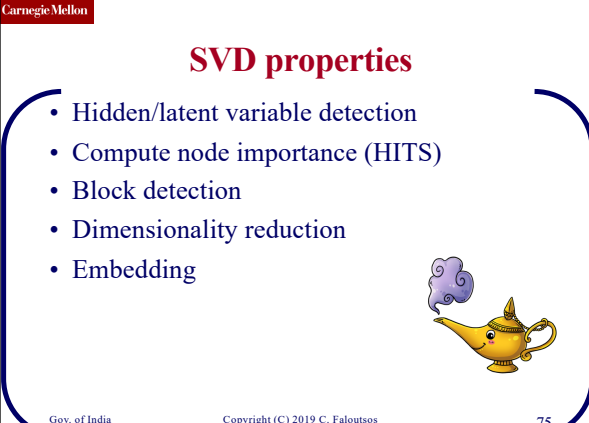
- Introduction – Motivation
- Part#1: Graphs
 - P1.1: properties/patterns in graphs
 - P1.2: node importance
 - PageRank and Personalized PR
 - HITS
 - (SVD)

Gov. of India
Copyright (C) 2019 C. Faloutsos
74

74

Carnegie Mellon

SVD properties



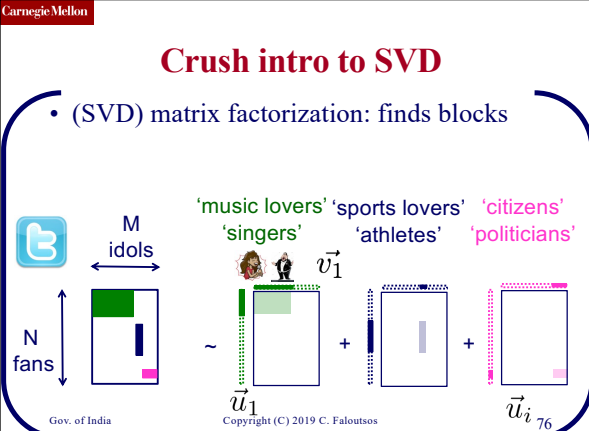
- Hidden/latent variable detection
- Compute node importance (HITS)
- Block detection
- Dimensionality reduction
- Embedding

Gov. of India
Copyright (C) 2019 C. Faloutsos
75

75

Carnegie Mellon

Crush intro to SVD



- (SVD) matrix factorization: finds blocks

Gov. of India
Copyright (C) 2019 C. Faloutsos

76

Crush intro to SVD

- (SVD) matrix factorization: finds blocks

Gov. of India
Copyright (C) 2019 C. Faloutsos

77

Crush intro to SVD

- (SVD) matrix factorization: finds blocks

Gov. of India
Copyright (C) 2019 C. Faloutsos

78

Crush intro to SVD

- (SVD) matrix factorization: finds blocks
- HITS: first singular vector, ie, fixates on largest group

Gov. of India

79

Crash intro to SVD

- Basis for anomaly detection – P1.4
- Basis for tensor/PARAFAC – P2.5

Gov. of India Copyright (C) 2019 C. Faloutsos

80

SVD properties

- ✓ Hidden/latent variable detection
- ✓ Compute node importance (HITS)
- ✓ Block detection
- Dimensionality reduction
- Embedding

Gov. of India Copyright (C) 2019 C. Faloutsos

81

SVD - intuition

SVD: gives best axis to project

- minimum RMS error

Gov. of India Copyright (C) 2019 C. Faloutsos

82

Carnegie Mellon

SVD properties

- ✓ Hidden/latent variable detection
- ✓ Compute node importance (HITS)
- ✓ Block detection
- ✓ Dimensionality reduction
- Embedding

The diagram illustrates the SVD decomposition of a matrix into two orthogonal matrices, U_0 and U_1 , and two diagonal matrices, V_0 and V_1 . A small teapot is also shown.

Gov. of India
Copyright (C) 2019 C. Faloutsos
83

83

Carnegie Mellon

Crush intro to SVD

- SVD compression is a linear **autoencoder**

The diagram shows a matrix of N fans and M idols being compressed into a set of k scores. These scores are then used to reconstruct the original row i (500 dim).

Gov. of India
Copyright (C) 2019 C. Faloutsos
84

84

Carnegie Mellon

Crush intro to SVD

- SVD compression is a linear **autoencoder**

The diagram shows a matrix of N fans and M idols being compressed into a set of k scores. These scores are then used to reconstruct the original row i (500 dim). A yellow arrow points to the matrix, indicating it is approximately $M \times 3$ idol matrix.

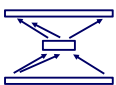

Gov. of India
Copyright (C) 2019 C. Faloutsos
85

85

Carnegie Mellon

SVD properties

- ✓ Hidden/latent variable detection
- ✓ Compute node importance (HITS)
- ✓ Block detection
- ✓ Dimensionality reduction
- ✓ Embedding (linear)
 - SVD is a special case of 'deep neural net'

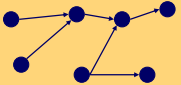
Gov. of India Copyright (C) 2019 C. Faloutsos 86

86

Carnegie Mellon

Node importance - Motivation:

- Given a graph (eg., web pages containing the desirable query word)
- Q1: Which node is the most important?
 - PageRank ($PR = RWR$), HITS, SALSA
- Q2: How close is node 'A' to node 'B'?
 - Personalized P.R. (/SALSA)



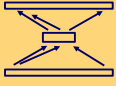

Gov. of India Copyright (C) 2019 C. Faloutsos 87

87

Carnegie Mellon

SVD properties

- ✓ Hidden/latent variable detection
- ✓ Compute node importance (HITS)
- ✓ Block detection
- ✓ Dimensionality reduction
- ✓ Embedding (linear)
 - SVD is a special case of 'deep neural net'

Gov. of India Copyright (C) 2019 C. Faloutsos 88

88

Carnegie Mellon

SVD properties

- ✓ Hidden/latent variable detection
- ✓ Compute node importance
- ✓ Block detection
- ✓ Dimensionality reduction
- ✓ Embedding (matrix factorization)
 - SVD is a special case of 'deep neural net'

Gov. of India
Copyright (C) 2019 C. Faloutsos
89

89

Carnegie Mellon

Roadmap

- Introduction – Motivation
- Part#1: Graphs
 - P1.1: properties/patterns in graphs
 - P1.2: node importance
 - ➔ – P1.3: community detection
 - P1.4: fraud/anomaly detection
 - P1.5: belief propagation

Gov. of India
Copyright (C) 2019 C. Faloutsos
90

90

Carnegie Mellon

Roadmap


- Introduction – Motivation
- Part#1: Graphs
 - P1.1: properties/patterns in graphs
 - P1.2: node importance
 - ➔ – P1.3: community detection
 - P1.4: fraud/anomaly detection
 - P1.5: belief propagation

Gov. of India
Copyright (C) 2019 C. Faloutsos
91




91

Carnegie Mellon

Roadmap



- Introduction – Motivation
- Part#1: Graphs
 - P1.1: properties/patterns in graphs
 - P1.2: node importance
 - P1.3: community detection
 - Algorithm
 - Warning: 'no good cuts'
 - P1.4: fraud/anomaly detection






Gov. of India Copyright (C) 2019 C. Faloutsos 92


92

Carnegie Mellon

Problem



- Given a graph, and k
- Break it into k (disjoint) communities




Gov. of India Copyright (C) 2019 C. Faloutsos P2-93

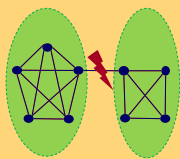
93

Carnegie Mellon

Short answer



- METIS [Karypis, Kumar]



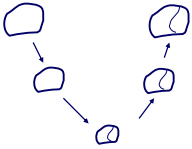
Gov. of India Copyright (C) 2019 C. Faloutsos P2-94

94

Carnegie Mellon

Solution#1: METIS

- Arguably, the best algorithm
- Open source, at
 - <http://glaros.dtc.umn.edu/gkhome/fetch/sw/metis/metis-5.1.0.tar.gz>
- and *many* related papers, at same url
- Main idea:
 - coarsen the graph;
 - partition;
 - un-coarsen




Gov. of India Copyright (C) 2019 C. Faloutsos P2-95

95

Carnegie Mellon

Solution #1: METIS

- G. Karypis and V. Kumar. *METIS 4.0: Unstructured graph partitioning and sparse matrix ordering system*. TR, Dept. of CS, Univ. of Minnesota, 1998.
- <and many extensions>



Gov. of India Copyright (C) 2019 C. Faloutsos P2-96

96

Carnegie Mellon

Solutions #2,3...


- **Fiedler vector** (2nd singular vector of Laplacian).
- **Modularity**: *Community structure in social and biological networks* M. Girvan and M. E. J. Newman, PNAS June 11, 2002. 99 (12) 7821-7826; <https://doi.org/10.1073/pnas.122653799>
- **Co-clustering**: [Dhillon+, KDD'03]
- **Clustering** on the A^2 (square of adjacency matrix) [Zhou, Woodruff, PODS'04]
- **Minimum cut** / maximum flow [Flake+, KDD'00]
-

Gov. of India Copyright (C) 2019 C. Faloutsos P2-97




97

Carnegie Mellon

Roadmap



- Introduction – Motivation
- Part#1: Graphs
 - P1.1: properties/patterns in graphs
 - P1.2: node importance
 - P1.3: community detection
 - Algorithm
 - Warning: ‘no good cuts’
 - P1.4: fraud/anomaly detection

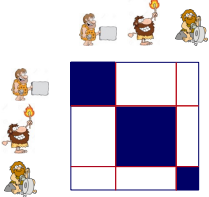
Gov. of India
Copyright (C) 2019 C. Faloutsos
98

98

Carnegie Mellon

A word of caution

- BUT: often, there are **no good cuts**:



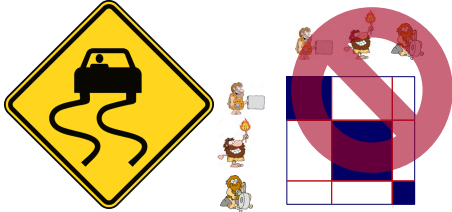
Gov. of India
Copyright (C) 2019 C. Faloutsos
P2-99

99

Carnegie Mellon

A word of caution

- BUT: often, there are **no good cuts**:




Gov. of India
Copyright (C) 2019 C. Faloutsos
P2-100

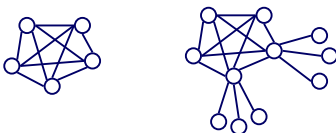
100

Carnegie Mellon

A word of caution



- Maybe there are no good cuts: "jellyfish" shape [Tauro+'01], [Siganos+', '06], strange behavior of cuts [Chakrabarti+'04], [Leskovec+', '08]




Gov. of India Copyright (C) 2019 C. Faloutsos P2-101

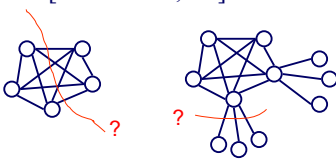
101

Carnegie Mellon

A word of caution



- Maybe there are no good cuts: "jellyfish" shape [Tauro+'01], [Siganos+', '06], strange behavior of cuts [Chakrabarti+', '04], [Leskovec+', '08]


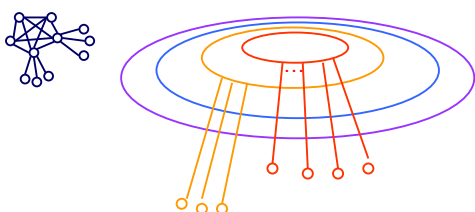


Gov. of India Copyright (C) 2019 C. Faloutsos P2-102

102

Carnegie Mellon

R1: Jellyfish model [Tauro+]

A Simple Conceptual Model for the Internet Topology, L. Tauro, C. Palmer, G. Siganos, M. Faloutsos, Global Internet, November 25-29, 2001

Jellyfish: A Conceptual Model for the AS Internet Topology G. Siganos, Sudhir L. Tauro, M. Faloutsos, J. of Communications and Networks, Vol. 8, No. 3, pp 339-350, Sept. 2006.

103

Carnegie Mellon

R1: Jellyfish model [Tauro+]

A Simple Conceptual Model for the Internet Topology, L. Tauro, C. Palmer, G. Siganos, M. Faloutsos, Global Internet, November 25-29, 2001

Jellyfish: A Conceptual Model for the AS Internet Topology G. Siganos, Sudhir L. Tauro, M. Faloutsos, J. of Communications and Networks, Vol. 8, No. 3, pp 339-350, Sept. 2006.

104

Carnegie Mellon

R1: Jellyfish model [Tauro+]

A Simple Conceptual Model for the Internet Topology, L. Tauro, C. Palmer, G. Siganos, M. Faloutsos, Global Internet, November 25-29, 2001

Jellyfish: A Conceptual Model for the AS Internet Topology G. Siganos, Sudhir L. Tauro, M. Faloutsos, J. of Communications and Networks, Vol. 8, No. 3, pp 339-350, Sept. 2006.

105

Carnegie Mellon

R2: 'Familiar strangers'

- Bipartite graph ('heterophily')

Gov. of India

Copyright (C) 2019 C. Faloutsos

106

106

Carnegie Mellon

R3: "Core-periphery"

- Bipartite graph + clique

Gov. of India

Copyright (C) 2019 C. Faloutsos

107

107

Carnegie Mellon

Strange behavior of min cuts

- 'negative dimensionality' (!)

Clickstream graph

NetMine: New Mining Tools for Large Graphs, by D. Chakrabarti, Y. Zhan, D. Blandford, C. Faloutsos and G. Blelloch, in the SDM 2004 Workshop on Link Analysis, Counter-terrorism and Privacy

Statistical Properties of Community Structure in Large Social and Information Networks, J. Leskovec, K. Lang, A. Dasgupta, M. Mahoney. WWW 2008.

108

Carnegie Mellon

Strange behavior of min cuts

- 'negative dimensionality' (!)

Clickstream graph

NetMine: New Mining Tools for Large Graphs, by D. Chakrabarti, Y. Zhan, D. Blandford, C. Faloutsos and G. Blelloch, in the SDM 2004 Workshop on Link Analysis, Counter-terrorism and Privacy



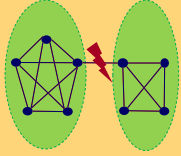
Statistical Properties of Community Structure in Large Social and Information Networks, J. Leskovec, K. Lang, A. Dasgupta, M. Mahoney. WWW 2008.

109

Carnegie Mellon

Short answer

- METIS [Karypis, Kumar]
- (but: maybe NO good cuts exist!)




Gov. of India Copyright (C) 2019 C. Faloutsos P2-110

110

Carnegie Mellon

Roadmap

- Introduction – Motivation
- Part#1: Graphs
 - P1.1: properties/patterns in graphs
 - P1.2: node importance
 - P1.3: community detection
 - ➔ – P1.4: fraud/anomaly detection
 - P1.5: belief propagation

Gov. of India Copyright (C) 2019 C. Faloutsos 111

111
