15-826: Multimedia Databases and Data Mining

Lecture #27: Graph mining - Generators & tools

Christos Faloutsos

Must-read material (1 of 2)

Fully Automatic Cross-Associations,
by D. Chakrabarti, S. Papadimitriou, D. Modha and C. Faloutsos, in KDD 2004 (pages 79-88), Washington, USA
Must-read material (2 of 2)


Main outline

• Introduction
• Indexing
• Mining
  – Graphs – patterns
  – Graphs – generators and tools
  – Association rules
  – …
Detailed outline

- Graphs – generators
  - Erdos-Renyi
  - Other generators
  - Kronecker
- Graphs - tools

Problem

- Q: How to generate realistic graphs?
Answer:

- Q: How to generate realistic graphs?
- A: self-similarity – ‘Kronecker’ graphs

Generators

- How to generate random, realistic graphs?
  - Erdos-Renyi model: beautiful, but unrealistic
  - degree-based generators
  - process-based generators
  - recursive/self-similar generators
Erdos-Renyi

- random graph – 100 nodes, avg degree = 2
- Fascinating properties (phase transition)
- But: unrealistic (Poisson degree distribution != power law)

E-R model & Phase transition

- vary avg degree D
- watch \( P_c = \) Prob( there is a giant connected component)
- How do you expect it to be?

\[
P_c \quad 0 \quad 1
\]

\[
D \quad ??
\]
E-R model & Phase transition

- vary avg degree D
- watch $P_c = \text{Prob( there is a giant connected component)}$
- How do you expect it to be?

Degree-based

- Figure out the degree distribution (eg., ‘Zipf’)
- Assign degrees to nodes
- Put edges, so that they match the original degree distribution
Process-based

- Barabasi; Barabasi-Albert: Preferential attachment -> power-law tails!
  - ‘rich get richer’
- [Kumar+]: preferential attachment + mimick
  - Create ‘communities’

Process-based (cont’d)

- [Fabrikant+, ‘02]: H.O.T.: connect to closest, high connectivity neighbor
- [Pennock+, ‘02]: Winner does NOT take all
Detailed outline

• Graphs – generators
  – Erdos-Renyi
  – Other generators
  – Kronecker
• Graphs - tools

Recursive generators

• (RMAT [Chakrabarti+, ’04])
• Kronecker product
Wish list for a generator:

- Power-law-tail in- and out-degrees
- Power-law-tail scree plots
- shrinking/constant diameter
- Densification Power Law
- communities-within-communities

Q: how to achieve all of them?
A: Kronecker matrix product [Leskovec+05b]

Graph gen.: Problem dfn

- Given a growing graph with count of nodes $N_1$, $N_2$, ...
- Generate a realistic sequence of graphs that will obey all the patterns
  - Static Patterns
    - S1 Power Law Degree Distribution
    - S2 Power Law eigenvalue and eigenvector distribution
    - Small Diameter
  - Dynamic Patterns
    - T2 Growth Power Law (2x nodes; 3x edges)
    - T1 Shrinking/Stabilizing Diameters
Graph Patterns

Power Laws
Count vs Indegree
Count vs Outdegree
Eigenvalue vs Rank

How to match all these properties (+ small diameters, etc)?

Hint: self-similarity

- A: RMAT/Kronecker generators
  - With self-similarity, we get all power-laws, automatically,
  - And small/shrinking diameter
  - And ‘no good cuts’

**R-MAT: A Recursive Model for Graph Mining**, by D. Chakrabarti, Y. Zhan and C. Faloutsos, SDM 2004, Orlando, Florida, USA

Kronecker Graphs

adjacency matrix}

\[
\begin{pmatrix}
1 & 1 & 0 \\
1 & 1 & 1 \\
0 & 1 & 1 \\
\end{pmatrix}
\]

$G'_1$

Adjacency matrix
**Kronecker Graphs**

![Diagram of Kronecker Graphs]

Intermediate stage

\[
\begin{array}{ccc}
1 & 1 & 0 \\
1 & 1 & 1 \\
0 & 1 & 1 \\
\end{array}
\]

Adjacency matrix

\[
G_1 = \begin{bmatrix}
G_1 & G_1 & 0 \\
G_1 & G_1 & G_1 \\
0 & G_1 & G_1 \\
\end{bmatrix}
\]

Adjacency matrix

\[G_2 = G_1 \otimes G_1\]

**Kronecker product**

![Diagram of Kronecker product]

Adjacency matrix

\[
\begin{array}{ccc}
1 & 1 & 0 \\
1 & 1 & 1 \\
0 & 1 & 1 \\
\end{array}
\]

\[
\begin{array}{ccc}
G & G & 0 \\
G & G & G \\
0 & G & G \\
\end{array}
\]

\[N^* N \quad N^{**4}\]
Kronecker Graphs

- Continuing multiplying with $G_1$ we obtain $G_4$ and so on …

$G_4$ adjacency matrix
Kronecker Graphs

- Continuing multiplying with $G_1$ we obtain $G_4$ and so on …

Holes within holes; Communities within communities

$G_4$ adjacency matrix

Copyright (c) 2019 C. Faloutsos
Properties:

- We can PROVE that
  - Degree distribution is multinomial ~ power law
  - Diameter: constant
  - Eigenvalue distribution: multinomial
  - First eigenvector: multinomial

Problem Definition

- Given a growing graph with nodes $N_1, N_2, ...$
- Generate a realistic sequence of graphs that will obey all the patterns
  - Static Patterns
    - ✔️ Power Law Degree Distribution
    - ✔️ Power Law eigenvalue and eigenvector distribution
    - ✔️ Small Diameter
  - Dynamic Patterns
    - ✔️ Growth Power Law
    - ✔️ Shrinking/Stabilizing Diameters
- First generator for which we can prove all these properties
Impact: Graph500

- Based on RMAT (= 2x2 Kronecker)
- Standard for graph benchmarks
- http://www.graph500.org/
- Competitions 2x year, with all major entities: LLNL, Argonne, ITC-U. Tokyo, Riken, ORNL, Sandia, PSC, …

To iterate is human, to recurse is divine

R-MAT: A Recursive Model for Graph Mining,
by D. Chakrabarti, Y. Zhan and C. Faloutsos,
SDM 2004, Orlando, Florida, USA

Conclusions - Generators

- Erdos-Renyi: phase transition
- Preferential attachment (Barabasi)
  - Power-law-tail in degree distribution
- Variations
- Recursion – Kronecker graphs
  - Numerous power-laws, + small diameters
Answer:

- Q: How to generate realistic graphs?
- A: self-similarity – ‘Kronecker’ graphs

Resources

Generators:
- Kronecker (christos@cs.cmu.edu)
- BRITE  http://www.cs.bu.edu/brite/
- INET: http://topology.eecs.umich.edu/inet
Other resources

Visualization - graph algo’s:
• Graphviz: http://www.graphviz.org/
• pajek: http://vlado.fmf.uni-lj.si/pub/networks/pajek/

Kevin Bacon web site:
http://www.cs.virginia.edu/oracle/

References

References, cont’d

• [Broder+, '00] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the web, WWW, 2000

References, cont’d

References, cont’d


References, cont’d

• [Leskovec+05b] Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos

References, cont’d

References, cont’d


Graph mining: tools

Main outline

- Introduction
- Indexing
- Mining
  - Graphs – patterns
  - Graphs – generators and tools
  - Association rules
  - …
Detailed outline

- Graphs – generators
- Graphs – tools
  - Community detection / graph partitioning
    - Algo’s
    - Observation: ‘no good cuts’
  - Node proximity – personalized RWR
  - Influence/virus propagation & immunization
  - ‘Belief Propagation’ & fraud detection
  - Anomaly detection

Problem

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

• METIS [Karypis, Kumar]

Problem

• Given a graph, and $k$
• Break it into $k$ (disjoint) communities
**Problem**

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

\[ k = 2 \]

**Solution #1: METIS**

- Arguably, the best algorithm
- Open source, at
  - \( \text{http://www.cs.umn.edu/~metis} \)
- and *many* related papers, at same url
- Main idea:
  - coarsen the graph;
  - partition;
  - un-coarsen
Solution #1: METIS

- <and many extensions>

Solution #2

(problem: hard clustering, $k$ pieces)
Spectral partitioning:
- Consider the $2^{nd}$ smallest eigenvector of the (normalized) Laplacian
Solutions #3, …

Many more ideas:
- Clustering on the $A^2$ (square of adjacency matrix) [Zhou, Woodruff, PODS’04]
- Minimum cut / maximum flow [Flake+, KDD’00]
- …

Detailed outline

- Motivation
- Hard clustering – $k$ pieces
- Hard co-clustering – $(k,l)$ pieces
- Hard clustering – optimal # pieces
- Soft clustering – matrix decompositions
- Observations
**Problem definition**

- Given a bi-partite graph, and $k$, $l$
- Divide it into $k$ row groups and $l$ row groups
- (Also applicable to uni-partite graph)

**Co-clustering**

- Given data matrix and the number of row and column groups $k$ and $l$
- Simultaneously
  - Cluster rows into $k$ disjoint groups
  - Cluster columns into $l$ disjoint groups
Co-clustering

• Let $X$ and $Y$ be discrete random variables
  – $X$ and $Y$ take values in $\{1, 2, \ldots, m\}$ and $\{1, 2, \ldots, n\}$
  – $p(X, Y)$ denotes the joint probability distribution—if not known, it is often estimated based on co-occurrence data
  – Application areas: text mining, market-basket analysis, analysis of browsing behavior, etc.

• Key Obstacles in Clustering Contingency Tables
  – High Dimensionality, Sparsity, Noise
  – Need for robust and scalable algorithms

Reference:
1. Dhillon et al. Information-Theoretic Co-clustering, KDD’03
Co-clustering

Observations

• uses KL divergence, instead of L2
• the middle matrix is **not** diagonal
  – Like in the Tucker tensor decomposition
• s/w at:
  www.cs.utexas.edu/users/dml/Software/cocluster.html
Detailed outline

- Motivation
- Hard clustering – k pieces
- Hard co-clustering – (k,l) pieces
- Hard clustering – optimal # pieces
- Soft clustering – matrix decompositions
- Observations

Problem with Information Theoretic Co-clustering

- Number of row and column groups must be specified

Desiderata:

- ✔ Simultaneously discover row and column groups
- ✗ Fully Automatic: No “magic numbers”
- ✔ Scalable to large graphs
Graph partitioning

- Documents x terms
- Customers x products
- Users x web-sites

Q: HOW MANY PIECES?
Graph partitioning

- Documents x terms
- Customers x products
- Users x web-sites

Q: HOW MANY PIECES?

A: MDL/ compression

Cross-association

Desiderata:
- Simultaneously discover row and column groups
- Fully Automatic: No “magic numbers”
- Scalable to large matrices

Reference:
1. Chakrabarti et al. Fully Automatic Cross-Associations, KDD’04
What makes a cross-association “good”? What makes a cross-association “good”?

versus

Column groups

Column groups

Why is this better?

simpler; easier to describe
easier to compress!

Why is this better?
What makes a cross-association “good”?

Problem definition: given an encoding scheme
• decide on the # of col. and row groups $k$ and $l$
• and reorder rows and columns,
• to achieve best compression

Main Idea

Total Encoding Cost = $\sum_i \text{size}_i \cdot H(x_i) + \text{Cost of describing cross-associations}$

Minimize the total cost (# bits)
for lossless compression
Algorithm

Experiments

“CLASSIC”
- 3,893 documents
- 4,303 words
- 176,347 “dots”

Combination of 3 sources:
- MEDLINE (medical)
- CISI (info. retrieval)
- CRANFIELD (aerodynamics)
“CLASSIC” graph of documents & words: k=15, l=19

Experiments

insipidus, alveolar, aortic, death, prognosis, intravenous

tissue, patient

blood, disease, clinical, shape, nose, leading,
cell, tissue, patient

MEDLINE (medical)
Experiments

“CLASSIC” graph of documents & words:

k=15, l=19

MEDLINE
(medical)

CISI
(Information Retrieval)

CRANFIELD
(aerodynamics)
Experiments

“CLASSIC” graph of documents & words: k=15, l=19

Algorithm

Code for cross-associations (matlab):


Variations and extensions:
• ‘Autopart’ [Chakrabarti, PKDD’04]
• www.cs.cmu.edu/~deepay
Algorithm

- Hadoop implementation [ICDM’08]

Detailed outline

- Motivation
- Hard clustering – $k$ pieces
- Hard co-clustering – $(k,l)$ pieces
- Hard clustering – optimal # pieces
- (Soft clustering – matrix decompositions
  – PCA, ICA, non-negative matrix factorization, …)
- Observations
Detailed outline

- Motivation
- Hard clustering – $k$ pieces
- Hard co-clustering – $(k,l)$ pieces
- Hard clustering – optimal # pieces
- (Soft clustering)
- Observations

Observation #1

- Skewed degree distributions – there are nodes with huge degree (> $O(10^4)$, in facebook/linkedIn popularity contests!)
- TRAP: ‘find all pairs of nodes, within 2 steps from each other’
Observation #2

• TRAP: *shortest-path between two nodes*
• (cheat: look for 2, at most 3-step paths)
• Why:
  – If they are close (within 2-3 steps): solved
  – If not, after ~6 steps, you’ll have ~ the whole graph, and the path won’t be very meaningful, anyway.

Observation #3

• Maybe there are no good cuts: ```jellyfish``` shape [Tauro+’01], [Siganos+,’06], strange behavior of cuts [Chakrabarti+’04], [Leskovec+,’08]
Observation #3

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

Jellyfish model [Tauro+]


Strange behavior of min cuts

• ‘negative dimensionality’ (!)


“Min-cut” plot

• Do min-cuts recursively.

Mincut size = sqrt(N)

log (mincut-size / #edges)

log (# edges)

N nodes
“Min-cut” plot

• Do min-cuts recursively.

New min-cut

\[ \log(\text{mincut-size} / \#\text{edges}) \]

N nodes

For a d-dimensional grid, the slope is \(-1/d\)

Slope = -0.5
“Min-cut” plot

For a d-dimensional graph, the slope is \(-1/d\)

For a random graph, the slope is 0

• What does it look like for a real-world graph?
Experiments

• Datasets:
  – Google Web Graph: 916,428 nodes and 5,105,039 edges
  – Lucent Router Graph: Undirected graph of network routers from [website link], 112,969 nodes and 181,639 edges
  – User Website Clickstream Graph: 222,704 nodes and 952,580 edges


Experiments

• Used the METIS algorithm [Karypis, Kumar, 1995]

- Google Web graph
- Values along the y-axis are averaged
- We observe a “lip” for large edges
- Slope of -0.4, corresponds to a 2.5-dimensional grid!
Experiments

- Used the METIS algorithm [Karypis, Kumar, 1995]

\[
\text{log (mincut-size / #edges)} - 0.57; -0.45
\]

- Similarly, for
  - Lucent routers
  - Clickstream

Conclusions – Practitioner’s guide

- Hard clustering – \(k\) pieces
- Hard co-clustering – \((k,l)\) pieces
- Hard clustering – optimal # pieces
- Observations

‘jellyfish’:
Maybe, there are no good cuts
Short answer

- METIS [Karypis, Kumar]

But: maybe there are NO good cuts!