Fast Influence-based Coarsening for Large Networks

Manish Purohit^, B. Aditya Prakash*,
Chanhyun Kang^, Yao Zhang*, V S Subrahmanian^

*Virginia Tech    ^University of Maryland

KDD, New York City
August 26, 2014
Networks are getting huge!

Flickr (friendship network): 87 million users and 8 billion photos until 2013

Amazon (friendship network): 237 million accounts until 2013

Facebook (friendship network): 829 million daily active users on average in June 2014

Twitter (follower network): 271 million monthly active users
Need for fast analysis

- Ever growing list of applications of network effects
- Viral Marketing
- Immunization
- Information Diffusion
- ...

However, scaling up traditional algorithms up to millions of nodes is hard 😞
How to handle large-scale networks

- **Approaches**
  - Use faster / simpler algorithms
  - Perform analysis locally
    - i.e., divide the large network into smaller subgraphs
  - **Zoom-out** the network to obtain a smaller representation of the network

this paper
Bird’s eye view of a network
Bird’s eye view of a network

- “Zoom-out” of the graph to get a quick picture

Called “coarsen” in this paper
Outline

• Motivation
• Challenges
• Problem Definition
• Our Proposed Method
• Experiments
• Applications
• Conclusion
Challenges

• C1: How do we maintain diffusive characteristics when coarsening networks?

• C2: How do we merge node to get the coarse network?

• C3: how do we find the best node to merge fast?
C1: Information Diffusion

- Cascading behavior in networks

```
Blog network

Source: [McGlohon et. al., SDM2007]
```

**Diffusion** is graph induced by a time ordered propagation of information (edges)
C1: Model information diffusion

- Information spreads over networks
  - e.g.: rumor/meme spreads over Twitter following network
- Independent cascade model (IC) [Kempe+, KDD03]
  - Weights $p_{ij}$: propagation prob. from i to j
  - Each node has only one chance to infect its neighbors

Meme spreading
C1: Diffusive characteristics

- First eigenvalue \( \lambda_1 \) (of adjacency matrix) is enough for most diffusion models. (Prakash et al. [ICDM’12])

\[ \lambda_1 \text{ is the epidemic threshold} \]

Increasing \( \lambda_1 \), Increasing vulnerability

(a) Chain(\( \lambda_1 = 1.73 \))  (b) Star(\( \lambda_1 = 2 \))  (c) Clique(\( \lambda_1 = 4 \))
C1: maintain diffusive characteristics

- Goal: maintain the diffusive characteristics of the original network in the coarsened network?

Make the coarsened network has the least change in the first eigenvalue

Original network

Coarsened network
C2: How to merge nodes

- Goal: Merge nodes of graph G to get the coarsened graph that “approximates” G with respect to diffusion

Influence from d to b: 0.5
Influence from d to a: 0.25
Average: 0.375

Merge b and a can get the least change of $\lambda_1$

Is this correct?
C2: How to merge nodes

- In general:

\[ \frac{a_x^i (1 + \beta_1) + b_y^i (1 + \beta_2)}{4} \]

\[ \frac{a_z^o (1 + \beta_2) + b_z^o (1 + \beta_1)}{4} \]
C3: which nodes to merge

- **Goal:**
  - Find the best nodes to merge
  - Fast, scalable to large network

Original network

Coarsened network

Talk about it later
Outline

• Motivation
• Challenges
• Problem Definition
• Our Proposed Method
• Experiments
• Applications
• Conclusion
Problem Definition

Graph Coarsening Problem (GCP)

*Given*: large graph $G(V, E)$, and reduction factor $\alpha$

*Find*: the best set of edges to merge

*Such that*: $|\lambda_G - \lambda_H|$ is minimized

• (i.e. $H$ is the coarsened graph with the least change in the first eigenvalue)
Naive Greedy Heuristic

Step:

- Score every edge by the change in eigenvalue
- Greedily choose the edge \((a,b)\) with the least score, and merge \((a,b)\)
- Re-evaluate the scores of every edge and repeat

- **Too slow!** \(O(m^2)\) time to score all edges
- Lose time benefits of analyzing the smaller graph

Purohit, Prakash, Kang, Zhang, Subrahmanian 2014
Outline

- Motivation
- Problem Definition
- Challenges
- **Our Proposed Method**
  - CoarseNet
- Experiments
- Applications
- Conclusion
CoarseNet: idea

• Can we approximate the edge scores faster?
  • Yes!

• Use matrix perturbation arguments to estimate (up to first order terms) the score of an edge in constant time!

• Score all edges in $O(m)$ time
  • Naive Heuristic: $O(m^2)$ time
CoarseNet: details

- Corollary 5.1: Given the first eigenvalue $\lambda$, and corresponding eigenvectors $u, v$, the score of a node pair $\text{score}(a, b)$ can be approximated in constant time.  

$\lambda$ (a,b) is a node-pair

We want to characterize the change of $\lambda$ after coarsening
\[ \Delta \lambda_{(a,b)} = \frac{-\lambda (u_a v_a + u_b v_b) + v_a \tilde{u}^T \tilde{c} + \beta_2 u_a v_b + \beta_1 u_b v_a}{\tilde{v}^T \tilde{u} - (u_a v_a + u_b v_b)} \]

- left eigenvector
- right eigenvector
- weight of (a,b)
- weight of (b,a)

The out-adjacency vector of merged node c

\[ u = \lambda \cdot u \]

See paper for details
CoarseNet: Complete algorithm

- **Step**
  1: compute scores for all edge pairs
  2: Merge nodes with smallest score
  3: Goto step 1 until αn nodes left

Original Network (weight=0.5)

Assigning scores

Coarsened Network

Merging edges

Purohit, Prakash, Kang, Zhang, Subrahmanian 2014
CoarseNet: running time

• Running time: $O(mln(m)+ann_\theta)$
  • $m$: number of edges
  • $n$: number of nodes
  • $n_\theta$: the maximum degree of any vertex during the merging process
Outline

- Motivation
- Challenges
- Problem Definition
- Our Proposed Method
- Experiments
- Applications
- Conclusion
How do we perform?

The first eigenvalue gets preserved well up to large coarsening factors!

(See more results in the paper)
Scalability w.r.t Reduction Factor ($\alpha$)

Amazon (334,863 vertices)  
DBLP (511,163 vertices)

Scales linearly with the desired reduction factor

(See more results in the paper)
Scalability w.r.t Graph Size ($n$)

We extracted 6 connected components (with 500K to 1M vertices in steps of 100K) of the Flickr network.

Scales linearly with the number of nodes.
Outline

• Motivation
• Challenges
• Problem Definition
• Our Proposed Method
• Experiments
• Applications
• Conclusion
Application 1: Influence Maximization

- How to market well?
  - Convince a subset of individuals to adopt a new product
  - Then, trigger a large cascade of further adoptions
- Influence maximization problem
  - [Kempe et. al, KDD03]
  - Find the best set of seeds in a network to achieve highest diffusion

Who is the most influential person?
Application 1: Influence Maximization

- Our fast algorithm CSPIN:
  
  Step 1: Coarsen the large social network using CoarsenNet
  
  Step 2: Solve influence maximization on the coarsened network
  
  Step 3: Randomly select one node from each selected “supernode”

We call it CSPIN
Quality of CSPIN

• We use and compare against the fast and popular PMIA algorithm (Chen et al. [KDD’07])

We obtain influence spread as good as by PMIA
Quality of CSPIN w.r.t $\alpha$

We can **merge up to 95%** of the vertices are merged without significantly affecting the influence spread!
Scalability w.r.t number of seeds

Portland (1.5 million vertices)

Finds good solutions in minutes instead of hours!
(See more results in the paper)
Application 2: Diffusion Characterization

- **Goal:** use Graph Coarsening to understand information cascades
- **Dataset:** Flixster
  - a friendship network with movie ratings
  - Cascade: the same movie rating from friends
- **Methodology**
  - coarsen the network using CoarseNet with the reduction factor $\alpha=0.5$
  - study the formed groups (supernodes)
**Diffusion observation**

Observation 1: a very large fraction of movies propagate in a small number of groups

Observation 2: a multi-modal distribution

**Stats:**
- 1891 groups
- mean group size: 16.6
- the largest group: 22061 nodes (roughly 40% of nodes)

(See more results in the paper)
Outline

• Motivation
• Challenges
• Problem Definition
• Our Proposed Method
• Experiments
• Applications
• Conclusion
Conclusion

Graph Coarsening Problem
• Given: a large graph and the reduction factor
• Find: "best" nodes to coarsen

CoarseNet
• estimate edge score in constant time
• Sub-quadratic

Applications
• Influence Maximization
• Diffusion Characterization

Original network

Coarsened network

Purohit, Prakash, Kang, Zhang, Subrahmanian 2014
Any Questions?

• Code at:
  http://www.cs.vt.edu/~badityap/

Funding: