



Commute Time and Related Methods for Seed-Set Expansion

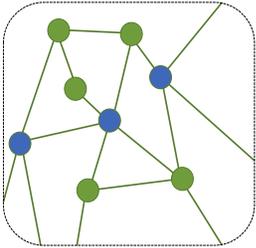
Victor Minden¹, Van Emden Henson², Geoff Sanders²

¹Stanford University
Institute for Computational and Mathematical Engineering
Stanford, CA

²Lawrence Livermore National Laboratory
Center for Applied Scientific Computing
Livermore, CA

Introduction

Problem: Given a set of **seed nodes** in an undirected, unweighted graph, find a **community** of nodes well-connected to the seeds



Difficulties: Nodes with abnormally high (hubs) or low (leaves) degree can mask underlying community structure.

Approach: Rank nodes based on the distance or affinity information in the spectrum of a matrix associated with the graph.

Applications



Recommendation
Seeds are friends, watched videos, etc.

Cybersecurity
Seeds are infected hosts, known spammers, etc.



Mathematical Methods

$$\left. \begin{aligned} L &= D - A = \Phi \Lambda \Phi^T \\ \hat{L} &= D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = \hat{\Phi} \hat{\Lambda} \hat{\Phi}^T \end{aligned} \right\} \text{Eigenvale decomposition}$$

Hitting Time

$$HT(v_i \rightarrow v_j) = \text{Expected length of a random walk from } v_i \text{ to } v_j \\ = r - \sum_{n=1}^{N-1} \frac{1}{\lambda_n} \frac{\hat{\phi}_n^{(i)} \hat{\phi}_n^{(j)}}{\sqrt{d_i} \sqrt{d_j}}$$

Commute Time

$$CT(v_i \leftrightarrow v_j) = HT(v_i \rightarrow v_j) + HT(v_j \rightarrow v_i) \\ = r \sum_{n=1}^{N-1} \frac{1}{\lambda_n} (\hat{\phi}_n^{(i)} - \hat{\phi}_n^{(j)})^2 = \sum_{n=1}^{N-1} \frac{1}{\lambda_n} \left(\frac{\hat{\phi}_n^{(i)}}{\sqrt{d_i}} - \frac{\hat{\phi}_n^{(j)}}{\sqrt{d_j}} \right)^2$$

Centered Commute Time

$$CCT(v_i \leftrightarrow v_j) = CT(v_i \leftrightarrow v_j) - \mathbb{E}_k [CT(v_i \leftrightarrow v_k)] - \mathbb{E}_k [CT(v_j \leftrightarrow v_k)] \\ = r - \sum_{n=1}^{N-1} \frac{1}{\lambda_n} \phi_n^{(i)} \phi_n^{(j)}$$

Regularization

Can reduce effects of “noisy” edges by enforcing *smoothness* of ranking vector with respect to some gradient – or regularize seed-set indicator vector, e_S , directly.

$$\|\nabla \mathbf{x}\|_2^2 = \sum_{v_j \in \mathcal{N}(v_i)} (\mathbf{x}^{(i)} - \mathbf{x}^{(j)})^2 \\ = \langle \mathbf{x}, L \mathbf{x} \rangle$$

$$\|\hat{\nabla} \mathbf{x}\|_2^2 = \sum_{v_j \in \mathcal{N}(v_i)} \left(\frac{\mathbf{x}^{(i)}}{\sqrt{d_i}} - \frac{\mathbf{x}^{(j)}}{\sqrt{d_j}} \right)^2 \\ = \langle \mathbf{x}, \hat{L} \mathbf{x} \rangle$$

Spectral Truncation

$$\frac{1}{\lambda_n} \rightarrow \begin{cases} 1/\lambda_n & n \leq k \\ 0 & n > k \end{cases}$$

Tikhonov-type

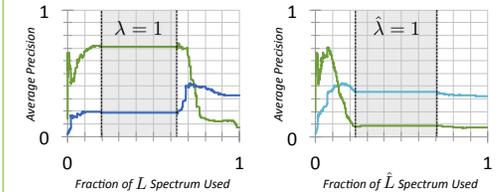
$$\frac{1}{\lambda_n} \rightarrow \frac{1}{1 + \alpha \lambda_n}$$

Simulation Results

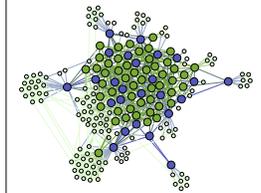
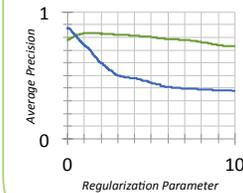
Algorithm

- Generate BTER graph
- Insert Erdős-Rényi community
- For each node, find mean distance from seed nodes
- Score ranking vector using average precision

Spectral Truncation Smoothing for HT , CT , and CCT



Tikhonov-type Smoothing of e_S with L and \hat{L}



Plots constructed from BTER graph with $O(4000)$ nodes and a 75-node inserted community with 10% edge density. 25 seeds were used.

Observations

- Truncation plots show large region of eigenvectors that don't affect average precision significantly.
- Truncation can improve CT and CCT scores, but no clear way to choose number of eigenvectors to use.
- Average precision varies smoothly with respect to choice of Tikhonov-type regularization parameter.
- Need to test on additional community models.