SPECTRAL CLUSTERING OF LARGE NETWORKS
A. Fender, N. Emad, S. Petiton, M. Naumov

May 8th, 2017
Agenda

Introduction
Laplacian
Modularity
Conclusions
THE CLUSTERING PROBLEM

Example: detect relevant groups based on frequent co-purchasing on Amazon.com

Data: V. Krebs. 2004
THE CLUSTERING PROBLEM

Pink  Liberal
Yellow Neutral
Green Conservative

Data: V. Krebs. 2004
CLUSTERING ALGORITHMS

• **Spectral**
  Build a matrix, solve an eigenvalue problem, use eigenvectors for clustering

• **Hierarchical / Agglomerative**
  Build a hierarchy (fine to coarse), partition coarse, propagate results back to fine level

• **Local refinements**
  Switch one node at a time
Laplacian
RATIO CUT COST FUNCTION

Objective function:

\[
Cost = \sum_{i=1}^{p} \frac{|\partial E_i|}{|V_i|}
\]

where \(|\partial E_i|\): # of edges cut and \(|V_i|\): # of nodes in i-th partition

A compromise between small edge-cut and balanced partitions

\[
Cost = \frac{2}{2} + \frac{2}{3} = \frac{5}{3}
\]

GRAPH LAPLACIAN

\[ L = D - A \]

D: degree matrix
A: adjacency matrix

GRAPH LAPLACIAN

For a vector $x$ with elements that are 0 or 1:

\[
\begin{align*}
|\partial E_1| &= 1 - 1 - 1 - 1 - 1 = 2 \\
|V_1| &= x^T x = 2
\end{align*}
\]
MINIMIZATION PROBLEM

\[
\min_{V_i} \sum_{i=1}^{p} \frac{|\partial E_i|}{|V_i|}
\]

\[
\min_{x_i} \sum_{i=1}^{p} \frac{x_i^T L x_i}{x_i^T x_i}
\]

where \(x_i \in \{0,1\}^n\) and \(x_i \perp x_j\)

Next step
Relax requirements on \(x\),
and let \(x_i\) take real values

K-MEANS POINTS CLUSTERING

Lloyd’s Algorithm:
- Select centroids
- Compute distance of points to centroids
- Assign points to the closest centroid
- Recompute centroid position

EDGE CUT MINIMIZATION PIPELINE

Graph Preprocessing

Laplacian

Eigensolver

Points Clustering
SPECTRAL EDGE CUT MINIMIZATION

80% hit rate

Balanced cut minimization

Ground truth
Modularity
Measures the difference between how well vertices are assigned into clusters for the current graph $G = (V, E)$ versus a random graph $R = (V, F)$.

\[
Q = \frac{1}{2\omega} \sum_i \sum_j (w_{ij} - \frac{v_i v_j}{2\omega}) \delta_{c(i)c(j)}
\]

for some assignment $c(.)$ into clusters.

A. Fender, N. Emad, S. Petiton, M. Naumov. “Parallel Modularity Clustering.” ICCS, 2017
MODULARITY MATRIX

Let matrix

\[ B = A - \frac{1}{2\omega} vv^T \]

then modularity

\[ Q = \frac{1}{2\omega} Tr(X^T BX) \]

where \( Tr(.) \) is the trace (sum of diagonal elements)

and \( X = [x_1, ..., x_p] \) is such that \( x_{ik} = 1 \) if \( c(i) = k \).

A. Fender, N. Emad, S. Petiton, M. Naumov. “Parallel Modularity Clustering.” ICCS, 2017
MODULARITY MAXIMIZATION PIPELINE

Graph Preprocessing → Modularity → Eigensolver → Points Clustering

\[-\frac{1}{2\omega} vv^T\]

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

\[
\begin{bmatrix}
-0.5 \\
0.0 \\
0.0 \\
0.6 \\
-0.5
\end{bmatrix}
\]

\[
\begin{bmatrix}
1 \\
1 \\
1 \\
1 \\
1
\end{bmatrix}
\]
SPECTRAL MODULARITY MAXIMIZATION

84% hit rate

Spectral Modularity maximization

Ground truth
The sparse matrix vector multiplication takes 90% of the time in the eigensolver.

The eigensolver takes 90% of the time.
**MODULARITY VS. LAPLACIAN CLUSTERING**

**Modularity**
- ✓ higher and steadier modularity score
- ✓ 3x speedup over Laplacian

**Laplacian**
- ✓ homogeneous cluster sizes

| # | Matrix          | $n = |V|$ | $m = |E|$ | Application          |
|---|----------------|-------|-------|----------------------|
| 1 | preferentialA... | 100,000 | 499,985 | Artificial           |
| 2 | caidaRouterLevel | 192,244 | 609,066 | Internet             |
| 3 | coAuthorsDBLP    | 299,067 | 977,676 | Coauthorship         |
| 4 | citationCiteSeer | 268,495 | 1,156,647 | Citation             |
| 5 | coPapersDBLP     | 540,486 | 15,245,729 | Affiliation         |
| 6 | coPapersCiteSeer | 434,102 | 16,036,720 | Affiliation         |
| 7 | as-Skitter       | 1,696,415 | 22,190,596 | Internet            |
| 8 | hollywood-2009  | 1,139,905 | 113,891,327 | Coauthorship        |

Nvidia Titan X (Pascal), Intel Core i7-3930K @3.2 GHz
**SPEEDUP AND QUALITY VS. AGGLOMERATIVE**

0.8s on network with 100 million edges on a single Titan X GPU

*Speedup* 3x over agglomerative* scheme

*Tradeoff* Speed vs. quality


Nvidia Titan X (Pascal), Intel Core i7-3930K @3.2 GHz
Spectral Clustering in CUDA Toolkit 9.0 release of nvGRAPH
CUDA TOOLKIT 9.0
nvGRAPH API

nvgraphStatus_t nvgraphSpectralClustering ( nvgraphHandle_t handle, const nvgraphGraphDescr_t graph_descr, const size_t weight_index, const struct SpectralClusteringParameter *params, int *clustering, void *eig_vals, void *eig_vects);

struct SpectralClusteringParameter {
    int n_clusters;
    int n_eig_vects;
    nvgraphSpectralClusteringType_t alg
    float evs_tolerance
    int evs_max_iter;
    float kmean_tolerance;
    ...
};
Conclusions
SPECTRAL CLUSTERING

• Software Framework
  • similar for both

• Laplacian - Minimum balanced cut\(^1\)
  • Probably the most common metric, with balancing involved in the cost function
  • Requires careful choice of eigensolver

• Modularity maximization \(^2\)
  • Widely used in analysis of social networks
  • Faster to compute

Thank you

H7129 - Accelerated Libraries
Monday and Wednesday @ 4:00, Pod B