Modularity-Maximizing Graph Communities via Mathematical Programming

Ying Xuan

February 24, 2009

Table of contents

Modularity Maximization

Given undirected graph $G = (V, E)$, find a clustering $\{C_1, \ldots, C_k\}$ which is a **disjoint** partition of V such that the modularity of the clustering $\mathcal{C}$[1]:

$$\mathcal{Q}(\mathcal{C}) = \frac{1}{2m} \sum_{u,v} (a_{u,v} - \frac{d_u d_v}{2m}) \cdot \delta(\gamma(u), \gamma(v))$$

is **maximized**. Here,

- $a_{u,v} = a_{v,u} = 1$ if $(u, v) \in E$, otherwise 0;
- d_u denotes the degree of any vertex u;
- m is the number of edges in G;
- $\gamma(v)$ denotes the (unique) index of the cluster to which v belongs;
- $\delta(x, y)$ is the Kronecker Delta, which equals to 1 if $x = y$, otherwise 0.

Recent Efforts

- This maximization problem is NP-complete[2];
- Correlation Clustering[3] interprets "partial membership of the same cluster" as a distance metric, group nearby ones together;
- Spectral Clustering[4] repeatedly divides clusters based on the largest eigenvalue and corresponding eigenvector of the modularity matrix.

Main Contributions

Two heuristics:

- LP relaxation and Distance-based Rounding Algorithm;
- Quadratic Programming and Randomized Rounding Algorithm.

One potential method ratio analysis:

- Similarity with Min-Disagree problem (4-approx) in LP formulation.

Pitfalls

- No ratio analysis over the LP rounding or randomized rounding for SDP;
- Significant huge resource requirement due to $\Theta(n^3)$ constraints in LP and $\Theta(n^2)$ variables in the vector programming; $\Longleftrightarrow$ Huge time complexity and computation overhead;
- Performance of LP rounding relies on the selection of center vertex.

Linear Programming Algorithm

- IP Formulation and LP relaxation;
- Distance-based Rounding;

IP formulation

Integer Program

Maximize

$$\frac{1}{2m} \cdot \sum_{u,v} (a_{u,v} - \frac{d_u d_v}{2m}) \cdot (1 - x_{u,v})$$

Subject to

$$x_{u,w} \leq x_{u,v} + x_{v,w} \qquad \forall u, v, w \in V$$

$$x_{u,v} \in \{0, 1\} \qquad \forall u, v \in V$$

Let

$$m_{u,v} = \frac{a_{u,v}}{2m} - \frac{d_u d_v}{4m^2}$$

LP formulation

Linear Program

Maximize

$$\sum_{u,v} m_{u,v} \cdot (1 - x_{u,v})$$

Subject to

$$x_{u,w} \leq x_{u,v} + x_{v,w} \qquad \forall u, v, w \in V$$

$$x_{u,v} \geq 0 \qquad \forall u, v \in V$$

Use CPlex to solve it $\Longrightarrow \Theta(n^3) constraints$

Distance-based rounding

Algorithm 1

$dist(u, v) \leftarrow x_{u,v}$ for LP solution X;

$S \leftarrow V$;

while $S \neq \emptyset$ **do**

  Select $u \in S$; $\triangleright$ randomly select

  $T_u \leftarrow \{v | dist(u, v) \leq 1/2\}$

  if average $dist(u, v) < 1/4$ for all $v \in \{T_u \setminus \{u\}\}$ **then**

    Make $C = T_u$ a cluster;

  else

    Make $C = \{u\}$ a singleton cluster

$S \leftarrow S \setminus C$;

Refine the result using local-search algorithm.

How good can this heuristic be?

Analysis

Definition

Min-Disagree problem formulated by [3]:

- Given a **complete** graph where all edges are labeled as respectively '+' or '-' to indicate the similarity or dis-similarity of vertex pair;
- partition the graph into clusters such that the number of errors ('-' edges within clusters and '+' edges between clusters) are minimized.

Analysis(Cont')

Min-Disagree LP

Minimize $\sum_{(u,v)\in E_+} x_{u,v} + \sum_{(u,v)\in E_-} (1 - x_{u,v})$

$\iff$ $|E_+| - \sum_{(u,v)\in E} \mu_{u,v}(1 - x_{u,v})$

Subject to

$$x_{u,w} \leq x_{u,v} + x_{v,w} \qquad \forall u, v, w \in V$$

$$x_{u,v} \geq 0 \qquad \forall u, v \in V$$

where $\mu_{u,v} = 1$ if (u, v) is '$+$' edge, otherwise 0.

Analysis(Cont')

Further results:

- if define

$$\mu_{u,v} = m_{u,v} = \frac{a_{u,v}}{2m} - \frac{d_u d_v}{4m^2}$$

  Mis-Degree formulation on **complete** graph is similar to the IP formulation of modularity maximization.

- Mis-Degree problem has a 4-approximation rounding algorithm;
- Due to existence of $|E_+|$, it is hard to get a same approximation algorithm for modularity maximization.

Vector Programming Algorithm

Kernel:

- Formulate the problem as Quadratic Program;
- Relax to vector program (Semi-definite programming);
- Use randomized cutting hyperplane to round the SDP solution.

Quadratic Programming Formulation

Considering **partitioning the graph into two communities** $(S, \bar{S})$ **of maximum modularity**, let $y_v = \pm 1$ indicate that vertex v belongs (or not) to S. Therefore, the formulation is:

Quadratic Program

Maximize

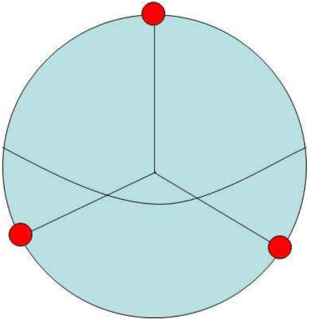
$$\frac{1}{4m} \sum_{u,v \in V} m_{u,v}(1 + y_u y_v)$$

Subject to

$$y_v^2 = 1 \text{ for all } v$$

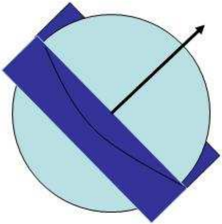
Semi-definite Relaxation

Key idea: relax $y = \pm 1$ to n-dimensional vector $\overrightarrow{y}$ with $\| y \| = 1$.



The product of $y_u y_v$ is corresponding to inner product of $\overrightarrow{y_u} \bullet \overrightarrow{y_v} = \cos\theta$

Randomized Rounding



- randomly select a n-dimension vector $\overrightarrow{s}$, where each component is following independent $\mathcal{N}(0,1)$ Gaussian.
- $S = \{v | \overrightarrow{y_v} \bullet \overrightarrow{s} \geq 0\}$
- $\bar{S} = \{v | \overrightarrow{y_v} \bullet \overrightarrow{s} < 0\}$

Algorithm

Algorithm 2

Hierarchical Clustering:

- use Semi-definite programming find a near-optimal division of a larger cluster $\Longleftrightarrow$ locally optimal;
- repeat the division until no further partition will increase the modularity;
- do local search post-processing to refine the solution.

Bibliography

M. Newman and M. Girvan. "Finding and evaluating community structure in networks". Physical Review E, 69, 2004.

U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, and D. Wagner. "On modularity clustering". IEEE Transactions on Knowledge and Data Engineering, 20(2):172 – 188, 2008.

M. Charikar, V. Guruswami, and A. Wirth. "Clustering with qualitative information". Journal of Computer and System Sciences, pages 360 – 383, 2005.

M. Newman. "Modularity and community structure in networks." Proc. Natl. Acad. Sci. USA, 103:8577 – 8582, 2006.