On the Approximability of Modularity Clustering

Newman's Community Finding Approach for Social Nets

Bhaskar DasGupta

Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607, USA
dasgupta@cs.uic.edu

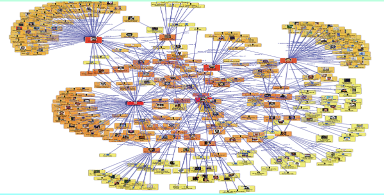
July 2, 2011

Joint work with Devendra Desai (Rutgers University)

Outline

Interaction systems in biology and social science

- modeled as pairwise interaction graphs
  - nodes are entities
  - edges are interactions between entities
- Goal: partition nodes into communities or clusters of **statistically significant** interactions



www.fmsasg.com/SocialNetworkAnalysis/

What are clusters of “statistically significant” interactions?

Unsatisfactory choices in practical applications (too strict, computationally difficult,. . .)

- cliques
- dense subgraphs

⋮

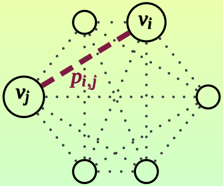
Outline

1. Introduction
2. Model Based Community Finding
3. Newman's Modularity Clustering
   - Generalization to other types of graphs
4. Previously known complexity results
5. Our results
   - Main results
   - Some proof ideas for main results
   - Other results

Model Based Clustering

Model: define a null model $\mathcal{G}$ of a background random graph

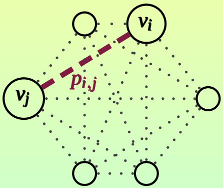
provides probability $p_{i,j}$ of edge between v_i and v_j (implicitly or explicitly)



Model Based Clustering

Model: define a null model $\mathcal{G}$ of a background random graph

provides probability $p_{i,j}$ of edge between v_i and v_j (implicitly or explicitly)



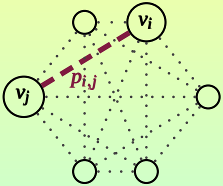
Input graph G: $0 < w_{i,j} \leq 1$

normalized weight

Model Based Clustering

Model: **define a null model $\mathcal{G}$ of a background random graph**

provides probability $p_{i,j}$ of edge between v_i and v_j (implicitly or explicitly)



Input graph G: $0 < w_{i,j} \leq 1$

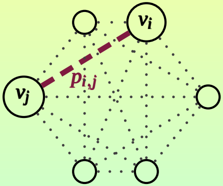
normalized weight

$| \, w_{i,j} - p_{i,j} \, |$ **is large**

Model Based Clustering

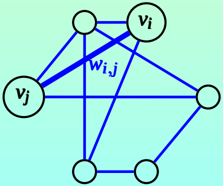
Model: define a null model $\mathcal{G}$ of a background random graph

provides probability $p_{i,j}$ of edge between v_i and v_j (implicitly or explicitly)



Input graph G**:** $0 < w_{i,j} \leq 1$

normalized weight



$| w_{i,j} - p_{i,j} |$ is large

$\Downarrow$

$\{v_i, v_j\}$ is statistically significant

$\{+,-\}$-correlation clustering

Goal: maximize number of $+$ edges minus number of $-$ edges inside clusters

e.g. **[Bansal, Blum, Chawla, 2002], [Charikar, Guruswami, Wirth, 2003], [Swamy, 2004]**

- given input graph H with each edge labeled as $+$ or $-$

Correlation Clustering as a Model Based Clustering

$\{+,-\}$-correlation clustering

Goal: maximize number of $+$ edges minus number of $-$ edges inside clusters

e.g. **[Bansal, Blum, Chawla, 2002], [Charikar, Guruswami, Wirth, 2003], [Swamy, 2004]**

- given input graph H with each edge labeled as $+$ or $-$
- let G be the graph consisting of all edges labeled $+$ in H
  ($\boldsymbol{a_{i,j}}$: $(i,j)^{\text{th}}$ entry in adjacency matrix)

Correlation Clustering as a Model Based Clustering

$\{+,-\}$-correlation clustering

Goal: maximize number of $+$ edges minus number of $-$ edges inside clusters

e.g. **[Bansal, Blum, Chawla, 2002], [Charikar, Guruswami, Wirth, 2003], [Swamy, 2004]**

- given input graph H with each edge labeled as $+$ or $-$
- let G be the graph consisting of all edges labeled $+$ in H
  ($\boldsymbol{a_{i,j}}$: $(i,j)^{\text{th}}$ entry in adjacency matrix)
- null model $\mathcal{G}$: $\boldsymbol{p_{i,j}} = \begin{cases} 0 & \text{if the edge was labeled } + \text{ or missing} \\ 1 & \text{otherwise } \textit{i.e.}\text{, labeled } - \end{cases}$

Correlation Clustering as a Model Based Clustering

$\{+,-\}$-correlation clustering

Goal: maximize number of $+$ edges minus number of $-$ edges inside clusters

e.g. **[Bansal, Blum, Chawla, 2002], [Charikar, Guruswami, Wirth, 2003], [Swamy, 2004]**

- given input graph H with each edge labeled as $+$ or $-$
- let G be the graph consisting of all edges labeled $+$ in H
  ($\boldsymbol{a_{i,j}}$: $(i,j)^{\text{th}}$ entry in adjacency matrix)
- null model $\mathcal{G}$: $\boldsymbol{p_{i,j}} = \begin{cases} 0 & \text{if the edge was labeled } + \text{ or missing} \\ 1 & \text{otherwise } \textit{i.e.}\text{, labeled } - \end{cases}$
- contribution of an edge **inside cluster** to total score: $\boldsymbol{a_{i,j} - p_{i,j}}$

Correlation Clustering as a Model Based Clustering

$\{+,-\}$-correlation clustering

Goal: maximize number of $+$ edges minus number of $-$ edges inside clusters

e.g. **[Bansal, Blum, Chawla, 2002], [Charikar, Guruswami, Wirth, 2003], [Swamy, 2004]**

- given input graph H with each edge labeled as $+$ or $-$
- let G be the graph consisting of all edges labeled $+$ in H
  ($\boldsymbol{a_{i,j}}$: $(i,j)^{\text{th}}$ entry in adjacency matrix)
- null model $\mathcal{G}$: $\boldsymbol{p_{i,j}} = \begin{cases} 0 & \text{if the edge was labeled } + \text{ or missing} \\ 1 & \text{otherwise } \textit{i.e.}\text{, labeled } - \end{cases}$
- contribution of an edge **inside cluster** to total score: $\boldsymbol{a_{i,j}} - \boldsymbol{p_{i,j}}$
- total score: appropriate function of individual scores of edges

Outline

Newman's Modularity Clustering

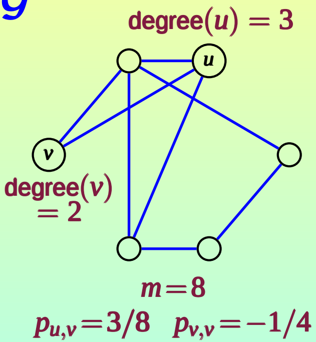
- A specific model based clustering
- Extremely popular in practice (in biology, social science, etc.) For example, see
  - (Ravasz et al., Science, 2002)
  - (Newman and Girvan, Physical Review E, 2004)
  - (Newman, Physical Review E, 2004)
  - (Newman, PNAS, 2006)
  - (Guimera et al, Nature Physics, 2007)
  - (Leicht and Newman, Physical Review Letters, 2008)
- null model dependent on the degree distribution of the input graph
- can be used for directed/undirected and weighted/unweighted graphs

Null Model $\mathcal{G}$

Input graph $G = (V, E)$ has m edges

$$\forall\, u, v \in V: p_{u,v} = \frac{\text{degree}(u) \times \text{degree}(v)}{2m}$$

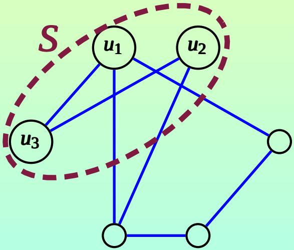
$u = v$ allowed



- Expected degree of a node v is precisely degree(v) and, thus, the expected number of edges is m

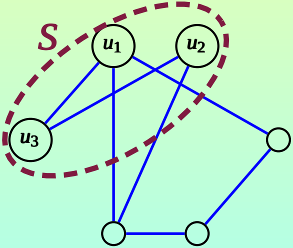
$$\sum_{v \in V} \text{degree}(u) \times \frac{\text{degree}(v)}{2m} = \text{degree}(u)$$

Fitness of a cluster (subset of nodes) $S \subseteq V$



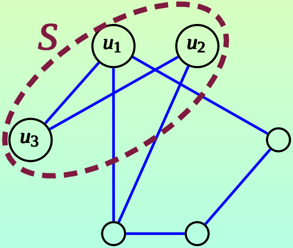
Fitness of a cluster (subset of nodes) $S \subseteq V$

Contribution for
an edge $\{u,v\} \in E$: $1 - p_{u,v}$
a non-edge $\{u,v\} \notin E$: $-p_{u,v}$



Fitness of a cluster (subset of nodes) $S \subseteq V$

Contribution for
an edge $\{u, v\} \in E$: $\mathbf{1 - p_{u,v}}$
a non-edge $\{u, v\} \notin E$: $\mathbf{-p_{u,v}}$



combining both cases:

Fitness of a cluster (subset of nodes) $S \subseteq V$

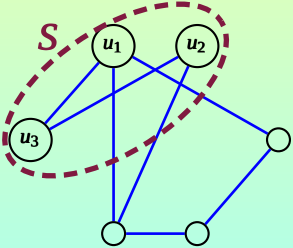
Contribution for

an edge $\{u,v\} \in E$: $1 - p_{u,v}$

a non-edge $\{u,v\} \notin E$: $-p_{u,v}$

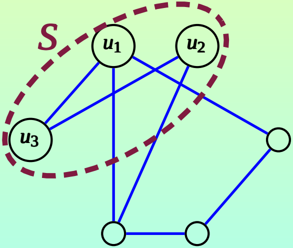
combining both cases:

$$a_{u,v} - p_{u,v} = a_{u,v} - \frac{\text{degree}(u) \times \text{degree}(v)}{2m}$$



Fitness of a cluster (subset of nodes) $S \subseteq V$

Contribution for
an edge $\{u, v\} \in E$: $1 - p_{u,v}$
a non-edge $\{u, v\} \notin E$: $-p_{u,v}$



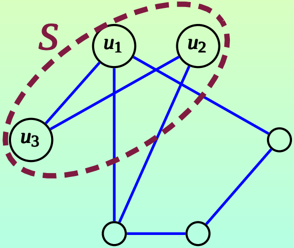
combining both cases:

$$a_{u,v} - p_{u,v} = a_{u,v} - \frac{\text{degree}(u) \times \text{degree}(v)}{2m}$$

Add for all pairs of nodes in S

Fitness of a cluster (subset of nodes) $S \subseteq V$

Contribution for
an edge $\{u,v\} \in E$: $1 - p_{u,v}$
a non-edge $\{u,v\} \notin E$: $-p_{u,v}$



combining both cases:

$$a_{u,v} - p_{u,v} = a_{u,v} - \frac{\text{degree}(u) \times \text{degree}(v)}{2m}$$

Add for all pairs of nodes in S

fitness of S

$$\mathsf{M}(S) = \sum_{u,v \in S} \left(a_{u,v} - \frac{\text{degree}(u) \times \text{degree}(v)}{2m} \right)$$

UIC

Modularity value of a clustering $\mathcal{C}$

- $\mathcal{C} = \{V_1, V_2, \ldots, V_k\}$ is a partition of V
- **modularity is sum of individual fitnesses**
  (normalized by dividing by $2m$ to get a value between 0 and 1)

$$\mathsf{M}(\mathcal{C}) = \frac{1}{2m} \times \sum_{i=1}^{k} \mathsf{M}(V_i)$$

- Goal: **find a clustering $\mathcal{C}$ to maximize $\mathsf{M}(\mathcal{C})$**
  (note: number of clusters k is unspecified)

Equivalent Formula for Modularity value (via simple algebraic manipulation)

Original modularity

$$\mathsf{M}(\mathcal{C}) = \frac{1}{2m} \times \sum_{i=1}^{k} \sum_{u,v \in V_i} \left(a_{u,v} - \frac{\text{degree}(u) \times \text{degree}(v)}{2m} \right)$$

Equivalent formula

$$\mathsf{M}(\mathcal{C}) = \sum_{i=1}^{k} \left(\frac{m_i}{m} - \left(\frac{D_i}{2m} \right)^2 \right)$$

m_i = number of edges whose both endpoints are in V_i
D_i = sum of degrees of nodes in V_i

Equivalent Formula for Modularity value (via simple algebraic manipulation)

Original modularity

$$\mathsf{M}(\mathcal{C}) = \frac{1}{2m} \times \sum_{i=1}^{k} \sum_{u,v \in V_i} \left(a_{u,v} - \frac{\text{degree}(u) \times \text{degree}(v)}{2m} \right)$$

Yet another equivalent formula

$$\mathsf{M}(\mathcal{C}) = \sum_{V_i, V_j \,:\, i<j} \left(\frac{D_i D_j}{2m^2} - \frac{m_{i,j}}{m} \right)$$

$m_{i,j}$ = number of edges with one endpoint in V_i and another in V_j

D_i = sum of degrees of nodes in V_i

Outline

Generalization to other types of graphs

Undirected graphs

$$\mathsf{M}(\mathcal{C}) = \frac{1}{2m} \times \sum_{i=1}^{k} \sum_{u,v \in V_i} \left(a_{u,v} - \frac{\text{degree}(u) \times \text{degree}(v)}{2m} \right)$$

Generalization to other types of graphs

Directed graphs

$$\mathsf{M}(\mathcal{C}) = \frac{1}{\underset{m}{\cancel{2m}}} \times \sum_{i=1}^{k}\sum_{u,v\in V_i}\left(a_{u,v} - \frac{\overset{\text{out-degree}}{\cancel{\text{degree}}}(u)\times \overset{\text{in-degree}}{\cancel{\text{degree}}}(v)}{\underset{m}{\cancel{2m}}}\right)$$

Generalization to other types of graphs

(Edge)-Weighted undirected graphs

$$\mathsf{M}(\mathcal{C}) = \frac{1}{2m} \times \sum_{i=1}^{k} \sum_{u,v \in V_i} \left(a_{u,v} - \frac{\overset{\text{weighted-degree}}{\sout{\text{degree}}}(u) \times \overset{\text{weighted-degree}}{\sout{\text{degree}}}(v)}{2m} \right)$$

- edge weights are **non-negative**
- weighted degree of v is sum of **weights** of edges incident on v
- $a_{u,v}$ is the **weight** of the edge $\{u, v\}$
- m is sum of edge **weights**

UIC

Outline

UIC

Previously known complexity results

$\text{OPT} = \max_{\mathcal{C}} \{ \text{M}(\mathcal{C}) \}$ denotes the maximum modularity value

Previously known complexity results

- computing OPT is NP-complete for sufficiently dense graphs
  (Brandes, Delling, Gaertler, Görke, Hoefer, Nikoloski and Wagner, 2007)
  - the reduction roughly requires $\Omega(\sqrt{n})$ degree for every node
  - NP-completeness result holds even if any solution is constrained to contain no more than two clusters
- Many results on heuristics and their experimental evaluations
- As (Agarwal and Kempe, 2008) observed:

  In spite of its extreme popularity, not much is known about the computational complexity aspect of modularity clustering beyond NP-completeness

Outline

Outline

Our main results (undirected graphs)

Inapproximability

Our main inapproximability results (undirected graphs)

Our main results (undirected graphs)

Inapproximability

Our main inapproximability results (undirected graphs)

- **computing OPT is APX-hard for dense graphs (edge-complement of 3-regular graphs)**

Our main inapproximability results (undirected graphs)

- computing OPT is APX-hard for dense graphs (edge-complement of 3-regular graphs)
- **optimally partitioning into 2 clusters is NP-complete even when the graph is sparse and regular (d-regular for any constant $d \geq 9$)**

Our main results (undirected graphs)

Approximation algorithms

Our main approximability results (undirected graphs)

Our main results (undirected graphs)

Approximation algorithms

Our main approximability results (undirected graphs)

- **small number of clusters well-approximate OPT**
  in particular, partitioning into two clusters achieves $\frac{1}{2} \times \text{OPT}$

$$\text{OPT}_2 \geq \frac{1}{2} \times \text{OPT}$$

Our main approximability results (undirected graphs)

- small number of clusters well-approximate OPT
  in particular, partitioning into just two clusters achieves $\frac{1}{2} \times$ OPT
  $$\text{OPT}_2 \geq \frac{1}{2} \times \text{OPT}$$
- **An approximation algorithm whose approximation ratio is logarithmic in the maximum degree**
  (provided, roughly speaking, maximum degree is $o(n)$)

Our main results (undirected graphs)

Approximation algorithms

Our main approximability results (undirected graphs)

- small number of clusters well-approximate OPT
  in particular, partitioning into just two clusters achieves $\frac{1}{2} \times \text{OPT}$
  $$\text{OPT}_2 \geq \frac{1}{2} \times \text{OPT}$$
- An approximation algorithm whose approximation ratio is logarithmic in the average degree
  (provided, roughly speaking, average degree is $o(n)$)
- **for locally-dense graphs (*i.e.*, every node has a degree of $\Omega(n)$) a solution within any constant additive error in polynomial time**
  via use of regularity lemma

Outline

APX-hardness for dense graphs

3-MIS ≡ maximum independent set for 3-regular graphs

$$\delta_\ell = \frac{94}{194} \qquad \delta_h = \frac{95}{194}$$

$L \in NP$	$\xrightarrow{[1]}$	3-MIS	$\longrightarrow$	Modularity Clustering
$I \in L$	$\longrightarrow$	$\Psi \geq \delta_h n$	$\longrightarrow$	$\text{OPT} \geq \frac{2\times(4\delta_h^2-\delta_h)}{n-4}$
$I \notin L$	$\longrightarrow$	$\Psi \leq \delta_\ell n$	$\longrightarrow$	$\text{OPT} \leq \frac{4\delta_\ell-1}{n-4}$

[1] Chlebík and Chlebíková, 2006

Logarithmic approximation algorithm

- **modularity function is neither monotone nor sub-modular, thus cannot use techniques from those domains**
- **we show that a natural** LP**-relaxation for modularity clustering has large integrality gap, so cannot use** LP**-based techniques**
- **standard algorithmic approaches such as greedy provably do not work well**
- **instead, we go via quadratic optimization and semi-definite programming (**SDP**) based approach**

Logarithmic approximation algorithm

Quadratic optimization and SDP-based approach

- $\text{OPT}_2 \geq \dfrac{\text{OPT}}{2}$, thus suffices to partition into 2-clusters
- express this 2-cluster partition problem as a quadratic integer program after some algebraic simplification

$$w(u,v) = \frac{a_{u,v} - \frac{\text{degree}(u) \times \text{degree}(v)}{2m}}{4m}, \quad W = [w_{u,v}] \in \mathbb{R}^{n \times n}$$

maximize $\mathbf{x}^{\mathrm{T}} W \mathbf{x}$ subject to $\mathbf{x} \in \{-1, 1\}^n$

- But, but, . . . , the diagonal entries $w_{u,u}$'s of W are negative

Logarithmic approximation algorithm (continued)

- **ignore diagonal entries; later show that it was OK to ignore**

$$\textbf{maximize} \sum_{u \neq v \in V} w_{u,v} x_u x_v \quad \textbf{subject to } \forall u \in V: x_u \in \{-1, 1\} \qquad (1)$$

- **obtain a lower bound on** OPT **using an explicit graph decomposition**

$$\text{OPT} = \Omega\left(\frac{1}{\textbf{average degree}}\right)$$

- **Approximate** (1) **within a factor of** $O\left(\frac{1}{\log \text{OPT}}\right)$ **by an appropriate adaptation of the algorithm of (Charikar & Wirth, FOCS 2004)**

Outline

Our other results for directed or weighted graphs

all the algorithmic results can be generalized to directed and/or weighted graphs via "appropriate modifications"

Our other results

alternate null model (undirected graphs)

Idea of alternative null models has been explored before empirically (Gaertler, Görke, Wagner, 2007) (Karrer and Newman, 2009)

We explore the classical Erdös-Rényi random graph null model $G(n, p)$

- each possible edge is selected uniformly and randomly with a probability of p
- set $p = \frac{2m}{n \times (n-1)}$ such that the expected number of edges in $G(n, p)$ is m

Our observation

This is same as computing Newman's modularity measure on a $\left(\frac{m}{n}\right)$-regular graph

UIC

Our other results

alternate overall modularity (undirected graphs)

Exact or approximate solutions to Newman's modularity measure may produce many trivial clusters of single nodes

Example

If the maximum degree is at most $\frac{\sqrt[4]{n}}{16 \ln n}$, then there always exists a clustering such that

- **every cluster except one consists of a single node**
- **modularity value is at least 25% of the maximum**

A possible reason

total modularity is **sum** of individual cluster modularities

Our other results

alternate overall modularity (undirected graphs)

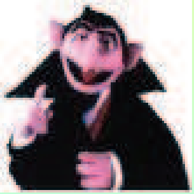
New modularity equation

total modularity is **minimum** of individual cluster modularities

Results

- **new objective indeed avoids generating trivial clusters**
- **its optimal value is precisely half of the optimal value of old objective**

The End



Any questions?