CS369M: Algorithms for Modern Massive Data Set Analysis **Lecture 12 - 11/04/2009**

Introduction to Graph Partitioning

Lecturer: Michael Mahoney *Scribes: Noah Youngs and Weidong Shao*

**Unedited Notes*

1 Graph Partition

A graph partition problem is to cut a graph into 2 or more "good" pieces. The methods are based on

1. spectral. Either global (e.g., Cheeger inequality,) or local.
2. flow-based. min-cut/max-flow theorem. LP formulation. Embeddings. Local Improvement.
3. combination of spectral and flow.

Note that not all graphs have good partitions.

Question: Can we certify that there are no good clusters in a graph?

"Good" clusters have the following properties:

1. internally (intra) - well connected.
2. externally (inter) - relatively poor

How do we quantify this?

Extreme cases:

1. split into 2 disconnected pieces
2. split into $S, \bar{S}$ on 2 maximum complete induced subgraphs.

2 Min cut problem

<u>DEFINE</u> Given $G = (V, E)$, a cut is a partition of V, $(S, \bar{S})$, where $S \subset V$.
Given $s, t \in V$, an (s, t) cut is a cut s.t. $s \in S, t \in \bar{S}$
A cut set of a cut is $(u, v) : (u, v) \in E, u \in S, v \in \bar{S}$

The min cut problem: find the cut of "smallest" edge weights

1. good: Polynomial time algorithm (min-cut = max flow)
2. bad: often get very inbalanced cut
3. in theory: cut algorithms are used as a sub-routine in divide and conquer algorithm
4. in practice: often want to "interpret" the clusters or partitions

3 Max Flow Problem

<u>DEFINE</u> Call the capacity of an edge $(u, v) \in E : e_{uv}$
Let there be a cost function: $c : E \to R^+$, delineated c_{uv} or c_e
Then a flow is function of $f : E \to R^+$

1. $f_{uv} \leq C_{uv} \forall u, v$ (capacity constraints)
2. $\sum_{(u,v)\in E} f_{uv} = \sum f_{vu}$ (conservation of flows)

Then the value of the flow

$$|f| = \sum_v f_{sv}$$

The MAX flow problem:

$$\max |f|$$

The capacity of (s, t) cut is $c(S, \bar{S}) = \sum C_{uv}$.

The min cut problem is

$$\min C(S, T)$$

Note: this is a "single flow problem" ... i.e. only one s and one t

Theorem: the max value of an $s - t$ flow is equal to the min capacity of an $s - t$ cut.

Proof idea:

$\max flow \leq \min cut$ (weak duality)

Does there exists a cut that achieves equality?
Yes, from the strong duality theorem we can also solve the dual of the max-flow problem, which is the min-flow problem

Primal: (max flow)

$$\max |f|$$

subject to

$$f_{uv} \leq C_{uv}$$

Dual: (min cut)

$$\min \sum_{(i,j)\in E} c_{ij} d_{ij}$$

s.t.

$$d_{ij} - p_i + p_j \geq 0, ij \in E$$

$$p_s = 1, p_t = 0, p_i \geq 0, \in V$$

$$d_{ij} \geq 0, ij \in E$$

Can we add a "balance" condition?

1. want a good cut value $E(S, \bar{S})$
2. want $S, \bar{S}$ both to be balanced - same size, or approximately same size

the answer is "Yes"

Explicit balance conditions:
Graph bisection - min cut s.t. $|S| = |\bar{S}| = n/2$
β balanced cut min cut s.t $|S| = \beta n$, $|\bar{S}| = (1-\beta)n$

Implicit Balance conditions:

1. input balance constraints
2. expansion. $\frac{E(S,\bar{S})}{\frac{|S|}{n}}$ (def this as :h(S))
3. sparsity $\frac{E(S,\bar{S})}{|S||\bar{S}|}$ (def this as :sp(S))
4. conductance $\frac{E(S,\bar{S})}{\frac{Vol(S)}{n}}$ (with $Vol(S) = \sum_{ij \in E} deg(V_i)$)
5. normalized cut $\frac{E(S,\bar{S})}{vol(|S|)vol(|\bar{S}|)}$
   (latter two are used in ML)
6. quotien cut $\frac{E(S,\bar{S})}{min(vol(|S|),vol(|\bar{S}|))}$

expansion and sparcity: are "same" (in the following sense:)

$$\min h(S) \approx \min sp(S)$$

Quotient cuts yield a tight bound on cheeger inequality
In-practice: bias towards high degree nodes

Note:

quotient cuts get balanced implicitly, no explicit constraints on inter or intra connectivity

Z^2 on random geometric graps or nice planer graphs yield good quotient cuts

More generally, - very inbalanced - disconnected clusters.

Example: extremely sparse random graph $G(n,p)$ model, $p \geq \log n^2/n$ expander $p\ logn/n$

4 Graph Partition Algorithms

4.1 Local Improvement

Developed in the 70's
Often it is a greedy improvemnt
Local minima are a big problem

Usual methods improve them by constant factors
- simulated annealing
- big difference in practice

Kernighan-Lin algorithm, fundamental work, no-longer used due to $\Theta(n^2)$ performance

Fiduccia-Mattheyses algorithm, linear time, still commonly used

METIS algorithm from Karypis and Kumar, works very well in practice, especially on low dimensional graphs

4.2 Spectral methods

Develped in the 70's and 80's
Serivce level gaurantee (Cheeger's inequality)
At root, this is relaxation or rounding method related to QIP formualation :

$MAX_{x\in(-1.1)^n} \frac{x^t Lx}{x^t x}$

- quadratic worst case.

- hyperplane rounding:
  -compute an eigenvector
  - cut according to some rules
  - post processing with local improvments

4.3 Flow-based methods

Developed in the 90's
Consider all pairs, multi-commodity flow problem.
Want to route the commodities s.t. the constraints are satisfied without bottlenecks.

Idea: bottleneck in flow computation corresponds to good cuts.

$k-$commodity problem: does not satisfy strong duality. does satisfy approx min-cut max flow value gap $\leq \Theta(logn)$

- releax flow to LP
- embed solution in l_1
- Round soltuion to $0, 1$, $\Theta(\log n)$ worst case.

4.4 Additional Graph Partitioning Notes

These methods "fail".... i.e. achieve the worst case, on the following graphs:
- spectral methods - fail on long stringy pieces —– ————–
- flow-based methods - fail on expander graphs. n choose 2 pairs but most pairs are far apart. (log n) apart.

Improvements/extensions for large data:

there exist hybrid flow based and local methods
(cut around the cut) local spectrum methods

— good cut around a start node of a given size
– time depends on the size of the output.

4.5 Methods that combine spectral and flow

- ARV algorithm (developed a few years ago by Arora, Rao, and Vazirani)
- most hyrbid algorithms are theoretical, but some implementations embed in SDP.
- approximate solution (two-player game).
- boosting & emsemble methods

5 References

1. Schaeffer, "Graph Clustering", Computer Science Review 1(1): 27-64, 2007
2. Kernighan, B. W.; Lin, Shen (1970). "An efficient heuristic procedure for partitioning graphs". Bell Systems Technical Journal 49: 291-307.
3. CM Fiduccia, RM Mattheyses. "A Linear-Time Heuristic for Improving Network Partitions". Design Automation Conference.
4. G Karypis, V Kumar (1999). "A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs". Siam Journal on Scientific Computing.