Graph Clustering and Minimum Cut Trees

Flake, Tarjan, Tsioutsiouliklis

April 4, 2007
Introduction

Goal: Clustering a Data Set

Criteria:
- large intra-cluster cuts
- small inter-cluster cuts

Approach:
- Add artificial sink to graph
- Utilize Minimum Cut Trees
Given $G = (V, E)$, construct $G_\alpha$ by introducing a new node $t$ and connecting it to all $v \in V$ with edges of capacity $\alpha$. 

$G_\alpha$
**Community**  Let $s, r \in V$. The *Community* of $s$ in $G$ with respect to $r$ is the minimal $S$ such that $s \in S$ and $(S, V - S)$ is a minimum $s - r$ cut.
**Terminology**

**Community** Let $s, r \in V$. The *Community* of $s$ in $G$ with respect to $r$ is the minimal $S$ such that $s \in S$ and $(S, V - S)$ is a minimum $s - r$ cut.

**Web Community** A *Web community* $S$ is a collection of nodes that has the property that all nodes of the Web community predominantly link to other Web community nodes. That is:

$$\sum_{v \in S} w(u, v) > \sum_{v \in \bar{S}} w(u, v), \quad \forall u \in S$$
Let $G(V, E)$ be a graph. A minimum cut tree of $G$ is a weighted tree, $T$, on vertex set $V$ such that for any pair $r, s \in V$, the capacity of the minimum $(r, s)$–cut in $G$ is equal to the weight of the minimum weight edge, $c(e^*)$, in $T$ on the unique path joining the two nodes. Moreover, the bipartition of $V$ obtained by removing $e^*$ from $T$ is a minimum $(r, s)$–cut.
Terminology

Minimum Cut Tree Example

Figure: A Graph and Two Minimum Cut Trees
Let \((S, \bar{S})\) be a cut in \(G\). We define the expansion of a cut as:

\[
\psi(S) = \frac{\sum_{u \in S, v \in \bar{S}} w(u, v)}{\min\{|S|, |\bar{S}|\}}
\]
Let \((S, \bar{S})\) be a cut in \(G\). We define the expansion of a cut as:

\[
\psi(S) = \frac{\sum_{u \in S, \; v \in \bar{S}} w(u, v)}{\min\{|S|, |\bar{S}|\}}
\]

The expansion of a subgraph is the minimum expansion over all cuts. The expansion of a clustering is the minimum expansion over all clusters.
Main Theorem

Theorem
Let $G = (V, E)$ be an undirected graph, $s \in V$ a source, and connect an artificial sink $t$ with edges of capacity $\alpha$ to all nodes. Let $S$ be the community of $s$ with respect to $t$. For any non-empty $P$ and $Q$, such that $P \cup Q = S$ and $P \cap Q = \emptyset$, the following bounds always hold:

$$\frac{c(S, V - S)}{|V - S|} \leq \alpha \leq \frac{c(P, Q)}{\min(|P|, |Q|)}$$

Proof.
Follows from following four Lemmas. \qed
Lemma

Let $s, r \in V$ be two nodes of $G$ and let $S$ be the community of $s$ with respect to $r$. Then, there exists a min-cut tree $T_G$ of $G$, and an edge $(a, b) \in T_G$, such that the removal of $(a, b)$ yields $S$ and $V - S$. 
Lemma

Let $s, r \in V$ be two nodes of $G$ and let $S$ be the community of $s$ with respect to $r$. Then, there exists a min-cut tree $T_G$ of $G$, and an edge $(a, b) \in T_G$, such that the removal of $(a, b)$ yields $S$ and $V - S$.

Proof.

Follows from Gomory-Hu Algorithm.
Start the algorithm by finding a minimum cut separating $s$ and $r$. Choose the cut $(S, V - S)$. □
Lemma

Let $T_G$ be a min-cut tree of a graph $G = (V, E)$, and let $(u, w)$ be an edge of $T_G$. Edge $(u, w)$ yields the cut $(U, W)$ in $G$, with $u \in U$, $w \in W$. Now, take any cut $(U_1, U_2)$ of $U$, so that $U_1$ and $U_2$ are non-empty, $u \in U_1$, $U_1 \cup U_2 = U$, and $U_1 \cap U_2 = \emptyset$. Then:

$$c(W, U_2) \leq c(U_1, U_2)$$
Lemma

Let $T_G$ be a min-cut tree of a graph $G = (V, E)$, and let $(u, w)$ be an edge of $T_G$. Edge $(u, w)$ yields the cut $(U, W)$ in $G$, with $u \in U$, $w \in W$. Now, take any cut $(U_1, U_2)$ of $U$, so that $U_1$ and $U_2$ are non-empty, $u \in U_1$, $U_1 \cup U_2 = U$, and $U_1 \cap U_2 = \emptyset$. Then:

$$c(W, U_2) \leq c(U_1, U_2)$$

Proof.

$(U, W)$ is a minimum $(u, w)$–cut.

$(U_1, W \cup U_2)$ is a $(u, w)$–cut.

Therefore,

$$c(U, W) \leq c(U_1, W \cup U_2)$$

$$c(U_1 \cup U_2, W) \leq c(U_1, W \cup U_2)$$

$$c(U_1, W) + c(U_2, W) \leq c(U_1, W) + c(U_1, U_2)$$

$$c(U_2, W) \leq c(U_1, U_2)$$
**Lemma**

Let $S$ be the community of $s$ in $G_\alpha$ with respect to $t$. For any non-empty $P$ and $Q$, such that $P \cup Q = S$ and $P \cap Q = \emptyset$, the following bound always holds

$$\alpha \leq \frac{c(P, Q)}{\min(|P|, |Q|)}$$
Lemma

Let $S$ be the community of $s$ in $G_\alpha$ with respect to $t$. For any non-empty $P$ and $Q$, such that $P \cup Q = S$ and $P \cap Q = \emptyset$, the following bound always holds

$$\alpha \leq \frac{c(P, Q)}{\min(|P|, |Q|)}$$

Proof.

Consider the $(s, t)$-cut $(S, V - S \cup \{t\})$. W.l.o.g., assume $s \in P$. By previous lemma, $c(Q, V - S \cup \{t\}) \leq c(P, Q)$ But $c(Q, V - S \cup \{t\}) \geq \alpha \cdot |Q|$ Therefore, $\alpha \cdot \min(|P|, |Q|) \leq c(P, Q)$ □
Lemma

Let $S$ be the community of $s$ in $G_\alpha$ with respect to $t$. Then, the following bound always holds:

$$\frac{c(S, V - S)}{|V - S|} \leq \alpha$$
Lemma

Let \( S \) be the community of \( s \) in \( G_\alpha \) with respect to \( t \). Then, the following bound always holds:

\[
\frac{c(S, V - S)}{|V - S|} \leq \alpha
\]

Proof.

\((S, V - S \cup \{t\})\) is a minimum \((s, t)\)-cut in \( G_\alpha \).

\( V - S \) and \( \{t\} \) form a partition of \( V - S \cup \{t\} \).

So, \( c(S, V - S) \leq c(V - S, \{t\}) = \alpha \cdot |V - S| \). \( \square \)
**CutClusteringAlgorithm** \((G(V, E), \alpha)\)

Let \(V' = V \cup t\)

Construct \(G_\alpha\)

Calculate the minimum-cut tree \(T'\) of \(G_\alpha\)

Remove \(t\) from \(T'\)

Return all connected components as clusters of \(G\)
Figure: Clusters for $\alpha = 1$ and $\alpha = 2$.

Figure: Clusters for $\alpha = 4$ and $\alpha = 5$. 
Lemma

Let $v_1, v_2 \in V$ and $S_1, S_2$ be their communities with respect to $t$ in $G_{\alpha}$. Then either $S_1$ and $S_2$ are disjoint or one is a subset of the other.
Lemma

Let $v_1, v_2 \in V$ and $S_1, S_2$ be their communities with respect to $t$ in $G_{\alpha}$. Then either $S_1$ and $S_2$ are disjoint or one is a subset of the other.

Proof.

Let $(S_1, V - S_1 \cup \{t\})$ be the initial partition in constructing a minimum cut tree.

Let $(a, b)$ be the edge corresponding to the cut.

If $s_2 \in S_1$, the path from $s_1$ to $t$ uses $(a, b)$.

So, a minimum $(s_2, t)$–cut is contained in $S_1$.

If $s_2 \not\in S_1$, there is a minimum $(s_2, t)$–cut disjoint from $S_1$. 

\qed
Heuristic

Sufficient to find neighbors of $t$ in the minimum cut tree. No need to calculate an entire min-cut tree of $G_\alpha$. By previous lemma, if we have a community $S$, no need to further branch.
Heuristic

Sufficient to find neighbors of $t$ in the minimum cut tree. No need to calculate an entire min-cut tree of $G_\alpha$. By previous lemma, if we have a community $S$, no need to further branch.

Heuristic:

Let $c(v) = c(\{v\}, V - \{v\})$. Sort nodes in decreasing order of $c(v)$. Calculate min-cuts between $t$ and ‘unmarked’ nodes in the given order. Reduces number of max-flow computations to almost the number of clusters.
Nesting Property

Observation

• For $\alpha$ small, communities are large (i.e., one large cluster)
• As $\alpha \to \infty$, communities become singleton nodes
Nesting Property

Observation

• For $\alpha$ small, communities are large (i.e., one large cluster)
• As $\alpha \to \infty$, communities become singleton nodes

Lemma (The Nesting Property)

For a source $s$ in $G_{\alpha_i}$, where $\alpha_i \in \{\alpha_1, \ldots, \alpha_{\text{max}}\}$, such that $\alpha_1 < \alpha_2 < \cdots < \alpha_{\text{max}}$, the communities $S_1, \ldots, S_{\text{max}}$ are such that $S_1 \subseteq S_2 \subseteq \ldots \subseteq S_{\text{max}}$, where $S_i$ is the community of $s$ with respect to $t$ in $G_{\alpha_i}$. 
Heirarchical-CutClustering (G(V, E))

Let $G^0 = G$

For $(i = 0; ; i++)$
  
  Set new, smaller value $a_i$
  
  Call CutCluster_Basic($G^i, \alpha_i$)
  
  If ((clusters returned are of desired number and size) or
    (clustering failed to create non-trivial clusters))
    
    break
  
  Contract clusters to produce $G^{i+1}$

Return all clusters at all levels
Experimental Results

Algorithm applied to

CiteSeer

- A digital library for scientific literature.
- Viewed as graph with documents as nodes and directed arcs denoting citations.
Experimental Results

Algorithm applied to

CiteSeer

- A digital library for scientific literature.
- Viewed as graph with documents as nodes and directed arcs denoting citations.

The Open Directory Project, *dmoz*

- Web pages as nodes, edges corresponding to hyperlinks (links between web-pages of same domain ignored)
Experimental Results

Algorithm applied to

CiteSeer

- A digital library for scientific literature.
- Viewed as graph with documents as nodes and directed arcs denoting citations.

The Open Directory Project, *dmoz*

- Web pages as nodes, edges corresponding to hyperlinks (links between web-pages of same domain ignored)

The 9/11 Community

- Identifying web pages related to 9/11
Experimental Results

Problem: Algorithm (and minimum cut trees) defined for undirected graphs

Fix: Normalize over outbound arcs for each node
Experimental Results

Problem: Algorithm (and minimum cut trees) defined for undirected graphs
Fix: Normalize over outbound arcs for each node

Outcomes

- Good hierarchical clustering for both CiteSeer and dmoz.
- Concentration of topics within 9/11 community