Alpha-Beta Community

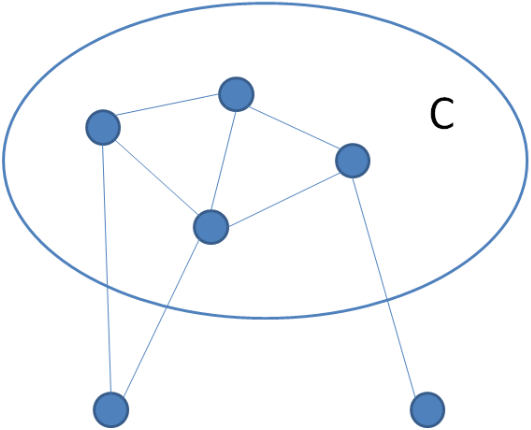
Supasorn Suwajanakorn

March 15, 2010

Definition from Nina's paper

Definition 1. Given a graph $G = (V, E)$, where every vertex has a self-loop $C \subset V$ is an (α, β)-cluster if

1) **Internally Dense** $\forall v \in C, |E(v, C)| \geq \beta|C|$
2) **Externally Sparse** $\forall u \in V \setminus C, |E(u, C)| \leq \alpha|C|$



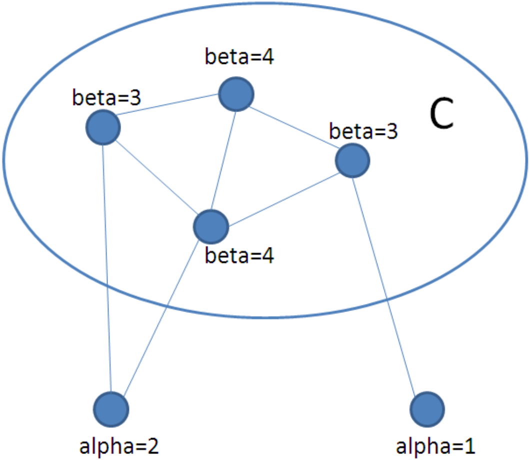
Example of (alpha = 0.5, beta = 0.75)-cluster

Our modified definition

Definition 2. Given a graph $G = (V, E)$, where every vertex has a self-loop $C \subset V$

1) $\forall v \in C, \beta(v) = |E(v, C)|$
2) $\forall v \notin C, \alpha(v) = |E(v, C)|$
3) $\beta(C) = \min_{v \in C} \beta(v)$
4) $\alpha(C) = \max_{v \notin C} \alpha(v)$

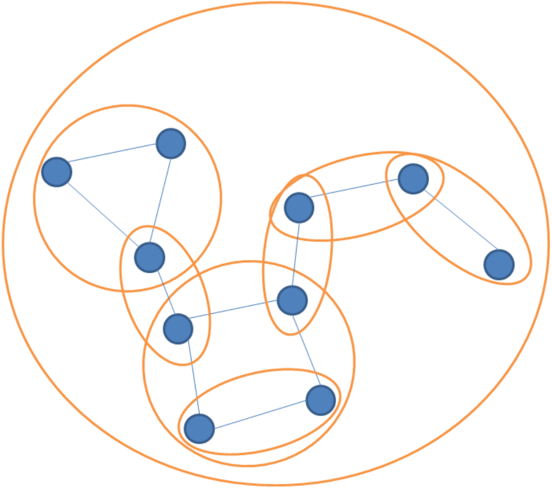
* Definition 1 uses α, β as a fraction of C
* Definition 2 uses α, β as a number of vertices in C



Example of (alpha = 0.5, beta = 0.75)-cluster

Trivial community

We're only interested in large size communities.



Our goal

We want to find a non-trivial (α, β) community such that $\beta > \alpha$.

Proof of existence of non-trivial community

Proof: Given a graph $G = (V, E)$. Let $K = V$ and we repeatedly remove a vertex with lowest beta value from K. Let r_1 be the first vertex removed from K whose $\beta(r_1) = \rho$. Once r_1 is removed, $\alpha(r_1) = \rho - 1$ since its self-loop is no longer counted. Suppose K is still not a community i.e. $\alpha(K) = \rho - 1 = \beta(K)$, then r_1 must be connected to a vertex r_2 and removal of r_1 must have decreased $\beta(r_2)$ by one.

If r_i will be removed from K by the algorithm, $\beta(r_i)$ before removal needs to equal $\rho - (i-1)$ which implies r_i has initial beta value ρ and is connected to all $r_1, r_2, \ldots, r_{i-1}$ already outside K. (By induction) if the last r_n, where $n = |V|$, is removed from K, all of the removed vertices $r_1, r_2, \ldots, r_n$ form a clique.

Swapping-Algorithm

C=SWAPPING($G = (V, E), C$)

while $\beta(C) < \alpha(C)$
    $a \leftarrow a \notin C$ whose $\alpha(a)$ is maximum
    $b \leftarrow b \in C$ whose $\beta(b)$ is minimum
    $C \leftarrow (C - \{b\}) \cup \{a\}$
return C

Swapping-Algorithm

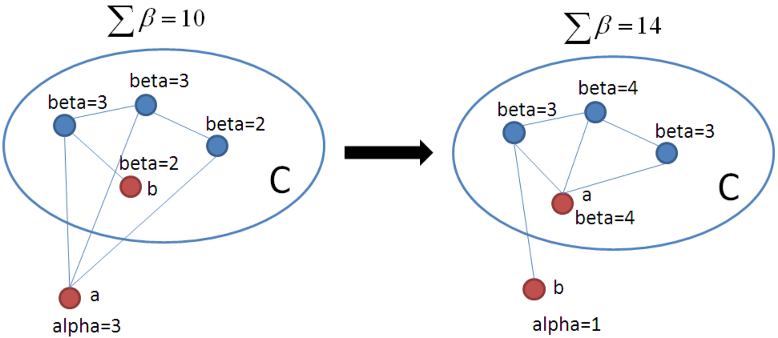
Claim 1. In the swapping algorithm, $\sum_{v \in C} \beta(v)$ is strictly increasing

Claim 2. The swapping algorithm always terminates and swapping any pair of vertices will not further increase $\sum_{v \in C} \beta(v)$.

Claim 3. The swapping algorithm returns C where $\beta(C) \geq \alpha(C)$.

Swapping-Algorithm: $(a, b) \notin E$

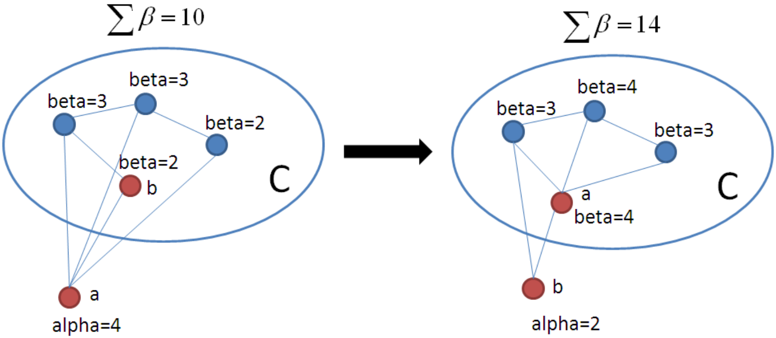
$\alpha(a) = 3, \beta(b) = 2$



If $(a, b) \notin E$, $\sum_{v \in C} \beta(v)$ increases by
$2\alpha(a) + 1 - (2\beta(b) - 1) = 2(\alpha(a) - \beta(b)) + 2$

Swapping-Algorithm: $(a, b) \in E$

$\alpha(a) = 4, \beta(b) = 2$



If $(a, b) \in E$, $\sum_{v \in C} \beta(v)$ increases by
$2\alpha(a) - 1 - (2\beta(b) - 1) = 2(\alpha(a) - \beta(b))$

Swapping-Algorithm

If $(a, b) \notin E$, $\sum_{v \in C} \beta(v)$ increases by $2(\alpha(a) - \beta(b)) + 2$
If $(a, b) \in E$, $\sum_{v \in C} \beta(v)$ increases by $2(\alpha(a) - \beta(b))$

$\sum_{v \in C} \beta(v)$ will always increase if
$(a, b) \notin E \wedge \alpha(a) \geq \beta(b)$ or
$(a, b) \in E \wedge \alpha(a) > \beta(b)$

Let A be the set of highest-alpha vertices not in C
Let B be the set of lowest-beta vertices in C

This implies that at the end of *Swapping* algorithm, $\beta(C) > \alpha(C)$ or $\beta(C) = \alpha(C)$ and the edges between A and B form a **biclique**.

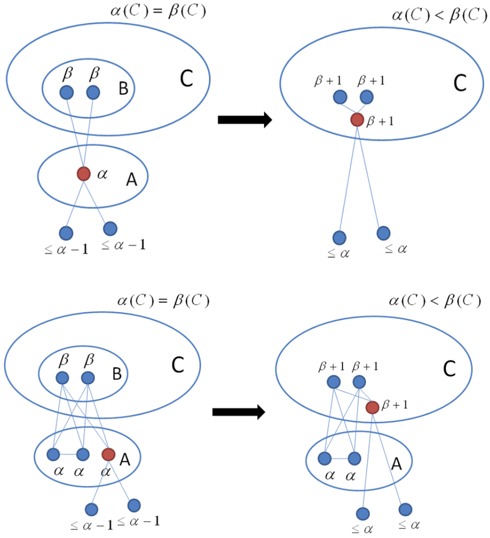
Community-Algorithm

C=COMMUNITY($G = (V, E)$)

$C \leftarrow$ any subset of V of constant size
while $\beta(C) \leq \alpha(C)$
  $C \leftarrow$ SWAPPINGALGORITHM(G, C)
  if $\beta(C) > \alpha(C)$ **then**
    return C
  $A \leftarrow$ set of highest-alpha vertices not in C
  $B \leftarrow$ set of lowest-beta vertices in C
  if $|A| = 1$ **then**
    return $C \cup A$
  if there exists $a_i \in A$ s.t. $(a_i, a_j) \notin E$ for all $a_j \in A$ **then**
    return $C \cup \{a_i\}$
  else
    $(a_j, a_k) \leftarrow$ any edge $(a_j, a_k) \in E$ s.t. $a_j, a_k \in A$
    $C \leftarrow C \cup \{a_j, a_k\}$
    if $\beta(C) \leq \alpha(C)$ **then**
      $v \leftarrow$ any vertex $v \in \alpha(C)$ s.t $(v, a_j) \in E$ **or** $(v, a_k) \in E$
      $C \leftarrow C \cup \{v\}$

return C

Community-Algorithm



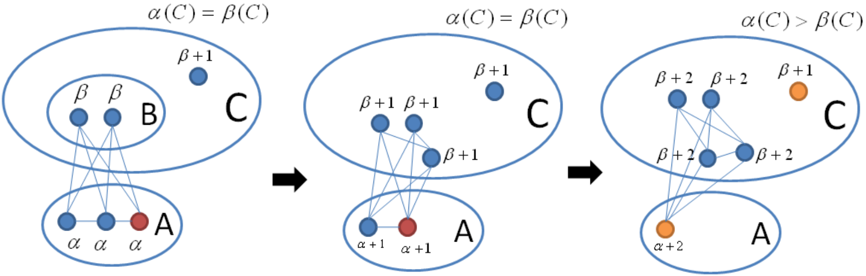
C=COMMUNITY$(G = (V, E))$

$C \leftarrow$ any subset of V of constant size
while $\beta(C) \le \alpha(C)$
  $C \leftarrow$ SWAPPINGALGORITHM(G, C)
  if $\beta(C) > \alpha(C)$ **then**
    return C
  $A \leftarrow$ set of highest-alpha vertices no
  $B \leftarrow$ set of lowest-beta vertices in C
  if $|A| = 1$ **then**
    return $C \cup A$
  if there exists $a_i \in A$ s.t. $(a_i, a_j) \notin$.
    return $C \cup \{a_i\}$
  else
    $(a_j, a_k) \leftarrow$ any edge $(a_j, a_k) \in$
    $C \leftarrow C \cup \{a_j, a_k\}$
    if $\beta(C) \le \alpha(C)$ **then**
      $v \leftarrow$ any vertex $v \in \alpha(C)$ s.
      $C \leftarrow C \cup \{v\}$

return C

Community-Algorithm

Problem



Our work

Convergence For 6-regular degree graph, we can prove termination.

Proof: Consider $d|C| - \sum_{v \in C} \beta(v)$ where d is the number of highest degree in the graph. If we add 3 vertices to C, $\sum_{v \in C} \beta(v)$ will increase by $6\alpha + 7$. Since $\alpha \geq 2$, $d|C| - \sum_{v \in C} \beta(v)$ will decrease by at least $d(3) - (6(2) + 7)$. If $d \leq 6$, $d(3) - (6(2) + 7) \leq -1$. Therefore, $d|C| - \sum_{v \in C} \beta(v)$ will eventually reach 0.

We need a prove of convergence for any graph whose $d > 6$.

Prof. Hopcroft's experiments

Random graph In a random regular-degree graph ($d/|V| = [0.02, 0.04]$), there are an enormous number of communities.

Real graph In a real graph with two strongly-knit core clusters, the algorithm terminates in most cases and returns a community concentrated around the cores but in a few cases, found a biclique.
Additionally, the cores contain many smaller communities that satisfy $\beta > \alpha$.

Constructed graph In a graph where there are 3 strongly-knit clusters connected to each other by a few edges, if we start with random initial vertices, the algorithm returns a community with an approximately equal number of vertices from the 3 clusters.

Future work

More strict definition to prevent disjoint components, increase gap between β and α, expand smaller community to its super-set community

Alternative definition to better represent communities in real graphs

Proof of convergence for the algorithm for any kind of graph