CoreCluster: A Degeneracy Based Graph Clustering Framework

Christos Giatsidis
Ecole Polytechnique
giatsidis@lix.polytechnique.fr

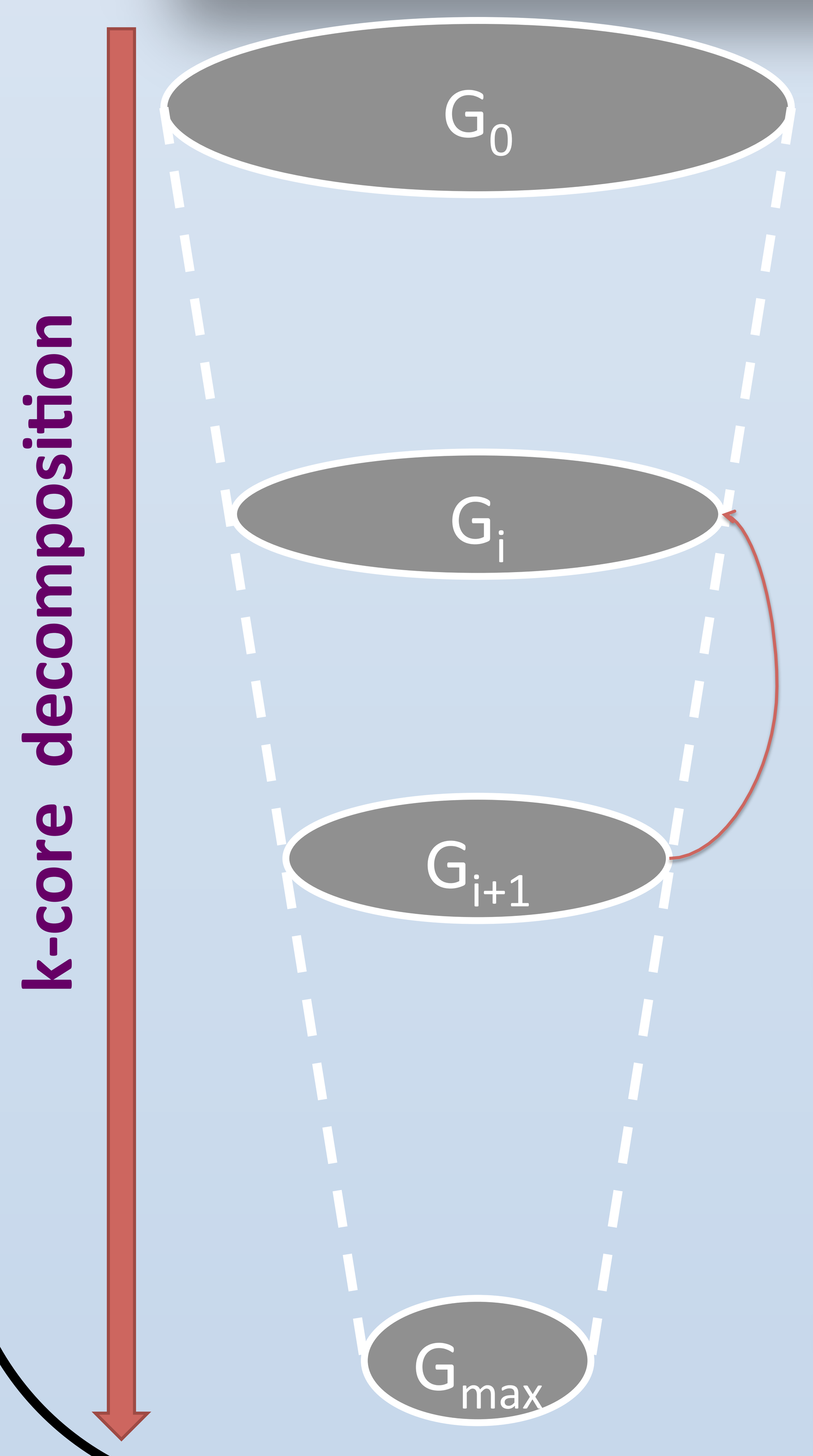
Fragkiskos D. Malliaros
Ecole Polytechnique
fmalliaros@lix.polytechnique.fr

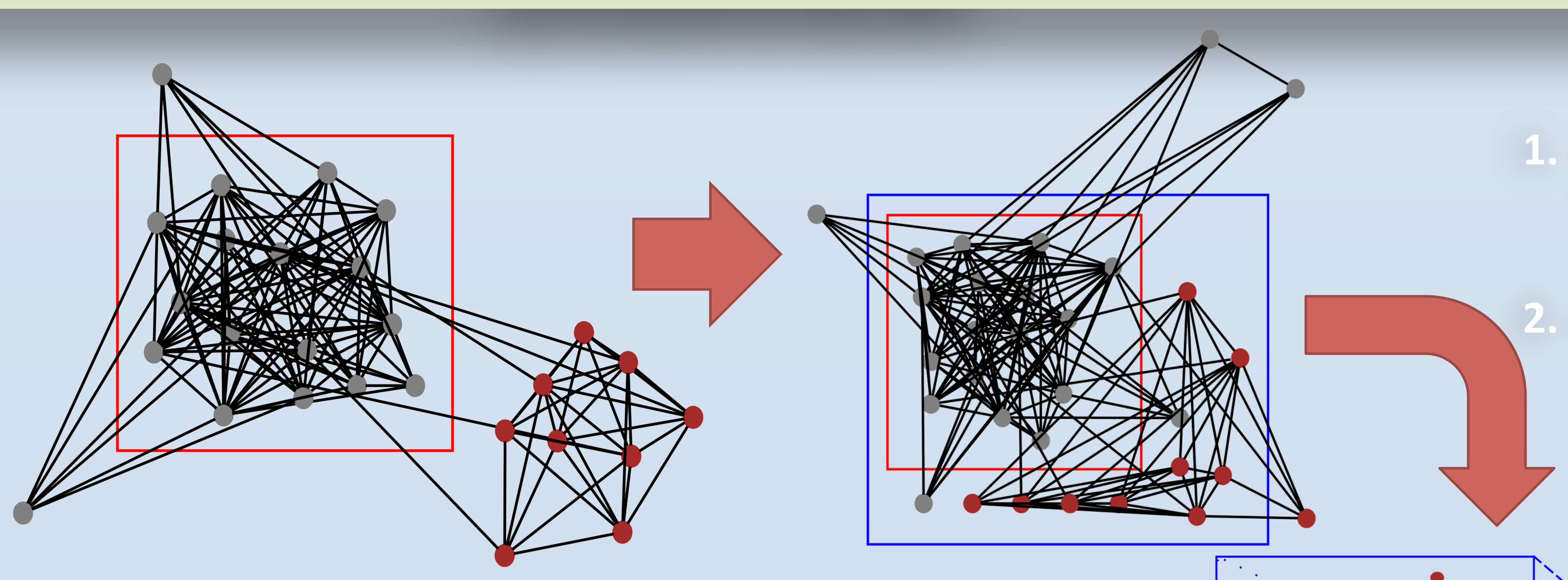
Dimitrios M. Thilikos
CNRS, LIRMM, and UoA
sedthilk@thilikos.info

Michalis Vazirgiannis
Ecole Polytechnique and AUEB
mvazirg@lix.polytechnique.fr

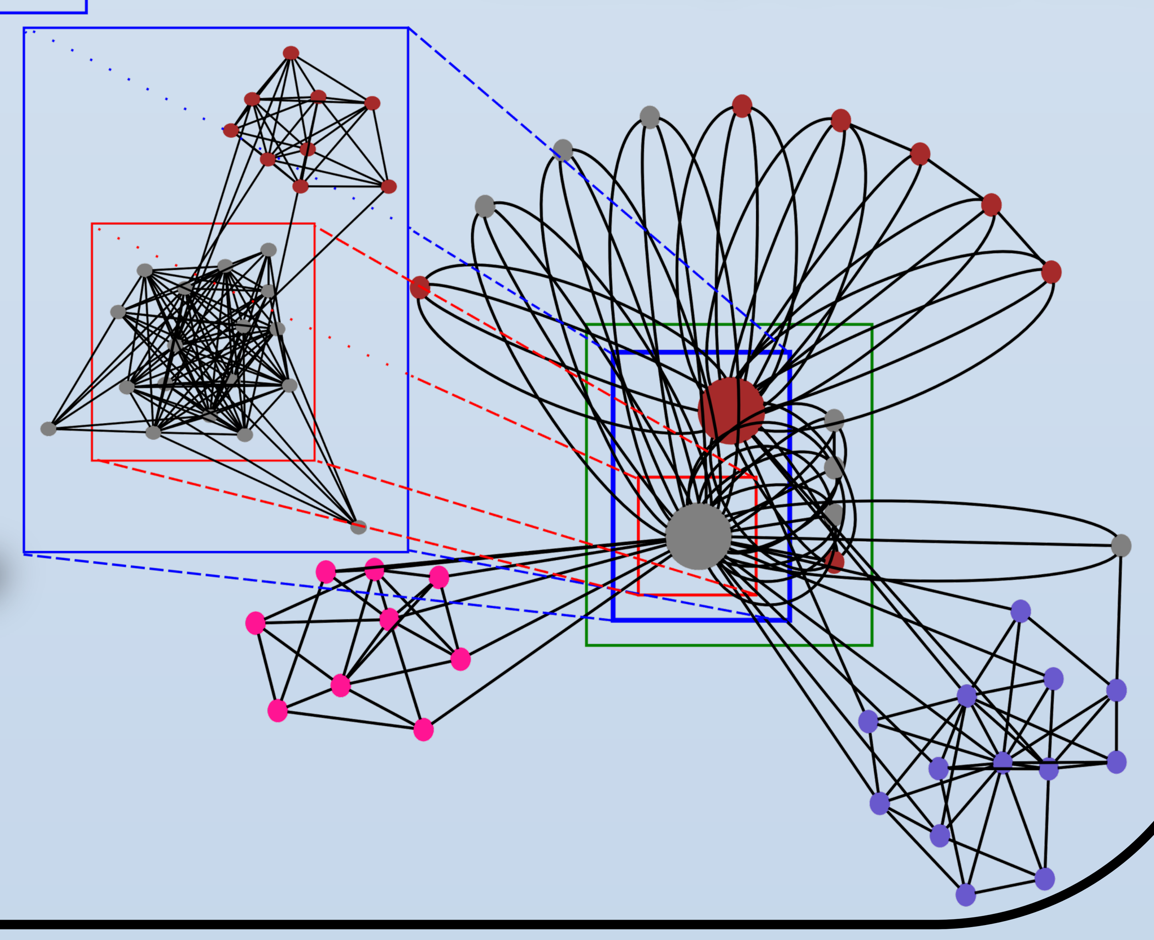
Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI-14), Québec City, Canada, 2014

OVERVIEW





1. Assign with a simple function nodes to existing clusters (from (i+1)-core)
2. Apply C to nodes less connected to the existing clusters than sub-graph nodes of {(i+1)-core}-{i-core}



Core decomposition partitions the graph in a hierarchical nested manner

Assume an "expensive" algorithm C e.g., Spectral Clustering as a black box

It is less expensive to compute in sections of the data separately

Utilize the vertical partition of k-core decomposition as incremental input to C

THE ALGORITHM

Procedure CoreCluster(G).
Input: A graph G.
Output: A partition of $V(G)$ into clusters.

1. $k := \delta^*(G)$.
2. $q := 0$.
3. Let $V_k, \ldots, V_0$ be the core expansion sequence of G.
4. For $i = 0, \ldots, k$, let G_i be the i-core of G,
5. Let $S_k = V_k$.
6. Let $\mathcal{A}_k = \{C_1^k, \ldots, C_{\rho_k}^k\} = \mathbf{Cluster}(G[S_k])$.
7. **for** $i = k-1$ **to** 0 **do**
8. $\quad S_i =$ *Select*$(G_i, \mathcal{A}_k \cup \ldots \cup \mathcal{A}_{i+1}, V_i)$,
9. $\quad$ let $\mathcal{A}_i = (C_1^i, \ldots, C_{\rho_i}^i) = \mathbf{Cluster}(G[S_i])$.
10. **Return** $\mathcal{A}^k \cup \cdots \cup \mathcal{A}^0$.

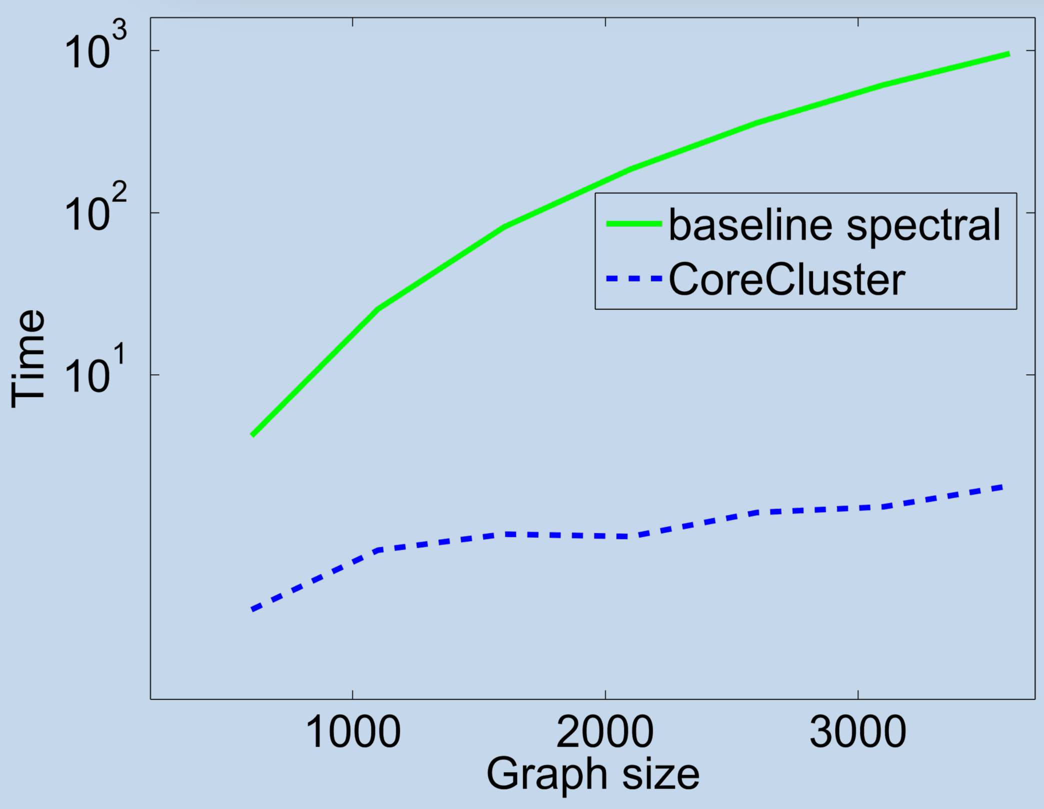
Procedure *Select*$(G, \mathcal{F}, V)$.
Input: A candidate triple $(G, \mathcal{F}, V)$
Output: A subset S of V and a partition $\mathcal{F}'$ of $V(\mathcal{F}) \cup (V \setminus S)$.

1. while $\mathbf{P}^{\alpha,\beta}(v)$ is true for some $v \in V$,
2. $\quad$ set $\mathcal{F} \leftarrow (\mathcal{F} \setminus \{C\}) \cup \{C \cup \{v\}\}$ where
3. $\quad\quad$ C is the certificate of v
4. $\quad$ and set $V \leftarrow V \setminus \{v\}$.
5. set $V^1 = N_G(V(\mathcal{F}))$ and $V^2 = V \setminus (V^1 \cup \mathcal{F})$.
6. if V^2 is either empty or an independent set of G,
7. $\quad$ then
8. $\quad\quad$ $\mathcal{F} \leftarrow$ *assign*$(G, \mathcal{F}, V, V^1)$
9. $\quad\quad$ $\mathcal{F} \leftarrow$ *assign*$(G, \mathcal{F}, V, V^2)$
10. $\quad\quad$ return $\emptyset$
11. $\quad$ else return $V^1 \cup V^2$.

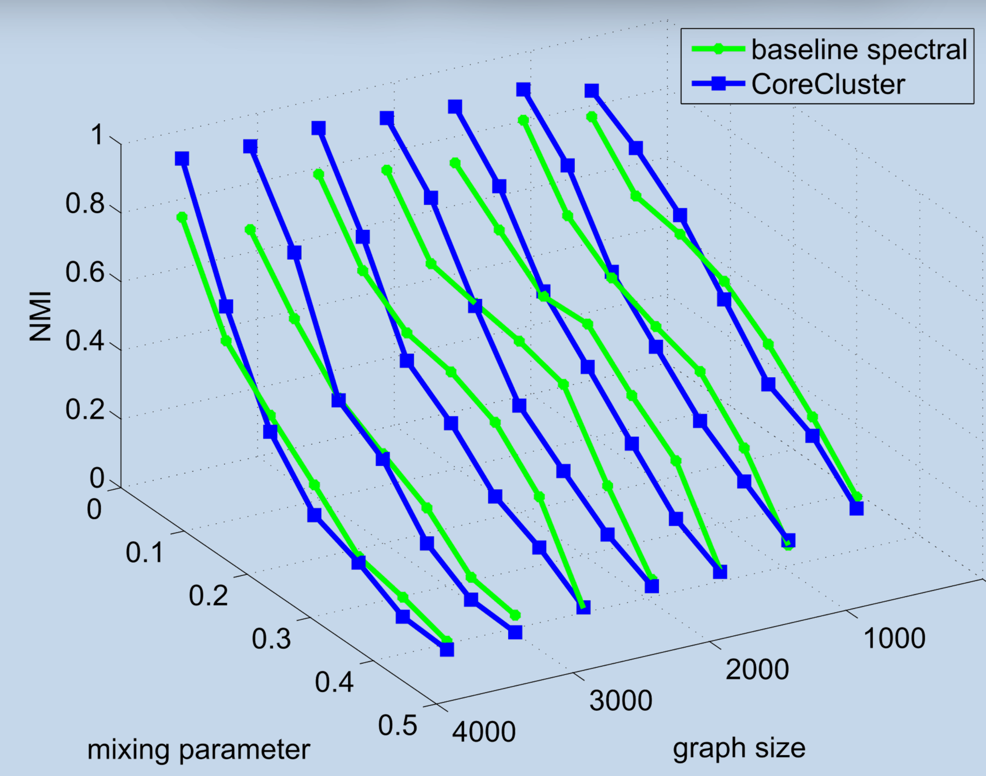
Procedure *assign*$(G, \mathcal{F}, V, S)$.
Input: A candidate triple $(G, \mathcal{F}, V)$ and a subset S of V
Output: A partition $\mathcal{F}$

1. while $S \neq \emptyset$,
2. $\quad$ let $l = \max\{\mathbf{span}(v) \mid v \in S\}$
3. $\quad$ let $L = \{v \in S \mid \mathbf{span}(v) = l\}$
4. $\quad$ for every $v \in L$,
5. $\quad\quad$ set $C' = C \cup \{v\}$ where $C = \mathbf{argspan}(v)$
6. $\quad\quad$ set $S \leftarrow S \setminus \{v\}$
7. $\quad\quad$ set $\mathcal{F} \leftarrow (\mathcal{F} \setminus \{C\}) \cup \{C'\}$
8. return $\mathcal{F}$.

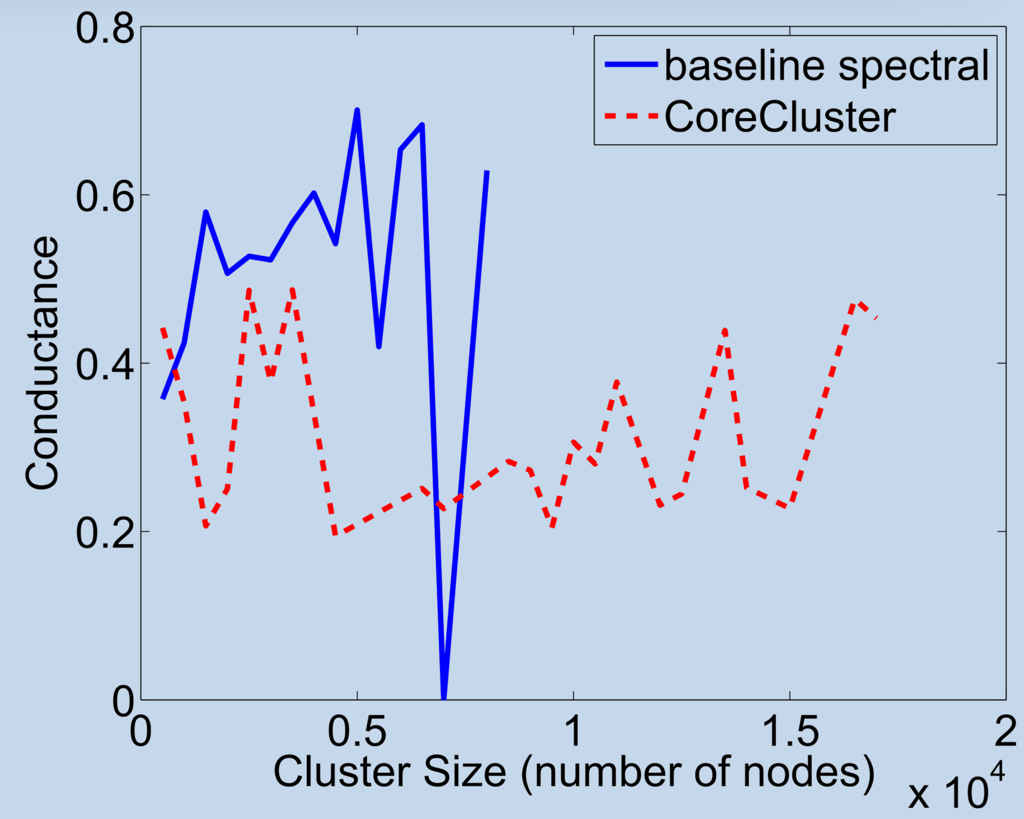
THE RESULTS



Execution time



Artificial data with ground truth



Real data (Facebook)

École Polytechnique, Université Paris-Saclay · CNRS · University of Athens · AUEB · DaScim · www.lix.polytechnique.fr/dascim · digiteo: Research in science and technology of information · Google Doctoral Fellowship · NSRF 2007-2013 programme for development