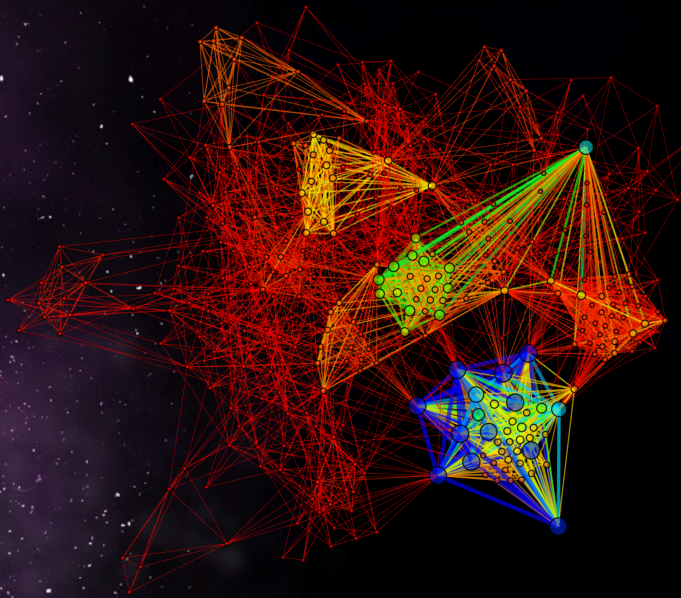
Fast Hierarchical Graph Clustering in Linear-Time

Ryan A. Rossi[1] | Nesreen K. Ahmed[2], Eunyee Koh[1], and Sungchul Kim[1]

[1] Adobe Research

[2] Intel Labs



MAKE IT AN
EXPERIENCE

Motivation

Communities are sets of densely connected nodes

- Important for many applications

Most previous work has two main limitations:

1. Most previous work does not address the hierarchical community detection problem
2. Inefficient for large graphs with a runtime that is not linear in the number of edges

This work proposes an approach called hLP that addresses both these limitations.

- Fast linear-time approach for revealing hierarchical communities in large graphs

Problem

Given G, hLP computes

(i) a hierarchy of communities $\mathbb{H} = \{\mathcal{C}^1, \ldots, \mathcal{C}^L\}$ s.t. $|\mathcal{C}^{t-1}| > |\mathcal{C}^t|, \forall t$

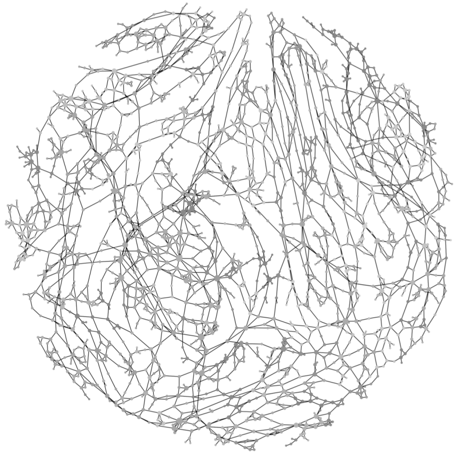
(ii) a hierarchy of super graphs $G_1, \ldots, G_t, \ldots, G_L$

$$V_t \leftarrow \mathcal{C}^t$$
$$E_t = \{(i,j) : r \in C_i^t, s \in C_j^t \wedge (r,s) \in E_{t-1} \wedge i \neq j\}$$

Desired properties:

- Finds "good" high quality communities
- Fast algorithm for large graphs – linear time and space complexity
- Summarizes structure at various levels of granularity

Overview



Two main steps:

1. Label propagation
2. Super graph construction

Repeat 1-2 until convergence

Algorithm 1 hLP

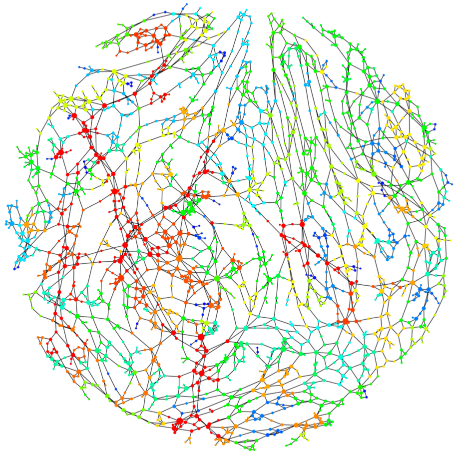
Input: a graph $G = (V, E)$

Output: hierarchical communities $\mathbb{H} = \{\mathcal{C}^1, \ldots, \mathcal{C}^L\}$

```
Set G_0 ← G to be the initial graph and t ← 0
repeat
    t ← t + 1
    C^t ← LabelProp(G_{t-1})
    G_t = (V_t, E_t) ← CreateSuperGraph(G_{t-1}, C^t)
until |V_t| < 2                          ▷ Stop when no nodes to combine
```

$G_0 \leftarrow G$; $t \leftarrow t+1$; $\mathcal{C}^t \leftarrow \textsc{LabelProp}(G_{t-1})$; $G_t = (V_t, E_t) \leftarrow \textsc{CreateSuperGraph}(G_{t-1}, \mathcal{C}^t)$; until $|V_t| < 2$

Overview



Two main steps:

1. Label propagation
2. Super graph construction

Repeat 1-2 until convergence

Algorithm 1 hLP

Input: a graph $G = (V, E)$

Output: hierarchical communities $\mathbb{H} = \{\mathcal{C}^1, \ldots, \mathcal{C}^L\}$

Set $G_0 \leftarrow G$ to be the initial graph and $t \leftarrow 0$

repeat

$\quad t \leftarrow t + 1$

$\quad \mathcal{C}^t \leftarrow \textsc{LabelProp}(G_{t-1})$

$\quad G_t = (V_t, E_t) \leftarrow \textsc{CreateSuperGraph}(G_{t-1}, \mathcal{C}^t)$

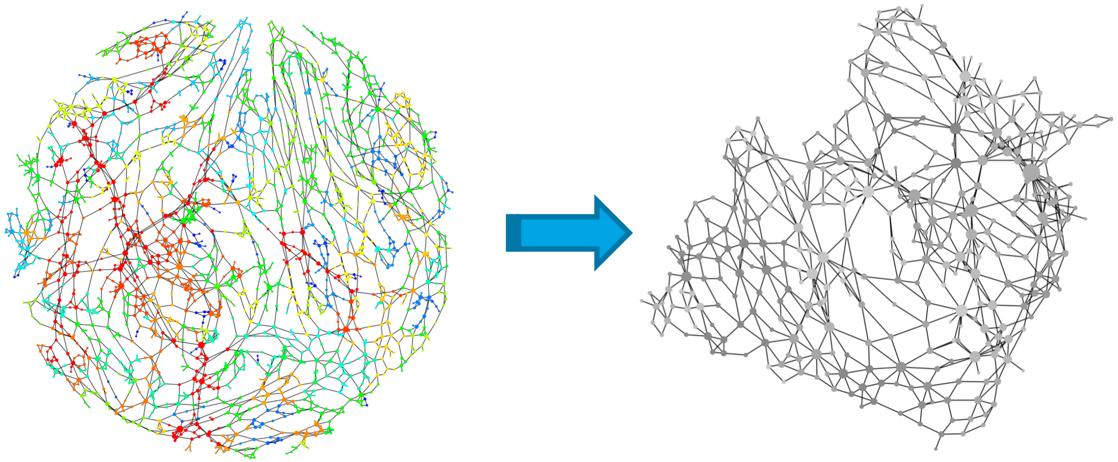
until $|V_t| < 2$ ▷ Stop when no nodes to combine

Overview

Two main steps:

1. Label propagation
2. Super graph construction

Repeat 1-2 until convergence



Algorithm 1 hLP

Input: a graph $G = (V, E)$

Output: hierarchical communities $\mathbb{H} = \{\mathcal{C}^1, \ldots, \mathcal{C}^L\}$

Set $G_0 \leftarrow G$ to be the initial graph and $t \leftarrow 0$

repeat

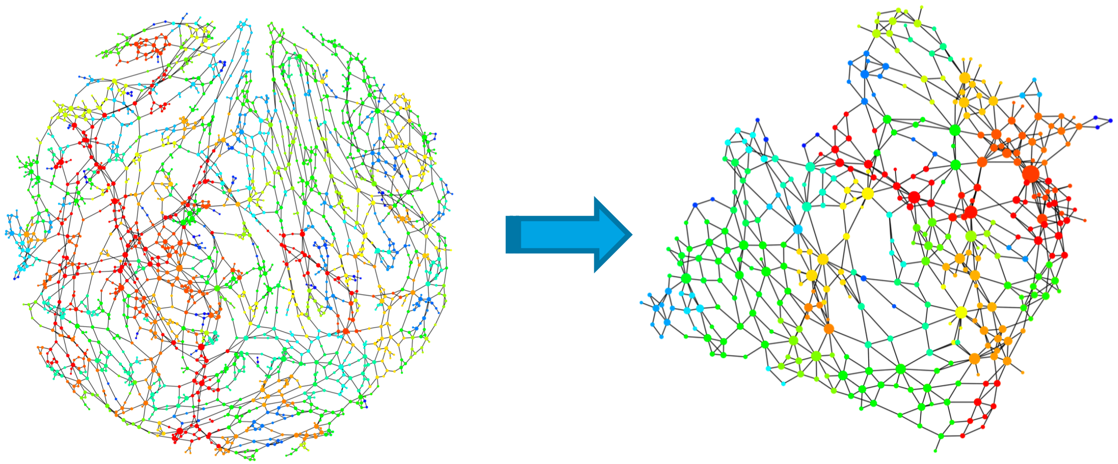
  $t \leftarrow t + 1$

  $\mathcal{C}^t \leftarrow \textsc{LabelProp}(G_{t-1})$

  $G_t = (V_t, E_t) \leftarrow \textsc{CreateSuperGraph}(G_{t-1}, \mathcal{C}^t)$

until $|V_t| < 2$ ▹ Stop when no nodes to combine

Overview



Two main steps:

1. Label propagation
2. Super graph construction

Repeat 1-2 until convergence

Algorithm 1 hLP

Input: a graph $G = (V, E)$

Output: hierarchical communities $\mathbb{H} = \{\mathcal{C}^1, \ldots, \mathcal{C}^L\}$

Set $G_0 \leftarrow G$ to be the initial graph and $t \leftarrow 0$

repeat

  $t \leftarrow t + 1$

  $\mathcal{C}^t \leftarrow \textsc{LabelProp}(G_{t-1})$

  $G_t = (V_t, E_t) \leftarrow \textsc{CreateSuperGraph}(G_{t-1}, \mathcal{C}^t)$

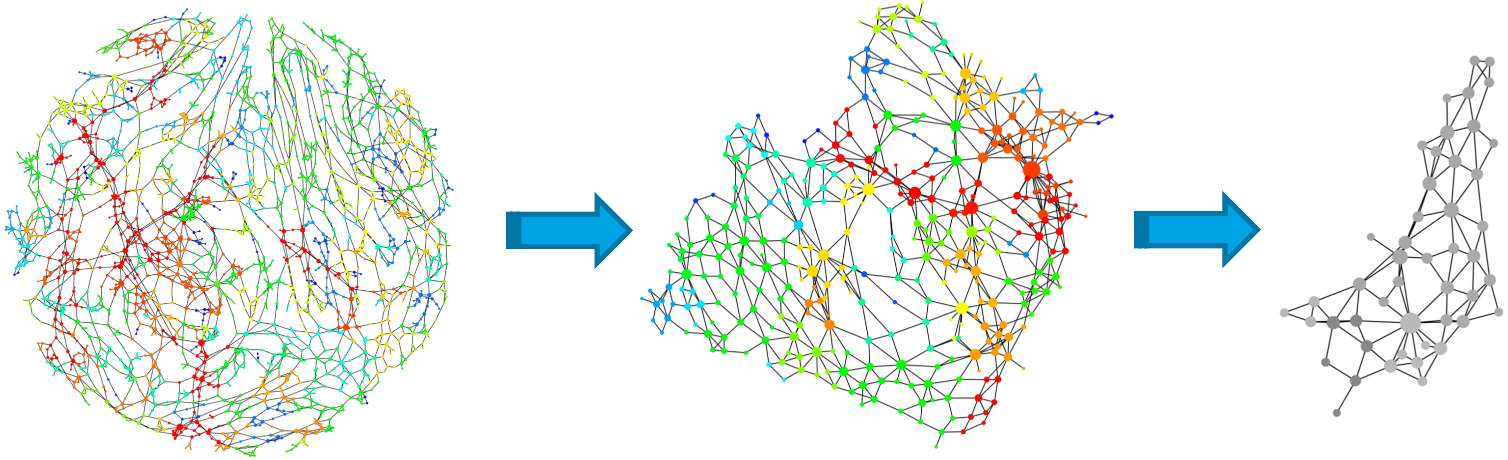
until $|V_t| < 2$ ▹ Stop when no nodes to combine

Overview

Two main steps:

1. Label propagation
2. Super graph construction

Repeat 1-2 until convergence



Algorithm 1 hLP

Input: a graph $G = (V, E)$

Output: hierarchical communities $\mathbb{H} = \{\mathcal{C}^1, \ldots, \mathcal{C}^L\}$

Set $G_0 \leftarrow G$ to be the initial graph and $t \leftarrow 0$

repeat

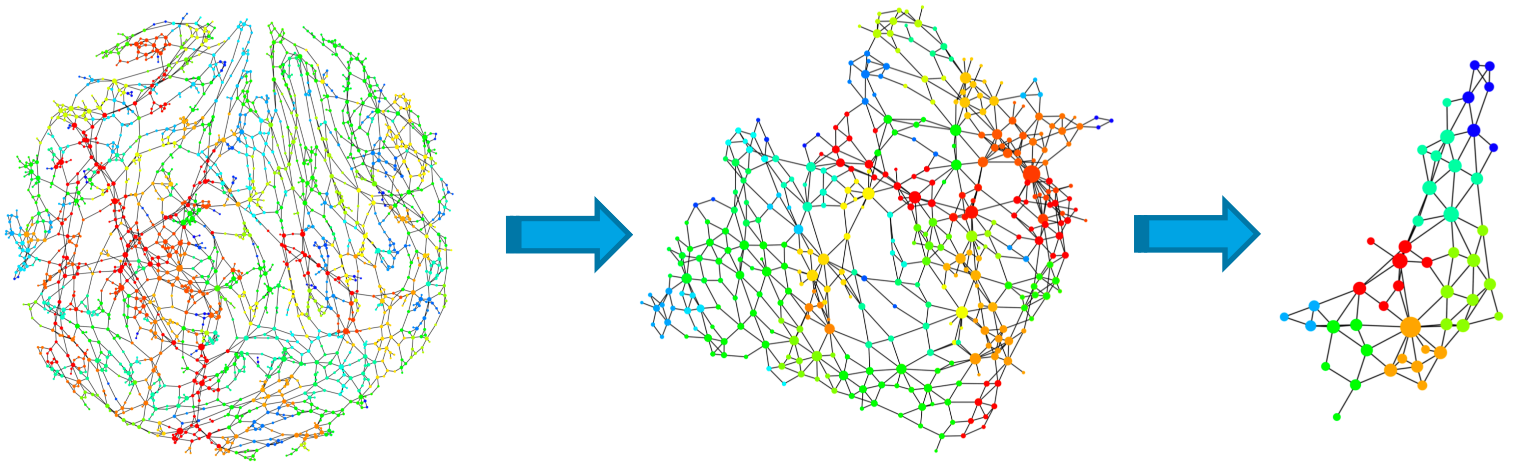
    $t \leftarrow t + 1$

    $\mathcal{C}^t \leftarrow \text{LabelProp}(G_{t-1})$

    $G_t = (V_t, E_t) \leftarrow \text{CreateSuperGraph}(G_{t-1}, \mathcal{C}^t)$

until $|V_t| < 2$ ▷ Stop when no nodes to combine

Overview



Two main steps:

1. Label propagation
2. Super graph construction

Repeat 1-2 until convergence

Time Complexity:

$$O(LTM)$$

L = number of layers
T = number of iterations
M = number of edges

Algorithm 1 hLP

Input: a graph $G = (V, E)$
Output: hierarchical communities $\mathbb{H} = \{\mathcal{C}^1, \ldots, \mathcal{C}^L\}$

```
Set G_0 ← G to be the initial graph and t ← 0
repeat
    t ← t + 1
    C^t ← LabelProp(G_{t−1})
    G_t = (V_t, E_t) ← CreateSuperGraph(G_{t−1}, C^t)
until |V_t| < 2                              ▷ Stop when no nodes to combine
```

Overview

Two main steps:

1. Label propagation
2. Construct super graph

Repeat 1-2 until convergence

Space Complexity:

$$O(L(M + N))$$

L = number of layers
M = number of edges
N = number of nodes

Algorithm 1 hLP

Input: a graph $G = (V, E)$
Output: hierarchical communities $\mathbb{H} = \{\mathcal{C}^1, \ldots, \mathcal{C}^L\}$

Set $G_0 \leftarrow G$ to be the initial graph and $t \leftarrow 0$
repeat
  $t \leftarrow t + 1$
  $\mathcal{C}^t \leftarrow \textsc{LabelProp}(G_{t-1})$
  $G_t = (V_t, E_t) \leftarrow \textsc{CreateSuperGraph}(G_{t-1}, \mathcal{C}^t)$
until $|V_t| < 2$ ▷ Stop when no nodes to combine

Algorithm 2 Create Super Graph

Input: a graph $G_{t-1} = (V_{t-1}, E_{t-1})$, communities $\mathcal{C}^t$ from G_{t-1}
Output: community (super) graph $G_t = (V_t, E_t)$ for layer t

$V_t \leftarrow \mathcal{C}^t = \{C_1, \ldots, C_k\}$ *and* $E_t \leftarrow \emptyset$
Let **c** be the community assignment vector where $c_i = k$ if $i \in C_k$
parallel for $i \in V_{t-1}$ **do**
  for $j \in \Gamma_i$ **do** ▷ Neighbor of vertex i
    if $c_i \neq c_j$ **and** $(c_i, c_j) \notin E_t$ **then**
      $E_t \leftarrow E_t \cup (c_i, c_j)$

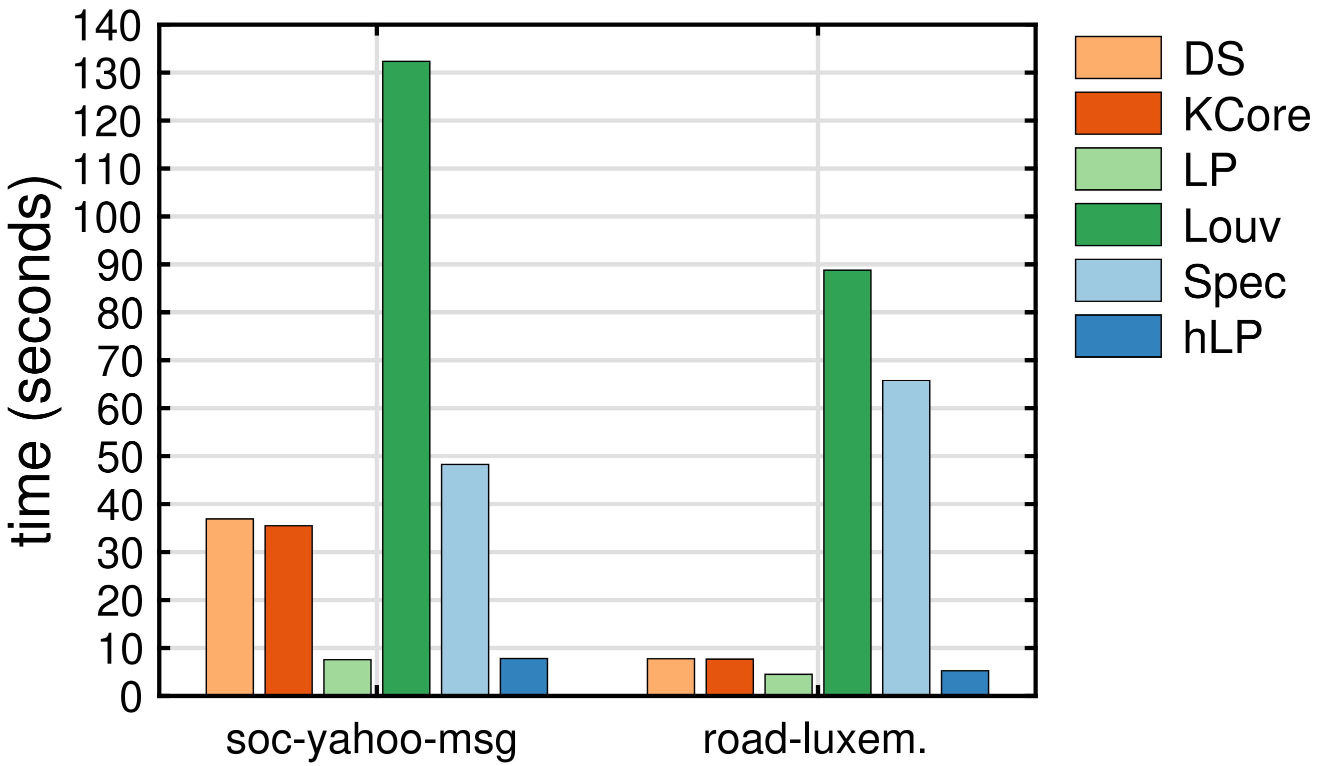
Results

- Use modularity for evaluation

Table 1: Quantitative evaluation using modularity.

	DS	KCore	LP	Louv	Spec	hLP
soc-yahoo-msg	0.0003	0.0004	0.0479	0.0394	0.0005	**0.0569**
bio-gene	0.0195	0.0217	0.0315	0.0408	-0.0208	**0.0846**
ca-cora	0.0089	0.0304	0.0444	0.0608	0.0164	**0.1026**
soc-terror	0.0888	0.0892	0.0967	0.0967	0.0999	**0.1243**
inf-US-powerGrid	0.0027	0.0027	0.0061	0.0212	0.1127	**0.1242**
web-google	0.0272	0.0275	0.0429	0.0471	0.1010	**0.1122**
ca-CSphd	0.0224	0.0224	0.0234	0.0198	0.0131	**0.1201**
ca-netscience	0.0164	0.0168	0.1063	0.0561	0.1229	**0.1233**
road-luxem.	0.0629	0.0629	0.0077	0.0046	-0.1170	**0.1141**
bio-DD21	0.0865	0.0866	0.0106	0.0202	0.1241	**0.1247**

Runtime comparison



Summary of Contributions

- Proposed a new hierarchical graph clustering algorithm
- Fast with linear time and space complexity
- Outperforms other algorithms in terms of cluster quality
- Useful for visualization and interactive exploration of large networks

Thanks for listening!

Appendix