# Sublinear Algorithms for Estimating Single-Linkage Clustering Costs




Pan Peng
University of Science & Technology of China

Christian Sohler
University of Cologne
Cologne, Germany

Yi Xu
University of Science & Technology of China

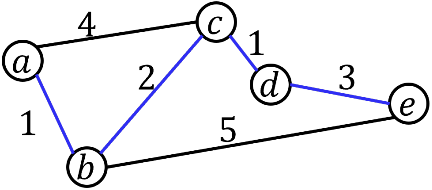

Women in TCS Workshop 2025

# Single Linkage Clustering

- Input: weighted graph $G = (V, E)$, **distance**/similarity
- SLC: bottom-up hierarchical clustering
  combine two **closest**/most similar clusters

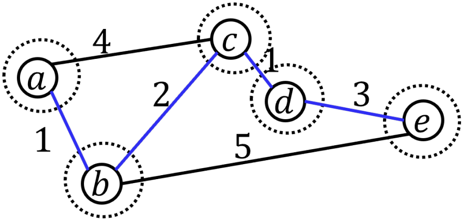
# Single Linkage Clustering

- Input: weighted graph $G = (V, E)$, **distance**/similarity
- SLC: bottom-up hierarchical clustering
  combine two **closest**/most similar clusters
- Example: $V = \{a, b, c, d, e\}$



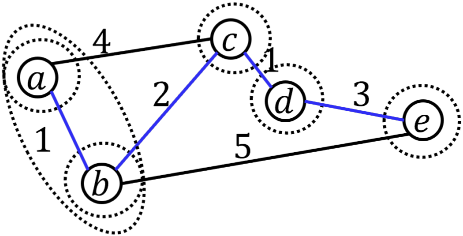
# Single Linkage Clustering

- Input: weighted graph $G = (V, E)$, **distance**/similarity
- SLC: bottom-up hierarchical clustering
  combine two **closest**/most similar clusters
- Example: $V = \{a, b, c, d, e\}$



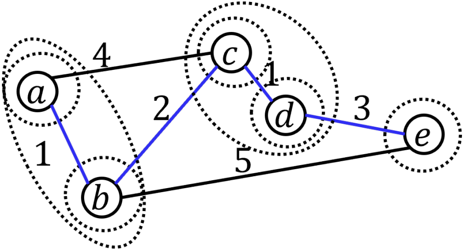
# Single Linkage Clustering

- Input: weighted graph $G = (V, E)$, **distance**/similarity
- SLC: bottom-up hierarchical clustering
  combine two **closest**/most similar clusters
- Example: $V = \{a, b, c, d, e\}$



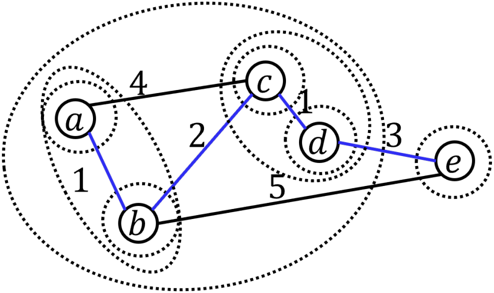
# Single Linkage Clustering

- Input: weighted graph $G = (V, E)$, **distance**/similarity
- SLC: bottom-up hierarchical clustering
  combine two **closest**/most similar clusters
- Example: $V = \{a, b, c, d, e\}$



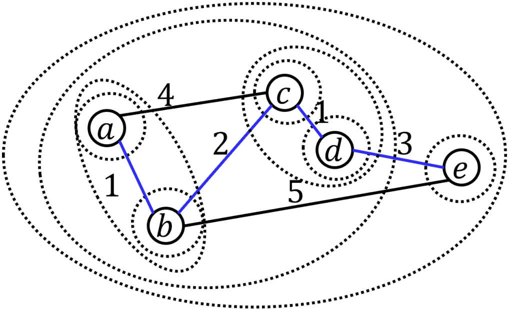
# Single Linkage Clustering

- Input: weighted graph $G = (V, E)$, **distance**/similarity
- SLC: bottom-up hierarchical clustering
  combine two **closest**/most similar clusters
- Example: $V = \{a, b, c, d, e\}$



# Single Linkage Clustering

- Input: weighted graph $G = (V, E)$, **distance**/similarity
- SLC: bottom-up hierarchical clustering
  combine two **closest**/most similar clusters
- Example: $V = \{a, b, c, d, e\}$



# Single Linkage Clustering

- Input: weighted graph $G = (V, E)$, **distance**/similarity
- SLC: bottom-up hierarchical clustering
  combine two **closest**/most similar clusters

$\mathrm{cost}_k$: sum of the costs of spanning trees within $k$ clusters
$\mathrm{cost}(G) := \sum_{k=1}^{n} \mathrm{cost}_k$, total clustering cost

# Single Linkage Clustering

- Input: weighted graph $G = (V, E)$, **distance**/similarity
- SLC: bottom-up hierarchical clustering
  combine two **closest**/most similar clusters

$\text{cost}_k$: sum of the costs of spanning trees within $k$ clusters
$\text{cost}(G) := \sum_{k=1}^{n} \text{cost}_k$, total clustering cost

Motivation:

- $\text{cost}_k$ captures important **structure**
- SLC minimizes these costs

# Single Linkage Clustering

- Input: weighted graph $G = (V, E)$, **distance**/similarity
- SLC: bottom-up hierarchical clustering
  combine two **closest**/most similar clusters

$\mathrm{cost}_k$: sum of the costs of spanning trees within $k$ clusters
$\mathrm{cost}(G) := \sum_{k=1}^{n} \mathrm{cost}_k$, total clustering cost

Motivation:

- $\mathrm{cost}_k$ captures important **structure**
- SLC minimizes these costs

Naive solution: compute an $\mathrm{MST}$ in $\tilde{O}(nd)$ time
Question: estimate $\mathrm{cost}(G)$ and $\mathrm{cost}_k$ in **sublinear** time?

# Main Results

$W$: max weight  $d$: average degree  query model: adj. list

| Setting | $\mathrm{cost}(G)$ | $\mathrm{cost}_k$ | Lower bound |
|---|---|---|---|
| Distance Case | $\tilde{O}(\frac{\sqrt{W}}{\varepsilon^3} d)$ | $\tilde{O}(\frac{\sqrt{W}}{\varepsilon^3} d)$ | $\Omega(\frac{\sqrt{W}}{\varepsilon^2} d)$ |
| Similarity Case | $\tilde{O}(\frac{W}{\varepsilon^3} d)$ | $\tilde{O}(\frac{W}{\varepsilon^3} d)$ | $\Omega(\frac{W}{\varepsilon^2} d)$ |

# Main Results

$W$: max weight    $d$: average degree    query model: adj. list

| Setting | cost$(G)$ | $\text{cost}_k$ | Lower bound |
|---|---|---|---|
| Distance Case | $\tilde{O}(\frac{\sqrt{W}}{\varepsilon^3}d)$ | $\tilde{O}(\frac{\sqrt{W}}{\varepsilon^3}d)$ | $\Omega(\frac{\sqrt{W}}{\varepsilon^2}d)$ |
| Similarity Case | $\tilde{O}(\frac{W}{\varepsilon^3}d)$ | $\tilde{O}(\frac{W}{\varepsilon^3}d)$ | $\Omega(\frac{W}{\varepsilon^2}d)$ |

**Succinct** representation of the SLC estimates $(\widehat{\text{cost}}_1, \ldots, \widehat{\text{cost}}_n)$ s.t. $\forall k$, recover $\widehat{\text{cost}}_k$ in a **short** time, and **on average** a $(1+\varepsilon)$ estimate

On average: $\sum_{k=1}^{n} |\widehat{\text{cost}_k} - \text{cost}_k| \leq \varepsilon \cdot \text{cost}(G) = \varepsilon \sum_{k=1}^{n} \text{cost}_k$
Short time: in $O(\log\log W)$ time

# Main Results

$W$: max weight   $d$: average degree   query model: adj. list

| Setting | cost$(G)$ | $\text{cost}_k$ | Lower bound |
|---|---|---|---|
| Distance Case | $\tilde{O}(\frac{\sqrt{W}}{\varepsilon^3}d)^1$ | $\tilde{O}(\frac{\sqrt{W}}{\varepsilon^3}d)$ | $\Omega(\frac{\sqrt{W}}{\varepsilon^2}d)$ |
| Similarity Case | $\tilde{O}(\frac{W}{\varepsilon^3}d)$ | $\tilde{O}(\frac{W}{\varepsilon^3}d)$ | $\Omega(\frac{W}{\varepsilon^2}d)$ |

**Succinct** representation of the SLC estimates $(\widehat{\text{cost}}_1, \ldots, \widehat{\text{cost}}_n)$ s.t. $\forall k$, recover $\widehat{\text{cost}}_k$ in a **short** time, and **on average** a $(1+\varepsilon)$ estimate

On average: $\sum_{k=1}^{n} |\widehat{\text{cost}}_k - \text{cost}_k| \leq \varepsilon \cdot \text{cost}(G) = \varepsilon \sum_{k=1}^{n} \text{cost}_k$
Short time: in $O(\log\log W)$ time

[1] Applying [CRT05], one can get: $(1+\varepsilon)$-estimate, $\tilde{O}(\frac{W}{\varepsilon^2}d)$ queries

# Proof Sketch for Total Cost Estimation (Distance Case)

CC: **C**onnected **C**omponent   $W$: max weight

**Step 1**

Reduction $\Rightarrow$ estimating # of **CC**s

# Proof Sketch for Total Cost Estimation (Distance Case)

CC: **C**onnected **C**omponent $W$: max weight

**Step 1**

Reduction $\Rightarrow$ estimating $\#$ of **CC**s

**Step 2 [CRT05]**

Estimate $\#$ of CCs $\Rightarrow$ sample & BFS

# Proof Sketch for Total Cost Estimation (Distance Case)

CC: **C**onnected **C**omponent   $W$: max weight

**Step 1**

Reduction $\Rightarrow$ estimating # of **CCs**   $\mathrm{cost}(G) \approx \sum_{j=1}^{W} c_j^2$

**Step 2 [CRT05]**

Estimate # of CCs $\Rightarrow$ sample & BFS

# Proof Sketch for Total Cost Estimation (Distance Case)

CC: **C**onnected **C**omponent $W$: max weight

## Step 1

Reduction $\Rightarrow$ estimating # of **CCs** $\mathrm{cost}(G) \approx \sum_{j=1}^{W} c_j^2$

## Step 2 [CRT05]

Estimate # of CCs $\Rightarrow$ sample & BFS in $\tilde{O}(\frac{\sqrt{W}}{\varepsilon^2})$ time

# Proof Sketch for Total Cost Estimation (Distance Case)

CC: **C**onnected **C**omponent   $W$: max weight

### Step 1

Reduction $\Rightarrow$ estimating # of **CC**s   $\mathrm{cost}(G) \approx \sum_{j=1}^{W} c_j^2$

### Step 2 [CRT05]

Estimate # of CCs $\Rightarrow$ sample & BFS   in $\tilde{O}(\frac{\sqrt{W}}{\varepsilon^2})$ time

Naive solution: estimate # of CCs for $W$ graphs, in $\tilde{O}(W \cdot \frac{\sqrt{W}}{\varepsilon^2})$ time!

# Proof Sketch for Total Cost Estimation (Distance Case)

CC: **C**onnected **C**omponent $W$: max weight

### Step 1

Reduction $\Rightarrow$ estimating # of **CCs** $\mathrm{cost}(G) \approx \sum_{j=1}^{W} c_j^2$

### Step 2 [CRT05]

Estimate # of CCs $\Rightarrow$ sample & BFS in $\tilde{O}(\frac{\sqrt{W}}{\varepsilon^2})$ time

Naive solution: estimate # of CCs for $W$ graphs, in $\tilde{O}(W \cdot \frac{\sqrt{W}}{\varepsilon^2})$ time!

### Step 3

Apply **binary search** to accelerate

# Proof Sketch for Total Cost Estimation (Distance Case)

CC: **C**onnected **C**omponent $W$: max weight

### Step 1

Reduction $\Rightarrow$ estimating # of **CCs** $\mathrm{cost}(G) \approx \sum_{j=1}^{W} c_j^2$

### Step 2 [CRT05]

Estimate # of CCs $\Rightarrow$ sample & BFS in $\tilde{O}(\frac{\sqrt{W}}{\varepsilon^2})$ time

Naive solution: estimate # of CCs for $W$ graphs, in $\tilde{O}(W \cdot \frac{\sqrt{W}}{\varepsilon^2})$ time!

### Step 3

Apply **binary search** to accelerate $\{c_j\}$ is monotonic

# Proof Sketch for Total Cost Estimation (Distance Case)

CC: **C**onnected **C**omponent  $W$: max weight

## Step 1

Reduction $\Rightarrow$ estimating # of **CC**s  $\mathrm{cost}(G) \approx \sum_{j=1}^{W} c_j^2$

## Step 2 [CRT05]

Estimate # of CCs $\Rightarrow$ sample & BFS  in $\tilde{O}(\frac{\sqrt{W}}{\varepsilon^2})$ time

Naive solution: estimate # of CCs for $W$ graphs, in $\tilde{O}(W \cdot \frac{\sqrt{W}}{\varepsilon^2})$ time!

## Step 3

Apply **binary search** to accelerate  $\{c_j\}$ is monotonic

$\Rightarrow$ ROBUST algo, works even on **not** monotonic estimates $\{\hat{c}_j\}$!

$\Rightarrow$ Estimate # of CCs upto $O(\log W/\varepsilon)$ graphs!

# Proof Sketch for Total Cost Estimation (Distance Case)

CC: **C**onnected **C**omponent   $W$: max weight

## Step 1

Reduction $\Rightarrow$ estimating # of **CC**s   $\mathrm{cost}(G) \approx \sum_{j=1}^{W} c_j^2$

## Step 2 [CRT05]

Estimate # of CCs $\Rightarrow$ sample & BFS  in $\tilde{O}(\frac{\sqrt{W}}{\varepsilon^2})$ time

Naive solution: estimate # of CCs for $W$ graphs, in $\tilde{O}(W \cdot \frac{\sqrt{W}}{\varepsilon^2})$ time!

## Step 3

Apply **binary search** to accelerate   $\{c_j\}$ is monotonic

$\Rightarrow$ ROBUST algo, works even on **not** monotonic estimates $\{\hat{c}_j\}$!

$\Rightarrow$ Estimate # of CCs upto $O(\log W/\varepsilon)$ graphs!

Total running time & queries: $\tilde{O}(\frac{\sqrt{W}}{\varepsilon^3})$