diameter, radius, discrete radius

$D : M \times M \to \mathbb{R}$ distance function, $S \subset M, |S| < \infty$

- $\text{diam}^D(S) := \max_{x,y \in S} D(x,y)$ (diameter of S)

diameter, radius, discrete radius

$D : M \times M \to \mathbb{R}$ distance function, $S \subset M, |S| < \infty$

- $\mathrm{diam}^D(S) := \max_{x,y \in S} D(x,y)$ (diameter of S)
- $\mathrm{rad}^D(S) := \min_{m \in M} \max_{x \in S} D(x,m)$ (radius of S)

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

diameter, radius, discrete radius

$D : M \times M \to \mathbb{R}$ distance function, $S \subset M, |S| < \infty$

- $\mathrm{diam}^D(S) := \max_{x,y \in S} D(x,y)$ (diameter of S)
- $\mathrm{rad}^D(S) := \min_{m \in M} \max_{x \in S} D(x,m)$ (radius of S)
- $\mathrm{drad}^D(S) := \min_{m \in S} \max_{x \in S} D(x,m)$ (discrete radius of S)

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

diameter, radius, discrete radius

$D : M \times M \to \mathbb{R}$ distance function, $S \subset M, |S| < \infty$

- $\mathrm{diam}^D(S) := \max_{x,y \in S} D(x,y)$ (diameter of S)
- $\mathrm{rad}^D(S) := \min_{m \in M} \max_{x \in S} D(x,m)$ (radius of S)
- $\mathrm{drad}^D(S) := \min_{m \in S} \max_{x \in S} D(x,m)$ (discrete radius of S)

$P \subset M, |P| < \infty, \mathcal{C} = \{C_1, \ldots, C_k\}$ partition of P

- $\mathrm{cost}^D_{\mathrm{diam}}(\mathcal{C}) := \max_{1 \leq i \leq k} \mathrm{diam}^D(C_i)$ (diameter cost)

diameter, radius, discrete radius

$D : M \times M \to \mathbb{R}$ distance function, $S \subset M, |S| < \infty$

- $\mathrm{diam}^D(S) := \max_{x,y \in S} D(x,y)$ (diameter of S)
- $\mathrm{rad}^D(S) := \min_{m \in M} \max_{x \in S} D(x,m)$ (radius of S)
- $\mathrm{drad}^D(S) := \min_{m \in S} \max_{x \in S} D(x,m)$ (discrete radius of S)

$P \subset M, |P| < \infty, \mathcal{C} = \{C_1, \ldots, C_k\}$ partition of P

- $\mathrm{cost}^D_{\mathrm{diam}}(\mathcal{C}) := \max_{1 \le i \le k} \mathrm{diam}^D(C_i)$ (diameter cost)
- $\mathrm{cost}^D_{\mathrm{rad}}(\mathcal{C}) := \max_{1 \le i \le k} \mathrm{rad}^D(C_i)$ (radius cost)

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

diameter, radius, discrete radius

$D : M \times M \to \mathbb{R}$ distance function, $S \subset M, |S| < \infty$

- $\mathrm{diam}^D(S) := \max_{x,y \in S} D(x,y)$ (diameter of S)
- $\mathrm{rad}^D(S) := \min_{m \in M} \max_{x \in S} D(x,m)$ (radius of S)
- $\mathrm{drad}^D(S) := \min_{m \in S} \max_{x \in S} D(x,m)$ (discrete radius of S)

$P \subset M, |P| < \infty, \mathcal{C} = \{C_1, \ldots, C_k\}$ partition of P

- $\mathrm{cost}^D_{\mathrm{diam}}(\mathcal{C}) := \max_{1 \le i \le k} \mathrm{diam}^D(C_i)$ (diameter cost)
- $\mathrm{cost}^D_{\mathrm{rad}}(\mathcal{C}) := \max_{1 \le i \le k} \mathrm{rad}^D(C_i)$ (radius cost)
- $\mathrm{cost}^D_{\mathrm{drad}}(\mathcal{C}) := \max_{1 \le i \le k} \mathrm{drad}^D(C_i)$ (discrete radius cost)

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Problem 6.1 (diameter k-clustering)

Given a set $P, |P| < \infty, k \in \mathbb{N}$, find a partition $\mathcal{C}$ of P into k clusters $C_1, \ldots, C_k$ that minimizes $cost^{D}_{diam}(\mathcal{C})$.

Problem 6.1 (diameter k-clustering)

Given a set $P, |P| < \infty, k \in \mathbb{N}$, find a partition $\mathcal{C}$ of P into k clusters $C_1, \ldots, C_k$ that minimizes $cost^D_{diam}(\mathcal{C})$.

Problem 6.2 (radius k-clustering)

Given a set $P, |P| < \infty, k \in \mathbb{N}$, find a partition $\mathcal{C}$ of P into k clusters $C_1, \ldots, C_k$ that minimizes $cost^D_{rad}(\mathcal{C})$.

Problem 6.1 (diameter k-clustering)

Given a set $P, |P| < \infty, k \in \mathbb{N}$, find a partition $\mathcal{C}$ of P into k clusters $C_1, \ldots, C_k$ that minimizes $cost^D_{diam}(\mathcal{C})$.

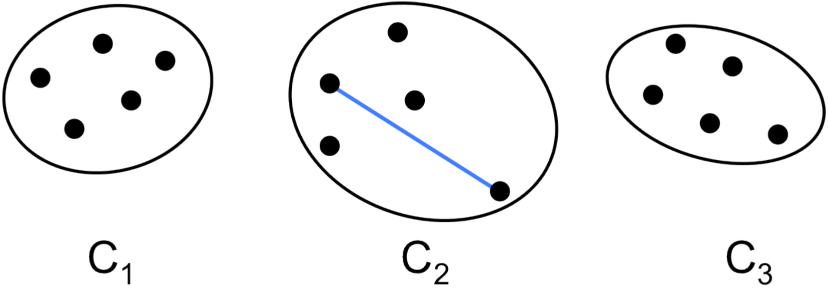
Problem 6.2 (radius k-clustering)

Given a set $P, |P| < \infty, k \in \mathbb{N}$, find a partition $\mathcal{C}$ of P into k clusters $C_1, \ldots, C_k$ that minimizes $cost^D_{rad}(\mathcal{C})$.

Problem 6.3 (discrete radius k-clustering)

Given a set $P, |P| < \infty, k \in \mathbb{N}$, find a partition $\mathcal{C}$ of P into k clusters $C_1, \ldots, C_k$ that minimizes $cost^D_{drad}(\mathcal{C})$.

Diameter clustering



Agglomerative clustering - setup and idea

$D : M \times M \to \mathbb{R}$ distance function,
$P \subset M, |P| = n, P = \{p_1, \ldots, p_n\}$

Agglomerative clustering - setup and idea

$D: M \times M \to \mathbb{R}$ distance function,
$P \subset M, |P| = n, P = \{p_1, \ldots, p_n\}$

Basic idea of agglomerative clustering

- start with n clusters $C_i, 1 \leq i \leq n, C_i := \{p_i\}$
- in each step replace two clusters C_i, C_j that are "closest" by their union $C_i \cup C_j$
- until single cluster is left.

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Agglomerative clustering - setup and idea

$D : M \times M \to \mathbb{R}$ distance function,
$P \subset M, |P| = n, P = \{p_1, \dots, p_n\}$

Basic idea of agglomerative clustering

- start with n clusters $C_i, 1 \leq i \leq n, C_i := \{p_i\}$
- in each step replace two clusters C_i, C_j that are "closest" by their union $C_i \cup C_j$
- until single cluster is left.

Observation Computes k-clustering for $k = n, \dots, 1$.

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Definition 6.4

For $C_1, C_2 \subset M$

$$D_{CL}(C_1, C_2) := \max_{x \in C_1, y \in C_2} D(x, y)$$

is called the complete linkage cost of C_1, C_2.

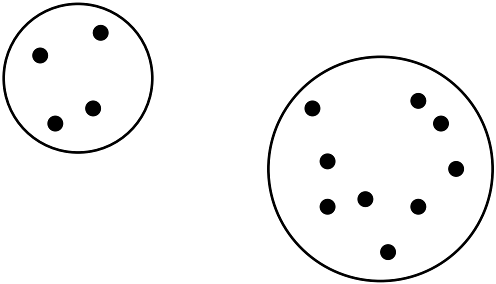
Complete linkage

Definition 6.4

For $C_1, C_2 \subset M$

$$D_{CL}(C_1, C_2) := \max_{x \in C_1, y \in C_2} D(x, y)$$

is called the complete linkage cost of C_1, C_2*.*

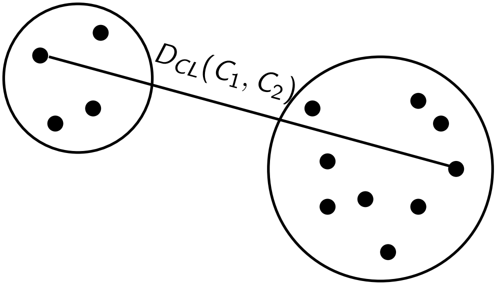


Definition 6.4

For $C_1, C_2 \subset M$

$$D_{CL}(C_1, C_2) := \max_{x \in C_1, y \in C_2} D(x, y)$$

is called the complete linkage cost of C_1, C_2.



Agglomerative clustering with complete linkage

$\textsc{AgglomerativeCompleteLinkage}(P)$

$\mathcal{C}_n := \{\{p_i\}|p_i \in P\}$;

for $i = n-1, \ldots, 1$ **do**

  find distinct cluster $A, B \in \mathcal{C}_{i+1}$ minimizing $D_{CL}(A, B)$;

  $\mathcal{C}_i := (\mathcal{C}_{i+1} \setminus \{A, B\}) \cup \{A \cup B\}$;

end

return $\mathcal{C}_1, \ldots, \mathcal{C}_n$ (or single $\mathcal{C}_k$)

Agglomerative clustering with complete linkage

$\textsc{AgglomerativeCompleteLinkage}(P)$

$\mathcal{C}_n := \{\{p_i\} | p_i \in P\};$

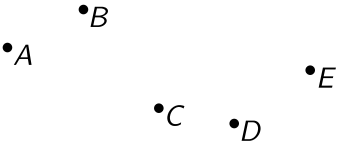
for $i = n-1, \ldots, 1$ **do**

  find distinct cluster $A, B \in \mathcal{C}_{i+1}$ minimizing $D_{CL}(A, B)$;

  $\mathcal{C}_i := (\mathcal{C}_{i+1} \setminus \{A, B\}) \cup \{A \cup B\};$

end

return $\mathcal{C}_1, \ldots, \mathcal{C}_n$ (or single $\mathcal{C}_k$)



Agglomerative clustering with complete linkage

AGGLOMERATIVECOMPLETELINKAGE(P)

$\mathcal{C}_n := \{\{p_i\} | p_i \in P\}$;

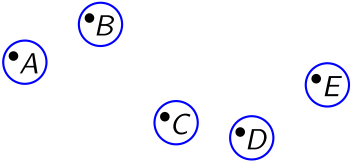
for $i = n-1, \ldots, 1$ **do**

  find distinct cluster $A, B \in \mathcal{C}_{i+1}$ minimizing $D_{CL}(A, B)$;

  $\mathcal{C}_i := (\mathcal{C}_{i+1} \setminus \{A, B\}) \cup \{A \cup B\}$;

end

return $\mathcal{C}_1, \ldots, \mathcal{C}_n$ (or single $\mathcal{C}_k$)



Agglomerative clustering with complete linkage

AGGLOMERATIVECOMPLETELINKAGE(P)

$\mathcal{C}_n := \{\{p_i\} | p_i \in P\}$;

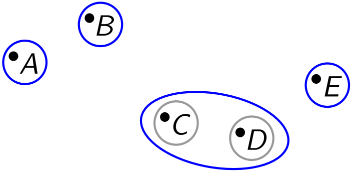
for $i = n-1, \ldots, 1$ **do**

  find distinct cluster $A, B \in \mathcal{C}_{i+1}$ minimizing $D_{CL}(A, B)$;

  $\mathcal{C}_i := (\mathcal{C}_{i+1} \setminus \{A, B\}) \cup \{A \cup B\}$;

end

return $\mathcal{C}_1, \ldots, \mathcal{C}_n$ (or single $\mathcal{C}_k$)



Agglomerative clustering with complete linkage

AGGLOMERATIVECOMPLETELINKAGE(P)

$\mathcal{C}_n := \{\{p_i\} | p_i \in P\}$;

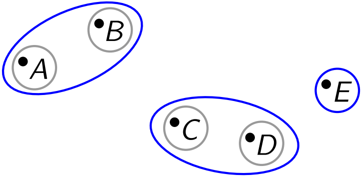
for $i = n-1, \ldots, 1$ **do**

  find distinct cluster $A, B \in \mathcal{C}_{i+1}$ minimizing $D_{CL}(A, B)$;

  $\mathcal{C}_i := (\mathcal{C}_{i+1} \setminus \{A, B\}) \cup \{A \cup B\}$;

end

return $\mathcal{C}_1, \ldots, \mathcal{C}_n$ (or single $\mathcal{C}_k$)



UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Agglomerative clustering with complete linkage

AGGLOMERATIVECOMPLETELINKAGE(P)

$\mathcal{C}_n := \{\{p_i\} | p_i \in P\}$;

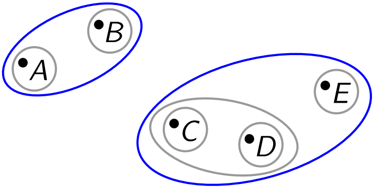
for $i = n-1, \ldots, 1$ **do**

  find distinct cluster $A, B \in \mathcal{C}_{i+1}$ minimizing $D_{CL}(A, B)$;

  $\mathcal{C}_i := (\mathcal{C}_{i+1} \setminus \{A, B\}) \cup \{A \cup B\}$;

end

return $\mathcal{C}_1, \ldots, \mathcal{C}_n$ (or single $\mathcal{C}_k$)



Agglomerative clustering with complete linkage

AGGLOMERATIVECOMPLETELINKAGE(P)

$\mathcal{C}_n := \{\{p_i\} | p_i \in P\}$;

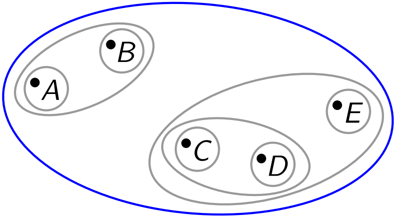
for $i = n-1, \dots, 1$ **do**

  find distinct cluster $A, B \in \mathcal{C}_{i+1}$ minimizing $D_{CL}(A, B)$;

  $\mathcal{C}_i := (\mathcal{C}_{i+1} \setminus \{A, B\}) \cup \{A \cup B\}$;

end

return $\mathcal{C}_1, \dots, \mathcal{C}_n$ (or single $\mathcal{C}_k$)



Agglomerative clustering with complete linkage

AgglomerativeCompleteLinkage(P)

$\mathcal{C}_n := \{\{p_i\} | p_i \in P\}$;

for $i = n-1, \ldots, 1$ **do**

  find distinct cluster $A, B \in \mathcal{C}_{i+1}$ minimizing $D_{CL}(A, B)$;

  $\mathcal{C}_i := (\mathcal{C}_{i+1} \setminus \{A, B\}) \cup \{A \cup B\}$;

end

return $\mathcal{C}_1, \ldots, \mathcal{C}_n$ (or single $\mathcal{C}_k$)

Agglomerative clustering with complete linkage

AgglomerativeCompleteLinkage(P)

$\mathcal{C}_n := \{\{p_i\} | p_i \in P\}$;

for $i = n-1, \ldots, 1$ **do**

  find distinct cluster $A, B \in \mathcal{C}_{i+1}$ minimizing $D_{CL}(A, B)$;

  $\mathcal{C}_i := (\mathcal{C}_{i+1} \setminus \{A, B\}) \cup \{A \cup B\}$;

end

return $\mathcal{C}_1, \ldots, \mathcal{C}_n$ (or single $\mathcal{C}_k$)

Theorem 6.5

Algorithm AgglomerativeCompleteLinkage *requires time* $O(n^2 \log n)$ *and space* $O(n^2)$.

Approximation guarantees

- $\mathrm{diam}^D(S) := \max_{x,y \in S} D(x,y)$ (diameter of S)
- $\mathrm{cost}^D_{\mathrm{diam}}(\mathcal{C}) := \max_{1 \le i \le k} \mathrm{diam}^D(C_i)$ (diameter cost)
- $\mathrm{opt}_k^{diam}(P) := \min_{|\mathcal{C}|=k} \mathrm{cost}^D_{\mathrm{diam}}(\mathcal{C})$

Approximation guarantees

- $\mathrm{diam}^D(S) := \max_{x,y \in S} D(x,y)$ (diameter of S)
- $\mathrm{cost}^D_{\mathrm{diam}}(\mathcal{C}) := \max_{1 \le i \le k} \mathrm{diam}^D(C_i)$ (diameter cost)
- $\mathrm{opt}_k^{diam}(P) := \min_{|\mathcal{C}|=k} \mathrm{cost}^D_{\mathrm{diam}}(\mathcal{C})$

Theorem 6.6

Let D be a distance metric on $M \subseteq \mathbb{R}^d$. Then for all sets P and all $k \le |P|$, Algorithm AgglomerativeCompleteLinkage *computes a k-clustering $\mathcal{C}_k$ with*

$$cost^D_{diam}(\mathcal{C}_k) \le O\left(opt_k^{diam}(P)\right),$$

where the constant hidden in the O-notation is double exponential in d.

Approximation guarantees

Theorem 6.7

There is a point set $P \subset \mathbb{R}^2$ such that for the metric D_{l_∞} algorithm AgglomerativeCompleteLinkage *computes a clustering $\mathcal{C}_k$ with*

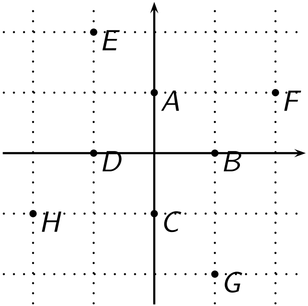
$$cost^{D}_{diam}(\mathcal{C}_k) = 3 \cdot opt^{diam}_{k}(P).$$

Approximation guarantees

Theorem 6.7

There is a point set $P \subset \mathbb{R}^2$ such that for the metric D_{l_∞} algorithm AGGLOMERATIVECOMPLETELINKAGE *computes a clustering $\mathcal{C}_k$ with*

$$cost_{diam}^{D}(\mathcal{C}_k) = 3 \cdot opt_k^{diam}(P).$$



Theorem 6.8

There is a point set $P \subset \mathbb{R}^d$, $d = k + \log k$ such that for the metric D_{l_1} algorithm AGGLOMERATIVECOMPLETELINKAGE *computes a clustering $\mathcal{C}_k$ with*

$$cost_{diam}^{D_{l_1}}(\mathcal{C}_k) \geq \frac{1}{2} \log k \cdot opt_k^{diam}(P).$$

Approximation garantees

Theorem 6.8

There is a point set $P \subset \mathbb{R}^d, d = k + \log k$ such that for the metric D_{l_1} algorithm $\textsc{AgglomerativeCompleteLinkage}$ *computes a clustering $\mathcal{C}_k$ with*

$$cost_{diam}^{D_{l_1}}(\mathcal{C}_k) \geq \frac{1}{2} \log k \cdot opt_k^{diam}(P).$$

Corollary 6.9

For every $1 \leq p < \infty$, there is a point set $P \subset \mathbb{R}^d, d = k + \log k$ such that for the metric D_{l_p} algorithm $\textsc{AgglomerativeCompleteLinkage}$ *computes a clustering $\mathcal{C}_k$ with*

$$cost_{diam}^{D_{l_p}}(\mathcal{C}_k) \geq \sqrt[p]{\frac{1}{2} \log k} \cdot opt_k^{diam}(P).$$

Theorem 6.10

For the metric D_{l_2} the diameter k-clustering problem is **NP**-*hard. Moreover, assuming* $\mathbf{P} \neq \mathbf{NP}$*, there is no polynomial time approximation for the diameter k-clustering with approximation factor* ≤ 1.96.

Hardness of diameter clustering

- $\Delta \in \mathbb{R}_{\geq 0}^{n \times n}, \Delta_{xy} := (x, y)$-entry in $\Delta, 1 \leq x, y \leq n$
- $\mathcal{C} = \{C_1, \ldots, C_k\}$ partition of $\{1, \ldots, n\}$
- $\text{cost}_{diam}^{\Delta} := \max_{1 \leq i \leq k} \max_{x,y \in C_i} \Delta_{xy}$

- $\Delta \in \mathbb{R}_{\geq 0}^{n \times n}, \Delta_{xy} :=$ (x, y)-entry in $\Delta, 1 \leq x, y \leq n$
- $\mathcal{C} = \{C_1, \ldots, C_k\}$ partition of $\{1, \ldots, n\}$
- $\text{cost}_{diam}^{\Delta} := \max_{1 \leq i \leq k} \max_{x,y \in C_i} \Delta_{xy}$

Problem 6.11 (matrix diameter k-clustering)

Given a matrix $\Delta \in \mathbb{R}_{\geq 0}^{n \times n}, k \in \mathbb{N}$, find a partition $\mathcal{C}$ of $\{1, \ldots, n\}$ into k clusters $C_1, \ldots, C_k$ that minimizes $\text{cost}_{\text{diam}}^{\Delta}(\mathcal{C})$.

Hardness of diameter clustering

- $\Delta \in \mathbb{R}_{\geq 0}^{n \times n}, \Delta_{xy} :=$ (x, y)-entry in $\Delta, 1 \leq x, y \leq n$
- $\mathcal{C} = \{C_1, \ldots, C_k\}$ partition of $\{1, \ldots, n\}$
- $\text{cost}_{diam}^{\Delta} := \max_{1 \leq i \leq k} \max_{x,y \in C_i} \Delta_{xy}$

Problem 6.11 (matrix diameter k-clustering)

Given a matrix $\Delta \in \mathbb{R}_{\geq 0}^{n \times n}$, $k \in \mathbb{N}$, find a partition $\mathcal{C}$ of $\{1, \ldots, n\}$ into k clusters $C_1, \ldots, C_k$ that minimizes $cost_{diam}^{\Delta}(\mathcal{C})$.

Theorem 6.12

The matrix diameter k-clustering problem is **NP***-hard. Moreover, assuming* $\mathbf{P} \neq \mathbf{NP}$*, there is no polynomial time approximation for the diameter k-clustering with approximation factor $\alpha \geq 1$ arbitrary.*

Problem 6.13 (maximum distance k-clustering)

Given distance measure $D : M \times M \to \mathbb{R}$, $k \in \mathbb{N}$, and $P \subset M$, find a partition $\mathcal{C} = \{C_1, \ldots, C_k\}$ of P into k clusters that maximizes

$$\min_{x \in C_i, y \in C_j, i \neq j} D(x, y),$$

i.e. a partition that maximizes the minimum distance between points in different clusters.

Problem 6.13 (maximum distance k-clustering)

Given distance measure $D : M \times M \to \mathbb{R}$, $k \in \mathbb{N}$, and $P \subset M$, find a partition $\mathcal{C} = \{C_1, \ldots, C_k\}$ of P into k clusters that maximizes

$$\min_{x \in C_i, y \in C_j, i \neq j} D(x, y),$$

i.e. a partition that maximizes the minimum distance between points in different clusters.

Definition 6.14

For $C_1, C_2 \subset M$

$$D_{SL}(C_1, C_2) := \min_{x \in C_1, y \in C_2} D(x, y)$$

is called the single linkage cost of C_1, C_2.

Agglomerative clustering with single linkage

AgglomerativeSingleLinkage(P)

$\mathcal{C}_n := \{\{p_i\} | p_i \in P\}$;

for $i = n-1, \ldots, 1$ **do**

  find distinct cluster $A, B \in \mathcal{C}_{i+1}$ minimizing $D_{SL}(A, B)$;

  $\mathcal{C}_i := (\mathcal{C}_{i+1} \setminus \{A, B\}) \cup \{A \cup B\}$;

end

return $\mathcal{C}_1, \ldots, \mathcal{C}_n$ (or single $\mathcal{C}_k$)

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Agglomerative clustering with single linkage

$\textsc{AgglomerativeSingleLinkage}(P)$

$\mathcal{C}_n := \{\{p_i\} | p_i \in P\}$;

for $i = n-1, \ldots, 1$ **do**

  find distinct cluster $A, B \in \mathcal{C}_{i+1}$ minimizing $D_{SL}(A, B)$;

  $\mathcal{C}_i := (\mathcal{C}_{i+1} \setminus \{A, B\}) \cup \{A \cup B\}$;

end

return $\mathcal{C}_1, \ldots, \mathcal{C}_n$ (or single $\mathcal{C}_k$)

Theorem 6.15

Algorithm $\textsc{AgglomerativeSingleLinkage}$ *optimally solves the maximum distance k-clustering problem.*

- $\mathrm{drad}^D(S) := \min_{m \in S} \max_{x \in S} D(x, m)$ (discrete radius of S)
- $\mathrm{cost}^D_{\mathrm{drad}}(\mathcal{C}) := \max_{1 \le i \le k} \mathrm{drad}^D(C_i)$ (discrete radius cost)
- find a partition $\mathcal{C}$ of P into k clusters $C_1, \ldots, C_k$ that minimizes $\mathrm{cost}^D_{\mathrm{drad}}(\mathcal{C})$ or $\mathrm{cost}^D_{\mathrm{rad}}(\mathcal{C})$.

- $\mathrm{drad}^D(S) := \min_{m \in S} \max_{x \in S} D(x, m)$ (discrete radius of S)
- $\mathrm{cost}^D_{\mathrm{drad}}(\mathcal{C}) := \max_{1 \le i \le k} \mathrm{drad}^D(C_i)$ (discrete radius cost)
- find a partition $\mathcal{C}$ of P into k clusters $C_1, \ldots, C_k$ that minimizes $\mathrm{cost}^D_{\mathrm{drad}}(\mathcal{C})$ or $\mathrm{cost}^D_{\mathrm{rad}}(\mathcal{C})$.

Theorem 6.16

Let $D : M \times M \to \mathbb{R}$ be a metric, $P \subset M$ and $\mathcal{C} = \{C_1, \ldots, C_k\}$ a partition of P. Then

1. $cost_{drad}(\mathcal{C}) \le cost_{diam}(\mathcal{C}) \le 2 \cdot cost_{drad}(\mathcal{C})$
2. $\frac{1}{2} \cdot cost_{drad}(\mathcal{C}) \le cost_{rad}(\mathcal{C}) \le cost_{drad}(\mathcal{C})$

Corollary 6.17

Let $D : M \times M \to \mathbb{R}$ *be a metric,* $k \in \mathbb{N}$*, and* $P \subset M$*. Then*

1. $opt_k^{drad}(P) \leq opt_k^{diam}(P) \leq 2 \cdot opt_k^{drad}(P)$
2. $\frac{1}{2} \cdot opt_k^{drad}(P) \leq opt_k^{rad}(P) \leq opt_k^{drad}(P)$

Corollary 6.17

Let $D : M \times M \to \mathbb{R}$ be a metric, $k \in \mathbb{N}$, and $P \subset M$. Then

1. $opt_k^{drad}(P) \leq opt_k^{diam}(P) \leq 2 \cdot opt_k^{drad}(P)$
2. $\frac{1}{2} \cdot opt_k^{drad}(P) \leq opt_k^{rad}(P) \leq opt_k^{drad}(P)$

Corollary 6.18

Assume there is a polynomial time c-approximation algorithm for the discrete radius k-clustering problem. Then there is a polynomial time 2c-approximation algorithm for the diameter k-clustering problem.

GonzalesAlgorithm(P, k)

$C := \{p\}$ for $p \in P$ arbitrary;

for $i = 1, \ldots, k-1$ **do**

  $q := \mathrm{argmax}_{y \in P} D(y, C)$;

  $C := C \cup \{q\}$;

end

compute partition $\mathcal{C} = \{C_1, \ldots, C_k\}$ corresponding to C;

return $\mathcal{C}$ and C

GonzalesAlgorithm(P, k)

$C := \{p\}$ for $p \in P$ arbitrary;
for $i = 1, \ldots, k-1$ **do**
  $q := \text{argmax}_{y \in P} D(y, C)$;
  $C := C \cup \{q\}$;
end
compute partition $\mathcal{C} = \{C_1, \ldots, C_k\}$ corresponding to C;
return $\mathcal{C}$ and C

Theorem 6.19

For any metric D, Algorithm GonzalesAlgorithm *is a 2-approximation algorithm for the diameter, radius, and discrete radius k-clustering problem.*

- $C := \{c_1, \dots, c_k\}$ set of points chosen by the algorithm
- chosen in order $c_1, \dots, c_k$
- $G_l := \{c_1, \dots, c_l\}$, i.e. G_l set of first l points chosen by the algorithm, $C = G_k$.
- $c_{k+1} := \text{argmax}_{q \in P} D(q, G_k)$, i.e. c_{k+1} is the point that would be chosen by the algorithm in an additional iteration

Show that

1. $\forall l \leq k-1 : D(c_{l+1}, G_l) \geq D(c_{k+1}, G_k)$
2. $\text{opt}_k^{\text{diam}}(P) \geq D(c_{k+1}, G_k)$
3. $\text{cost}_{\text{diam}}^{D}(\mathcal{C}) \leq 2 \cdot D(c_{k+1}, G_k)$.

Proof of Theorem 6.19 (for diameter)

- 2 and 3 imply the theorem
- 1 is used to prove 2

Proof of 1

- Assume there is an index $l, 1 \leq l \leq k-1$, such that $D(c_{l+1}, G_l) < D(c_{k+1}, G_k)$.
- Then $D(c_{l+1}, G_l) < D(c_{k+1}, G_k) \leq D(c_{k+1}, G_l)$, since $G_| \subset G_k$.
- This contradicts the choice of c_{l+1} as $\text{argmax}_{q \in p} D(q, G_l)$.

Proof of Theorem 6.19 (for diameter)

Proof of 2

- It suffice to prove that for any $k$clustering $\hat{\mathcal{C}}$ of P we have $\mathrm{cost}^D_{\mathrm{diam}}(\hat{\mathcal{C}}) \geq D(c_{k+1}, G_k)$.
- For any k-clustering $\hat{\mathcal{C}}$ of P there are two elements c_i, c_j of $C \cup \{c_{k+1}\}$ that belong to the same cluster. Assume without loss of generality that $i < j$.
- Then $\mathrm{cost}^D_{\mathrm{diam}}(\hat{\mathcal{C}}) \geq D(c_j, c_i) \geq D(c_j, G_{j-1}) \geq D(c_{k+1}, G_k)$, where the second inequality follows from $c_i \in G_{j-1}$ and the last inequality follows from 1.
- This contradicts the choice of c_{l+1} as $\mathrm{argmax}_{q \in p} D(q, G_l)$.

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Proof of Theorem 6.19 (for diameter)

Proof of 3

- Let C_l be a cluster in the clustering $\mathcal{C}$ computed by Gonzales' algorithm and let $u, v \in C_l$. It suffices to show that $D(u, v) \leq 2 \cdot D(c_{k+1}, G_k)$.
- Let c_l be the element of C such that C_l consists of all elements that are closer to c_l than to any other element in C.
- Then $D(u, c_l) = D(u, G_k) \leq D(c_{k+1}, G_k)$, where the inequality follows from the definition of c_{k+1} as $\mathrm{argmax}_{q \in p} D(q, G_k)$. Similarly, we get $D(v, c_l) \leq D(c_{k+1}, G_k)$.
- By the triangle inequality $D(u, v)|leD(u, c_l) + D(v, c_i) \leq 2 \cdot D(c_{k+1}, G_k)$.

- $\mathrm{drad}^D(S) := \min_{m \in S} \max_{x \in S} D(x, m)$ (discrete radius of S)
- $\mathrm{cost}^D_{\mathrm{drad}}(\mathcal{C}) := \max_{1 \leq i \leq k} \mathrm{drad}^D(C_i)$ (discrete radius cost)
- find a partition $\mathcal{C}$ of P into k clusters $C_1, \dots, C_k$ that minimizes $\mathrm{cost}^D_{\mathrm{drad}}(\mathcal{C})$.

Agglomerative clustering and discrete radius clustering

- $\mathrm{drad}^D(S) := \min_{m \in S} \max_{x \in S} D(x, m)$ (discrete radius of S)
- $\mathrm{cost}^D_{\mathrm{drad}}(\mathcal{C}) := \max_{1 \le i \le k} \mathrm{drad}^D(C_i)$ (discrete radius cost)
- find a partition $\mathcal{C}$ of P into k clusters $C_1, \ldots, C_k$ that minimizes $\mathrm{cost}^D_{\mathrm{drad}}(\mathcal{C})$.

Discrete radius measure

$$D_{\mathrm{drad}}(C_1, C_2) = \mathrm{drad}(C_1 \cup C_2)$$

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Agglomerative clustering with dradius cost

$\textsc{AgglomerativeDiscreteRadius}(P)$

$\mathcal{C}_n := \{\{p_i\} | p_i \in P\}$;

for $i = n-1, \ldots, 1$ **do**

find distinct clusters $A, B \in \mathcal{C}_{i+1}$ minimizing $D_{\text{drad}}(A, B)$;

$\mathcal{C}_i := (\mathcal{C}_{i+1} \setminus \{A, B\}) \cup \{A \cup B\}$;

end

return $\mathcal{C}_1, \ldots, \mathcal{C}_n$ (or single $\mathcal{C}_k$)

Agglomerative clustering with dradius cost

AGGLOMERATIVEDISCRETERADIUS(P)

$\mathcal{C}_n := \{\{p_i\} | p_i \in P\}$;
for $i = n-1, \ldots, 1$ **do**
    find distinct clusters $A, B \in \mathcal{C}_{i+1}$ minimizing $D_{\mathrm{drad}}(A, B)$;
    $\mathcal{C}_i := (\mathcal{C}_{i+1} \setminus \{A, B\}) \cup \{A \cup B\}$;
end
return $\mathcal{C}_1, \ldots, \mathcal{C}_n$ (or single $\mathcal{C}_k$)

Theorem 6.20

Let D be a distance metric on $M \subseteq \mathbb{R}^d$. Then for all sets $P \subset M$ and all $k \leq |P|$, Algorithm AGGLOMERATIVEDISCRETERADIUS *computes a k-clustering $\mathcal{C}_k$ with*

$$cost_k^{drad}(\mathcal{C}_k) < O(d) \cdot opt_k.$$

Hierarchical clustering Given distance measure $D : M \times M \to \mathbb{R}, k \in \mathbb{N}$, and $P \subset M, |P| = n$, a sequence of clusterings $\mathcal{C}_n, \ldots, \mathcal{C}_1$ with $|\mathcal{C}_k| = k$ is called *hierarchical clustering of P* if for all $A \in \mathcal{C}_k$

1. $A \in \mathcal{C}_{k+1}$ or
2. $\exists B, C \in \mathcal{C}_{k+1} : A = B \cup C \quad \text{and} \quad \mathcal{C}_k = \mathcal{C}_{k+1} \setminus \{B, C\} \cup \{A\}$.

Hierarchical clustering Given distance measure $D : M \times M \to \mathbb{R}, k \in \mathbb{N}$, and $P \subset M, |P| = n$, a sequence of clusterings $\mathcal{C}_n, \ldots, \mathcal{C}_1$ with $|\mathcal{C}_k| = k$ is called *hierarchical clustering of P* if for all $A \in \mathcal{C}_k$

1. $A \in \mathcal{C}_{k+1}$ or
2. $\exists B, C \in \mathcal{C}_{k+1} : A = B \cup C \quad and \quad \mathcal{C}_k = \mathcal{C}_{k+1} \setminus \{B, C\} \cup \{A\}$.

Dendrograms A dendrogram on n nodes is a rooted binary tree $T = (V, E)$ with an index function $\chi : V \setminus \{\text{leaves of } T\} \to \{1, \ldots, n\}$ such that

- $\forall v \neq w : \chi(v) \neq \chi(w)$
- $\chi(\text{root}) = n$
- $\forall u, v$: if v parent of u, then $\chi(v) > \chi(u)$.

From hierarchical clusterings to dendrograms

$\mathcal{C}_n, \ldots, \mathcal{C}_1$ hierarchical clustering of P.

Construction of dendrogram

- create leaf for each point $p \in P$
- interior nodes correspond to union of clusters
- if k-th cluster is obtained by union of clusters B, C, create new node with index k and with children B, C.

AgglomerativeCompleteLinkage

- Start with one cluster for each input object.
- Iteratively merge the two closest clusters.

Complete linkage measure

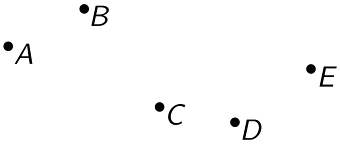
$$D_{CL}(C_1, C_2) = \max_{x \in C_1, y \in C_2} D(x, y)$$

AgglomerativeCompleteLinkage

- Start with one cluster for each input object.
- Iteratively merge the two closest clusters.

Complete linkage measure

$$D_{CL}(C_1, C_2) = \max_{x \in C_1, y \in C_2} D(x, y)$$



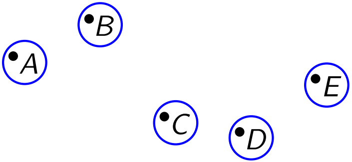
AgglomerativeCompleteLinkage

- Start with one cluster for each input object.
- Iteratively merge the two closest clusters.

Complete linkage measure

$$D_{CL}(C_1, C_2) = \max_{x \in C_1, y \in C_2} D(x, y)$$

A B C D E

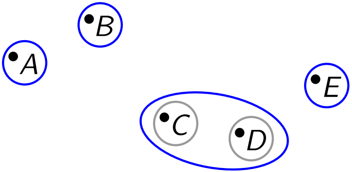


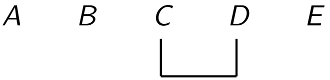
AgglomerativeCompleteLinkage

- Start with one cluster for each input object.
- Iteratively merge the two closest clusters.

Complete linkage measure

$$D_{CL}(C_1, C_2) = \max_{x \in C_1, y \in C_2} D(x, y)$$



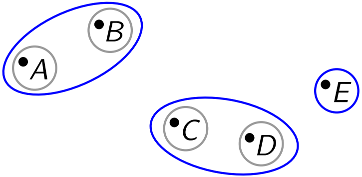


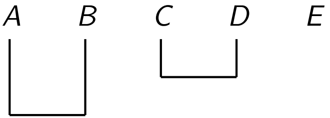
AGGLOMERATIVECOMPLETELINKAGE

- Start with one cluster for each input object.
- Iteratively merge the two closest clusters.

Complete linkage measure

$$D_{CL}(C_1, C_2) = \max_{x \in C_1, y \in C_2} D(x, y)$$



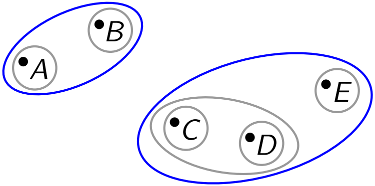


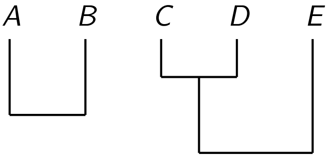
AgglomerativeCompleteLinkage

- Start with one cluster for each input object.
- Iteratively merge the two closest clusters.

Complete linkage measure

$$D_{CL}(C_1, C_2) = \max_{x \in C_1, y \in C_2} D(x, y)$$



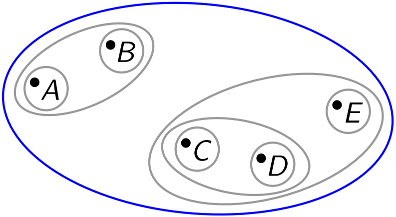


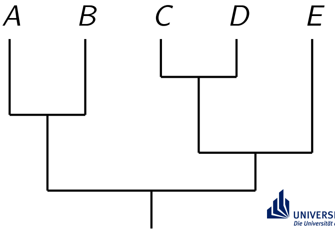
AgglomerativeCompleteLinkage

- Start with one cluster for each input object.
- Iteratively merge the two closest clusters.

Complete linkage measure

$$D_{CL}(C_1, C_2) = \max_{x \in C_1, y \in C_2} D(x, y)$$



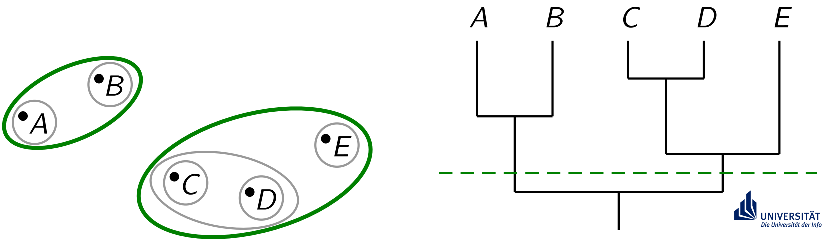


AgglomerativeCompleteLinkage

- Start with one cluster for each input object.
- Iteratively merge the two closest clusters.

Complete linkage measure

$$D_{CL}(C_1, C_2) = \max_{x \in C_1, y \in C_2} D(x, y)$$



AgglomerativeCompleteLinkage

- Start with one cluster for each input object.
- Iteratively merge the two closest clusters.

Complete linkage measure

$$D_{CL}(C_1, C_2) = \max_{x \in C_1, y \in C_2} D(x, y)$$