Advances in Explainable Clustering and Hierarchical Clustering

Eduardo Laber

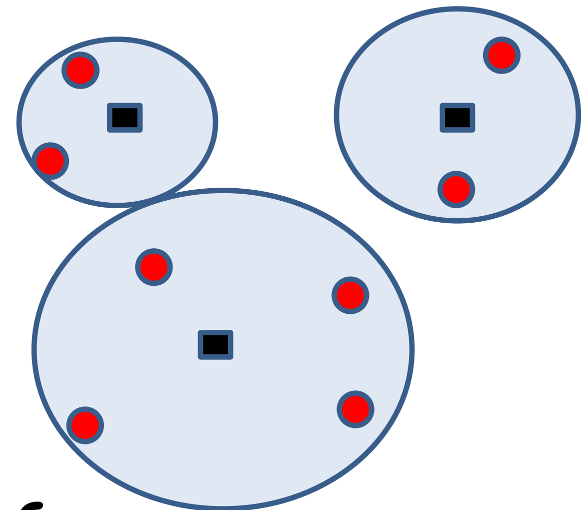
Departamento de Informática, PUC-RIO

laber@inf.puc-rio.br

Clustering

- Wide range of applications
  - Reducing computational resources
  - Data analysis
- Testbed problem for developing algorithmic techniques
- Vast literature available

(Hard) Clustering Problem



Input

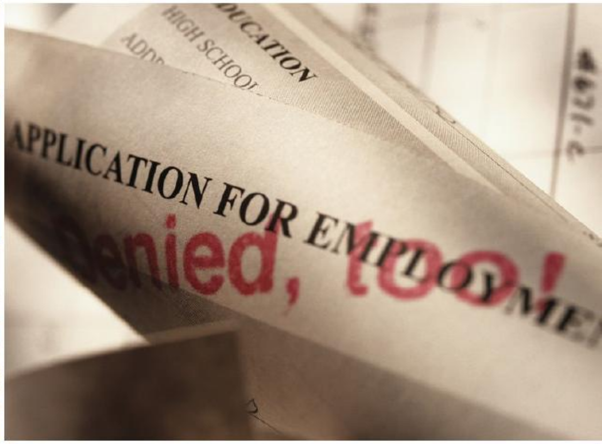
- $X=\{x_1,\ldots,x_n\}$ points
- k: #clusters
- Optimization criterion *f*

Output

- Partition of X into k groups optimizing *f*

Part I: Explainable Clustering

Machine Learning



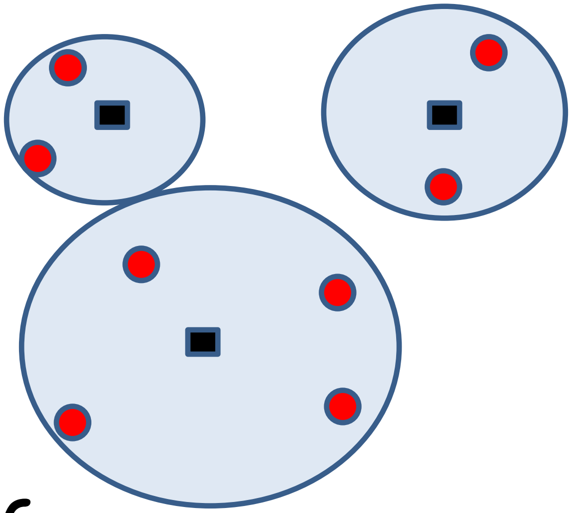
Machine Learning

Issues

- We don't trust models
- The rational behind some decision is not clear
- We don't know what happens in extreme cases
- Mistakes can be expensive/harmful
- How to change model when things go wrong?

Interpretability is one way we try to deal with these problems

Clustering Problem (Explainable)



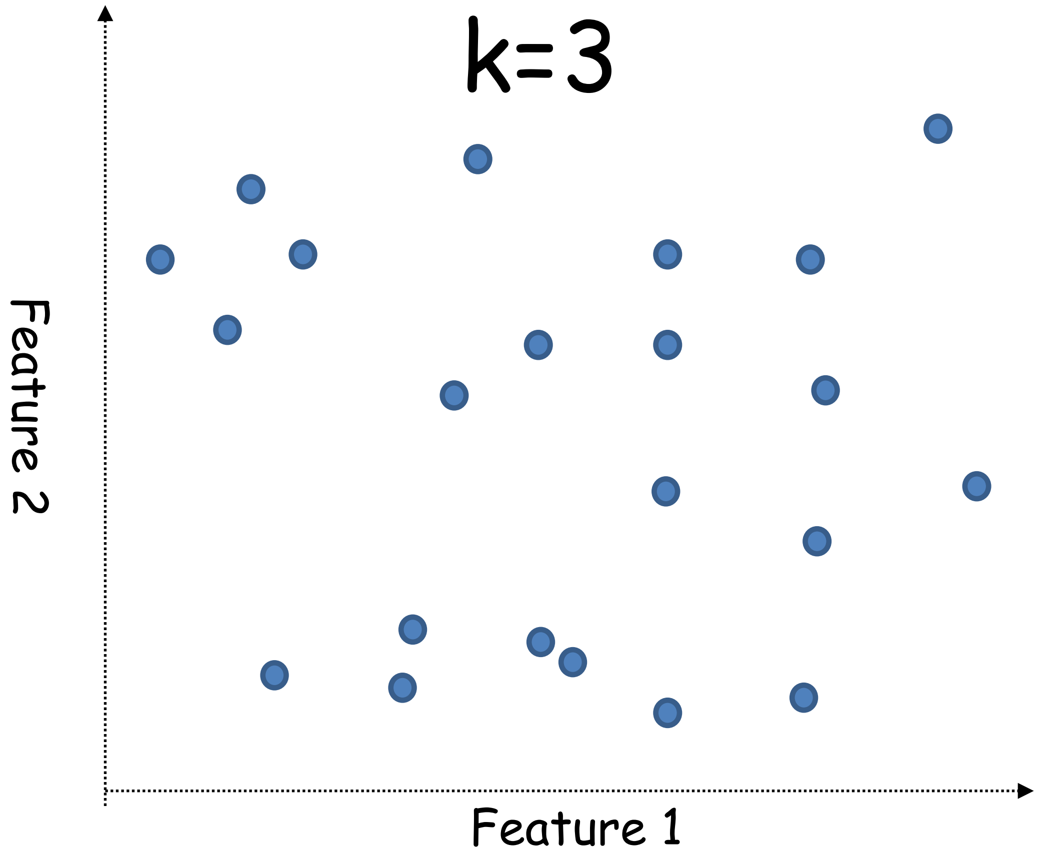
Input

- $X=\{x_1,\ldots,x_n\}$ points
- k: #clusters
- Optimization criterion f

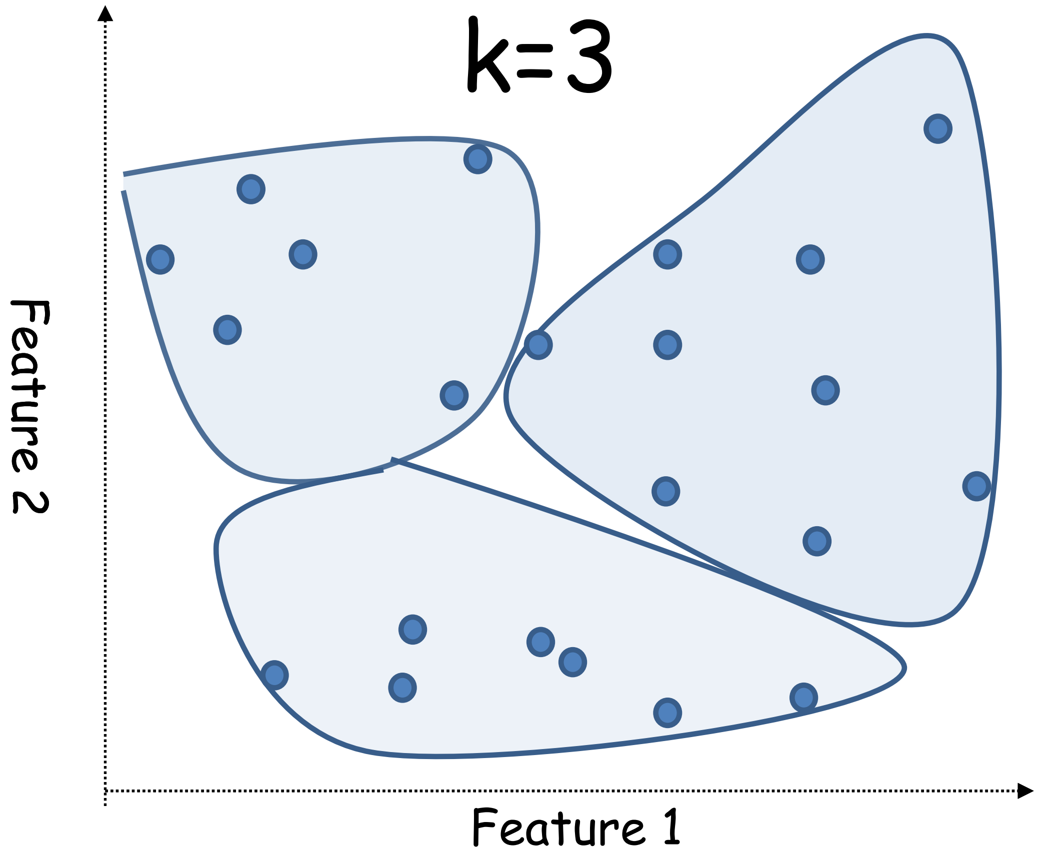
Output

- Partition of X into k groups optimizing f
- Partition must have a simple explanation

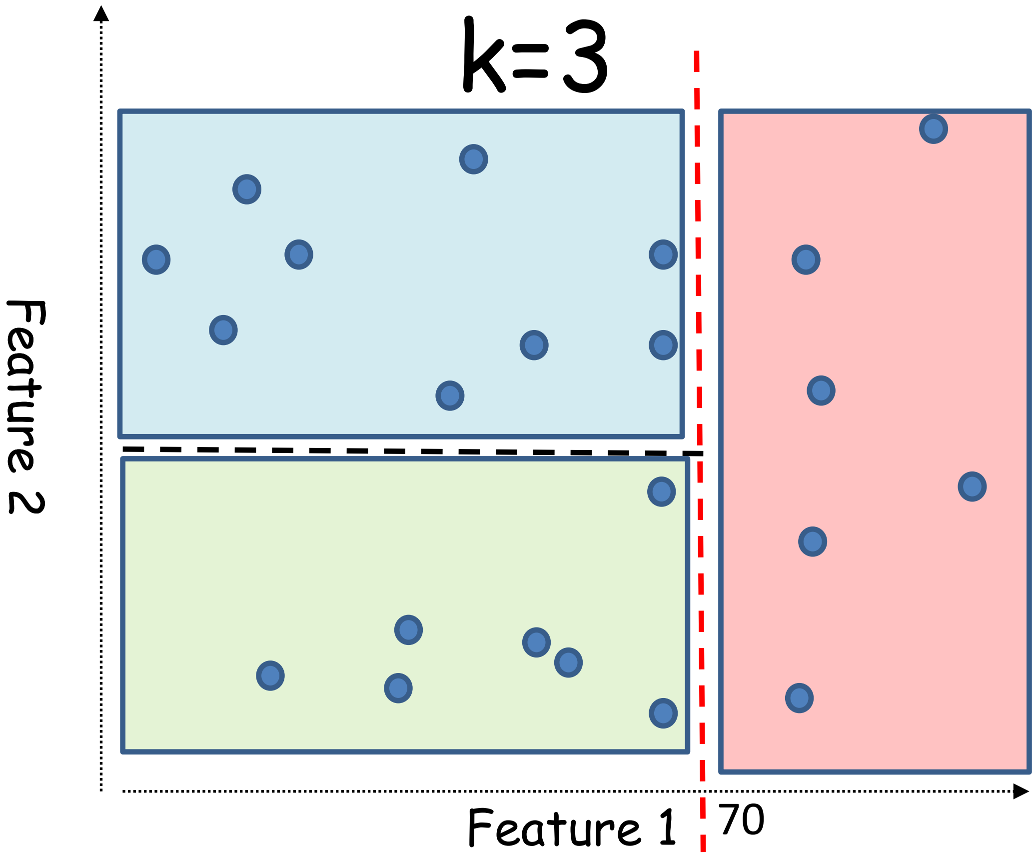
Explainable Clustering

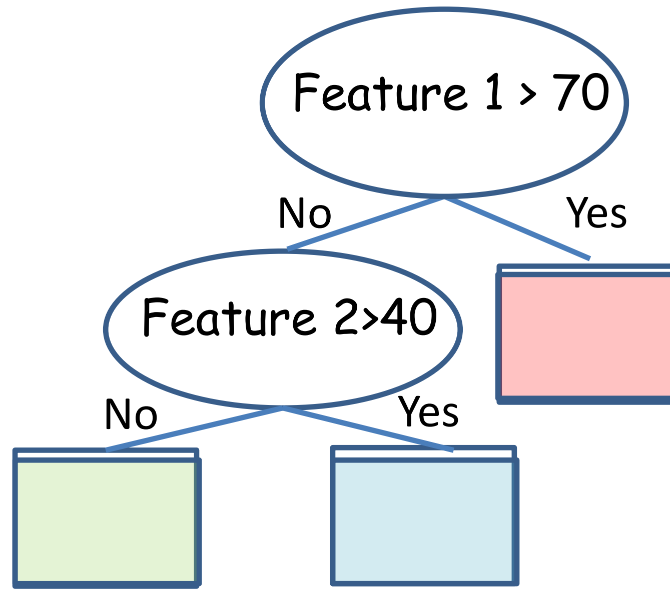


Explainable Clustering



Decision Tree Explanation





Decision Tree Clustering

Input

- $X=\{x_1,\ldots,x_n\}$ points in R^d
- k: #clusters
- Optimization criterion *f*

Output

- Partition of X into k groups optimizing *f* via decision trees with k leaves

Decision Tree Clustering

Research Questions

- Efficient algorithms for explainable clustering
- Price of Explainability

Price of the Explainability

Mathematically …

For a minimization criterion

$$Price = MAX_I \left\{ \frac{OPT_{Explainable}(I)}{OPT_{unrestricted}\ (I)} \right\}$$

Instances

Some Optimization Criteria

k-center

- Worst case
- Intra clustering

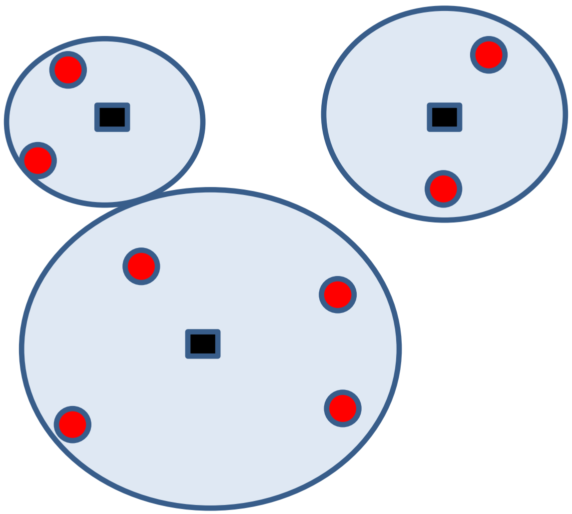
k-medians / k-means

- Average case
- Intra Cluster

Maximum spacing

- Worst Case
- Inter Clustering
- Hierarchical Clustering (single-link)

k-medians



Input

- $X=\{x_1,\dots,x_n\}$ points in R^d
- k: #clusters

Output

- k centers so that the sum of the ℓ_1 distances from the points in X to their closest centers is minimized

$$kmedians(X) = \sum_{x \in X} |x - center(x)\,|_1$$

k-medians

Theorem [Dasgupta et al, ICML 20]

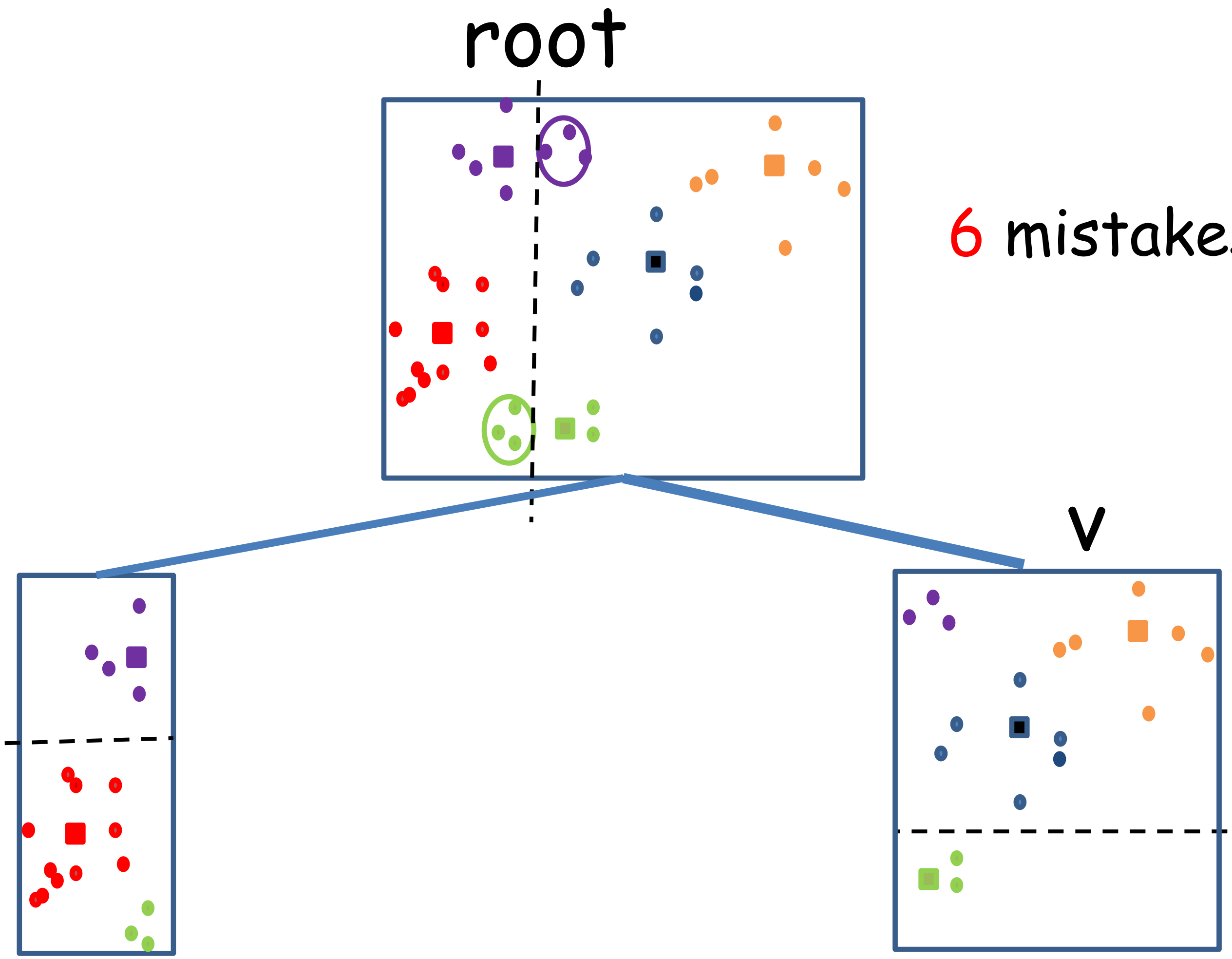
The price of explainability for k-medians is $O(k)$ and $\Omega(\log k)$

k-medians

IMM Algorithm

1. Obtain k reference centers (via some standard clustering algorithm)
2. While there is a cluster with more than one reference center
   - Apply an axis-aligned cut that minimizes the number of mistakes among those that separate two centers

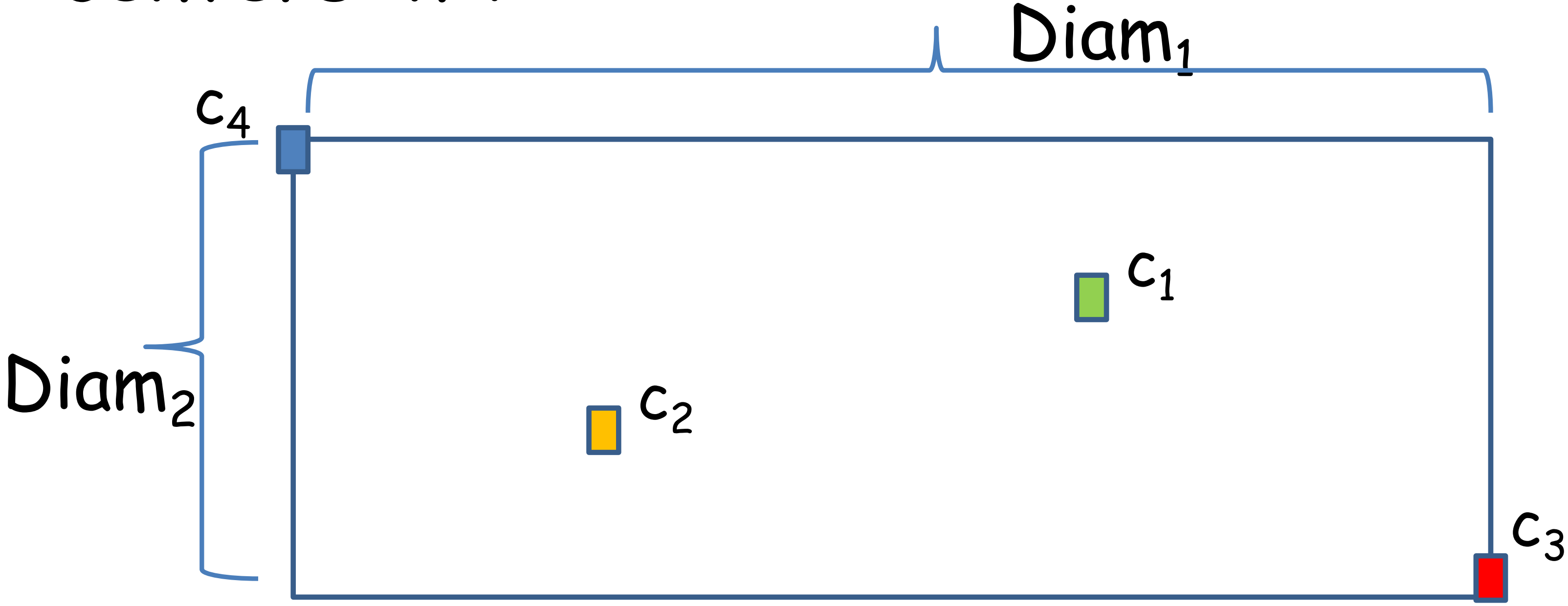
IMM



6 mistakes

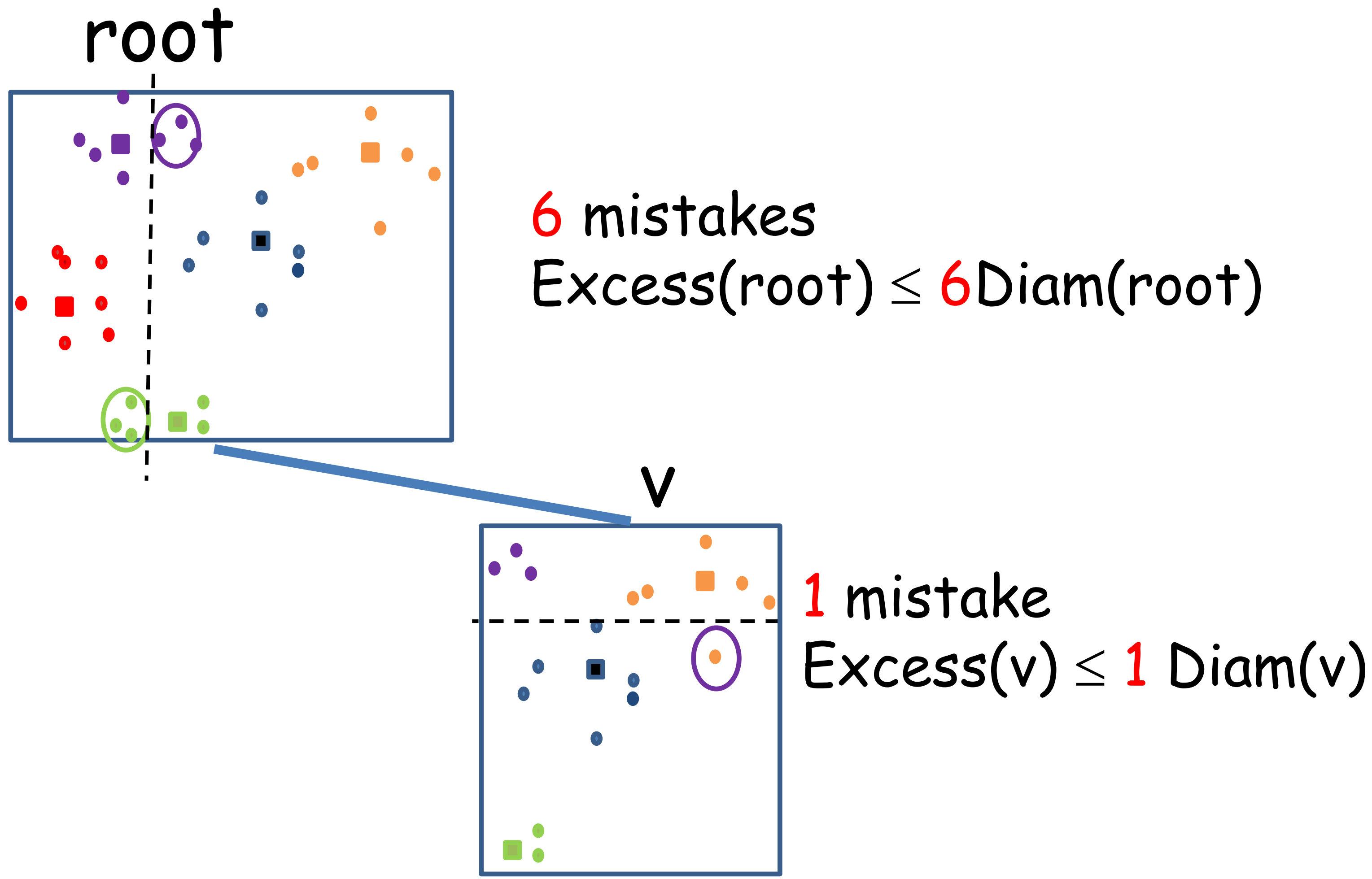
IMM Analysis

Diam(v): sum of the lengths of the bounding box that contains all reference centers in v



$$Diam = Diam_1 + Diam_2$$

IMM analysis: Upper Bound



$$Cost(D) \leq OPT_{unrest} + \sum_{v \in D} Excess(v)$$

IMM analysis: Upper Bound

6 mistakes

Excess(root) ≤ 6Diam(root)

1 mistake

Excess(v) ≤ 1 Diam(v)

$$Cost(D) \leq OPT_{unrest} + \sum_{v \in D} MinMistakes(v)\ Diam(v)$$

IMM Analysis: Lower Bound

- $center_i(p)$: component i of the center that is closest to point *p*
- OPT_i : contribution of component i to OPT_{unrest}

$$OPT_i = \sum_p |p_i - center_i(p)|$$

- Write OPT_i as a function of the mistakes introduced by the cuts

$$OPT_i = \sum_p |p_i - center_i(p)| \geq MinMistakes_i \times Diam_i$$

$$OPT_{unrest} = \sum_i OPT_i \geq MinMistakes \times Diam$$

k-median: Price of Explainability

Theorem. IMM is an O(k) approximation

- Upper Bound

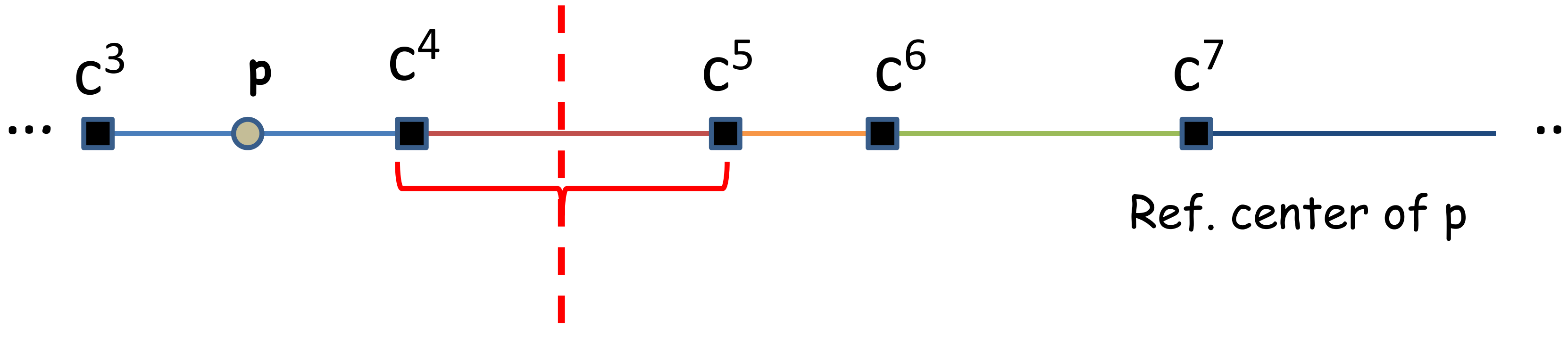
$$Cost(D) \leq OPT_{unrest} + \sum_{v \in D} MinMistakes(v)\ Diam(v)$$

- Lower Bound

$$2k \times OPT_{unrest} \geq \sum_{v \in D} MinMistakes(v)\ Diam(v)$$

IMM Analysis: Lower Bound

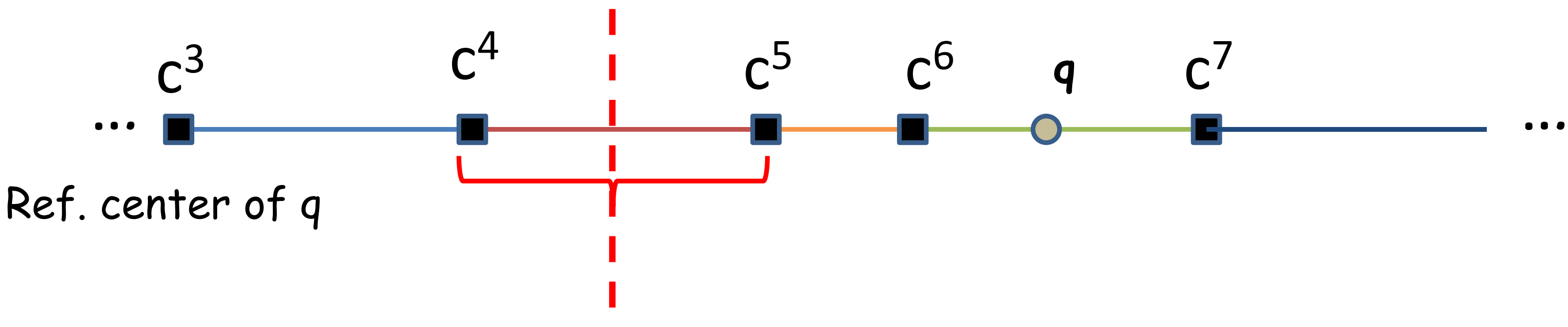
Consider component i



- red cut makes mistake on **p**
- we can add $|c^4 - c^5|$ to the lower bound

IMM Analysis: Lower Bound

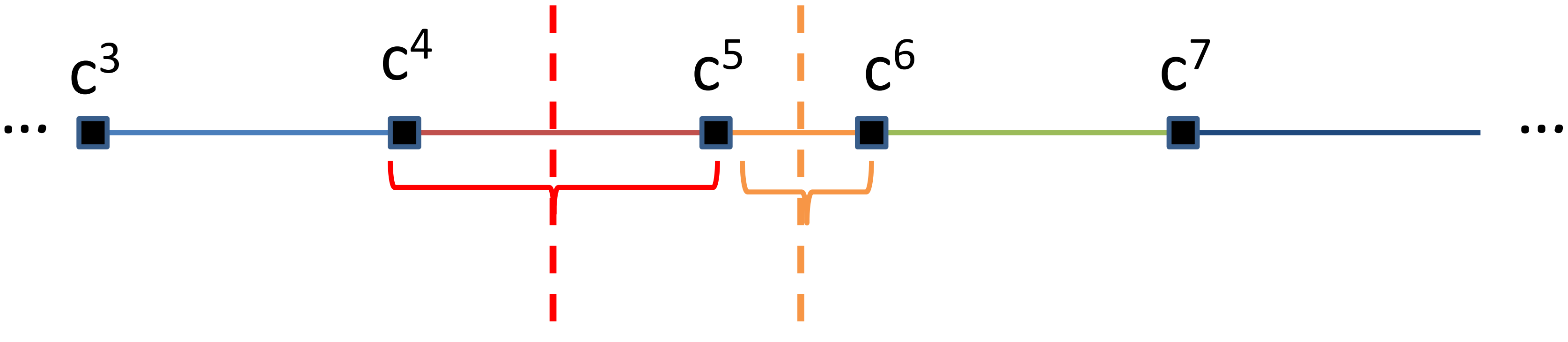
Consider component i



- red cut makes mistake on **p**
- $|p-c^7|>|c^4-c^5|$. Add $|c^4-c^5|$ to the lower bound
- red cut makes mistake on **q**
- $|q-c^3|>|c^4-c^5|$. Add $|c^4-c^5|$ to the lower bound

IMM Analysis: Lower Bound

Consider component i

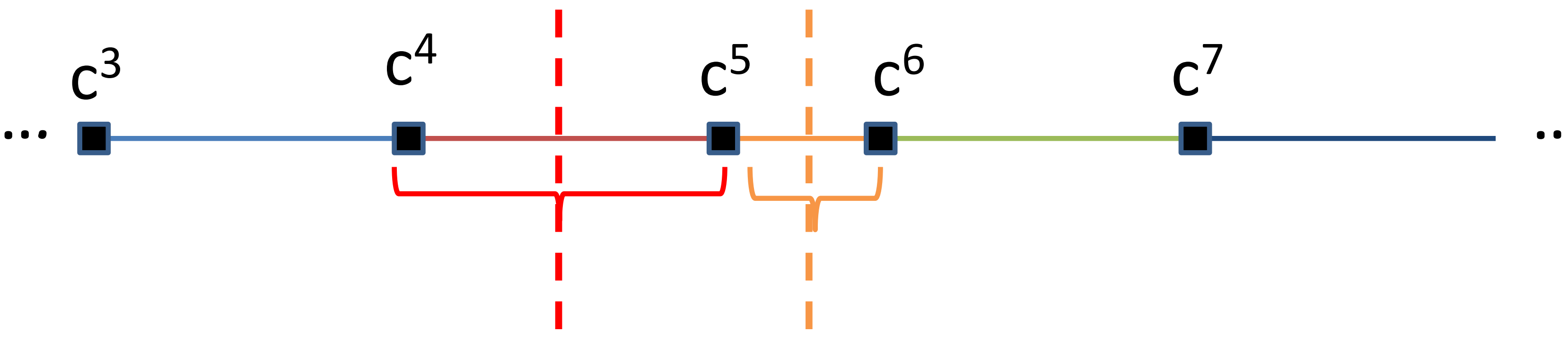


$$OPT_i = \sum_p |p - center(p)| \geq \quad * \left|c_i^4 - c_i^5\right| +$$

mistakes

IMM Analysis: Lower Bound

Consider component i



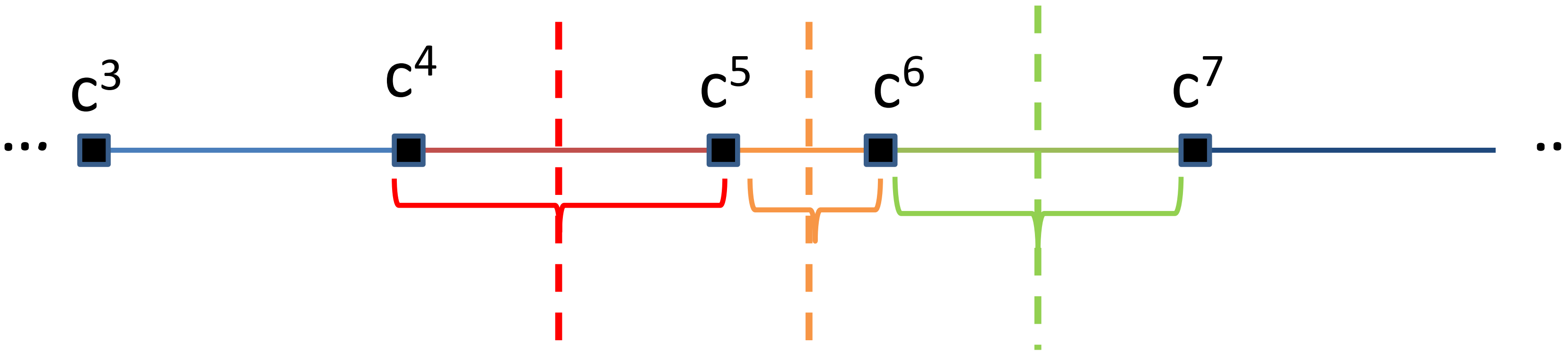
$$OPT_i = \sum_p |p - center(p)| \geq \quad * \left|c_i^4 - c_i^5\right| + \quad * \left|c_i^6 - c_i^5\right| +$$

\# mistakes

\# mistakes

IMM Analysis: Lower Bound

Consider component i



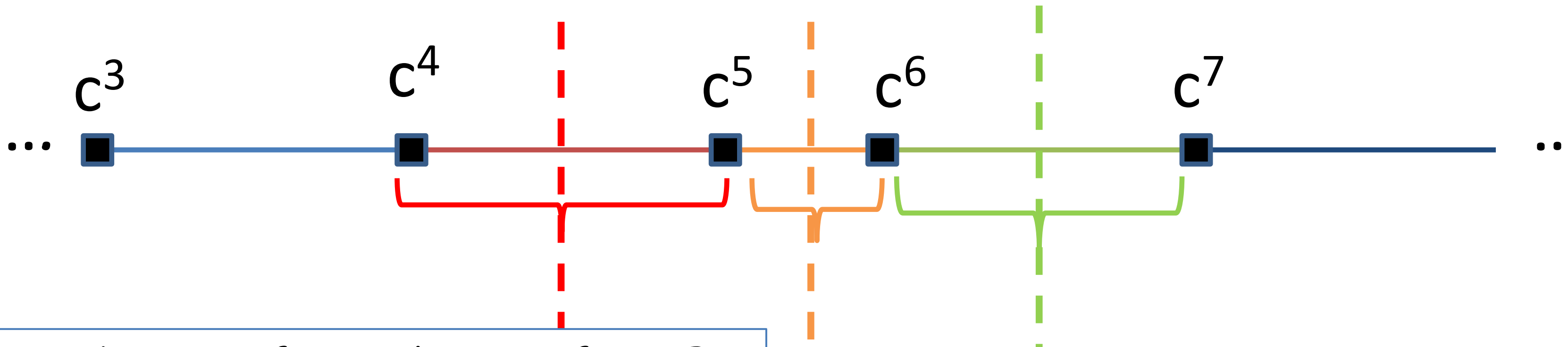
$$OPT_i = \sum_p |p - center(p)| \geq \;\; * \left|c_i^4 - c_i^5\right| + \;\; * \left|c_i^6 - c_i^5\right| +$$

mistakes

mistakes

IMM Analysis: Lower Bound

Consider component i



Contribution of coordinate i for OPT

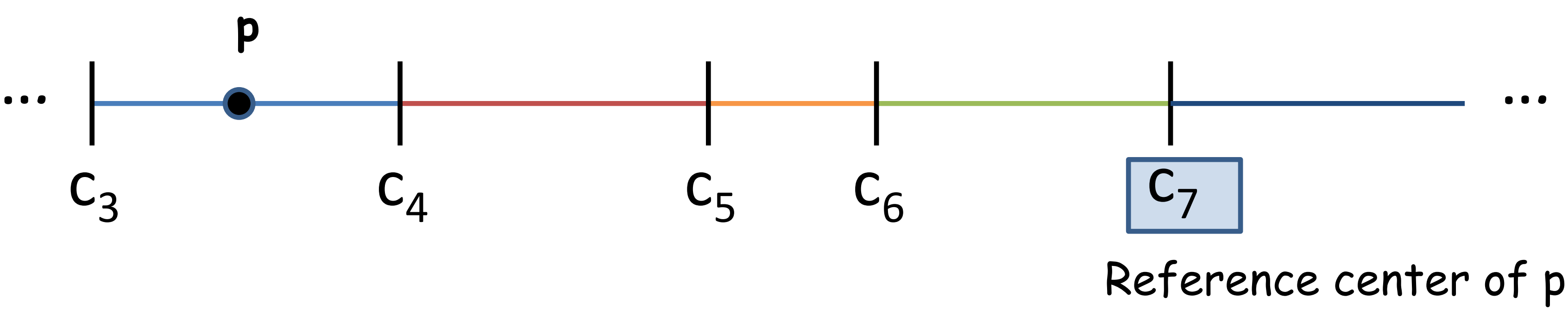
$$OPT_i = \sum_p |p_i - center(p)| \geq \;\; * \left|c_i^4 - c_i^5\right| + \;\; * \left|c_i^6 - c_i^5\right| + \;\; * \left|c_i^7 - c_i^6\right|$$

\# mistakes # mistakes # mistakes

$$OPT_i = \sum_p |p - center_i(p)| \geq MinMistakes_i \times Diam_i$$

IMM Analysis: Lower Bound

Consider component i



$$OPT_{unrest} = \sum_{p \in v} |p - center_i(p)| \geq$$

$$\sum_i MinMistakes_i \times Diam_i(v) =$$

$$MinMistakes \times Diam(v)$$

k-median: Price of Explainability

Theorem. IMM is an O(k) approximation

- Upper Bound

$$Cost(D) \leq OPT_{unrest} + \sum_{v \in D} MinMistakes(v)\ Diam(v)$$

- Lower Bound

$$2k \times OPT_{unrest} \geq \sum_{v \in D} MinMistakes(v)\ Diam(v)$$

$\Omega(\log k)$ Lower Bound

Bad instance

- First pick k random centers $c_1,\ldots,c_k$ from the hypercube $\{-1, 1\}^d$;
- Create k clusters $C_1,\ldots,C_k$
  - C_i has d points
  - jth point of C_i : replace the j-th component of center c_i with 0

$$c_i = (-1,-1,1,-1,1) \rightarrow p_{i3} = (-1,-1,0,-1,1)$$

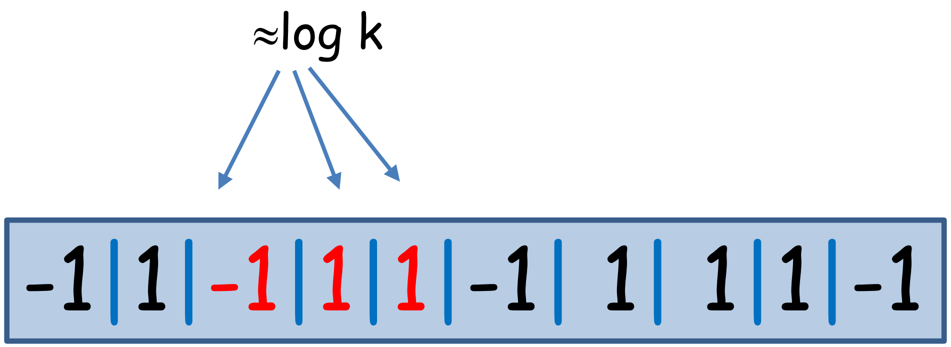
$\Omega(\log k)$ Lower Bound

Properties of Bad Instance

- $d=k^3$
- $OPT \leq dk$
- $dist(c_i,c_j) \geq d/4$ (centers are far apart)

$\Omega(\log k)$ Lower Bound

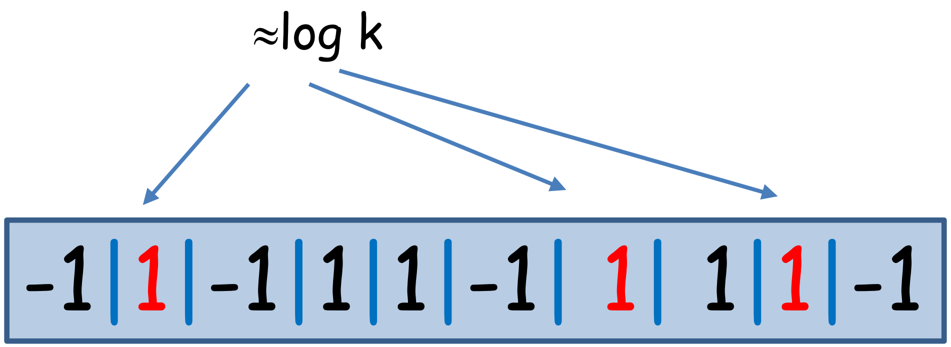
Properties of Bad Instance



$k^{49/50}$ *centers agree with the red values*

Ω(log k) Lower Bound

Properties of Bad Instance



$k^{49/50}$ *centers agree with the red values*

$\Omega(\log k)$ Lower Bound

Properties of Bad Instance

- for any subset $S \subset \{1,\ldots, d\}$, with

$$|S| = \log k / 50$$

and any $\{-1,1\}$-assignment for the components in S, there are about

$$k^{49/50}$$

centers that agree with this assignment

Ω(log k) Lower Bound

- The last property implies that any tree with k leaves will have "many" points from different clusters in the same leaf
- These points are at least d/4
- Thus, the cost of any tree is

$$\Omega(\log(k)\, d\, k) = \Omega(\log(k)\, OPT)$$

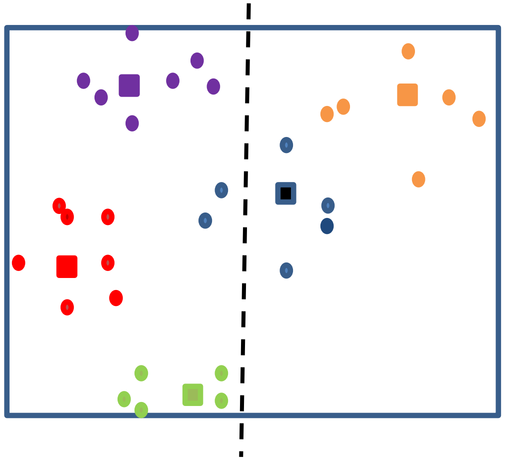
O(log k) Upper bound

Random Cuts Algorithm

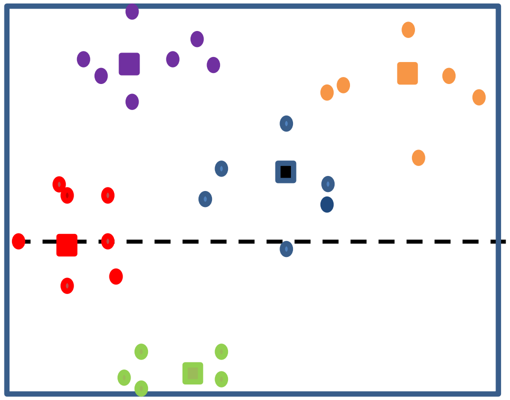
1. Run an unrestricted clustering algorithm to obtain k reference centers

2. *Repeatedly select threshold cuts uniformly at random among those that separate reference centers*

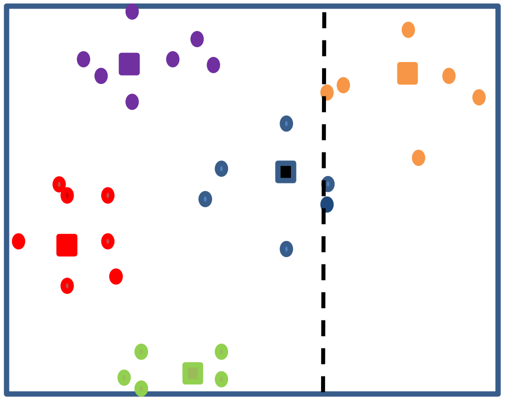
Random Cut



Random Cut



Random Cut



Random Cuts Algorithm

OPT: optimal solution of unrestricted clustering

Theorem [Gupta 23 & Makarychev 23]
The random cut algorithm builds a threshold trees with expected cost

$$O(\log k \; OPT)$$

Random Cuts Algorithm

Theorem [Weak version] With probability $\geq(1-1/k)$ the algorithm produces a tree with

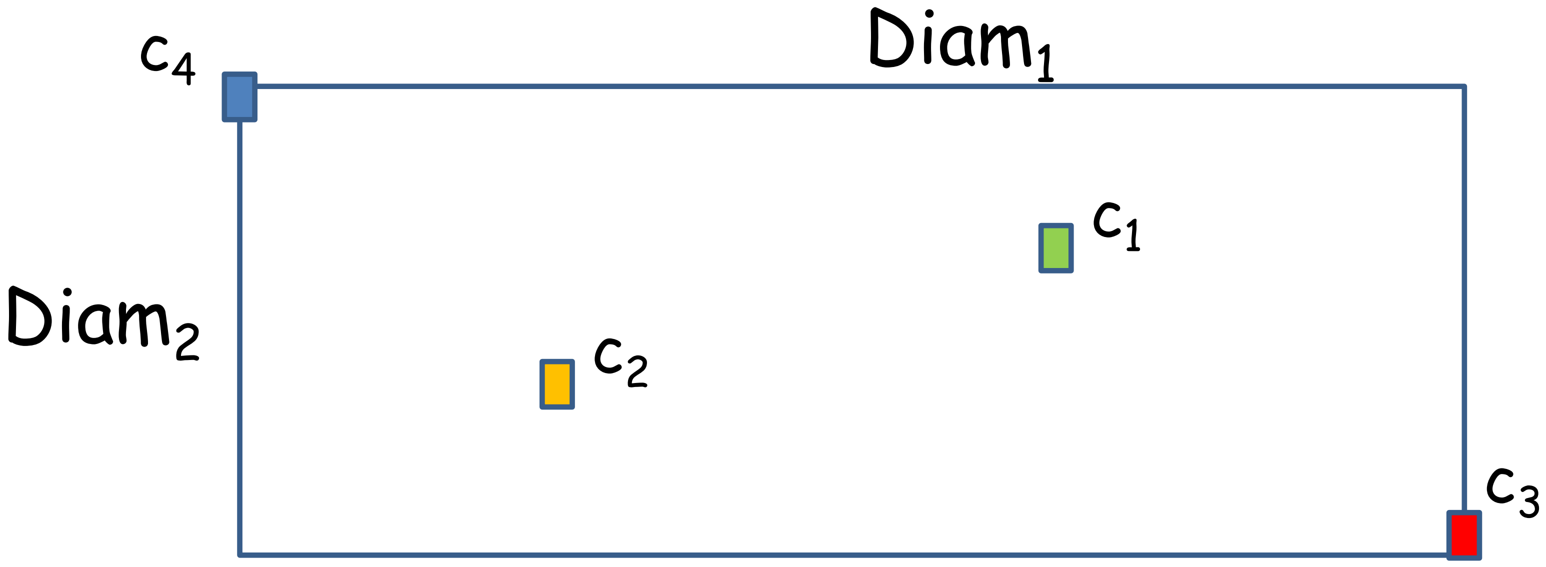
$$\text{cost} \leq \log\left(\frac{c_{max}}{c_{min}}\right) \log(k)\ OPT$$

c_{max}: maximum distance between two reference centers

c_{min}: minimum distance between two reference centers

Random Cut Algorithm

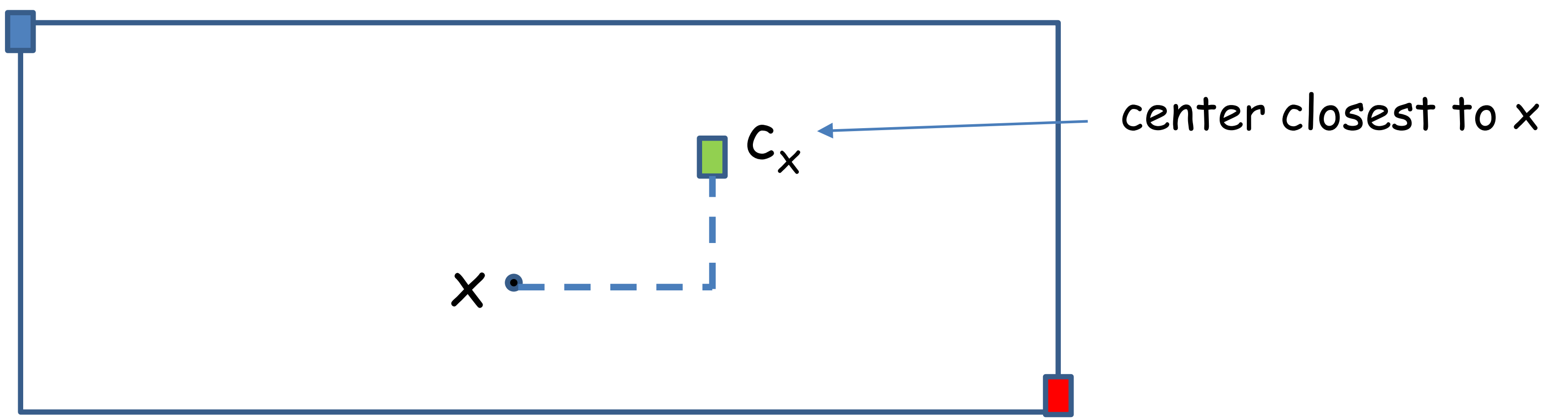
Diam(v): sum of the lengths of the bounding box that contains all reference centers in node v



$$\text{Diam} = \text{Diam}_1 + \text{Diam}_2$$

Random Cuts Algorithm

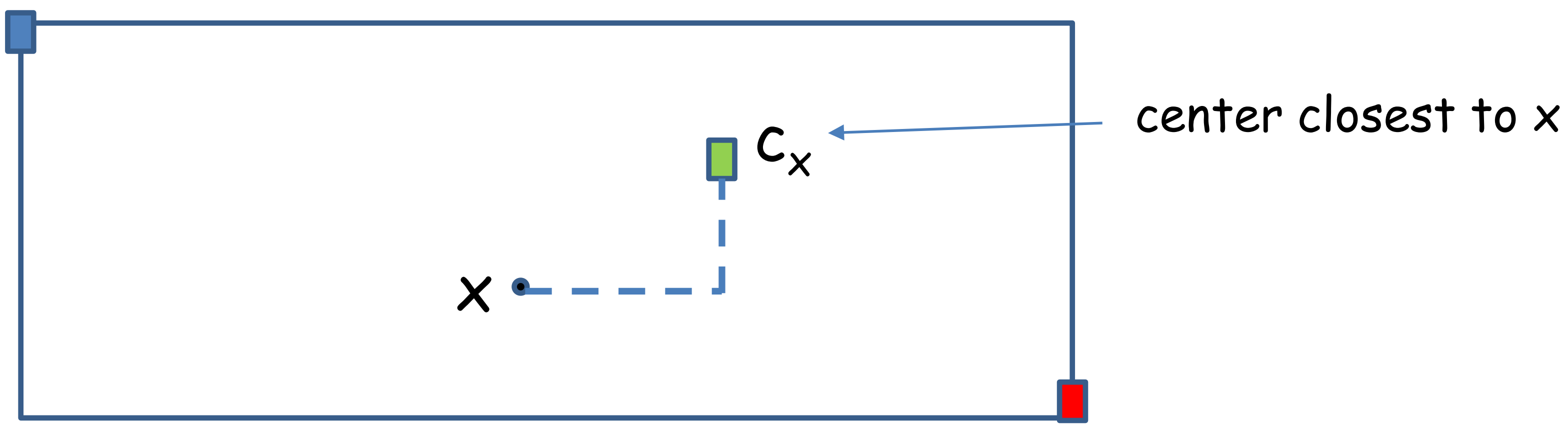
Lemma. The expected number of points separated from their closest centers by a random cut is ≤ OPT/Diam



$$Prob[cut\ separates\ x\ from\ c_x] = \frac{|x - c_x|_1}{Diam}$$

Random Cuts Algorithm

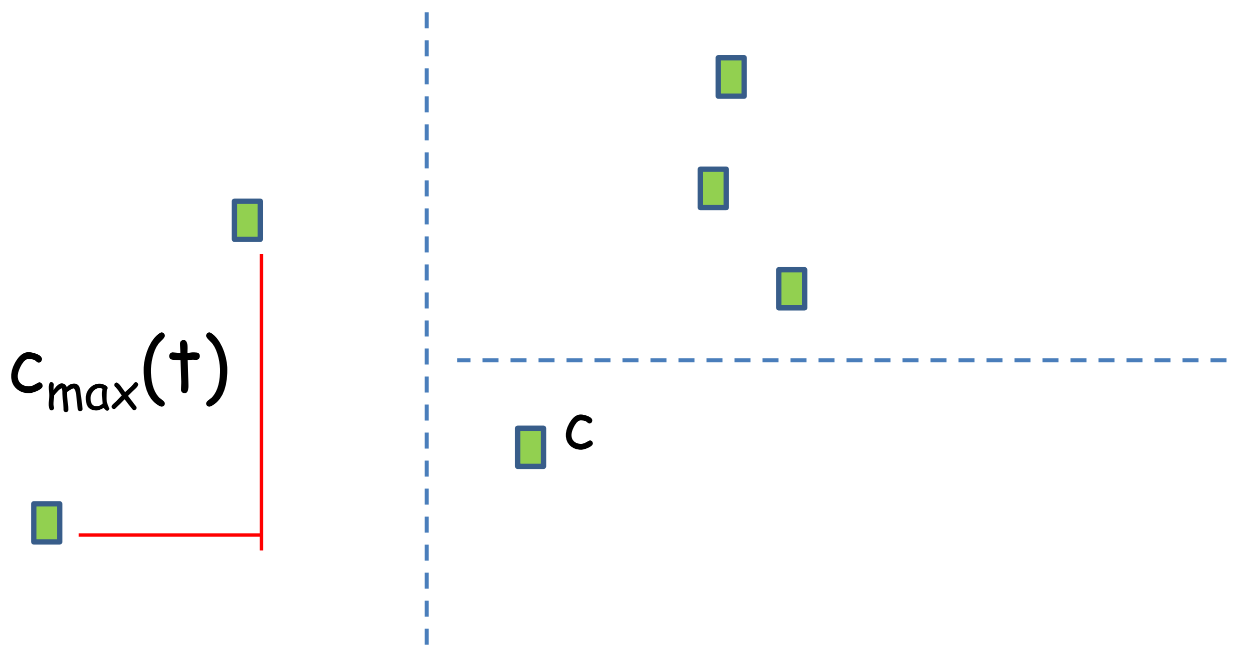
Lemma. The expected number of points separated from their closest centers by a random cut is ≤ OPT/Diam



$$\sum_x Prob[cut\ separates\ x\ \ from\ c_x] = \frac{\sum_x |x - c_x|_1}{Diam} = \frac{OPT}{Diam}$$

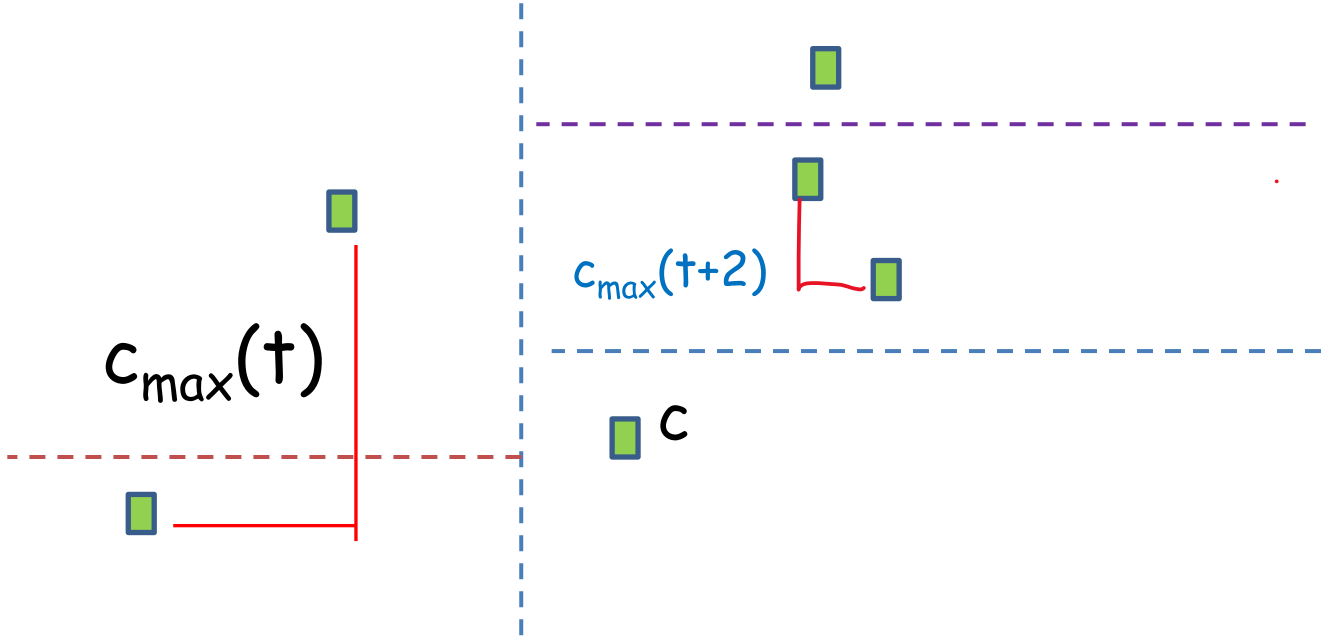
Random Cuts Algorithm

$c_{max}(t)$: maximum distance between centers in the same leaf/region after t iterations.



Random Cuts Algorithm

Lemma. After $M = 3\ \mathrm{Diam}\ \ln(k) / c_{max}(t)$ iterations with high probability the maximum distance between two centers is divided by 2



Random Cuts Algorithm

Lemma. After $M=3\ Diam\ \ln(k) / c_{max}(t)$ iterations with high probability the maximum distance between two centers is divided by 2

Proof

- Pick 2 centers at distance $\geq c_{max}(t)/2$
- The probability they are not separated in the M iterations is

$$\left(1 - \frac{c_{max}(t)/2}{Diam}\right)^M \leq \frac{1}{k^3}$$

- Union bound on k^2 centers

Random Cuts Algorithm

$sep_i(t)$: set of points separated from their closest centers at iteration t

$$cost\ (Alg) \leq OPT + E\left[\sum_t c_{\max(t)} sep_i\ (t)\right]$$

- R(i): number of iterations with $c_{max}(t)$ in $\left[\frac{c_{max}}{2^i}, \frac{c_{max}}{2^{i+1}}\right]$

$$cost(Alg) \leq OPT + E\left[\sum_{i=0}^{\log(c_{max}/c_{min})} \sum_{t \in R(i)} c_{\max(t)} sep_i\ (t)\right]$$

Random Cuts Algorithm

$$cost(Alg) \leq OPT + E\left[\sum_{i=0}^{\log(c_{max}/c_{min})} \sum_{t \in R(i)} c_{\max(t)} sep_i(t)\right]$$

– $sep_i(t) \leq \frac{OPT}{Diam}$ (Lemma 1)

– $\mathrm{R(i)} \leq \frac{3\, Diam \ln(k)}{c_{max}(t)}$, with high probability (Lemma 2)

$$\sum_{t \in R(i)} c_{\max(t)} sep_i(t) \approx \frac{3 Diam \log(k)}{c_{max}(t)} \times \frac{\mathrm{OPT}}{Diam} \times c_{max}(t) \approx 3 \ln(k)\, OPT$$

$$cost(Alg) \leq OPT + \log\left(\frac{c_{max}}{c_{min}}\right) \log k\ OPT$$

Random Cuts Algorithm

Modified Algorithm

- Sample uniformly a cut that does not separate two centers that are within distance at most $c_{max}(t)/k^4$

Theorem With probability $\geq (1-1/k)$ the algorithm produces a threshold tree with

$$\text{cost} \leq \log^2(k)\ \mathrm{OPT}$$

Extensions

K-means

- Random cut sampling from a different distributions

Theorem[Gupta 23] Random Cuts produces a tree with

$$\text{cost } O(k \log(k)\, OPT)$$

Experiments

		Normalized Partition Cost					
Dataset	k	SHA	BIS	GRD	IMM	KMC	RDM
anuran	10	1.16	1.21	**1.15**	1.28	1.32	1.71
avila	12	**1.05**	1.13	1.05	1.07	1.18	1.35
beer	104	1.16	**1.07**	1.19	1.83	1.27	1.55
bng	24	1.05	**1.01**	1.02	1.04	1.03	1.05
cifar10	10	1.16	**1.15**	1.17	1.22	1.19	1.26
collins	30	1.18	**1.16**	1.17	1.23	1.23	1.42
covtype	7	1.03	1.10	1.03	**1.03**	1.13	1.34
digits	10	**1.19**	1.19	1.21	1.23	1.22	1.42
iris	3	**1.04**	1.10	**1.04**	**1.04**	**1.04**	1.45
letter	26	**1.19**	1.30	1.23	1.30	1.36	1.53
mice	8	**1.07**	1.09	1.09	1.12	1.15	1.37
newsgroups	20	1.05	1.01	**1.01**	1.01	1.01	1.01
pendigits	10	**1.14**	1.18	1.14	1.24	1.32	1.70
poker	10	**1.10**	1.11	1.10	1.10	1.12	1.14
sensorless	11	1.02	1.05	**1.02**	1.03	1.07	1.32
vowel	11	**1.21**	1.21	1.25	1.36	1.29	1.50

Random Cuts are not great in practice ☹

References

- *Explainable k-Means and k-Medians Clustering*
  Dasgupta et. al , ICML 2020

- *Nearly-Tight and Oblivious Algorithms for Explainable Clustering*
  Gamlath et. al, Neurips 2021

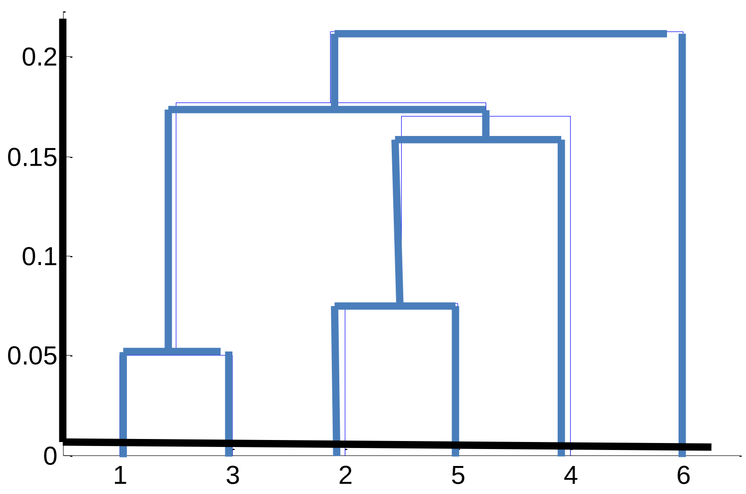
- Random Cuts are Optimal for Explainable k-Medians
  Makarychev & Shan, Neurips 2023

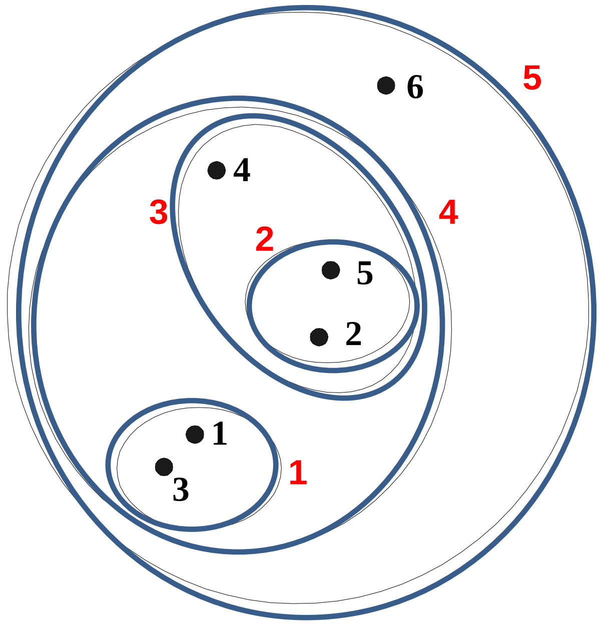
- Price of Explainable Clutering
  Gupta et. al, Arxiv 2023

Part II: Hierarchical Clustering

Hierarchical Clustering

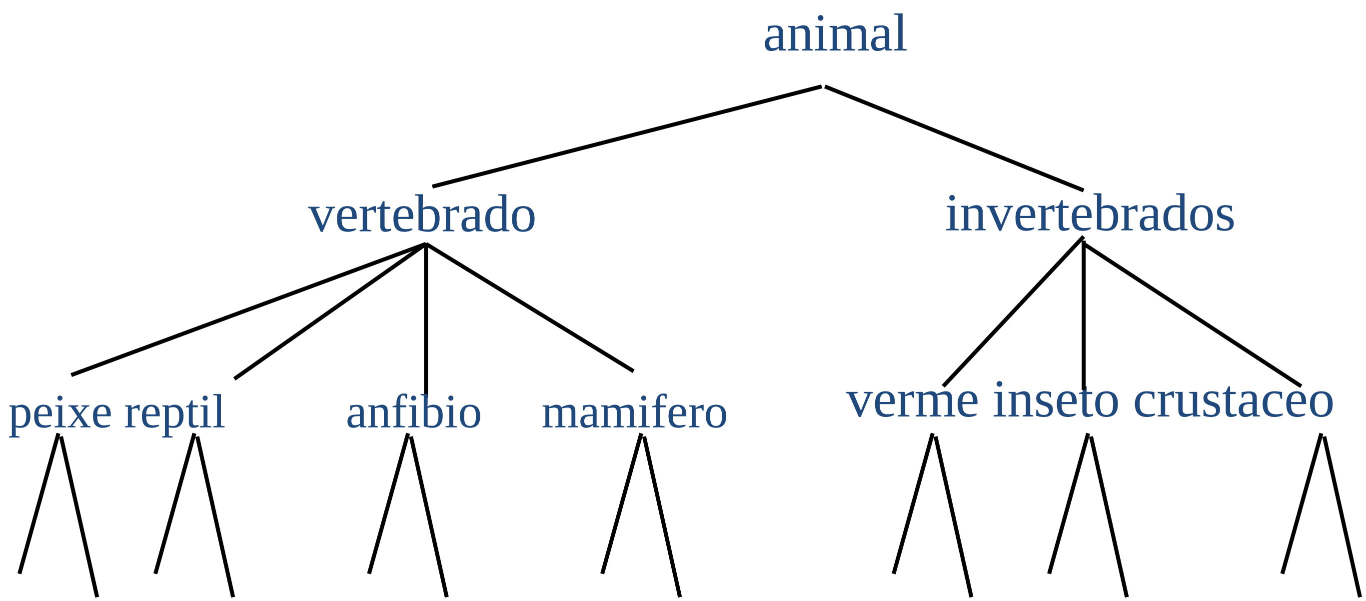
- For every k, it induces a clustering with k clusters
- Can be visualized by a dendogram
  - Trees that keep track of the merges or divisions employed to build the clustering

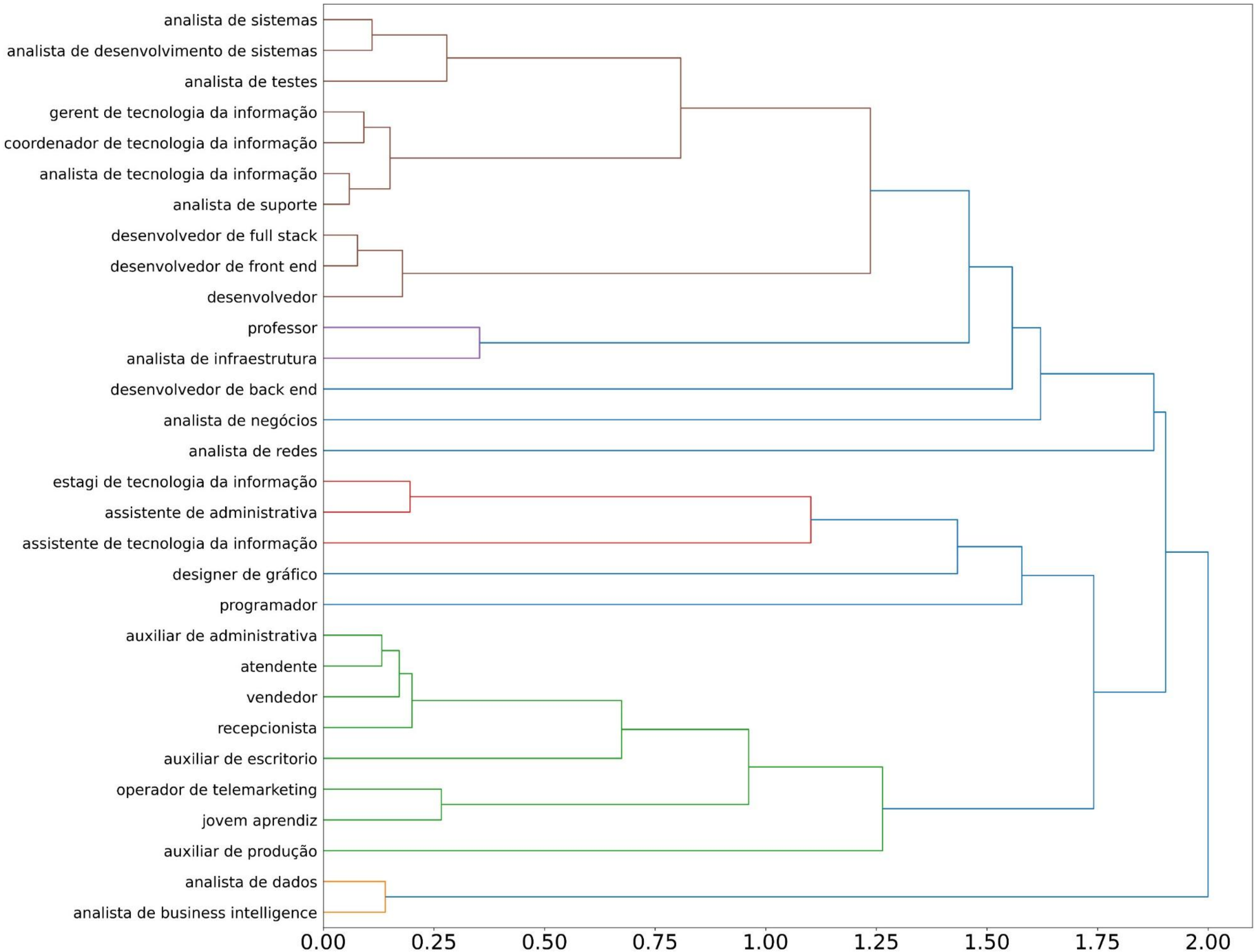




Hierarchical Clustering

- Number of clusters not pre-defined
- Tree may correspond to a natural taxonomy





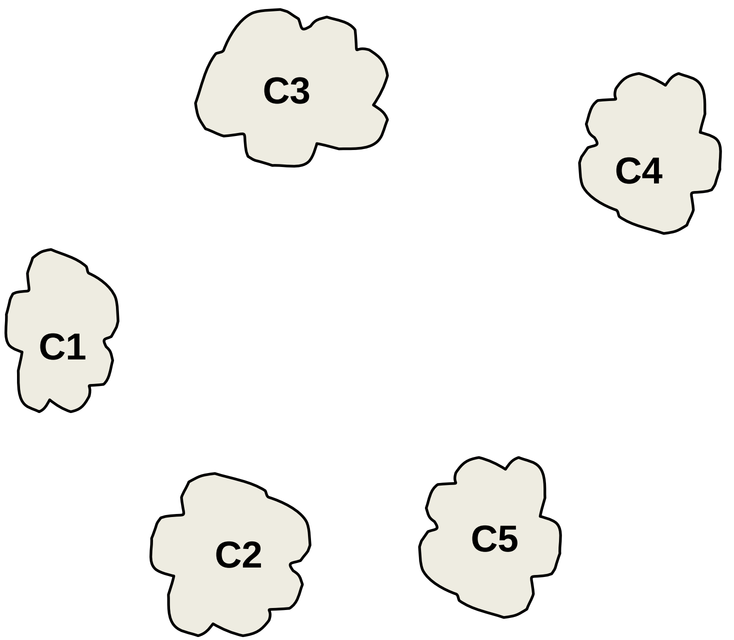
analista de sistemas
analista de desenvolvimento de sistemas
analista de testes
gerent de tecnologia da informação
coordenador de tecnologia da informação
analista de tecnologia da informação
analista de suporte
desenvolvedor de full stack
desenvolvedor de front end
desenvolvedor
professor
analista de infraestrutura
desenvolvedor de back end
analista de negócios
analista de redes
estagi de tecnologia da informação
assistente de administrativa
assistente de tecnologia da informação
designer de gráfico
programador
auxiliar de administrativa
atendente
vendedor
recepcionista
auxiliar de escritorio
operador de telemarketing
jovem aprendiz
auxiliar de produção
analista de dados
analista de business intelligence
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00

Basic Agglomerative Algorithm

- Compute the proximity between the points
- **Repeat** n-1 times
  - Merge the two “closest” clusters
  - Compute the proximity between the new group and the others

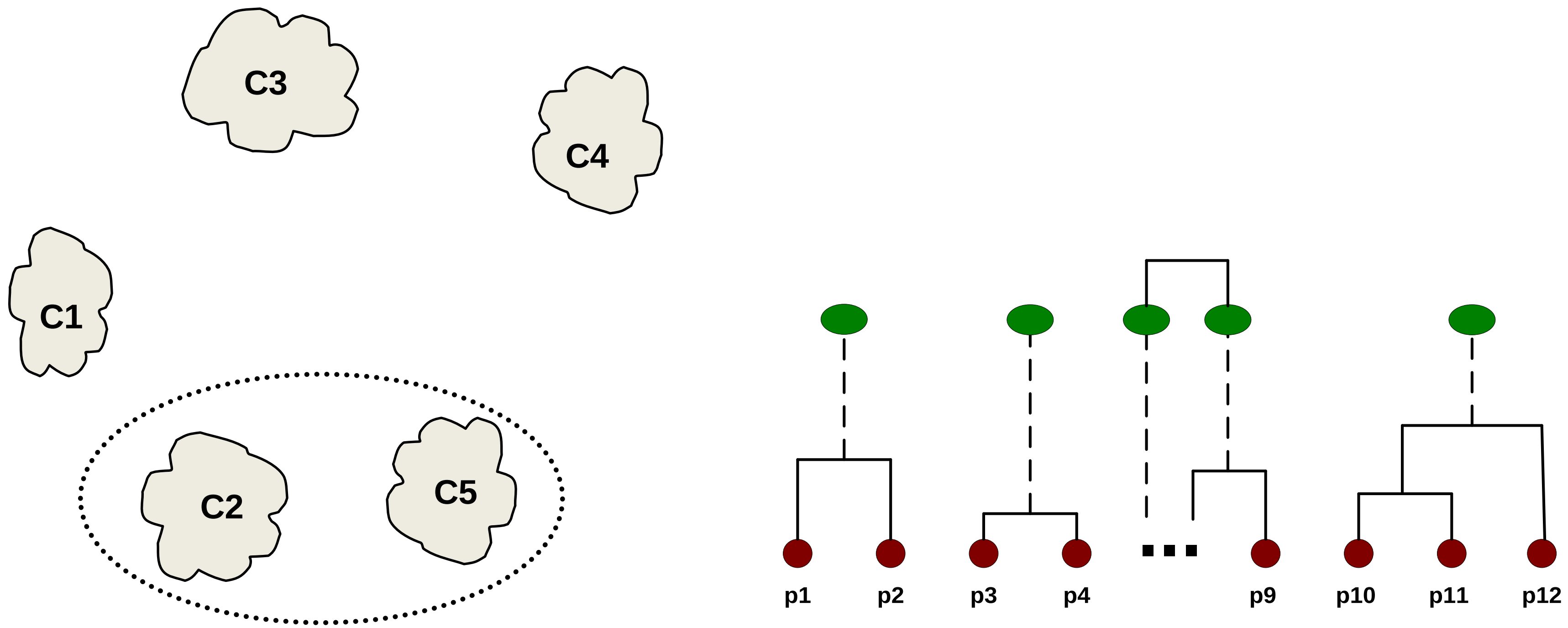
Basic Agglomerative Algorithm

- After a couple of merges we have some clusters:



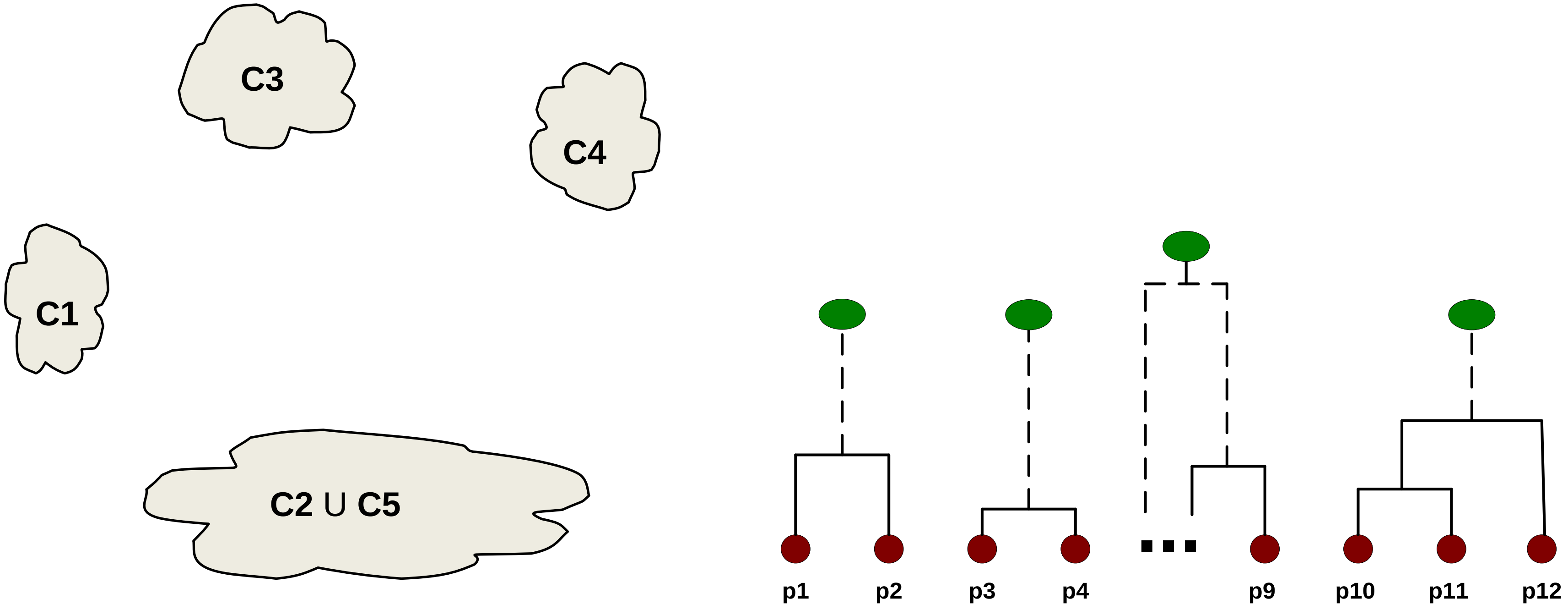
Basic Agglomerative Algorithm

- Merge the "closest" pair of groups and update the dendogram

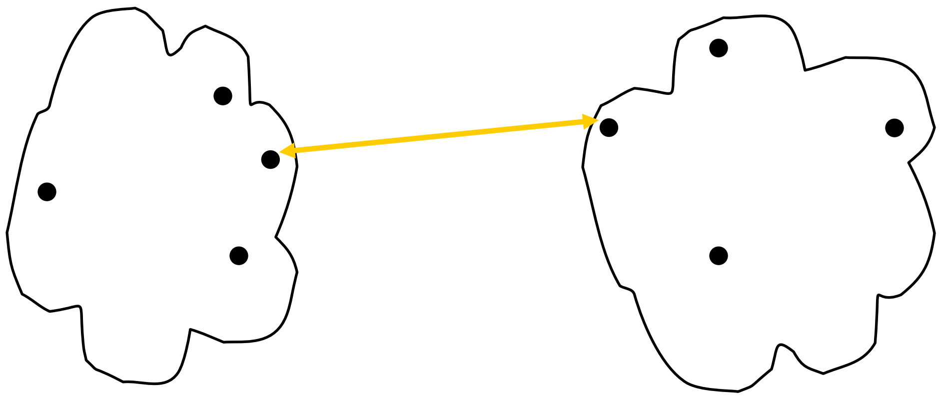


Basic Agglomerative Algorithm

- After the merge



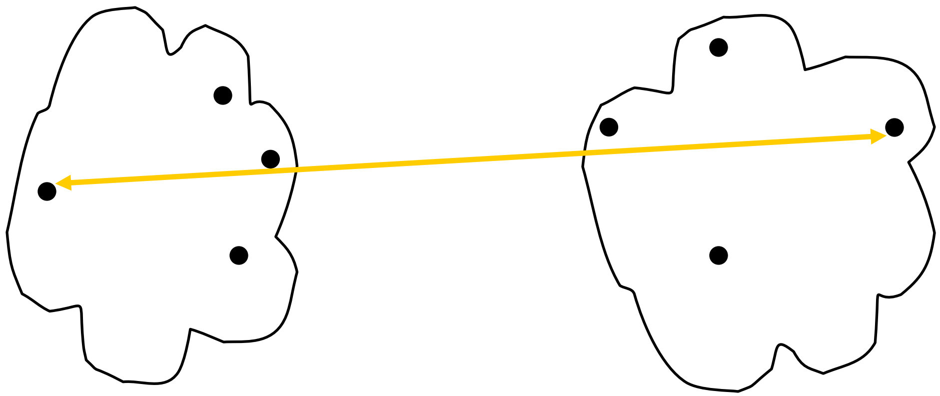
Proximity between groups



Linkage Methods

- Single-Link: two closest points
- Complete-Link:
- Average-Link

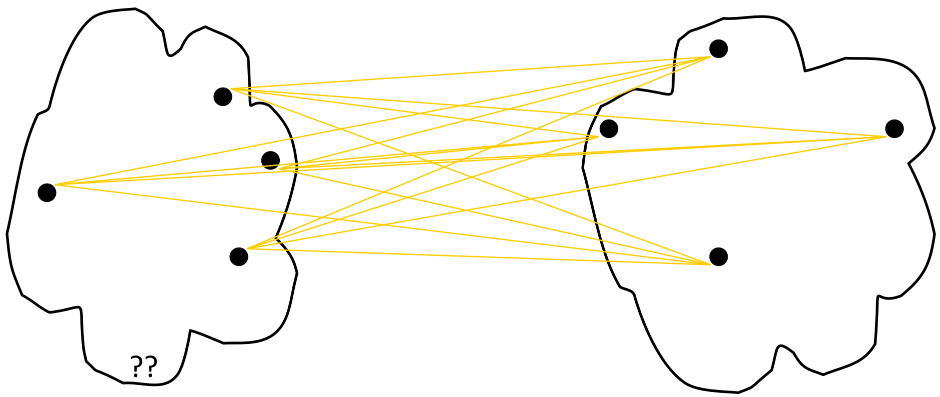
Proximity between groups



Linkage Methods

- Single-Link
- Complete-Link: two farthest points
- Average-Link

Proximity between groups

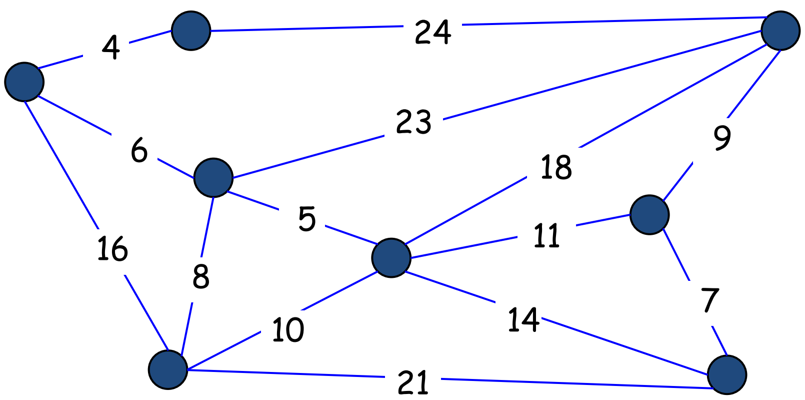


Linkage Methods

- Single-Link
- Complete-Link
- Average-Link: average distance among points

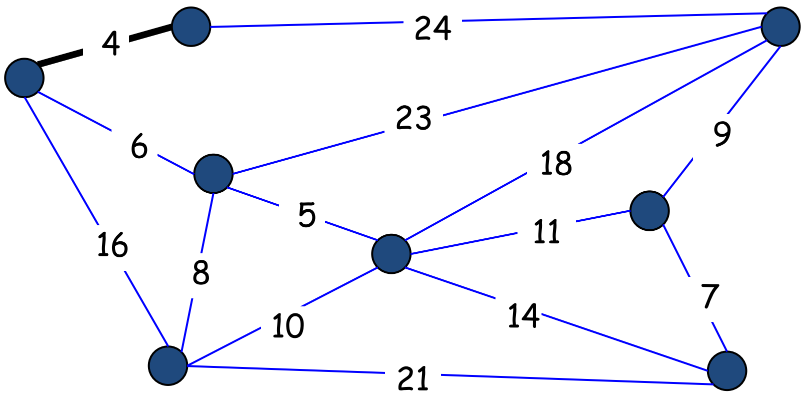
Single-Linkage

k=4



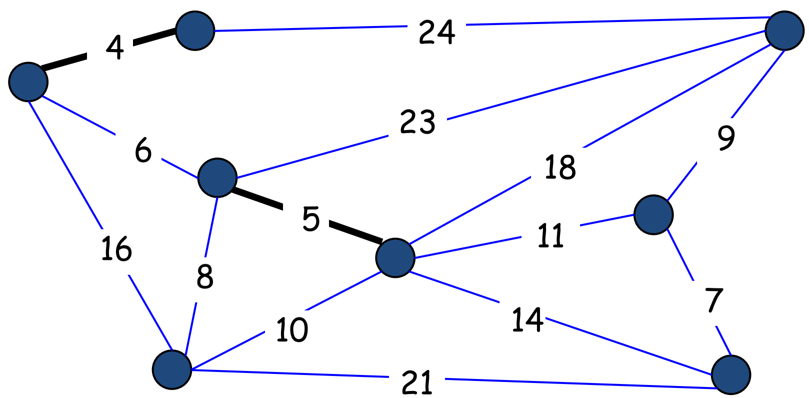
Single-Linkage

k=4



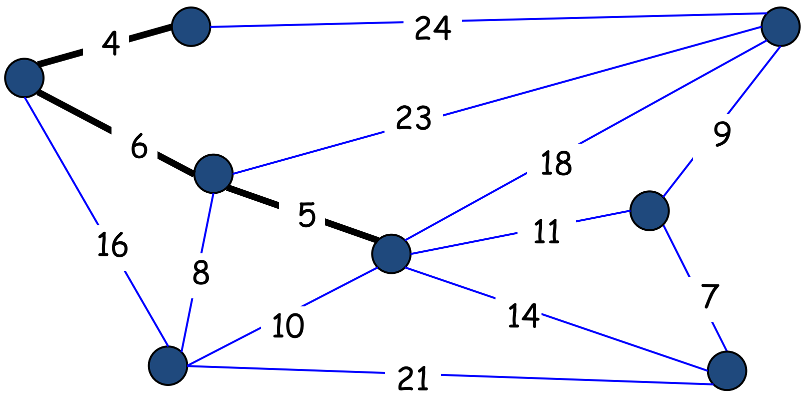
Single-Linkage

k=4



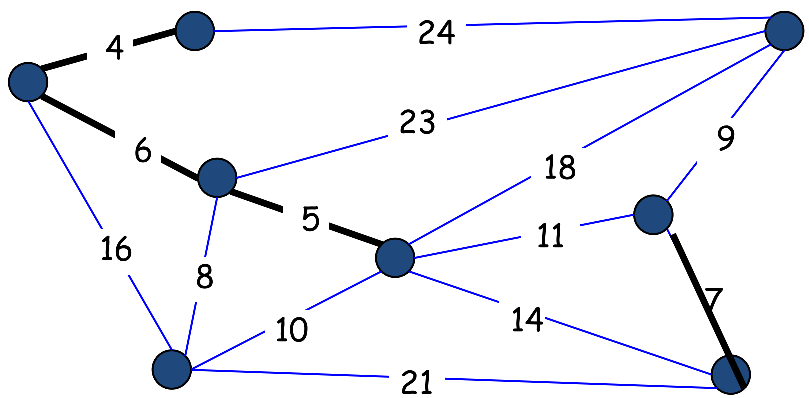
Single-Linkage

k=4



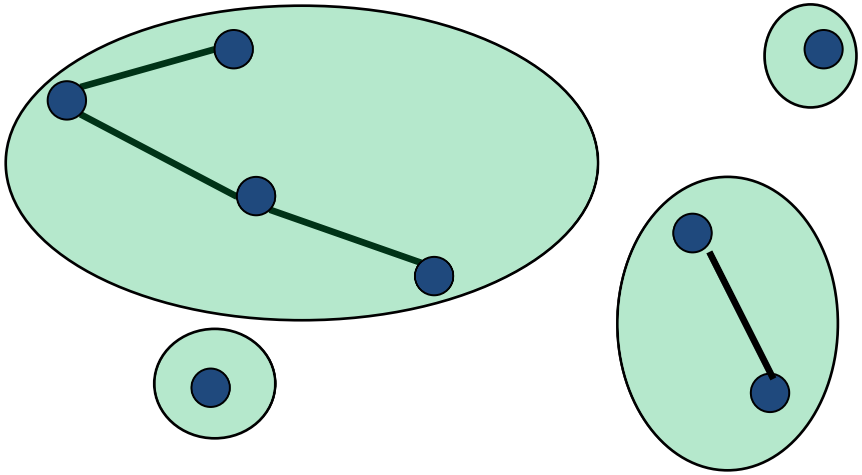
Single-Linkage

k=4



Single-Linkage

k=4



Linkage Methods

- Taught in introductory Machine Learning courses
- Available in many libraries as scipy, matlab, R, etc
- Good results reported in the literature for some methods (e.g. average-link and Ward)

Linkage Methods

- Many (recent) works proposing more efficient and scalable implementations
  - [Yu et al., VLDB 21]
  - [Dhulipala et al, ICML 21]
  - [Dhulipala et al, Neurips 22]

- Many (recent) work studying its theoretical properties
  - [Cohen-Addad et al., JACM 19]
  - [Mosely and Wang, JMLR 23]
  - [Arutyunova et al., Machine Learning 23]

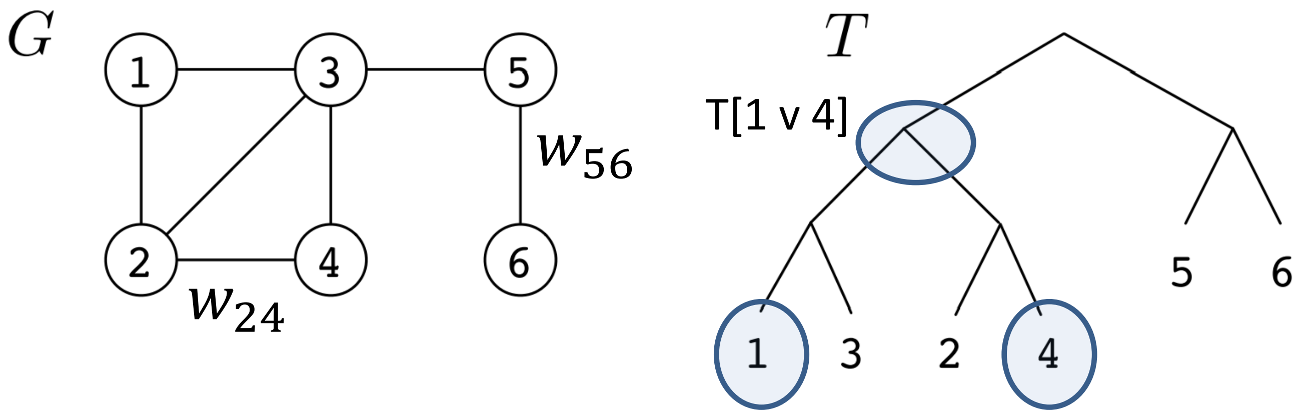
Research Questions

- More efficient and scalable methods?
- What cost functions do these methods optimize?
- Foundations for the good results reported in practice?
- Methods with better guarantees?

Research Questions

- More efficient and scalable methods?
- What cost functions do these methods optimize?
- Foundations for the good results reported in practice?
- Methods with better guarantees?

Dasgupta Objective Function



$$\mathrm{cost}_G(T) = \sum_{\{i,j\} \in E} w_{ij} \, |\mathtt{leaves}(T[i \vee j])|.$$

Similarity between i and j

common subtree of i and j

Similar items shall be merged early → tree below them has few leaves

Dasgupta's Objective Function

Pros

- One single objective function encompassing the tree hierarchy
- Work well for planted partition models

Cons

- All methods have approximately the same performance in metric spaces
- Interpretabilty
  - Not easy to explain for a practitioner

Cohesion and Separability

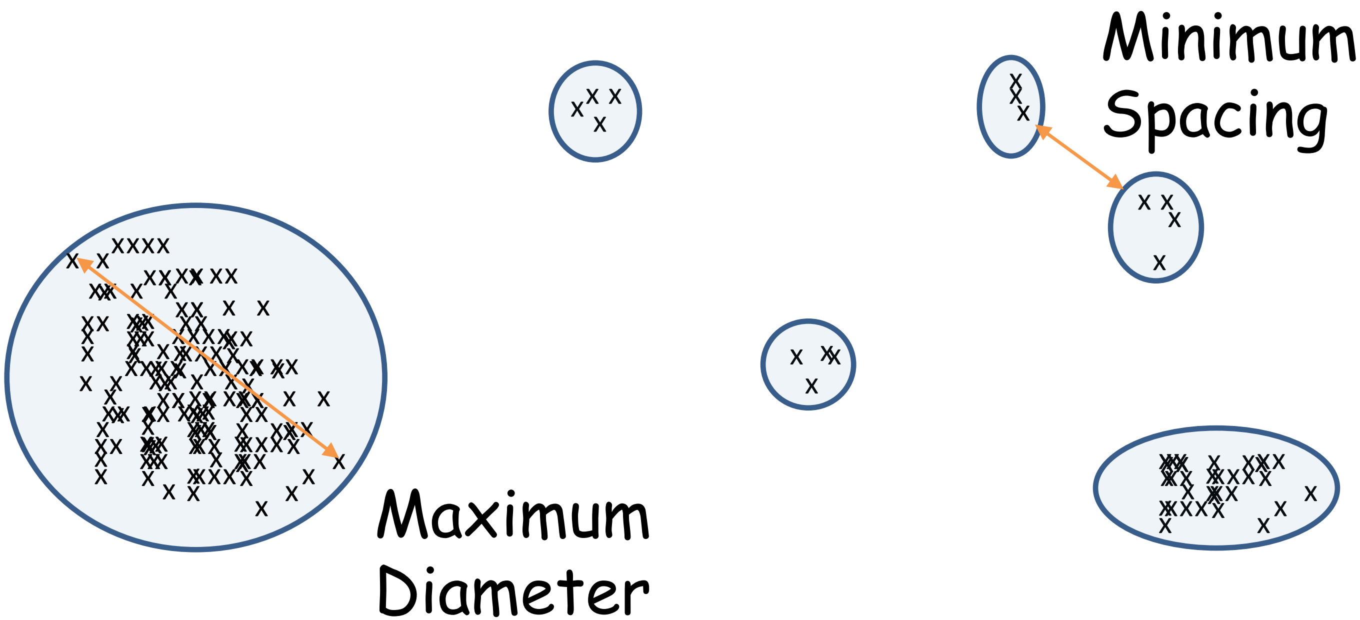
Cohesion (Intra-Group)

- Measure how compact are the clusters
  - Maximum diameter
  - Sum of pairwise distance
  - Sum of quadratic errors (k-means cost function)

Separability (Inter-Group)

- Measure how separated are distinct clusters
  - Minimum spacing
  - Average spacing

Cohesion and Separability



Clustering with k=6 clusters

Research

- *Optimization of inter-groups criteria for clustering with minimum size constraints*
  with L. Murtinho, Neurips 2023

- *New bounds on the cohesion of complete-link and other linkage methods for agglomerative clustering*
  With S. Dasgupta, ICML 2024

- *On the cohesion and separability of average-link forhierarchical agglomerative clustering*
  with M. Batista, Neurips 2024

Cohesion Criteria

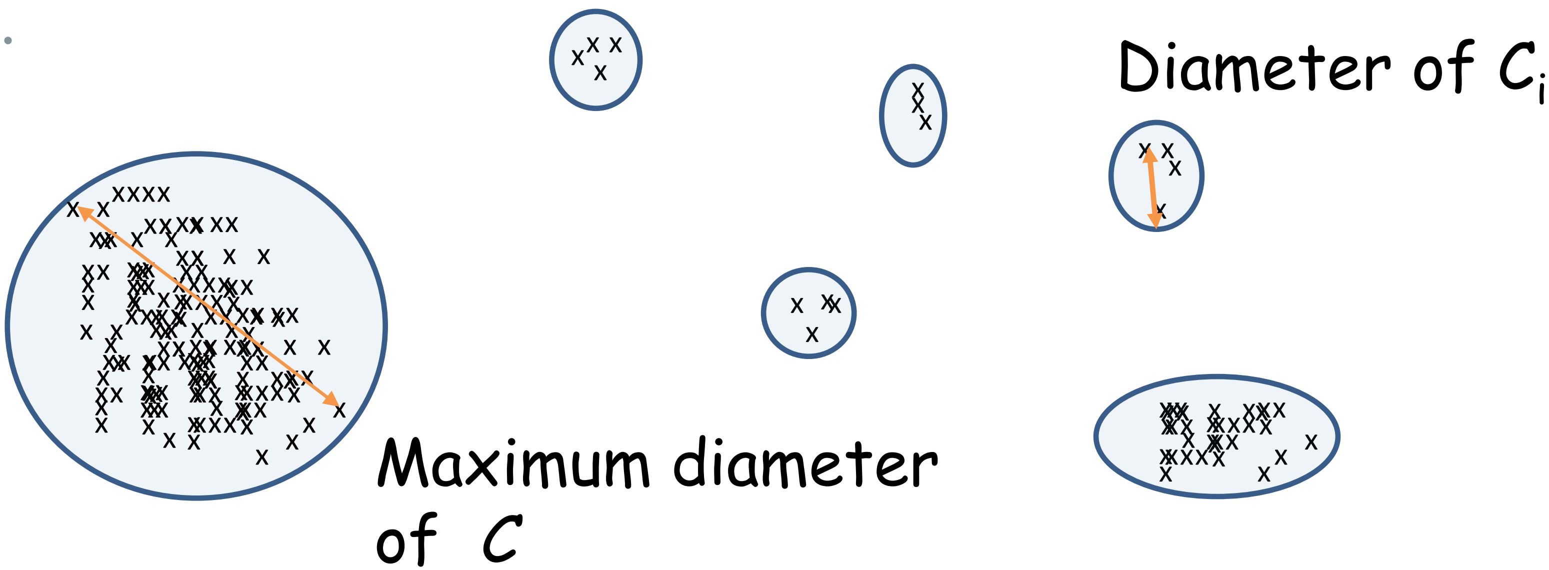
For a cluster C_i

- Diameter(C_i): maximum distance between points in C_i

For a clustering $C=(C_1,\ldots,C_k)$

- Diameter(C): maximum diameter among the clusters C_i

Cohesion Criteria



clustering C with k=6 clusters

Metric Spaces

We assume the points lie in a metric space (mostly)

- Triangle inequality: for every a,b and c

  dist(a,b)+dist(b,c) ≤ dist(a,c)

- Many relevant distances are metrics
  - Euclidean distance
  - Manhattan distance

Diameter of Complete-Link

Theorem [Arutyunova et. al 23] For every instance, the k-clustering $\mathcal{C}$ built by complete-link satisfies

$$\text{diameter}(\mathcal{C}) \leq k^{1.59}\ OPT_{DIAM}$$

Theorem [Arutyunova et. al 23] There exists an instance for which the k-clustering $\mathcal{C}$ built by complete-link satisfies

$$\text{diameter}(\mathcal{C}) \geq k\ OPT_{DIAM}$$

OPT_{DIAM}: diameter of the k-clustering with minimum diameter

Diameter of Single-Linkage

Theorem [Arutyunova et. al 23] For every instance, the k-clustering S built by single-link satisfies

$$\text{diameter}(S) \leq (2k-2)\, OPT_{DIAM}$$

Theorem [Dasgupta 05] There exists an instance for which the k-clustering S built by single-link satisfies

$$\text{diameter}(S) \geq k\, OPT_{DIAM}$$

Takeaway

- Single-link outperforms complete-link in term of cohesion (diameter)
  - Not expected since complete-link greedily minimizes the diameter
  - Single-link suffers from chaining effect

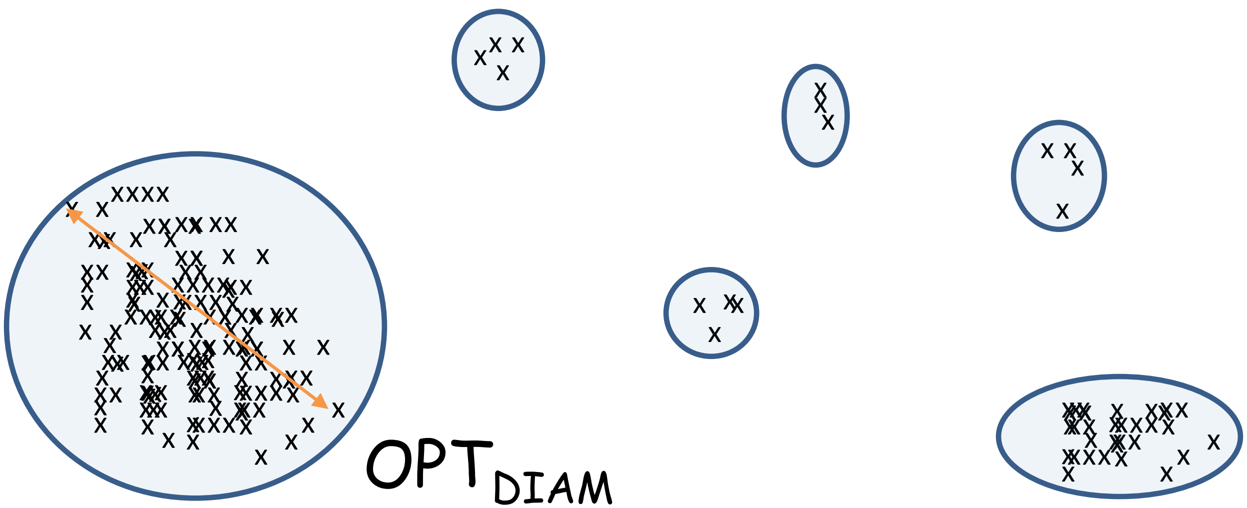
Our Results

- Average diameter of k-clustering $C=(C_1,\ldots,C_k)$ is

$$\frac{1}{k}\sum_{i=1}^{k} diameter(C_i)$$

- OPT_{AVG} : average diameter of the k-clustering with minimum average diameter

Our Results



$OPT_{AVG} \leq OPT_{DIAM}$

OPT_{AVG} may be up to k times smaller than OPT_{DIAM}

Diameter of Complete-Link

Theorem [Dasgupta & L. 24] For every instance the k-clustering $\mathcal{C}$ built by complete-link satisfies

(i) $\text{diameter}(\mathcal{C}) \leq k^{1.59}\, OPT_{AV}$

$\leq k^{1.59}\, OPT_{DIAM}$

(ii) $\text{diameter}(\mathcal{C}) \leq k^{1.30}\, OPT_{DIAM}$

$\leq k^{1.59}\, OPT_{DIAM}$

Diameter of Single-link

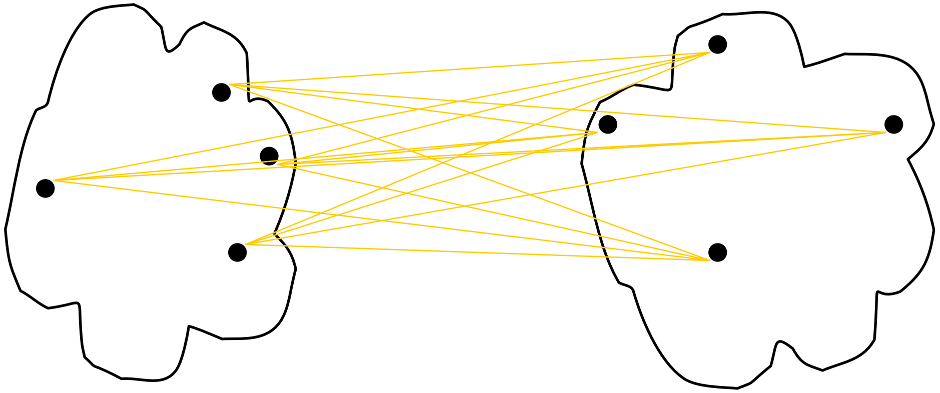
Theorem [Dasgupta & L. 24] There exists an instance for which the k-clustering S built by single-link satisfies

$$\text{diameter}(S) \geq k^2 \, OPT_{AVG}$$

Consequence: Separation between complete-link and single-link using OPT_{AVG}

Average Link

- Usually considered one of the most effective linkage methods
- Few theoretical analysis are available



Cohesion of Average-Link

avg(A): average paiwise distance between points in A

Theorem. [Dasgupta and L. 24] Every cluster A in the k-clustering built by average-link satisfies

$$\text{avg}(A) \leq k^{1.59}\, OPT_{AVG}$$

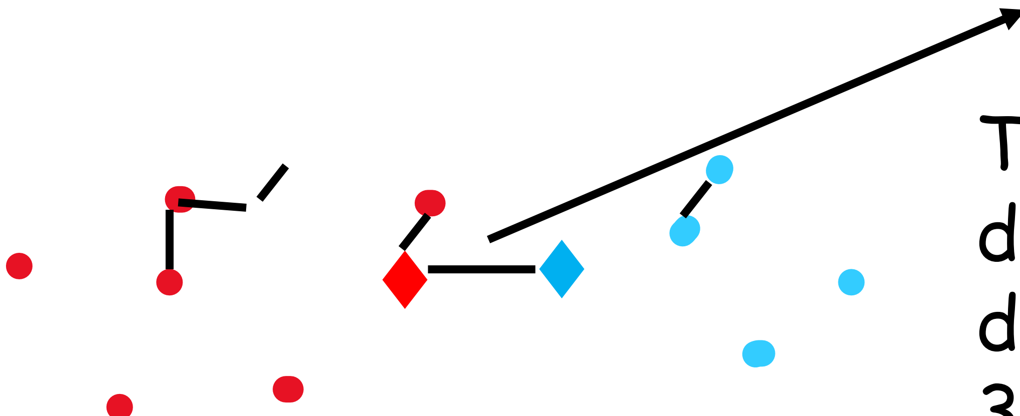
Cohesion of Average Link

Theorem [L. & Batista 24] For every instance the k-clustering A built by average-link satisfies

$$\text{diameter}(A) \leq \min(k, 4 \log n+1)\, k^{1.59}\, OPT_{AV}$$

Theorem [L. & Batista 24] There is an instance I for which the k-clustering A built by average-link satisfies diameter(A)$\geq k OPT_{Diam}$

Proof Strategy

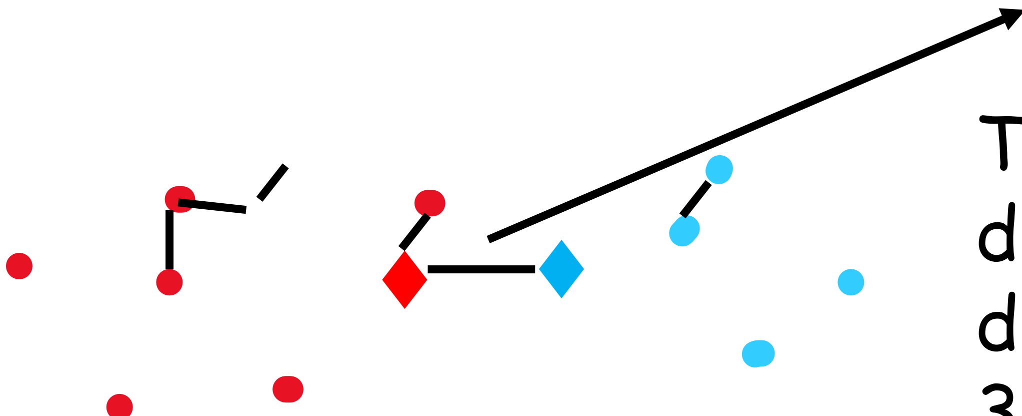


Complete-link rule ⇒
dist(♦, ♦) $\leq OPT_{DIAM}$

Triangle Ineq. ⇒
dist(•, •) $\leq$ dist(•, ♦)+ dist(♦, ♦)+dist(♦, •) $\leq$ $3\ OPT_{DIAM}$

Clustering with optimal diameter for k=2

Proof Strategy



Complete-link rule ⇒
dist(♦, ♦) ≤ OPT_{DIAM}

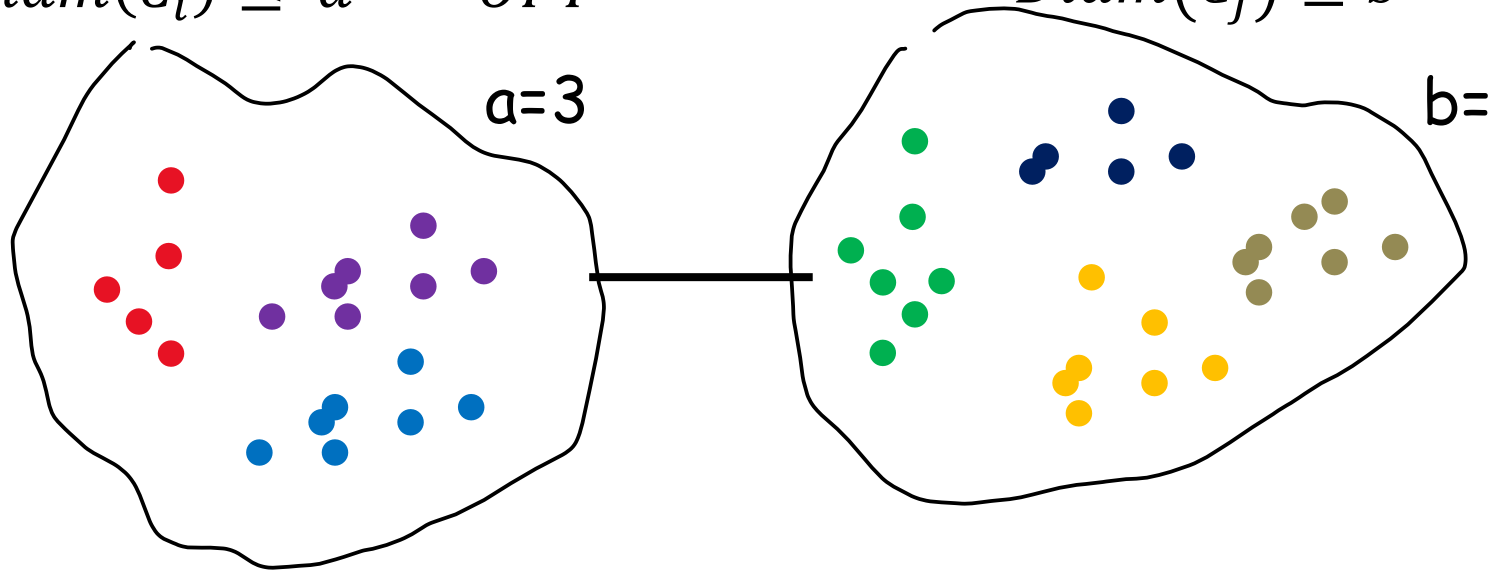
Triangle Ineq. ⇒
dist(•, •) ≤ dist(•, ♦)+
dist(♦, ♦)+dist(♦, •) ≤
3 OPT_{DIAM}

At most 3 times optimal diameter!

k=2 and $3=2^{1.59}$

Proof Strategy

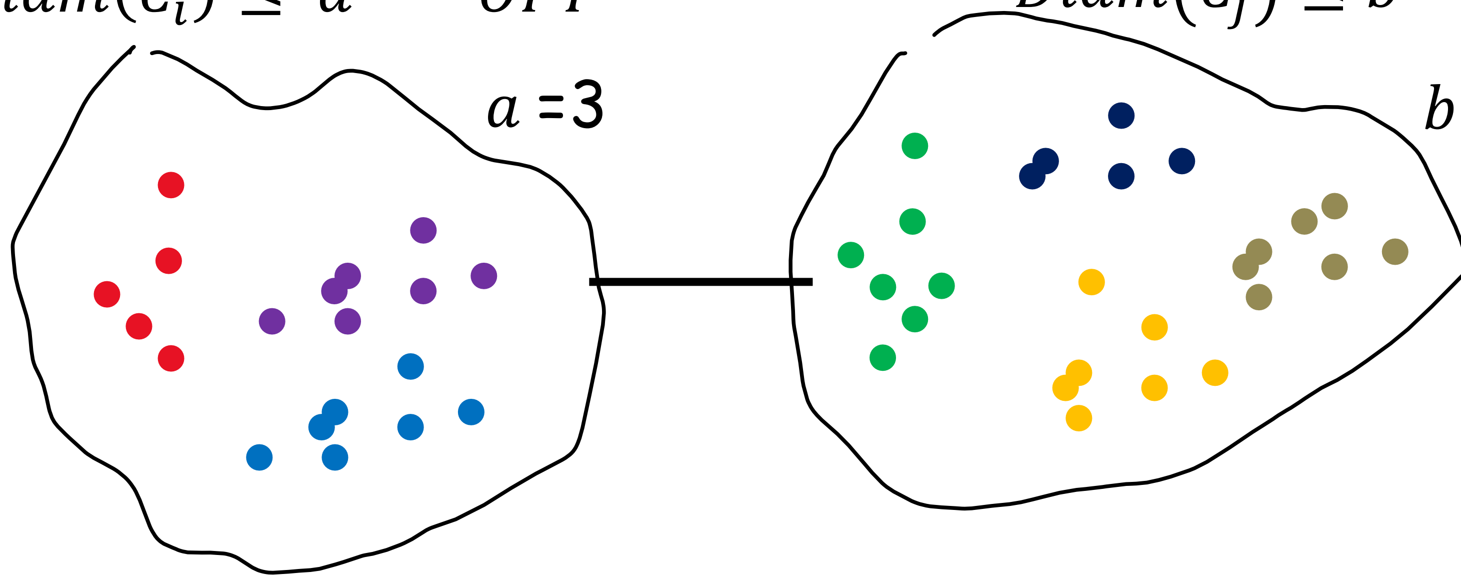
$Diam(C_i) \leq a^{\log 3}\, OPT$ $\qquad$ $Diam(C_j) \leq b^{\log 3}\, OPT$



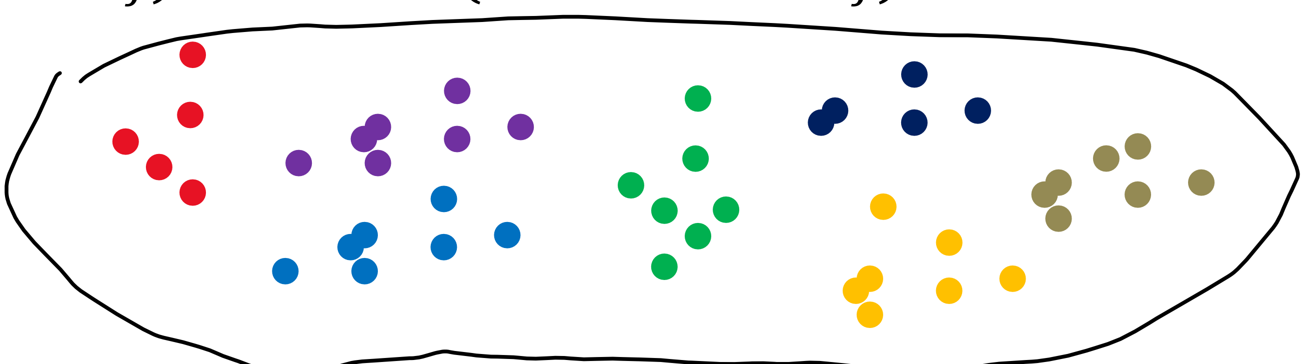
Points of same color are toghether in the optimal clustering

Proof Strategy

$Diam(C_i) \leq a^{\log 3}\, OPT$ $\quad$ $Diam(C_j) \leq b^{\log 3}\, OPT$



$$Diam(C_i \cup C_j) \leq 2Diam(C_i) + Diam(C_j) \leq (a+b)^{\log 3}\, OPT$$



Lower Bounds

- There is an $\Omega(k)$ lower bound for all methods
  - Lower bounds for complete-link and average-link use 2^k data points
  - It does not imply an $\Omega(n)$ lower bound

Takeaway

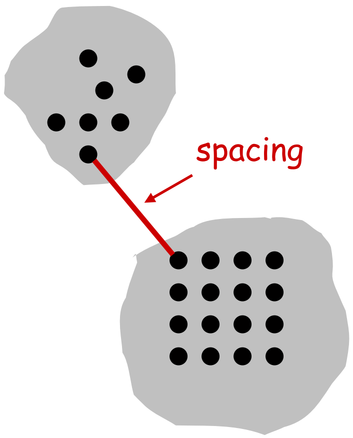
- Linkage methods work well for small k and bad for large k
  - Not expected, for small k the error of the greedy choices should lead to bad situations
  - For k=n-1 complete-link is optimal

Open Questions

- Better understing of the performance of complete-link and average-link for large k
- Do they obtain a logarithmic approximation to the diameter?

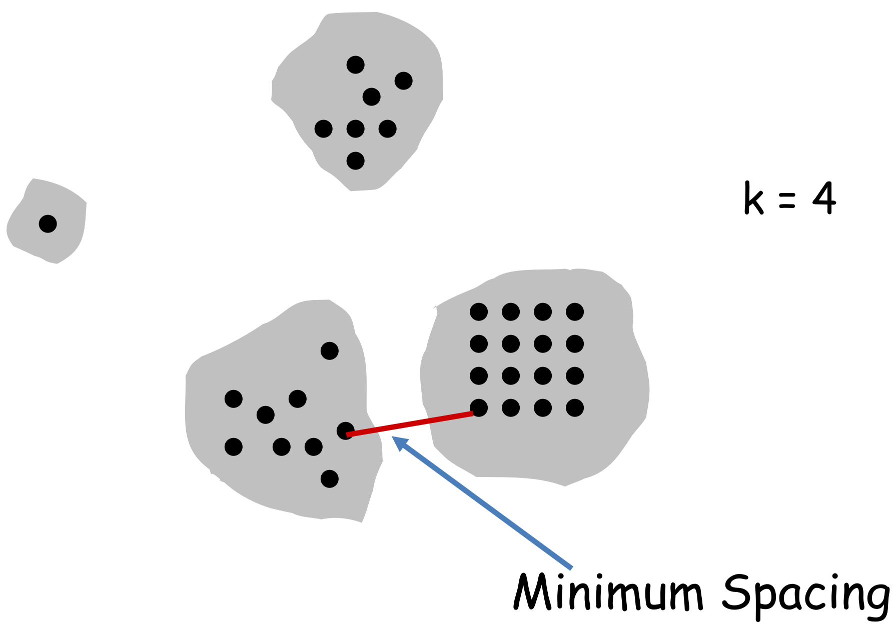
Separability: Spacing

Definition. The spacing of a pair of clusters A and B is the minimum distance between a point in A and a point in B



Separability: Minimum Spacing

Definition. The minimum spacing of a clustering is the spacing of the pair of clusters with mininum spacing

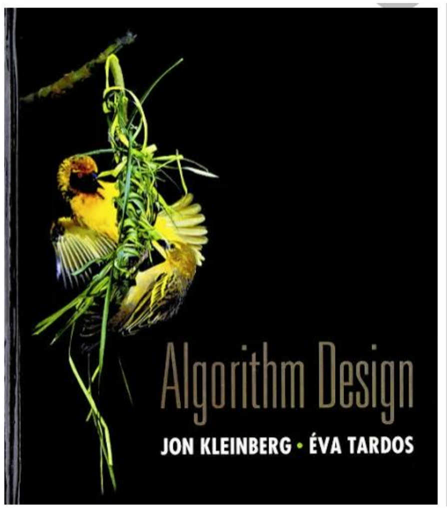


Separability: Max Minimum Spacing

Theorem. [Max-Min Spacing]

For all k, single-Link builds a k-clustering with **maximum** minimum spacing

Proof. Exchange argument



Max Minimum Spacing

Observation. The minimum spacing does not characterize well the behaviour of Single-Link.

Not built by Single-Link

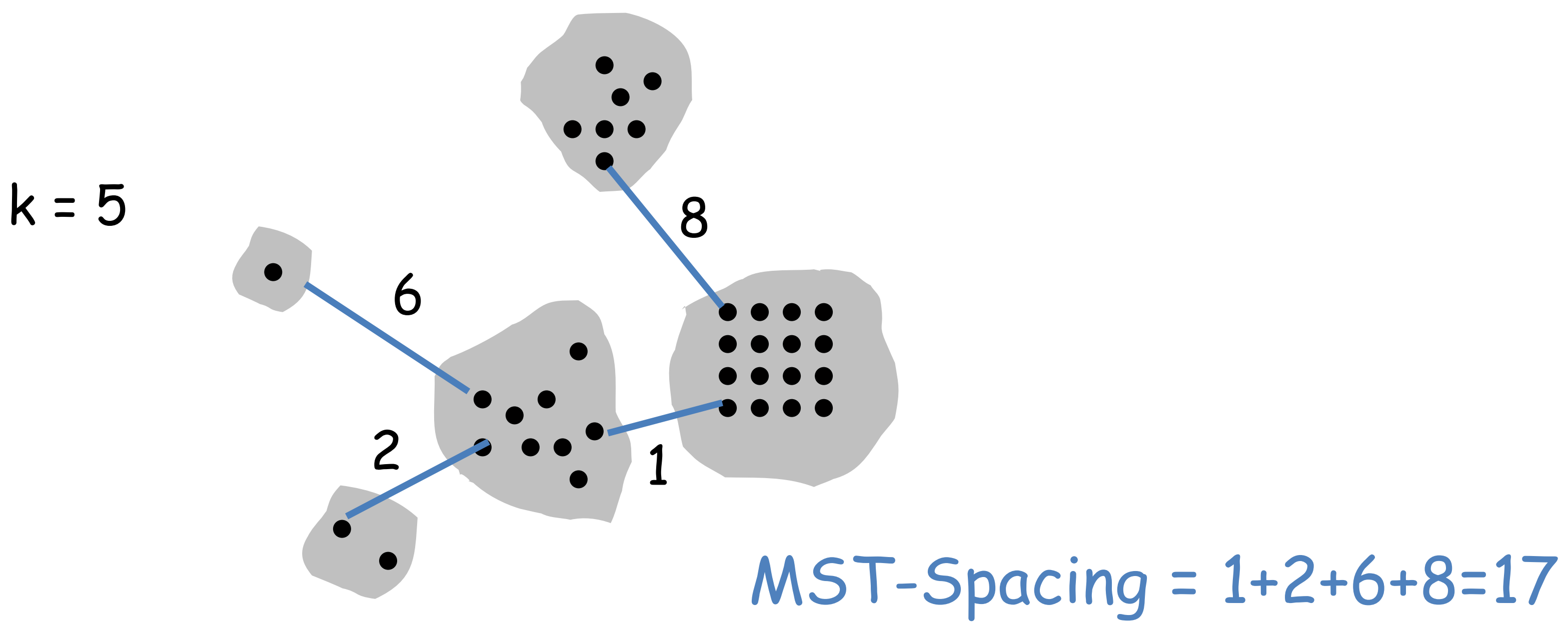
Built by Single-Link

Both examples maximize minimum spacing (k=3):

Mininum Spanning Tree Spacing

Def. MST-Spacing of Clustering C

- Each cluster of C is a node
- cost(u,v): spacing between u and v
- MST-Spacing: cost of the MST



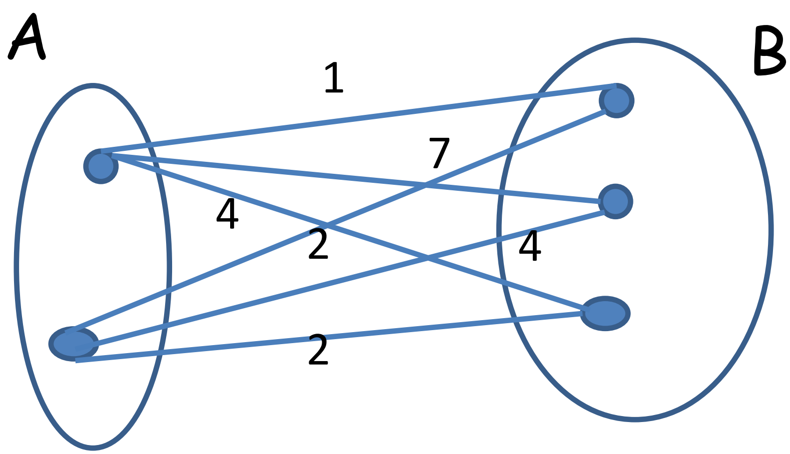
Mininum Spanning Tree Spacing

Theorem [L. & Murtinho 23] Single-link maximizes the MST-Spacing

Theorem [L. & Murtinho 23] If a clustering maximizes the MST-Spacing then it **also** maximizes the Minimum Spacing.

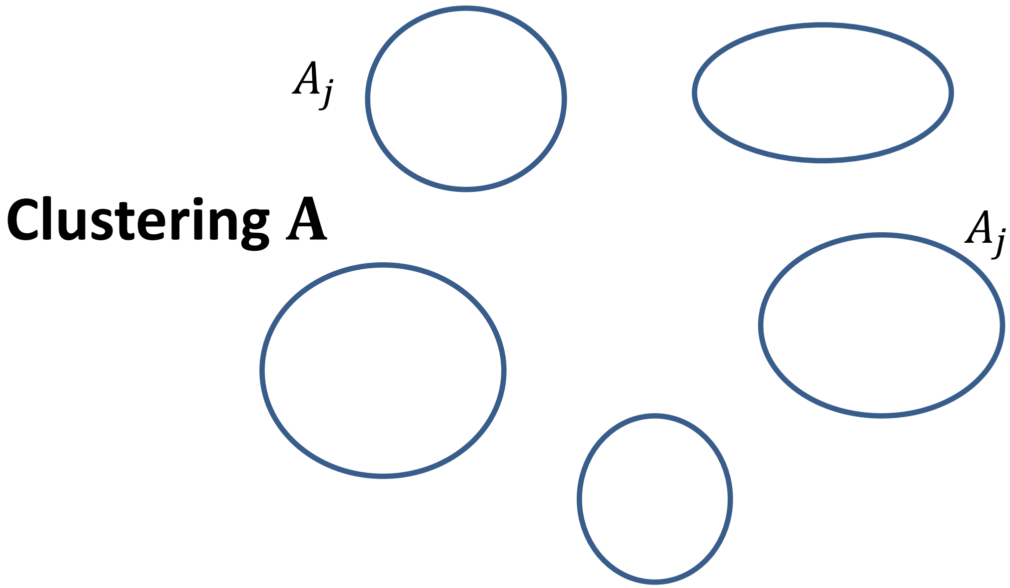
Consequence. MST-Spacing is more relevant than Minimum Spacing in terms of optimization

Separability: Average spacing



$$avg(A, B) = \frac{1}{|A||B|} \sum_{a,b \in A \times B} dist(a, b)$$

Separability: Average spacing



$$\mathrm{sep}_{\mathrm{av}}(\mathbf{A}) := \frac{\sum avg(A_i, A_j)}{k(k-1)/2}$$

Separation of Average Link

Theorem [L. & Batista 24] For every instance the k-clustering $\mathbf{A}=(A_1,..,A_k)$ built by average-link satisfies

$$\mathrm{sep}_{\mathrm{av}}(\mathbf{A}) := \frac{\sum avg(A_i, A_j)}{k(k-1)/2} \geq \frac{OPT_{av}}{k + \ln n}$$

and the bound is nearly tight

- There are instances for which the clustering **C** and **S** built by complete-link and single-link are $(k+\sqrt{n})$ from the optimal [exponential gap]

Separation of Average Link

Proof:

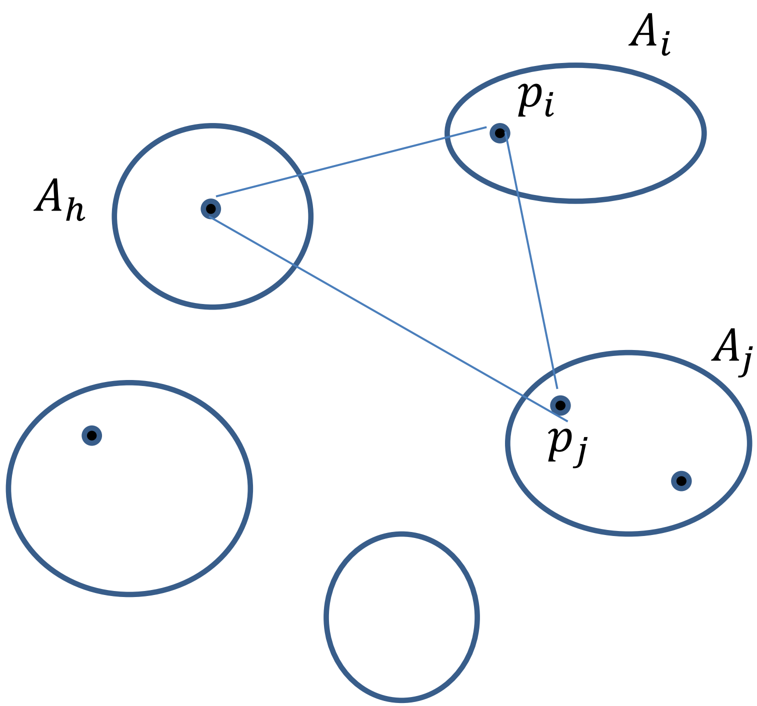
- There is a set of k points P={p_1,..., p_k} that that satisfy

  average distance in P$\geq OPT_{av}$

- It is enough relate average distance in P with $sep_{av}(\boldsymbol{A})$

Separation of Average Link

Proof:



$$dist(p_i, p_j) \leq$$
$$avg(A_j, A_h) + avg(A_i, A_h) +$$
$$avg(p_i, A_i) + avg(p_j, A_j) \leq$$
$$avg(A_j, A_h) + av(A_i, A_h)$$
$$+ \ln(n)\, sep_{av}(\boldsymbol{A})$$

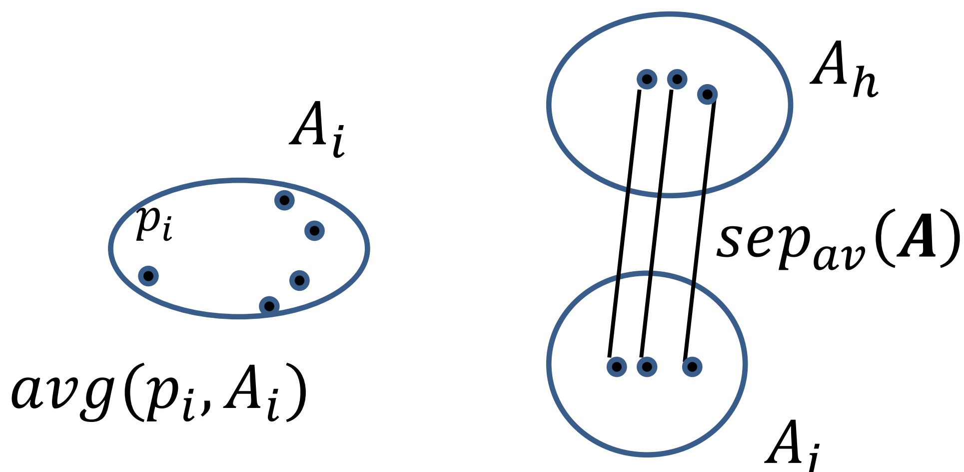
Result is established averaging over all p_i, p_j *and* A_h

Separation of Average Link

Key Lemma: Let **A**=(A_1,..., A_k) be a cluster built by average-link. Let $p_i \in A_i$. Then,

$$avg(p_i, A_i) \leq ln\ (|A_i|) sep_{av}(\boldsymbol{A})\ (*)$$

Proof Idea



Separation of Average Link

Key Lemma: Let **A**=(A_1,..., A_k) be a cluster built by average-link. Let $p_i \in A_i$. Then,

$$avg(p_i, A_i) \leq ln\,(|A_i|) sep_{av}(\boldsymbol{A})\ (*)$$

Proof Idea

- Pick A_j and A_h such that $\mathrm{avg}(\mathrm{A_j}, \mathrm{A_h}) \leq sep_{av}(\boldsymbol{A})$
- If the inequality (*) does not hold, then at some step average-link would have merged a subset of A_j with a subset of A_h

Cohesion/Separation of Avg Link

Theorem [L. & Batista 24] For every instance (not necesarily in a metric-space) the k-clustering $A=(A_1,\dots,A_k)$ built by average-link satisfies

$$\frac{max\{avg(A_1),\dots,avg(A_k)\}}{\min_{i \neq j} avg(A_i, A_j)} \leq 1$$

- There are instances for which complete-link and single-link have value $\geq$ n and $\geq \sqrt{n}$ for the above criterion

Cohesion/Separation of Avg Link

Theorem [L. & Batista 24] For every instance in a metric space, the k-clustering $A=(A_1,\ldots,A_k)$ built by average-link satisfies

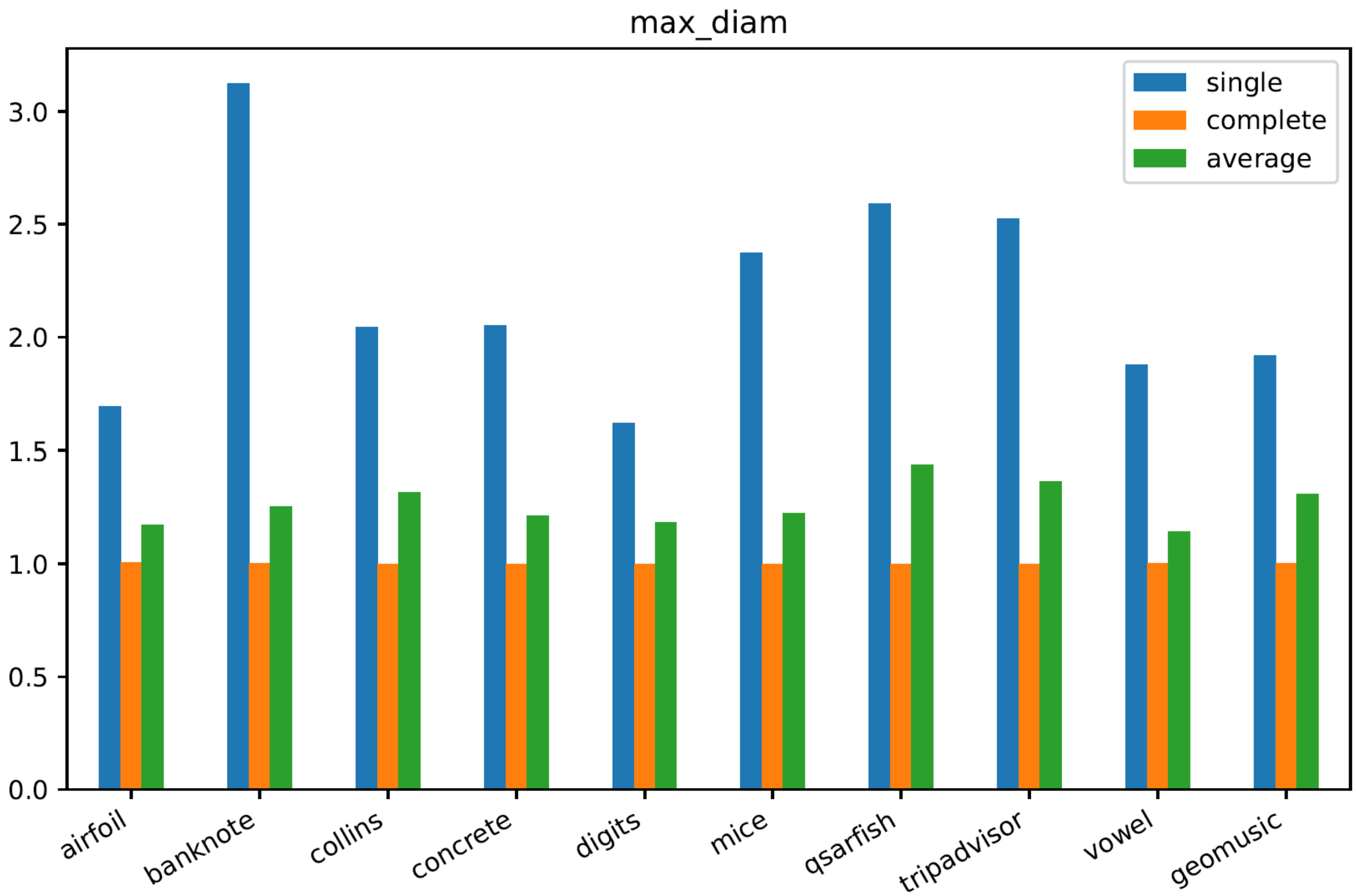
$$\frac{max\{diam(A_1),\ldots,diam\,(A_k)\}}{\min_{i\neq j} avg(A_i,A_j)} \leq \log n$$

- There are instances for which complete-link and single-link have value $\geq$ n and $\geq \sqrt{n}$ for the above criterion

Experiments

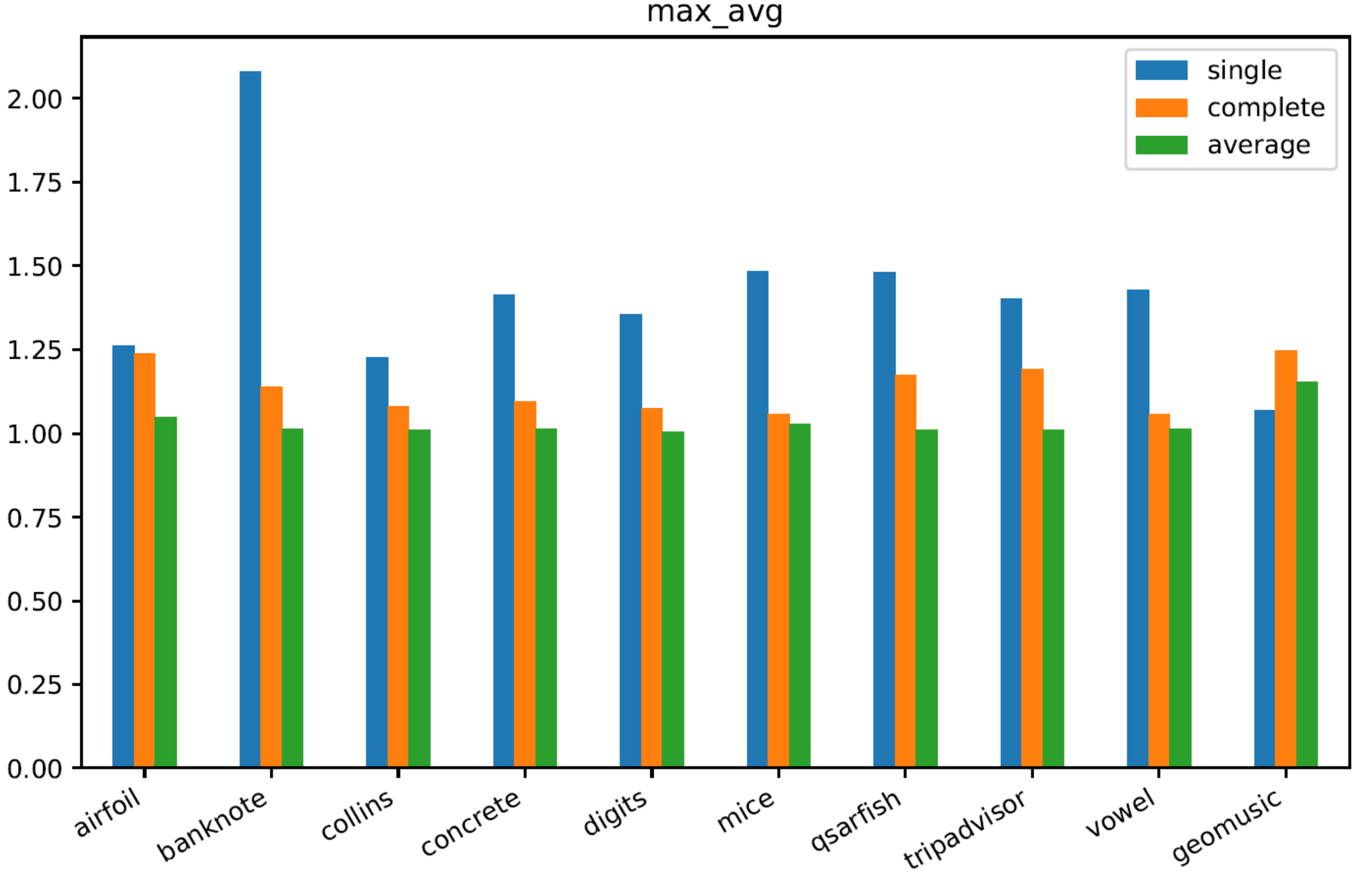
Dataset	n	d	Source
Airfoil	1501	5	Brooks and Marcolini [2014]
Banknote	1371	5	Lohweg [2013]
Collins	1000	19	OpenML
Concrete	1028	8	Yeh [2007]
Digits	1797	64	Alpaydin [1998]
Geographical Music	1057	116	Zhou [2014]
Mice	552	77	Higuera and Cios [2015]
Qsarfish	906	10	Ballabio and Todeschini [2019]
Tripdvisor	979	10	Renjith [2018]
Vowel	990	10	UCI

Experiments: cohesion



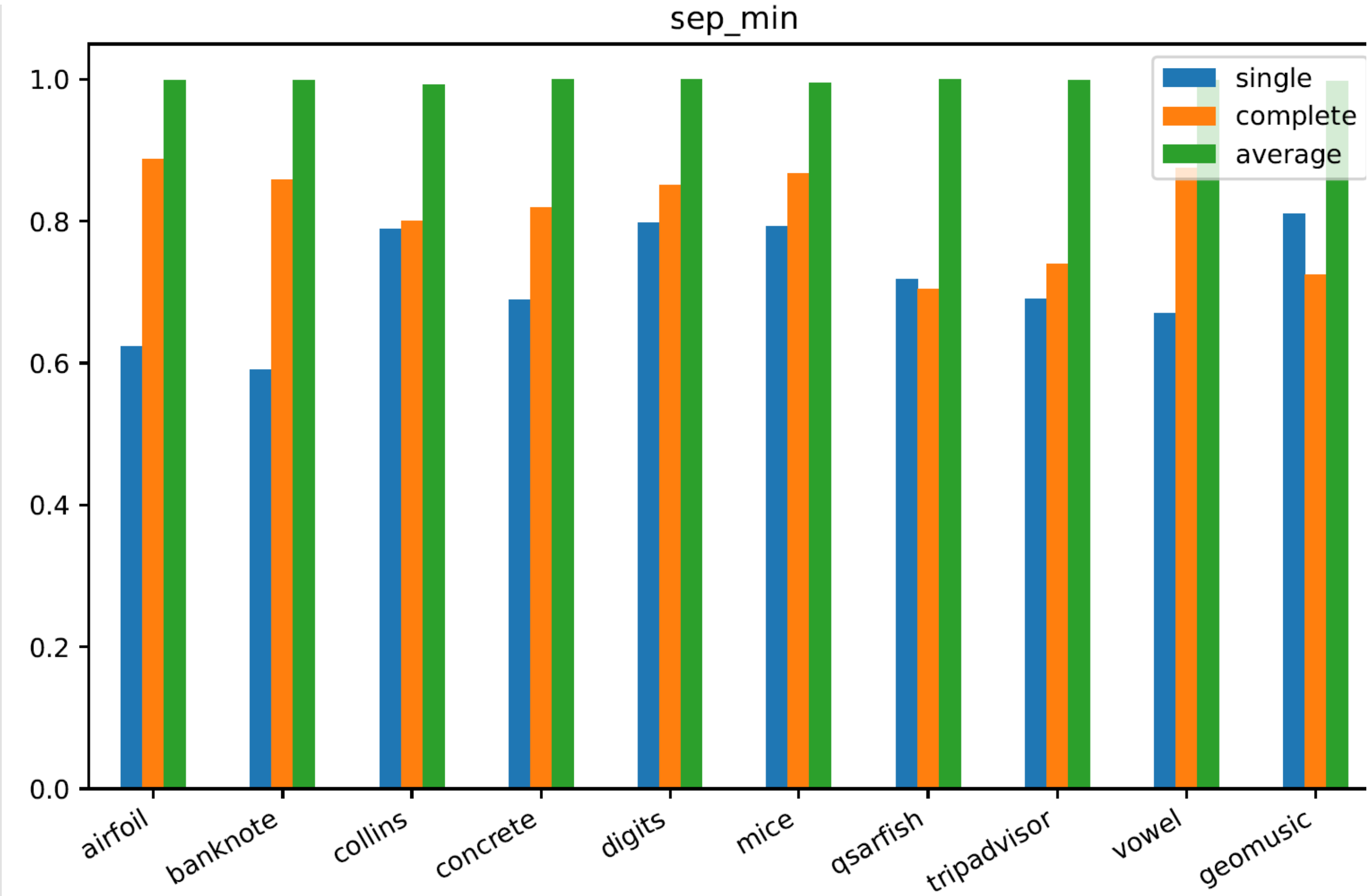
$\min_i diam(A_i)$: The lower the better

Experiments: cohesion



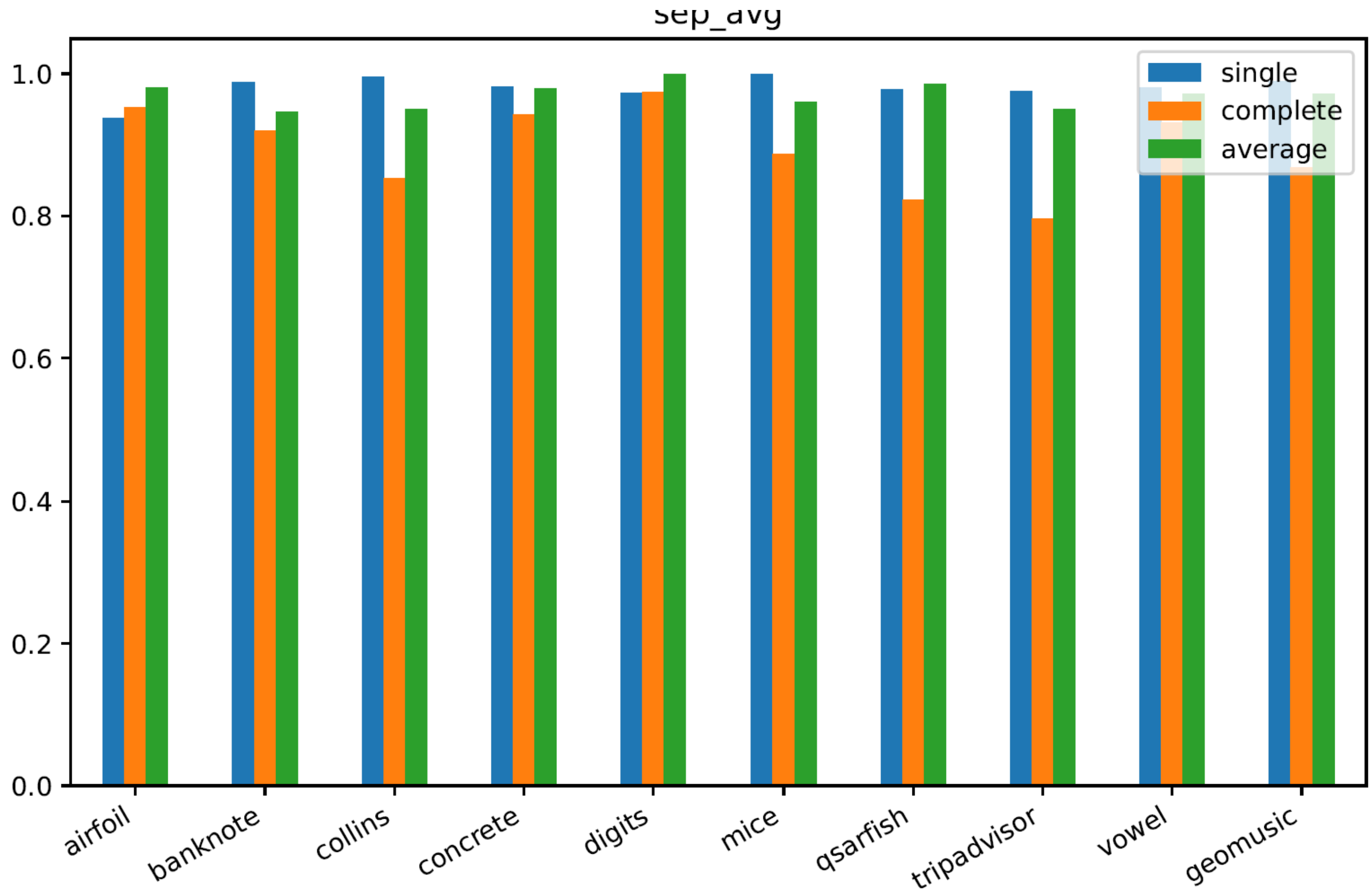
$\min_i avg(A_i)$: The lower the better

Experiments: separability



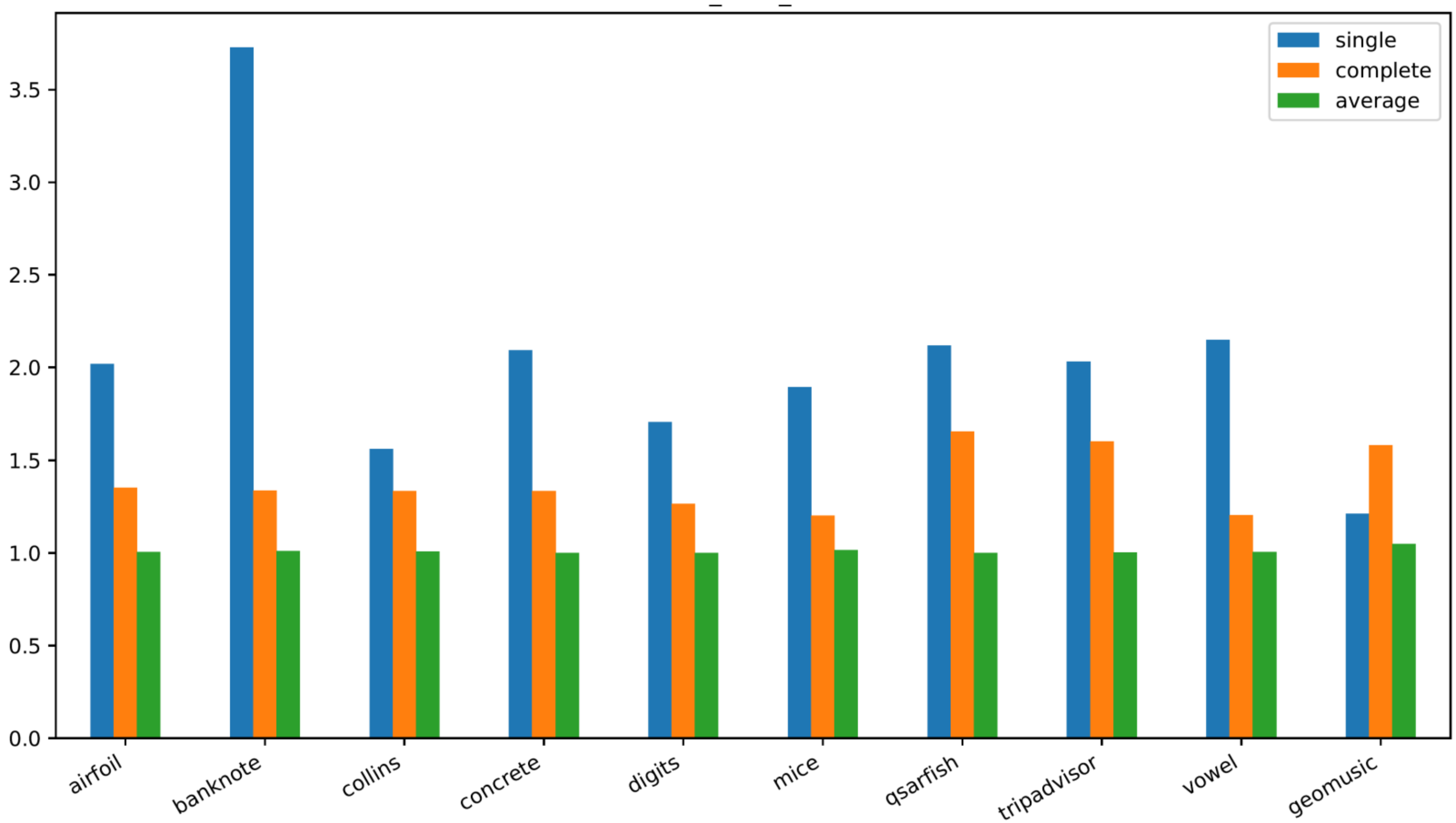
$\min_{i \neq j} avg(A_i, A_j)$: The higher the better

Experiments: separability



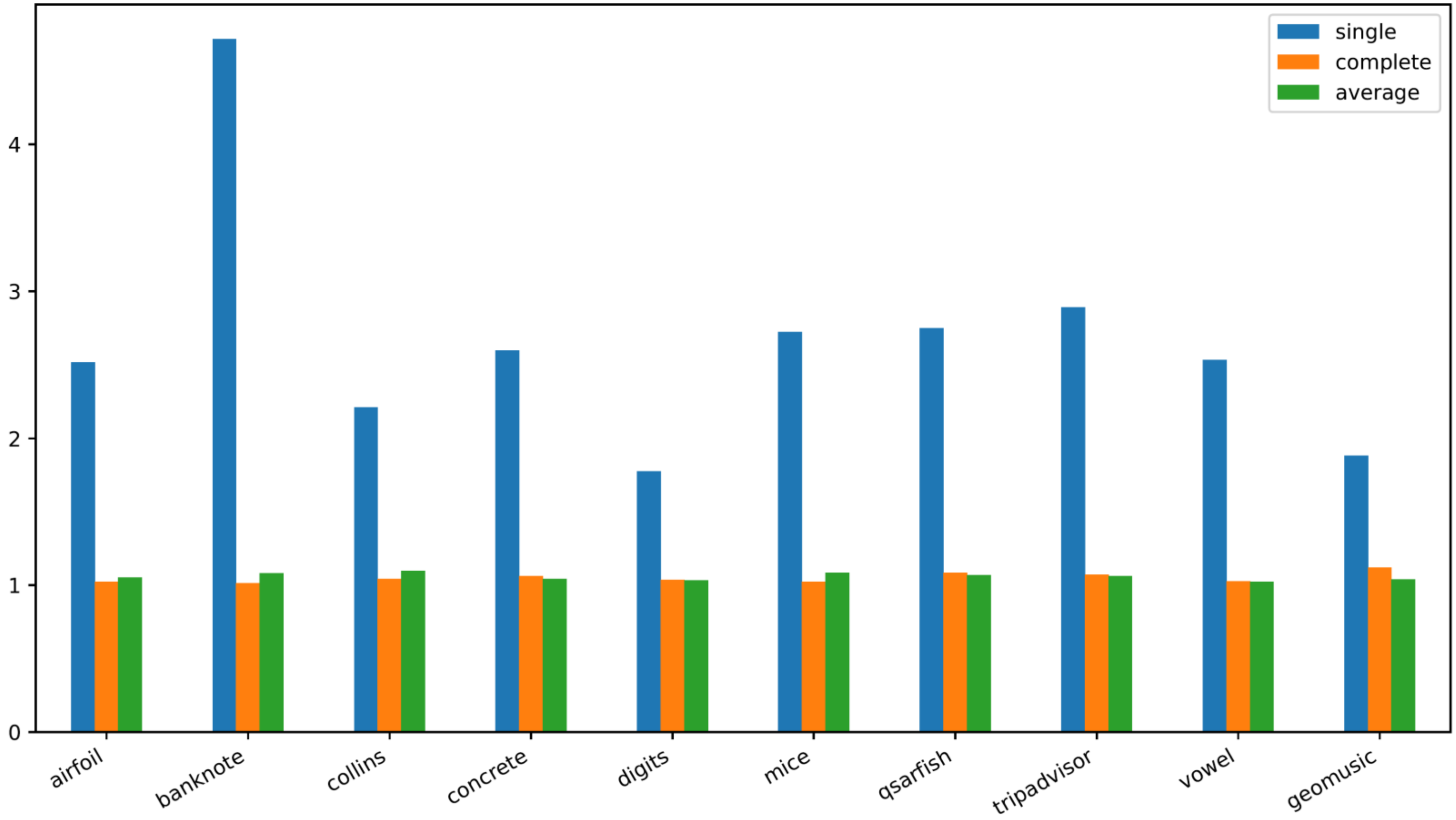
$\sum_{i \neq j} avg(A_i, A_j)$: The higher the better

Experiments: combined



$$\frac{\max_i avg(A_i)}{\min_{i \neq j} avg(A_i, A_j)}$$: The lower the better

Experiments: combined



$$\frac{\max_i diam\,(A_i)}{\min_{i \neq j} avg(A_i, A_j)}$$: The lower the better

Conclusions

- New and improved interpretable bounds for the cohesion and separability of classical linkage methods
- Alignment between theoretical results and those observed in practice

Future Work

- Simple linkage methods with better guarantees
- Results for large k

Thank you