Statistical Foundations of Learning

Debarghya Ghoshdastidar

School of Computation, Information and Technology

Technical University of Munich

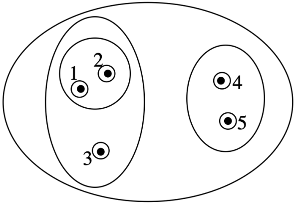
Hierarchical clustering

Outline

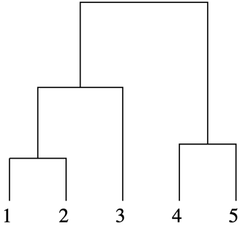
- Introduction to hierarchical clustering (agglomerative/divisive)
- Approximation guarantee of hierarchical clustering based on induced k-clustering
- Hierarchical clustering as an optimisation problem, and its approximation guarantee

Hierarchical clustering

- Aim: Group data $\mathcal{X}$ at different levels of granularity
  - Hierarchy of clusters
  - One can derive k large clusters or many small clusters
- Dendrogram: Binary tree depicting hierarchy of clusters



Clusters at different levels



Dendrogram / Tree

Agglomerative vs divisive clustering

- Agglomerative clustering
  - Initialisation: $|\mathcal{X}|$ number of clusters, each containing a single element
  - Recursion: Merge most similar clusters at each level
  - Example: Average linkage; Single linkage
- Divisive clustering
  - Initialisation: Entire set $\mathcal{X}$ is a single cluster
  - Recursion: Split each cluster into smaller clusters
  - Example: Repeated 2-means; Divisive Analysis (DIANA)

Average and single linkage (based on distances)

- Linkage function (distance) between two clusters C, C':
  - Average linkage: $$d_{avg}(C, C') = \frac{1}{|C| \cdot |C'|} \sum_{x \in C, x' \in C'} d(x, x')$$
  - Single linkage: $$d_{sin}(C, C') = \min_{x \in C, x' \in C'} d(x, x')$$
- Average linkage clustering algorithm:
  1. Start with m singleton clusters, $C_i = \{x_i\}$
  2. Merge clusters C_i, C_j that have smallest linkage $d_{avg}(C_i, C_j)$
  3. Repeat step-2 till all clusters merged

Analysing hierarchical clustering

- $T =$ tree / dendrogram retuned by algorithm $\mathcal{A}$
- How can we measure goodness of output T?
  - There is no inherent optimisation problem / notion of cost
- Approach 1: Cost of induced k-clustering
  - $G_k(\cdot) =$ clustering cost, defined for each k
  - $\mathcal{C}_k = k$-way clustering, obtained from T
  - Is $\mathcal{C}_k$ optimal k-clustering for every k?
- Approach 2: Define new cost / value function for the tree T

Cannot have optimal k-clustering for every k

- $\mathcal{X} = \{1, 2, 3, 4, 5, 6\} \subset \mathbb{R}$
- $G_k(\cdot) = k$-means cost
- Optimal 2-means clusters: $\{1, 2, 3\}$ and $\{4, 5, 6\}$
- Optimal 3-means clusters: $\{1, 2\}$, $\{3, 4\}$ and $\{5, 6\}$
- Above two clusterings cannot be obtained from same tree

Digression: k-center problem

- k-means problem: $\underset{\boldsymbol{\mu}=(\mu_1,\ldots,\mu_k)}{\text{minimise}} \sum_{x\in\mathcal{X}} d(x,\boldsymbol{\mu})^2$ $\qquad d(x,\boldsymbol{\mu}) = \min_{j=1,\ldots,k} \|x-\mu_j\|$

- k-center problem: $\underset{\boldsymbol{\mu}}{\text{minimise}} \max_{x\in\mathcal{X}} d(x,\boldsymbol{\mu})$

- There is a simple 2-approximation algorithm for k-center problem
  - The algorithm is based on farthest first traversal
  - If $P \neq NP$, then there is no poly-time algorithm for k-center problem with approximation ratio < 2

Farthest first traversal

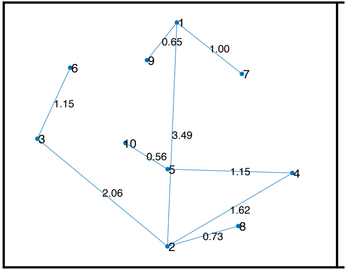
- Input: Data $\mathcal{X} = \{x_1, \ldots, x_m\}$ and distance $d(\cdot, \cdot)$

- Relabel points and draw spanning tree:

  - Denote farthest two points as $1, 2$

  - For $i = 3, \ldots, m$

    — Use i to denote point in $\mathcal{X}$ farthest from $\{1, \ldots, i-1\}$

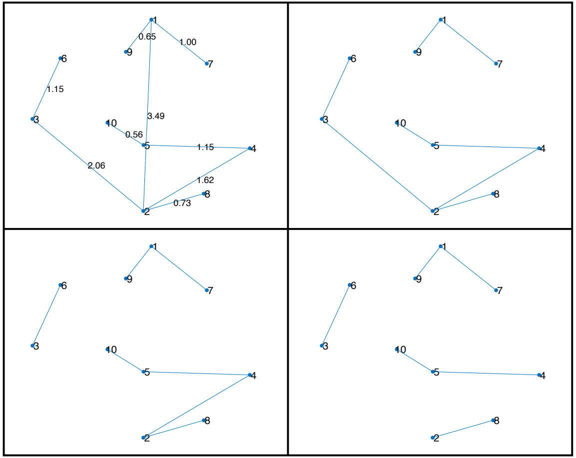


- k-center solution: Choose relabeled points $\{1, \ldots, k\}$ as the k centers

(prove this gives a 2-factor approx)

Divisive algorithm based on farthest first traversal

- Add edge $(1, 2)$
- For $i = 3, \ldots, m$: Add edge (i, j) where $j \in \{1, \ldots, i-1\}$ is closest to i
- Root of T:
  — Entire set $\mathcal{X}$
- Recursions: At each stage,
  — Delete longest edge
  — This splits a cluster (subgraph split into two connected components)



Constant factor approximation of induced k clusterings

Theorem Hier.1 (O(1)-approximation algorithm for cost-induced clustering)

Consider k-center clustering cost

$$G(\boldsymbol{\mu}) = \max_{x \in \mathcal{X}} d(x, \boldsymbol{\mu}) \qquad \ldots \boldsymbol{\mu} = \text{ set of } k \text{ centers}$$

- $G_k(T) =$ *cost of k-center clustering obtained from hierarchical tree T*
- $G_{opt,k} =$ *optimal k-center clustering cost for $\mathcal{X}$*

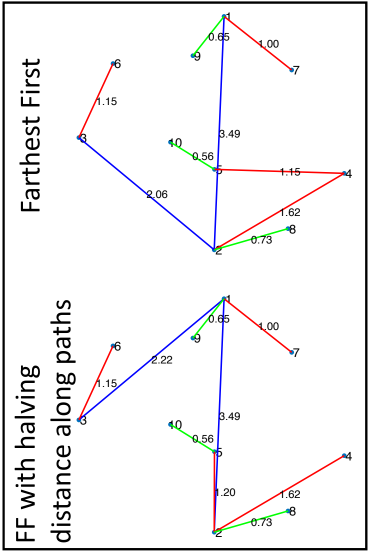
There is a divisive algorithm such that for any $\mathcal{X}$ and every k

$$G_k(T) \quad \leq \quad 8 \cdot G_{opt,k}$$

Proof skipped, but we will see the algorithm

The algorithm

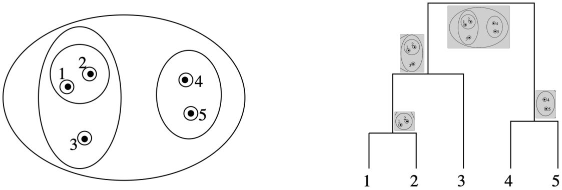
- Connecting i to its closest point in $\{1, \ldots, i-1\}$ can lead to long chains (bad for induced k-center cost)
- Idea: Any path from root must have edge lengths of geometrically decreasing length
- Modified algorithm (assuming points relabelled):
  - Let $R_i = \min\limits_{j \in \{1,\ldots,i-1\}} d(x_i, x_j)$
  - Let $S_0 = \{1\}$ and $S_j = \{i : R_2/2^j < R_i \leq R_2/2^{j-1}\}$
  - Connect $i \in S_j$ to its closest point in $S_0, \ldots, S_{j-1}$ ... call this neighbour $\pi(i)$



Outline

- Introduction to hierarchical clustering (agglomerative/divisive)
- Approximation guarantee of hierarchical clustering based on induced k-clustering
- Hierarchical clustering as an optimisation problem, and its approximation guarantee

Formulating hierarchical clustering as optimisation problem



- N = node in tree = corresponding cluster = sub-tree rooted at N
  - N_1, N_2 = children of N (or corresponding clusters)
  - Distance between two nodes = sum of pairwise distances between elements

$$d(N_1, N_2) = \sum_{x \in N_1} \sum_{x' \in N_2} d(x, x')$$

Value of dendrogram (Cohen-Addad et al, *Journal of the ACM*, 2019)

- Notation: Let points be indexed as $x_1, \ldots, x_n$

- Value function for T

$$\begin{aligned} \text{value}(T) &= \sum_{N \in T} d(N_1, N_2) \cdot |N| \\ &= \sum_{i<j} d(x_i, x_j) \cdot |\text{lca}(x_i, x_j)| \end{aligned}$$

- $\text{lca}(x, x') =$ smallest cluster / node containing both x, x' (least common ancestor)

- High $\text{value}(T)$ if
  - whenever $d(x, x')$ high, merge x, x' later ... both $d(x, x')$ and $|\text{lca}(x, x')|$ large
  - hence, merge closer x, x' earlier

Hierarchical clustering as optimisation

- Formal hierarchical clustering problem (version 1)

$$\underset{T}{\text{maximise}}\ \text{value}(T)$$

  - Maximisation over all binary trees T

Theorem Hier.2 (Approximation guarantee for average linkage)

If $\widehat{T}$ = tree obtained from average linkage, then

$$\mathit{value}(\widehat{T}) \quad \geq \quad \frac{1}{2} \cdot \underbrace{\max_{T} \ \mathit{value}(T)}_{\textit{optimal value}}$$

Proof: Recap average linkage based on distances

- Average linkage between two clusters C, C'

$$d_{avg}(C, C') = \frac{1}{|C| \cdot |C'|} \sum_{x \in C, x' \in C'} d(x, x')$$

1. Start with m singleton clusters, $C_i = \{x_i\}$
2. Merge clusters C_i, C_j that have smallest distance $d_{avg}(C_i, C_j)$
3. Repeat step-2 till all clusters merged

Proof: Warm up

- Exercise:

  - For any tree T, $value(T) \leq m \cdot \sum_{i<j} d(x_i, x_j)$

  - $\sum_{N \in T} d(N_1, N_2) = \sum_{i<j} d(x_i, x_j)$ (hint: every x_i, x_j is merged exactly once in the tree)

    In particular, setting $d(x, x') = 1$, we get $\sum_{N \in T} |N_1| \cdot |N_2| = \binom{m}{2}$

  - (bounds on ratio of sums) If $a_1, \ldots, a_k, b_1, \ldots, b_k > 0$ are positive scalars. Then

$$\min_i \frac{a_i}{b_i} \leq \frac{\sum_i a_i}{\sum_i b_i} \leq \max_i \frac{a_i}{b_i}$$

- Will show: $value(\widehat{T}) \geq \frac{m}{2} \sum_{i<j} d(x_i, x_j)$, which implies $value(\widehat{T}) \geq \frac{1}{2} \cdot value(T_{optimal})$

Proof: By induction

- Notation: Let N be root of tree T. We use term $d(T) := d(N) := \sum_{i,j \in N \,:\, i<j} d(x_i, x_j)$

- Inductive hypothesis:
  If T is constructed by average linkage on m points, then $value(T) \geq \frac{m}{2} d(T)$
  - Holds for base case $m = 2$
  - Assume it holds for all $m' < m$. Need to show it holds for m.

- Observe: If N is root of T with children N_1, N_2 ($T_i =$ sub-tree rooted at N_i)
  - $value(T) \;=\; m \cdot d(N_1, N_2) + value(T_1) + value(T_2)$
  - $d(T) = d(N) \;=\; d(N_1, N_2) + d(N_1) + d(N_2) \;=\; d(N_1, N_2) + d(T_1) + d(T_2)$

Proof: Lower bound for $value(T)$

$$
\begin{aligned}
\text{value}(T) &= m \cdot d(N_1, N_2) + \text{value}(T_1) + \text{value}(T_2) \\
&\geq m \cdot d(N_1, N_2) + \frac{|T_1| \cdot d(T_1)}{2} + \frac{|T_2| \cdot d(T_2)}{2} && \ldots\ T_1, T_2 \text{ have } < m \text{ leaves} \\
&= \frac{m}{2} \cdot d(N_1, N_2) + \frac{m(d(T) - d(T_1) - d(T_2))}{2} \\
&\quad + \frac{|T_1| \cdot d(T_1)}{2} + \frac{|T_2| \cdot d(T_2)}{2} \\
&= \frac{m}{2} \cdot d(T) \\
&\quad + \underbrace{\frac{m \cdot d(N_1, N_2)}{2} - \frac{|T_2| \cdot d(T_1)}{2} - \frac{|T_1| \cdot d(T_2)}{2}}_{\frac{m}{2|N_1| \cdot |N_2|} \underbrace{\left(\frac{d(N_1, N_2)}{|N_1| \cdot |N_2|} - \frac{d(N_1)}{m \cdot |N_1|} - \frac{d(N_2)}{m \cdot |N_2|} \right)}_{\text{suffices to show this} \geq 0}} && \ldots\ m = |T_1| + |T_2|
\end{aligned}
$$

Proof: Using a claim

- Claim $(\star)$:
  Let A be any node in the tree T obtained from average linkage algorithm with children A_1, A_2. Then
  $$\frac{d(A_1,A_2)}{|A_1|\cdot|A_2|} \geq \frac{d(A_1)}{\binom{|A_1|}{2}} \qquad \text{and} \qquad \frac{d(A_1,A_2)}{|A_1|\cdot|A_2|} \geq \frac{d(A_2)}{\binom{|A_2|}{2}}.$$

- Using claim $(\star)$ in previous slide:

$$\begin{aligned}\frac{d(N_1)}{m\cdot|N_1|} + \frac{d(N_2)}{m\cdot|N_2|} &\leq \frac{d(N_1)}{|N_1|(|N_1|-1)} + \frac{d(N_2)}{|N_2|(|N_2|-1)} \\ &\leq \frac{1}{2}\left(\frac{d(N_1)}{\binom{|N_1|}{2}} + \frac{d(N_2)}{\binom{|N_2|}{2}}\right) \quad \leq \frac{d(N_1,N_2)}{|N_1|\cdot|N_2|} \quad \text{proves Theorem}\end{aligned}$$

Proof of Claim $(\star)$

- Let T_1, T_2 be trees rooted at A_1, A_2

- By ratio of sums bound: $$\frac{d(A_1)}{\binom{|A_1|}{2}} = \frac{\sum\limits_{N \in T_1} d(N_1, N_2)}{\sum\limits_{N \in T_1} |N_1| \cdot |N_2|} \leq \max_{N \in T_1} \frac{d(N_1, N_2)}{|N_1| \cdot |N_2|}$$

- Above means that there are $N_1, N_2 \subset A_1$ such that $\dfrac{d(A_1)}{\binom{|A_1|}{2}} \leq \dfrac{d(N_1, N_2)}{|N_1| \cdot |N_2|}$

  - Consider stage where N_1, N_2 were merged

  - Suppose, at that stage A_1, A_2 comprised of clusters $A_1 = \bigcup\limits_{i=1}^{k} N_i$ and $A_2 = \bigcup\limits_{j=1}^{l} M_j$

  - Since N_1, N_2 was merged, its average linkage $\dfrac{d(N_1, N_2)}{|N_1| \cdot |N_2|}$ was lower than other pairs

Proof of Claim ($\star$)

$$\frac{d(A_1)}{\binom{|A_1|}{2}} \leq \frac{d(N_1, N_2)}{|N_1| \cdot |N_2|} \leq \min_{i,j} \frac{d(N_i, M_j)}{|N_i| \cdot |M_j|}$$

$$\leq \frac{\sum_{i,j} d(N_i, M_j)}{\sum_{i,j} |N_i| \cdot |M_j|} \qquad \text{. . . ratio of sums bound}$$

$$= \frac{d(\cup_i N_i, \cup_j M_j)}{\left(\sum_i |N_i|\right)\left(\sum_j |M_j|\right)} \qquad \text{. . . } \textbf{why?}$$

$$= \frac{d(A_1, A_2)}{|A_1| \cdot |A_2|}$$

Bound for A_2 proved similarly

Hierarchical clustering with similarities

- Instead of distance $d(x,x')$, assume we have similarity $w(x,x')$
- Average linkage of clusters C, C'

$$w_{avg}(C,C') = \frac{1}{|C|\cdot|C'|}\sum_{x\in C, x'\in C'} w(x,x')$$

  - Recursion: Merge C, C' with largest linkage $w_{avg}(C,C')$

- How do we incorporate w in an optimisation formulation?

Cost of hierarchical tree (Dasgupta, *STOC*, 2016)

- Dasgupta's cost for T

$$\begin{aligned} \mathrm{cost}(T) &= \sum_{N \in T} w(N_1, N_2) \cdot |N| \qquad \ldots \; w(N_1, N_2) = \sum_{\substack{x \in N_1 \\ x' \in N_2}} w(x, x') \\ &= \sum_{x \neq x'} w(x, x') \cdot |\mathrm{lca}(x, x')| \end{aligned}$$

Hierarchical clustering as cost minimisation

- Formal hierarchical clustering problem (version 2)

$$\underset{T}{\text{minimise}} \ \text{cost}(T)$$

  - Minimisation over all trees T, not only binary trees

- Interesting results:
  - Tree with minimum cost is binary
  - There is divisive algorithm with $\text{cost}(\widehat{T}) \ \leq \ O(\sqrt{\ln m}) \cdot \min_{T} \text{cost}(T)$

- Open problem: Can average linkage be analysed under Dasgupta's cost?