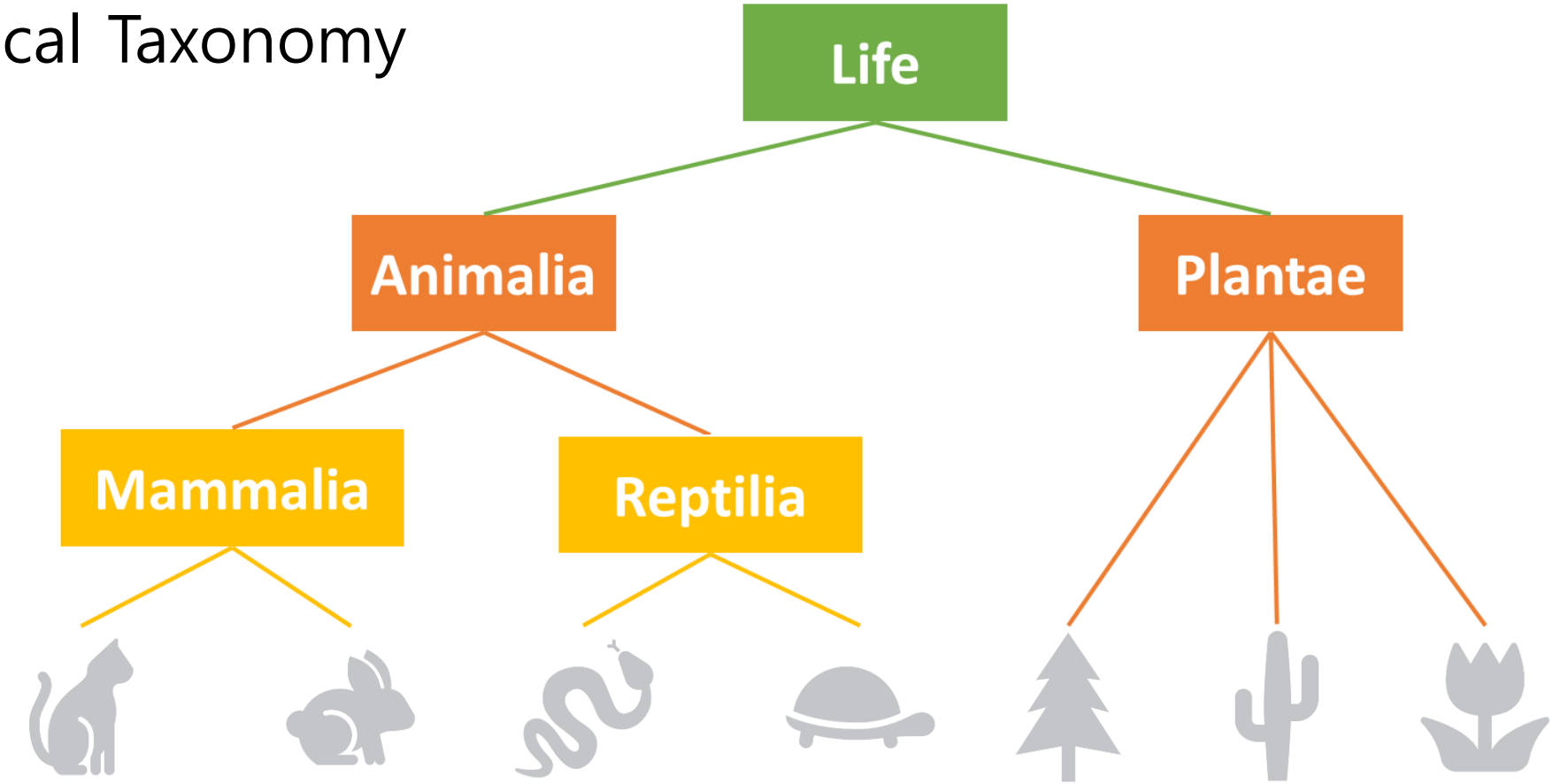
A Framework for Parallelizing Hierarchical Clustering Methods

Silvio Lattanzi, **Thomas Lavastida**, Kefu Lu, Benjamin Moseley

September 19, 2019

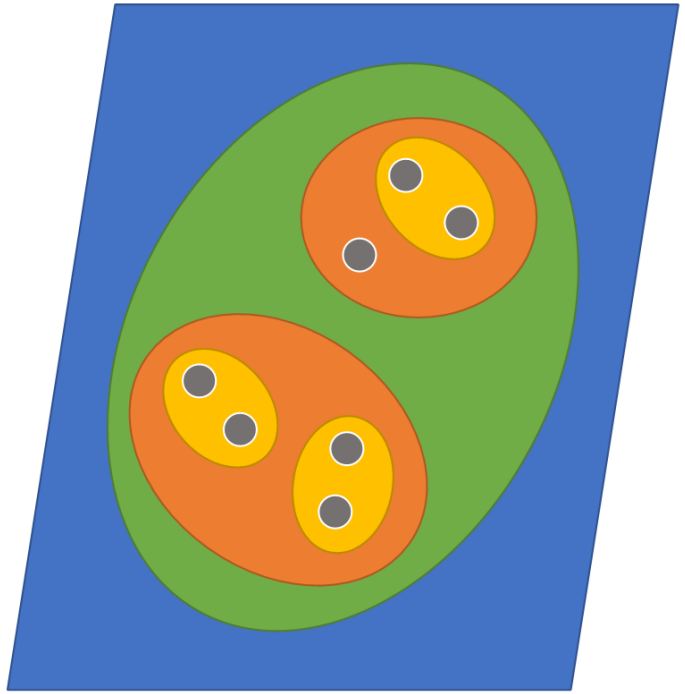
What is Hierarchical Clustering?

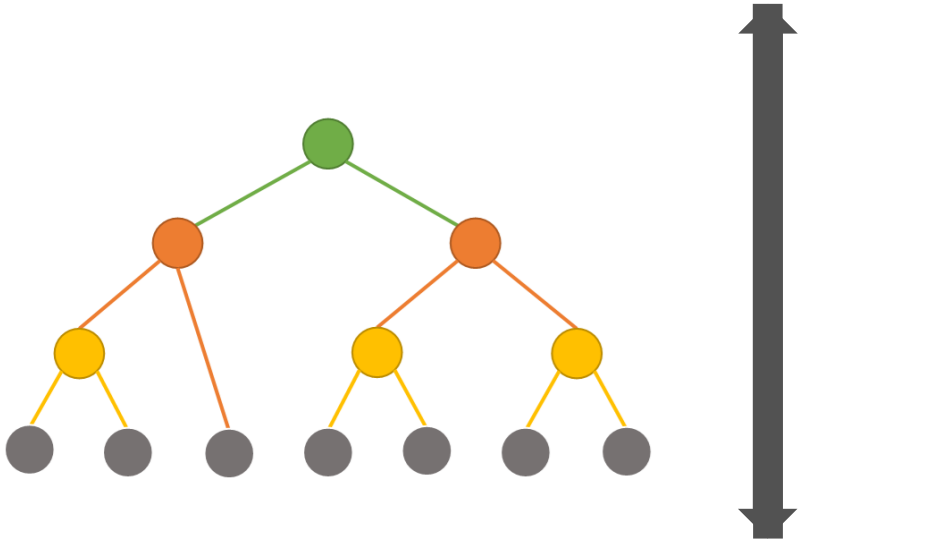
- Biological Taxonomy



What is Hierarchical Clustering?

- Set S of n data points
- Pairwise dissimilarity → Euclidean distance



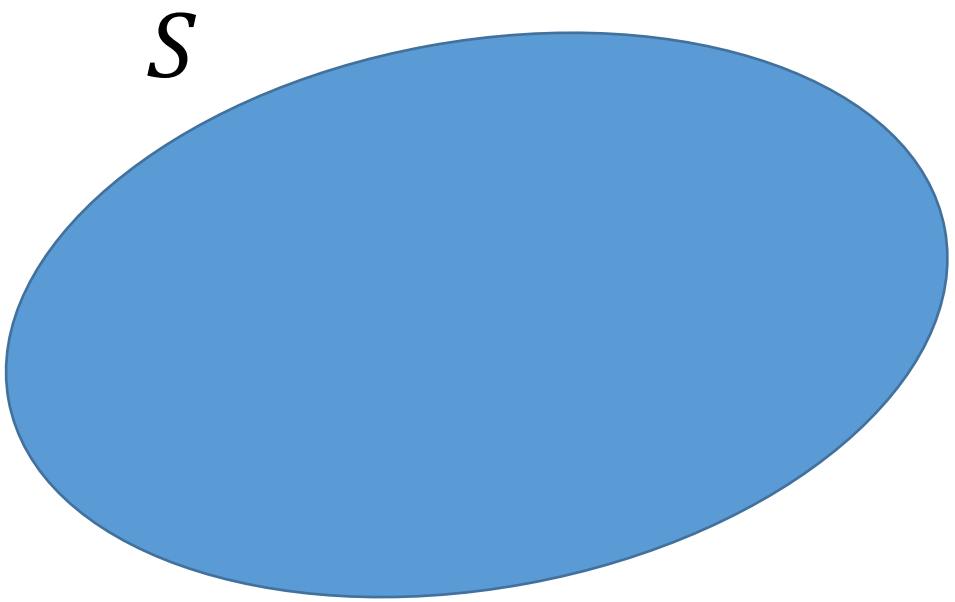


Divisive Methods

- Split using k-means with $k = 2$
- k-means objective: $\sum_{i=1}^{k} \sum_{x \in S_i} d\left(x, \mu(S_i)\right)^2$
  - Centroid: $\mu(S) := \frac{1}{|S|} \sum_{x \in S} x$

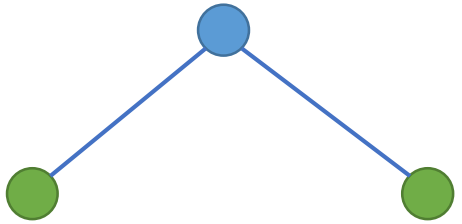
Divisive Methods

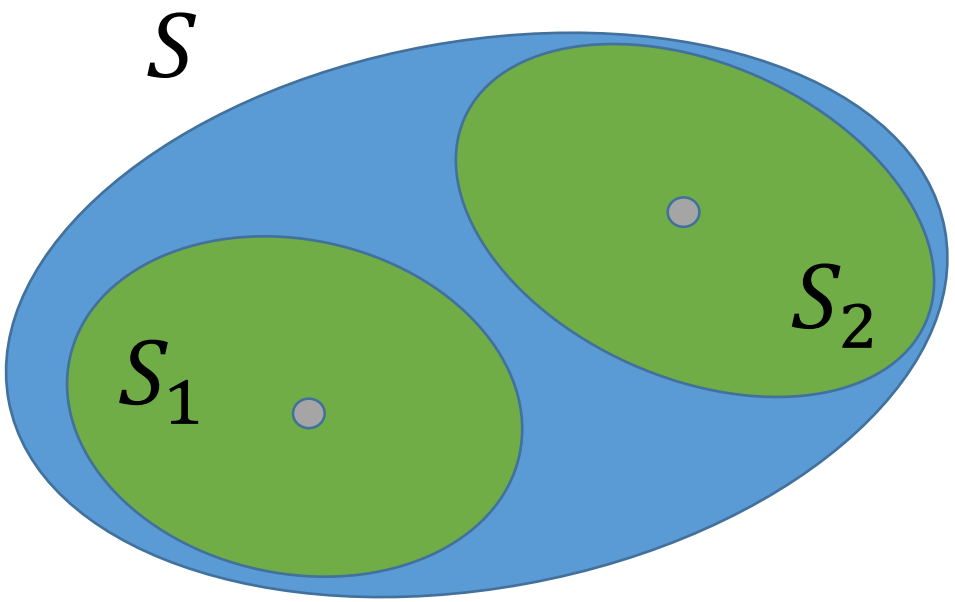
- Split using k-means with $k = 2$
- k-means objective: $\sum_{i=1}^{k} \sum_{x \in S_i} d\left(x, \mu(S_i)\right)^2$
  - Centroid: $\mu(S) := \frac{1}{|S|} \sum_{x \in S} x$



Divisive Methods

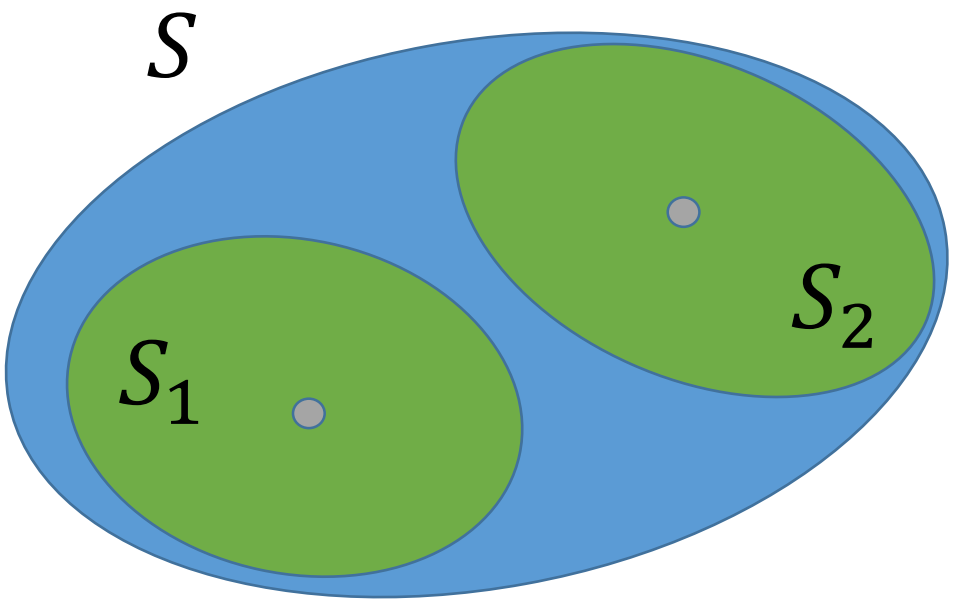
- Split using k-means with $k = 2$
- k-means objective: $\sum_{i=1}^{k} \sum_{x \in S_i} d\left(x, \mu(S_i)\right)^2$
  - Centroid: $\mu(S) := \frac{1}{|S|} \sum_{x \in S} x$

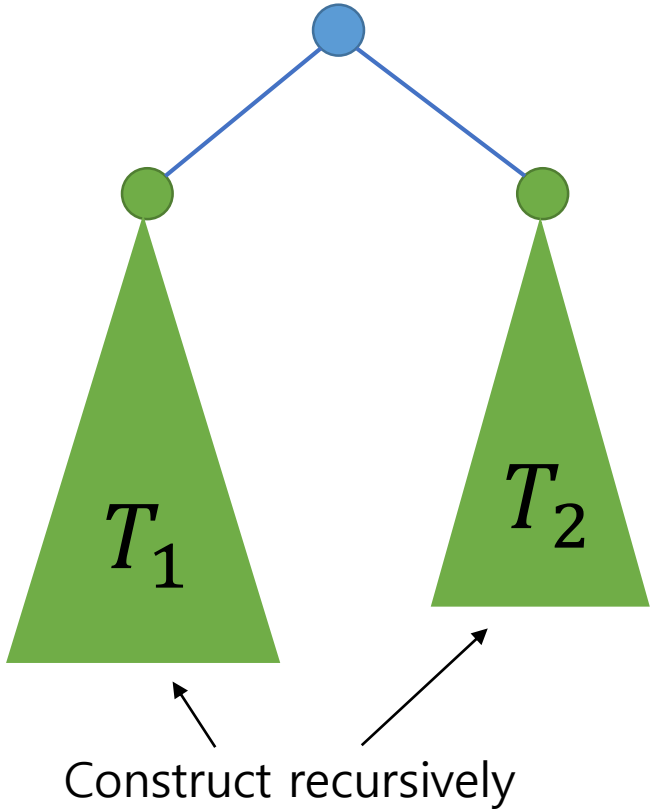




Divisive Methods

- Split using k-means with $k = 2$
- k-means objective: $\sum_{i=1}^{k} \sum_{x \in S_i} d\left(x, \mu(S_i)\right)^2$
  - Centroid: $\mu(S) := \frac{1}{|S|} \sum_{x \in S} x$





Agglomerative Methods

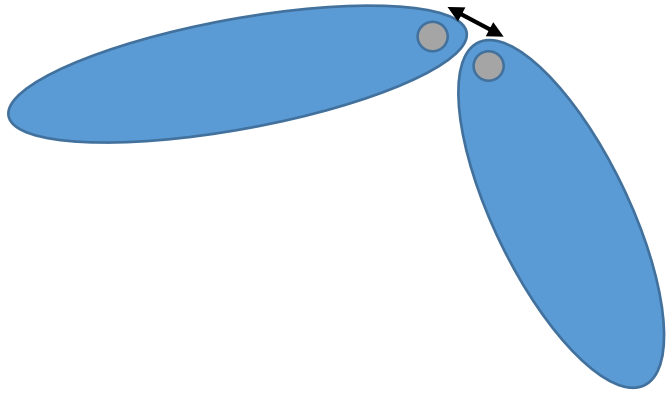
- Assign cost to merging a pair of clusters
- Iteratively merge pairs with lowest cost

Agglomerative Methods

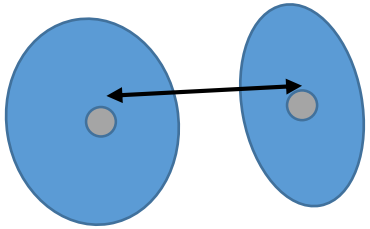
- Assign cost to merging a pair of clusters
- Iteratively merge pairs with lowest cost
- How to assign the cost?

Agglomerative Methods

- Assign cost to merging a pair of clusters
- Iteratively merge pairs with lowest cost
- How to assign the cost?



Single Linkage: $\min_{A,B} \min_{x \in A, y \in B} d(x, y)$



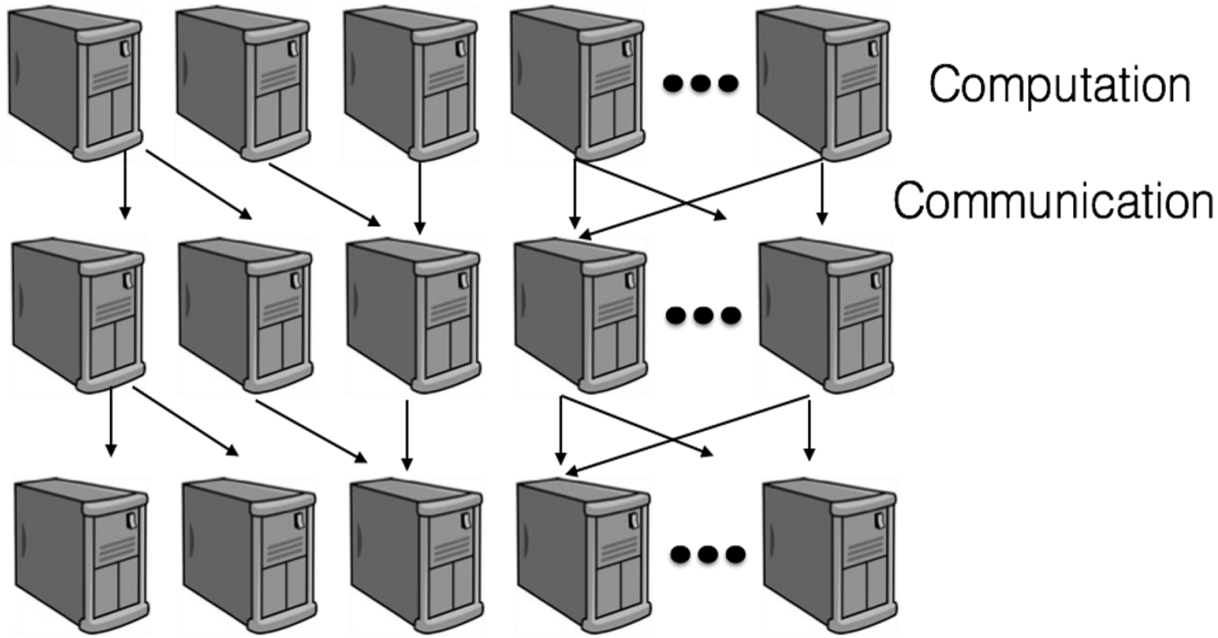
Centroid Linkage: $\min_{A,B} d(\mu(A), \mu(B))$

Motivation

- Popular data analysis tool
- Large sequential runtime observed in practice
  - $\Omega(n^2)$ time complexity for many methods
- Is there a way to improve scalability of standard methods?

Massively Parallel Computation

- Inspired by Spark and MapReduce
- m machines with space s
  - Ideally $s \approx \frac{n}{m}$
- Computation runs for r rounds
- Want small number of rounds - $O(\log n)$

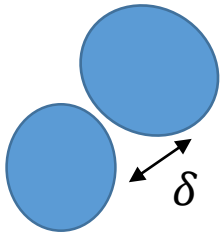


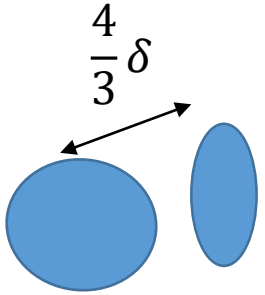
Related Work

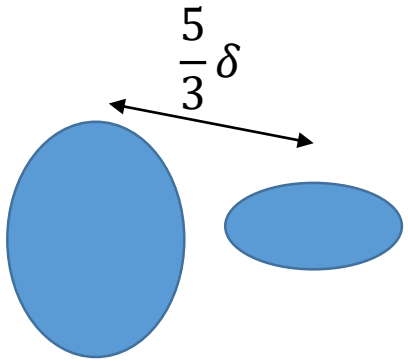
- MPC algorithms for Single Linkage
  - [Bateni et al., Jin et al., Yaroslatsev et al.]
- Single Linkage ↔ Minimum Spanning Tree
- Many results for Minimum Spanning Tree in MPC
  - [Andoni et al., Bader et al., Karloff et al., Lattanzi et al., Qin et al.]

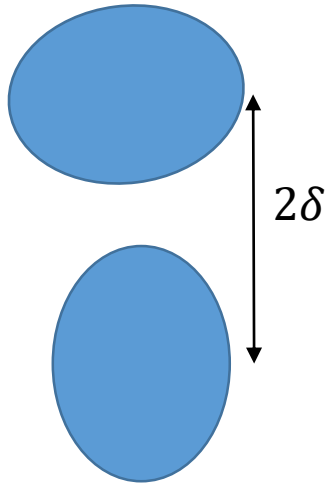
The Main Difficulty

- Need to make decisions sequentially!



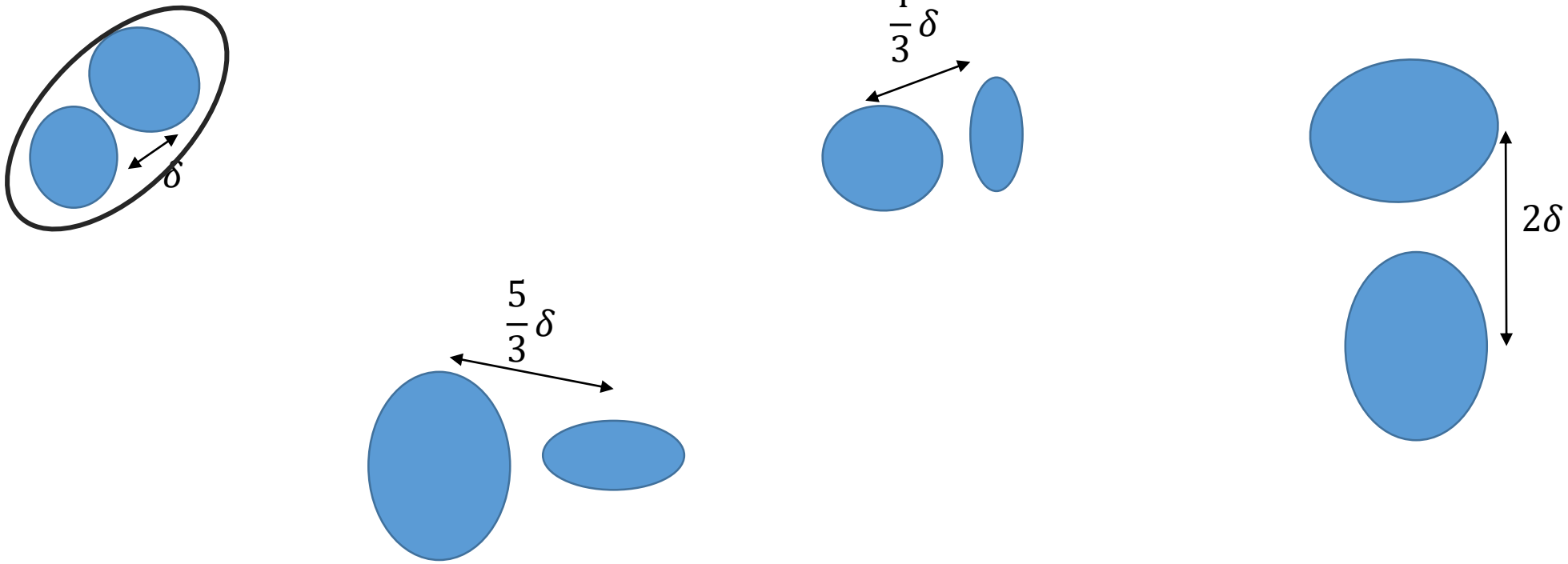






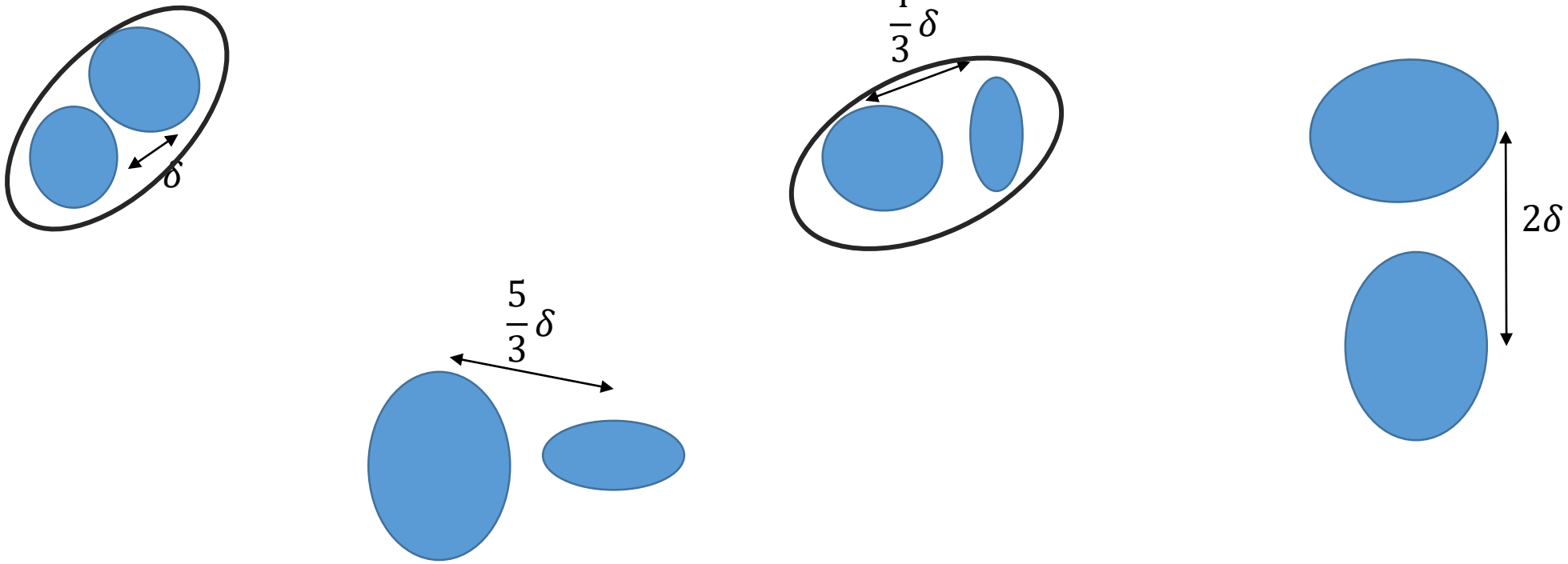
The Main Difficulty

- Need to make decisions sequentially!



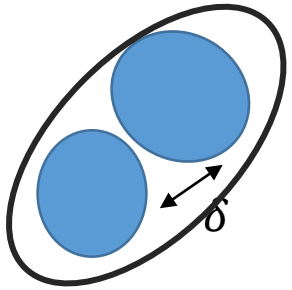
The Main Difficulty

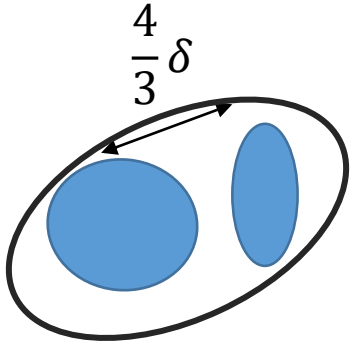
- Need to make decisions sequentially!

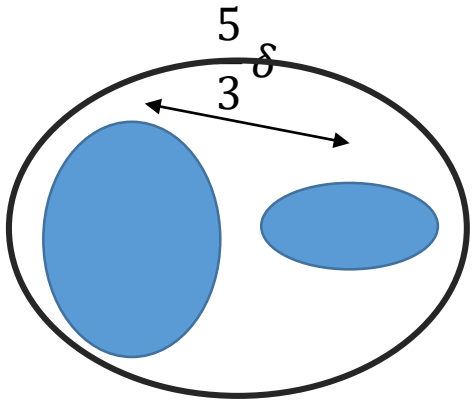


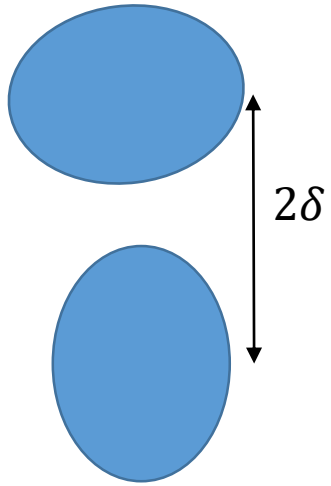
The Main Difficulty

- Need to make decisions sequentially!



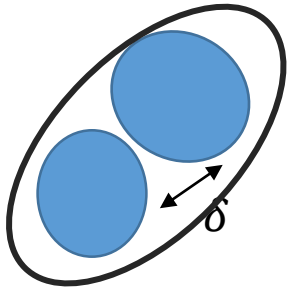


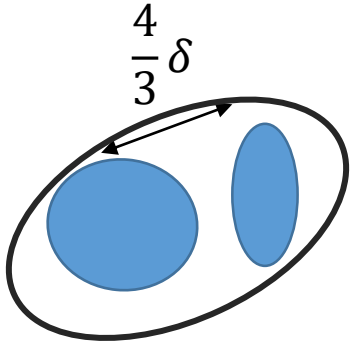


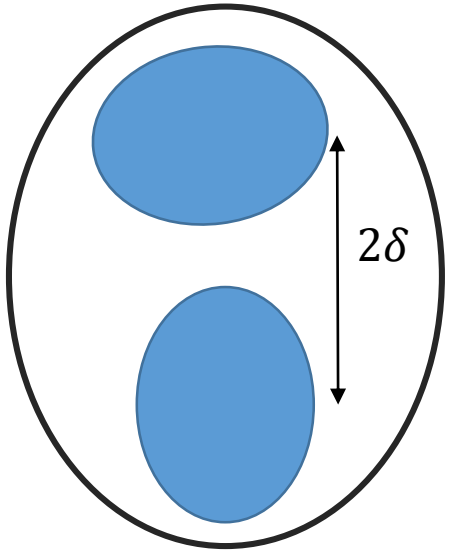


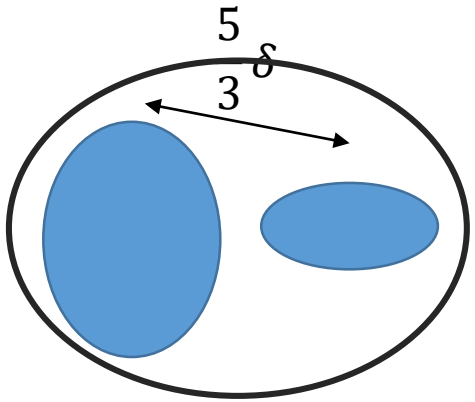
The Main Difficulty

- Need to make decisions sequentially!



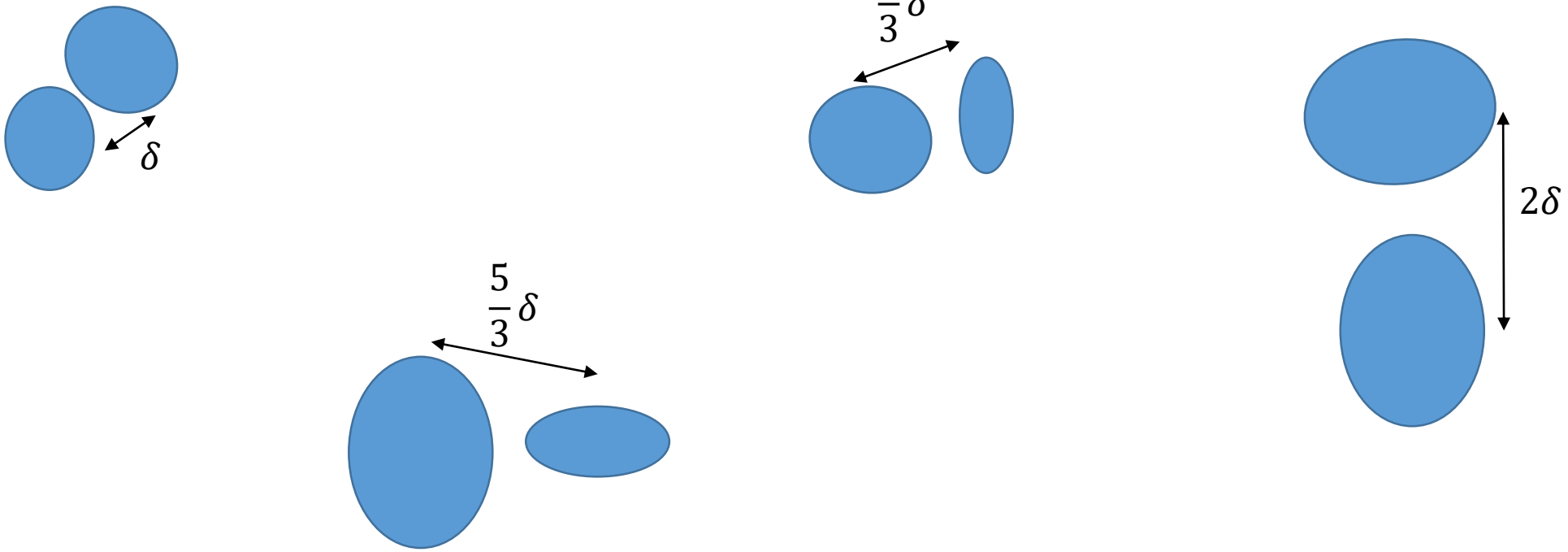






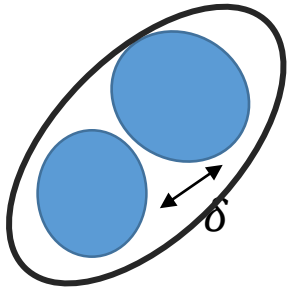
The Main Difficulty

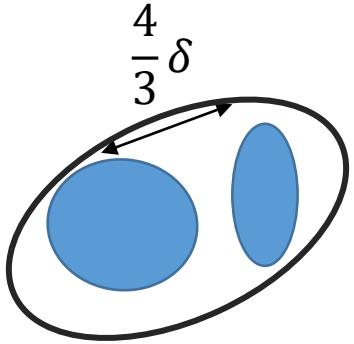
- What if we could merge anything within 2δ?

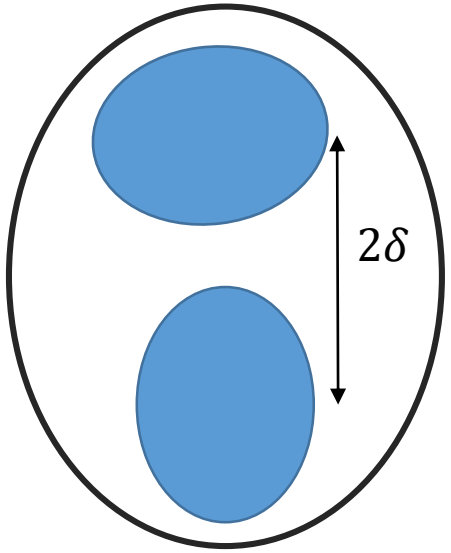


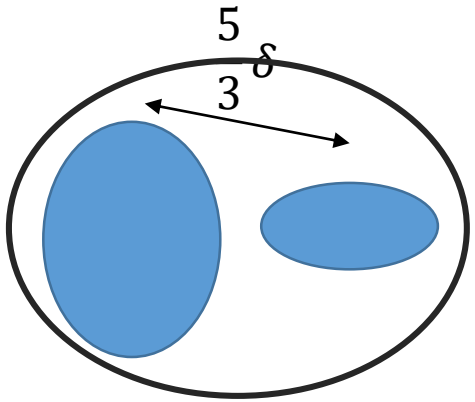
The Main Difficulty

- What if we could merge anything within 2δ?









Close Hierarchical Clustering

- Relaxation of standard methods
- Closeness - worst case measure for quality
- Algorithm α-close if for every merge/split: algorithm chooses sets A', B' such that

$$c(A', B') \leq \alpha \cdot \min_{A,B \in F} c(A, B)$$

e.g. 2-means cost, distance b/w centroids

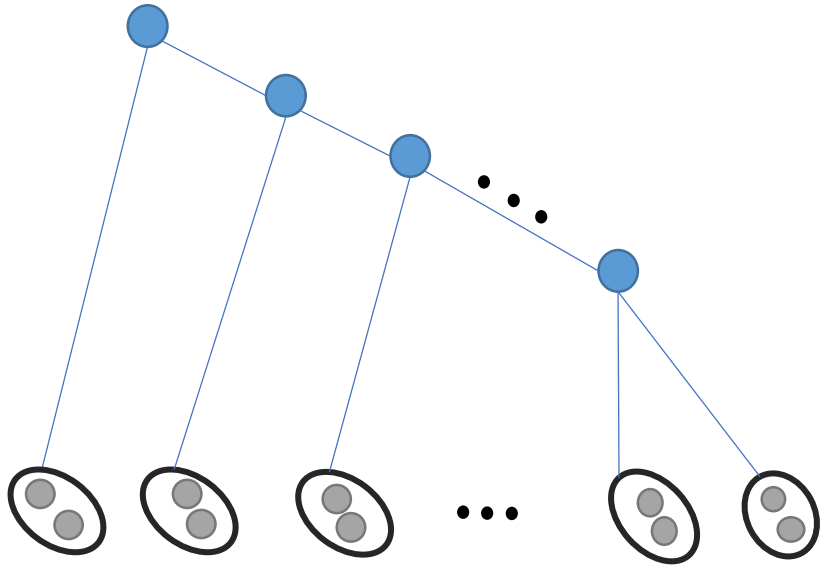
Current feasible splits/merges

Results

- Give efficient α-close MPC algorithms
- Divisive k-means
  - $\alpha = O(1)$, $O(\log n)$ rounds
- Centroid Linkage
  - $\alpha = O(\log^2 n)$, $\tilde{O}(\log^2 n)$ rounds
- Empirical Results
  - Closeness typically small
  - # of rounds scales logarithmically

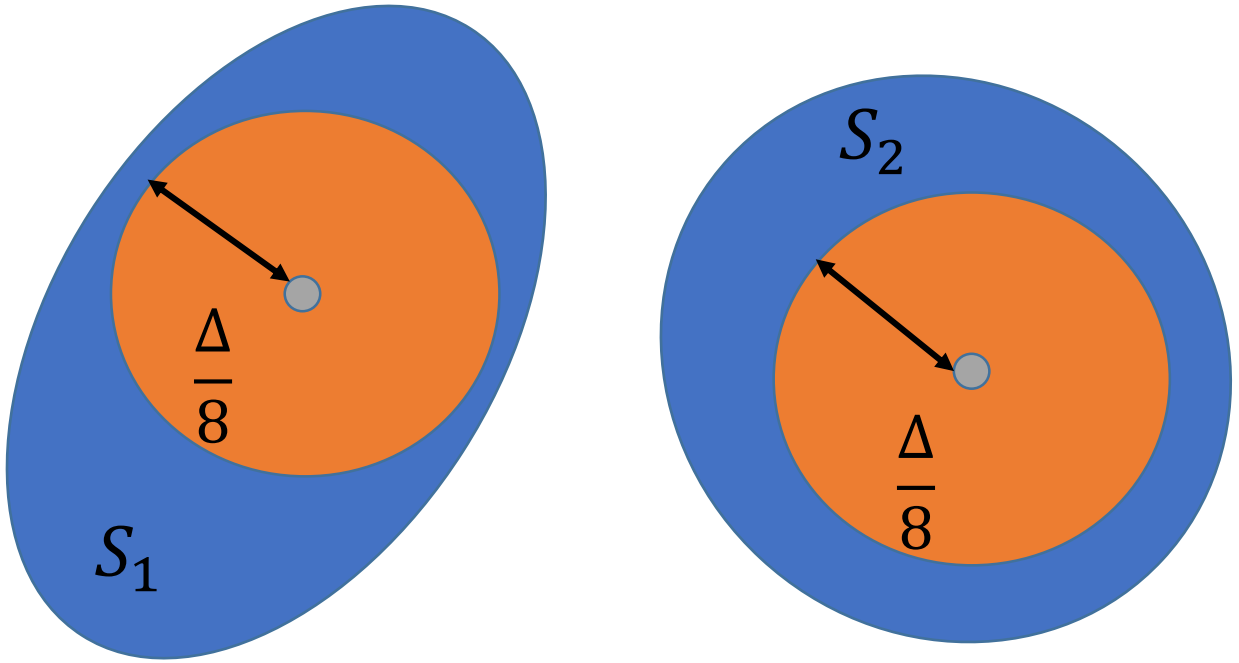
Divisive k-means

- Computing k-means well studied in MPC
  - $O(1)$ round algorithms known
- Problem: split given by k-means may be very unbalanced



Divisive k-means

- Want to reassign points in split S_1, S_2 while maintaining:
  - Reassignment only increases 2-means cost by $O(1)$-factor
  - Either size or diameter of each set decreases geometrically



- Let Δ be the diameter of S
- Reassign points in
  (● points) – (● points)

Centroid Linkage

- Want to find pairs of clusters w/ nearby centroids
- Idea: use ideas from approximate near neighbor search
  - Locality Sensitive Hashing [Indyk et al., Datar et al.]
  - Hash function where nearby points more likely to collide
- Use LSH to partition point set
- Do merges within a partition in parallel

Experiments - Closeness

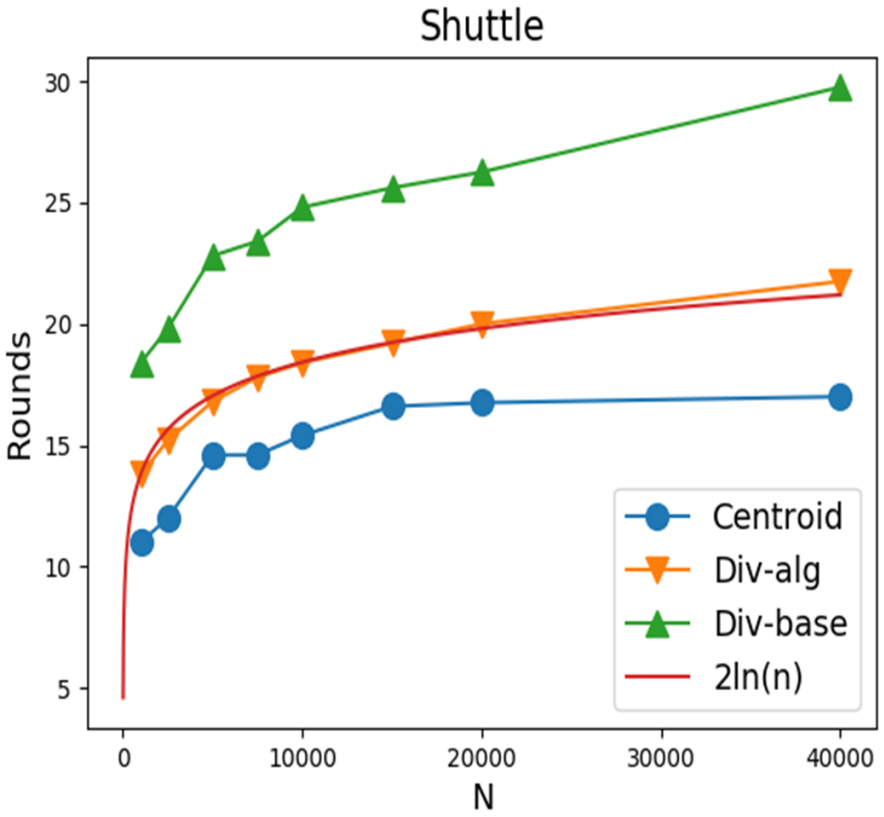
Size	Shuttle	Skin	Covertype
≤1000	1.51	1.61	1.51
2000	1.69	1.74	1.58
3000	1.74	1.91	1.22
4000	1.57	2.10	1.74
5000	-	1.19	-
6000	-	2.30	-
8000	1.64	-	2.01
≥10000	1.74	1.84	1.07
Overall	1.52	1.61	1.51

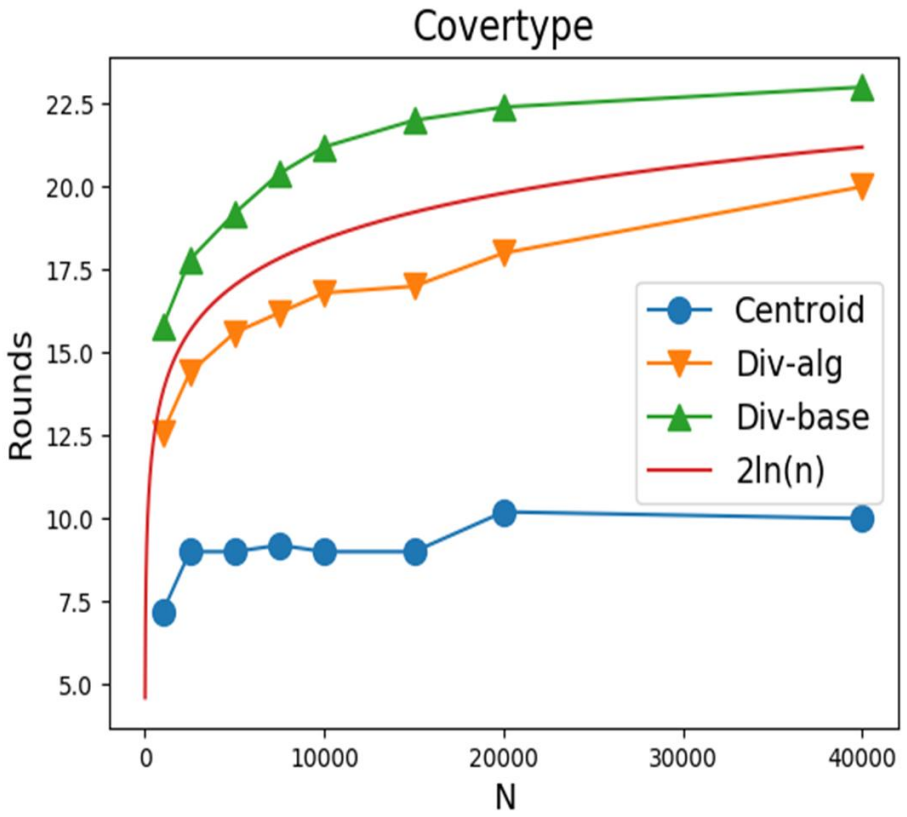
Divisive 2-means (Reassignment)

Size	Shuttle	Skin	Covertype
≤1000	2.74	2.66	2.38
2000	2.66	2.56	2.70
3000	2.76	2.25	2.72
4000	2.50	2.89	-
5000	-	3.16	1.81
6000	1.84	-	-
7000	2.48	3.40	2.11
8000	2.72	1.16	-
9000	-	-	1.92
≥10000	1	2.84	1
Overall	2.74	2.66	2.38

Centroid Linkage (LSH)

Experiments - Rounds





Conclusion

- Give efficient MPC algorithms which approximately simulate divisive k-means and centroid linkage methods
- Future work: extend to other agglomerative methods
  - Average Linkage
  - Ward's Method

Conclusion

- Give efficient MPC algorithms which approximately simulate divisive k-means and centroid linkage methods
- Future work: extend to other agglomerative methods
  - Average Linkage
  - Ward's Method

Thank You!

References

- Jin, C., Liu, R., Chen, Z., Hendrix, W., Agrawal, A., Choudhary, A.N.: A scalable hierarchical clustering algorithm using spark. In: BigDataService 2015. pp. 418–426 (2015)
- Karloff, H.J., Suri, S., Vassilvitskii, S.: A model of computation for mapreduce. In: SODA 2010. pp. 938–948 (2010)
- Lattanzi, S., Moseley, B., Suri, S., Vassilvitskii, S.: Filtering: a method for solving graph problems in mapreduce. In: SPAA 2011 (Co-located with FCRC 2011). pp. 85–94 (2011)
- Qin, L., Yu, J.X., Chang, L., Cheng, H., Zhang, C., Lin, X.: Scalable big graph processing in mapreduce. In: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. pp. 827–838. SIGMOD '14 (2014)
- Yaroslavtsev, G., Vadapalli, A.: Massively parallel algorithms and hardness for single-linkage clustering under lp distances. In: ICML 2018. pp. 5596–5605 (2018)

References – continued

- Andoni, A., Nikolov, A., Onak, K., Yaroslavtsev, G.: Parallel algorithms for geometric graph problems. In: Symposium on Theory of Computing (STOC) 2014
- Bateni, M., Behnezhad, S., Derakhshan, M., Hajiaghayi, M., Lattanzi, S., Mirrokni, V.: On distributed hierarchical clustering. In: NIPS 2017 (2017)
- Bader, D.A., Cong, G.: Fast shared-memory algorithms for computing the minimum spanning forest of sparse graphs. J. Parallel Distrib. Comput. 66(11), 1366–1378
- Datar, M., Immorlica, N., Indyk, P., Mirrokni, V.S.: Locality-sensitive hashing scheme based on p-stable distributions. In: SoCG 2004 (2004)
- Indyk, P., Motwani, R.: Approximate nearest neighbors: Towards removing the curse of dimensionality. In: STOC 1998. pp. 604–613 (1998)