Divisive clustering algorithms

Procheta Sen

UNIVERSITY OF LIVERPOOL

Divisive (top-down) methods: main idea

- A top-down approach is used to successively partition the data objects into a tree-like structure.

- A flat clustering algorithm $\mathscr{A}$ (e.g. k-means algorithm) may be used for the partitioning in a given step. The algorithm $\mathscr{A}$ can be any arbitrary clustering algorithm, not necessarily a distance based algorithm.

Divisive (top-down) methods: main idea

- Provide flexibility in terms of choosing the trade-off between the balance in the tree structure and the balance in the number of objects in each cluster. For example
  - **Strategy 1**: split the heaviest node (a cluster with the maximum number of objects). Will result in leaf nodes (clusters) with a similar number of objects in them.
  - **Strategy 2**: split each cluster into the same number of subclusters. Will result in a balanced tree structure with the same number of children at each node, but the leaf nodes (most granular clusters) will have varying numbers of objects.

Generic divisive clustering algorithm

Input: dataset: $\mathscr{D}$; flat algorithm: $\mathscr{A}$

1. **Initialise** tree $\mathscr{T}$ to contain a single (root) vertex with entire dataset $\mathscr{D}$
2. **Repeat**:
    1. Select a leaf node L in $\mathscr{T}$ based on pre-defined criterion;
    2. Use algorithm $\mathscr{A}$ split L into $L_1, \ldots, L_k$;
    3. Add $L_1, \ldots, L_k$ as children of L in $\mathscr{T}$;
3. **Until** termination criterion
4. **Return** current clustering **or** hierarchy or clusterings

Generic divisive clustering algorithm

Input: dataset: $\mathscr{D}$; flat algorithm: $\mathscr{A}$

1. **Initialise** tree $\mathscr{T}$ to contain a single (root) vertex with entire dataset $\mathscr{D}$
2. **Repeat**:
    1. Select a leaf node L in $\mathscr{T}$ based on pre-defined criterion;
    2. Use algorithm $\mathscr{A}$ split L into $L_1, \ldots, L_k$;
    3. Add $L_1, \ldots, L_k$ as children of L in $\mathscr{T}$;
3. **Until** termination criterion
4. **Return** current clustering **or** hierarchy or clusterings

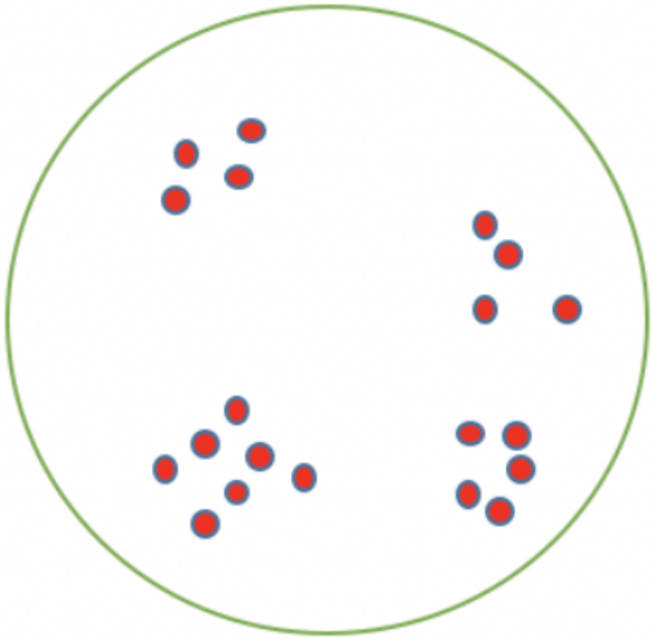
Need to specify

Bisecting k-Means algorithm

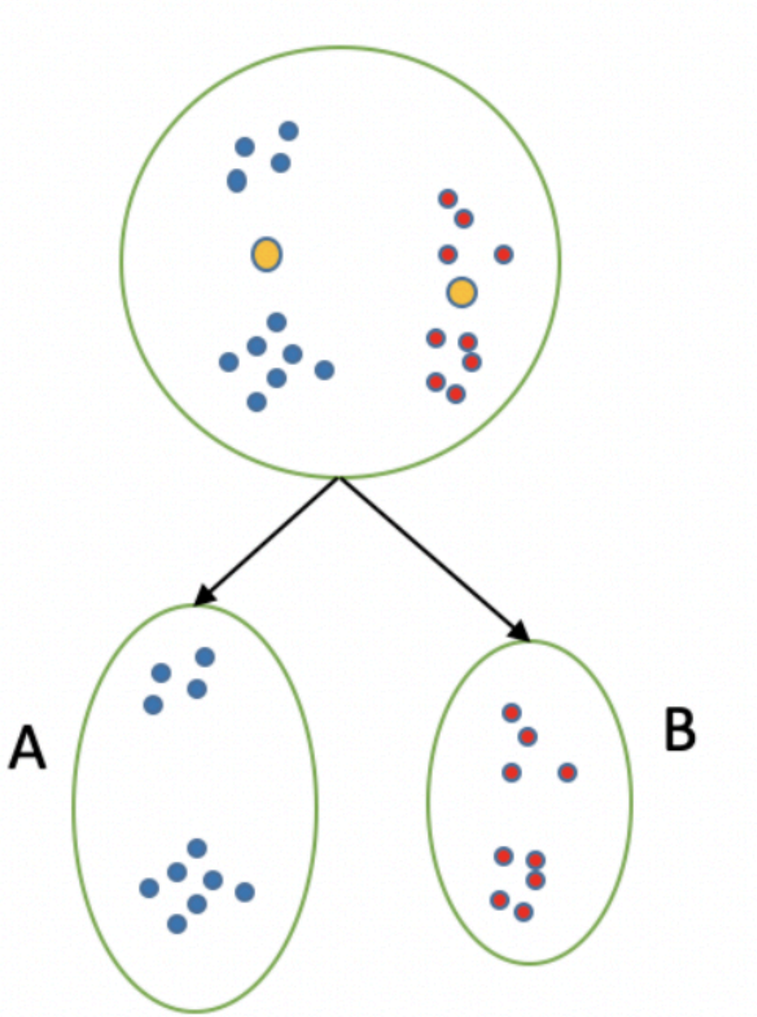
Input: dataset: $\mathscr{D}$; number of clusters: s

1. **Initialise** tree $\mathscr{T}$ to contain a single (root) vertex with entire dataset $\mathscr{D}$
2. **Repeat**:
    1. Select a leaf node (cluster) L in $\mathscr{T}$ that has the largest sum of square distance $\sum_{\overline{X},\overline{Y}\in L} dist(\overline{X},\overline{Y})^2$;
    2. Split L into 2 clusters L_1, L_2 using k-means algorithm;
    3. Add L_1, L_2 as children of L in $\mathscr{T}$;
3. **Until** the number of leaf clusters is s
4. **Return** the leaf clusters

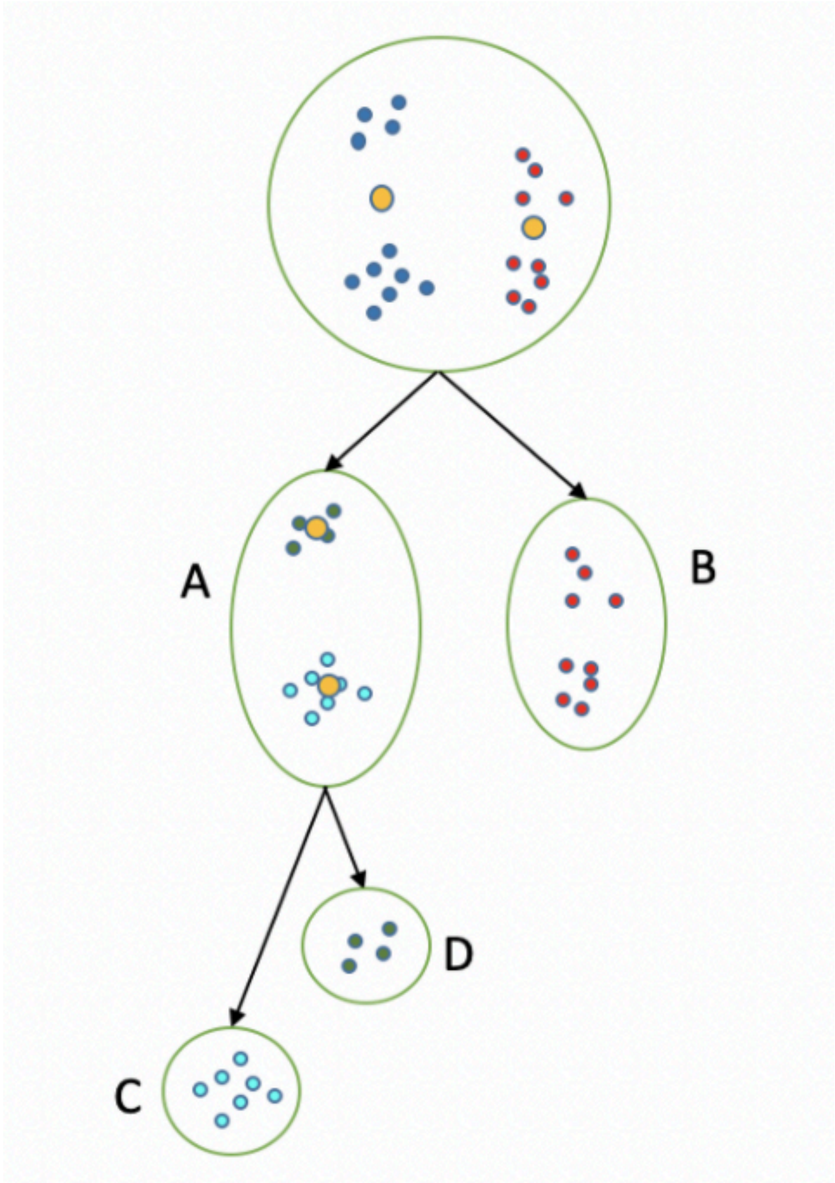
Bisecting k-Means algorithm: example



Bisecting k-Means algorithm: example



Bisecting k-Means algorithm: example



Bisecting k-Means algorithm: example