An Application of Minimum Spanning Trees

T. M. Murali

September 28, 2009

Minimum Spanning Trees

- We motivated MSTs through the problem of finding a low-cost network connecting a set of nodes.
- MSTs are useful in a number of seemingly disparate applications.
- We will consider the problem of clustering.

Motivation for Clustering

- Given a set of objects and distances between them.
- Objects can be images, web pages, people, species ....
- Distance function: increasing distance corresponds to decreasing similarity.
- Goal: group objects into clusters, where each cluster is a set of similar objects.

Formalising the Clustering Problem

- Let U be the set of n objects labelled $p_1, p_2, \ldots, p_n$.
- For every pair p_i and p_j, we have a distance $d(p_i, p_j)$.
- We require $d(p_i, p_i) = 0$, $d(p_i, p_j) > 0$, if $i \neq j$, and $d(p_i, p_j) = d(p_j, p_i)$

Formalising the Clustering Problem

- Let U be the set of n objects labelled $p_1, p_2, \ldots, p_n$.
- For every pair p_i and p_j, we have a distance $d(p_i, p_j)$.
- We require $d(p_i, p_i) = 0$, $d(p_i, p_j) > 0$, if $i \neq j$, and $d(p_i, p_j) = d(p_j, p_i)$
- Given a positive integer k, a *k-clustering* of U is a partition of U into k non-empty subsets or "clusters" $C_1, C_2, \ldots C_k$.

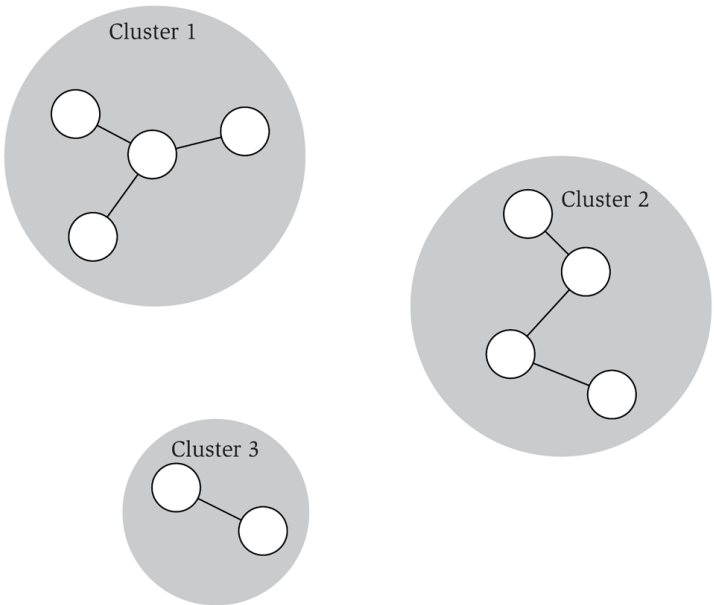

Figure 4.14 An example of single-linkage clustering with $k = 3$ clusters. The clusters are formed by adding edges between points in order of increasing distance.

Formalising the Clustering Problem

- Let U be the set of n objects labelled $p_1, p_2, \ldots, p_n$.
- For every pair p_i and p_j, we have a distance $d(p_i, p_j)$.
- We require $d(p_i, p_i) = 0$, $d(p_i, p_j) > 0$, if $i \neq j$, and $d(p_i, p_j) = d(p_j, p_i)$
- Given a positive integer k, a *k-clustering* of U is a partition of U into k non-empty subsets or "clusters" $C_1, C_2, \ldots C_k$.

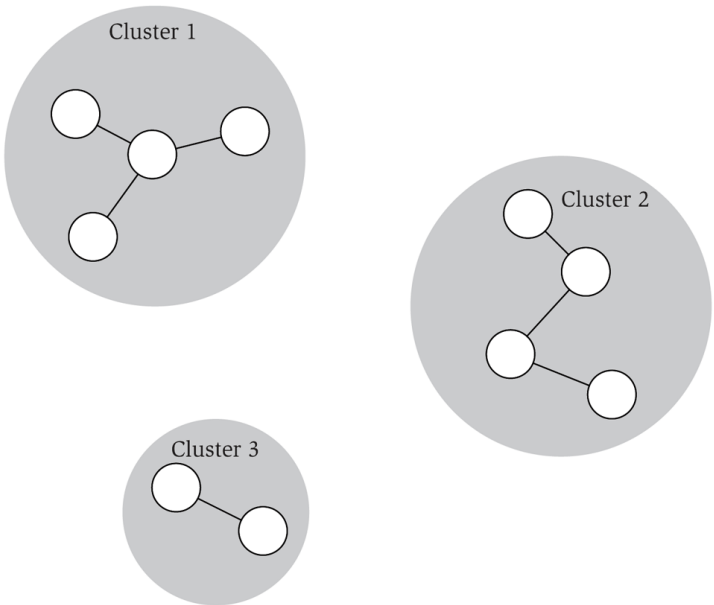

Figure 4.14 An example of single-linkage clustering with $k = 3$ clusters. The clusters are formed by adding edges between points in order of increasing distance.

- The *spacing* of a clustering is

Formalising the Clustering Problem

- Let U be the set of n objects labelled $p_1, p_2, \ldots, p_n$.
- For every pair p_i and p_j, we have a distance $d(p_i, p_j)$.
- We require $d(p_i, p_i) = 0$, $d(p_i, p_j) > 0$, if $i \neq j$, and $d(p_i, p_j) = d(p_j, p_i)$
- Given a positive integer k, a *k-clustering* of U is a partition of U into k non-empty subsets or "clusters" $C_1, C_2, \ldots C_k$.

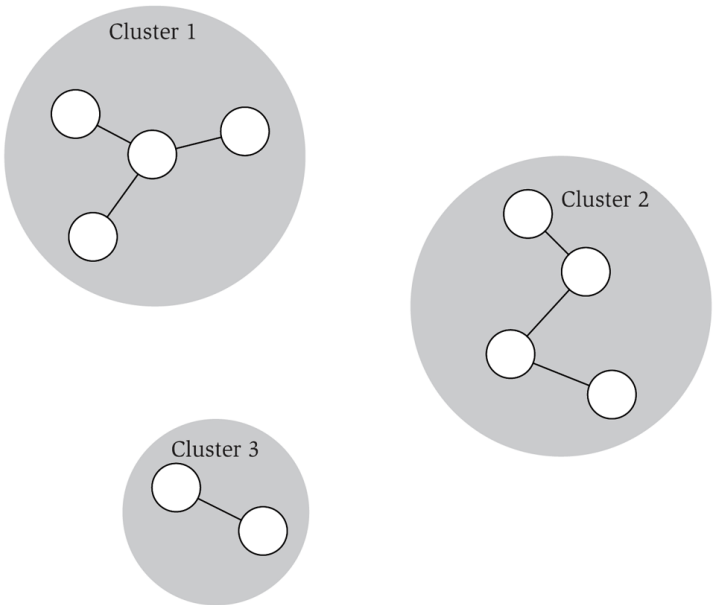

Figure 4.14 An example of single-linkage clustering with $k = 3$ clusters. The clusters are formed by adding edges between points in order of increasing distance.

- The *spacing* of a clustering is the smallest distance between objects in two different subsets:

$$\text{spacing}(C_1, C_2, \ldots C_k) = \min_{\substack{1 \leq i,j \leq k \\ i \neq j, \\ p \in C_i, q \in C_j}} d(p, q)$$

The Clustering Problem

- The *spacing* of a clustering is the smallest distance between objects in two different subsets:

$$\text{spacing}(C_1, C_2, \ldots C_k) = \min_{\substack{1 \le i,j \le k \\ i \neq j, \\ p \in C_i, q \in C_j}} d(p, q)$$

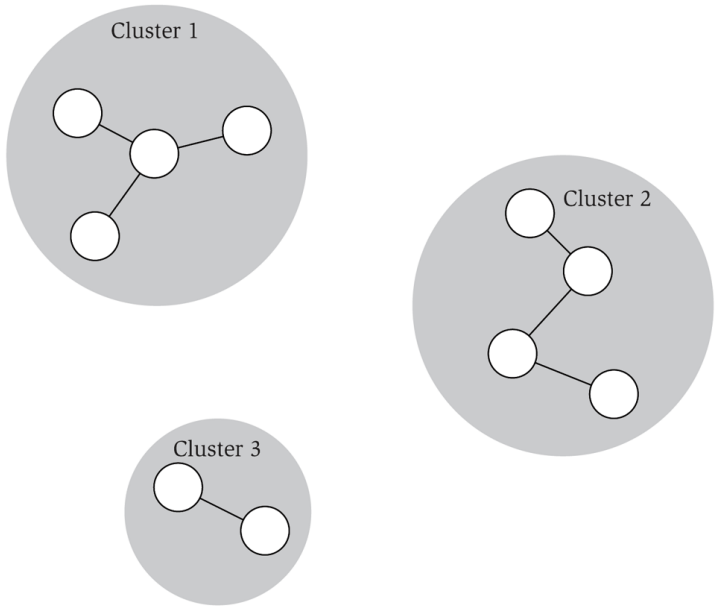

Figure 4.14 An example of single-linkage clustering with $k = 3$ clusters. The clusters are formed by adding edges between points in order of increasing distance.

The Clustering Problem

- The *spacing* of a clustering is the smallest distance between objects in two different subsets:

$$\text{spacing}(C_1, C_2, \ldots C_k) = \min_{\substack{1 \le i,j \le k \\ i \ne j, \\ p \in C_i, q \in C_j}} d(p, q)$$

Clustering of Maximum Spacing

INSTANCE: A set U of objects, a distance function $d : U \times U \to \mathbb{R}^+$, and a positive integer k

SOLUTION: A k-clustering of U whose spacing is the largest over all possible k-clusterings.

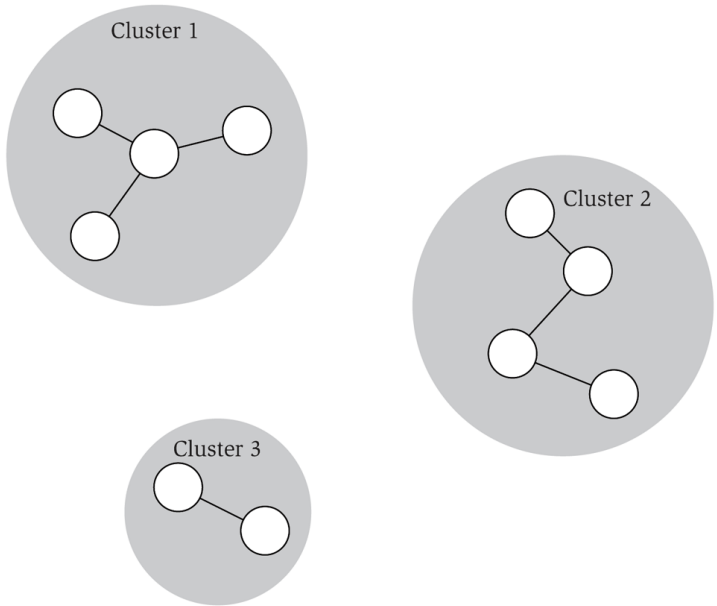

Figure 4.14 An example of single-linkage clustering with $k = 3$ clusters. The clusters are formed by adding edges between points in order of increasing distance.

Algorithm for Clustering of Maximum Spacing

- Intuition: greedily cluster objects in increasing order of distance.

Algorithm for Clustering of Maximum Spacing

- Intuition: greedily cluster objects in increasing order of distance.
- Let $\mathcal{C}$ be a set of n clusters, with each object in U in its own cluster.
- Process pairs of objects in increasing order of distance.
  - Let (p, q) be the next pair with $p \in C_p$ and $q \in C_q$.
  - If $C_p \neq C_q$, add new cluster $C_p \cup C_q$ to $\mathcal{C}$, delete C_p and C_q from $\mathcal{C}$.
- Stop when there are k clusters in $\mathcal{C}$.

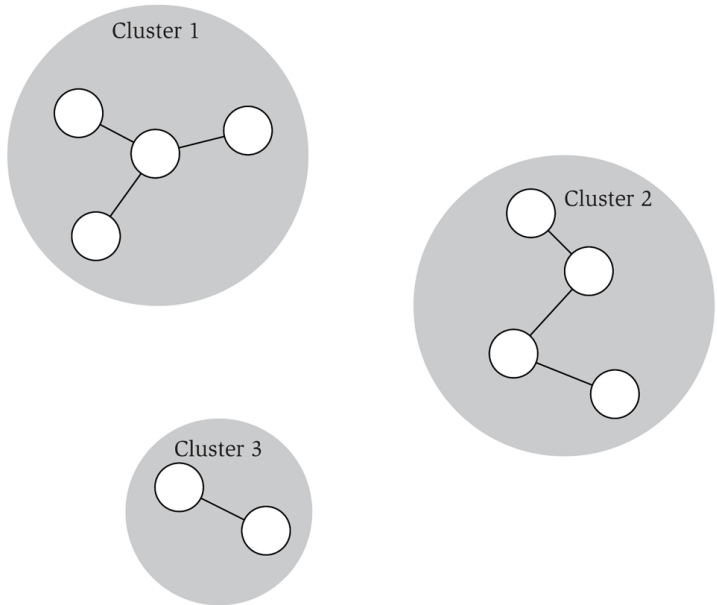

Figure 4.14 An example of single-linkage clustering with $k = 3$ clusters. The clusters are formed by adding edges between points in order of increasing distance.

Algorithm for Clustering of Maximum Spacing

- Intuition: greedily cluster objects in increasing order of distance.
- Let $\mathcal{C}$ be a set of n clusters, with each object in U in its own cluster.
- Process pairs of objects in increasing order of distance.
  - Let (p, q) be the next pair with $p \in C_p$ and $q \in C_q$.
  - If $C_p \neq C_q$, add new cluster $C_p \cup C_q$ to $\mathcal{C}$, delete C_p and C_q from $\mathcal{C}$.
- Stop when there are k clusters in $\mathcal{C}$.
- Same as Kruskal's algorithm but do not add last $k - 1$ edges in MST.

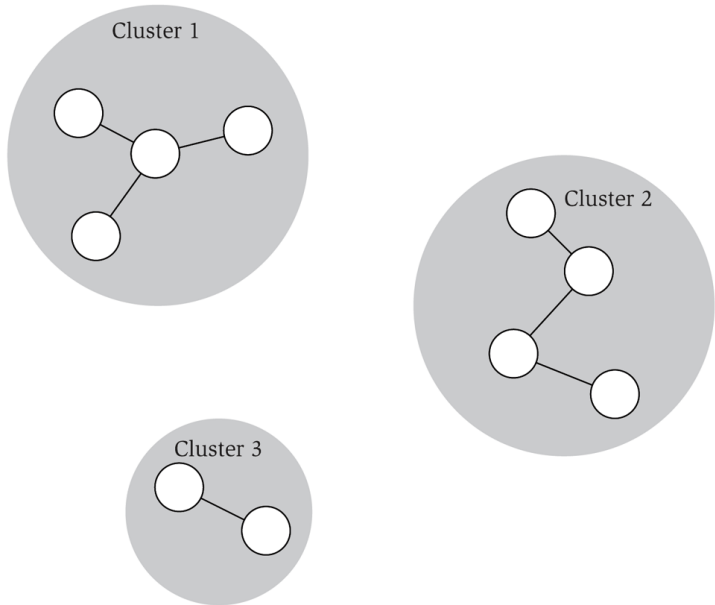

Figure 4.14 An example of single-linkage clustering with $k = 3$ clusters. The clusters are formed by adding edges between points in order of increasing distance.

Why is the Algorithm Optimal?

- Let $\mathcal{C}$ be the clustering produced by the algorithm and let $\mathcal{C}'$ be any other clustering.
- What is spacing($\mathcal{C}$)?

Why is the Algorithm Optimal?

- Let $\mathcal{C}$ be the clustering produced by the algorithm and let $\mathcal{C}'$ be any other clustering.
- What is spacing($\mathcal{C}$)? It is the cost of the $(k-1)$st most expensive edge in the MST. Let this cost be d^*.
- We will prove that spacing($\mathcal{C}'$) $\leq d^*$.

$\textbf{spacing}(\mathcal{C}') \leq d^*$

- There must be two points p_i and p_j in U in the same cluster C_r in $\mathcal{C}$ but in different clusters in $\mathcal{C}'$:

spacing$(\mathcal{C}') \leq d^*$

- There must be two points p_i and p_j in U in the same cluster C_r in $\mathcal{C}$ but in different clusters in $\mathcal{C}'$: spacing$(\mathcal{C}') \leq d(p_i, p_j)$.

spacing$(\mathcal{C}') \leq d^*$

- There must be two points p_i and p_j in U in the same cluster C_r in $\mathcal{C}$ but in different clusters in $\mathcal{C}'$: spacing$(\mathcal{C}') \leq d(p_i, p_j)$. But $d(p_i, p_j)$ could be $> d^*$.
- Suppose $p_i \in C'_s$ and $p_j \in C'_t$ in $\mathcal{C}'$.

spacing$(\mathcal{C}') \leq d^*$

- There must be two points p_i and p_j in U in the same cluster C_r in $\mathcal{C}$ but in different clusters in $\mathcal{C}'$: spacing$(\mathcal{C}') \leq d(p_i, p_j)$. But $d(p_i, p_j)$ could be $> d^*$.
- Suppose $p_i \in C'_s$ and $p_j \in C'_t$ in $\mathcal{C}'$.
- All edges in the path Q connecting p_i and p_j in the MST have length $\leq d^*$.
- In particular, there is a point $p \in C'_s$ and a point $p' \notin C'_s$ such that p and p' are adjacent in Q.
- $d(p, p') \leq d* \Rightarrow$ spacing$(\mathcal{C}') \leq d(p, p') \leq d^*$.

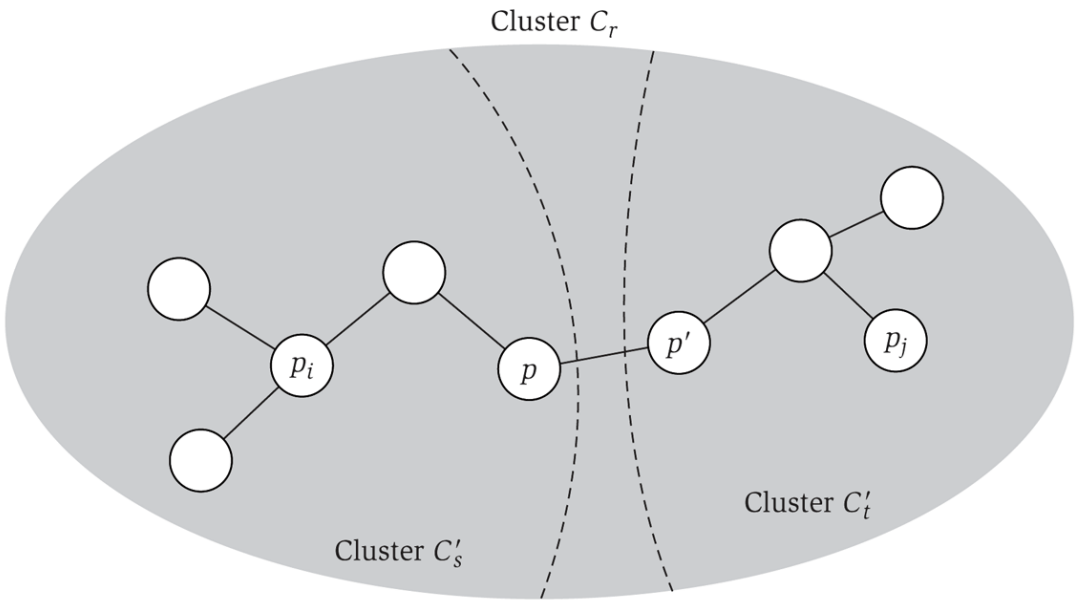

Figure 4.15 An illustration of the proof of (4.26), showing that the spacing of any other clustering can be no larger than that of the clustering found by the single-linkage