Cluster Analysis and other unsupervised learning methods

Minimal spanning trees and single linkage clustering

Werner Stuetzle
Department of Statistics
University of Washington

March 27, 2017

Minimal spanning trees

$G = (V, E)$ connected, undirected graph.

$\tilde{G} = (V, \tilde{E})$ is called spanning tree of G if $\tilde{G}$ has unique path between any two vertices in V.

Suppose there is a cost $c(e)$ associated with any edge $e \in E$.

Definition: Minimal spanning tree = spanning tree with minimal total edge cost.

Special case of interest:

- V = set of points in $\mathbf{R}^p$.
- G = complete graph
- Cost of edge = edge length

Prim's Lemma

Let (V_1, V_2) denote a partitioning of the vertices of V. There is a MST containing a cheapest edge with one end in V_1 and the other end in V_2.

Proof

Let (v_1, v_2) denote a cheapest edge. Suppose MST did not contain a cheapest edge.

- The MST has to contain a path between v_1 and v_2.
- The path has to contain an edge (v_1, v_2) with $v_1 \in V_1$ and $v_2 \in V_2$.
- Removing this edge breaks the MST into two subtrees, one containing v_1, the other containing v_2.
- Adding edge (v_1, v_2) connects the two subtrees into a spanning tree with lower cost. Therefore the original tree could not have been an MST.

Prim's principles for MST construction

1. Any isolated node can be connected to a nearest neighbor
2. Any tree fragment can be connected to a nearest neighbor by a shortest available link.

Nearest neighbor = vertex connected by incident edge with minimum cost

---

The MST and single linkage clustering

Prop

The k-partition obtained by breaking the $k-1$ longest edges is the same as the k-partition constructed by SL-HAC.

Proof

Every merge step in the SL algorithm is in accordance with Prim's principles.

Note

We can compute the SL dendogram by first computing the MST and then recursively breaking longest edges.

We cannot obtain MST from SL dendrogram. Dendrogram does not contain info about closest pair of points in the two clusters that were merged.

The MST and graph thresholding

Given: A $n \times n$ matrix D of dissimilarities between n objects with $d_{ij} \geq 0$.

We can regard the d_{ij} as edge weights in an edge weighted complete graph G with n vertices.

There is an obvious way of generating a hierarchical clustering of the n objects from the graph:

- Define the threshold graph $G^+(\lambda)$ as the graph obtained by $G = G^+(\infty)$ by removing all edges with $d_{ij} \geq \lambda$
- $\mathcal{P}(\lambda)$ = partition of V defined by connected components of $G^+(\lambda)$
- $\mathcal{P}(\infty) = V$
- $\mathcal{P}(0) = (\{v_1\}, \ldots, \{v_n\})$
- The partitions are nested

Let T be the MST of G.

Prop

The connected components of $T^+(\lambda)$ are the same as the connected components of $G^+(\lambda)$.

Proof

Suppose v_1, v_2 are in the same connected component (cc) of $T^+(\lambda) \Rightarrow$ they are in the same cc of $G^+(\lambda)$ because the edges of $T^+(\lambda)$ are a subset of the edges of $G^+(\lambda)$.

Now suppose v_1, v_2 are in different cc's C_1, C_2 of $T^+(\lambda)$. Therefore the unique path in T between v_1 and v_2 has an edge e with weight $\geq \lambda$.

If v_1, v_2 were in the same cc of $G^+(\lambda)$ there would be a path between v_1 and v_2 with all edge weights $< \lambda$.

In this path there has to be an edge e' connecting C_1 and C_2.

We could then remove e from T and replace it with e' and get a tree with smaller total edge weight $\Rightarrow$ T could not have been the MST.