Minimum Spanning Trees

Application to Clustering

Algorithms: Design and Analysis, Part II

Clustering

[aka "unsupervised learning"]

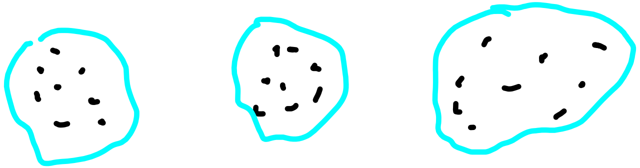
Informal goal: given n "points" [web pages, images, genome fragments, etc.] classify into "coherent groups".

Assumptions: (1) as input, given a (dis)similarity measure — a distance $d(p,q)$ between each point pair.

(2) symmetric [i.e., $d(p,q) = d(q,p)$]

Examples: Euclidean distance, genome similarity, etc.

Goal: same cluster $\Longleftrightarrow$ "nearby"



Tim Roughgarden

Max-Spacing k-Clusterings

Assume: we know $k :=$ # of clusters desired.

[in practice, can experiment with a range of values]

Call points p & q separated if they're assigned to different clusters.

Definition: the spacing of a k-clustering is

$$\min_{\text{separated } p,q} d(p,q).$$

[the bigger, the better]

Problem statement: given a distance measure d and k, compute the k-clustering with maximum spacing.

A Greedy Algorithm

- initially, each point in a separate cluster
- repeat until only k clusters:
  - let p,q = closest pair of separated points
    (determines the current spacing)
  - merge the clusters containing p & q into a single cluster

Note: just like Kruskal's MST algorithm, but stopped early.
- points ⟷ vertices; distances ⟷ edge costs; point pairs ⟷ edges

⇒ called single-link clustering