Outline:

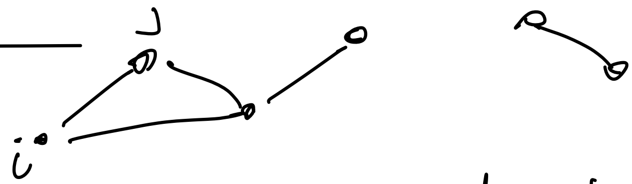
Graph

Clustering

  Dendrograms

union-find algorithm

Spectral Clustering.



Graph: $G(V, E)$

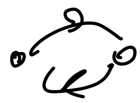
$$|V| = N, \quad |E| = M$$

$$E \subseteq V \times V$$

$$e = (i, j) \quad i, j \in V$$

directed edges

$i \to j \neq i \leftarrow j$



undirected graph

$(i, j) \sim (j, i)$

Self-loop

(i, i)

Examples:

Social Networks (vertices individuals
edge: friend relation)

transportation networks (vertices: cities,
edges: roads)

food webs (species, who eats who)

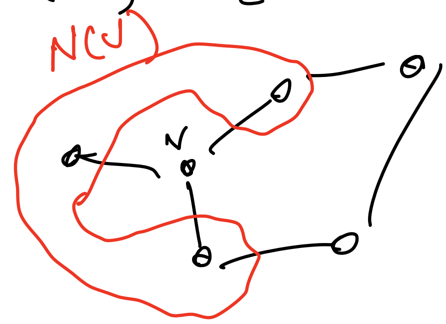
generally, edges encode relationship btw. entities.

Nearest neighbors graph: edge if two points are within some distance from each other.

Assumptions today: undirected, unweighted graphs

$V = \{1, \dots N\}$

Def. the neighborhood of an a vertex $v \in V$ is the set $N(v) = \{ w \in V \mid (v,w) \in E \}$



Def. a path from $i \in V$ to $j \in V$ is a sequence of edges $(i, k_0), (k_0, k_1), \dots (k_{p-1}, k_p), (k_p, j)$

i and j are are in the same connected component if $\exists$ a path between i and j.

Prop: path connectedness is an equivalence relation. $i \sim j$ if $\exists$ path $i \to j$

identity, symmetry, reflexivity.

$i \sim j$, $j \sim k \Rightarrow i \sim k$ through concatenation of paths.

Equivalence class is a connected component.

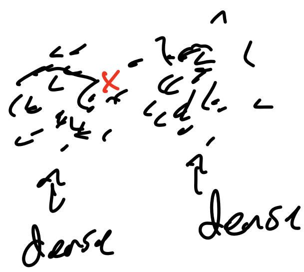
Clustering: there are many ways to define this, and many algs. (ref. on difficulty in clustering Kleinberg)

k-means, DBSCAN, ...

we'll focus on notion that is topologically meaningful: single linkage clustering.

Examples:

"easy to cluster"

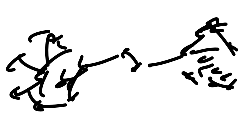


"harder to cluster"

---

idea of single-linkage clustering:

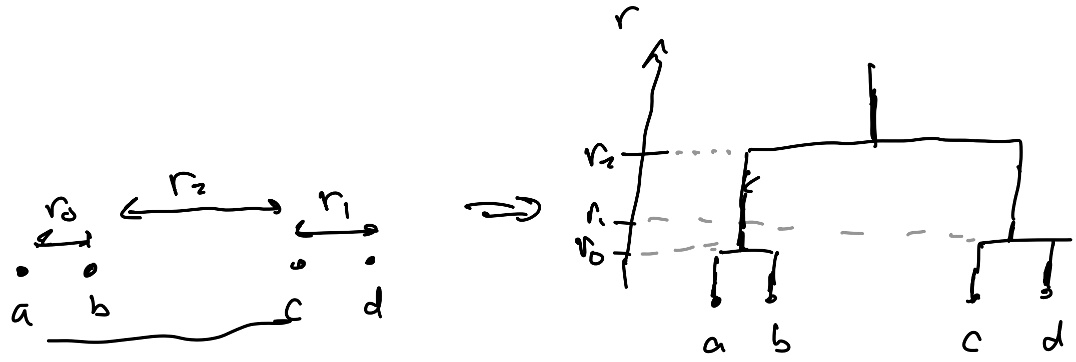
we form a graph that connects points that are near each other. The two clusters merge if there is a single link btw them.



we identify connected components of nbhd graph.

We run into a problem: how to choose nbhd parameter? In practice: use all nbhd parameters.

produce what is called a dendrogram, this shows how clusters merge. ↳ tree



$r_0 < r_1 < r_2$

edge (a, c) at param $r_0 + r_2$
but already in same cluster

---

single linkage: single edge merges clusters
average linkage: merge at avg. distance
Complete linkage: need to add all edges btw. clusters to merge.

Union-find / Disjoint set data structure

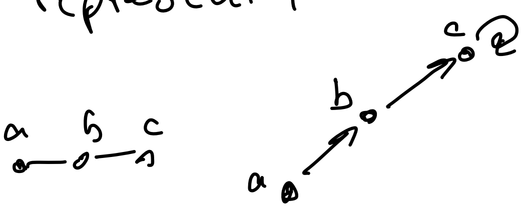
can use to compute dendrogram.

Disjoint set data structure:

two operations:

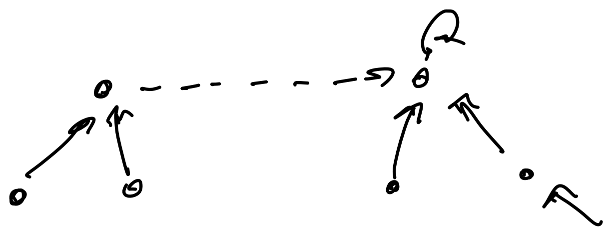
find (find connected component)
union / merge (merge two connected components)

idea: every cluster has a representative point
every vertex has a parent in same cluster
representative point is its own parent.

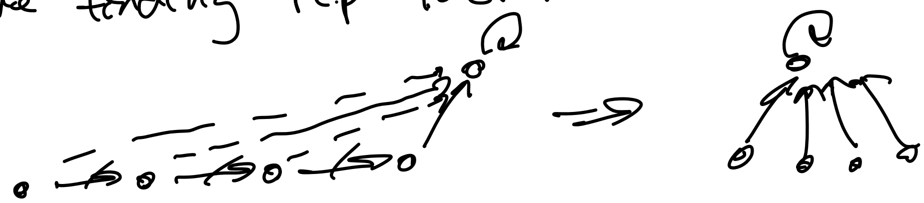


c is the representative point

To merge two clusters, simply find the representative for each cluster, and then make the parent of the rep for smaller cluster the rep for larger cluster.



idea (important for performance) "path compression"
make finding rep faster each time.



how to represent on computer:

list data structure: N points Array

"parent" array of length N

parent[i] = j $\Leftrightarrow$ $\underset{i}{\circ} \longrightarrow \underset{j}{\circ}$

parent[i] = i if i rep of cluster.

---

to form dendrogram every time we add an edge to nbhd graph (i,j) try merging clusters that contain i and j.

If $rep(i) = rep(j)$ then already same cluster.

if $rep(i) \neq rep(j)$ then merge two components dendrogram just needs to remember which components merged, and which edge caused this to happen, so we can look up parameter value.

analysis: $O(M \underbrace{\alpha}_{\text{inverse ackerman function}}(N))$ time

---

Spectral Clustering:

recall incidence matrix: $B \in \mathbb{R}^{N \times M}$

$$\left.\begin{array}{l} B[i,k] = -1 \\ B[j,k] = +1 \end{array}\right\} e_k = (i,j)$$

$$B[\cdot,k] = 0 \text{ ow.}$$

we define graph Laplacian $L = BB^T$, $L \in \mathbb{R}^{N \times N}$

Exercise: $L = D - A$, D is degree matrix, A adjacency matrix

Prop: L satisfies the following properties:

1) $x^T L x = \sum_{(i,j) \in E} (x_i - x_j)^2$

2) L is symmetric, positive semi-definite

3) The null eigenspace of L is spanned by indicator vectors on CC.

Pf: 1: $x^T L x = (x^T B)(B^T x) = (B^T x)^T (B^T x)$

$$\left.\begin{aligned} B[i,k] &= -1 \\ B[j,k] &= +1 \\ B[:,k] &= 0 \end{aligned}\right\} e_k = (i,j)$$

$$B^T[k] = x_j - x_i$$

$$(B^T x)^T (B^T x) = \sum_{(i,j)\in E} (x_j - x_i)^2$$

2) symmetry obvious. Positive semi-definite:

$x^T L x \geq 0 \;\; \forall x$ implied by (1)

3) we can verify this let $\mathbb{1}_C$ be an indicator on C. C.

$$\mathbb{1}_C[i] = \begin{cases} 1 & \text{if } i \in C \\ 0 & \text{if } i \notin C \end{cases}$$

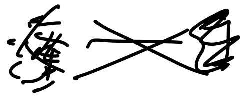
$x_i - x_j = 0 = 1 - 1$ if $i, j \in C$ ←

$x_i - x_j = 0 = 0 - 0$ if $i, j \notin C$ ←

no other edges.

$\mathbb{1}_C^T L \mathbb{1}_C = 0$ $\forall$ indicators on CC. $\square$

What abt. weak connections? e.g. SBM.



want to partition V into $S \subseteq V$ $\bar{S} \subseteq V$

$$S \cup \bar{S} = V, \quad S \cap \bar{S} = \emptyset$$

minimize quantity

$$h_G(S) = \frac{|E(S, \bar{S})|}{\min(|S|, |\bar{S}|)}$$

Cheeger inequality:

$$2h_G \le \lambda_1 \le \frac{h_G^2}{2}$$ (λ_1 = smallest non-zero eigenvalue of L)

Idea to use eigenvector v_1 for an embedding and do clustering in embedded space.