SPECTRAL clustering

The $\boxed{\text{RANDOM WALK}}$ Method

Remember $\boxed{\text{RANDOM WALK} \equiv \text{MARKOV CHAIN}}$

~~similarity~~

- Random walk on a graph

- Consider a random walk on an undirected graph

- Spectral clustering can be interpreted as trying to find a partition of the graph such that the random walk stays long within the same cluster and seldom jumps between clusters.

- This is consistent with the GRAPH CUT intuition we discussed.

  A partition with a low CUT will have the property that the random walk does not have many opportunities to jump between clusters

- Defining the RANDOM WALK

  The probability of jumping from vertex i to vertex j is $p_{ij} = \frac{w_{ij}}{d_i}$, where $\left\{ \begin{array}{l} w_{ij} \text{ is the weight of edge } (i,j) \end{array} \right.$

  and d_i = degree of vertex i

The transition matrix of the random walk

$$P = \{p_{ij}\}_{i,j=1,2,\dots}$$

defined as $P = D^{-1} W$

where D is the degree matrix

$$D = \begin{bmatrix} d_1 & & 0 \\ & \ddots & \\ 0 & & d_n \end{bmatrix}$$

$$D^{-1} = \begin{bmatrix} \frac{1}{d_1} & \cdots & 0 \\ 0 & & \frac{1}{d_n} \end{bmatrix}$$

W = the graph matrix

- If the graph G is connected and not bipartite, the RANDOM WALK has a unique stationary distribution

$$\pi = (\pi_1, \dots, \pi_n)^T$$

which is given by

$$\pi_i = \frac{d_i}{vol(G)}$$

- A beautiful relation between the normalized Laplacian L_{rw} and the Random Walk matrix

$$\boxed{L_{rw} = I - P}$$

As a consequence if λ is an eigenvalue of L_{rw} with eigenvector v <u>iff</u> $1-\lambda$ is an eigenvalue of P with eigenvector v

<u>conclusion:</u> The largest eigenvalues of P and

the smallest eigenvalues of L_{RW} can be used to describe cluster properties of the graph.

RANDOM WALKS & Ncut

- A formal equivalence between Ncut and $\underline{P}$:

PROPOSITION

Let G be a connected graph and non-bipartite.
Assume that we run the random walk $(X_t)_{t \geq 0}$

starting with X_0 in the stationary distribution π.

For disjoint subsets $A, B \subset V$ denote

$$P(A|B) = \underset{\uparrow}{P}(X_1 \in B \mid X_0 \in A)$$

RANDOM WALK P

Then:

$$\boxed{Ncut(A, \bar{A}) = P(\bar{A}|A) + P(A|\bar{A})}$$

This property tells us that when minimizing Ncut, we actually look for a cut through the graph such that the random walk seldom transitions from A to $\bar{A}$ or vice-versa

The COMMUTE DISTANCE

A tight connection between random walks & graph Laplacians can be made via the <u>commute distance</u> on the ~~the~~ the graph.

The <u>commute distance</u> is also known as

<u>resistance distance</u>

in electronics graph theory.

The <u>commute distance</u> $c(i,j)$
between vertices i and j is the expected time of it takes the random walk to travel from vertex i to vertex j <u>and back</u>.

- The commuting distance has some nice properties which makes it particularly appealing for machine learning.

- As opposed to the shortest path distance on a graph, the commute distance between two vertices decreases if there are many different short ways to get from vertex i to vertex j.

- So instead of just looking for one shortest path, the commute distance looks at the set of short paths.

- Points which are connected by a short path and lie in the same cluster of the graph are <u>much closer</u> to each other than points which are connected by a short path but lie in different clusters.

- $\boxed{\text{Remarkably}}$ the commute distance on a graph can be computed by the <u>pseudo-inverse</u> or <u>Moore-Penrose inverse</u> L^+ of the graph Laplacian L

- The above shows that

$$\sqrt{c(i,j)}$$

can be considered as a Euclidian distance function on the vertices of the graph.

This means that we can construct an embedding which maps the vertices v_i of the graph on points $z_i \in \mathbb{R}^n$ such that the Euclidean between the points z_i coincide with the commute distance on the graph.

$\boxed{\text{Avoiding this way the "curse of dimensionality"}}$