CS 59000-ENS: Similarity

Assefaw Gebremedhin
Purdue University
agebreme@purdue.edu
http://www.cs.purdue.edu/homes/agebreme/Networks
Spring 2013
Similarity

- In what ways can vertices in a network be similar?
- How can we quantify that similarity?
- Which vertices in a given network are most similar to one another?
- Which vertex $v$ is most similar to a given vertex $u$?

Answers to such questions can help tease apart the types and relationships of vertices in social and information networks.
Two types of similarity

- **Structural equivalence**
  - Vertices share many of the same network neighbors

- **Regular equivalence**
  - Vertices have neighbors who are themselves similar

We will look at some mathematical measures that quantify these ideas of similarity.
Cosine similarity

- Simplest measure: count number of common neighbors.
- In an undirected network, the number of common neighbors between \( i \) and \( j \) is
  \[
  n_{ij} = \sum_k A_{ik}A_{kj}
  \]
  Which is the \( ij \)th element of \( A^2 \).
- This, however, does not tell us much on its own. We need some sort of normalization.
- Cosine similarity (suggested by Salton (89)) is one such normalized quantity.
Cosine similarity

- Idea, the inner product between two vectors \( \mathbf{x} \) and \( \mathbf{y} \) is
  \[ \mathbf{x} \cdot \mathbf{y} = |\mathbf{x}| |\mathbf{y}| \cos \theta \]
  where \( \theta \) is the angle between the two vectors.

  Rearranging,
  \[ \cos \theta = \frac{\mathbf{x} \cdot \mathbf{y}}{|\mathbf{x}| \cdot |\mathbf{y}|} \]

- Salton proposed to regard the \( i \)th and \( j \)th row (or column) of the adjacency matrix as two vectors and use the cosine of the angle between them as the similarity measure.

- Noting that the dot product is \( \sum_k A_{ik} A_{kj} \)
  The similarity measure is
  \[ \sigma_{ij} = \cos \theta = \frac{\sum_k A_{ik} A_{kj}}{\sqrt{\sum_k A^2_{ik}} \sqrt{\sum_k A^2_{jk}}} \]

- Assuming unweighted graph, entries of \( \mathbf{A} \) are either zero or one, thus
  \[ \sigma_{ij} = \frac{\sum_k A_{ik} A_{kj}}{\sqrt{d_i} \sqrt{d_j}} = \frac{n_{ij}}{\sqrt{d_id_j}} \]
Pearson coefficients

- **Normalization factor**: the expected value the count would take on a network in which vertices choose their neighbors at random.

- Suppose vertices $i$ and $j$ have degrees $d_i$ and $d_j$.

- Suppose further that vertex $i$ chooses the $d_i$ neighbors uniformly at random from the $n$ possibilities, and vertex $j$ similarly chooses $d_j$ neighbors at random.

- For the first neighbor that $j$ chooses, there is a probability of $d_i/n$ that it will choose the one of the ones $i$ chose, and similarly for each succeeding choice.

- Then in total the expected number of common neighbors between the two vertices is will be $d_id_j/n$
A reasonable measure of similarity between two vertices is the actual number of common neighbors they have minus the expected number that they would have if they chose their neighbors at random:

\[
\sum_k A_{ik} A_{jk} - \frac{d_i d_j}{n} = \sum_k A_{ik} A_{jk} - 1/n \sum_k A_{ik} \sum_l A_{jl} \\
= \sum_k A_{ik} A_{jk} - n \bar{A}_i \bar{A}_j \\
= \sum_k [A_{ik} A_{jk} - \bar{A}_i \bar{A}_j] \\
= \sum_k (A_{ik} - \bar{A}_i)(A_{jk} - \bar{A}_j)
\]

This equation is simply \( n \) times the covariance \( \text{cov}(A_i, A_j) \) of the two rows of the matrix.
Pearson coefficient

- It is common to normalize this quantity, so that its maximum value is 1.

- The maximum value of the covariance of any two sets occurs when the sets are exactly the same, in which case their covariance is equal to the variance of either set, $\sigma_i \sigma_j$.

- Normalizing by this quantity gives us the standard Pearson correlation coefficient:

$$r_{ij} = \frac{\text{cov}(A_i, A_j)}{\sigma_i \sigma_j}$$

- This quantity lies between -1 and 1.
Euclidean distance

- Measures the number of vertices that are neighbors of $i$ but not of $j$ (more of a dissimilarity measure)
- In terms of the adjacency matrix, Euclidean distance can be written as
  \[ d_{ij} = \sum_{k} (A_{ik} - A_{jk})^2 \]
  Convenient to normalize by dividing by maximum possible value.
- The maximum value of $d_{ij}$ occurs when two vertices have no neighbor in common, in which case $d_{ij} = d_i + d_j$.
- Dividing by this quantity, the normalized distance becomes
  \[ 1 - 2 \frac{n_{ij}}{d_i + d_j} \]
Regular equivalence
(less developed compared structural equivalence)

- Similarity score $\sigma_{ij}$ such that $i$ and $j$ have high similarity if they have neighbors $k$ and $l$ that themselves have high similarity. For an undirected network,

$$\sigma_{ij} = \alpha \sum_{kl} A_{ik} A_{jl} \sigma_{kl}$$

Or in matrix terms

$$\sigma = \alpha A \sigma A$$

- A type of eigenvector equation.

- This formula has some problems:
  - It does not necessarily give a high value for self similarity.
  - It does not necessarily give a high similarity score to vertex pairs that have a lot of common neighbors.

- Fix: introduce an extra diagonal term in the similarity thus

$$\sigma_{ij} = \alpha \sum_{kl} A_{ik} A_{jl} \sigma_{kl} + \delta_{ij}$$

- or in matrix notation

$$\sigma = \alpha A \sigma A + I$$

- Can be further generalized (for all paths) as

$$\sigma = \sum_{m=0}^{\infty} (\alpha A)^m = (I - \alpha A)^{-1}$$

(reminiscent of Katz centrality)
Homophily

- People form friendships with those that are similar to them
- This is called homophily or assortative mixing
- More rarely one also sees disassortative mixing
Figure 4.14: The tendency of people to live in racially homogeneous neighborhoods produces spatial patterns of segregation that are apparent both in everyday life and when superimposed on a map — as here, in these maps of Chicago from 1940 and 1960 [302]. In blocks colored yellow and orange the percentage of African-Americans is below 25, while in blocks colored brown and black the percentage is above 75.
Groups of vertices

- Cliques
- Plexes
- Cores
- Components
Groups of vertices

- **Clique:**
  - Maximal subset of vertices such that every member of the set is connected by an edge to every other.
  - Presence indicates a highly cohesive group.
  - Finding maximum clique is NP-hard.
  - Fast algorithm for sparse graphs possible (our group has developed one such, state-of-the-art)

- **K-core:**
  - Maximal subset of vertices such that each is connected to at least $k$ others in the subset.
  - The max $k$-core (and in fact all $k$-cores) in a graph can be computed in linear time.
  - Algorithm: start with the vertex of degree less than $k$, remove, and iterate.

- **K-plex:**
  - A maximal subset of $n$ vertices such that each vertex is connected with at least $n-k$ of the others. ($k=1$ corresponds to clique).
  - Cliques and $k$-plexes can overlap, but cores do not (they are rather layered).