2019-06-14

1 Centrality and ranking

For a given network, how important is any given node or edge? The answer to this question depends a great deal on context. Do we care about the immediate neighbors of a node (degree centrality), how close the node is to other nodes (closeness centrality), or if a node sits in a cross roads for traffic through the network (node between-ness centrality)? Do we care about the edges that are most important in keeping a graph together (edge between-ness centrality)? Do we want an egocentric measure that focuses on the graph near a particular target node (personalized PageRank)? All these different approaches lead to different — though sometimes related — measures of *centrality* or importance of a node or edge.

When we think about computations involving centrality measures, we care about a number of different aspects:

- Does a particular notion of centrality capture the notion of importance that matters for our application? Does it make sense given how the network is constructed?
- Is the measure stable, or can "small" changes to the network (for some application-dependent notion of "small") change it dramatically?
- Can we efficiently compute the centrality for the types of graphs we care about? If we are concerned with large graphs, we may seek a cheaper approximate computation instead of an exact computation; how good is the approximation?

In the rest of the lecture, we discuss these different aspects of centrality computation for several different types of centrality.

2 Degree centrality

The simplest measure of centrality is (weighted) node degree. In a directed graph, we distinguish between in-degree and out-degree. For example, in a social network, we might declare someone to be famous if they have many followers (a high in-degree). This measure is easy to compute, and it is easy

to understand the stability to small changes in the network. Unfortunately, the degree is also a very local property, and may miss network features that matter a lot if we are concerned with the bigger picture. For example, a paper in a citation network may be heavily cited by the author and his students; but if nobody else cites the work of the group, it has little importance.

3 Spectral centrality measures

Many centrality measures are based on the idea of a balance or exchange between neighboring nodes, and these centrality measures typically lead to eigenvalue problems, usually involving non-negative matrices. A useful preliminary to discussing these measures is the *Perron-Frobenius* theorem.

In the least restrictive case, the Perron-Frobenius theorem says that every non-negative matrix has a positive real eigenvalue with absolute value at least as great as that of any other eigenvalue, and there is an associated non-negative eigenvector with some elementwise non-negative row and column eigenvectors. We typically assume the non-negative matrix is *irreducible* (i.e. it cannot be rewritten as a block upper triangular matrix); in this case, this largest positive eigenvalue (the Perron-Frobenius eigenvalue) has both algebraic and geometric multiplicity one and has associated positive row and column eigenvectors. The Perron eigenvalue is bounded from above and below by the largest and smallest column sums.

One of the simplest centrality measures is *eigenvector* centrality (also known as Bonacich centrality). The idea of eigenvector centrality is that the importance of a node is determined by whether it has important neighbors; that is, if x is the vector of centralities, we want

$$x_i = \frac{1}{\lambda} \sum_j a_{ij} x_j,$$

or, in matrix form,

$$Ax = \lambda x.$$

The vector x is only determined up to a scaling factor; in this setting, we often choose to enforce that the entries sum to one, i.e.

$$\sum_{i=1}^{n} x_i = e^T x = 1.$$

Eigenvector centrality is correlated with degree centrality, but is more sensitive to the overall shape of the network. Unfortunately, it may be rather sensitive. For example, consider the graph consisting of two cliques connected by a single edge. The eigenvector centrality in this case may change enormously if we remove the edge in one direction.

The HITS algorithm generalizes the notion of eigenvector centrality to the case where we have two complementary centralities: *hubs* and *authorities*. A hub is important if it is points to important authorities, and an authority is important if it points to important hubs. So if a_{ij} denotes the weight from j to i, then the hub importance vector x and the authority vector y should satisfy

$$y = \lambda A x, \quad x = \lambda A^T y.$$

That is, x and y are the singular vectors associated with the largest singular value of A.

If we scale the edge weights on the graph so that all nodes have weighted out-degree one, we are left with the stationary distribution for a Markov chain, i.e.

$$AD^{-1}x = x.$$

If the original graph was undirected, then $d = Ae$ is the vector of node degrees, and

$$AD^{-1}d = Ae = d,$$

so the stationary distribution is proportional to the vector of node degrees in the original graph. In this case, we recover degree centrality as a case of eigenvector centrality. But even with this scaling, the two notions differ when the graph is directed.

The eigenvector centrality vector, the stationary state of a random walk on a graph, and the dominant singular pair for a graph can all be computed by power iteration or by more rapidly convergent Krylov subspace iterations (Lanczos and Arnoldi). However, convergence may be slow if another eigenvalue is close to the Perron eigenvalue, which may happen in networks that exhibit strong clusters with weak connections between them. In this case, the slow convergence relates to the fact that the solution may be rather sensitive to small changes in the network.

The *PageRank* algorithm can be seen as a regularized version of computing the usual stationary distribution for a random walk on a graph. Rather than the usual model of a random *walker* that transitions only along edges

in a graph, we consider a random *surfer* who usually behaves like a walker, but sometimes (with probability α) teleports to a new network location. The PageRank vector is the stationary vector of

$$P_{\text{PR}} = (1-\alpha)P + \alpha\pi_{\text{ref}}e^T$$

where π_{ref} is a reference distribution used for teleporting. The PageRank vector x can either be written as the stationary vector for P_{PR}, i.e.

$$\left[(1-\alpha)P + \alpha\pi_{\text{ref}}e^T\right] x = x,$$

or we can substitute the normalization condition $e^T x = 1$ to get

$$\left[I - (1-\alpha P)\right] x = \alpha\pi_{\text{ref}}.$$

The PageRank iteration, which can be seen as a power method on P_{PR} or as a simple linear solver (Richardson iteration) on the linear system, is

$$x^{k+1} = (1-\alpha)Px^k + \alpha\pi_{\text{ref}}.$$

The iteration satisfies the error equation

$$\|x^{k+1} - x\|_1 \leq (1-\alpha)\|x^k - x\|_1,$$

and so converges rapidly when α is far enough from one – a choice on the order of 0.1 is typical. The iteration converges rapidly because there can be no eigenvalue (other than the one at 1) with modulus greater than $1-\alpha$; this same fact guarantees that the solutions to the PageRank problem are relatively insensitive to small changes in the transition probabilities.

4 Path-based centrality measures

Recall that if A is an adjacency matrix with entries a_{ij} denoting the weight of an edge from j to i, then $[A^k]_{ij}$ denotes the total of all length k paths from j to i. Path-based centrality measures combine these counts to get summary information about nodes in a graph. For example, the *Katz* centrality measure of a node i is a weighted sum of the number of paths going through

node i from any source, where the weight for paths of length k is α^k for some positive α sufficiently less than one. That is,

$$\begin{aligned} x_i &= \sum_{k=1}^{\infty} \sum_{j=1}^{n} \alpha^k [A^k]_{ij} \\ &= \left[\sum_{k=1}^{\infty} \alpha^k A^k e \right]_i \\ x &= \left([I - \alpha A]^{-1} - I \right) e. \end{aligned}$$

Assuming α times the Perron eigenvalue of A is reasonably less than one, we can compute the Katz vector by the iteration

$$x^{k+1} = \alpha A(x^k + e).$$

Different choices of weights give different centrality measures; for example, we could also consider

$$x = \sum_{k=0}^{\infty} \frac{\alpha^k}{k!} A^k e = \exp(\alpha A) e.$$

An alternative is to consider not *all* paths in the network, but only *closed* paths. With the weights $\alpha^k/k!$, this gives us the *Estrada* centrality

$$x_i = [\exp(\alpha A)]_{ii};$$

for the weights α^k, we have the *resolvent* centrality

$$x_i = \left[(I - \alpha A)^{-1} \right]_{ii}.$$

Computing these quantities exactly is typically rather expensive, but there are fast estimation mechanisms. In particular, we can combine Krylov methods for quickly computing $\exp(\alpha A)v$ or $(I - \alpha)^{-1}v$ for an arbitrary vector v with the *stochastic diagonal estimation* idea: if z is a vector of independent entries with mean zero and variance one, then

$$\mathbb{E}\left[z_i [f(A)z]_i \right] = f(A)_{ii}.$$

5 Closeness centrality measures

Yet another notion of centrality involves how close nodes are to each other in the graph. If d_{ij} is the shortest distance from node j to node i, then the *closeness centrality* of j is

$$c_j = \left[\sum_i d_{ij} \right]^{-1}.$$

This version of closeness centrality rewards short paths *from* j; there is also a version that rewards short paths *to* j. The *harmonic centrality* of j is

$$h_j = \sum_i d_{ij}^{-1},$$

i.e. it looks like closeness centrality, but with the sum and the inverse swapped. As with Isomap, the expensive part of a closeness centrality computation is generally the all-pairs shortest path computation used to obtain the distances. Hence, there is an issue to the scalability of closeness cetnrality and harmonic centrality. As with Isomap, though, we can approximate these centrality measures by only looking at paths through intermediate *landmark* or *pivot* vertices in the graph.

The closeness centrality and harmonic centrality consider only the geodesic (shortest path) distances. The *current flow centrality* or equivalently (*information centrality*) instead uses the resistance distance. Let

$$C^I = (L + ee^T)^{-1}$$

be the so-called conductance matrix associated with the graph, and recall that the effective resistance from i to j is

$$C^I_{ii} + C^I_{jj} - C^I_{ji} - C^I_{ij}.$$

Summing over i, we have that the current centrality of j is

$$c_i^{-1} = nc^I_{jj} + \mathrm{tr}(C^I) - \frac{2}{n}.$$

As in the previous section, we can estimate the traces and diagonal elements that appear in this formula via stochastic methods.

The *random walk* closeness centrality is computed in terms of *mean first hitting time* (or *mean first passage*) for a random walker in the graph. Let P be the transition matrix for a random walk in a graph; the mean first hitting time to reach node j from a starting distribution vector π^0 is

$$h_j = \sum_{t=0}^{\infty} e^T [P_{-j}]^t [\pi^0]_{-j}$$

where P_{-j} denotes the matrix P with column and row j omitted, and $[\pi^0]_{-j}$ is the vector π^0 with row j omitted. This is a geometric series, and so we can write the infinite sum as

$$h_j = e^T (I - P_{-j})^{-1} [\pi^0]_{-j}.$$

While this might initially seem awkwardly expensive even for small graphs, we can use the same trick that we saw in our discussion of leave-one-out cross validation, and incorporate the effect of deleting a row and column of P by instead adding a border:

$$h_j = \begin{bmatrix} e \\ 0 \end{bmatrix}^T \begin{bmatrix} I - P & e_j \\ e_j^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} \pi^0 \\ 0 \end{bmatrix}.$$

If P is small enough that factorization of $I-P$ is plausible, we can solve these systems using an LU factorization of the (singular) matrix $I - P$. However, we can go further.

Recognizing that h_j looks very much like a Schur complement in a larger matrix, we rearrange again to obtain

$$\begin{bmatrix} I - P & \pi_0 & e_j \\ e^T & 0 & 0 \\ e_j & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ h^{-1} \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}.$$

Gaussian elimination of y gives us

$$-\begin{bmatrix} e_j \\ 0 \end{bmatrix}^T \begin{bmatrix} I - P & \pi^0 \\ e^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} e_j \\ 0 \end{bmatrix} y = \begin{bmatrix} e_j \\ 0 \end{bmatrix}^T \begin{bmatrix} I - P & \pi^0 \\ e^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

and back-substitution yields

$$\begin{bmatrix} I - P & \pi^0 \\ e^T & 0 \end{bmatrix} \begin{bmatrix} x \\ h_j^{-1} \end{bmatrix} = -\begin{bmatrix} y e_j \\ 1 \end{bmatrix}.$$

or

$$h_j^{-1} = -\begin{bmatrix} 0 \\ 1 \end{bmatrix}^T \begin{bmatrix} I - P & \pi^0 \\ e^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} ye_j \\ 1 \end{bmatrix}.$$

Let B denote the matrix that appears in all these expressions after elimination of y, i.e.

$$B = \begin{bmatrix} I - P & \pi^0 \\ e^T & 0 \end{bmatrix}$$

Then define

$$f = B^{-1}e_{n+1}, \quad g = B^{-T}e_{n+1}$$

and observe that we have written h_j^{-1} as

$$h_j^{-1} = g_j f_j / (B^{-1})_{jj} - g_{n+1}.$$

Hence, we can compute the mean hitting time for *every* node in the network in terms of two linear solves with B and B^T and a diagonal computation for B^{-1}, which we might tackle directly or by using the stochastic diagonal estimator described before.

6 Between-ness centrality measures

Where closeness centrality is used to find nodes that are close to many other nodes, *betweenness centrality* measures are used to find nodes that appear in many paths between nodes. The most common form of betweenness centrality focuses on shortest paths. As in the case of closeness centrality, the expensive part of betweenness centrality computations is an all pairs shortest path computation. A method due to Brandes combines some steps to go in time proportional to the number of nodes times the number of edges; a later approximation scheme involving intermediate pivot or landmark vertices runs more quickly (in time proportional to the number of pivots times the number of edges).

Just as we can transition from geodesic distances to flow distances or random walk distances in closeness centrality, we can do the same for betweenness centrality. Similar techniques to those we saw with closeness centrality also apply for attempting to accelerate these betweenness centrality computations.