Exact Non-Iterative SimRank

Given two undirected graphs F and G and an initial similarity matrix S_0, SimRank propagates the similarity via the recurrence:

$$S = cF_{adj}^{\top} S G_{adj} + S_0$$

where c is a damping or decay factor between 0 and 1. Here the *adj* matrices are the column-normalized adjacency matrices of the graphs.

The recurrence can be rewritten as an infinite sum, i.e.,

$$S = \sum_{k=0}^{\infty} c^k F_{adj}^{\top k} S_0 G_{adj}^k$$

Because F and G are undirected, the adjacency matrices are symmetric which implies that they are diagonalizable. It follows that the column-normalized versions are also diagonalizable, because the normalization can be written as a product of the original symmetric matrix and a positive-definite diagonal matrix (assuming no orphaned nodes).

We eigendecompose the matrices as follows:

$$\begin{aligned} F_{adj}^{\top} &= P_F D_F P_F^{-1} \\ G_{adj} &= P_G D_G P_G^{-1} \end{aligned}$$

Solving,

$$\begin{aligned} S &= \sum_{k=0}^{\infty} c^k F_{adj}^{\top k} S_0 G_{adj}^k \\ &= \sum_{k=0}^{\infty} c^k (P_F D_F P_F^{-1})^k S_0 (P_G D_G P_G^{-1})^k \\ &= \sum_{k=0}^{\infty} c^k P_F D_F^k P_F^{-1} S_0 P_G D_G^k P_G^{-1} \\ &= P_F \left(\sum_{k=0}^{\infty} c^k D_F^k P_F^{-1} S_0 P_G D_G^k \right) P_G^{-1} \\ &= P_F \left(\sum_{k=0}^{\infty} (c \cdot eigs_F \cdot eigs_G^{\top})^k \odot (P_F^{-1} S_0 P_G) \right) P_G^{-1} \end{aligned}$$

where $eigs_F$ and $eigs_G$ are column vectors corresponding to the diagonals of D_F and D_G, respectively. Note that, because the adjacency matrices are column-normalized and $0 < c < 1$, we have $|c\lambda_F \lambda_G| < 1$ for any eigenvalue λ_F of F_{adj} and λ_G of G_{adj}.

Thus, the last equation contains the sum of an infinite geometric series, where $c \cdot eigs_F \cdot eigs_G^\top$ is the common ratio. Rewriting one last time, we arrive at the final form:

$$\begin{aligned} R &= c \cdot eigs_F \cdot eigs_G^\top \\ A &= P_F^{-1} S_0 P_G \\ S &= P_F (A \oslash (1 - R)) P_G^{-1} \end{aligned}$$

A Quick Note About Eigendecomposition

Because column-normalized adjacency matrices may no longer be symmetric, we cannot use the `np.linalg.eigh` routine. However, there's a trick we can apply.

Let S be the adjacency matrix in question. Then column-normalizing S is equivalent to calculating the product SD, where D is a diagonal matrix containing the inverse column sums. If we assume our graph has no orphaned nodes, then all diagonal entries of D are positive and thus D is positive definite. This implies an invertible square root of D exists.

We can then write:

$$D^{\frac{1}{2}}(SD)D^{-\frac{1}{2}} = D^{\frac{1}{2}} S D^{\frac{1}{2}} = Q$$

where Q is symmetric because $D^{\frac{1}{2}}$ and S are both symmetric. The above shows that Q is similar to SD, and thus they share the same eigenvalues. Moreover, we can now apply the `np.linalg.eigh` routine to Q because it is symmetric.

To recover the original eigenvectors, let v be an eigenvector of Q with eigenvalue λ. Then,

$$\begin{aligned} Qv = D^{\frac{1}{2}}(SD)D^{-\frac{1}{2}} v &= \lambda v \\ (SD)D^{-\frac{1}{2}} v &= \lambda D^{-\frac{1}{2}} v \\ (SD)w &= \lambda w \end{aligned}$$

which shows that $w = D^{-\frac{1}{2}} v$ is an eigenvector of SD with eigenvalue λ of Q.

Finally, we are interested in calculating the inverse eigenvectors of SD (P_F^{-1} and P_G^{-1} from previous section). Let B be the eigenvectors of Q. Because Q is symmetric, B is an orthogonal matrix.

Thus we can calculate the inverse eigenvectors of SD without calling `np.linalg.inv`:

$$\begin{aligned} (D^{-\frac{1}{2}} B)^{-1} &= B^{-1} D^{\frac{1}{2}} \\ &= B^\top D^{\frac{1}{2}} \end{aligned}$$