Notes about Function Prediction

Yiftach Kolb

April 1, 2021

1 RWR Methods

1.a go by rows or columns?

we have a graph $G(V, E)$ and we associate with it a transition matrix T of size $n \times n$. Now one thing to be carefule about is whether the rows or the columns of T are normalized.

If the rows are normalized, then $T_{i,j}$ represents the transition probability from i to j. If $p = (p_1, \dots p_n)$ is a row vector representing then $p \cdot T$ is the next probability in the process.

But for me at least, I prefer to use the standard matrix multiplication so we can transpose T and p: $A = T^t, u = p^t$, so now A is column normalized and u is a column vector and $A \cdot u$ is the next distribution of the process.

Anyway from now on lets assume by convention that a transition matrix is column-normalized and vectors are column vectors by default.

1.b RWR by matrix representation

So here we have a transition matrix T derived from the graph G. Let q a fixed distribution on the vertices $\{1 \dots n\}$ (column vector). Fix $\alpha \in [0, 1]$, representating the restart probability.

We can now define the RWR as the sequence of distributions p_i defined by:

- $p_0 := q$ (doesn't really matter what $p_0 \neq 0$ we choose, they all converge to the same, greatest eigenvector)
- $p_{k+1} := (1 - \alpha)T \cdot p_k + \alpha q$
  If we let Q be the matrix with all columns q, $Q := (q| \dots |q)$ then we may rewrite the transition step as:
  $p_{k+1} = [(1 - \alpha)T + \alpha Q] \cdot p_k$

This process converges to a stationary distribution $p = \lim p_k$. So we get

$$p = (1 - \alpha)T \cdot p + \alpha q \tag{1}$$

We can rewite this:

$$(I - (1 - \alpha)T) \cdot p = \alpha q. \tag{2}$$

The matrix $I - (1 - \alpha)T$ is invertible, so we can solve p from q and we get:

$$p = \alpha(I - (1 - \alpha)T)^{-1} q := K \cdot q \tag{3}$$

This is in my opinion a very important result. It means we choose a restart distibution and from it we get the stationary distribution of the corresponding RWR process.

Let say that we choose $q = (1, 0, \dots 0)$ Then $p = K \cdot q$ will be the stationary distribution for the RWR with restart to node 1 (with probability α. This is what propagating from node 1 means.

If we choose $q = (1/n \dots 1/n)$ then the corrsponding stationary $p = Kq$ is the pageRank distribution. so $p[i]$ (its i'th component) would be the pagerank of node i.

The matrix K from 3 is closely related to the graph Laplacian and in fact K^{-1} and this is why in the Laplacian we are interested in the smallest eigentvectors whereas in the transition matrices (K is also a transition matrice) we are interested in the largest eigenvalues. More on that in the next subsection.

1.c Relation to the Laplacian matriced and spectral clustering

This is an excerpt from the spectral clustering part which had been left out of my bachelor's thesis. I include it here because there is a very strong connection between graph laplacians and the diffusion matrix. We talked about how in the Laplacian we look for the smallest eigenvalues instead of the largest and the reason for that is very simple- the Laplacian's eigenvalues correspond to the diffusion matrix's (and to the orifinal transition matrix of G) in inverse order.

Definition 1.1. Let G be a bidirectional graph with adjacency matrix A and diagonal degree matrix D. Then the following matrices are its **graph Laplacian**, and its **symmetric—, row–normalized—and column–normalized—Laplacians**:

$$\begin{aligned} L &= D - A \\ L_s &= D^{-1/2} L D^{-1/2} = I - D^{-1/2} A D^{-1/2} \\ L_r &= D^{-1} L = I - D^{-1} A \\ L_c &= L D^{-1} = I - A D^{-1} \end{aligned} \tag{4}$$

Consider the connected graph G and its adjacency matrix A. We create the transition matrix by normalizing it: $T = AD^{-}1$, and finally we choose a restart parameter α and create the diffusion matrix $K = \alpha[I - (1-\alpha)T]^{-1}$. K and T have the same eigenvectors and the orders of their corresponding eigenvalues are the same. The eigenvalues are all real and therefore the eigenvectors are real as well. And the eigenvalues of K are all non-negative.

The matrix $K^{-1} = \alpha^{-1}[I - (1-\alpha)T]$ is closely related to the symmetric Laplacian L_s and thecolumn-normalized Laplacian L_c. The added parameter $0 < \alpha < 1$ neither changes the eigenvectors, nor the order of the corresponding eigenvalues.

The smallest eigenvalue ,0 of L and L_s corresponds to the largest eigenvalue 1 of and T and its eigenvector is (can be chosen as) all positive. Any other eigenvector of L (which corresponds to an eigenvector of T) must contain both positive and negative components (Perron-Froebenius).

1.d Using RWR to predict a function of an unlabled protein

Definition 1.2. Let $V = \{1 \dots n\}$ be the vertex set, let $\mathcal{L} = \{f_1, \dots f_p\}$ a set of labels.

A **partial (multi) labeling** of V is a function $\delta : V \times \mathcal{L} \to \{0, 1\}$ $v \in V$ has label $f \in \mathcal{L}$ iff $\delta(v, f) = 1$. This definition allows for vertex to have multiple labels.

We saw in 3 that for each restart distribution there is a corresponding stationary distribution of the RWR. The idea of propagation is to set for each v a restart distribution q_v (as a column vector), which then yields a stationary distribution $q_v = K \cdot q_v$ that we associate with vertex v.

We can use matrix notation: let $QR = (q_1 | \dots | q_n)$ be the matrix of restart distribution, such that column v of QR is q_v. The let $PR = K \cdot QR = (p_1 | \dots | p_n)$ (so the p_i's are **column vectors**), the corresponding matrix of stationary distributions, so that the first column is $p_1 = K \cdot q_1$ the stationary distribution we associate with vertex 1 etc.

The question is how to choose the right restart distribution for each v. In my bachelor thesis, I picked $q_v = e_v$ where e_v is that standard indicator vector (so it has 1 on coordinate v and is 0 otherwise). This means in the RWR process for v we restart from v with probability α. In the article of Zhang et. al, they defined a different q_v which is a distribution on all the vertices based on some similarity calculations they do which takes into account both graph topology (shared vs. unique neighbors) as well as protein domains comonality. They also didn't use the PPI graph itself for the propagation, rather they defined a correlation matrice based on that graph.

Now that we have for each vertex v an associated stationary distribution p_v, we need a way to claculate the label score for each label for each protein, and it is straitforward:

Definition 1.3. The score function s is the function $s : V \times \mathcal{L} \to [0, 1]$ defined by $s(v, f) = \sum_{i \in V} p_v[i] \cdot \delta(i, f)$

So $s(v, f)$ is the score of label f for vertex v, which in my bachelor's thesis I called the 'volume' but I now use the score term which I like more and is used by Zhang et. al.

In my thesis, for methods 1 and 2 I took $g_v = \text{argmax}\{s(v, f) | f \in (f)\}$ as the predicted function of v. In Zhang et. al. they pick the K largest argument as the predicted function set for v.

2 a note about the difference between scoring and ranking

Using the same notation as above, if we set the restart distribution as the uniform one, $q = (1/n, \ldots, 1/n)$, which means in our RWR we make every node equally likely to be restarted from. We obtain the corresponding stationary distribution $p = K \cdot q$ of this walk.

Now in PageRank, the vertices are then ranked according to their components in p. So the highest ranked vertex is $v = \text{argmax}\{p[i] | i \in V\}$. $p[i]$ is the frequency in which i is visited in the RWR, so v is the most frequently visited vertex. In the case of context search we would say that v is the top result and

In my bachelor thesis I used this principle for method 5. Basically suppose we have two disjoint subsets $A, B \subset V$. Now we set q_a to be the uniform distribution over members of A (and 0 outside A), and similarly with q_b for B.

we get two different RWR processes dependent whether we pick q_a or q_b for the restart, so we also get the two corresponding stationary distributions of these RWRs: $p_a = K \cdot q_a$ and $p_b = K \cdot q_b$.

Now given vertex $i \in V - (A \cup B)$, we can compare is i visited more frequently when we restart from q_a (a random member of A) or from B? $p_a[i] \square ? p_b[i]$ And according decide if i more likely belong to A or to B.

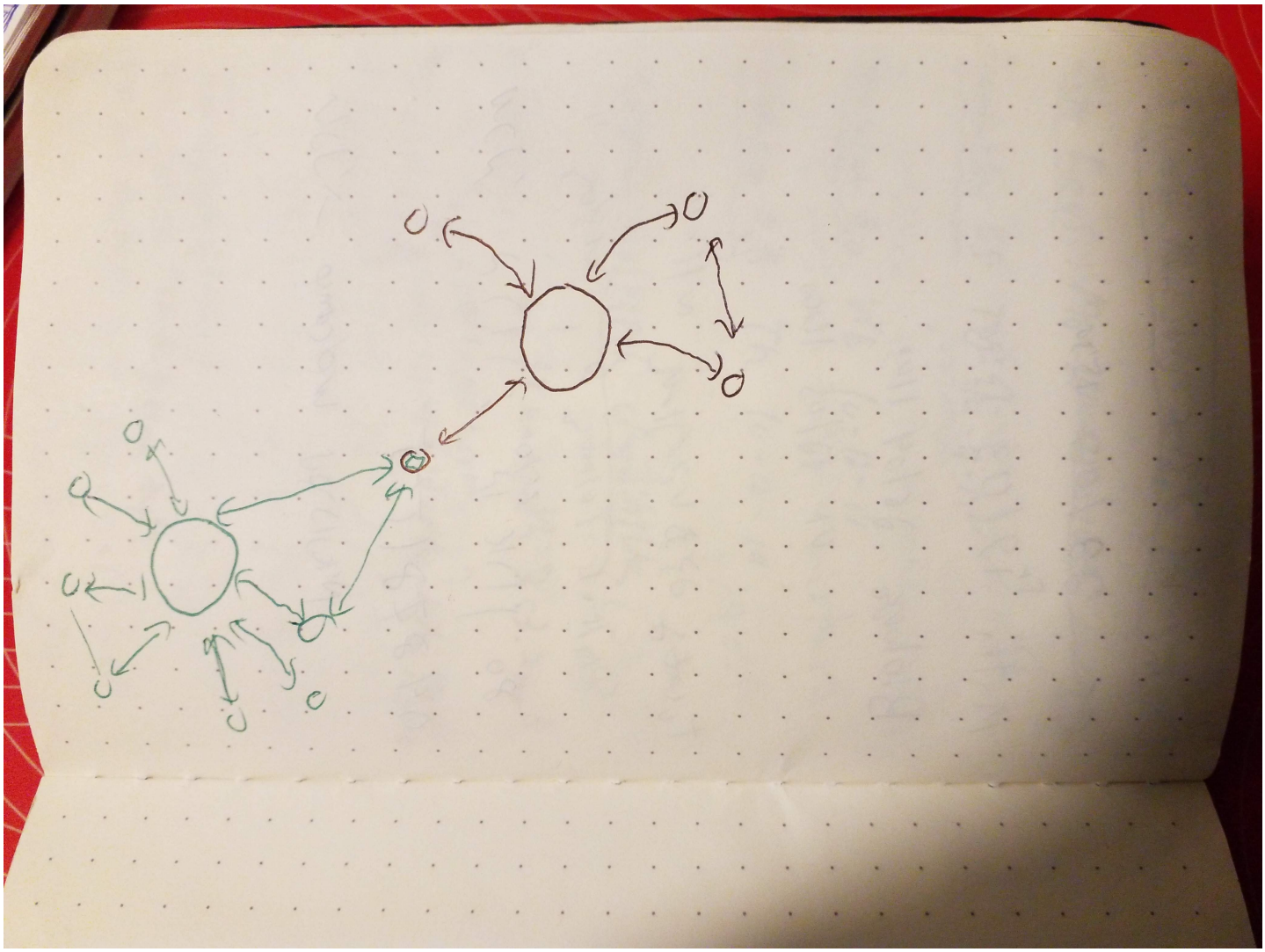

Figure 1: Example where ranking fails but scoring works

In figure **??** I try to demonstrate this scoring vs. ranking idea. We want to predict the function of the vertex in the middle, is it black or green? All the other vertices are known to be black or green.

If we do a random walk with uniform restart to a any green node, and a random walk with restart to any black node, then compare the frequencies in which the node in question is visited in each process, then we would probably find out that it is visited more frequently when we restart to the black group, simply because the black module is smaller than the green module so it is more likely to get out of it and visit the node in the middle.

If we do a restart to the green group, since this module is much larger we will probably visit the node in question less frequently even though it it is more connected to the green module than to the black module.

However, if we try to predict to which module the node in the middle

belongs to, by doing a RWR with restart to the node itself, then clearly in this process we visit a green node more frequently than a black node. Therefore the score of the greens (which is the sum of the frequencies of all the green nodes in this RWR) will be higher.