1. Bipartite Graph model

This is a ranking model for set $A = [1, 2, 3, \cdots, n]$, set $B = [1, 2, 3, \cdots, m]$ is said to be the "bonus" set (It can be can considered as the categorization of properties we concern about A: each $l \in B$ represents certain property of A). And for each edge incident with $l \in B$, that edge is then given by a bonus function$b(l)$ decided by users as weight.

First, we have our initial vector $\mathbf{r} = [r_1, \cdots, r_n]$, where f_i represents the resources or investment we input initially for the vertice $i \in A$ waited to be ranked. Usually it is set to be all 1's.

Second, we need to form the edges from A to B. Let a 3-tuple (i, l, w_{il}^a) represent the edge from $i \in A$ to $l \in B$ with the weight

$$w_{il}^a = \frac{c_{il}}{deg(i)}$$

meaning the ratio of the counts inside i having property of l over the total counts of i.

Third, form the edges from B back to A. Similarly, we have a 3-tuple (l, i, w_{li}^b) for the edge from l to i, where now w_2 is given by the bonus function $b(l)$.

Fourth, let the resource flow from A to B and then back to A:
The edges from A to B re-allocate our initial resource to B, the resource in B now is then blended and re-generated by the bonus function. As last step, the edges back to A then re-allocate the blended resource to each of the $i \in A$ again to get us the output vector $\mathbf{I_1} = [I_1, \cdots, I_n]$, which can then be viewed as the interest vector out of our initial investment. we can use this to make a rank of units in A.

The interest vector can be calculated by the following mathematical formula:

$$\mathbf{I} = W \cdot \mathbf{f} \tag{1}$$

The re-allocation matrix W is calculated by

$$W_{ij} = \frac{1}{\deg(j)} \sum_{l=1}^{m} \frac{b(l)}{\deg(l)} \cdot c_{il} c_{jl} \tag{2}$$

Due to the fact that $\deg(i)$ may be highly biased. There are some variations of (2) to balance this effect. One of them is

$$W_{ij} = \frac{1}{deg(i)} \frac{1}{\deg(j)} \sum_{l=1}^{m} \frac{b(l)}{\deg(l)} \cdot c_{il} c_{jl} \tag{3}$$

Take the "high school - GPA" pair as example.
All the different high schools are indexed by integers to form the set A. We form set B and the bonus function as below:

category	l	b(l)
GPA<2.0	1	-2
GPA$\in$ [2.0, 2.8)	2	0
GPA$\in$ [2.8, 3.3)	3	1
GPA$\in$ [3.3, 3.8)	4	2
GPA$\geq$ 3.8	5	4

The initial vector can be implemented as "how interest are we in recruiting students from each of the high school".

The blending effect in B can be then considered as the fact that even though they come from different schools, they are all learning in UW, there might be some connections between them.

The set B and bonus function corresponding with it then depend on "how shall we measure the performance of a student based on the GPA"

The numerical experiment shows the fact(Initial vector are set to be all ones):

- If we adopt (2), schools contributing more students will generally stand out at top of our ranking list(i.e. not balanced enough);
- If we adopt (3), schools contributing few but excellent students will generally stand out at top of our ranking list(i.e. maybe go too far in balancing).

It seems that we may consider other variations for the matrix to "appropriately " balance the effect of highly biased number of students of each high school.(Or just take "average" of the two scheme?)

I am currently considering another strategy:
The initial vector are not necessary all ones, but we require $\|\mathbf{f}\|$ to be a fixed number, the general norm are usually L^2 norm or just summing up all f_i's. Then it turns out to be an optimization problem:

$$\sup_{\|\mathbf{f}\|=F} \|\mathbf{I}\| \tag{4}$$

where F is a fixed positive number.

2. currently considered pairs

HighSchool-GPA, Major-GPA, College-GPA, GraduateTerm-GPA, Level-GPA

3. extension

GPA can be replaced by Financial aid to feed the model. It is much efficient to do the analysis if we have a totally combined file.