Calculating Author-Level Eigenfactors (Pseudocode)

Jevin West and Carl T. Bergstrom*

June 4, 2008

1 Overview

There are seven steps for calculating Author-Level Eigenfactors:

1. Data Input
2. Creating an Adjacency Matrix
3. Modifying the Adjacency Matrix
4. Identifying the Dangling Nodes
5. Calculating the Stationary Vector
6. Calculationg EigenFactor (EF) and ArticleInfluence (AI)
7. Outputting the Results

*Both authors are at the Department of Biology, University of Washington, Seattle WA 98115. If you have any questions, feel free to email Jevin at jevinw@u.washington.edu.

The following pseudocode can be used to calculate author-level Eigenfactors for scholars in the Social Science Research Network (SSRN). Eigenfactor measures the number of times that authors in the SSRN provide citations to other authors in the SSRN since the inception of this scholarly community.

1.1 Data Input

Four inputs — two files and two constants — are needed:

- Author File: this is the file that contains which author is citing which author and the total credit an author receives from a citing author. The file contains three columns. The first column is the author that is doing the citing (A1). The second column is the author receiving the citations (A2) from the author in the first column (A1). And the third column is the total, weighted credit (C) the author in column two has received from the author in column one since the inception of the SSRN. For each paper written by A1 that cites A2, the following information is needed: the number of references, the number of source authors (A1 being one of them), the number of target authors (A2 being one of them) and the number of versions of that paper in the SSRN. You add up all those numbers and then divide by one. You then do this for every paper from A1 citing a paper by A2. For example, the row in the file that contains (57, 108, 2.8) means that author (57) cited author (108) 2.8 times since the inception of SSRN.

- Article File: this is the file that contains the number of articles each author has in the SSRN database. This file has two columns. The first column is the author and the second column is the number of articles. There are two type of article files–weighted and non-weighted.

We typically are interested in using the weighted number of articles to calculate the Eigenfactor. For the weighted file, credit is divided among the authors of a multi-authored paper. For the non-weighted file, authors receive the full value of every paper they author, regardless of whether they are the solo author or a co-author. For example, if there were four authors on a paper, each author would get credit for 1/4 of a paper.

- Alpha ($\alpha = 0.85$)
- Epsilon ($\epsilon = 0.00001$)

1.2 Creating an Adjacency Matrix

The author citation network can be conveniently represented as an adjacency matrix $\mathbf{Z}$, where the $\mathbf{Z_{ij}}$-th entry indicates the number of times that author j cite author i since the inceptin of the SSRN. It can also be thought of as the amount of credit that author j gives to author i. Using the example above, an entry in the column 57 and row 108 would be 2.8. The dimension of this square matrix is n x n where n is the number of unique authors in the SSRN. For example, suppose there are authors A, B, and C.

	A	B	C
A	2	0	3
B	4	1	1
C	0	2	7

In the adjacency matrix above, author A cites itself 2 times, it cites author

B 4 times, and it doesn't cite author C at all. Author B receives 4 citations from author A, 1 citation from itself, and 1 citation from author C.

1.3 Modifying the Adjacency Matrix

There are some modifications that need to be done to $\mathbf{Z}$ before the eigenvectors can be calculated.

- First, we set the diagonal of $\mathbf{Z}$ to zero (i.e., we set all of the entries $Z_{ii} = 0$). This is done so that authors do not receive credit for self-citations.

- Second, we normalize the columns of the matrix $\mathbf{Z}$ (i.e., divide each entry in a column by the sum of that column). To do this, compute the column sums for each column j as $Z_j = \sum_i \mathbf{Z_{ij}}$. Then divide the entries from each column by the corresponding column sum to get the entries of the $\mathbf{H}$ matrix: $\mathbf{H_{ij}} = \mathbf{Z_{ij}}/Z_j$. There may be columns that sum up to zero (i.e., authors that cite no other authors). These are danlging nodes, and we will deal with them in the next section.

In the example below, we take an adjacency matrix through these two modifications. The matrix you get after these two modifications is $\mathbf{H}$. This example matrix will be used throughout the pseudocode as an example of how to calculate the EF of an author. The numbers in parentheses next to each author letter represent the number of papers that each author has published.

Example raw adjacency matrix (Z)

	A	B	C	D	E	F
$A(3)$	1	0	2	0	4	3
$B(2)$	3	0	1	1	0	0
$C(5)$	2	0	4	0	1	0
$D(1)$	0	0	1	0	0	1
$E(2)$	8	0	3	0	5	2
$F(1)$	0	0	0	0	0	0

1. Set the diagonal to zero

↓

	A	B	C	D	E	F
$A(3)$	0	0	2	0	4	3
$B(2)$	3	0	1	1	0	0
$C(5)$	2	0	0	0	1	0
$D(1)$	0	0	1	0	0	1
$E(2)$	8	0	3	0	0	2
$F(1)$	0	0	0	0	0	0

2. Normalize the columns. This matrix is H.

↓

	A	B	C	D	E	F
$A(3)$	0	0	2/7	0	4/5	3/6
$B(2)$	3/13	0	1/7	1	0	0
$C(5)$	2/13	0	0	0	1/5	0
$D(1)$	0	0	1/7	0	0	1/6
$E(2)$	8/13	0	3/7	0	0	2/6
$F(1)$	0	0	0	0	0	0

1.4 Identifying the Dangling Nodes

As mentioned in the previous section, there will be authors that don't cite any other authors. These authors are called dangling nodes and can be identified by looking for columns that contain all zeros. These columns need to be identified with a row vector of 1's and 0's. Call this vector d. The 1's indicate that a author is a dangling node; the 0's indicate a non-dangling node. For the example above, d would be the following row vector:

	A	B	C	D	E	F
d_i	0	1	0	0	0	0

1.5 Calculating the Influence Vector

The next step is to construct a transition matrix $\mathbf{P}$ and compute its leading eigenvector. This eigenvector, normalized so that its components sum to 1, will be called the influence vector π^*. This vector gives us the author weights that we will use in assigning eigenfactor scores.

To calculate the influence vector π^*, we need six inputs: the matrix $\mathbf{H}$ that we just created, an initial start vector $\pi^{(\mathbf{0})}$, the constants α and ϵ, the dangling node vector d and the article vector a.

Article Vector. Let A_{tot} be the total number of articles published by all of the authors. The article vector a is a column vector of the number of articles published by each author over the (five-year) target window, normalized so that its entries sum to 1. (To do this normalization, divide the number of

articles that each author publishes by A_{tot}). Using the example from above, $A_{\text{tot}} = 3 + 5 + 2 + 1 + 2 + 1 = 14$ and the article vector would be

Article Vector

	a_i
A	3/14
B	2/14
C	5/14
D	1/14
E	2/14
F	1/14

Initial start vector $\pi^{(\mathbf{0})}$. This vector is used in iterating the influence vector. Set each entry of this column vector to $1/n$. For our example, this vector would look like

	$\pi_{\mathbf{i}}^{(\mathbf{0})}$
A	1/6
B	1/6
C	1/6
D	1/6
E	1/6
F	1/6

Calculating the influence vector π^*. The influence vector π^* is the leading eigenvector (normalized so that its terms sum to one) of the matrix $\mathbf{P}$, defined as follows:[1]

$$\mathbf{P} = \alpha \mathbf{H}' + (1 - \alpha) a.e^T,$$

Here e^T is a row vector of all 1's and $a.e^T$ is thus a matrix with identical columns a. The matrix $\mathbf{H}'$ is the matrix $\mathbf{H}$, with all columns corresponding to dangling nodes replaced with the article vector a. In the example, $\mathbf{H}'$ would be the following matrix (notice the replacement of the B column):

	A	B	C	D	E	F
$A(3)$	0	3/14	2/7	0	4/5	3/6
$B(2)$	3/13	2/14	1/7	1	0	0
$C(5)$	2/13	5/14	0	0	1/5	0
$D(1)$	0	1/14	1/7	0	0	1/6
$E(2)$	8/13	2/14	3/7	0	0	2/6
$F(1)$	0	1/14	0	0	0	0

Because $\mathbf{P}$ will be an irreducible aperiodic Markov chain by construction, it will have a unique leading eigenvector by the Perron-Frobenius theorem. We could compute the normalized leading eigenvector of the matrix P directly using the power method, but this involves repeated matrix multiplication operations on the dense matrix $\mathbf{P}$ and thus is computationally

[1] This matrix describes a stochastic process in which a random walker moves through the scientific literature; it is analogous to the "google matrix" that Google uses to compute the PageRank scores of websites. The stochastic process can be interpreted as follows: a fraction α of the time the random walker follows citations and a fraction $1 - \alpha$ of the time the random walker "teleports" to a random author chosen at a frequency proportional to the number of articles published.

intensive. Instead, we can use an alternative approach that involves only operations on the sparse matrix $\mathbf{H}$ and thus is far faster[2]. To compute the influence vector rapidly, we will iterate the following equation

$$\pi^{(k+1)} = \alpha\, \mathbf{H}\, \pi^{(k)} + [\alpha\, d.\pi^{(k)} + (1 - \alpha)]a$$

This iteration will converge uniquely to the leading eigenvector of $\mathbf{P}$, normalized so that its terms sum to 1. To find this eigenvector, iterate repeatedly. After each iteration, check to see if the residual ($\tau = \pi^{(k+1)} - \pi^{(k)}$) is less than ϵ. If it is, then $\pi^* \approx \pi^{(k+1)}$ is the influence vector. Typically, this does not take more than 100 iterations with $\epsilon = 0.00001$. Using the raw adjacency matrx example above and the corresponding article vector, the stationary vector converges after 16 iterations to the following vector with $\alpha = 0.85$ and $\epsilon = 0.00001$:

	$\pi^*_{\mathbf{i}}$
A	0.3040
B	0.1636
C	0.1898
D	0.0466
E	0.2753
F	0.0206

[2]Notice that the equation below uses the matrix $\mathbf{H}$, without the dangling node columns replaced, not the matrix $\mathbf{H}'$. In fact, one does not need to ever construct the matrix $\mathbf{H}'$ in the process of doing these calculations.

1.6 Calculationg Eigenfactor ($\mathbf{EF_i}$) and Article Influence ($\mathbf{AI_i}$)

The vector of eigenfactor values for each author is given by the dot product of the H matrix and the influence vector π^*, normalized to sum to 1 and then multiplied by 100 to convert the values from fractions to percentages:

$$\mathrm{EF} = 100\,\frac{\mathbf{H}.\pi^*}{\sum_i [\mathbf{H}.\pi^*]_i}$$

The Eigenfactor values for our example are thus

	EF_i
A	34.0510
B	17.2037
C	12.1755
D	3.6532
E	32.9166
F	0.0000

The ArticleInfluence $\mathbf{AI_i}$ for each author (i) is calculated using the following equation:

$$\mathbf{AI_i} = 0.01 \frac{\mathbf{EF}_i}{a_i}$$

where $\mathbf{EF_i}$ is the Eigenfactor for author i and a_i is the normalized article vector. In words, the Article Influence is essentially the Eigenfactor/100, divided by the fraction of all articles that each author has published. The Article Influence values for our example are

	AI_i
A	1.5890
B	1.2043
C	0.3409
D	0.5114
E	2.3042
F	0.0000

1.7 Outputting the Results

To get the author rankings, just sort in descending order the **EF** and **AI** vectors. Output the results in whatever format is easiest to compare rankings. Right now, we are using Excel. The following is what we include in our data output:

- Author Last Name
- Author First Name
- Author Middle Name
- Eigenfactor
- Eigenvector
- ArticleInfluence
- Total Articles Authored
- Citations Out
- Citations In