Krylov Subspace Methods to Calculate PageRank

B. Vadala-Roth

REU Final Presentation
August 1st, 2013

How does Google Rank Web Pages?

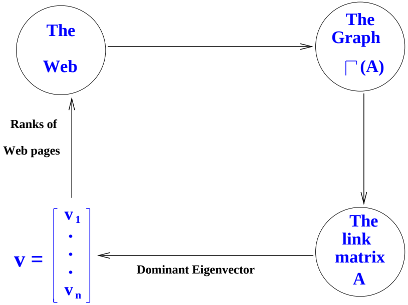

Figure : Diagram of the Page ranking process

Basics of the Model

Consider the web as a directed graph.

Definition

Link Matrix: Given a *directed* graph $\Gamma(A)$ with nodes $P_1, P_2, \ldots, P_n$, the adjacency matrix A_{nxn} has an entry in $a_{ij} \iff P_i$ has a link to P_j. In the context of the web, the value given to each entry will be $\frac{1}{r}$, where r is the number of outlinks from a node P_i, and we will call this matrix the **Link Matrix H**.

Remark : This Link Matrix is (nearly) row stochastic.

Stochasticity

Definition

Row Stochastic: A_{nxn} is **row Stochastic** if it is nonnegative and the entries of each row add up to 1.

Theorem

If A_{nxn} is a stochastic matrix, then its eigenvalues all have the property:

$$|\lambda_i| \leq 1, \quad i = 1, 2, \ldots \tag{1}$$

and $\lambda = 1$ is always an eigenvalue.

How does this Relate to Web Searches?

An ideal ranking would consider both the incoming and outgoing links of a webpage.

To formalize this concept, let's denote the rank of a webpage P_j by $r(P_j)$, then

$$r(P_j) = \sum_{P_i \epsilon B_{P_j}} \frac{r(P_i)}{|P_i|} \tag{2}$$

where B_{P_j} is the set of all pages leading to P_j and $|P_i|$ is the number of links coming from P_i.

Therefore, to calculate the ranking of one webpage, one must know the rankings of all the pages that link to it. In other words, a vector of page rankings multiplied by *row j* of the Link matrix should give the ranking of page j.

The Power Method

To quickly calculate this vector, an eigenvector with eigenvalue one, we will initially consider the Power Method.

Theorem

If a matrix A_{nxn} has a dominant eigenpair and n linearly independent eigenvectors, then for an arbitrary x_0 the iteration $x_{k+1} = Ax_k$ $(k = 0,1,2, \ldots)$ converges to the dominant eigenvector with a rate of convergence $q = \frac{|\lambda_2|}{|\lambda_1|}$.

Note : When A is a stochastic matrix, the steps of the iteration form a Markov Chain and the dominant eigenvector is the stationary distribution vector.

Background Definitions

Definition

Dominant Eigenpair: Given an nxn matrix A, a **dominant eigenvalue** is an eigenvalue λ_1 of A that satisfies the following inequality:

$$|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_m| \tag{3}$$

where $m \leq n$. A **dominant eigenvector** is the associated vector of the dominant eigenvalue.

Definition

Left Eigenvector: A vector $x \neq 0$ such that $x^T A = \lambda x^T$, or equivalently, the right eigenvector of A^T . In any case, A and A^T share eigenvalues.

From Graphs to Matrices

Definition

Strongly Connected: A directed graph $\Gamma(A)$ is **Strongly Connected** if there exists a finite path leading from one node to any other node in the graph.

Definition

Irreducible: We may call a matrix A_{nxn} **irreducible** if these two equivalent conditions are met:

- The associated graph $\Gamma(A)$ is Strongly Connected.
- There exists a permutation matrix P_{nxn} and an integer $1 \leq r \leq n-1$ such that $P^TAP = \begin{bmatrix} C & D \\ 0 & E \end{bmatrix}$, where 0 is an (n-r) by r block matrix.

If a matrix is not irreducible, then it is **reducible**

Perron-Frobenius Theorem

Theorem

Given A_{nxn}, an irreducible and nonnegative matrix, then:

1. *$\rho(A)$, denoting the Spectral Radius of A, is a positive (real) eigenvalue of A.*
2. *There exists a positive eigenvector associated to $\rho(A)$.*
3. *$\rho(A)$ has algebraic and geometric multiplicity one.*

Remark : This theorem has many equivalent and nearly equivalent phrasings, and this is only one of them.

Problems

Example

Given an irreducible matrix $A = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$. Then eigenvalues are $\{1, -1, \pm i\}$, and there is a largest real eigenvalue.

However, in this example there is no dominant eigenvalue, and that is exactly what we need.

Primitive Matrices

Definition

Primitive Matrix: A nonnegative matrix A_{nxn} is **Primitive** $\iff \quad A^k > 0, \quad \textit{for} \ k \geq 1$

Theorem

A primitive matrix is irreducible and has only one eigenvalue of largest magnitude.

Therefore, our inequality is met.

Stochastic, Irreducible, Primitve Matrices

If you haven't noticed, not every matrix is Stochastic, Irreducible, and Primitive.

$$\begin{aligned} B &= H + au^T,\ \textit{where u is our Personalization Vector and} \\ a_i &= \begin{cases} 1 & \textit{if page i is a dangling node} \\ 0 & \textit{otherwise} \end{cases} \end{aligned} \tag{4}$$

This adds Probability row vectors to rows with all zeros. The matrix is now a stochastic approximation of the original.

Note : The Personalization Vector u can be any probability vector used to assign heavier weight to certain sites. It may also be $\frac{e_n}{n}$, in which case each entry would be very small if n is large.

Making Our Matrix Irreducible and Primitive

By taking steps to make the matrix irreducible, we also make it primitive.

$$G = \alpha B + (1-\alpha) e_n u^T, \quad \textit{where alpha is the damping factor}. \tag{5}$$

The new matrix G, our "Google Matrix," is now irreducible and primitive in addition to being stochastic.

This means that $\lambda_1 = 1$ will be a unique largest eigenvalue and its corresponding eigenspaces is 1-dimensional.

Our Damping Factor α

Theorem

Denote the spectra of B and G with $\sigma(B) = \{\mu_1, \mu_2, \ldots, \mu_n\}$ *and* $\sigma(G) = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$. *Then*

$$\lambda_k = \alpha\mu_k \quad \textit{for} \;\; k = 2, 3, \ldots$$

Therefore, the separation of λ_1 and λ_2 will be greater, and the rate of convergence will be faster.

Remark : A damping factor closer to 1 indicates a more accurate approximation of the Link Matrix H, but a slower rate of convergence.

Other Methods

The dominant eigenvector that we are approximating is referred to as the PageRank vector.

A common method to solving the PageRank problem is by using a numerical method called an Arnoldi Process to construct an approximation of a subspace spanned by the largest eigenvectors of the Link Matrix. We call these subspaces Krylov Subspaces.

But first we must review some linear algebra.

Orthogonal Matrices & QR Factorizations

Theorem

Given matrix A_{mxn} with n linearly independent columns, there exists a factorization

$$A = QR$$

where Q_{mxn} is an orthogonal matrix and R_{nxn} is an invertible triangular matrix. The columns of Q form an orthonormal basis for col(A).

Facts About Orthogonal Matrices :

1. Q_{mxn} is orthogonal iff $Q^TQ = I$: its left inverse is Q^T.
2. Given Q_{nxn} and $x \epsilon \mathbb{R}^n$, then $||Qx||_2 = ||x||_2$.

Modifications for Arnoldi Methods

Definition

Hessenberg Matrix: A matrix A_{mxn} is upper Hessenberg if for all $i > j+1$, $a_{ij} = 0$. A matrix is unreduced upper Hessenberg if for all $i = j+1$, $a_{ij} \neq 0$.

Example

$H = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 0 & 8 \\ 0 & 9 & 1 & 2 \\ 0 & 0 & 4 & 5 \end{bmatrix}$ is a square, unreduced upper Hessenberg matrix. $G = \begin{bmatrix} 1 & 1 & 2 & 3 \\ 5 & 8 & 13 & 21 \\ 0 & 0 & 34 & 55 \\ 0 & 0 & 89 & 144 \\ 0 & 0 & 0 & 233 \end{bmatrix}$ is a rectangular, reduced upper Hessenberg matrix.

Eigenspaces

Definition

Let A be of order n and let χ be a subspace of $\mathbb{C}^n$. Then χ is an **Eigenspace** or **Invariant Subspace** of A if

$$A\chi \equiv \{Ax : x \epsilon \chi\} \subset \chi. \tag{6}$$

Theorem

Let υ be a subspace and let U be a basis for υ. Let V be a left inverse of U and set

$$B = VAU. \tag{7}$$

If $\chi \subset \upsilon$ is an eigenspace of A, there is an eigenpair (L,W) of B, such that (L,UW) is an eigenpair of A with col(UW) $= \chi$.

We call (7) the **Rayleigh Quotient**.

Rayleigh-Ritz Procedure

If the space υ contains an approximate eigenspace $\tilde{\chi}$ of A, there would be an eigenpair $(\tilde{L}, \tilde{W})$ of B such that $(\tilde{L}, U\tilde{W})$ is an approximate eigenpair of A with col(U$\tilde{W}$) $\cong \tilde{\chi}$.

Definition

Let μ be a Ritz value (eigenvalue approximation) associated to υ. A **Refined Ritz Vector** is the solution of the following problem:

$$\min_{\substack{x \in \upsilon \\ ||x||=1}} ||Ax - \mu x|| \tag{8}$$

Solving for the Refined Ritz Vector

Equivalently, we may write (8) as:

$$\min_{\substack{x \in \kappa_k \\ ||x||=1}} ||(A - \mu I)x|| \tag{9}$$

The solution to such problem is the right singular vector corresponding to the smallest singular value in the SVD factorization of $(A - \mu I)$.

We will now define SVD.

Singular Value Decompositions

Theorem

Given a matrix A_{mxn} there exist orthogonal matrices U_{mxm} and V_{nxn} and a diagonal matrix Σ_{mxn} such that $A = U\Sigma V^T$ and the following conditions hold:

1. *The columns of U are the eigenvectors of AA^T.*
2. *The columns of V are the eigenvectors of A^TA.*
3. *Let λ_i be the eigenvalues of AA^T or $(A^TA\ )$. The entries σ_i along the principal diagonal of Σ are the* ***singular values*** *of A. That is, $\sigma_i = \sqrt{\lambda_i}$, $i = 1, 2, \ldots p$, $p = min\{m, n\}$.*

We call this the Singular Value Decomposition.

Krylov Subspaces

Definition

Let A be *nxn* and $u \neq 0 \epsilon \mathbb{R}^n$. Then:

- We call the sequence $u, Au, A^2u, \ldots, A^{k-1}u$ the **Krylov Sequence of order k**.
- We call the matrix $K_k(A,u) = (u \; Au \; A^2u \; \cdots \; A^{k-1}u)$ the **kth Krylov Matrix**.
- We call $\kappa_k(A,u) = \mathrm{col}[K_k(A,u)]$ the **kth Krylov Subspace**.

For the sake of brevity, we will omit A and u from our references and use simply K_k or κ_k.

Properties of Krylov Subspaces

Theorem

Let A and $u \neq 0$ be given. Then:

1. $\kappa_k \subset \kappa_{k+1}$ *and* $A\kappa_k \subset \kappa_{k+1}$
2. *If* (λ, u) *is an eigenpair of* A*, then* $A^k u = \lambda^k u$ *and* $\kappa_k = \kappa_1$ $(k = 1, 2, \ldots)$.
3. *We say that the Krylov Sequence* **terminates** *at* ℓ *if* $\kappa_{\ell+1} = \kappa_\ell$.

Arnoldi Decomposition

Definition

Let $U_{k+1} \epsilon \mathbb{C}^{nx(k+1)}$ be orthogonal and let U_k consist of the first k columns of U_{k+1}. If there is a (k+1) by k unreduced upper Hessenberg matrix $\hat{H}_k$ such that

$$AU_k = U_{k+1}\hat{H}_k, \tag{10}$$

then we call (10) an **Arnoldi Decomposition of order k**. If $\hat{H}_k$ is reduced, we say the Arnoldi decomposition is **Reduced**.

Note: (10) can also be written as $AU_k = U_k H_k + \beta_k u_{k+1} e_k^T$, where $\beta_k = h_{k+1,k}$

Arnoldi Process

Given any starting vector q_0, we may construct an orthonormal set that spans a Krylov subspace of order $k+1$ for matrix A by doing the following:

$$\begin{aligned} q_1 &= \tfrac{q_0}{||q_0||} \\ \tilde{q}_2 &= Aq_1 \; - ((Aq_1)^T q_1)q_1, \;\; q_2 = \tfrac{\tilde{q}_2}{||\tilde{q}_2||} \\ &\vdots \\ \tilde{q}_{k+1} &= Aq_k \; - \sum_{i=1}^{k}(Aq_k^T q_i)q_i, \;\; q_{k+1} = \tfrac{\tilde{q}_{k+1}}{||\tilde{q}_{k+1}||} \end{aligned} \tag{11}$$

Putting it All Together

How do we use Krylov Subspaces to find an approximation to the dominant eigenvector?

Firstly, let Q_{k+1}, produced from the Arnoldi process, denote our Krylov Matrix, whose left inverse is Q_{k+1}^T

Denote the Google matrix as A. Approximating an eigenvector of the dominant eigenvalue is equivalent to minimizing:

$$||Ax - \lambda_1 x||, \ \textit{subject to } x \ \epsilon \ \kappa_k, ||x||_2 = 1. \tag{12}$$

When $\lambda_1 = 1$, (12) becomes much simpler and bares a striking resemblance to a Refined Ritz Vector calculation.

Simplifying Some More

As a result of the relationship $AQ_k = Q_{k+1}H_{k+1,k}$ we also have the following:

$$\begin{aligned}
Q_k^T A Q_k &= Q_k^T(Q_{k+1}H_{k+1,k}) \\
Q_k^T A Q_k &= Q_k^T(Q_k H_{k,k} + \beta_k q_{k+1} e_k^T) \\
Q_k^T A Q_k &= Q_k^T Q_k H_{k,k} + Q_k^T \beta_k q_{k+1} e_k^T \\
Q_k^T A Q_k &= H_{k,k}
\end{aligned} \tag{13}$$

The matrix H_k is significantly smaller than Q_k or A, and we also know from this form that if (λ_i, v_i) is an eigenpair of H_k, then $(\lambda_i, Q_k v_i)$ is an approximate eigenpair of A.

Our new estimate to the dominant eigenvector is created by taking the SVD of $(H_k - I)$.

Generalized Arnoldi Method

Generalizing the Arnoldi Process, we can try to make our algorithm a lot faster.

The idea is to generalize the dot product used to create the Krylov Subspace.

Definition

SPD: A symmetric matrix A_{nxn} is **Symmetric Positive Definite** if $x^TAx > 0$ for all vectors $x \neq 0$

Definition

G - Inner Product: Let G ϵ $\mathbb{R}^{nxn}$ be an SPD matrix and let x,y ϵ $\mathbb{R}^n$, then a $\mathbf{G-}$ **Inner Product** is

$$(x,y)_G = x^TGy \tag{14}$$

G - Orthonormality

The G-norm is defined as

$$||u||_G = \sqrt{(x,x)_G} \tag{15}$$

Theorem

Given a set of k vectors $\tilde{q}_i$ such that the G - Inner Product is zero when any two are multiplied and the G - norm of each equals one, we call the set a G - orthonormal set of vectors.

Denote $\tilde{Q}_k = \{\tilde{q}_1 \ \tilde{q}_2 \ \dots \ \tilde{q}_k\} \ \epsilon \ \mathbb{R}^{nxk}$, then

$$\tilde{Q}_k^T G \tilde{Q}_k = I \tag{16}$$

Changes to the Algorithm

We now have,

$$\begin{aligned} A\tilde{Q}_k &= \tilde{Q}_k \tilde{H}_{k,k+1} \\ \tilde{Q}_k^T G A \tilde{Q}_k &= \tilde{H}_k \end{aligned} \tag{17}$$

So if (λ_i, v_i) is an eigenpair of $\tilde{H}_k$, then $(\lambda_i, \tilde{Q}_k v_i)$ is an eigenpair of A.

Additionally, the residual $r = Aq - q = \tilde{\sigma}_k \tilde{Q}_{k+1} u_k$, where u_k is the left singular vector associated to the singular value $\tilde{\sigma}_k$.

Choosing a Matrix G

An initial choice for G was a diagonal matrix $G_0 = diag\{\frac{|r_i|}{||r||_1}\}$, where r is the residual defined above.
This choice is not necessarily the fastest nor is it necessarily faster than the regular Arnoldi Method.

Other choices include:

$$\begin{aligned} G_1 &= diag\{|r_i|\} \\ G_2 &= diag\{\frac{1}{|r_i|}\} \end{aligned} \tag{18}$$

Stanford Matrix, n = 281,903 and Stanford-Berkeley Matrix, n = 685,230 ($tolerance = 1 * 10^6$)

α	*Power Method*	*Arnoldi*	G_0	G_1
.85	5.49	3.60	4.98	4.42
.90	8.30	5.39	6.75	6.07
.95	16.82	8.07	10.78	9.80
.99	81.82	30.59	27.64	25.04
.998	412.47	73.85	59.02	47.76

α	*Power Method*	*Arnoldi*	G_0	G_1
.85	8.76	6.30	8.75	8.24
.90	13.18	9.45	12.91	12.37
.95	26.89	17.58	23.70	22.97
.99	138.11	78.20	96.96	100.76
.998	681.05	228.60	321.19	310.65

Table : Method Comparison

K = 7

Matrix Reorderings

Other methods of speeding up calculation of the PageRank vector include inserting iterations of the Power Method at various points in the code and reordering the original matrix. We found reorderings more useful.

Possible reorderings are:

- Reordering each row and column by nonzero count
- Reordering by dangling nodes
- Reverse Cuthill-McKee Reordering

Why do Reorderings Speed Up Calculations?

Not much is known about why reordering a matrix can reduce the computational complexity of certain matrix calculations.

Conjecture

Matrix reorderings that speed up the PageRank calculation, or equivalently, the calculation of the stationary distribution of the Markov chain, further separate the dominant eigenvalue from the subdominant eigenvalue.

Remember that the rate of convergence is: $q = \frac{|\lambda_2|}{|\lambda_1|}$, so further separation (in addition to that already due to the damping factor α) should speed up the convergence.

Dangling Nodes

Dangling Nodes are the pages that are linked to by other pages but have no outlinks themselves. In our model, these correspond to rows of zeros, meaning that some of the row vectors of our matrix will be identical.

This suggests that the Markov Chain is lumpable. Other attempts at computing PageRank more quickly have made use of reorderings that exploit this trait.

Lumpability

Definition

Lumpability Let $A \epsilon \mathbb{R}^{nxn}$ be the transition probability matrix for a Markov Chain of n states. Let $S_1, S_2, \ldots, S_p$ be disjoint subsets of states whose union contains all the states of the Markov Chain. Then the Markov Chain is lumpable if for all partitions $S_n \, and \, S_m$ and states $i, i' \epsilon S_n$

$$\sum_{j \epsilon S_m} P(i,j) = \sum_{j \epsilon S_m} P(i',j), \tag{19}$$

where $P(i,j)$ denotes the probability of going from state i to state j.

Uses of Lumpability

- Lumping the dangling nodes together reduces the number of states of the Markov Chain.
- A solution to the PageRank problem proposed by Lee, Golub and Zenio involves the Power Method and Lumpability.
- An analogous method proposed by Langville and Meyer also uses Lumpability.
- Both these methods involve using a matrix of order $card(S_{ND}) + 1$.

Effect of Reordering

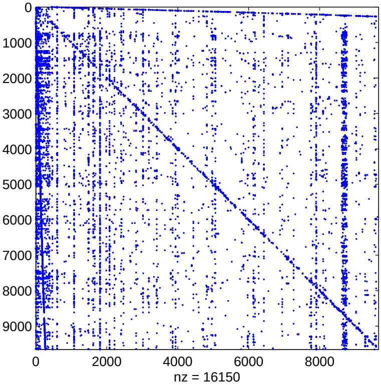

Figure : "California" Matrix, n= 9664

Effect of Reordering

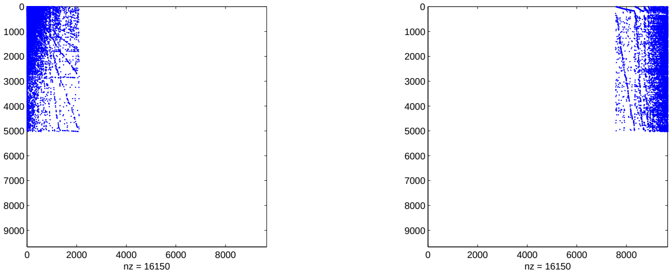

Figure : Reordering by nonzeros Method A and Method B

Some Preliminary Findings

On randomly generated matrices of order 200, we found that reordering by nonzeros reduced the absolute value of the subdominant eigenvalue most of the time. Here are results for two reorderings.

α	Decreases	Average Change	Average Percent Change
.85	75.40%	-.0493	-5.97%
.95	78.60%	-.0727	-8.10%
.99	76.60%	-.0831	-8.56%

α	Decreases	Average Change	Average Percent Change
.85	71.40%	-.0530	-6.06%
.95	72.80%	.0685	-7.24%
.99	72.60%	-.0659	-6.53%

Table : Decreases in Subdominant Eigenvalue

Reordering by Nonzeros vs Other Arnoldi Methods

α	Arnoldi	It.	Adaptive Arnoldi	It.	Reordered Arnoldi	It.
.85	3.7278	12	3.5950	8	.7285	3
.90	5.5164	18	5.2855	12	.7524	3
.95	9.6857	31	8.3817	19	0.7231	3
.99	38.1776	126	20.1734	45	0.9803	4
.85	2.7272	7	2.8631	5	1.1732	3
.90	3.5000	9	3.4557	6	1.5193	4
.95	4.6826	12	4.565	8	1.5263	4
.99	12.4978	32	9.6114	16	1.5442	4
.85	6.7539	13	6.2199	8	2.9065	5
.90	9.8673	19	10.1653	13	3.4734	6
.95	18.7197	36	19.1228	24	3.4903	6
.99	82.5691	157	64.4411	80	4.0604	7

Table : Stanford Matrix ($n = 281803$), Amazon Matrix ($n = 410236$), Stanford-Berkley Matrix ($n = 685230$)

Accelerated Methods

Inserting iterations of the Power Method after each iteration of the Arnoldi process yields:

$$x^i_{Acc\ I} = \frac{A^d x^i}{||A^d x^i||_2} \tag{20}$$

after the *ith* iteration, where x^i is the approximate eigenvector produced from the Arnoldi process and d is a chosen number of iterations for the power method (e.g. d $=$ 10).

Another Accelerated Method

Another accelerated Arnoldi Method, proposed by Wu, Zhang, and Wei, solves the following minimization problem:

$$q = \min_{y \in \mathbb{C}^k} ||A(A^{d-1}x^i + Q_k y) - (A^{d-1}x^i + Q_k y)||_2 \tag{21}$$

And this is used find the next iterate:

$$x^i_{Acc\ II} = \frac{A(A^{d-1}x^i + Q_k q)}{||A(A^{d-1}x^i + Q_k q)||_2} \tag{22}$$

Reordering by Nonzeros vs Accelerated Methods

α	Arn (3)	It.	Arn (13)	It.	Acc (I)	It.
.85	7.3929	25	7.4948	4	3.4584	5
.90	11.2047	38	11.1594	6	5.3084	7
.95	21.9187	74	20.5443	11	9.7856	13
.99	95.7948	318	87.4143	44	41.141	53

α	Acc (II)	It.	Re (3)	It	Re (13)	It.
.85	3.4341	4	2.9544	9	4.0481	2
.90	5.0474	5	3.3044	10	4.2328	2
.95	9.2086	11	3.682	11	4.1995	2
.99	37.3108	44	3.9675	12	4.0997	2

Table : Stanford-Berkley Matrix ($n = 685{,}230$)

Reordering by Nonzeros vs Accelerated Methods (Cont.)

α	Arn (3)	It.	Arn (13)	It.	Acc (I)	It.
.85	30.3154	18	25.6454	3	18.8871	4
.90	43.865	26	35.3022	4	24.3876	5
.95	84.1615	52	42.3668	5	30.0845	6
.99	457.3376	271	70.54	8	126.0145	23
α	Acc (II)	It.	Re (3)	It	Re (13)	It.
.85	17.5136	3	9.3331	7	7.0687	1
.90	19.1123	4	9.0994	7	7.0620	1
.95	28.9946	5	9.1397	7	7.0252	1
.99	80.3484	14	9.0382	7	7.0634	1

Table : Wikipedia Matrix (n = 1,634,989)

Future and Current Problems

- Are there faster choices of G?
- Which values of k and alpha are optimal?
- Are certain methods better for matrices of certain sizes or densities?
- Can we precondition the Matrix or change it further?
- Why do matrix reorderings tend to speed up calculations?
- If so, why do all improvements make a difference?

Preconditioning

To reduce reduce the condition number in solving linear systems, one can use preconditioning. Instead of solving

$$Ax = b, \tag{23}$$

we can solve

$$MAx = Mb. \tag{24}$$

where $M = \tilde{L}\tilde{U}$ is an incomplete LU factorization. It is expected that when preconditioning is complemented with reordering, our algorithms will be even faster. However, some issues may arise.

Some References

[1] S. Brin, L. Page, R. Motwami, T. Winograd "The PageRank citation ranking: bringing order to the web." Technical Report 1999-0120, Stanford University (1999).

[2] C. Lee, G. H. Golub, S. Zenios "A Fast Two-Stage Algorithm for Computing PageRank and its Extensions." Scientific Computation and Computational Mathematics (2003).

[3] G. H. Golub, C. Greif "An Arnoldi-Type Algorithm for Computing Page Rank." BIT Numerical Mathematics (2006).

[4] A. Langville, C. D. Meyer "A Reordering for the PageRank Problem." SIAM Journal on Scientific Computing (2006).

[5] J. Yin, G. Yin, M. Ng "On Adaptively Accelerated Arnoldi Method for Computing PageRank." Numerical Linear Algebra with Applications 19:73-85. Wiley Online Library (2012).

[6] G. Wu, Y. Zhang, Y. Wei "Accelerating the Arnoldi-Type Algorithm for the PageRank Problem and the ProteinRank Problem." Journal of Scientific Computing (2013).