The $25,000,000,000 Eigenvector
The Linear Algebra Behind Google

Kurt Bryan

May 19, 2011

Searching the Web

A search engine like Google has to:

- Crawl the web and collect information from web pages.

Searching the Web

A search engine like Google has to:

- Crawl the web and collect information from web pages.
- Store this information in a suitable format.

Searching the Web

A search engine like Google has to:

- Crawl the web and collect information from web pages.
- Store this information in a suitable format.
- When queried, retrieve the relevant links and present them in some sensible order.

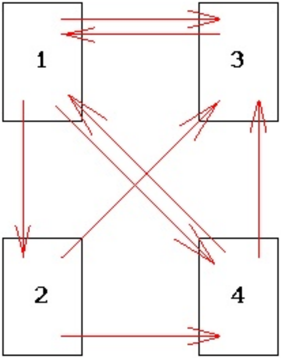
Searching the Web

A search engine like Google has to:

- Crawl the web and collect information from web pages.
- Store this information in a suitable format.
- When queried, retrieve the relevant links and present them in some sensible order.

An essential factor in this ordering is the importance of each web page. How should "importance" be computed?

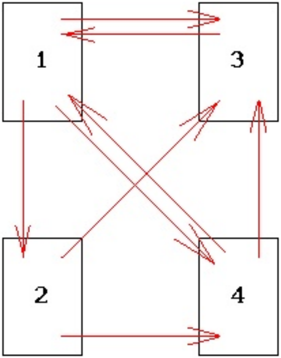
A Simple Idea



Let x_k = importance of page k.

Idea: $x_k = \#$ of backlinks for page k. Then

A Simple Idea

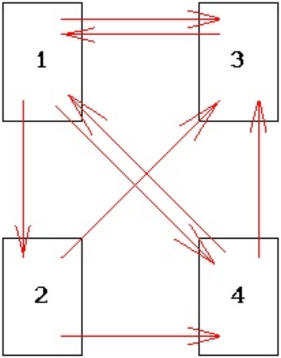


Let x_k = importance of page k.

Idea: $x_k = \#$ of backlinks for page k. Then

- $x_1 = 2$

A Simple Idea

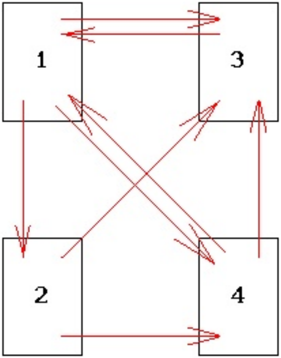


Let x_k = importance of page k.

Idea: $x_k = \#$ of backlinks for page k. Then

- $x_1 = 2$
- $x_2 = 1$ (least important)

A Simple Idea

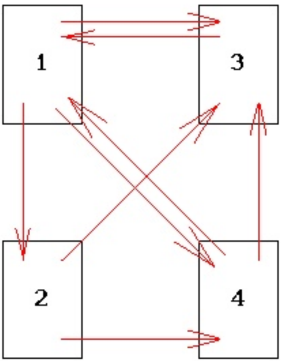


Let x_k = importance of page k.

Idea: $x_k = \#$ of backlinks for page k. Then

- $x_1 = 2$
- $x_2 = 1$ (least important)
- $x_3 = 3$ (most important)

A Simple Idea

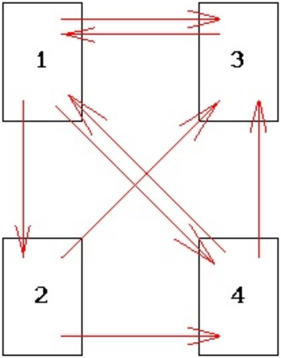


Let x_k = importance of page k.

Idea: $x_k = \#$ of backlinks for page k. Then

- $x_1 = 2$
- $x_2 = 1$ (least important)
- $x_3 = 3$ (most important)
- $x_4 = 2$

A Shortcoming

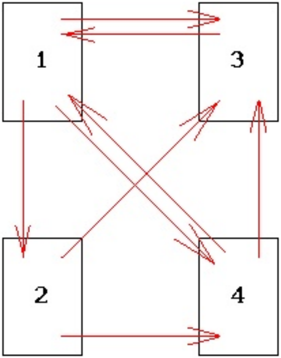


Backlink counting yields $x_1 = 2, x_2 = 1, x_3 = 3, x_4 = 2.$

But shouldn't $x_1 > x_4$?

Shouldn't a link from `Yahoo` count more than a link from `www.kurtbryan.com`?

A Shortcoming



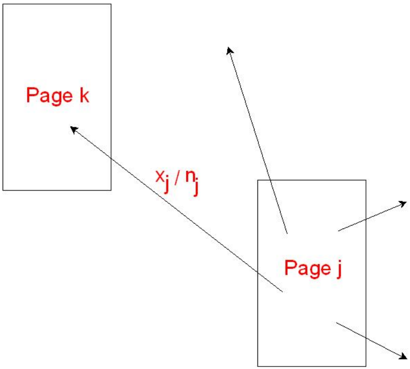
Backlink counting yields $x_1 = 2, x_2 = 1, x_3 = 3, x_4 = 2$.

But shouldn't $x_1 > x_4$?

Shouldn't a link from `Yahoo` count more than a link from `www.kurtbryan.com`?

Better Idea: Links from important pages should count more than links from less important pages.

An Improvement

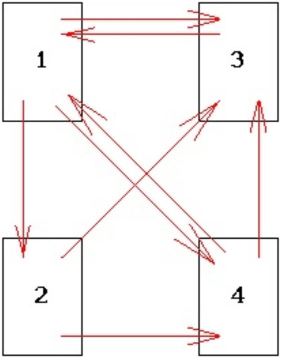


If page j links to page k then page j casts a vote for page k's importance, in amount x_j/n_j, where n_j is the number of links out of page j.

The importance score for any page is the sum of its votes from its backlinks.

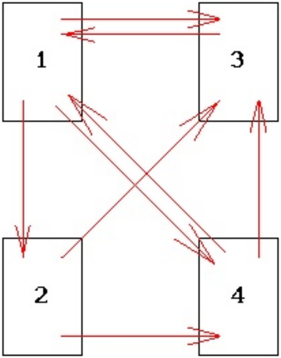
$$x_k = \cdots + x_j/n_j + \cdots$$

An Improvement



The importance score for any page is the sum of the votes from its backlinks:

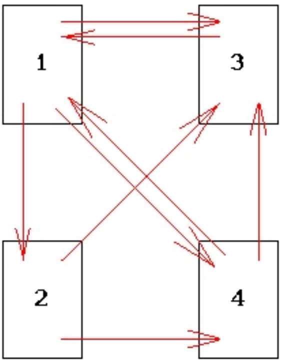
An Improvement



The importance score for any page is the sum of the votes from its backlinks:

$$x_1 = x_3/1 + x_4/2$$

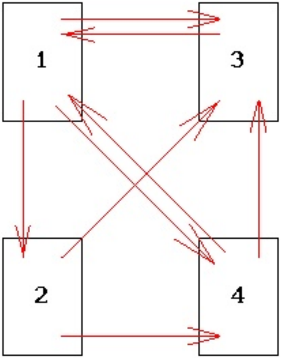
An Improvement



The importance score for any page is the sum of the votes from its backlinks:

$$\begin{aligned} x_1 &= x_3/1 + x_4/2 \\ x_2 &= x_1/3 \end{aligned}$$

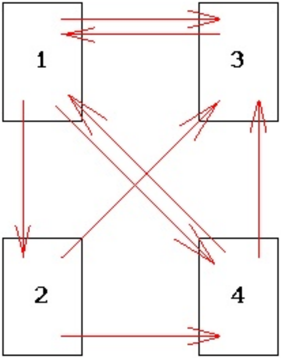
An Improvement



The importance score for any page is the sum of the votes from its backlinks:

$$
\begin{aligned}
x_1 &= x_3/1 + x_4/2 \\
x_2 &= x_1/3 \\
x_3 &= x_1/3 + x_2/2 + x_4/2
\end{aligned}
$$

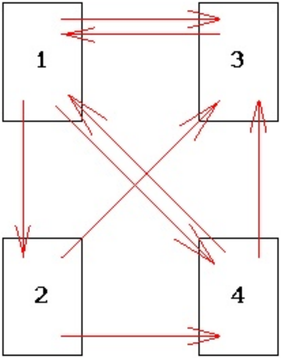
An Improvement



The importance score for any page is the sum of the votes from its backlinks:

$$
\begin{aligned}
x_1 &= x_3/1 + x_4/2 \\
x_2 &= x_1/3 \\
x_3 &= x_1/3 + x_2/2 + x_4/2 \\
x_4 &= x_1/3 + x_2/2
\end{aligned}
$$

The Matrix Form



If $\mathbf{x} = [x_1 \;\; x_2 \;\; x_3 \;\; x_4]^T$ then

$$\mathbf{x} = \mathbf{A}\mathbf{x}$$

where $\mathbf{A}$ is the link matrix

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 1/2 \\ 1/3 & 0 & 0 & 0 \\ 1/3 & 1/2 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix}$$

So $\mathbf{x}$ is an eigenvector for $\mathbf{A}$ with eigenvalue 1.

Eigenvectors

Recall that if $\mathbf{A}$ is an $n \times n$ matrix then a NONZERO vector $\mathbf{x}$ is an eigenvector for $\mathbf{A}$ if

$$\mathbf{A}\mathbf{x} = \lambda \mathbf{x}$$

for some scalar λ (the eigenvalue for $\mathbf{x}$).
"In general",

Eigenvectors

Recall that if $\mathbf{A}$ is an $n \times n$ matrix then a NONZERO vector $\mathbf{x}$ is an eigenvector for $\mathbf{A}$ if

$$\mathbf{Ax} = \lambda \mathbf{x}$$

for some scalar λ (the eigenvalue for $\mathbf{x}$).
"In general",

- An $n \times n$ matrix has n distinct eigenvalues $\lambda_1, \ldots, \lambda_n$, (some may be complex), and

Eigenvectors

Recall that if $\mathbf{A}$ is an $n \times n$ matrix then a NONZERO vector $\mathbf{x}$ is an eigenvector for $\mathbf{A}$ if

$$\mathbf{A}\mathbf{x} = \lambda \mathbf{x}$$

for some scalar λ (the eigenvalue for $\mathbf{x}$).
"In general",

- An $n \times n$ matrix has n distinct eigenvalues $\lambda_1, \ldots, \lambda_n$, (some may be complex), and
- n corresponding linearly independent eigenvectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$

Eigenvectors

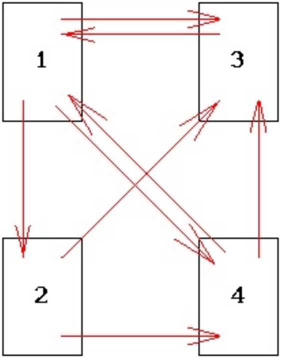
Recall that if $\mathbf{A}$ is an $n \times n$ matrix then a NONZERO vector $\mathbf{x}$ is an eigenvector for $\mathbf{A}$ if

$$\mathbf{A}\mathbf{x} = \lambda \mathbf{x}$$

for some scalar λ (the eigenvalue for $\mathbf{x}$).
"In general",

- An $n \times n$ matrix has n distinct eigenvalues $\lambda_1, \ldots, \lambda_n$, (some may be complex), and
- n corresponding linearly independent eigenvectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$
- Any nonzero multiple of an eigenvector is again an eigenvector.

The Importance Eigenvector



Such an eigenvector exists here:

$\mathbf{x} \approx [0.387, 0.129, 0.290, 0.194]^T$

(or any multiple) satisfies
$\mathbf{x} = \mathbf{A}\mathbf{x}$.

Such an eigenvector must ALWAYS exist if $\mathbf{A}$ is *column-stochastic* (columns of $\mathbf{A}$ sum to one) and there are no dangling nodes.

Desirable Properties of the Eigenvector

Desirable Properties of the Eigenvector

- Should be non-negative.

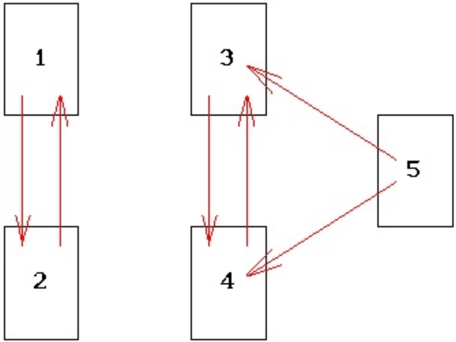
Desirable Properties of the Eigenvector

- Should be non-negative.
- Should be unique up to scaling (one-dimensional eigenspace).

Desirable Properties of the Eigenvector

- Should be non-negative.
- Should be unique up to scaling (one-dimensional eigenspace).
- Should be "easy" to compute for an eight billion by eight billion matrix.

Problems



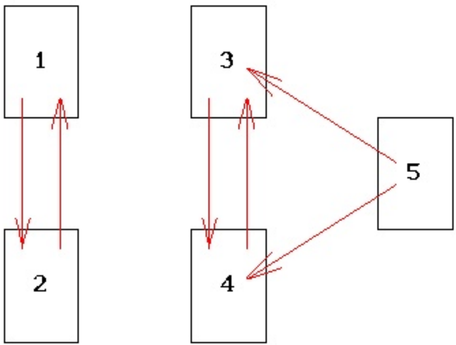
The link matrix here is

$$\mathbf{A} = \left[\begin{array}{cc|ccc} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 1/2 \\ 0 & 0 & 1 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right].$$

Two-dimensional eigenspace, basis $\mathbf{x}_1 = [1/2, 1/2, 0, 0, 0]^T$, $\mathbf{x}_2 = [0, 0, 1/2, 1/2, 0]^T$.

Any linear combination $c_1\mathbf{x}_1 + c_2\mathbf{x}_2$ is an eigenvector too!

Web With Multiple Components



Which eigenvector should we use?

Notation: $V_1(\mathbf{A})$ is the subspace of eigenvectors for $\mathbf{A}$ with eigenvalue 1.

Theorem: If a web has r components (considered as a graph) then $V_1(\mathbf{A})$ has at least dimension r.

A Solution for Multiple Components

- Give every page a "weak link" to every other page!

A Solution for Multiple Components

- Give every page a “weak link” to every other page!
- Replace n by n matrix $\mathbf{A}$ with weighted combination

$$\mathbf{M} = (1 - m)\mathbf{A} + m\mathbf{S}$$

where $0 \leq m \leq 1$ and $\mathbf{S}$ is n by n,

$$\mathbf{S} = \begin{bmatrix} 1/n & 1/n & \cdots & 1/n \\ \vdots & \vdots & \vdots & \vdots \\ 1/n & 1/n & \cdots & 1/n \end{bmatrix}$$

A Solution for Multiple Components

- Give every page a "weak link" to every other page!
- Replace n by n matrix $\mathbf{A}$ with weighted combination

$$\mathbf{M} = (1-m)\mathbf{A} + m\mathbf{S}$$

where $0 \leq m \leq 1$ and $\mathbf{S}$ is n by n,

$$\mathbf{S} = \begin{bmatrix} 1/n & 1/n & \cdots & 1/n \\ \vdots & \vdots & \vdots & \vdots \\ 1/n & 1/n & \cdots & 1/n \end{bmatrix}$$

- Then $\mathbf{M}$ is column-stochastic. Original problem is $m = 0$, while $m = 1$ is egalitarian web—all pages equal importance.

Desirable Properties of $\mathbf{M}$

$\mathbf{M}$ is the matrix we want. If $m > 0$ one can show that

Desirable Properties of $\mathbf{M}$

$\mathbf{M}$ is the matrix we want. If $m > 0$ one can show that

- The dominant eigenvalue for $\mathbf{M}$ is 1, and $V_1(\mathbf{M})$ is one-dimensional, so

Desirable Properties of M

M is the matrix we want. If $m > 0$ one can show that

- The dominant eigenvalue for **M** is 1, and $V_1(\mathbf{M})$ is one-dimensional, so
- There is a unique non-negative eigenvector **x** with $\sum_i x_i = 1$.

Desirable Properties of **M**

M is the matrix we want. If $m > 0$ one can show that

- The dominant eigenvalue for **M** is 1, and $V_1(\mathbf{M})$ is one-dimensional, so
- There is a unique non-negative eigenvector **x** with $\sum_i x_i = 1$.
- We can use this **x** for unambiguous importance ratings.

Desirable Properties of M

M is the matrix we want. If $m > 0$ one can show that

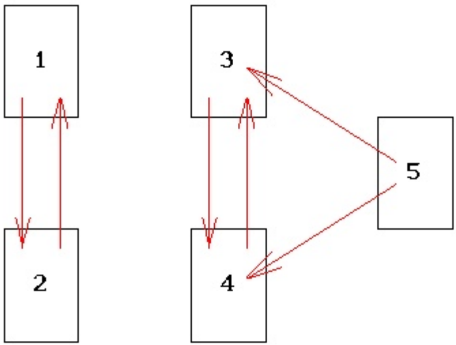
- The dominant eigenvalue for **M** is 1, and $V_1(\mathbf{M})$ is one-dimensional, so
- There is a unique non-negative eigenvector **x** with $\sum_i x_i = 1$.
- We can use this **x** for unambiguous importance ratings.
- Google (reputedly) uses $m = 0.15$.

Desirable Properties of **M**

M is the matrix we want. If $m > 0$ one can show that

- The dominant eigenvalue for **M** is 1, and $V_1(\mathbf{M})$ is one-dimensional, so
- There is a unique non-negative eigenvector **x** with $\sum_i x_i = 1$.
- We can use this **x** for unambiguous importance ratings.
- Google (reputedly) uses $m = 0.15$.
- All we need is a method for finding an eigenvector for an eight billion by eight billion matrix!

Multiple Component Example



With $m = 0.15$ the link matrix is

$$\begin{bmatrix} 0.03 & 0.88 & 0.03 & 0.03 & 0.03 \\ 0.88 & 0.03 & 0.03 & 0.03 & 0.03 \\ 0.03 & 0.03 & 0.03 & 0.88 & 0.46 \\ 0.03 & 0.03 & 0.88 & 0.03 & 0.46 \\ 0.03 & 0.03 & 0.03 & 0.03 & 0.03 \end{bmatrix}$$

The importance eigenvector is
$\mathbf{x} =$
$[0.2, 0, 2, 0.285, 0.285, 0.03]^T$.

Interesting observation: As
$m \to 0^+$,
$\mathbf{x} \to [0.2, 0, 2, 0.3, 0.3, 0.0]^T$.

Computing the Eigenvector

The Power Method for find the dominant eigenvector of a matrix $\mathbf{M}$:

Computing the Eigenvector

The Power Method for find the dominant eigenvector of a matrix $\mathbf{M}$:

1. Take a non-negative guess $\mathbf{x}_0$ at the eigenvector, scaled so $\sum_j x_0^{(j)} = 1$. Set counter $k = 0$.

Computing the Eigenvector

The Power Method for find the dominant eigenvector of a matrix $\mathbf{M}$:

1. Take a non-negative guess $\mathbf{x}_0$ at the eigenvector, scaled so $\sum_j x_0^{(j)} = 1$. Set counter $k = 0$.
2. Let $\mathbf{x}_{k+1} = \mathbf{M}\mathbf{x}_k$.

Computing the Eigenvector

The Power Method for find the dominant eigenvector of a matrix $\mathbf{M}$:

1. Take a non-negative guess $\mathbf{x}_0$ at the eigenvector, scaled so $\sum_j x_0^{(j)} = 1$. Set counter $k = 0$.
2. Let $\mathbf{x}_{k+1} = \mathbf{M}\mathbf{x}_k$.
3. If $\mathbf{x}_{k+1} - \mathbf{x}_k$ is small, terminate with estimated dominant eigenvector $\mathbf{x}_{k+1}$. Otherwise, increment k and return to step 2.

Computing the Eigenvector

The Power Method for find the dominant eigenvector of a matrix $\mathbf{M}$:

1. Take a non-negative guess $\mathbf{x}_0$ at the eigenvector, scaled so $\sum_j x_0^{(j)} = 1$. Set counter $k = 0$.
2. Let $\mathbf{x}_{k+1} = \mathbf{M}\mathbf{x}_k$.
3. If $\mathbf{x}_{k+1} - \mathbf{x}_k$ is small, terminate with estimated dominant eigenvector $\mathbf{x}_{k+1}$. Otherwise, increment k and return to step 2.

In short, $\mathbf{M}^k\mathbf{x}_0$ will converge to the eigenvector we want, for “any” initial guess $\mathbf{x}_0$.

Power Method Example

With the **M** for the five page network above we find

$$\mathbf{x}_0 = \begin{bmatrix} 0.1 \\ 0.1 \\ 0.2 \\ 0.2 \\ 0.4 \end{bmatrix}, \quad \mathbf{M}^{20}\mathbf{x}_0 = \begin{bmatrix} 0.196 \\ 0.196 \\ 0.289 \\ 0.289 \\ 0.03 \end{bmatrix}, \quad \mathbf{M}^{40}\mathbf{x}_0 = \begin{bmatrix} 0.199 \\ 0.199 \\ 0.286 \\ 0.286 \\ 0.03 \end{bmatrix},$$

Convergence and Implementation

Convergence and Implementation

- In general the power method converges geometrically, as $\left(\frac{|\lambda_2|}{|\lambda_1|}\right)^k$, where λ_2 is the second largest eigenvalue.

Convergence and Implementation

- In general the power method converges geometrically, as $\left(\frac{|\lambda_2|}{|\lambda_1|}\right)^k$, where λ_2 is the second largest eigenvalue.
- For $\mathbf{M}$ we have $\lambda_1 = 1$ and it can be shown that $\lambda_2 = 1 - m$. If $m = 0.15$ the method converges in proportion to $(0.85)^k$.

Convergence and Implementation

- In general the power method converges geometrically, as $\left(\frac{|\lambda_2|}{|\lambda_1|}\right)^k$, where λ_2 is the second largest eigenvalue.
- For **M** we have $\lambda_1 = 1$ and it can be shown that $\lambda_2 = 1 - m$. If $m = 0.15$ the method converges in proportion to $(0.85)^k$.
- But **M** is dense (mostly non-zero). Computing the matrix-vector product **Mv** requires about 1.3×10^{20} operations for a web with eight billion pages!

Convergence and Implementation

- $\mathbf{Mv}$ can be computed efficiently: $\mathbf{M} = (1 - m)\mathbf{A} + m\mathbf{S}$, where $\mathbf{S}$ has all entries $1/n$ and $\mathbf{A}$ is sparse (mostly zeros), since most web pages link to only a few other pages.

Convergence and Implementation

- $\mathbf{Mv}$ can be computed efficiently: $\mathbf{M} = (1 - m)\mathbf{A} + m\mathbf{S}$, where $\mathbf{S}$ has all entries $1/n$ and $\mathbf{A}$ is sparse (mostly zeros), since most web pages link to only a few other pages.
- Note $\mathbf{Sv}$ is a vector with all entries $\frac{1}{n}\sum_j v_j$. In our case $\mathbf{Sv} = [1/n, 1/n, \ldots, 1/n]^T$.

Convergence and Implementation

- $\mathbf{Mv}$ can be computed efficiently: $\mathbf{M} = (1-m)\mathbf{A} + m\mathbf{S}$, where $\mathbf{S}$ has all entries $1/n$ and $\mathbf{A}$ is sparse (mostly zeros), since most web pages link to only a few other pages.
- Note $\mathbf{Sv}$ is a vector with all entries $\frac{1}{n}\sum_j v_j$. In our case $\mathbf{Sv} = [1/n, 1/n, \ldots, 1/n]^T$.
- If each page has, on average, 10 outgoing links, $\mathbf{A}$ has only 10 non-zero entries per column. $\mathbf{Av}$ can be computed in about 1.6×10^{11} operations.

Convergence and Implementation

- $\mathbf{Mv}$ can be computed efficiently: $\mathbf{M} = (1 - m)\mathbf{A} + m\mathbf{S}$, where $\mathbf{S}$ has all entries $1/n$ and $\mathbf{A}$ is sparse (mostly zeros), since most web pages link to only a few other pages.
- Note $\mathbf{Sv}$ is a vector with all entries $\frac{1}{n}\sum_j v_j$. In our case $\mathbf{Sv} = [1/n, 1/n, \ldots, 1/n]^T$.
- If each page has, on average, 10 outgoing links, $\mathbf{A}$ has only 10 non-zero entries per column. $\mathbf{Av}$ can be computed in about 1.6×10^{11} operations.
- Then $\mathbf{Mv} = (1-m)\mathbf{Av} + m\mathbf{Sv} = (1-m)\mathbf{Av} + [m/n, m/n, \ldots, m/n]^T$ can be computed in a reasonable time.

Conclusions

Conclusions

- The “Google” approach to importance ranking has found other applications, e.g., “food webs,” game theory, medicine.

Conclusions

- The "Google" approach to importance ranking has found other applications, e.g., "food webs," game theory, medicine.
- Sophisticated linear algebra is at the core of much scientific computation, and pops up when you least expect it.

Conclusions

- The "Google" approach to importance ranking has found other applications, e.g., "food webs," game theory, medicine.
- Sophisticated linear algebra is at the core of much scientific computation, and pops up when you least expect it.
- If you pay attention to your math professors, you might become a billionaire.

Resources

1. *The $25,000,000,000 Eigenvector: The Linear Algebra Behind Google*, SIAM Review (Education Section), Vol 48 (3), August 2006.
2. `www.rose-hulman.edu/~bryan/google.html`