COMP4121 Advanced Algorithms

Aleks Ignjatović

School of Computer Science and Engineering
University of New South Wales Sydney

The PageRank, Markov chains and random walks on graphs

Basic tools: Eigenvalues and Eigenvectors

Before we can start studying the PageRank, we need to remind ourselves about some basic matrix theory.

- Matrices of size $M \times M$ [illegible] ve much fewer than M^2 non zero entries are called *sparse matrices*.
- We say that λ is a *left* [illegible] of a matrix G if there exist a vector $\mathbf{x}$ such that

$$\mathbf{x}^\top G = \lambda \mathbf{x}^\top$$

- Vectors satisfying this property are called the (left) eigenvectors corresponding to the (left) eigenvalue λ.

- A matrix of size $M \times M$ can have up to M distinct eigenvalues, which are, in general, complex numbers.

- $\mathbf{x}^\top G = \lambda \mathbf{x}^\top$ is equivalent to $\mathbf{x}^\top (G - \lambda I) = 0$, where I is the identity matrix having 1 on the diagonal and zeros elsewhere.

$$G - \lambda I = \begin{pmatrix} g_{1,1} - \lambda & g_{1,2} & \dots & g_{1,n-1} & g_{1,n} \\ g_{2,1} & g_{2,2} - \lambda & \dots & g_{2,n-1} & g_{2,n} \\ & & \dots & & \\ g_{n,1} & g_{n,2} & \dots & g_{n,n-1} & g_{n,n} - \lambda \end{pmatrix}$$

Basic tools: Eigenvalues and Eigenvectors

Before we can start studying the PageRank, we need to remind ourselves about some basic matrix theory.

- Matrices of size $M \times M$ which have much fewer than M^2 non zero entries are called *sparse matrices*.
- We say that λ is a *left eigenvalue* of a matrix G if there exist a vector $\mathbf{x}$ such that

$$\mathbf{x}^{\top} G = \lambda \mathbf{x}^{\top}$$

- Vectors satisfying this property are called the (left) eigenvectors corresponding to the (left) eigenvalue λ.

- A matrix of size $M \times M$ can have up to M distinct eigenvalues, which are, in general, complex numbers.

- $\mathbf{x}^{\top} G = \lambda \mathbf{x}^{\top}$ is equivalent to $\mathbf{x}^{\top}(G - \lambda I) = 0$, where I is the identity matrix having 1 on the diagonal and zeros elsewhere.

$$G - \lambda I = \begin{pmatrix} g_{1,1} - \lambda & g_{1,2} & \dots & g_{1,n-1} & g_{1,n} \\ g_{2,1} & g_{2,2} - \lambda & \dots & g_{2,n-1} & g_{2,n} \\ & & \dots & & \\ g_{n,1} & g_{n,2} & \dots & g_{n,n-1} & g_{n,n} - \lambda \end{pmatrix}$$

Basic tools: Eigenvalues and Eigenvectors

Before we can start studying the PageRank, we need to remind ourselves about some basic matrix theory.

- Matrices of size $M \times M$ which have much fewer than M^2 non zero entries are called *sparse matrices*.
- We say that λ is a *left eigenvalue* of a matrix G if there exist a vector $\mathbf{x}$ such that

$$\mathbf{x}^\intercal G = \lambda \mathbf{x}^\intercal$$

- Vectors satisfying this property are called the (left) eigenvectors corresponding to the (left) eigenvalue λ.
- A matrix of size $M \times M$ can have up to M distinct eigenvalues, which are, in general, complex numbers.
- $\mathbf{x}^\intercal G = \lambda \mathbf{x}^\intercal$ is equivalent to $\mathbf{x}^\intercal (G - \lambda I) = 0$, where I is the identity matrix having 1 on the diagonal and zeros elsewhere.

$$G - \lambda I = \begin{pmatrix} g_{1,1} - \lambda & g_{1,2} & \dots & g_{1,n-1} & g_{1,n} \\ g_{2,1} & g_{2,2} - \lambda & \dots & g_{2,n-1} & g_{2,n} \\ & & \dots & & \\ g_{n,1} & g_{n,2} & \dots & g_{n,n-1} & g_{n,n} - \lambda \end{pmatrix}$$

Before we can start studying the PageRank, we need to remind ourselves about some basic matrix theory.

- Matrices of size $M \times M$ which have much fewer than M^2 non zero entries are called *sparse matrices*.
- We say that λ is a *left eigenvalue* of a matrix G if there exist a vector $\mathbf{x}$ such that

$$\mathbf{x}^{\intercal} G = \lambda \mathbf{x}^{\intercal}$$

- Vectors satisfying this property are called the (left) eigenvectors corresponding to the (left) eigenvalue λ.

- A matrix of size $M \times M$ can have up to M distinct eigenvalues, which are, in general, complex numbers.

- $\mathbf{x}^{\intercal} G = \lambda \mathbf{x}^{\intercal}$ is equivalent to $\mathbf{x}^{\intercal}(G - \lambda I) = 0$, where I is the identity matrix having 1 on the diagonal and zeros elsewhere.

$$G - \lambda I = \begin{pmatrix} g_{1,1} - \lambda & g_{1,2} & \dots & g_{1,n-1} & g_{1,n} \\ g_{2,1} & g_{2,2} - \lambda & \dots & g_{2,n-1} & g_{2,n} \\ & & \dots & & \\ g_{n,1} & g_{n,2} & \dots & g_{n,n-1} & g_{n,n} - \lambda \end{pmatrix}$$

Before we can start studying the PageRank, we need to remind ourselves about some basic matrix theory.

- Matrices of size $M \times M$ which have much fewer than M^2 non zero entries are called *sparse matrices*.
- We say that λ is a *left eigenvalue* of a matrix G if there exist a vector $\mathbf{x}$ such that

$$\mathbf{x}^{\intercal} G = \lambda \mathbf{x}^{\intercal}$$

- Vectors satisfying this property are called the (left) eigenvectors corresponding to the (left) eigenvalue λ.
- A matrix of size $M \times M$ can have up to M distinct eigenvalues, which are, in general, complex numbers.
- $\mathbf{x}^{\intercal} G = \lambda \mathbf{x}^{\intercal}$ is equivalent to $\mathbf{x}^{\intercal}(G - \lambda I) = 0$, where I is the identity matrix having 1 on the diagonal and zeros elsewhere.

$$G - \lambda I = \begin{pmatrix} g_{1,1} - \lambda & g_{1,2} & \dots & g_{1,n-1} & g_{1,n} \\ g_{2,1} & g_{2,2} - \lambda & \dots & g_{2,n-1} & g_{2,n} \\ & & \dots & & \\ g_{n,1} & g_{n,2} & \dots & g_{n,n-1} & g_{n,n} - \lambda \end{pmatrix}$$

Basic tools: Eigenvalues and Eigenvectors

Before we can start studying the PageRank, we need to remind ourselves about some basic matrix theory.

- Matrices of size $M \times M$ that have much fewer than M^2 non zero entries are called *sparse matrices*.
- We say that λ is a *left eigenvalue* of a matrix G if there exist a vector $\mathbf{x}$ such that

$$\mathbf{x}^\intercal G = \lambda \mathbf{x}^\intercal$$

- Vectors satisfying this property are called the (left) eigenvectors corresponding to the (left) eigenvalue λ.
- A matrix of size $M \times M$ can have up to M distinct eigenvalues, which are, in general, complex numbers.
- $\mathbf{x}^\intercal G = \lambda \mathbf{x}^\intercal$ is equivalent to $\mathbf{x}^\intercal (G - \lambda I) = 0$, where I is the identity matrix having 1 on the diagonal and zeros elsewhere.

$$G - \lambda I = \begin{pmatrix} g_{1,1} - \lambda & g_{1,2} & \dots & g_{1,n-1} & g_{1,n} \\ g_{2,1} & g_{2,2} - \lambda & \dots & g_{2,n-1} & g_{2,n} \\ & & \dots & & \\ g_{n,1} & g_{n,2} & \dots & g_{n,n-1} & g_{n,n} - \lambda \end{pmatrix}$$

- Such a homogeneous li… n has a non zero solution just in case the determinant of the sys… , i.e., $\mathrm{Det}(G - \lambda I) = 0$, which produces a polynomial equation o… in λ, $p_n(\lambda) = 0$.

$$\mathrm{Det}(G - \lambda I) = \begin{vmatrix} g_{1,1} - \lambda & g_{1,2} & \dots & g_{1,n-1} & g_{1,n} \\ g_{2,1} & g_{2,2} - \lambda & \dots & g_{2,n-1} & g_{2,n} \\ & & \dots & & \\ g_{n,1} & g_{n,2} & \dots & g_{n,n-1} & g_{n,n} - \lambda \end{vmatrix} = 0$$

- Polynomial $p_n(\lambda)$ is called the characteristic polynomial of matrix G.

- Such a homogeneous li... n has a non zero solution just in case the determinant of the sys..., i.e., $\text{Det}(G - \lambda I) = 0$, which produces a polynomial equation o... in λ, $p_n(\lambda) = 0$.

$$\text{Det}(G - \lambda I) = \begin{vmatrix} g_{1,1} - \lambda & g_{1,2} & \dots & g_{1,n-1} & g_{1,n} \\ g_{2,1} & g_{2,2} - \lambda & \dots & g_{2,n-1} & g_{2,n} \\ & & \dots & & \\ g_{n,1} & g_{n,2} & \dots & g_{n,n-1} & g_{n,n} - \lambda \end{vmatrix} = 0$$

- Polynomial $p_n(\lambda)$ is called the characteristic polynomial of matrix G.

- Such a homogeneous li[illegible]n has a non zero solution just in case the determinant of the sys[illegible], i.e., $\mathrm{Det}(G - \lambda I) = 0$, which produces a polynomial equation of degree n in λ, $p_n(\lambda) = 0$.

$$\mathrm{Det}(G - \lambda I) = \begin{vmatrix} g_{1,1} - \lambda & g_{1,2} & \dots & g_{1,n-1} & g_{1,n} \\ g_{2,1} & g_{2,2} - \lambda & \dots & g_{2,n-1} & g_{2,n} \\ & & \dots & & \\ g_{n,1} & g_{n,2} & \dots & g_{n,n-1} & g_{n,n} - \lambda \end{vmatrix} = 0$$

- Polynomial $p_n(\lambda)$ is called the characteristic polynomial of matrix G.

Basic tools: Eigenvalues and Eigenvectors

- Similarly, λ_r is a right eigenvalue of G and $\mathbf{r}$ is a right eigenvector of G corresponding to the eigenvalue λ_r if

$$G\mathbf{r} = \lambda_r \mathbf{r}$$

- Note that this yields $(G - \lambda_r I)\mathbf{r} = 0$ which has a solution just in case $\mathrm{Det}(G - \lambda_r I) = 0$. This yields exactly the same polynomial equation for λ_r and consequently the left and the right eigenvalues coincide.
- However, the left and the right eigenvectors are different and we can see that $G\mathbf{r} = \lambda_r \mathbf{r}$ implies $\mathbf{r}^\top G^\top = \lambda_r \mathbf{r}^\top$.
- Thus, the left eigenvectors of G are the right eigenvectors of $G^\top$ and vice versa.

Basic tools: Eigenvalues and Eigenvectors

- Similarly, λ_r is a right eigenvalue of G and $\mathbf{r}$ is a right eigenvector of G corresponding to the eigenvalue λ_r if

$$G\mathbf{r} = \lambda_r \mathbf{r}$$

- Note that this yields $(G - \lambda_r I)\mathbf{r} = \mathbf{0}$ which has a solution just in case $\mathrm{Det}(G - \lambda_r I) = 0$. This yields exactly the same polynomial equation for λ_r and consequently the left and the right eigenvalues coincide.

- However, the left and the right eigenvectors are different and we can see that $G\mathbf{r} = \lambda_r\mathbf{r}$ implies $\mathbf{r}^\top G^\top = \lambda_r \mathbf{r}^\top$.

- Thus, the left eigenvectors of G are the right eigenvectors of $G^\top$ and vice versa.

- Similarly, λ_r is a right [illegible] of G and $\mathbf{r}$ is a right eigenvector of G corresponding to the e[illegible] if

$$G\mathbf{r} = \lambda_r \mathbf{r}$$

- Note that this yields $(G - \lambda_r I)\mathbf{r} = \mathbf{0}$ which has a solution just in case $\mathrm{Det}(G - \lambda_r I) = 0$. This yields exactly the same polynomial equation for λ_r and consequently the left and the right eigenvalues coincide.
- However, the left and the right eigenvectors are different and we can see that $G\mathbf{r} = \lambda_r \mathbf{r}$ implies $\mathbf{r}^\top G^\top = \lambda_r \mathbf{r}^\top$.
- Thus, the left eigenvectors of G are the right eigenvectors of $G^\top$ and vice versa.

Basic tools: Eigenvalues and Eigenvectors

- Similarly, λ_r is a right eigenvalue of G and $\mathbf{r}$ is a right eigenvector of G corresponding to the eigenvalue λ_r if

$$G\mathbf{r} = \lambda_r \mathbf{r}$$

- Note that this yields $(G - \lambda_r I)\mathbf{r} = \mathbf{0}$ which has a solution just in case $\mathrm{Det}(G - \lambda_r I) = 0$. This yields exactly the same polynomial equation for λ_r and consequently the left and the right eigenvalues coincide.

- However, the left and the right eigenvectors are different and we can see that $G\mathbf{r} = \lambda_r \mathbf{r}$ implies $\mathbf{r}^\intercal G^\intercal = \lambda_r \mathbf{r}^\intercal$.

- Thus, the left eigenvectors of G are the right eigenvectors of $G^\intercal$ and vice versa.

Problem: ordering webpages according to their importance

- **Setup:** Consider all the webpages P_i on the **entire** WWW as vertices of a directed graph.
- A directed edge $P_i \rightarrow P_j$ exists just in case page P_i points to page P_j (i.e., P_i has a link to page P_j).
- **Problem:** Rank all the webpages of the WWW according to their "importance".
- **First attempt:** P_0 should have a high rank if many pages point to a page P_0.
- This would be easy to manipulate by creating a lot of bogus pages which point to P_0.

- **Setup:** Consider all the webpages P_i on the **entire** WWW as vertices of a directed graph.
- A directed edge $P_i \rightarrow P_j$ exists just in case page P_i points to page P_j (i.e., P_i has a link to page P_j).

- **Problem:** Rank all the webpages of the WWW according to their "importance".

- **First attempt:** P_0 should have a high rank if many pages point to a page P_0.

- This would be easy to manipulate by creating a lot of bogus pages which point to P_0.

- **Setup:** Consider all t[illegible]s P_i on the **entire** WWW as vertices of a directed graph.
- A directed edge $P_i \rightarrow P_j$ exists just in case page P_i points to page P_j (i.e., P_i has a link to page P_j).

- **Problem:** Rank all the webpages of the WWW according to their "importance".

- **First attempt:** P_0 should have a high rank if many pages point to a page P_0.

- This would be easy to manipulate by creating a lot of bogus pages which point to P_0.

Problem: ordering webpages according to their importance

- **Setup:** Consider all t[illegible]s P_i on the **entire** WWW as vertices of a directed graph.
- A directed edge $P_i \to P_j$ exists just in case page P_i points to page P_j (i.e., P_i has a link to page P_j).

- **Problem:** Rank all the webpages of the WWW according to their "importance".

- **First attempt:** P_0 should have a high rank if many pages point to a page P_0.

- This would be easy to manipulate by creating a lot of bogus pages which point to P_0.

- **Setup:** Consider all the webpages P_i on the **entire** WWW as vertices of a directed graph.

- A directed edge $P_i \rightarrow P_j$ exists just in case page P_i points to page P_j (i.e., P_i has a link to page P_j).

- **Problem:** Rank all the webpages of the WWW according to their "importance".

- **First attempt:** P_0 should have a high rank if many pages point to a page P_0.

- This would be easy to manipulate by creating a lot of bogus pages which point to P_0.

- **Second attempt:** P ... ve a high rank if many pages which themselves are pointed ... y other pages point to a page P_0.

- This would also be eas ... ulate by creating a lot of bogus pages which point to P_0 and which also points to other bogus pages.

- **One of the main ideas behind the PageRank:** In order to make it hard to manipulate, the page rank of each webpage on the web should depend on page ranks of ALL other web pages on the web!

- In this case no individual player can manipulate the rank of a page because everybody can control only a minuscule fraction of all the webpages on the web!

- How can we accomplish this?

- **Second attempt:** P[illegible]ve a high rank if many pages which themselves are pointed[illegible]y other pages point to a page P_0.
- This would also be eas[illegible]ulate by creating a lot of bogus pages which point to P_0 and which also points to other bogus pages.

- **One of the main ideas behind the PageRank:** In order to make it hard to manipulate, the pag[illegible] should depend on page ranks of ALL other web pages on the web!

- In this case no individual player can manipulate the rank of a page because everybody can control [illegible]fraction of all the webpages on the web!

- How can we accomplish this?

Problem: ordering webpages according to their importance

- **Second attempt:** P… ve a high rank if many pages which themselves are pointed … y other pages point to a page P_0.
- This would also be eas… ulate by creating a lot of bogus pages which point to P_0 and which also points to other bogus pages.

- **One of the main ideas behind the PageRank:** In order to make it hard to manipulate, the page rank of each webpage on the web should depend on page ranks of ALL other web pages on the web!

- In this case no individual player can manipulate the rank of a page because everybody can control only a minuscule fraction of all the webpages on the web!

- How can we accomplish this?

- **Second attempt:** P... ve a high rank if many pages which themselves are pointed ... y other pages point to a page P_0.

- This would also be eas ... ulate by creating a lot of bogus pages which point to P_0 and which also points to other bogus pages.

- **One of the main ideas behind the PageRank:** In order to make it hard to manipulate, the page rank of each webpage on the web should depend on page ranks of ALL other web pages on the web!

- In this case no individual player can manipulate the rank of a page because everybody can control only a minuscule fraction of all the webpages on the web!

- How can we accomplish this?

- **Second attempt:** P ... ve a high rank if many pages which themselves are pointed ... y other pages point to a page P_0.

- This would also be eas ... ulate by creating a lot of bogus pages which point to P_0 and which also points to other bogus pages.

- **One of the main ideas behind the PageRank:** In order to make it hard to manipulate, the page rank of each webpage on the web should depend on page ranks of ALL other web pages on the web!

- In this case no individual player can manipulate the rank of a page because everybody can control only a minuscule fraction of all the webpages on the web!

- How can we accomplish this?

- **Notation:**
  - $\rho(P)$ = the rank of a page (to be assigned);
  - $\#(P)$ = the number of outgoing links on a web page.
- A web page P should have a high rank $\rho(P)$ only if it is pointed at by many pages P_i which:
  1. themselves have a high rank $\rho(P_j)$,
  2. and do not point to an excessive number of other web pages, i.e., $\#(P_j)$ is reasonably small.

- So we would like to have valid something like this:

$$\rho(P) = \sum_{P_i \rightarrow P} \frac{\rho(P_i)}{\#(P_i)} \qquad (1)$$

- **Note:** this formula is circular and thus cannot be directly used to compute $\rho(P)$: in order to get $\rho(P)$ we would need to have all $\rho(P_j)$ already assigned!
- Rather, this is just a condition which the ranks might satisfy.

Problem: ordering webpages according to their importance

- **Notation:**
  - $\rho(P)$ = the rank of a page (to be assigned);
  - $\#(P)$ = the number of outgoing links on a web page.

- A web page P should have a high rank $\rho(P)$ only if it is pointed at by many pages P_i which:
  1. themselves have a high rank $\rho(P_j)$,
  2. and do not point to an excessive number of other web pages, i.e., $\#(P_j)$ is reasonably small.

- So we would like to have valid something like this:

$$\rho(P) = \sum_{P_i \to P} \frac{\rho(P_i)}{\#(P_i)} \qquad (1)$$

- **Note:** this formula is circular and thus cannot be directly used to compute $\rho(P)$: in order to get $\rho(P)$ we would need to have all $\rho(P_j)$ already assigned!
- Rather, this is just a condition which the ranks might satisfy.

- **Notation:**
  - $\rho(P)$ = the rank of a page (to be assigned);
  - $\#(P)$ = the number of outgoing links on a web page.
- A web page P should have a high rank $\rho(P)$ only if it is pointed at by many pages P_i which:
  1. themselves have a high rank $\rho(P_j)$,
  2. and do not point to an excessive number of other web pages, i.e., $\#(P_j)$ is reasonably small.
- So we would like to have valid something like this:

$$\rho(P) = \sum_{P_i \to P} \frac{\rho(P_i)}{\#(P_i)} \qquad (1)$$

- **Note:** this formula is circular and thus cannot be directly used to compute $\rho(P)$: in order to get $\rho(P)$ we would need to have all $\rho(P_j)$ already assigned!
- Rather, this is just a condition which the ranks might satisfy.

Problem: ordering webpages according to their importance

- **Notation:**
  - $\rho(P)$ = the rank of a page (to be assigned);
  - $\#(P)$ = the number of outgoing links on a web page.
- A web page P should have a high rank $\rho(P)$ only if it is pointed at by many pages P_i which:
  1. themselves have a high rank $\rho(P_j)$,
  2. and do not point to an excessive number of other web pages, i.e., $\#(P_j)$ is reasonably small.

- So we would like to have valid something like this:

$$\rho(P) = \sum_{P_i \to P} \frac{\rho(P_i)}{\#(P_i)} \tag{1}$$

- **Note:** this formula is circular and thus cannot be directly used to compute $\rho(P)$: in order to get $\rho(P)$ we would need to have all $\rho(P_j)$ already assigned!
- Rather, this is just a condition which the ranks might satisfy.

Problem: ordering webpages according to their importance

- **Notation:**
  - $\rho(P)$ = the rank of a page (to be assigned);
  - $\#(P)$ = the number of outgoing links on a web page.
- A web page P should have a high rank $\rho(P)$ only if it is pointed at by many pages P_i which:
  1. themselves have a high rank $\rho(P_j)$,
  2. and do not point to an excessive number of other web pages, i.e., $\#(P_j)$ is reasonably small.
- So we would like to have valid something like this:

$$\rho(P) = \sum_{P_i \to P} \frac{\rho(P_i)}{\#(P_i)} \tag{1}$$

- **Note:** this formula is circular and thus cannot be directly used to compute $\rho(P)$: in order to get $\rho(P)$ we would need to have all $\rho(P_j)$ already assigned!
- Rather, this is just a condition which the ranks might satisfy.

Problem: ordering webpages according to their importance

- **Notation:**
  - $\rho(P)$ = the rank of a page (to be assigned);
  - $\#(P)$ = the number of outgoing links on a web page.
- A web page P should have a high rank $\rho(P)$ only if it is pointed at by many pages P_i which:
  1. themselves have a high rank $\rho(P_j)$,
  2. and do not point to an excessive number of other web pages, i.e., $\#(P_j)$ is reasonably small.

- So we would like to have valid something like this:

$$\rho(P) = \sum_{P_i \to P} \frac{\rho(P_i)}{\#(P_i)} \tag{1}$$

- **Note:** this formula is circular and thus cannot be directly used to compute $\rho(P)$: in order to get $\rho(P)$ we would need to have all $\rho(P_j)$ already assigned!
- Rather, this is just a condition which the ranks might satisfy.

Problem: ordering webpages according to their importance

- **Notation:**
  - $\rho(P)$ = the rank of a page (to be assigned);
  - $\#(P)$ = the number of outgoing links on a web page.
- A web page P should have a high rank $\rho(P)$ only if it is pointed at by many pages P_i which:
  1. themselves have a high rank $\rho(P_j)$,
  2. and do not point to an excessive number of other web pages, i.e., $\#(P_j)$ is reasonably small.

- So we would like to have valid something like this:

$$\rho(P) = \sum_{P_i \to P} \frac{\rho(P_i)}{\#(P_i)} \tag{1}$$

- **Note:** this formula is circular and thus cannot be directly used to compute $\rho(P)$: in order to get $\rho(P)$ we would need to have all $\rho(P_j)$ already assigned!
- Rather, this is just a condition which the ranks might satisfy.

Problem: ordering webpages according to their importance

- **Notation:**
  - $\rho(P)$ = the rank of a page (to be assigned);
  - $\#(P)$ = the number of outgoing links on a web page.
- A web page P should have a high rank $\rho(P)$ only if it is pointed at by many pages P_i which:
  1. themselves have a high rank $\rho(P_j)$,
  2. and do not point to an excessive number of other web pages, i.e., $\#(P_j)$ is reasonably small.

- So we would like to have valid something like this:

$$\rho(P) = \sum_{P_i \to P} \frac{\rho(P_i)}{\#(P_i)} \tag{1}$$

- **Note:** this formula is circular and thus cannot be directly used to compute $\rho(P)$: in order to get $\rho(P)$ we would need to have all $\rho(P_j)$ already assigned!
- Rather, this is just a condition which the ranks might satisfy.

Problem: ordering webpages according to their importance

- Two equally important questions:
  1. Why should suc... at all?
  2. Even if such ρ e... the **unique** solution satisfying (1)?

- Question 2 above is as important as Question 1, why?
- Relative ordering of web pages should NOT involve any randomness, because it might decide which e-commerce outlet gets an order and this should not be random!

- Note that the collection

$$\left\{ \rho(P) = \sum_{P_i \to P} \frac{\rho(P_i)}{\#(P_i)} \right\}_{P \in WWW} \tag{2}$$

can be seen as a system of equations in the variables $\rho(P_i)$, one for each page P on the web, which we would like to hold.

Problem: ordering webpages according to their importance

- Two equally important questions:
  1. Why should suc … at all?
  2. Even if such ρ e … the **unique** solution satisfying (1)?

- Question 2 above is as important as Question 1, why?
- Relative ordering of web pages should NOT involve any randomness, because it might decide which e-commerce outlet gets an order and this should not be random!

- Note that the collection

$$\left\{ \rho(P) = \sum_{P_i \to P} \frac{\rho(P_i)}{\#(P_i)} \right\}_{P \in WWW} \qquad (2)$$

can be seen as a system of equations in the variables $\rho(P_i)$, one for each page P on the web, which we would like to hold.

Problem: ordering webpages according to their importance

- Two equally important questions:
  1. Why should suc[illegible]at all?
  2. Even if such ρ e[illegible] the **unique** solution satisfying (1)?

- Question 2 above is as important as Question 1, why?

- Relative ordering of web pages should NOT involve any randomness, because it might decide which e[illegible] this should not be random!

- Note that the collection

$$\left\{ \rho(P) = \sum_{P_i \to P} \frac{\rho(P_i)}{\#(P_i)} \right\}_{P \in WWW} \tag{2}$$

can be seen as a system of equations in the variables $\rho(P_i)$, one for each page P on the web, which we would like to hold.

- Two equally important questions:
  1. Why should suc[illegible]at all?
  2. Even if such ρ e[illegible] the **unique** solution satisfying (1)?

- Question 2 above is as important as Question 1, why?

- Relative ordering of web pages should NOT involve any randomness, because it might decide which e-commerce outlet gets an order and this should not be random!

- Note that the collection

$$\left\{ \rho(P) = \sum_{P_i \to P} \frac{\rho(P_i)}{\#(P_i)} \right\}_{P \in WWW} \tag{2}$$

can be seen as a system of equations in the variables $\rho(P_i)$, one for each page P on the web, which we would like to hold.

Problem: ordering webpages according to their importance

- Two equally important questions:
  1. Why should such ρ exist at all?
  2. Even if such ρ exists, is it the **unique** solution satisfying (1)?

- Question 2 above is as important as Question 1, why?

- Relative ordering of web pages should NOT involve any randomness, because it might decide which e-commerce outlet gets an order and this should not be random!

- Note that the collection

$$\left\{\rho(P) = \sum_{P_i \to P} \frac{\rho(P_i)}{\#(P_i)}\right\}_{P \in WWW} \tag{2}$$

can be seen as a system of equations in the variables $\rho(P_i)$, one for each page P on the web, which we would like to hold.

Problem: ordering webpages according to their importance

- Two equally important questions:
  1. Why should such ρ exist at all?
  2. Even if such ρ exists, is it the **unique** solution satisfying (1)?

- Question 2 above is as important as Question 1, why?

- Relative ordering of web pages should NOT involve any randomness, because it might decide which e-commerce outlet gets an order and this should not be random!

- Note that the collection

$$\left\{\rho(P)=\sum_{P_i\to P}\frac{\rho(P_i)}{\#(P_i)}\right\}_{P\in WWW} \tag{2}$$

can be seen as a system of equations in the variables $\rho(P_i)$, one for each page P on the web, which we would like to hold.

- Such a system of equations, one for each page P on the web:

$$\left\{ \rho(P) = \sum_{P_i \to P} \frac{\rho(P_i)}{\#(P_i)} \right\}_{P \in WWW} \tag{3}$$

can be represented in matrix form as

$$\mathbf{r}^{\top} = \mathbf{r}^{\top} G_1 \tag{4}$$

where

$$G_1 = \begin{pmatrix} & \vdots & & \vdots & & \vdots \\ & \vdots & & \vdots & & \vdots \\ \ldots 0\ 0 & \frac{1}{\#(P_i)} & 0\ 0 \ldots \quad \ldots 0\ 0 & \frac{1}{\#(P_i)} & 0\ 0 \ldots \quad \ldots \quad \ldots 0\ 0 & \frac{1}{\#(P_i)} & 0\ 0 \ldots \\ & \vdots & & \vdots & & \vdots \\ & \vdots & & \vdots & & \vdots \end{pmatrix}$$

and $\mathbf{r}^{\top} = (\rho(P_1), \rho(P_2), \ldots, \rho(P_M))$.

- This way we get

$$(\rho(P_1), \rho(P_2), \ldots, \rho(P_i), \ldots, \rho(P_M)) \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \frac{1}{\#(P_{i_1})} \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \frac{1}{\#(P_{i_2})} \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} = \sum_{P_{i_p} \to P_j} \frac{\rho(P_{i_p})}{\#(P_{i_p})} = \rho(P_j).$$

Problem: ordering webpages according to their importance

- Note that matrix G_1 of size $M \times M$ is a sparse matrix.
- Given that $\mathbf{r}$ multiplies G_1 from the left, matrix equation $\mathbf{r}^T = \mathbf{r}^T G_1$ simply says that:
  - 1 is a left eigenvalue of G_1;
  - $\mathbf{r}$ is a *left eigenvector* of G_1 corresponding to the eigenvalue 1.
- Thus, it appears that finding ranks of the web pages would amount to finding the eigenvector of G_1 which corresponds to the eigenvalue 1 (providing that 1 is indeed one of the eigenvalues of G_1).

- But:
  - Why should 1 be one of the eigenvalues of G_1?
  - Even if 1 is indeed an eigenvalue of G_1, it could have multiplicity larger than one, so there could be several corresponding linearly independent eigenvectors $\mathbf{r}$,.
  - In fact, these two conditions might fail for such a G_1; we need to refine our model.

Problem: ordering webpages according to their importance

- Note that matrix G_1 of size $M \times M$ is a sparse matrix.
- Given that $\mathbf{r}$ multiplies [illegible] the left, matrix equation $\mathbf{r}^{\,T} = \mathbf{r}^{\,T} G_1$ simply says that:
  - 1 is a left eigen[illegible];
  - $\mathbf{r}$ is a *left eigen*[illegible] G_1 corresponding to the eigenvalue 1.
- Thus, it appears that finding ranks of the web pages would amount to finding the eigenvector of G_1 which corresponds to the eigenvalue 1 (providing that 1 is indeed one of the eigenvalues of G_1).

- But:
  - Why should 1 be one of the eigenvalues of G_1?
  - Even if 1 is indeed an eigenvalue of G_1, it could have multiplicity larger than one, so there could be several corresponding linearly independent eigenvectors $\mathbf{r}$,.
  - In fact, these two conditions might fail for such a G_1; we need to refine our model.

Problem: ordering webpages according to their importance

- Note that matrix G_1 of size $M \times M$ is a sparse matrix.

- Given that $\mathbf{r}$ multiplies G_1 from the left, matrix equation $\mathbf{r}^T = \mathbf{r}^T G_1$ simply says that:
  - 1 is a left eigenvalue of G_1;
  - $\mathbf{r}$ is a *left eigenvector* of G_1 corresponding to the eigenvalue 1.
- Thus, it appears that finding ranks of the web pages would amount to finding the eigenvector of G_1 which corresponds to the eigenvalue 1 (providing that 1 is indeed one of the eigenvalues of G_1).

- But:
  - Why should 1 be one of the eigenvalues of G_1?
  - Even if 1 is indeed an eigenvalue of G_1, it could have multiplicity larger than one, so there could be several corresponding linearly independent eigenvectors $\mathbf{r}$,.
  - In fact, these two conditions might fail for such a G_1; we need to refine our model.

Problem: ordering webpages according to their importance

- Note that matrix G_1 of size $M \times M$ is a sparse matrix.
- Given that $\mathbf{r}$ multiplies G_1 from the left, matrix equation $\mathbf{r}^T = \mathbf{r}^T G_1$ simply says that:
  - 1 is a left eigenvalue of G_1;
  - $\mathbf{r}$ is a *left eigenvector* of G_1 corresponding to the eigenvalue 1.
- Thus, it appears that finding ranks of the web pages would amount to finding the eigenvector of G_1 which corresponds to the eigenvalue 1 (providing that 1 is indeed one of the eigenvalues of G_1).
- But:
  - Why should 1 be one of the eigenvalues of G_1?
  - Even if 1 is indeed an eigenvalue of G_1, it could have multiplicity larger than one, so there could be several corresponding linearly independent eigenvectors $\mathbf{r}$,.
  - In fact, these two conditions might fail for such a G_1; we need to refine our model.

Problem: ordering webpages according to their importance

- Note that matrix G_1 of size $M \times M$ is a sparse matrix.

- Given that $\mathbf{r}$ multiplies G_1 from the left, matrix equation $\mathbf{r}^T = \mathbf{r}^T G_1$ simply says that:
  - 1 is a left eigenvalue of G_1;
  - $\mathbf{r}$ is a *left eigenvector* of G_1 corresponding to the eigenvalue 1.
- Thus, it appears that finding ranks of the web pages would amount to finding the eigenvector of G_1 which corresponds to the eigenvalue 1 (providing that 1 is indeed one of the eigenvalues of G_1).

- But:
  - Why should 1 be one of the eigenvalues of G_1?
  - Even if 1 is indeed an eigenvalue of G_1, it could have multiplicity larger than one, so there could be several corresponding linearly independent eigenvectors $\mathbf{r}$,.
  - In fact, these two conditions might fail for such a G_1; we need to refine our model.

Problem: ordering webpages according to their importance

- Note that matrix G_1 of size $M \times M$ is a sparse matrix.
- Given that $\mathbf{r}$ multiplies G_1 from the left, matrix equation $\mathbf{r}^T = \mathbf{r}^T G_1$ simply says that:
  - 1 is a left eigenvalue of G_1;
  - $\mathbf{r}$ is a *left eigenvector* of G_1 corresponding to the eigenvalue 1.
- Thus, it appears that finding ranks of the web pages would amount to finding the eigenvector of G_1 which corresponds to the eigenvalue 1 (providing that 1 is indeed one of the eigenvalues of G_1).
- But:
  - Why should 1 be one of the eigenvalues of G_1?
  - Even if 1 is indeed an eigenvalue of G_1, it could have multiplicity larger than one, so there could be several corresponding linearly independent eigenvectors $\mathbf{r}$,.
  - In fact, these two conditions might fail for such a G_1; we need to refine our model.

Problem: ordering webpages according to their importance

- Note that matrix G_1 of size $M \times M$ is a sparse matrix.

- Given that $\mathbf{r}$ multiplies G_1 from the left, matrix equation $\mathbf{r}^T = \mathbf{r}^T G_1$ simply says that:
  - 1 is a left eigenvalue of G_1;
  - $\mathbf{r}$ is a *left eigenvector* of G_1 corresponding to the eigenvalue 1.
- Thus, it appears that finding ranks of the web pages would amount to finding the eigenvector of G_1 which corresponds to the eigenvalue 1 (providing that 1 is indeed one of the eigenvalues of G_1).

- But:
  - Why should 1 be one of the eigenvalues of G_1?

  - Even if 1 is indeed an eigenvalue of G_1, it could have multiplicity larger than one, so there could be several corresponding linearly independent eigenvectors $\mathbf{r}$,.

  - In fact, these two conditions might fail for such a G_1; we need to refine our model.

Problem: ordering webpages according to their importance

- Note that matrix G_1 of size $M \times M$ is a sparse matrix.
- Given that $\mathbf{r}$ multiplies [illegible] the left, matrix equation $\mathbf{r}^T = \mathbf{r}^T G_1$ simply says that:
  - 1 is a left eigenv[illegible];
  - $\mathbf{r}$ is a *left eigenvector* of G_1 corresponding to the eigenvalue 1.
- Thus, it appears that finding ranks of the web pages would amount to finding the eigenvector of G_1 which corresponds to the eigenvalue 1 (providing that 1 is indeed one of the eigenvalues of G_1).
- But:
  - Why should 1 be one of the eigenvalues of G_1?
  - Even if 1 is indeed an eigenvalue of G_1, it could have multiplicity larger than one, so there could be several corresponding linearly independent eigenvectors $\mathbf{r}$,.
  - In fact, these two conditions might fail for such a G_1; we need to refine our model.

The Random Surfer heuristics

- Let us consider a web surfer which starts at a random web page and then just follows the links on these webpages by clicking on randomly chosen links.

- Assume he is doing this for a very long time of $T >> 10^{10}$ many clicks in total.

- For every web page P let $N(P)$ be the number of times he visits that page.

- Intuitively, rank of each web page P should equal to the ratio $N(P)/T$.

- Why is this so?

- Problems with this approach:
  - What happens if he arrives to a webpage without any outgoing links?
  - More generally, what happens if he gets in an “island” of web pages that point at each other but have no outgoing links to pages outside such an “island”?
  - If such an “island” is sufficiently large, the surfer might not even notice that he is in a trap without exit links.

The Random Surfer heuristics

- Let us consider a web surfer which starts at a random web page and then just follows the links on these webpages by clicking on randomly chosen links.
- Assume he is doing thi[illegible] long time of $T >> 10^{10}$ many clicks in total.
- For every web page P [illegible]e the number of times he visits that page.
- Intuitively, rank of each web page P should equal to the ratio $N(P)/T$.
- Why is this so?
- Problems with this approach:
  - What happens if he arrives to a webpage without any outgoing links?
  - More generally, what happens if he gets in an "island" of web pages that point at each other but have no outgoing links to pages outside such an "island"?
  - If such an "island" is sufficiently large, the surfer might not even notice that he is in a trap without exit links.

The Random Surfer heuristics

- Let us consider a web surfer which starts at a random web page and then just follows the links on these webpages by clicking on randomly chosen links.
- Assume he is doing thi[...] long time of $T >> 10^{10}$ many clicks in total.
- For every web page P [...]e the number of times he visits that page.
- Intuitively, rank of each web page P should equal to the ratio $N(P)/T$.
- Why is this so?
- Problems with this approach:
  - What happens if he arrives to a webpage without any outgoing links?
  - More generally, what happens if he gets in an "island" of web pages that point at each other but have no outgoing links to pages outside such an "island"?
  - If such an "island" is sufficiently large, the surfer might not even notice that he is in a trap without exit links.

The Random Surfer heuristics

- Let us consider a web surfer which starts at a random web page and then just follows the links on these webpages by clicking on randomly chosen links.
- Assume he is doing thi[illegible] long time of $T >> 10^{10}$ many clicks in total.
- For every web page P [illegible]e the number of times he visits that page.
- Intuitively, rank of each web page P should equal to the ratio $N(P)/T$.
- Why is this so?
- Problems with this approach:
  - What happens if he arrives to a webpage without any outgoing links?
  - More generally, what happens if he gets in an "island" of web pages that point at each other but have no outgoing links to pages outside such an "island"?
  - If such an "island" is sufficiently large, the surfer might not even notice that he is in a trap without exit links.

The Random Surfer heuristics

- Let us consider a web surfer which starts at a random web page and then just follows the links on these webpages by clicking on randomly chosen links.
- Assume he is doing thi[illegible] long time of $T >> 10^{10}$ many clicks in total.
- For every web page P [illegible]e the number of times he visits that page.
- Intuitively, rank of each web page P should equal to the ratio $N(P)/T$.
- Why is this so?
- Problems with this approach:
  - What happens if he arrives to a webpage without any outgoing links?
  - More generally, what happens if he gets in an "island" of web pages that point at each other but have no outgoing links to pages outside such an "island"?
  - If such an "island" is sufficiently large, the surfer might not even notice that he is in a trap without exit links.

- Let us consider a web surfer which starts at a random web page and then just follows the links on these webpages by clicking on randomly chosen links.
- Assume he is doing thi[illegible] long time of $T >> 10^{10}$ many clicks in total.
- For every web page P [illegible]e the number of times he visits that page.
- Intuitively, rank of each web page P should equal to the ratio $N(P)/T$.
- Why is this so?
- Problems with this approach:
  - What happens if he arrives to a webpage without any outgoing links?
  - More generally, what happens if he gets in an "island" of web pages that point at each other but have no outgoing links to pages outside such an "island"?
  - If such an "island" is sufficiently large, the surfer might not even notice that he is in a trap without exit links.

The Random Surfer heuristics

- Let us consider a web surfer which starts at a random web page and then just follows the links on these webpages by clicking on randomly chosen links.
- Assume he is doing thi … long time of $T >> 10^{10}$ many clicks in total.
- For every web page P … e the number of times he visits that page.
- Intuitively, rank of each web page P should equal to the ratio $N(P)/T$.
- Why is this so?
- Problems with this approach:
  - What happens if he arrives to a webpage without any outgoing links?
  - More generally, what happens if he gets in an "island" of web pages that point at each other but have no outgoing links to pages outside such an "island"?
  - If such an "island" is sufficiently large, the surfer might not even notice that he is in a trap without exit links.

The Random Surfer heuristics

- Let us consider a web surfer which starts at a random web page and then just follows the links on these webpages by clicking on randomly chosen links.
- Assume he is doing this for a very long time of $T >> 10^{10}$ many clicks in total.
- For every web page P let $N(P)$ be the number of times he visits that page.
- Intuitively, rank of each web page P should equal to the ratio $N(P)/T$.
- Why is this so?
- Problems with this approach:
  - What happens if he arrives to a webpage without any outgoing links?
  - More generally, what happens if he gets in an "island" of web pages that point at each other but have no outgoing links to pages outside such an "island"?
  - If such an "island" is sufficiently large, the surfer might not even notice that he is in a trap without exit links.

The Random Surfer heuristics

- Let us consider a web surfer which starts at a random web page and then just follows the links on these webpages by clicking on randomly chosen links.
- Assume he is doing thi[illegible] long time of $T >> 10^{10}$ many clicks in total.
- For every web page P [illegible]e the number of times he visits that page.
- Intuitively, rank of each web page P should equal to the ratio $N(P)/T$.
- Why is this so?
- Problems with this approach:
  - What happens if he arrives to a webpage without any outgoing links?
  - More generally, what happens if he gets in an "island" of web pages that point at each other but have no outgoing links to pages outside such an "island"?
  - If such an "island" is sufficiently large, the surfer might not even notice that he is in a trap without exit links.

- To get out of large tra[illegible]his browser would have to have a very large memory to backtrack, [illegible]cking is not a good idea.
- Moreover, the statistical features of his web surfing should not depend on any particular choices of links the surfer makes;
- $N(P)/T$ should be approximately the same for every particular surfing history.
- That is to say that, if we let $T \to \infty$, the values $N(P)/T$ should converge, so that the page ranks eventually stay essentially the same, independent of his random choices of links to follow.

- To get out of large tra his browser would have to have a very large memory to backtrack, king is not a good idea.
- Moreover, the statistical features of his web surfing should not depend on any particular choices of links the surfer makes;

- $N(P)/T$ should be approximately the same for every particular surfing history.
- That is to say that, if we let $T \to \infty$, the values $N(P)/T$ should converge, so that the page ranks eventually stay essentially the same, independent of his random choices of links to follow.

The Random Surfer heuristics

- To get out of large tra[illegible]his browser would have to have a very large memory to backtrack, [illegible]cking is not a good idea.
- Moreover, the statistical features of his web surfing should not depend on any particular choices of links the surfer makes;
- $N(P)/T$ should be approximately the same for every particular surfing history.
- That is to say that, if we let $T \to \infty$, the values $N(P)/T$ should converge, so that the page ranks eventually stay essentially the same, independent of his random choices of links to follow.

The Random Surfer heuristics

- To get out of large tra[illegible]his browser would have to have a very large memory to backtrack, [illegible]cking is not a good idea.

- Moreover, the statistical features of his web surfing should not depend on any particular choices of links the surfer makes;

- $N(P)/T$ should be approximately the same for every particular surfing history.

- That is to say that, if we let $T \to \infty$, the values $N(P)/T$ should converge, so that the page ranks eventually stay essentially the same, independent of his random choices of links to follow.

The Random Surfer heuristics

- Another, related heuristic is to have a surfer choose a fixed starting web page and then follow a large number M of links .

- When the surfer stops, we could register the webpage he stopped at.

- We now perform such an experiment a very large number T of times and calculate $\widetilde{N}(P)/T$ where $\widetilde{N}(P)$ is the number of times the surfer stopped at page P.

- Again, the values $\widetilde{N}(P)/T$ should converge and should be independent of surfer's choices, including what the starting page was.

- Another, related heuristic is to have a surfer choose a fixed starting web page and then follow a large number M of links .
- When the surfer stops, we could register the webpage he stopped at.

- We now perform such an experiment a very large number T of times and calculate $\widetilde{N}(P)/T$ where $\widetilde{N}(P)$ is the number of times the surfer stopped at page P.
- Again, the values $\widetilde{N}(P)/T$ should converge and should be independent of surfer's choices, including what the starting page was.

The Random Surfer heuristics

- Another, related heuri[illegible] have a surfer choose a fixed starting web page and then follow a large [illegible] of links .

- When the surfer stops, we could register the webpage he stopped at.

- We now perform such an experiment a very large number T of times and calculate $\widetilde{N}(P)/T$ where $\widetilde{N}(P)$ is the number of times the surfer stopped at page P.

- Again, the values $\widetilde{N}(P)/T$ should converge and should be independent of surfer's choices, including what the starting page was.

The Random Surfer heuristics

- Another, related heuri[illegible]ave a surfer choose a fixed starting web page and then follow a large[illegible] M of links .

- When the surfer stops, we could register the webpage he stopped at.

- We now perform such an experiment a very large number T of times and calculate $\widetilde{N}(P)/T$ where $\widetilde{N}(P)$ is the number of times the surfer stopped at page P.

- Again, the values $\widetilde{N}(P)/T$ should converge and should be independent of surfer's choices, including what the starting page was.

- Clearly, for some graphs such ratio cannot converge; for example if graph of websites is bipartite, th[illegible] T would depend on whether M is even or odd.

Figure: A “bipartite internet”

The Random Surfer heuristics

- What can prevent convergence of the quantities $N(P)/T$ or $\widetilde{N}(P)/T$?
- Remarkably, there are only two types of problems:
  1. arriving at “dan[illegible] pages” (web pages without any outgoing links) and more [illegible]aps, which are “islands” of webpages with incoming links f[illegible]tside such islands but without links leaving these isl[illegible]
  2. “periodic webpages” (web pages such that the length of every cycle of hyperlinks containing them is divisible by some fixed integer k; an example are the bipartite graphs where $k = 2$.)
- Both of these problems are solved by making our model more realistic!
- We refine our Random Surfer model by letting him every now and then to get bored with following links on pages he visits and instead letting him jump to a randomly chosen webpage.
- It turns out that this is all we need to have a suitable model which guarantees that both the values $N(P)/T$ and $\widetilde{N}(P)/T$ will converge to the same magnitude which is independent of particular choices made during such an “impatient” random surfing of the web!

The Random Surfer heuristics

- What can prevent convergence of the quantities $N(P)/T$ or $\widetilde{N}(P)/T$?
- Remarkably, there are only two types of problems:
  1. arriving at “dan[illegible] pages” (web pages without any outgoing links) and more [illegible]aps, which are “islands” of webpages with incoming links f[illegible]tside such islands but without links leaving these isl[illegible]
  2. “periodic webpages” (web pages such that the length of every cycle of hyperlinks containing them is divisible by some fixed integer k; an example are the bipartite graphs where $k = 2$.)
- Both of these problems are solved by making our model more realistic!
- We refine our Random Surfer model by letting him every now and then to get bored with following links on pages he visits and instead letting him jump to a randomly chosen webpage.
- It turns out that this is all we need to have a suitable model which guarantees that both the values $N(P)/T$ and $\widetilde{N}(P)/T$ will converge to the same magnitude which is independent of particular choices made during such an “impatient” random surfing of the web!

The Random Surfer heuristics

- What can prevent convergence of the quantities $N(P)/T$ or $\widetilde{N}(P)/T$?
- Remarkably, there are only two types of problems:
  1. arriving at “dar… pages” (web pages without any outgoing links) and more … aps, which are “islands” of webpages with incoming links f… utside such islands but without links leaving these isl…
  2. “periodic webpages” (web pages such that the length of every cycle of hyperlinks containing them is divisible by some fixed integer k; an example are the bipartite graphs where $k = 2$.)
- Both of these problems are solved by making our model more realistic!
- We refine our Random Surfer model by letting him every now and then to get bored with following links on pages he visits and instead letting him jump to a randomly chosen webpage.
- It turns out that this is all we need to have a suitable model which guarantees that both the values $N(P)/T$ and $\widetilde{N}(P)/T$ will converge to the same magnitude which is independent of particular choices made during such an “impatient” random surfing of the web!

The Random Surfer heuristics

- What can prevent convergence of the quantities $N(P)/T$ or $\widetilde{N}(P)/T$?
- Remarkably, there are only two types of problems:
  1. arriving at "dar[illegible] pages" (web pages without any outgoing links) and more [illegible]aps, which are "islands" of webpages with incoming links f[illegible]utside such islands but without links leaving these isl[illegible]
  2. "periodic webpages" (web pages such that the length of every cycle of hyperlinks containing them is divisible by some fixed integer k; an example are the bipartite graphs where $k = 2$.)
- Both of these problems are solved by making our model more realistic!
- We refine our Random Surfer model by letting him every now and then to get bored with following links on pages he visits and instead letting him jump to a randomly chosen webpage.
- It turns out that this is all we need to have a suitable model which guarantees that both the values $N(P)/T$ and $\widetilde{N}(P)/T$ will converge to the same magnitude which is independent of particular choices made during such an "impatient" random surfing of the web!

The Random Surfer heuristics

- What can prevent convergence of the quantities $N(P)/T$ or $\widetilde{N}(P)/T$?
- Remarkably, there are only two types of problems:
  1. arriving at “dar[illegible] pages” (web pages without any outgoing links) and more [illegible]aps, which are “islands” of webpages with incoming links f[illegible]utside such islands but without links leaving these isl[illegible]
  2. “periodic webpages” (web pages such that the length of every cycle of hyperlinks containing them is divisible by some fixed integer k; an example are the bipartite graphs where $k = 2$.)
- Both of these problems are solved by making our model more realistic!
- We refine our Random Surfer model by letting him every now and then to get bored with following links on pages he visits and instead letting him jump to a randomly chosen webpage.
- It turns out that this is all we need to have a suitable model which guarantees that both the values $N(P)/T$ and $\widetilde{N}(P)/T$ will converge to the same magnitude which is independent of particular choices made during such an “impatient” random surfing of the web!

The Random Surfer heuristics

- What can prevent convergence of the quantities $N(P)/T$ or $\widetilde{N}(P)/T$?
- Remarkably, there are only two types of problems:
  1. arriving at “dangling pages” (web pages without any outgoing links) and more general traps, which are “islands” of webpages with incoming links from outside such islands but without links leaving these islands.
  2. “periodic webpages” (web pages such that the length of every cycle of hyperlinks containing them is divisible by some fixed integer k; an example are the bipartite graphs where $k = 2$.)
- Both of these problems are solved by making our model more realistic!
- We refine our Random Surfer model by letting him every now and then to get bored with following links on pages he visits and instead letting him jump to a randomly chosen webpage.
- It turns out that this is all we need to have a suitable model which guarantees that both the values $N(P)/T$ and $\widetilde{N}(P)/T$ will converge to the same magnitude which is independent of particular choices made during such an “impatient” random surfing of the web!

The Random Surfer heuristics

- What can prevent convergence of the quantities $N(P)/T$ or $\widetilde{N}(P)/T$?
- Remarkably, there are only two types of problems:
  1. arriving at “dar[illegible] pages” (web pages without any outgoing links) and more [illegible]aps, which are “islands” of webpages with incoming links f[illegible]utside such islands but without links leaving these isl[illegible]
  2. “periodic webpages” (web pages such that the length of every cycle of hyperlinks containing them is divisible by some fixed integer k; an example are the bipartite graphs where $k = 2$.)

- Both of these problems are solved by making our model more realistic!
- We refine our Random Surfer model by letting him every now and then to get bored with following links on pages he visits and instead letting him jump to a randomly chosen webpage.
- It turns out that this is all we need to have a suitable model which guarantees that both the values $N(P)/T$ and $\widetilde{N}(P)/T$ will converge to the same magnitude which is independent of particular choices made during such an “impatient” random surfing of the web!

The Random Surfer heuristics

- So we agree on the following behaviour of the random surfer:
  1. If he ends up at a "dangling webpage" without any outgoing links, he jumps to a randomly chosen webpage, with all web pages equally likely.
  2. Once he has reaches a web page, he can either choose with a probability α to follow a randomly chosen link on that webpage, or, with probability $1-\alpha$, he can jump to a randomly chosen webpage, again with all web pages equally likely.

- Our aim is to show that for every webpage P both $N(P)/T$ and $\widetilde{N}(P)/T$ converge to the same limit $\rho(P)$, regardless of the particular surfing history.
- Thus, such a limit $\rho(P)$ can be used as an indication of the importance of the webpage and is what is called the Google PageRank.
- We first want to produce a matrix representation of the behaviour of our random surfer.

The Random Surfer heuristics

- So we agree on the following behaviour of the random surfer:
  1. If he ends up at a "dangling webpage" without any outgoing links, he jumps to a randomly chosen webpage, with all web pages equally likely.
  2. Once he has reaches a web page, he can either choose with a probability α to follow a randomly chosen link on that webpage, or, with probability $1-\alpha$, he can jump to a randomly chosen webpage, again with all web pages equally likely.
- Our aim is to show that for every webpage P both $N(P)/T$ and $\widetilde{N}(P)/T$ converge to the same limit $\rho(P)$, regardless of the particular surfing history.
- Thus, such a limit $\rho(P)$ can be seen as an indication of the importance of the webpage and is what is called the Google PageRank.
- We first want to produce a matrix representation of the behaviour of our random surfer.

- So we agree on the following behaviour of the random surfer:
  1. If he ends up at a “dangling webpage” without any outgoing links, he jumps to a randomly chosen webpage, with all web pages equally likely.
  2. Once he has reaches a web page, he can either choose with a probability α to follow a randomly chosen link on that webpage, or, with probability $1-\alpha$, he can jump to a randomly chosen webpage, again with all web pages equally likely.

- Our aim is to show that for every webpage P both $N(P)/T$ and $\widetilde{N}(P)/T$ converge to the same limit $\rho(P)$, regardless of the particular surfing history.
- Thus, such a limit $\rho(P)$ can be taken as an indication of the importance of the webpage and is what is called the Google PageRank.
- We first want to produce a matrix representation of the behaviour of our random surfer.

The Random Surfer heuristics

- So we agree on the foll[illegible]aviour of the random surfer:
  1. If he ends up at [illegible]ng webpage" without any outgoing links, he jumps to a r[illegible]hosen webpage, with all web pages equally likely.
  2. Once he has reaches a web page, he can either choose with a probability α to follow a randomly chosen link on that webpage, or, with probability $1-\alpha$, he can jump to a randomly chosen webpage, again with all web pages equally likely.

- Our aim is to show that for every webpage P both $N(P)/T$ and $\widetilde{N}(P)/T$ converge to the same limit $\rho(P)$, regardless of the particular surfing history.
- Thus, such a limit $\rho(P)$ can be taken as an indication of the importance of the webpage and is what is called the Google PageRank.
- We first want to produce a matrix representation of the behaviour of our random surfer.

- So we agree on the following behaviour of the random surfer:
  1. If he ends up at a "dangling webpage" without any outgoing links, he jumps to a randomly chosen webpage, with all web pages equally likely.
  2. Once he has reaches a web page, he can either choose with a probability α to follow a randomly chosen link on that webpage, or, with probability $1-\alpha$, he can jump to a randomly chosen webpage, again with all web pages equally likely.
- Our aim is to show that for every webpage P both $N(P)/T$ and $\widetilde{N}(P)/T$ converge to the same limit $\rho(P)$, regardless of the particular surfing history.
- Thus, such a limit $\rho(P)$ can be taken as an indication of the importance of the webpage and is what is called the Google PageRank.
- We first want to produce a matrix representation of the behaviour of our random surfer.

The Random Surfer heuristics

- So we agree on the foll[illegible]aviour of the random surfer:
  1. If he ends up at [illegible]ng webpage" without any outgoing links, he jumps to a r[illegible]hosen webpage, with all web pages equally likely.
  2. Once he has reaches a web page, he can either choose with a probability α to follow a randomly chosen link on that webpage, or, with probability $1 - \alpha$, he can jump to a randomly chosen webpage, again with all web pages equally likely.

- Our aim is to show that for every webpage P both $N(P)/T$ and $\widetilde{N}(P)/T$ converge to the same limit $\rho(P)$, regardless of the particular surfing history.
- Thus, such a limit $\rho(P)$ can be taken as an indication of the importance of the webpage and is what is called the Google PageRank.
- We first want to produce a matrix representation of the behaviour of our random surfer.

- To produce a matrix representation of such modified surfing session let us consider again the **init**[illegible]ified matrix G_1 construed as **the matrix of probabilities** g_{ij} **to** [illegible] **page** i **to page** j:

$$
\begin{array}{r l}
 & \\
 & \\
\textit{dangling page} & \\
G_1 = & \\
 & \\
 & \\
 &
\end{array}
\begin{pmatrix}
\vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots \\
\dots 0\ 0\ 0 \dots & \dots 0\ 0\ 0\ 0\ 0 \dots & \dots 0\ 0\ 0\ 0\ 0 \dots \\
\vdots & \vdots & \vdots \\
\dots 0\ 0\ \frac{1}{\#(P_i)}\ 0\ 0 \dots & \dots 0\ 0\ \frac{1}{\#(P_i)}\ 0\ 0 \dots & \dots 0\ 0\ \frac{1}{\#(P_i)}\ 0\ 0 \dots \\
\vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots
\end{pmatrix}
$$

The Random Surfer heuristics

- After fixing dangling webpages we get a new matrix G_2 which looks as follows:

$$G_2 = \begin{pmatrix} \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \ldots \frac{1}{M}\ \frac{1}{M}\ \frac{1}{M}\ \frac{1}{M}\ \frac{1}{M} \ldots & \ldots \frac{1}{M}\ \frac{1}{M}\ \frac{1}{M}\ \frac{1}{M}\ \frac{1}{M} \ldots & \ldots \frac{1}{M}\ \frac{1}{M}\ \frac{1}{M}\ \frac{1}{M}\ \frac{1}{M} \ldots \\ \vdots & \vdots & \vdots \\ \ldots 0\ 0\ \frac{1}{\#(P_i)}\ 0\ 0 \ldots & \ldots 0\ 0\ \frac{1}{\#(P_i)}\ 0\ 0 \ldots & \ldots 0\ 0\ \frac{1}{\#(P_i)}\ 0\ 0 \ldots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \end{pmatrix}$$

- Such a matrix is **row stochastic**, meaning that each row sums up to 1.

The Random Surfer heuristics

- After fixing dangling webpages we get a new matrix G_2 which looks as follows:

$$G_2 = \begin{pmatrix} & \vdots & & \vdots & & \vdots & \\ & \vdots & & \vdots & & \vdots & \\ \ldots \frac{1}{M}\ \frac{1}{M}\ \frac{1}{M}\ \frac{1}{M}\ \frac{1}{M} \ldots & & \ldots \frac{1}{M}\ \frac{1}{M}\ \frac{1}{M}\ \frac{1}{M}\ \frac{1}{M} \ldots & & \ldots \frac{1}{M}\ \frac{1}{M}\ \frac{1}{M}\ \frac{1}{M}\ \frac{1}{M} \ldots \\ & \vdots & & \vdots & & \vdots & \\ \ldots 0\ 0\ \frac{1}{\#(P_i)}\ 0\ 0 \ldots & & \ldots 0\ 0\ \frac{1}{\#(P_i)}\ 0\ 0 \ldots & & \ldots 0\ 0\ \frac{1}{\#(P_i)}\ 0\ 0 \ldots \\ & \vdots & & \vdots & & \vdots & \\ & \vdots & & \vdots & & \vdots & \end{pmatrix}$$

- Such a matrix is **row stochastic**, meaning that each row sums up to 1.

The Random Surfer heuristics

- We now add **teleportat[...]**domly chosen webpage:

$$G = \begin{pmatrix} & & \vdots & & & & \vdots & & & \\ \cdots \frac{1}{M} & \frac{1}{M} & \frac{1}{M} & \frac{1}{M} & \cdots & \frac{1}{M} & \frac{1}{M} & \frac{1}{M} & \frac{1}{M} & \frac{1}{M} \cdots \\ & & \vdots & & & & \vdots & & & \\ \cdots \frac{1-\alpha}{M} & \frac{1-\alpha}{M} & \frac{\alpha}{\#(P_i)} + \frac{1-\alpha}{M} & \frac{1-\alpha}{M} \cdots & \cdots & \frac{1-\alpha}{M} & \frac{\alpha}{\#(P_i)} + \frac{1-\alpha}{M} & \frac{1-\alpha}{M} & \frac{1-\alpha}{M} & \frac{1-\alpha}{M} \cdots \\ & & \vdots & & & & \vdots & & & \end{pmatrix}$$

- The last transformation [...] corresponding to dangling webpages: $\alpha/M + (1-\alpha)/M = 1/M$.

- We now add **teleportation** to a randomly chosen webpage:

$$G = \begin{pmatrix} & & & \vdots & & & & & \vdots & & & \\ & & & \vdots & & & & & \vdots & & & \\ \cdots \frac{1}{M} & \frac{1}{M} & & \frac{1}{M} & \frac{1}{M} & \cdots & \cdots & \frac{1}{M} & \frac{1}{M} & \frac{1}{M} & \frac{1}{M} & \frac{1}{M} \cdots \\ & & & \vdots & & & & & \vdots & & & \\ \cdots \frac{1-\alpha}{M} & \frac{1-\alpha}{M} & & \frac{\alpha}{\#(P_i)} + \frac{1-\alpha}{M} & \frac{1-\alpha}{M} & \cdots & \cdots & \frac{1-\alpha}{M} & \frac{1-\alpha}{M} & \frac{\alpha}{\#(P_i)} + \frac{1-\alpha}{M} & \frac{1-\alpha}{M} & \frac{1-\alpha}{M} \cdots \\ & & & \vdots & & & & & \vdots & & & \\ & & & \vdots & & & & & \vdots & & & \end{pmatrix}$$

- The last transformation does not change the rows corresponding to dangling webpages: $\alpha/M + (1-\alpha)/M = 1/M$.

- We now add **teleportation** to a randomly chosen webpage:

$$G = \begin{pmatrix}
 & & \vdots & & & & \vdots & & \\
 & & \vdots & & & & \vdots & & \\
\cdots \frac{1}{M} & \frac{1}{M} & \frac{1}{M} & \frac{1}{M} \cdots & \cdots \frac{1}{M} & \frac{1}{M} & \frac{1}{M} & \frac{1}{M} & \frac{1}{M} \cdots \\
 & & \vdots & & & & \vdots & & \\
\cdots \frac{1-\alpha}{M} & \frac{1-\alpha}{M} & \frac{\alpha}{\#(P_i)} + \frac{1-\alpha}{M} & \frac{1-\alpha}{M} \cdots & \cdots \frac{1-\alpha}{M} & \frac{1-\alpha}{M} & \frac{\alpha}{\#(P_i)} + \frac{1-\alpha}{M} & \frac{1-\alpha}{M} & \frac{1-\alpha}{M} \cdots \\
 & & \vdots & & & & \vdots & & \\
 & & \vdots & & & & \vdots & &
\end{pmatrix}$$

- The last transformation does not change the rows corresponding to dangling webpages: $\alpha/M + (1-\alpha)/M = 1/M$.

The Random Surfer heuristics

- The first fix:

$$
G_2 = G_1 + \begin{pmatrix}
0 & \ldots 0\ 0\ 0 \ldots & \ldots 0\ 0\ 0 \ldots & 0 \\
 & \vdots & \vdots & \\
0 & \ldots 0\ 0\ 0 \ldots & \ldots 0\ 0\ 0 \ldots & 0 \\
\frac{1}{M} & \ldots \frac{1}{M}\ \frac{1}{M}\ \frac{1}{M} \ldots & \ldots \frac{1}{M}\ \frac{1}{M}\ \frac{1}{M} \ldots & \frac{1}{M} \\
0 & \ldots 0\ 0\ 0 \ldots & \ldots 0\ 0\ 0 \ldots & 0 \\
 & \vdots & \vdots & \\
0 & \ldots 0\ 0\ 0 \ldots & \ldots 0\ 0\ 0 \ldots & 0 \\
\frac{1}{M} & \ldots \frac{1}{M}\ \frac{1}{M}\ \frac{1}{M} \ldots & \ldots \frac{1}{M}\ \frac{1}{M}\ \frac{1}{M} \ldots & \frac{1}{M} \\
0 & \ldots 0\ 0\ 0 \ldots & \ldots 0\ 0\ 0 \ldots & 0 \\
 & \vdots & \vdots & \\
0 & \ldots 0\ 0\ 0 \ldots & \ldots 0\ 0\ 0 \ldots & 0
\end{pmatrix}
$$

- Let d be 1 at positions i which correspond to dangling webpages and 0 elsewhere;

$$\mathbf{d}^{\intercal} = (0 \ldots 0\ 1\ 0 \ldots 0\ 1\ 0 \ldots 0\ 1\ 0 \ldots 0)$$

- Let $\mathbf{e}$ be 1 everywhere: $\mathbf{e}^{\intercal} = (1\ 1\ \ldots\ldots\ 1)$.
- Then

$$G_2 = G_1 + \frac{1}{M}\,\mathbf{d}\,\mathbf{e}^{\intercal}$$

- The first fix:

$$
G_2 = G_1 + \begin{pmatrix}
0 & \dots 0\ 0\ 0 \dots & \dots 0\ 0\ 0 \dots & 0 \\
 & \vdots & \vdots & \\
 & \dots 0\ 0\ 0 \dots & \dots 0\ 0\ 0 \dots & 0 \\
 & \frac{1}{M}\ \frac{1}{M}\ \frac{1}{M} \dots & \dots \frac{1}{M}\ \frac{1}{M}\ \frac{1}{M} \dots & \frac{1}{M} \\
 & \dots 0\ 0\ 0 \dots & \dots 0\ 0\ 0 \dots & 0 \\
 & \vdots & \vdots & \\
0 & \dots 0\ 0\ 0 \dots & \dots 0\ 0\ 0 \dots & 0 \\
\frac{1}{M} & \dots \frac{1}{M}\ \frac{1}{M}\ \frac{1}{M} \dots & \dots \frac{1}{M}\ \frac{1}{M}\ \frac{1}{M} \dots & \frac{1}{M} \\
0 & \dots 0\ 0\ 0 \dots & \dots 0\ 0\ 0 \dots & 0 \\
 & \vdots & \vdots & \\
0 & \dots 0\ 0\ 0 \dots & \dots 0\ 0\ 0 \dots & 0
\end{pmatrix}
$$

- Let d be 1 at positions i which correspond to dangling webpages and 0 elsewhere;

$$\mathbf{d}^\mathsf{T} = (0 \ \dots \ 0\ 1\ 0 \dots 0\ 1\ 0 \dots \ 0\ 1\ 0 \ \dots 0)$$

- Let $\mathbf{e}$ be 1 everywhere: $\mathbf{e}^\mathsf{T} = (1\ 1\ \dots\dots\ 1)$.
- Then

$$G_2 = G_1 + \frac{1}{M}\,\mathbf{d}\,\mathbf{e}^\mathsf{T}$$

- The first fix:

$$G_2 = G_1 + \begin{pmatrix}
0 & \dots 0 & 0 & 0 \dots & \dots 0 & 0 & 0 \dots & 0 \\
 & & \vdots & & & \vdots & & \\
0 & \dots 0 & 0 & 0 \dots & \dots 0 & 0 & 0 \dots & 0 \\
\frac{1}{M} & \dots \frac{1}{M} & \frac{1}{M} & \frac{1}{M} \dots & \dots \frac{1}{M} & \frac{1}{M} & \frac{1}{M} \dots & \frac{1}{M} \\
0 & \dots 0 & 0 & 0 \dots & \dots 0 & 0 & 0 \dots & 0 \\
 & & \vdots & & & \vdots & & \\
0 & \dots 0 & 0 & 0 \dots & \dots 0 & 0 & 0 \dots & 0 \\
\frac{1}{M} & \dots \frac{1}{M} & \frac{1}{M} & \frac{1}{M} \dots & \dots \frac{1}{M} & \frac{1}{M} & \frac{1}{M} \dots & \frac{1}{M} \\
0 & \dots 0 & 0 & 0 \dots & \dots 0 & 0 & 0 \dots & 0 \\
 & & \vdots & & & \vdots & & \\
0 & \dots 0 & 0 & 0 \dots & \dots 0 & 0 & 0 \dots & 0
\end{pmatrix}$$

- Let d be 1 at positions i which correspond to dangling webpages and 0 elsewhere;

$$\mathbf{d}^{\mathsf{T}} = (0 \ \dots \ 0 \ 1 \ 0 \ \dots \ 0 \ 1 \ 0 \dots \ 0 \ 1 \ 0 \ \dots 0)$$

- Let $\mathbf{e}$ be 1 everywhere: $\mathbf{e}^{\mathsf{T}} = (1 \ 1 \ \dots\dots \ 1)$.
- Then

$$G_2 = G_1 + \frac{1}{M}\,\mathbf{d}\,\mathbf{e}^{\mathsf{T}}$$

The Random Surfer heuristics

- The first fix:

$$G_2 = G_1 + \begin{pmatrix} 0 & \dots 0\ 0\ 0 \dots & \dots 0\ 0\ 0 \dots & 0 \\ & \vdots & \vdots & \\ 0 & \dots 0\ 0\ 0 \dots & \dots 0\ 0\ 0 \dots & 0 \\ \frac{1}{M} & \dots \frac{1}{M}\ \frac{1}{M}\ \frac{1}{M} \dots & \dots \frac{1}{M}\ \frac{1}{M}\ \frac{1}{M} \dots & \frac{1}{M} \\ 0 & \dots 0\ 0\ 0 \dots & \dots 0\ 0\ 0 \dots & 0 \\ & \vdots & \vdots & \\ 0 & \dots 0\ 0\ 0 \dots & \dots 0\ 0\ 0 \dots & 0 \\ \frac{1}{M} & \dots \frac{1}{M}\ \frac{1}{M}\ \frac{1}{M} \dots & \dots \frac{1}{M}\ \frac{1}{M}\ \frac{1}{M} \dots & \frac{1}{M} \\ 0 & \dots 0\ 0\ 0 \dots & \dots 0\ 0\ 0 \dots & 0 \\ & \vdots & \vdots & \\ 0 & \dots 0\ 0\ 0 \dots & \dots 0\ 0\ 0 \dots & 0 \end{pmatrix}$$

- Let d be 1 at positions i which correspond to dangling webpages and 0 elsewhere;

$$\mathbf{d}^{\mathsf{T}} = (0\ \dots\ 0\ 1\ 0\ \dots\ 0\ 1\ 0\ \dots\ 0\ 1\ 0\ \dots 0)$$

- Let $\mathbf{e}$ be 1 everywhere: $\mathbf{e}^{\mathsf{T}} = (1\ 1\ \dots\dots\ 1)$.
- Then

$$G_2 = G_1 + \frac{1}{M}\,\mathbf{d}\,\mathbf{e}^{\mathsf{T}}$$

- The product of a vector (i.e., a matrix $M \times 1$) and a transposed vector (a matrix $1 \times M$) such as $\mathbf{d}\,\mathbf{e}^\intercal$ is a matrix of size $M \times M$, and such a product is called *the outer product* or *the tensor product* of two vectors.
- We now get the final random surfer matrix:

$$G = \alpha G_2 + \frac{1-\alpha}{M}\,\mathbf{e}\,\mathbf{e}^\intercal = \alpha\left(G_1 + \frac{1}{M}\,\mathbf{d}\,\mathbf{e}^\intercal\right) + \frac{1-\alpha}{M}\,\mathbf{e}\,\mathbf{e}^\intercal, \tag{5}$$

- Note that G is still raw stochastic because in each non-dangling row we have $\#(P_i)$ many entries $\frac{\alpha}{\#(P_i)}$ totalling α and M entries $\frac{1-\alpha}{M}$ totalling $1-\alpha$, so all together $\alpha + 1 - \alpha = 1$.
- G is no longer sparse, but the vector - matrix product $\mathbf{x}^\intercal G$ for vectors $\mathbf{x}$ whose coordinates sum up to 1 can be decomposed as follows:

$$\begin{aligned}
\mathbf{x}^\intercal G &= \alpha\left(\mathbf{x}^\intercal G_1 + \frac{1}{M}\,\mathbf{x}^\intercal\mathbf{d}\,\mathbf{e}^\intercal\right) + \frac{1-\alpha}{M}\,\mathbf{x}^\intercal\mathbf{e}\,\mathbf{e}^\intercal \\
&= \alpha\mathbf{x}^\intercal G_1 + \frac{1}{M}\left(\alpha\mathbf{x}^\intercal\mathbf{d} + (1-\alpha)\mathbf{x}^\intercal\mathbf{e}\right)\mathbf{e}^\intercal \\
&= \alpha\mathbf{x}^\intercal G_1 + \frac{1}{M}\left(\alpha\mathbf{x}^\intercal\mathbf{d} + 1 - \alpha\right)\mathbf{e}^\intercal \\
&= \alpha\mathbf{x}^\intercal G_1 + \frac{1}{M}\left(1 - \alpha(1 - \mathbf{x}^\intercal\mathbf{d})\right)\mathbf{e}^\intercal,
\end{aligned}$$

The Random Surfer heuristics

- The product of a vector (i.e., a matrix $M \times 1$) and a transposed vector (a matrix $1 \times M$) such as $\mathbf{d}\,\mathbf{e}^{\intercal}$ is a matrix of size $M \times M$, and such a product is called *the outer product* or *the tensor product* of two vectors.
- We now get the final matrix G:

$$G = \alpha G_2 + \frac{1-\alpha}{M}\,\mathbf{e}\,\mathbf{e}^{\intercal} = \alpha\left(G_1 + \frac{1}{M}\,\mathbf{d}\,\mathbf{e}^{\intercal}\right) + \frac{1-\alpha}{M}\,\mathbf{e}\,\mathbf{e}^{\intercal}, \tag{5}$$

- Note that G is still row stochastic because in each non-dangling row we have $\#(P_i)$ many entries $\frac{\alpha}{\#(P_i)}$ totalling α and M entries $\frac{1-\alpha}{M}$ totalling $1-\alpha$, so all together $\alpha + 1 - \alpha = 1$.
- G is no longer sparse, but the vector - matrix product $\mathbf{x}^{\intercal} G$ for vectors $\mathbf{x}$ whose coordinates sum up to 1 can be decomposed as follows:

$$\begin{aligned}
\mathbf{x}^{\intercal} G &= \alpha\left(\mathbf{x}^{\intercal} G_1 + \frac{1}{M}\,\mathbf{x}^{\intercal}\,\mathbf{d}\,\mathbf{e}^{\intercal}\right) + \frac{1-\alpha}{M}\,\mathbf{x}^{\intercal}\,\mathbf{e}\,\mathbf{e}^{\intercal} \\
&= \alpha \mathbf{x}^{\intercal} G_1 + \frac{1}{M}\left(\alpha \mathbf{x}^{\intercal}\mathbf{d} + (1-\alpha)\mathbf{x}^{\intercal}\mathbf{e}\right)\mathbf{e}^{\intercal} \\
&= \alpha \mathbf{x}^{\intercal} G_1 + \frac{1}{M}\left(\alpha \mathbf{x}^{\intercal}\mathbf{d} + 1 - \alpha\right)\mathbf{e}^{\intercal} \\
&= \alpha \mathbf{x}^{\intercal} G_1 + \frac{1}{M}\left(1 - \alpha(1 - \mathbf{x}^{\intercal}\mathbf{d})\right)\mathbf{e}^{\intercal},
\end{aligned}$$

- The product of a vector (i.e., a matrix $M \times 1$) and a transposed vector (a matrix $1 \times M$) such as $\mathbf{d}\,\mathbf{e}^{\mathsf{T}}$ is a matrix of size $M \times M$, and such a product is called *the outer produc*[...] *product* of two vectors.
- We now get the final r[...]:

$$G = \alpha G_2 + \ldots = \alpha\left(G_1 + \frac{1}{M}\,\mathbf{d}\,\mathbf{e}^{\mathsf{T}}\right) + \frac{1-\alpha}{M}\,\mathbf{e}\,\mathbf{e}^{\mathsf{T}}, \tag{5}$$

- Note that G is still raw stochastic because in each non-dangling row we have $\#(P_i)$ many entries $\frac{\alpha}{\#(P_i)}$ totalling α and M entries $\frac{1-\alpha}{M}$ totalling $1-\alpha$, so all together $\alpha + 1 - \alpha = 1$.
- G is no longer sparse, but the vector - matrix product $\mathbf{x}^{\mathsf{T}} G$ for vectors $\mathbf{x}$ whose coordinates sum up to 1 can be decomposed as follows:

$$\begin{aligned}
\mathbf{x}^{\mathsf{T}} G &= \alpha\left(\mathbf{x}^{\mathsf{T}} G_1 + \frac{1}{M}\,\mathbf{x}^{\mathsf{T}}\,\mathbf{d}\,\mathbf{e}^{\mathsf{T}}\right) + \frac{1-\alpha}{M}\,\mathbf{x}^{\mathsf{T}}\,\mathbf{e}\,\mathbf{e}^{\mathsf{T}} \\
&= \alpha \mathbf{x}^{\mathsf{T}} G_1 + \frac{1}{M}\left(\alpha \mathbf{x}^{\mathsf{T}}\mathbf{d} + (1-\alpha)\mathbf{x}^{\mathsf{T}}\mathbf{e}\right)\mathbf{e}^{\mathsf{T}} \\
&= \alpha \mathbf{x}^{\mathsf{T}} G_1 + \frac{1}{M}\left(\alpha \mathbf{x}^{\mathsf{T}}\mathbf{d} + 1 - \alpha\right)\mathbf{e}^{\mathsf{T}} \\
&= \alpha \mathbf{x}^{\mathsf{T}} G_1 + \frac{1}{M}\left(1 - \alpha(1 - \mathbf{x}^{\mathsf{T}}\mathbf{d})\right)\mathbf{e}^{\mathsf{T}},
\end{aligned}$$

The Random Surfer heuristics

- The product of a vector (i.e., a matrix $M \times 1$) and a transposed vector (a matrix $1 \times M$) such as $\mathbf{d}\,\mathbf{e}^\intercal$ is a matrix of size $M \times M$, and such a product is called *the outer produc* *product* of two vectors.
- We now get the final r :

$$G = \alpha G_2 + \quad = \alpha\left(G_1 + \frac{1}{M}\,\mathbf{d}\,\mathbf{e}^\intercal\right) + \frac{1-\alpha}{M}\,\mathbf{e}\,\mathbf{e}^\intercal, \tag{5}$$

- Note that G is still raw stochastic because in each non-dangling row we have $\#(P_i)$ many entries $\frac{\alpha}{\#(P_i)}$ totalling α and M entries $\frac{1-\alpha}{M}$ totalling $1-\alpha$, so all together $\alpha + 1 - \alpha = 1$.
- G is no longer sparse, but the vector - matrix product $\mathbf{x}^\intercal G$ for vectors $\mathbf{x}$ whose coordinates sum up to 1 can be decomposed as follows:

$$\begin{aligned}
\mathbf{x}^\intercal G &= \alpha\left(\mathbf{x}^\intercal G_1 + \frac{1}{M}\,\mathbf{x}^\intercal\,\mathbf{d}\,\mathbf{e}^\intercal\right) + \frac{1-\alpha}{M}\,\mathbf{x}^\intercal\,\mathbf{e}\,\mathbf{e}^\intercal \\
&= \alpha\mathbf{x}^\intercal G_1 + \frac{1}{M}\left(\alpha\mathbf{x}^\intercal\mathbf{d} + (1-\alpha)\mathbf{x}^\intercal\mathbf{e}\right)\mathbf{e}^\intercal \\
&= \alpha\mathbf{x}^\intercal G_1 + \frac{1}{M}\left(\alpha\mathbf{x}^\intercal\mathbf{d} + 1 - \alpha\right)\mathbf{e}^\intercal \\
&= \alpha\mathbf{x}^\intercal G_1 + \frac{1}{M}\left(1 - \alpha(1 - \mathbf{x}^\intercal\mathbf{d})\right)\mathbf{e}^\intercal,
\end{aligned}$$

The Random Surfer heuristics

- Thus, the last expression

$$\frac{1}{M}\Big(1-\alpha(1-\mathbf{x}^{\mathsf{T}}\mathbf{d})\Big)\,\mathbf{e}^{\mathsf{T}}$$

can be computed very fast!

- Only the original matrix G_1 and vector $\mathbf{d}$ need to be stored.
- But why does G work??
- Why is there $\mathbf{x}$ such that $\mathbf{x}^{\mathsf{T}} = \mathbf{x}^{\mathsf{T}}G$ and why it is unique??
- The reason why G works is because our Random Surfer model is a special case of something much more general, a well behaved *Markov Chain*.

The Random Surfer heuristics

- Thus, the last expression

$$\frac{1}{M}\left(1-\alpha(1-\mathbf{x}^{\intercal}\mathbf{d})\right)\mathbf{e}^{\intercal}$$

can be computed very fast!

- Only the original matrix G_1 and vector $\mathbf{d}$ need to be stored.

- But why does G work??

- Why is there $\mathbf{x}$ such that $\mathbf{x}^{\intercal} = \mathbf{x}^{\intercal}G$ and why it is unique??

- The reason why G works is because our Random Surfer model is a special case of something much more general, a well behaved *Markov Chain.*

The Random Surfer heuristics

- Thus, the last expressi

$$\frac{1}{M}\left(1-\alpha(1-\mathbf{x}^{\intercal}\mathbf{d})\right)\mathbf{e}^{\intercal}$$

  can be computed very fast!

- Only the original matrix G_1 and vector $\mathbf{d}$ need to be stored.

- But why does G work??

- Why is there $\mathbf{x}$ such that $\mathbf{x}^{\intercal} = \mathbf{x}^{\intercal}G$ and why it is unique??

- The reason why G works is because our Random Surfer model is a special case of something much more general, a well behaved *Markov Chain.*

The Random Surfer heuristics

- Thus, the last expressi

$$\frac{1}{M}\left(1-\alpha(1-\mathbf{x}^{\intercal}\mathbf{d})\right)\mathbf{e}^{\intercal}$$

  can be computed very fast!

- Only the original matrix G_1 and vector $\mathbf{d}$ need to be stored.

- But why does G work??

- Why is there $\mathbf{x}$ such that $\mathbf{x}^{\intercal} = \mathbf{x}^{\intercal}G$ and why it is unique??

- The reason why G works is because our Random Surfer model is a special case of something much more general, a well behaved *Markov Chain.*

- Thus, the last expressi

$$\frac{1}{M}\Big(1-\alpha(1-\mathbf{x}^{\intercal}\mathbf{d})\Big)\,\mathbf{e}^{\intercal}$$

  can be computed very fast!

- Only the original matrix G_1 and vector $\mathbf{d}$ need to be stored.

- But why does G work??

- Why is there $\mathbf{x}$ such that $\mathbf{x}^{\intercal} = \mathbf{x}^{\intercal} G$ and why it is unique??

- The reason why G works is because our Random Surfer model is a special case of something much more general, a well behaved *Markov Chain*.

- A **Markov Chain** (also called a *Discrete Time Markov Process*) is given by
  - a set of states $S = \{P_1, \ldots, P_M\}$; (we only consider cases when S is finite)
  - a row stochastic matrix G.
- At every instant of discrete time $t = 0, 1, 2, \ldots$ the chain is in one of the states $X(t) = P_i \in S$; at the next instant $t + 1$ the state changes to another state $X(t + 1) = P_j$ in a random manner.
- If $X(t) = P_i$, the probability that $X(t+1) = P_j$ is given by the entry $g(i, j) = (G)_{i,j}$ of the matrix G.
- Note that the probability of entering state $X(t+1) = P_j$ from the previous state $X(t) = P_i$ DOES NOT depend on the way how the state P_i was reached.
- Thus, the Markov chains have no memory.
- The state $X(0)$ at which the Markov chain starts can also be random; let us denote the probability distribution of the initial state by $q^{(0)}$.
- Thus $\mathbf{q}^{(0)}(i)$ denotes the probability that the initial state was $X(0) = P_i$.

Markov Chains (Discrete Time Markov Processes)

- A **Markov Chain** (al… *Discrete Time Markov Process*) is given by
  - a set of states S … $_M$; (we only consider cases when S is finite)
  - a row stochastic … .
- At every instant of discrete time $t = 0, 1, 2, \ldots$ the chain is in one of the states $X(t) = P_i \in S$; at the next instant $t+1$ the state changes to another state $X(t+1) = P_j$ in a random manner.
- If $X(t) = P_i$, the probability that $X(t+1) = P_j$ is given by the entry $g(i,j) = (G)_{i,j}$ of the matrix G.
- Note that the probability … $= P_j$ from the previous state $X(t) = P_i$ DOES NOT depend on the way how the state P_i was reached.
- Thus, the Markov chains have no memory.
- The state $X(0)$ at which the Markov chain starts can also be random; let us denote the probability distribution of the initial state by $q^{(0)}$.
- Thus $\mathbf{q}^{(0)}(i)$ denotes the probability that the initial state was $X(0) = P_i$.

Markov Chains (Discrete Time Markov Processes)

- A **Markov Chain** (al[illegible] *Discrete Time Markov Process*) is given by
  - a set of states S[illegible]$_M$; (we only consider cases when S is finite)
  - a row stochastic [illegible].
- At every instant of discrete time $t = 0, 1, 2, \ldots$ the chain is in one of the states $X(t) = P_i \in S$; at the [illegible] 1 the state changes to another state $X(t+1) = P_j$ in a random manner.
- If $X(t) = P_i$, the proba[illegible] the entry $g(i,j) = (G)_{i,j}$ of the matrix G.
- Note that the probabil[illegible] P_j from the previous state $X(t) = P_i$ DOES NOT depend on the way how the state P_i was reached.
- Thus, the Markov chai[illegible] memory.
- The state $X(0)$ at which the Markov chain starts can also be random; let us denote the probability [illegible] of the initial state by $q^{(0)}$.
- Thus $\mathbf{q}^{(0)}(i)$ denotes the probability that the initial state was $X(0) = P_i$.

- A **Markov Chain** (al[illegible] *Discrete Time Markov Process*) is given by
  - a set of states S[illegible]$_M$; (we only consider cases when S is finite)
  - a row stochastic[illegible].
- At every instant of discrete time $t = 0, 1, 2, \ldots$ the chain is in one of the states $X(t) = P_i \in S$; at the next time instant $t+1$ the state changes to another state $X(t+1) = P_j$ in a random manner.
- If $X(t) = P_i$, the probability that $X(t+1) = P_j$ is given by the entry $g(i,j) = (G)_{i,j}$ of the matrix G.
- Note that the probability of transition to $X(t+1) = P_j$ from the previous state $X(t) = P_i$ DOES NOT depend on the way how the state P_i was reached.
- Thus, the Markov chains have no memory.
- The state $X(0)$ at which the Markov chain starts can also be random; let us denote the probability distribution of the initial state by $q^{(0)}$.
- Thus $\mathbf{q}^{(0)}(i)$ denotes the probability that the initial state was $X(0) = P_i$.

Markov Chains (Discrete Time Markov Processes)

- A **Markov Chain** (al[illegible] *Discrete Time Markov Process*) is given by
  - a set of states S [illegible] $_M$; (we only consider cases when S is finite)
  - a row stochastic [illegible].
- At every instant of discrete time $t = 0, 1, 2, \ldots$ the chain is in one of the states $X(t) = P_i \in S$; at the next time instant $t + 1$ the state changes to another state $X(t+1) = P_j$ in a random manner.
- If $X(t) = P_i$, the probability that $X(t+1) = P_j$ is given by the entry $g(i, j) = (G)_{i,j}$ of the matrix G.
- Note that the probability of entering state $X(t+1) = P_j$ from the previous state $X(t) = P_i$ DOES NOT depend on the way how the state P_i was reached.
- Thus, the Markov chains have no memory.
- The state $X(0)$ at which the Markov chain starts can also be random; let us denote the probability distribution of the initial state by $q^{(0)}$.
- Thus $\mathbf{q}^{(0)}(i)$ denotes the probability that the initial state was $X(0) = P_i$.

- A **Markov Chain** (al *Discrete Time Markov Process*) is given by
  - a set of states S $_M$; (we only consider cases when S is finite)
  - a row stochastic .
- At every instant of discrete time $t = 0, 1, 2, \ldots$ the chain is in one of the states $X(t) = P_i \in S$; at the next time instant $t+1$ the state changes to another state $X(t+1) = P_j$ in a random manner.
- If $X(t) = P_i$, the probability that $X(t+1) = P_j$ is given by the entry $g(i,j) = (G)_{i,j}$ of the matrix G.
- Note that the probability to enter state $X(t+1) = P_j$ from the previous state $X(t) = P_i$ DOES NOT depend on the way how the state P_i was reached.
- Thus, the Markov chains have no memory.
- The state $X(0)$ at which the Markov chain starts can also be random; let us denote the probability distribution of the initial state by $q^{(0)}$.
- Thus $\mathbf{q}^{(0)}(i)$ denotes the probability that the initial state was $X(0) = P_i$.

Markov Chains (Discrete Time Markov Processes)

- A **Markov Chain** (al[illegible] *Discrete Time Markov Process*) is given by
  - a set of states S[illegible]$_M$; (we only consider cases when S is finite)
  - a row stochastic[illegible].
- At every instant of discrete time $t = 0, 1, 2, \ldots$ the chain is in one of the states $X(t) = P_i \in S$; at the next time instant $t + 1$ the state changes to another state $X(t+1) = P_j$ in a random manner.
- If $X(t) = P_i$, the probability that $X(t+1) = P_j$ is given by the entry $g(i,j) = (G)_{i,j}$ of the matrix G.
- Note that the probability to enter state $X(t+1) = P_j$ from the previous state $X(t) = P_i$ DOES NOT depend on the way how the state P_i was reached.
- Thus, the Markov chains have no memory.
- The state $X(0)$ at which the Markov chain starts can also be random; let us denote the probability distribution of the initial state by $q^{(0)}$.
- Thus $\mathbf{q}^{(0)}(i)$ denotes the probability that the initial state was $X(0) = P_i$.

Markov Chains (Discrete Time Markov Processes)

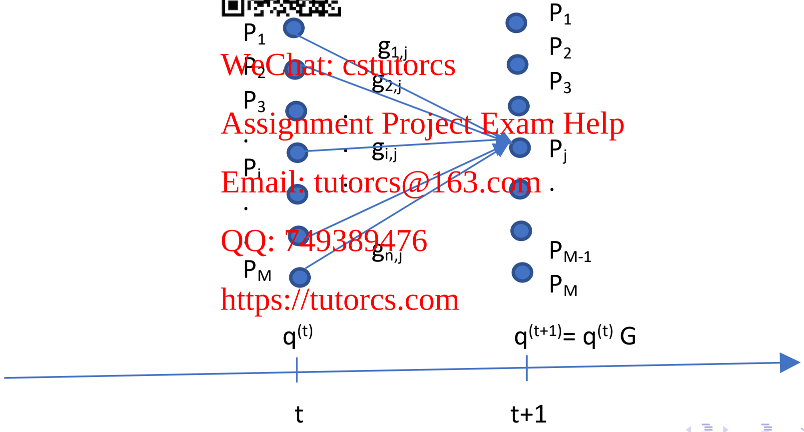
- A **Markov Chain** (al[illegible] *Discrete Time Markov Process*) is given by
  - a set of states S[illegible]$_M$; (we only consider cases when S is finite)
  - a row stochastic[illegible].
- At every instant of discrete time $t = 0, 1, 2, \ldots$ the chain is in one of the states $X(t) = P_i \in S$; at the next time instant $t+1$ the state changes to another state $X(t+1) = P_j$ in a random manner.
- If $X(t) = P_i$, the probability that $X(t+1) = P_j$ is given by the entry $g(i,j) = (G)_{i,j}$ of the matrix G.
- Note that the probability to enter state $X(t+1) = P_j$ from the previous state $X(t) = P_i$ DOES NOT depend on the way how the state P_i was reached.
- Thus, the Markov chains have no memory.
- The state $X(0)$ at which the Markov chain starts can also be random; let us denote the probability distribution of the initial state by $q^{(0)}$.
- Thus $\mathbf{q}^{(0)}(i)$ denotes the probability that the initial state was $X(0) = P_i$.

Markov Chains (Discrete Time Markov Processes)

- A **Markov Chain** (al[illegible] *Discrete Time Markov Process*) is given by
  - a set of states S[illegible]$_M$; (we only consider cases when S is finite)
  - a row stochastic[illegible].
- At every instant of discrete time $t = 0, 1, 2, \ldots$ the chain is in one of the states $X(t) = P_i \in S$; at the next time instant $t + 1$ the state changes to another state $X(t + 1) = P_j$ in a random manner.
- If $X(t) = P_i$, the probability that $X(t + 1) = P_j$ is given by the entry $g(i, j) = (G)_{i,j}$ of the matrix G.
- Note that the probability to enter state $X(t + 1) = P_j$ from the previous state $X(t) = P_i$ DOES NOT depend on the way how the state P_i was reached.
- Thus, the Markov chains have no memory.
- The state $X(0)$ at which the Markov chain starts can also be random; let us denote the probability distribution of the initial state by $q^{(0)}$.
- Thus $\mathbf{q}^{(0)}(i)$ denotes the probability that the initial state was $X(0) = P_i$.

- If $q^{(t)}$ is the probability distribution of states at an instant t then the probability distribution of the states at instant $t+1$ is uniquely determined:

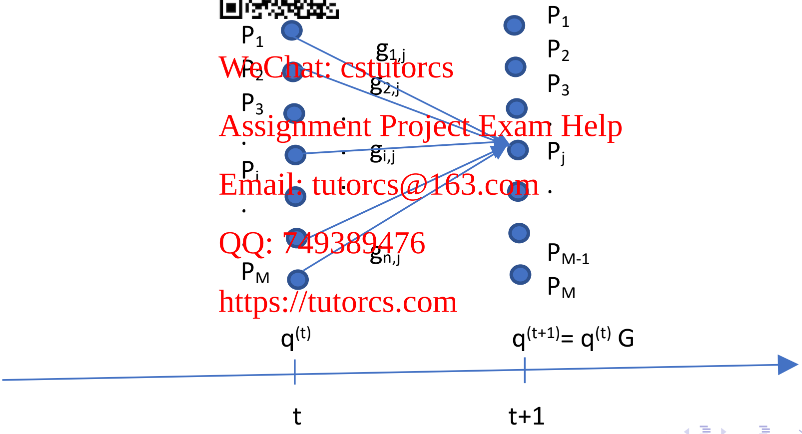
$$[illegible]) = \sum_{1 \le i \le M} \mathbf{q}^{(t)}(i) g_{i,j}$$



Markov Chains (Discrete Time Markov Processes)

- If $q^{(t)}$ is the probability distribution of states at an instant t then the probability distribution of the states at instant $t+1$ is uniquely determined:

$$\ldots) = \sum_{1 \le i \le M} \mathbf{q}^{(t)}(i) g_{i,j}$$



- Putting all of these equalities in matrix form we get

$$(\mathbf{q}^{(t+1)})^\top = (\mathbf{q}^{(t)})^\top G$$

- Note that the probability distribution $q^{(0)}$ of the initial state and matrix G uniquely determine the probability distribution of states at any future instant:

$$(\mathbf{q}^{(1)})^\top = (\mathbf{q}^{(0)})^\top G$$

$$(\mathbf{q}^{(2)})^\top = (\mathbf{q}^{(1)})^\top G = ((\mathbf{q}^{(0)})^\top G)\, G = (\mathbf{q}^{(0)})^\top G^2 \qquad \vdots$$

$$(\mathbf{q}^{(t)})^\top = (\dots(\mathbf{q}^{(0)})^\top G)G)\dots G = (\mathbf{q}^{(0)})^\top G^t$$

- Putting all of these equalities in matrix form we get

$$(\mathbf{q}^{(t+1)})^{\mathsf{T}} = (\mathbf{q}^{(t)})^{\mathsf{T}} G$$

- Note that the probability distribution $q^{(0)}$ of the initial state and matrix G uniquely determine the probability distribution of states at any future instant:

$$(\mathbf{q}^{(1)})^{\mathsf{T}} = (\mathbf{q}^{(0)})^{\mathsf{T}} G$$

$$(\mathbf{q}^{(2)})^{\mathsf{T}} = (\mathbf{q}^{(1)})^{\mathsf{T}} G = ((\mathbf{q}^{(0)})^{\mathsf{T}} G)\, G = (\mathbf{q}^{(0)})^{\mathsf{T}} G^2 \qquad \vdots$$

$$(\mathbf{q}^{(t)})^{\mathsf{T}} = (\ldots(\mathbf{q}^{(0)})^{\mathsf{T}} G) G) \ldots G = (\mathbf{q}^{(0)})^{\mathsf{T}} G^t$$

- Putting all of these equalities in matrix form we get

$$(\mathbf{q}^{(t+1)})^\top = (\mathbf{q}^{(t)})^\top G$$

- Note that the probability distribution $q^{(0)}$ of the initial state and matrix G uniquely determine the probability distribution of states at any future instant:

$$(\mathbf{q}^{(1)})^\top = (\mathbf{q}^{(0)})^\top G$$

$$(\mathbf{q}^{(2)})^\top = (\mathbf{q}^{(1)})^\top G = ((\mathbf{q}^{(0)})^\top G)\, G = (\mathbf{q}^{(0)})^\top G^2 \qquad \vdots$$

$$(\mathbf{q}^{(t)})^\top = (\dots(\mathbf{q}^{(0)})^\top G)G)\dots G = (\mathbf{q}^{(0)})^\top G^t$$

Markov Chains (Discrete Time Markov Processes)

- Putting all of these equalities in matrix form we get

$$(\mathbf{q}^{(t+1)})^\mathsf{T} = (\mathbf{q}^{(t)})^\mathsf{T} G$$

- Note that the probability distribution $q^{(0)}$ of the initial state and matrix G uniquely determine the probability distribution of states at any future instant:

$$(\mathbf{q}^{(1)})^\mathsf{T} = (\mathbf{q}^{(0)})^\mathsf{T} G$$

$$(\mathbf{q}^{(2)})^\mathsf{T} = (\mathbf{q}^{(1)})^\mathsf{T} G = ((\mathbf{q}^{(0)})^\mathsf{T} G)\, G = (\mathbf{q}^{(0)})^\mathsf{T} G^2 \qquad \vdots$$

$$(\mathbf{q}^{(t)})^\mathsf{T} = (\ldots(\mathbf{q}^{(0)})^\mathsf{T} G)G)\ldots G = (\mathbf{q}^{(0)})^\mathsf{T} G^t$$

- Putting all of these equalities in matrix form we get

$$(\mathbf{q}^{(t+1)})^\mathsf{T} = (\mathbf{q}^{(t)})^\mathsf{T} G$$

- Note that the probability distribution $q^{(0)}$ of the initial state and matrix G uniquely determine the probability distribution of states at any future instant:

$$(\mathbf{q}^{(1)})^\mathsf{T} = (\mathbf{q}^{(0)})^\mathsf{T} G$$

$$(\mathbf{q}^{(2)})^\mathsf{T} = (\mathbf{q}^{(1)})^\mathsf{T} G = ((\mathbf{q}^{(0)})^\mathsf{T} G)\, G = (\mathbf{q}^{(0)})^\mathsf{T} G^2 \qquad \vdots$$

$$(\mathbf{q}^{(t)})^\mathsf{T} = (\ldots(\mathbf{q}^{(0)})^\mathsf{T} G)G)\ldots G = (\mathbf{q}^{(0)})^\mathsf{T} G^t$$

Random Surfer model as a Markov Chain

- Our random surfer is an example of a Markov Chain:
  - States are "being on page P_i", so we have in total M states, one for each web page P_i, $1 \leq i \leq M$.
  - He starts from a randomly chosen starting web page.
  - Thus, $q^{(0)}(i) = \frac{1}{M}$ for all i, because all pages are equally likely to be the starting page.
  - If a webpage P_i is not a dangling webpage, he follows a link on the page P_i which points to another page P_j with probability $\frac{\alpha}{\#(P_i)}$, or he jumps directly to a page P_j with probability $\frac{1-\alpha}{M}$.
  - So if a page P_i points to a page P_j then $g_{i,j} = \frac{\alpha}{\#(P_i)} + \frac{1-\alpha}{M}$ (because he can also jump to P_j randomly).
  - If a page P_i does not point to a page P_j then $g_{i,j} = \frac{1-\alpha}{M}$ (because from P_i he can reach P_j only by jumping randomly).
  - If P_i is a dangling page then $g_{i,j} = \frac{1}{M}$ for every page P_j (including the same page P_i, for simplicity).

Random Surfer model as a Markov Chain

- Our random surfer is an example of a Markov Chain:
  - States are "being at page P_i", so we have in total M states, one for each web page P_i, $1 \leq i \leq M$.
  - He starts from a randomly chosen starting web page.
  - Thus, $q^{(0)}(i) = \frac{1}{M}$ for all i, because all pages are equally likely to be the starting page.
  - If a webpage P_i is not a dangling webpage, he follows a link on the page P_i which points to another page P_j with probability $\frac{\alpha}{\#(P_i)}$, or he jumps directly to a page P_j with probability $\frac{1-\alpha}{M}$.
  - So if a page P_i points to a page P_j then $g_{i,j} = \frac{\alpha}{\#(P_i)} + \frac{1-\alpha}{M}$ (because he can also jump to P_j randomly).
  - If a page P_i does not point to a page P_j then $g_{i,j} = \frac{1-\alpha}{M}$ (because from P_i he can reach P_j only by jumping randomly).
  - If P_i is a dangling page then $g_{i,j} = \frac{1}{M}$ for every page P_j (including the same page P_i, for simplicity).

Random Surfer model as a Markov Chain

- Our random surfer is an example of a Markov Chain:
  - States are "being on page P_i", so we have in total M states, one for each web page P_i, $1 \leq i \leq M$.
  - He starts from a randomly chosen starting web page.
  - Thus, $q^{(0)}(i) = \frac{1}{M}$ for all i, because all pages are equally likely to be the starting page.
  - If a webpage P_i is not a dangling webpage, he follows a link on the page P_i which points to another page P_j with probability $\frac{\alpha}{\#(P_i)}$, or he jumps directly to a page P_j with probability $\frac{1-\alpha}{M}$.
  - So if a page P_i points to a page P_j then $g_{i,j} = \frac{\alpha}{\#(P_i)} + \frac{1-\alpha}{M}$ (because he can also jump to P_j randomly).
  - If a page P_i does not point to a page P_j then $g_{i,j} = \frac{1-\alpha}{M}$ (because from P_i he can reach P_j only by jumping randomly).
  - If P_i is a dangling page then $g_{i,j} = \frac{1}{M}$ for every page P_j (including the same page P_i, for simplicity).

- Our random surfer is an example of a Markov Chain:
  - States are "being at page P_i", so we have in total M states, one for each web page P_i, $1 \leq i \leq M$.
  - He starts from a randomly chosen starting web page.
  - Thus, $q^{(0)}(i) = \frac{1}{M}$ for all i, because all pages are equally likely to be the starting page.

  - If a webpage P_i is not a dangling webpage, he follows a link on the page P_i which points to another page P_j with probability $\frac{\alpha}{\#(P_i)}$, or he jumps directly to a page P_j with probability $\frac{1-\alpha}{M}$.

  - So if a page P_i points to a page P_j then $g_{i,j} = \frac{\alpha}{\#(P_i)} + \frac{1-\alpha}{M}$ (because he can also jump to P_j randomly).

  - If a page P_i does not point to a page P_j then $g_{i,j} = \frac{1-\alpha}{M}$ (because from P_i he can reach P_j only by jumping randomly).

  - If P_i is a dangling page then $g_{i,j} = \frac{1}{M}$ for every page P_j (including the same page P_i, for simplicity).

Random Surfer model as a Markov Chain

- Our random surfer is an example of a Markov Chain:
  - States are "bein[illegible]age P_i", so we have in total M states, one for each web pa[illegible] $i \leq M$.
  - He starts from a [illegible] chosen starting web page.
  - Thus, $q^{(0)}(i) = \frac{1}{M}$ for all i, because all pages are equally likely to be the starting page.
  - If a webpage P_i is not a dangling webpage, he follows a link on the page P_i which points to another web page P_j with probability $\frac{\alpha}{\#(P_i)}$, or he jumps directly to a page P_j with probability $\frac{1-\alpha}{M}$.
  - So if a page P_i points to a page P_j then $g_{i,j} = \frac{\alpha}{\#(P_i)} + \frac{1-\alpha}{M}$ (because he can also jump to P_j randomly).
  - If a page P_i does not point to a page P_j then $g_{i,j} = \frac{1-\alpha}{M}$ (because from P_i he can reach P_j only by jumping randomly).
  - If P_i is a dangling page then $g_{i,j} = \frac{1}{M}$ for every page P_j (including the same page P_i, for simplicity).

Random Surfer model as a Markov Chain

- Our random surfer is an example of a Markov Chain:
  - States are "being on page P_i", so we have in total M states, one for each web page P_i, $1 \leq i \leq M$.
  - He starts from a randomly chosen starting web page.
  - Thus, $q^{(0)}(i) = \frac{1}{M}$ for all i, because all pages are equally likely to be the starting page.
  - If a webpage P_i is not a dangling webpage, he follows a link on the page P_i which points to another web page P_j with probability $\frac{\alpha}{\#(P_i)}$, or he jumps directly to a page P_j with probability $\frac{1-\alpha}{M}$.
  - So if a page P_i points to a page P_j then $g_{i,j} = \frac{\alpha}{\#(P_i)} + \frac{1-\alpha}{M}$ (because he can also jump to P_j randomly).
  - If a page P_i does not point to a page P_j then $g_{i,j} = \frac{1-\alpha}{M}$ (because from P_i he can reach P_j only by jumping randomly).
  - If P_i is a dangling page then $g_{i,j} = \frac{1}{M}$ for every page P_j (including the same page P_i, for simplicity).

- Our random surfer is an example of a Markov Chain:
  - States are "being on page P_i", so we have in total M states, one for each web page $1 \leq i \leq M$.
  - He starts from a randomly chosen starting web page.
  - Thus, $q^{(0)}(i) = \frac{1}{M}$ for all i, because all pages are equally likely to be the starting page.
  - If a webpage P_i is not a dangling webpage, he follows a link on the page P_i which points to another web page P_j with probability $\frac{\alpha}{\#(P_i)}$, or he jumps directly to a page P_j with probability $\frac{1-\alpha}{M}$.
  - So if a page P_i points to a page P_j then $g_{i,j} = \frac{\alpha}{\#(P_i)} + \frac{1-\alpha}{M}$ (because he can also jump to P_j randomly).
  - If a page P_i does not point to a page P_j then $g_{i,j} = \frac{1-\alpha}{M}$ (because from P_i he can reach P_j only by jumping randomly).
  - If P_i is a dangling page then $g_{i,j} = \frac{1}{M}$ for every page P_j (including the same page P_i, for simplicity).

Random Surfer model as a Markov Chain

- Our random surfer is an example of a Markov Chain:
  - States are "being on page P_i", so we have in total M states, one for each web page P_i, $1 \leq i \leq M$.
  - He starts from a randomly chosen starting web page.
  - Thus, $q^{(0)}(i) = \frac{1}{M}$ for all i, because all pages are equally likely to be the starting page.
  - If a webpage P_i is not a dangling webpage, he follows a link on the page P_i which points to another web page P_j with probability $\frac{\alpha}{\#(P_i)}$, or he jumps directly to a page P_j with probability $\frac{1-\alpha}{M}$.
  - So if a page P_i points to a page P_j then $g_{i,j} = \frac{\alpha}{\#(P_i)} + \frac{1-\alpha}{M}$ (because he can also jump to P_j randomly).
  - If a page P_i does not point to a page P_j then $g_{i,j} = \frac{1-\alpha}{M}$ (because from P_i he can reach P_j only by jumping randomly).
  - If P_i is a dangling page then $g_{i,j} = \frac{1}{M}$ for every page P_j (including the same page P_i, for simplicity).

Back to the general Markov Chains

- To a non-negative matrix $M \geq 0$ (i.e., $M_{i,j} \geq 0$ for all i, j) of size $n \times n$ we associate a direct[illegible] G with n vertices $V = \{P_1, P_2, \ldots, P_n\}$ and a directed edge [illegible] in case $M_{ij} > 0$.
- Such a graph has the [illegible] matrix $\widetilde{M} = \mathrm{sign}(M)$.
- **Lemma:** There is a path of length k from P_i to P_j in G just in case $(M^k)_{ij} > 0$. (Proof is an easy induction on k).
- **Homework:** Prove by induction on k that $(\widetilde{M}^k)_{i,j}$ is exactly the number of directed paths from P_i to P_j of length exactly k.
- **Definition:** We say that a directed graph G is *strongly connected* just in case for any two vertices P_i, P_j in G there exists a directed path in G from P_i to P_j (and thus also a directed path from P_j to P_i).
- **Definition:** A Markov chain is *irreducible* if the graph corresponding to its transition probabilities matrix is strongly connected.
- Google matrix induces a strongly connected graph.
- This is trivially true because, in fact, there is a directed edge between any two vertices.

Back to the general Markov Chains

- To a non-negative matrix $M \geq 0$ (i.e., $M_{i,j} \geq 0$ for all i, j) of size $n \times n$ we associate a direct[illegible] G with n vertices $V = \{P_1, P_2, \ldots, P_n\}$ and a directed edge [illegible] in case $M_{ij} > 0$.
- Such a graph has the [illegible] matrix $\widetilde{M} = \text{sign}(M)$.
- **Lemma:** There is a path of length k from P_i to P_j in G just in case $(M^k)_{ij} > 0$. (Proof is an easy induction on k).

- **Homework:** Prove by induction on k that $(\widetilde{M}^k)_{i,j}$ is exactly the number of directed paths from P_i to P_j of length exactly k.

- **Definition:** We say that a directed graph G is *strongly connected* just in case for any two vertices $P_i, P_j \in G$ there exists a directed path in G from P_i to P_j (and thus also a directed path from P_j to P_i).

- **Definition:** A Markov chain is *irreducible* if the graph corresponding to its transition probabilities matrix is strongly connected.
- Google matrix induces a strongly connected graph.

- This is trivially true because, in fact, there is a directed edge between any two vertices.

- To a non-negative matrix $M \geq 0$ (i.e., $M_{i,j} \geq 0$ for all i, j) of size $n \times n$ we associate a directed graph G with n vertices $V = \{P_1, P_2, \ldots, P_n\}$ and a directed edge $e(i,j)$ just in case $M_{ij} > 0$.
- Such a graph has the adjacency matrix $\widetilde{M} = \mathrm{sign}(M)$.
- **Lemma:** There is a path of length k from P_i to P_j in G just in case $(M^k)_{ij} > 0$. (Proof is an easy induction on k).

- **Homework:** Prove by induction on k that $(\widetilde{M}^k)_{i,j}$ is exactly the number of directed paths from P_i to P_j of length exactly k.

- **Definition:** We say that a directed graph G is *strongly connected* just in case for any two vertices $P_i, P_j \in G$ there exists a directed path in G from P_i to P_j (and thus also a directed path from P_j to P_i).

- **Definition:** A Markov chain is *irreducible* if the graph corresponding to its transition probabilities matrix is strongly connected.

- Google matrix induces a strongly connected graph.

- This is trivially true because, in fact, there is a directed edge between any two vertices.

- To a non-negative matrix $M \geq 0$ (i.e., $M_{i,j} \geq 0$ for all i, j) of size $n \times n$ we associate a direct[illegible] G with n vertices $V = \{P_1, P_2, \ldots, P_n\}$ and a directed edge [illegible] in case $M_{ij} > 0$.
- Such a graph has the [illegible] matrix $\widetilde{M} = \mathrm{sign}(M)$.
- **Lemma:** There is a path of length k from P_i to P_j in G just in case $(M^k)_{ij} > 0$. (Proof is an easy induction on k).
- **Homework:** Prove by induction on k that $(\widetilde{M}^k)_{i,j}$ is exactly the number of directed paths from P_i to P_j of length exactly k.
- **Definition:** We say that a directed graph G is *strongly connected* just in case for any two vertices P_i, P_j in G there exists a directed path in G from P_i to P_j (and thus also a directed path from P_j to P_i).
- **Definition:** A Markov chain is *irreducible* if the graph corresponding to its transition probabilities matrix is strongly connected.
- Google matrix induces a strongly connected graph.
- This is trivially true because, in fact, there is a directed edge between any two vertices.

Back to the general Markov Chains

- To a non-negative matrix $M \geq 0$ (i.e., $M_{i,j} \geq 0$ for all i, j) of size $n \times n$ we associate a directed graph G with n vertices $V = \{P_1, P_2, \ldots, P_n\}$ and a directed edge $e(i,j)$ just in case $M_{ij} > 0$.
- Such a graph has the adjacency matrix $\widetilde{M} = \mathrm{sign}(M)$.
- **Lemma:** There is a path of length k from P_i to P_j in G just in case $(M^k)_{ij} > 0$. (Proof is an easy induction on k).

- **Homework:** Prove by induction on k that $(\widetilde{M}^k)_{i,j}$ is exactly the number of directed paths from P_i to P_j of length exactly k.

- **Definition:** We say that a directed graph G is *strongly connected* just in case for any two vertices $i, j \in G$ there exists a directed path in G from P_i to P_j (and thus also a directed path from P_j to P_i).

- **Definition:** A Markov chain is *irreducible* if the graph corresponding to its transition probabilities matrix is strongly connected.

- Google matrix induces a strongly connected graph.

- This is trivially true because, in fact, there is a directed edge between any two vertices.

Back to the general Markov Chains

- To a non-negative matrix $M \geq 0$ (i.e., $M_{i,j} \geq 0$ for all i, j) of size $n \times n$ we associate a directed graph G with n vertices $V = \{P_1, P_2, \ldots, P_n\}$ and a directed edge e_{ij} just in case $M_{ij} > 0$.
- Such a graph has the adjacency matrix $\widetilde{M} = \mathrm{sign}(M)$.
- **Lemma:** There is a path of length k from P_i to P_j in G just in case $(M^k)_{ij} > 0$. (Proof is an easy induction on k).

- **Homework:** Prove by induction on k that $(\widetilde{M^k})_{i,j}$ is exactly the number of directed paths from P_i to P_j of length exactly k.

- **Definition:** We say that a directed graph G is *strongly connected* just in case for any two vertices $i, j \in G$ there exists a directed path in G from P_i to P_j (and thus also a directed path from P_j to P_i).

- **Definition:** A Markov chain is *irreducible* if the graph corresponding to its transition probabilities matrix is strongly connected.

- Google matrix induces a strongly connected graph.

- This is trivially true because, in fact, there is a directed edge between any two vertices.

Back to the general Markov Chains

- To a non-negative matrix $M \geq 0$ (i.e., $M_{i,j} \geq 0$ for all i, j) of size $n \times n$ we associate a directed graph G with n vertices $V = \{P_1, P_2, \ldots, P_n\}$ and a directed edge $e(i,j)$ just in case $M_{ij} > 0$.
- Such a graph has the adjacency matrix $\widetilde{M} = \mathrm{sign}(M)$.
- **Lemma:** There is a path of length k from P_i to P_j in G just in case $(M^k)_{ij} > 0$. (Proof is an easy induction on k).

- **Homework:** Prove by induction on k that $(\widetilde{M^k})_{i,j}$ is exactly the number of directed paths from P_i to P_j of length exactly k.

- **Definition:** We say that a directed graph G is *strongly connected* just in case for any two vertices $i, j \in G$ there exists a directed path in G from P_i to P_j (and thus also a directed path from P_j to P_i).

- **Definition:** A Markov chain is *irreducible* if the graph corresponding to its transition probabilities matrix is strongly connected.

- Google matrix induces a strongly connected graph.

- This is trivially true because, in fact, there is a directed edge between any two vertices.

- To a non-negative matrix $M \geq 0$ (i.e., $M_{i,j} \geq 0$ for all i, j) of size $n \times n$ we associate a directed graph G with n vertices $V = \{P_1, P_2, \ldots, P_n\}$ and a directed edge from P_i to P_j just in case $M_{ij} > 0$.
- Such a graph has the adjacency matrix $\widetilde{M} = \mathrm{sign}(M)$.
- **Lemma:** There is a path of length k from P_i to P_j in G just in case $(M^k)_{ij} > 0$. (Proof is an easy induction on k).

- **Homework:** Prove by induction on k that $(\widetilde{M}^k)_{i,j}$ is exactly the number of directed paths from P_i to P_j of length exactly k.

- **Definition:** We say that a directed graph G is *strongly connected* just in case for any two vertices $i, j \in G$ there exists a directed path in G from P_i to P_j (and thus also a directed path from P_j to P_i).

- **Definition:** A Markov chain is *irreducible* if the graph corresponding to its transition probabilities matrix is strongly connected.

- Google matrix induces a strongly connected graph.

- This is trivially true because, in fact, there is a directed edge between any two vertices.

Back to the general Markov Chains

- **Definition:** A state P_i of a Markov chain is periodic if there exists an integer $K > 1$ such that all loops in its underlying graph which contain vertex P_i have length divisible by K.
- Example: in bipartite graphs every loop through every vertex has length divisible by 2.
- Markov chains which do not have any periodic states are called *aperiodic Markov chains.*
- Markov chain which corresponds to the Google matrix G is aperiodic.
- This is trivial, because the corresponding graph is complete; thus for every vertex there exists a loop containing that vertex of every length ≥ 2.

Back to the general Markov Chains

- **Definition:** A state [illegible]arkov chain is periodic if there exists an integer $K > 1$ such t[illegible]ps in its underlying graph which contain vertex P_i have length [illegible] by K.
- Example: in bipartite graphs every loop through every vertex has length divisible by 2.

- Markov chains which do not have any periodic states are called *aperiodic Markov chains.*

- Markov chain which corresponds to the Google matrix G is aperiodic.

- This is trivial, because the corresponding graph is complete; thus for every vertex there exists a loop containing that vertex of every length ≥ 2.

- **Definition:** A state [illegible]arkov chain is periodic if there exists an integer $K > 1$ such t[illegible]ps in its underlying graph which contain vertex P_i have length [illegible] by K.
- Example: in bipartite graphs every loop through every vertex has length divisible by 2.
- Markov chains which do not have any periodic states are called *aperiodic Markov chains.*
- Markov chain which corresponds to the Google matrix G is aperiodic.
- This is trivial, because the corresponding graph is complete; thus for every vertex there exists a loop containing that vertex of every length ≥ 2.

Back to the general Markov Chains

- **Definition:** A state P_i in a Markov chain is periodic if there exists an integer $K > 1$ such that all loops in its underlying graph which contain vertex P_i have length divisible by K.
- Example: in bipartite graphs every loop through every vertex has length divisible by 2.
- Markov chains which do not have any periodic states are called *aperiodic Markov chains*.
- Markov chain which corresponds to the Google matrix G is aperiodic.
- This is trivial, because the corresponding graph is complete; thus for every vertex there exists a loop containing that vertex of every length ≥ 2.

Back to the general Markov Chains

- **Definition:** A state arkov chain is periodic if there exists an integer $K > 1$ such t ps in its underlying graph which contain vertex P_i have length by K.
- Example: in bipartite graphs every loop through every vertex has length divisible by 2.
- Markov chains which do not have any periodic states are called *aperiodic Markov chains.*
- Markov chain which corresponds to the Google matrix G is aperiodic.
- This is trivial, because the corresponding graph is complete; thus for every vertex there exists a loop containing that vertex of every length ≥ 2.

- A general property of Markov chains insures that the Google page rank is well defined (i.e., exi[illegible]unique) and can be computed iteratively.
- **Theorem:** Any finit[illegible]ble and aperiodic Markov chain has the following properties:
  1. For every initial probability distribution of states $\mathbf{q}^{(0)}$ the value of $\mathbf{q}^{(t)} = \mathbf{q}^{(0)} G^t$ converges as $t \to \infty$ to a unique stationary distribution $\mathbf{q}$, i.e., converges to a unique distribution $\mathbf{q}$ which is independent of $\mathbf{q}^{(0)}$ and satisfies $\mathbf{q} = \mathbf{q}G$.
  2. Note: in the above $\mathbf{q}$ is a row vector, to avoid having to always transpose it.
  3. Let $N(P_i, T)$ be the number of times the system has been in state P_i during T many transitions of such a Markov chain; then

$$\lim_{T \to \infty} \frac{N(P_i, T)}{T} = \mathbf{q}_i$$

Back to the general Markov Chains

- A general property of Markov chains insures that the Google page rank is well defined (i.e., exists and is unique) and can be computed iteratively.
- **Theorem:** Any finite, irreducible and aperiodic Markov chain has the following properties:
  1. For every initial probability distribution of states $\mathbf{q}^{(0)}$ the value of $\mathbf{q}^{(t)} = \mathbf{q}^{(0)} G^t$ converges as $t \to \infty$ to a unique stationary distribution $\mathbf{q}$, i.e., converges to a unique distribution $\mathbf{q}$ which is independent of $\mathbf{q}^{(0)}$ and satisfies $\mathbf{q} = \mathbf{q}G$.
  2. Note: in the above $\mathbf{q}$ is a row vector, to avoid having to always transpose it.
  3. Let $N(P_i, T)$ be the number of times the system has been in state P_i during T many transitions of such a Markov chain; then

$$\lim_{T \to \infty} \frac{N(P_i, T)}{T} = \mathbf{q}_i$$

Back to the general Markov Chains

- A general property of Markov chains insures that the Google page rank is well defined (i.e., exists and is unique) and can be computed iteratively.
- **Theorem:** Any finite, irreducible and aperiodic Markov chain has the following properties:
  1. For every initial probability distribution of states $\mathbf{q}^{(0)}$ the value of $\mathbf{q}^{(t)} = \mathbf{q}^{(0)} G^t$ converges as $t \to \infty$ to a unique stationary distribution $\mathbf{q}$, i.e., converges to a unique distribution $\mathbf{q}$ which is independent of $\mathbf{q}^{(0)}$ and satisfies $\mathbf{q} = \mathbf{q}G$.
  2. Note: in the above $\mathbf{q}$ is a row vector, to avoid having to always transpose it.
  3. Let $N(P_i, T)$ be the number of times the system has been in state P_i during T many transitions of such a Markov chain; then

$$\lim_{T\to\infty} \frac{N(P_i, T)}{T} = \mathbf{q}_i$$

Back to the general Markov Chains

- A general property of Markov chains insures that the Google page rank is well defined (i.e., exists and is unique) and can be computed iteratively.
- **Theorem:** Any finite, irreducible and aperiodic Markov chain has the following properties:
  1. For every initial probability distribution of states $\mathbf{q}^{(0)}$ the value of $\mathbf{q}^{(t)} = \mathbf{q}^{(0)} G^t$ converges as $t \to \infty$ to a unique stationary distribution $\mathbf{q}$, i.e., converges to a unique distribution $\mathbf{q}$ which is independent of $\mathbf{q}^{(0)}$ and satisfies $\mathbf{q} = \mathbf{q}\,G$.
  2. Note: in the above $\mathbf{q}$ is a row vector, to avoid having to always transpose it.
  3. Let $N(P_i, T)$ be the number of times the system has been in state P_i during T many transitions of such a Markov chain; then

$$\lim_{T \to \infty} \frac{N(P_i, T)}{T} = \mathbf{q}_i$$

- A general property of Markov chains insures that the Google page rank is well defined (i.e., exists and is unique) and can be computed iteratively.
- **Theorem:** Any finite, irreducible and aperiodic Markov chain has the following properties:
  1. For every initial probability distribution of states $\mathbf{q}^{(0)}$ the value of $\mathbf{q}^{(t)} = \mathbf{q}^{(0)}\,G^{\,t}$ converges as $t \to \infty$ to a unique stationary distribution $\mathbf{q}$, i.e., converges to a unique distribution $\mathbf{q}$ which is independent of $\mathbf{q}^{(0)}$ and satisfies $\mathbf{q} = \mathbf{q}G$.
  2. Note: in the above $\mathbf{q}$ is a row vector, to avoid having to always transpose it.
  3. Let $N(P_i, T)$ be the number of times the system has been in state P_i during T many transitions of such a Markov chain; then

$$\lim_{T\to\infty} \frac{N(P_i, T)}{T} = \mathbf{q}_i$$

Back to the PageRank

- The general theorem on Markov chains implies that:
  - 1 is the left eigenvalue of the Google matrix G of the largest absolute value, and *the stationary distribution* $\mathbf{q}$ is the corresponding left-hand eigenvector, $\mathbf{q}^\intercal = \mathbf{q}^\intercal G$;
  - such stationary distribution $\mathbf{q}$ is unique, i.e., if $\mathbf{q}_1^\intercal = \mathbf{q}_1^\intercal G$, then $\mathbf{q}_1 = \mathbf{q}$;
  - distribution $\mathbf{q}$ can be obtained by starting with an arbitrary initial probability distribution $\mathbf{q}_0$ and obtain $\mathbf{q}$ as $\lim_{k\to\infty} \mathbf{q}_0^\intercal G^k$;
  - an approximation $\tilde{\mathbf{q}}$ of $\mathbf{q}$ can be obtained by taking $\mathbf{q}_0^\intercal = (1/M, 1/M, \ldots, 1/M)$ and a sufficiently large K and computing $\tilde{\mathbf{q}} = \mathbf{q}_0 G^K$ iteratively (NOT by computing G^K!) via:

$$\mathbf{q}(0) = \mathbf{q}_0$$

$$(\mathbf{q}(n+1))^\intercal = (\mathbf{q}(n))^\intercal G \qquad \text{for } 0 \le n < K; \tag{6}$$

  - the i^{th} coordinate of such obtained distribution $\mathbf{q}^\intercal = (q_1, \ldots, q_i, \ldots, q_M)$ roughly gives the ratio $N(P_i, T)/T$ where $N(P_i, T)$ is the number of times P_i has been visited during a surfing session of length T.

Back to the PageRank

- The general theorem on Markov chains implies that:
  - 1 is the left eigenvalue of the Google matrix G of the largest absolute value, and *the stationary distribution* $\mathbf{q}$ is the corresponding left hand eigenvector, $\mathbf{q}^\intercal = \mathbf{q}^\intercal G$;
  - such stationary distribution $\mathbf{q}$ is unique, i.e., if $\mathbf{q}_1^\intercal = \mathbf{q}_1^\intercal G$, then $\mathbf{q}_1 = \mathbf{q}$;
  - distribution $\mathbf{q}$ can be obtained by starting with an arbitrary initial probability distribution $\mathbf{q}_0$ and obtain $\mathbf{q}$ as $\lim_{k\to\infty} \mathbf{q}_0^\intercal G^k$;
  - an approximation $\tilde{\mathbf{q}}$ of $\mathbf{q}$ can be obtained by taking $\mathbf{q}_0^\intercal = (1/M, 1/M, \ldots, 1/M)$ and a sufficiently large K and computing $\tilde{\mathbf{q}} = \mathbf{q}_0 G^K$ iteratively (NOT by computing G^K!) via:

$$\mathbf{q}(0) = \mathbf{q}_0$$

$$(\mathbf{q}(n+1))^\intercal = (\mathbf{q}(n))^\intercal G \qquad \text{for } 0 \le n < K; \tag{6}$$

  - the i^{th} coordinate of such obtained distribution $\mathbf{q}^\intercal = (q_1, \ldots, q_i, \ldots, q_M)$ roughly gives the ratio $N(P_i, T)/T$ where $N(P_i, T)$ is the number of times P_i has been visited during a surfing session of length T.

- The general theorem on Markov chains implies that:
  - 1 is the left eigenvalue of the Google matrix G of the largest absolute value, and *the stationary distribution* $\mathbf{q}$ is the corresponding left-hand side eigenvector, $\mathbf{q}^\intercal = \mathbf{q}^\intercal G$;
  - such stationary distribution $\mathbf{q}$ is unique, i.e., if $\mathbf{q}_1^\intercal = \mathbf{q}_1^\intercal G$, then $\mathbf{q}_1 = \mathbf{q}$;
  - distribution $\mathbf{q}$ can be obtained by starting with an arbitrary initial probability distribution $\mathbf{q}_0$ and obtain $\mathbf{q}$ as $\lim_{k\to\infty} \mathbf{q}_0^\intercal G^k$;
  - an approximation $\tilde{\mathbf{q}}$ of $\mathbf{q}$ can be obtained by taking $\mathbf{q}_0^\intercal = (1/M, 1/M, \ldots, 1/M)$ and a sufficiently large K and computing $\tilde{\mathbf{q}} = \mathbf{q}_0\, G^K$ iteratively (NOT by computing G^K!) via:

$$\mathbf{q}(0) = \mathbf{q}_0$$

$$(\mathbf{q}(n+1))^\intercal = (\mathbf{q}(n))^\intercal G \qquad \text{for } 0 \le n < K; \tag{6}$$

  - the i^{th} coordinate of such obtained distribution $\mathbf{q}^\intercal = (q_1, \ldots, q_i, \ldots, q_M)$ roughly gives the ratio $N(P_i, T)/T$ where $N(P_i, T)$ is the number of times P_i has been visited during a surfing session of length T.

Back to the PageRank

- The general theorem on Markov chains implies that:
  - 1 is the left eigenvalue of the Google matrix G of the largest absolute value, and *the stationary distribution* $\mathbf{q}$ is the corresponding left hand side eigenvector, $\mathbf{q}^{\intercal} = \mathbf{q}^{\intercal}\, G$;
  - such stationary distribution $\mathbf{q}$ is unique, i.e., if $\mathbf{q}_1^{\intercal} = \mathbf{q}_1^{\intercal}\, G$, then $\mathbf{q}_1 = \mathbf{q}$;
  - distribution $\mathbf{q}$ can be obtained by starting with an arbitrary initial probability distribution $\mathbf{q}_0$ and obtain $\mathbf{q}$ as $\lim_{k\to\infty} \mathbf{q}_0^{\intercal}\, G^k$;
  - an approximation $\tilde{\mathbf{q}}$ of $\mathbf{q}$ can be obtained by taking $\mathbf{q}_0^{\intercal} = (1/M, 1/M, \ldots, 1/M)$ and a sufficiently large K and computing $\tilde{\mathbf{q}} = \mathbf{q}_0\, G^K$ iteratively (NOT by computing G^K!) via:

$$\mathbf{q}(0) = \mathbf{q}_0$$

$$(\mathbf{q}(n+1))^{\intercal} = (\mathbf{q}(n))^{\intercal}\, G \qquad \text{for } 0 \leq n < K; \tag{6}$$

  - the i^{th} coordinate of such obtained distribution $\mathbf{q}^{\intercal} = (q_1, \ldots, q_i, \ldots, q_M)$ roughly gives the ratio $N(P_i, T)/T$ where $N(P_i, T)$ is the number of times P_i has been visited during a surfing session of length T.

- The general theorem on Markov chains implies that:
  - 1 is the left eigenvalue of the Google matrix G of the largest absolute value, and *the stationary distribution* $\mathbf{q}$ is the corresponding left-hand eigenvector, $\mathbf{q}^\intercal = \mathbf{q}^\intercal G$;
  - such stationary distribution $\mathbf{q}$ is unique, i.e., if $\mathbf{q}_1^\intercal = \mathbf{q}_1^\intercal G$, then $\mathbf{q}_1 = \mathbf{q}$;
  - distribution $\mathbf{q}$ can be obtained by starting with an arbitrary initial probability distribution $\mathbf{q}_0$ and obtain $\mathbf{q}$ as $\lim_{k\to\infty} \mathbf{q}_0^\intercal G^k$;
  - an approximation $\tilde{\mathbf{q}}$ of $\mathbf{q}$ can be obtained by taking $\mathbf{q}_0^\intercal = (1/M, 1/M, \ldots, 1/M)$ and a sufficiently large K and computing $\tilde{\mathbf{q}} = \mathbf{q}_0 G^K$ iteratively (NOT by computing G^K!) via:

$$\mathbf{q}(0) = \mathbf{q}_0$$

$$(\mathbf{q}(n+1))^\intercal = (\mathbf{q}(n))^\intercal G \qquad \text{for } 0 \leq n < K; \tag{6}$$

  - the i^{th} coordinate of such obtained distribution $\mathbf{q}^\intercal = (q_1, \ldots, q_i, \ldots, q_M)$ roughly gives the ratio $N(P_i, T)/T$ where $N(P_i, T)$ is the number of times P_i has been visited during a surfing session of length T.

- The general theorem on Markov chains implies that:
  - 1 is the left eigenvalue of the Google matrix G of the largest absolute value, and *the stationary distribution* $\mathbf{q}$ is the corresponding left hand side eigenvector, $\mathbf{q}^\intercal = \mathbf{q}^\intercal G$;
  - such stationary distribution $\mathbf{q}$ is unique, i.e., if $\mathbf{q}_1^\intercal = \mathbf{q}_1^\intercal G$, then $\mathbf{q}_1 = \mathbf{q}$;
  - distribution $\mathbf{q}$ can be obtained by starting with an arbitrary initial probability distribution $\mathbf{q}_0$ and obtain $\mathbf{q}$ as $\lim_{k\to\infty} \mathbf{q}_0^\intercal G^k$;
  - an approximation $\tilde{\mathbf{q}}$ of $\mathbf{q}$ can be obtained by taking $\mathbf{q}_0^\intercal = (1/M, 1/M, \ldots, 1/M)$ and a sufficiently large K and computing $\tilde{\mathbf{q}} = \mathbf{q}_0\, G^K$ iteratively (NOT by computing G^K!) via:

$$\mathbf{q}(0) = \mathbf{q}_0$$

$$(\mathbf{q}(n+1))^\intercal = (\mathbf{q}(n))^\intercal G \qquad \text{for } 0 \le n < K; \qquad (6)$$

  - the i^{th} coordinate of such obtained distribution $\mathbf{q}^\intercal = (q_1, \ldots, q_i, \ldots, q_M)$ roughly gives the ratio $N(P_i, T)/T$ where $N(P_i, T)$ is the number of times P_i has been visited during a surfing session of length T.

Back to the PageRank

- How close to 1 should α be?
- Let us compute the *expected* length lh of surfing between two "teleportations".

$$
\begin{aligned}
E(lh) &= 0\,(1-\alpha) + 1\,\alpha(1-\alpha) + \ldots + k\,\alpha^k(1-\alpha) + \ldots \\
&= \alpha(1-\alpha)\underbrace{(1 + 2\,\alpha + \ldots + k\,\alpha^{k-1} + \ldots)}_{\frac{1}{(1-\alpha)^2}} \\
&= \frac{\alpha}{1-\alpha}
\end{aligned}
$$

- Google uses $\alpha = 0.85$.
- Google inventors Page and Brin obtained the value .85 empirically.
- The expected surfing length is $\frac{.85}{1-.85} = \frac{.85}{.15} \approx 5.7$.
- This looks like very short surfing before random teleportations!
- Have you heard about the *six degrees of separation*?
- With $\alpha = .85$ the number of iterations needed for an approximate convergence is $50 - 100$, i.e., $\mathbf{q} \approx \mathbf{q}_0\, G^{50}$.

Back to the PageRank

- How close to 1 should α be??
- Let us compute the *expected* length lh of surfing between two "teleportations".

$$
\begin{aligned}
E(lh) &= 0\,(1-\alpha) + 1\,\alpha(1-\alpha) + \ldots + k\,\alpha^k(1-\alpha) + \ldots \\
&= \alpha(1-\alpha)\underbrace{(1 + 2\,\alpha + \ldots + k\,\alpha^{k-1} + \ldots)}_{\frac{1}{(1-\alpha)^2}} \\
&= \frac{\alpha}{1-\alpha}
\end{aligned}
$$

- Google uses $\alpha = 0.85$.
- Google inventors Page and Brin obtained the value .85 empirically.
- The expected surfing length is $\frac{.85}{1-.85} = \frac{.85}{.15} \approx 5.7$.
- This looks like very short surfing before random teleportations!
- Have you heard about the *six degrees of separation*?
- With $\alpha = .85$ the number of iterations needed for an approximate convergence is $50 - 100$, i.e., $\mathbf{q} \approx \mathbf{q}_0\, G^{50}$.

Back to the PageRank

- How close to 1 should α be?
- Let us compute the *expected* length lh of surfing between two "teleportations".

$$
\begin{aligned}
E(lh) &= 0\,(1-\alpha) + 1\,\alpha(1-\alpha) + \ldots + k\,\alpha^k(1-\alpha) + \ldots \\
&= \alpha(1-\alpha)\underbrace{(1 + 2\,\alpha + \ldots + k\,\alpha^{k-1} + \ldots)}_{\frac{1}{(1-\alpha)^2}} \\
&= \frac{\alpha}{1-\alpha}
\end{aligned}
$$

- Google uses $\alpha = 0.85$.
- Google inventors Page and Brin obtained the value .85 empirically.
- The expected surfing length is $\frac{.85}{1-.85} = \frac{.85}{.15} \approx 5.7$.
- This looks like very short surfing before random teleportations!
- Have you heard about the *six degrees of separation*?
- With $\alpha = .85$ the number of iterations needed for an approximate convergence is $50 - 100$, i.e., $\mathbf{q} \approx \mathbf{q}_0\, G^{50}$.

Back to the PageRank

- How close to 1 should α be??
- Let us compute the *expected* length lh of surfing between two "teleportations".

$$
\begin{aligned}
E(lh) &= 0\,(1-\alpha) + 1\,\alpha(1-\alpha) + \ldots + k\,\alpha^k(1-\alpha) + \ldots \\
&= \alpha(1-\alpha)\underbrace{(1 + 2\,\alpha + \ldots + k\,\alpha^{k-1} + \ldots)}_{\frac{1}{(1-\alpha)^2}} \\
&= \frac{\alpha}{1-\alpha}
\end{aligned}
$$

- Google uses $\alpha = 0.85$.
- Google inventors Page and Brin obtained the value .85 empirically.
- The expected surfing length is $\frac{.85}{1-.85} = \frac{.85}{.15} \approx 5.7$.
- This looks like very short surfing before random teleportations!
- Have you heard about the *six degrees of separation*?
- With $\alpha = .85$ the number of iterations needed for an approximate convergence is $50 - 100$, i.e., $\mathbf{q} \approx \mathbf{q}_0\, G^{50}$.

Back to the PageRank

- How close to 1 should α be?
- Let us compute the *expected* length lh of surfing between two "teleportations".

$$\begin{aligned} E(lh) &= 0\,(1-\alpha) + 1\,\alpha(1-\alpha) + \ldots + k\,\alpha^k(1-\alpha) + \ldots \\ &= \alpha(1-\alpha)\underbrace{(1 + 2\,\alpha + \ldots + k\,\alpha^{k-1} + \ldots)}_{\frac{1}{(1-\alpha)^2}} \\ &= \frac{\alpha}{1-\alpha} \end{aligned}$$

- Google uses $\alpha = 0.85$.
- Google inventors Page and Brin obtained the value .85 empirically.
- The expected surfing length is $\frac{.85}{1-.85} = \frac{.85}{.15} \approx 5.7$.
- This looks like very short surfing before random teleportations!
- Have you heard about the *six degrees of separation*?
- With $\alpha = .85$ the number of iterations needed for an approximate convergence is $50 - 100$, i.e., $\mathbf{q} \approx \mathbf{q}_0\, G^{50}$.

- How close to 1 should α be??
- Let us compute the *expected* length lh of surfing between two "teleportations".

$$
\begin{aligned}
E(lh) &= 0\,(1-\alpha) + 1\,\alpha(1-\alpha) + \ldots + k\,\alpha^k(1-\alpha) + \ldots \\
&= \alpha(1-\alpha)\underbrace{(1 + 2\,\alpha + \ldots + k\,\alpha^{k-1} + \ldots)}_{\frac{1}{(1-\alpha)^2}} \\
&= \frac{\alpha}{1-\alpha}
\end{aligned}
$$

- Google uses $\alpha = 0.85$.
- Google inventors Page and Brin obtained the value .85 empirically.
- The expected surfing length is $\frac{.85}{1-.85} = \frac{.85}{.15} \approx 5.7$.
- This looks like very short surfing before random teleportations!
- Have you heard about the *six degrees of separation*?
- With $\alpha = .85$ the number of iterations needed for an approximate convergence is $50 - 100$, i.e., $\mathbf{q} \approx \mathbf{q}_0\, G^{50}$.

- How close to 1 should α be??
- Let us compute the *expected* length lh of surfing between two "teleportations".

$$\begin{aligned} E(lh) &= 0\,(1-\alpha) + 1\,\alpha(1-\alpha) + \ldots + k\,\alpha^k(1-\alpha) + \ldots \\ &= \alpha(1-\alpha)\underbrace{(1 + 2\,\alpha + \ldots + k\,\alpha^{k-1} + \ldots)}_{\frac{1}{(1-\alpha)^2}} \\ &= \frac{\alpha}{1-\alpha} \end{aligned}$$

- Google uses $\alpha = 0.85$.
- Google inventors Page and Brin obtained the value .85 empirically.
- The expected surfing length is $\frac{.85}{1-.85} = \frac{.85}{.15} \approx 5.7$.
- This looks like very short surfing before random teleportations!
- Have you heard about the *six degrees of separation*?
- With $\alpha = .85$ the number of iterations needed for an approximate convergence is $50-100$, i.e., $\mathbf{q} \approx \mathbf{q}_0\, G^{50}$.

Back to the PageRank

- How close to 1 should α be??
- Let us compute the *expected* length lh of surfing between two "teleportations".

$$\begin{aligned} E(lh) &= 0\,(1-\alpha) + 1\,\alpha(1-\alpha) + \ldots + k\,\alpha^k(1-\alpha) + \ldots \\ &= \alpha(1-\alpha)\underbrace{(1 + 2\,\alpha + \ldots + k\,\alpha^{k-1} + \ldots)}_{\frac{1}{(1-\alpha)^2}} \\ &= \frac{\alpha}{1-\alpha} \end{aligned}$$

- Google uses $\alpha = 0.85$.
- Google inventors Page and Brin obtained the value .85 empirically.
- The expected surfing length is $\frac{.85}{1-.85} = \frac{.85}{.15} \approx 5.7$.
- This looks like very short surfing before random teleportations!
- Have you heard about the *six degrees of separation*?
- With $\alpha = .85$ the number of iterations needed for an approximate convergence is $50 - 100$, i.e., $\mathbf{q} \approx \mathbf{q}_0\, G^{50}$.

Back to the PageRank

- How close to 1 should α be??
- Let us compute the *expected* length lh of surfing between two "teleportations".

$$\begin{aligned} E(lh) &= 0\,(1-\alpha) + 1\,\alpha(1-\alpha) + \ldots + k\,\alpha^k(1-\alpha) + \ldots \\ &= \alpha(1-\alpha)\underbrace{(1 + 2\,\alpha + \ldots + k\,\alpha^{k-1} + \ldots)}_{\frac{1}{(1-\alpha)^2}} \\ &= \frac{\alpha}{1-\alpha} \end{aligned}$$

- Google uses $\alpha = 0.85$.
- Google inventors Page and Brin obtained the value .85 empirically.
- The expected surfing length is $\frac{.85}{1-.85} = \frac{.85}{.15} \approx 5.7$.
- This looks like very short surfing before random teleportations!
- Have you heard about the *six degrees of separation*?
- With $\alpha = .85$ the number of iterations needed for an approximate convergence is $50 - 100$, i.e., $\mathbf{q} \approx \mathbf{q_0}\, G^{50}$.

- Larger values produce more accurate representation of "importance" of a webpage ...
- ... but the convergence slows down fast.
- Error of approximation of $\mathbf{q}$ by $\mathbf{q}_0 G^k$ decreases approximately as α^k.
- Comparing the number of iterations with $\alpha = .85$ and with $\alpha = .95$ needed for the same accuracy:
  $.95^m = .85^k$ implies $m = k \log_{10} .85 / \log_{10} .95 \approx 3.17k$.
- But, even more importantly, increasing α increases the sensitivity of the resulting PageRank.
- Why would that be bad?
- Internet content and structure changes on daily basis, but the PageRank should change slowly, otherwise it would not be useful.

- Larger values produce a more accurate representation of "importance" of a webpage ...
- ... but the convergence slows down fast.
- Error of approximation of $\mathbf{q}$ by $\mathbf{q}_0\, G^k$ decreases approximately as α^k.
- Comparing the number of iterations with $\alpha = .85$ and with $\alpha = .95$ needed for the same accuracy:
  $.95^m = .85^k$ implies $m = k \log_{10} .85 / \log_{10} .95 > 3.1k$.
- But, even more importantly, increasing α increases the sensitivity of the resulting PageRank.
- Why would that be bad?
- Internet content and structure changes on daily basis, but the PageRank should change slowly, otherwise it would not be useful.

Back to the PageRank

- Larger values produce more accurate representation of "importance" of a webpage ...
- ... but the convergence slows down fast.
- Error of approximation of $\mathbf{q}$ by $\mathbf{q_0}\, G^k$ decreases approximately as α^k.
- Comparing the number of iterations with $\alpha = .85$ and with $\alpha = .95$ needed for the same accuracy:
  $.95^m = .85^k$ implies $m = k \log_{10} .85/\log_{10} .95 \approx 3.17k$
- But, even more importantly, increasing α increases the sensitivity of the resulting PageRank.
- Why would that be bad?
- Internet content and structure changes on daily basis, but the PageRank should change slowly, otherwise it would not be useful.

- Larger values produce more accurate representation of "importance" of a webpage …
- … but the convergence slows down fast.
- Error of approximation of $\mathbf{q}$ by $\mathbf{q_0}\, G^k$ decreases approximately as α^k.
- Comparing the number of iterations with $\alpha = .85$ and with $\alpha = .95$ needed for the same accuracy:
  $.95^m = .85^k$ implies $m = k \log_{10} .85 / \log_{10} .95 \approx 3.17k$.
- But, even more importantly, increasing α increases the sensitivity of the resulting PageRank.
- Why would that be bad?
- Internet content and structure changes on daily basis, but the PageRank should change slowly, otherwise it would not be useful.

Back to the PageRank

- Larger values produc[illegible]curate representation of "importance" of a webpage ...
- ... but the converge[illegible]own fast.
- Error of approximation of $\mathbf{q}$ by $\mathbf{q_0}\,G^k$ decreases approximately as α^k.
- Comparing the number of iterations with $\alpha = .85$ and with $\alpha = .95$ needed for the same accuracy:
  $.95^m = .85^k$ implies $m = k \log_{10} .85 / \log_{10} .95 > 3.7k$.
- But, even more importantly, increasing α increases the sensitivity of the resulting PageRank.
- Why would that be bad?
- Internet content and structure changes on daily basis, but the PageRank should change slowly, otherwise it would not be useful.

- Larger values produce a more accurate representation of "importance" of a webpage ...
- ... but the convergence slows down fast.
- Error of approximation of $\mathbf{q}$ by $\mathbf{q_0}\, G^k$ decreases approximately as α^k.
- Comparing the number of iterations with $\alpha = .85$ and with $\alpha = .95$ needed for the same accuracy:
  $.95^m = .85^k$ implies $m = k \log_{10} .85 / \log_{10} .95 \approx 3.17k$.
- But, even more importantly, increasing α increases the sensitivity of the resulting PageRank.
- Why would that be bad?
- Internet content and structure changes on daily basis, but the PageRank should change slowly, otherwise it would not be useful.

- Larger values produce more accurate representation of "importance" of a webpage ...
- ... but the convergence slows down fast.
- Error of approximation of $\mathbf{q}$ by $\mathbf{q_0}\, G^k$ decreases approximately as α^k.
- Comparing the number of iterations with $\alpha = .85$ and with $\alpha = .95$ needed for the same accuracy:
  $.95^m = .85^k$ implies $m = k \log_{10} .85 / \log_{10} .95 \approx 3.17k$.
- But, even more importantly, increasing α increases the sensitivity of the resulting PageRank.
- Why would that be bad?
- Internet content and structure changes on daily basis, but the PageRank should change slowly, otherwise it would not be useful.

Back to the PageRank

- Larger values produc[illegible]curate representation of "importance" of a webpage ...
- ... but the converge[illegible]own fast.
- Error of approximation of $\mathbf{q}$ by $\mathbf{q_0}\, G^k$ decreases approximately as α^k.
- Comparing the number of iterations with $\alpha = .85$ and with $\alpha = .95$ needed for the same accuracy:
  $.95^m = .85^k$ implies $m = k \log_{10} .85 / \log_{10} .95 > 3.17k$.
- But, even more importantly, increasing α increases the sensitivity of the resulting PageRank.
- Why would that be bad?
- Internet content and structure changes on daily basis, but the PageRank should change slowly, otherwise it would not be useful.

Refinements of the PageRank algorithm

- Not all outgoing links from a web page are equally important.
- Maybe we should just count the number of clicks on each webpage and assign the probability of following these links accordingly.
- Higher probability should be given to pages of similar kind of content.
- One can in the equation

$$G = \alpha \left(G_1 + \frac{1}{M} \mathbf{d}\, \mathbf{e}^{\intercal} \right) + \frac{1-\alpha}{M} \mathbf{e}\, \mathbf{e}^{\intercal} \qquad (7)$$

- instead of using $\frac{\alpha}{M}\mathbf{d}\,\mathbf{e}^{\intercal}$ and $\frac{1-\alpha}{M}\mathbf{e}\,\mathbf{e}^{\intercal}$ one could use a topic specific teleportation matrix of the form $\alpha\,\mathbf{d}\,\mathbf{v}^{\intercal}$ and $(1-\alpha)\,\mathbf{e}\,\mathbf{v}^{\intercal}$.
- Unfortunately, this involves *the semantics*, i.e., the *meaning* of words and computers are not very good handling that...
- Difficulty with personalising PageRanks.
- AI system *Kaltix* for classifying web pages according to 16 categories of topics.
- Somewhat personalised PageRank is obtained as a weighted average of 16 topic PageRanks.
- Present algorithms incorporates many other features such as methods for defeating farm-links.

Refinements of the PageRank algorithm

- Not all outgoing links from a web page are equally important.
- Maybe we should just count the number of clicks on each webpage and assign the probabilit[illegible]ing these links accordingly.
- Higher probability sh[illegible]ven to pages of similar kind of content.
- One can in the equat[illegible]

$$G = \alpha\left(G_1 + \frac{1}{M}\,\mathbf{d}\,\mathbf{e}^{\intercal}\right) + \frac{1-\alpha}{M}\,\mathbf{e}\,\mathbf{e}^{\intercal} \tag{7}$$

- instead of using $\frac{\alpha}{M}\,\mathbf{d}\,\mathbf{e}^{\intercal}$ and $\frac{1-\alpha}{M}\,\mathbf{e}\,\mathbf{e}^{\intercal}$ one could use a topic specific teleportation matrix of the form $\alpha\,\mathbf{d}\,\mathbf{v}^{\intercal}$ and $(1-\alpha)\,\mathbf{e}\,\mathbf{v}^{\intercal}$.
- Unfortunately, this i[illegible]he *meaning* of words and computers are not very good handling that...
- Difficulty with perso[illegible]ks.
- AI system *Kaltix* for classifying web pages according to 16 categories of topics.
- Somewhat personalised PageRank is obtained as a weighted average of 16 topic PageRanks.
- Present algorithms incorporates many other features such as methods for defeating farm-links.

Refinements of the PageRank algorithm

- Not all outgoing links from a web page are equally important.
- Maybe we should just count the number of clicks on each webpage and assign the probability [illegible] these links accordingly.
- Higher probability should be given to pages of similar kind of content.
- One can in the equat[illegible]

$$G = \alpha\left(G_1 + \frac{1}{M}\,\mathbf{d}\,\mathbf{e}^{\mathsf{T}}\right) + \frac{1-\alpha}{M}\,\mathbf{e}\,\mathbf{e}^{\mathsf{T}} \tag{7}$$

- instead of using $\frac{\alpha}{M}\,\mathbf{d}\,\mathbf{e}^{\mathsf{T}}$ and $\frac{1-\alpha}{M}\,\mathbf{e}\,\mathbf{e}^{\mathsf{T}}$ one could use a topic specific teleportation matrix of the form $\alpha\,\mathbf{d}\,\mathbf{v}^{\mathsf{T}}$ and $(1-\alpha)\,\mathbf{e}\,\mathbf{v}^{\mathsf{T}}$.
- Unfortunately, this in[illegible] the *meaning* of words and computers are not very good handling that...
- Difficulty with personalising PageRanks.
- AI system *Kaltix* for classifying web pages according to 16 categories of topics.
- Somewhat personalised PageRank is obtained as a weighted average of 16 topic PageRanks.
- Present algorithms incorporates many other features such as methods for defeating farm-links.

- Not all outgoing links from a web page are equally important.
- Maybe we should just count the number of clicks on each webpage and assign the probability [illegible]ing these links accordingly.
- Higher probability sh[illegible]ven to pages of similar kind of content.
- One can in the equat[illegible]

$$G = \alpha\left(G_1 + \frac{1}{M}\,\mathbf{d}\,\mathbf{e}^\top\right) + \frac{1-\alpha}{M}\,\mathbf{e}\,\mathbf{e}^\top \qquad (7)$$

- instead of using $\frac{\alpha}{M}\,\mathbf{d}\,\mathbf{e}^\top$ and $\frac{1-\alpha}{M}\,\mathbf{e}\,\mathbf{e}^\top$ one could use a topic specific teleportation matrix of the form $\alpha\,\mathbf{d}\,\mathbf{v}^\top$ and $(1-\alpha)\,\mathbf{e}\,\mathbf{v}^\top$.
- Unfortunately, this i[illegible]the *meaning* of words and computers are not very good handling that...
- Difficulty with personalising PageRanks.
- AI system *Kaltix* for classifying web pages according to 16 categories of topics.
- Somewhat personalised PageRank is obtained as a weighted average of 16 topic PageRanks.
- Present algorithms incorporates many other features such as methods for defeating farm-links.

Refinements of the PageRank algorithm

- Not all outgoing links from a web page are equally important.
- Maybe we should just count the number of clicks on each webpage and assign the probability ... ing these links accordingly.
- Higher probability sh... ven to pages of similar kind of content.
- One can in the equat...

$$G = \alpha\left(G_1 + \frac{1}{M}\,\mathbf{d}\,\mathbf{e}^{\top}\right) + \frac{1-\alpha}{M}\,\mathbf{e}\,\mathbf{e}^{\top} \tag{7}$$

- instead of using $\frac{\alpha}{M}\,\mathbf{d}\,\mathbf{e}^{\top}$ and $\frac{1-\alpha}{M}\,\mathbf{e}\,\mathbf{e}^{\top}$ one could use a topic specific teleportation matrix of the form $\alpha\,\mathbf{d}\,\mathbf{v}^{\top}$ and $(1-\alpha)\,\mathbf{e}\,\mathbf{v}^{\top}$.

- Unfortunately, this in... the *meaning* of words and computers are not very good handling that...
- Difficulty with personalising PageRanks.
- AI system *Kaltix* for classifying web pages according to 16 categories of topics.
- Somewhat personalised PageRank is obtained as a weighted average of 16 topic PageRanks.
- Present algorithms incorporates many other features such as methods for defeating farm-links.

Refinements of the PageRank algorithm

- Not all outgoing links from a web page are equally important.
- Maybe we should just count the number of clicks on each webpage and assign the probability of following these links accordingly.
- Higher probability should be given to pages of similar kind of content.
- One can in the equation

$$G = \alpha \left(G_1 + \frac{1}{M}\,\mathbf{d}\,\mathbf{e}^\intercal \right) + \frac{1-\alpha}{M}\,\mathbf{e}\,\mathbf{e}^\intercal \tag{7}$$

- instead of using $\frac{\alpha}{M}\,\mathbf{d}\,\mathbf{e}^\intercal$ and $\frac{1-\alpha}{M}\,\mathbf{e}\,\mathbf{e}^\intercal$ one could use a topic specific teleportation matrix of the form $\alpha\,\mathbf{d}\,\mathbf{v}^\intercal$ and $(1-\alpha)\,\mathbf{e}\,\mathbf{v}^\intercal$.
- Unfortunately, this involves: *the semantics* i.e., the *meaning* of words and computers are not very good handling that...
- Difficulty with personalising PageRanks.
- AI system *Kaltix* for classifying web pages according to 16 categories of topics.
- Somewhat personalised PageRank is obtained as a weighted average of 16 topic PageRanks.
- Present algorithms incorporates many other features such as methods for defeating farm-links.

Refinements of the PageRank algorithm

- Not all outgoing links from a web page are equally important.
- Maybe we should just count the number of clicks on each webpage and assign the probability of following these links accordingly.
- Higher probability should be given to pages of similar kind of content.
- One can in the equation

$$G = \alpha\left(G_1 + \frac{1}{M}\,\mathbf{d}\,\mathbf{e}^{\intercal}\right) + \frac{1-\alpha}{M}\,\mathbf{e}\,\mathbf{e}^{\intercal} \tag{7}$$

- instead of using $\frac{\alpha}{M}\,\mathbf{d}\,\mathbf{e}^{\intercal}$ and $\frac{1-\alpha}{M}\,\mathbf{e}\,\mathbf{e}^{\intercal}$ one could use a topic specific teleportation matrix of the form $\alpha\,\mathbf{d}\,\mathbf{v}^{\intercal}$ and $(1-\alpha)\,\mathbf{e}\,\mathbf{v}^{\intercal}$.
- Unfortunately, this involves *the semantics* i.e., the *meaning* of words and computers are not very good handling that...
- Difficulty with personalising PageRanks.
- AI system *Kaltix* for classifying web pages according to 16 categories of topics.
- Somewhat personalised PageRank is obtained as a weighted average of 16 topic PageRanks.
- Present algorithms incorporates many other features such as methods for defeating farm-links.

Refinements of the PageRank algorithm

- Not all outgoing links from a web page are equally important.
- Maybe we should just count the number of clicks on each webpage and assign the probability of following these links accordingly.
- Higher probability should be given to pages of similar kind of content.
- One can in the equation

$$G = \alpha \left(G_1 + \frac{1}{M} \mathbf{d}\, \mathbf{e}^{\intercal} \right) + \frac{1-\alpha}{M} \mathbf{e}\, \mathbf{e}^{\intercal} \tag{7}$$

- instead of using $\frac{\alpha}{M} \mathbf{d}\, \mathbf{e}^{\intercal}$ and $\frac{1-\alpha}{M} \mathbf{e}\, \mathbf{e}^{\intercal}$ one could use a topic specific teleportation matrix of the form $\alpha\, \mathbf{d}\, \mathbf{v}^{\intercal}$ and $(1-\alpha)\, \mathbf{e}\, \mathbf{v}^{\intercal}$.
- Unfortunately, this involves *the semantics* i.e., the *meaning* of words and computers are not very good handling that...
- Difficulty with personalising PageRanks.
- AI system *Kaltix* for classifying web pages according to 16 categories of topics.
- Somewhat personalised PageRank is obtained as a weighted average of 16 topic PageRanks.
- Present algorithms incorporates many other features such as methods for defeating farm-links.

Refinements of the PageRank algorithm

- Not all outgoing links from a web page are equally important.
- Maybe we should just count the number of clicks on each webpage and assign the probability [illegible]ing these links accordingly.
- Higher probability sh[illegible]ven to pages of similar kind of content.
- One can in the equat[illegible]

$$G = \alpha \left(G_1 + \frac{1}{M}\,\mathbf{d}\,\mathbf{e}^{\intercal} \right) + \frac{1-\alpha}{M}\,\mathbf{e}\,\mathbf{e}^{\intercal} \tag{7}$$

- instead of using $\frac{\alpha}{M}\,\mathbf{d}\,\mathbf{e}^{\intercal}$ and $\frac{1-\alpha}{M}\,\mathbf{e}\,\mathbf{e}^{\intercal}$ one could use a topic specific teleportation matrix of the form $\alpha\,\mathbf{d}\,\mathbf{v}^{\intercal}$ and $(1-\alpha)\,\mathbf{e}\,\mathbf{v}^{\intercal}$.
- Unfortunately, this involves *the semantics* i.e., the *meaning* of words and computers are not very good handling that...
- Difficulty with personalising PageRanks.
- AI system *Kaltix* for classifying web pages according to 16 categories of topics.
- Somewhat personalised PageRank is obtained as a weighted average of 16 topic PageRanks.
- Present algorithms incorporates many other features such as methods for defeating farm-links.

Refinements of the PageRank algorithm

- Not all outgoing links from a web page are equally important.
- Maybe we should just count the number of clicks on each webpage and assign the probabilit[illegible]ing these links accordingly.
- Higher probability sh[illegible]ven to pages of similar kind of content.
- One can in the equat[illegible]

$$G = \alpha \left(G_1 + \frac{1}{M}\, \mathbf{d}\, \mathbf{e}^{\intercal} \right) + \frac{1-\alpha}{M}\, \mathbf{e}\, \mathbf{e}^{\intercal} \qquad (7)$$

- instead of using $\frac{\alpha}{M}\, \mathbf{d}\, \mathbf{e}^{\intercal}$ and $\frac{1-\alpha}{M}\, \mathbf{e}\, \mathbf{e}^{\intercal}$ one could use a topic specific teleportation matrix of the form $\alpha\, \mathbf{d}\, \mathbf{v}^{\intercal}$ and $(1-\alpha)\, \mathbf{e}\, \mathbf{v}^{\intercal}$.
- Unfortunately, this involves *the semantics* i.e., the *meaning* of words and computers are not very good handling that...
- Difficulty with personalising PageRanks.
- AI system *Kaltix* for classifying web pages according to 16 categories of topics.
- Somewhat personalised PageRank is obtained as a weighted average of 16 topic PageRanks.
- Present algorithms incorporates many other features such as methods for defeating farm-links.

Refinements of the PageRank algorithm

- Not all outgoing links from a web page are equally important.
- Maybe we should just count the number of clicks on each webpage and assign the probability of following these links accordingly.
- Higher probability should be given to pages of similar kind of content.
- One can in the equation

$$G = \alpha\left(G_1 + \frac{1}{M}\,\mathbf{d}\,\mathbf{e}^{\intercal}\right) + \frac{1-\alpha}{M}\,\mathbf{e}\,\mathbf{e}^{\intercal} \qquad (7)$$

- instead of using $\frac{\alpha}{M}\,\mathbf{d}\,\mathbf{e}^{\intercal}$ and $\frac{1-\alpha}{M}\,\mathbf{e}\,\mathbf{e}^{\intercal}$ one could use a topic specific teleportation matrix of the form $\alpha\,\mathbf{d}\,\mathbf{v}^{\intercal}$ and $(1-\alpha)\,\mathbf{e}\,\mathbf{v}^{\intercal}$.
- Unfortunately, this involves *the semantics* i.e., the *meaning* of words and computers are not very good handling that...
- Difficulty with personalising PageRanks.
- AI system *Kaltix* for classifying web pages according to 16 categories of topics.
- Somewhat personalised PageRank is obtained as a weighted average of 16 topic PageRanks.
- Present algorithms incorporates many other features such as methods for defeating farm-links.

- While both the Markov [illegible] have been studied “ad nauseam” and are thus far from being a novelty, the Google inventors deserve a huge credit for finding the ulti[illegible]t day application of these “ancient” concepts.
- The PageRank has seen [illegible]ations; at the class website you can find a paper done by a former 4121 st[illegible]en Rezvani who applied the PageRank to a security problem of eval[illegible]sks of hosts and risk of flows in computer networks.
- As a homework, try applying the PageRank to the following problem:

  The present day “publish [illegible] academia involves counting number of papers researchers have published, as well as the number of citations their papers got.

  1. One might argue t[illegible] a citation in a paper that is itself often cited is more valuable than a citation in a paper that no one cites. Desig[illegible]hich would rank papers according to their “importance”, and then use such an algorithm to rank researchers by their “importance”.
  2. Assume now that you do not have information on the citations in each published paper, but instead you have for every researcher a list of other researchers who have cited him and how many times they cited him. Design again a PageRank style algorithm which would rank researchers by their importance.
- At the class website you can find very detailed lecture notes on the PageRank which also contain a detailed mathematical analysis of the algorithm for those of you that are Math double majors.

The PageRank algorithm conclusion:

- While both the Markov chains and random walks on graphs have been studied "ad nauseam" and are thus far from being a novelty, the Google inventors deserve a huge credit for finding the ulti[illegible]t day application of these "ancient" concepts.
- The PageRank has seen [illegible]ations; at the class website you can find a paper done by a former 4121 st[illegible]en Rezvani who applied the PageRank to a security problem of eval[illegible]sks of hosts and risk of flows in computer networks.
- As a homework, try applying the PageRank to the following problem:

  The present day "publish [illegible] academia involves counting number of papers researchers have published, as well as the number of citations their papers got.

  1. One might argue t[illegible] a citation in a paper that is itself often cited is more valuable than a citation in a paper that no one cites. Desig[illegible]hich would rank papers according to their "importance", and then use such an algorithm to rank researchers by their "importance".
  2. Assume now that you do not have information on the citations in each published paper, but instead you have for every researcher a list of other researchers who have cited him and [illegible] cited him. Design again a PageRank style algorithm which would rank researchers by their importance.
- At the class website you can find very detailed lecture notes on the PageRank which also contain a detailed mathematical analysis of the algorithm for those of you that are Math double majors.

The PageRank algorithm conclusion:

- While both the Markov chains and random walks on graphs have been studied “ad nauseam” and are thus far from being a novelty, the Google inventors deserve a huge credit for finding the ulti[illegible]t day application of these “ancient” concepts.
- The PageRank has seen [illegible]ations; at the class website you can find a paper done by a former 4121 st[illegible]en Rezvani who applied the PageRank to a security problem of evalu[illegible]sks of hosts and risk of flows in computer networks.
- As a homework, try applying the PageRank to the following problem:

  The present day “publish [illegible] academia involves counting number of papers researchers have published, as well as the number of citations their papers got.

  1. One might argue t[illegible] a citation in a paper that is itself often cited is more valuable than a citation in a paper that no one cites. Design a PageRank style algorithm which would rank papers according to their “importance”, and then use such an algorithm to rank researchers by their “importance”.
  2. Assume now that you do not have information on the citations in each published paper, but instead you have for every researcher a list of other researchers who have cited him and how many times they cited him. Design again a PageRank style algorithm which would rank researchers by their importance.
- At the class website you can find very detailed lecture notes on the PageRank which also contain a detailed mathematical analysis of the algorithm for those of you that are Math double majors.

The PageRank algorithm conclusion:

- While both the Markov chains and random walks on graphs have been studied “ad nauseam” and are thus far from being a novelty, the Google inventors deserve a huge credit for finding the ulti[illegible]t day application of these “ancient” concepts.
- The PageRank has seen [illegible]ations; at the class website you can find a paper done by a former 4121 st[illegible]en Rezvani who applied the PageRank to a security problem of evalu[illegible]sks of hosts and risk of flows in computer networks.
- As a homework, try applying the PageRank to the following problem:

  The present day “publish [illegible] academia involves counting number of papers researchers have published, as well as the number of citations their papers got.

  1. One might argue t[illegible] a citation in a paper that is itself often cited is more valuable than a citation in a paper that no one cites. Desig[illegible]hich would rank papers according to their “importance”, and then use such an algorithm to rank researchers by their “importance”.
  2. Assume now that you do not have information on the citations in each published paper, but instead you have for every researcher a list of other researchers who have cited him and how many times they cited him. Design again a PageRank style algorithm which would rank researchers by their importance.
- At the class website you can find very detailed lecture notes on the PageRank which also contain a detailed mathematical analysis of the algorithm for those of you that are Math double majors.

The PageRank algorithm conclusion:

- While both the Markov chains and random walks on graphs have been studied "ad nauseam" and are thus far from being a novelty, the Google inventors deserve a huge credit for finding the ulti[illegible]t day application of these "ancient" concepts.
- The PageRank has seen [illegible]ations; at the class website you can find a paper done by a former 4121 st[illegible]en Rezvani who applied the PageRank to a security problem of evalu[illegible]sks of hosts and risk of flows in computer networks.
- As a homework, try applying the PageRank to the following problem:

  The present day "publish or perish" madness in academia involves counting number of papers researchers have published, as well as the number of citations their papers got.

  1. One might argue that not all citations are equally valuable: a citation in a paper that is itself often cited is more valuable than a citation in a paper that no one cites. Design a PageRank style algorithm which would rank papers according to their "importance", and then use such an algorithm to rank researchers by their "importance".
  2. Assume now that you do not have information on the citations in each published paper, but instead you have for every researcher a list of other researchers who have cited him and how many times they cited him. Design again a PageRank style algorithm which would rank researchers by their importance.
- At the class website you can find very detailed lecture notes on the PageRank which also contain a detailed mathematical analysis of the algorithm for those of you that are Math double majors.

The PageRank algorithm conclusion:

- While both the Markov chains and random walks on graphs have been studied "ad nauseam" and are thus far from being a novelty, the Google inventors deserve a huge credit for finding the ulti[illegible]t day application of these "ancient" concepts.
- The PageRank has seen [illegible]ations; at the class website you can find a paper done by a former 4121 st[illegible]en Rezvani who applied the PageRank to a security problem of evalu[illegible]sks of hosts and risk of flows in computer networks.
- As a homework, try applying the PageRank to the following problem:

  The present day "publish or perish" madness in academia involves counting number of papers researchers have published, as well as the number of citations their papers got.

  1. One might argue that not all citations are equally valuable: a citation in a paper that is itself often cited is more valuable than a citation in a paper that no one cites. Design a PageRank style algorithm which would rank papers according to their "importance", and then use such an algorithm to rank researchers by their "importance".
  2. Assume now that you do not have information on the citations in each published paper, but instead you have for every researcher a list of other researchers who have cited him and how many times they cited him. Design again a PageRank style algorithm which would rank researchers by their importance.
- At the class website you can find very detailed lecture notes on the PageRank which also contain a detailed mathematical analysis of the algorithm for those of you that are Math double majors.

The PageRank algorithm conclusion:

- While both the Markov chains and random walks on graphs have been studied "ad nauseam" and are thus far from being a novelty, the Google inventors deserve a huge credit for finding the ulti[illegible]t day application of these "ancient" concepts.
- The PageRank has seen [illegible]ations; at the class website you can find a paper done by a former 4121 st[illegible]en Rezvani who applied the PageRank to a security problem of evalu[illegible]sks of hosts and risk of flows in computer networks.
- As a homework, try applying the PageRank to the following problem:

  The present day "publish or perish" madness in academia involves counting number of papers researchers have published, as well as the number of citations their papers got.

  1. One might argue that not all citations are equally valuable: a citation in a paper that is itself often cited is more valuable than a citation in a paper that no one cites. Design a PageRank style algorithm which would rank papers according to their "importance", and then use such an algorithm to rank researchers by their "importance".
  2. Assume now that you do not have information on the citations in each published paper, but instead you have for every researcher a list of other researchers who have cited him and how many times they cited him. Design again a PageRank style algorithm which would rank researchers by their importance.
- At the class website you can find very detailed lecture notes on the PageRank which also contain a detailed mathematical analysis of the algorithm for those of you that are Math double majors.