Big Data Computing

Master's Degree in Computer Science

2025-2026

Gabriele Tolomei

Department of Computer Science

Sapienza Università di Roma

tolomei@di.uniroma1.it

SAPIENZA
UNIVERSITÀ DI ROMA

Goal: The "Importance" of a Node

- We want to find an effective way to measure the **trustworthiness** of a page within the Web graph

Goal: The "Importance" of a Node

- We want to find an effective way to measure the **trustworthiness** of a page within the Web graph
- More generally, we want to assign a score which indicates the **importance** of a node in a graph

Goal: The "Importance" of a Node

- We want to find an effective way to measure the **trustworthiness** of a page within the Web graph
- More generally, we want to assign a score which indicates the **importance** of a node in a graph
- Derive such a score from the structural properties of the graph only (i.e., via **link analysis**)

Goal: The "Importance" of a Node

- We want to find an effective way to measure the **trustworthiness** of a page within the Web graph
- More generally, we want to assign a score which indicates the **importance** of a node in a graph
- Derive such a score from the structural properties of the graph only (i.e., via **link analysis**)
- Exploit the fact that the Web is an example of a **scale-free network**

Computing Node Importance

Several link analysis approaches to compute web page importance

PageRank

Hubs and Authorities (HITS)

Personalized PageRank

Web Spam Detection

PageRank

One Slide PageRank

- A link analysis approach to the definition of web page importance

One Slide PageRank

- A link analysis approach to the definition of web page importance
- Introduced in 1998 by Sergey Brin and Larry Page*

*The Anatomy of a Large-Scale Hypertextual Web Search Engine. In Computer Networks, vol. 30, n. 1-7, pp. 107-117, 1998.

One Slide PageRank

- A link analysis approach to the definition of web page importance
- Introduced in 1998 by Sergey Brin and Larry Page*
- The core of Google search engine

*The Anatomy of a Large-Scale Hypertextual Web Search Engine. In Computer Networks, vol. 30, n. 1-7, pp. 107-117, 1998.

One Slide PageRank

- A link analysis approach to the definition of web page importance
- Introduced in 1998 by Sergey Brin and Larry Page*
- The core of Google search engine
- Assigns a numerical score to each web page with the purpose of indicating its relative importance within the whole collection

*The Anatomy of a Large-Scale Hypertextual Web Search Engine. In Computer Networks, vol. 30, n. 1-7, pp. 107-117, 1998.

PageRank's Intuition: Links as Votes

$\boxed{\text{Based on 2 intuitions}}$

PageRank's Intuition: Links as Votes

$\boxed{\text{Based on 2 intuitions}}$

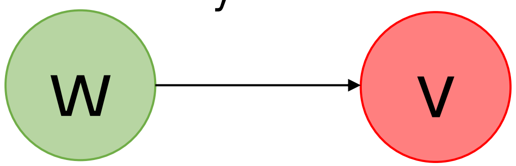
The more incoming links a web page has the more important it is

PageRank's Intuition: Links as Votes

Based on 2 intuitions

The more incoming links a web page has the more important it is

Each link from a web page w to a web page v is interpreted as a vote by w to v

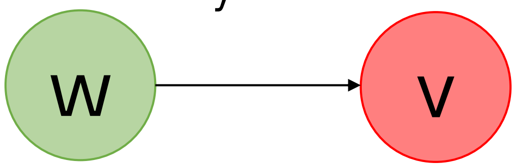


PageRank's Intuition: Links as Votes

Based on 2 intuitions

The more incoming links a web page has the more important it is

Each link from a web page w to a web page v is interpreted as a vote by w to v



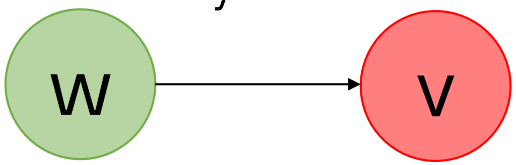
Links (i.e., votes) from important web pages should count more!

PageRank's Intuition: Links as Votes

Based on 2 intuitions

The more incoming links a web page has the more important it is

Each link from a web page w to a web page v is interpreted as a vote by w to v



Links (i.e., votes) from important web pages should count more!

Different web pages have different in-degree (scale-free network)

www.stanford.edu has more than 23K in-links

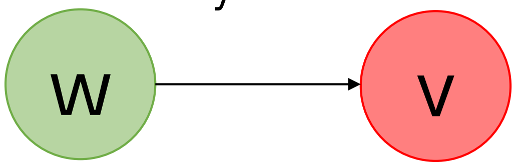
www.uniroma1.it/~tolomei has one or two in-links!

PageRank's Intuition: Links as Votes

Based on 2 intuitions

The more incoming links a web page has the more important it is

Each link from a web page w to a web page v is interpreted as a vote by w to v



Links (i.e., votes) from important web pages should count more!

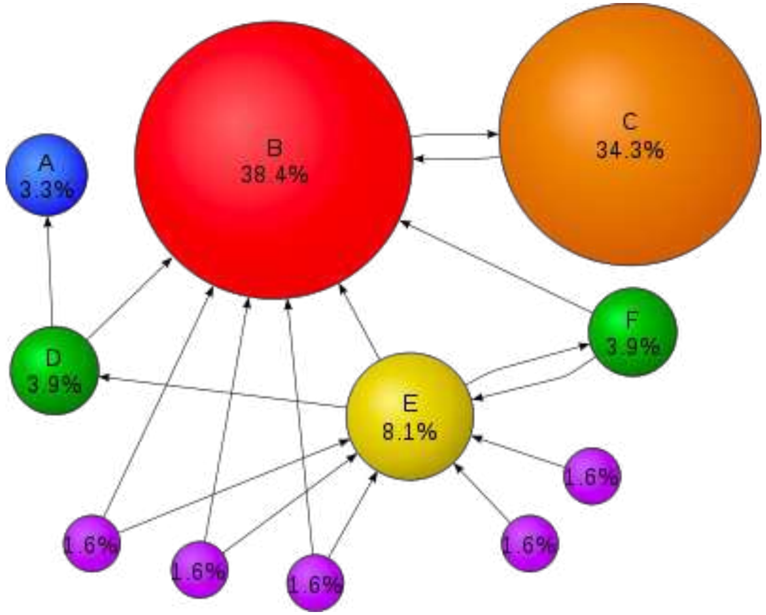
Different web pages have different in-degree (scale-free network)

www.stanford.edu has more than 23K in-links

www.uniroma1.it/~tolomei has one or two in-links!

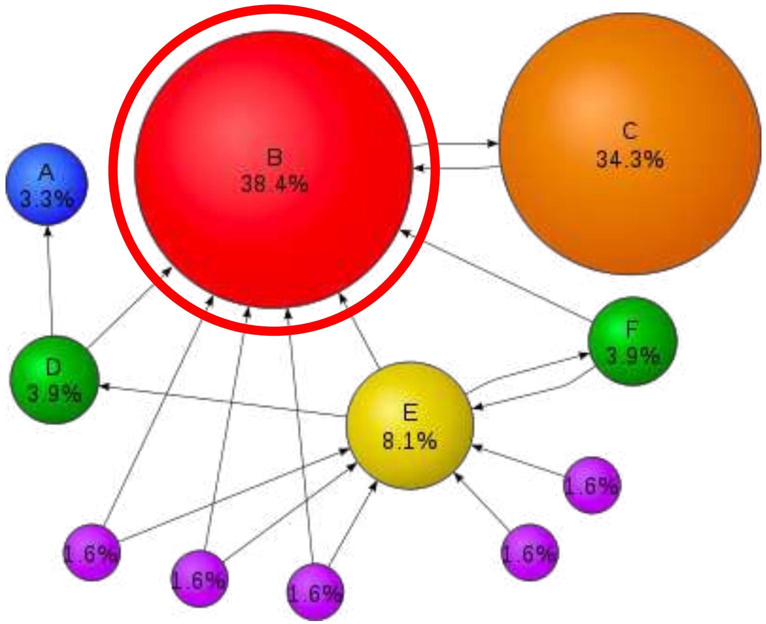
Recursive definition

PageRank Scores: Example



Circle size proportional to the node importance

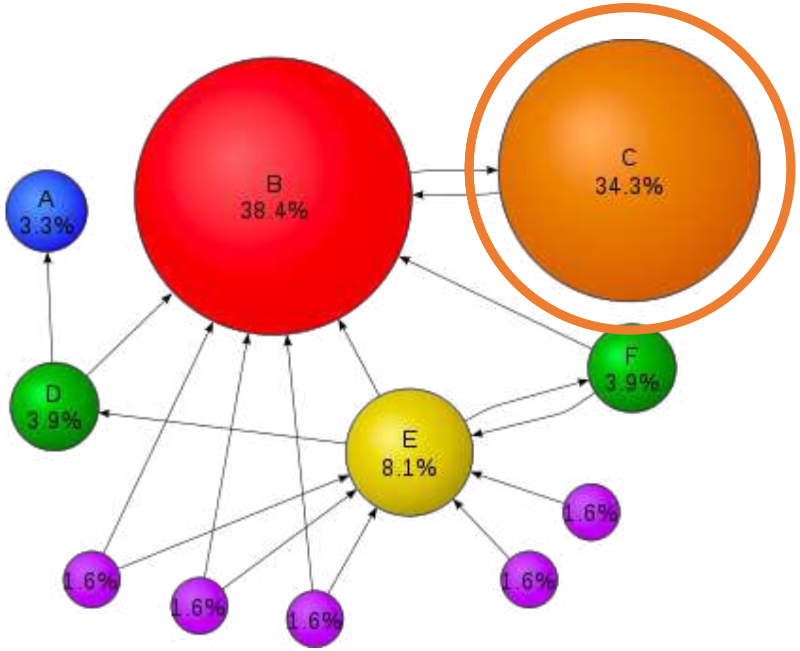
PageRank Scores: Example



Circle size proportional to the node importance

B has a high score since many nodes point to it

PageRank Scores: Example

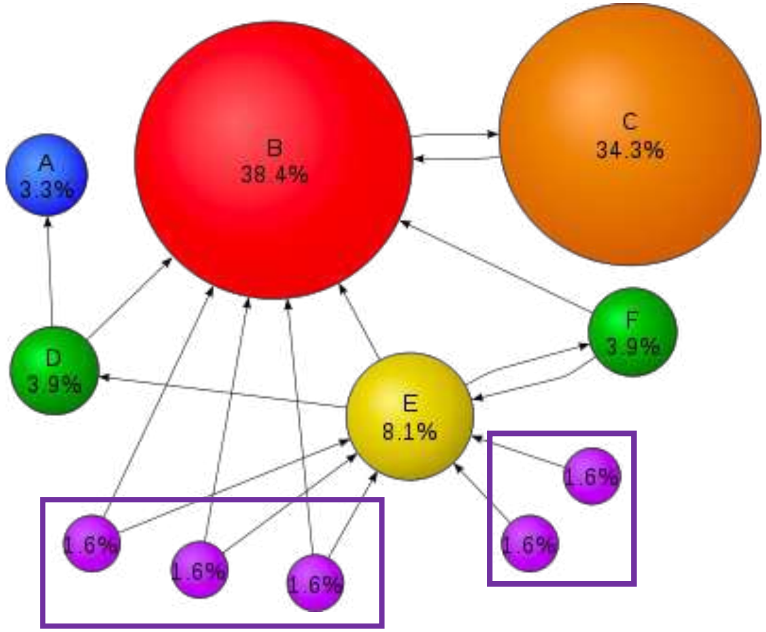


Circle size proportional to the node importance

B has a high score since many nodes point to it

C also has a high score even though it has only one incoming link but from an important node B

PageRank Scores: Example



Circle size proportional to the node importance

B has a high score since many nodes point to it

C also has a high score even though it has only one incoming link but from an important node B

Many other less important nodes

PageRank: Prelminaries

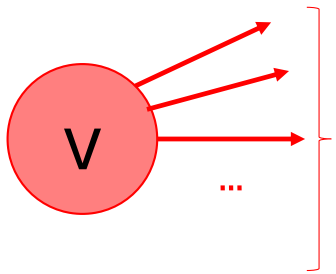
$G = (V, E)$ The Web Graph $|V| = N$ Number of Nodes (pages)

PageRank: Prelminaries

$G = (V, E)$ The Web Graph $|V| = N$ Number of Nodes (pages)

$O_v = \{w \in V : (v, w) \in E\}$ Set of pages linked by v

$|O_v| = o_v$ Out-degree of node v



PageRank: Prelminaries

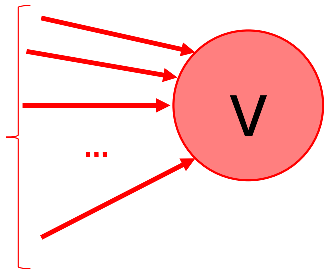
$G = (V, E)$ The Web Graph $|V| = N$ Number of Nodes (pages)

$O_v = \{w \in V : (v, w) \in E\}$ Set of pages linked by v

$|O_v| = o_v$ Out-degree of node v

$I_v = \{w \in V : (w, v) \in E\}$ Set of pages linked to v

$|I_v| = i_v$ In-degree of node v



PageRank: First Simple Recursive Formulation

Each link's vote to a page v is proportional to the importance of the source page w, which the link comes from

PageRank: First Simple Recursive Formulation

Each link's vote to a page v is proportional to the importance of the source page w, which the link comes from

If a page w has importance r_w and out-degree o_w, each out-link will get an equal proportion of the importance, i.e., r_w/o_w

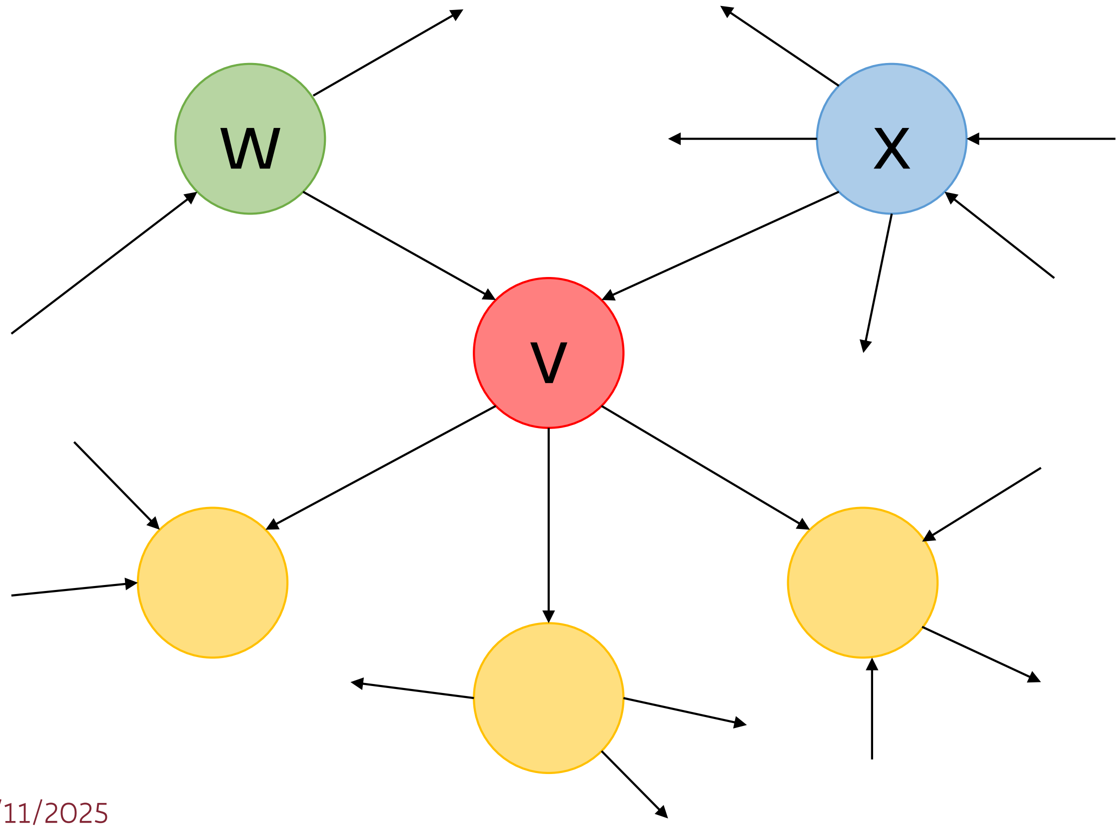
PageRank: First Simple Recursive Formulation

Each link's vote to a page v is proportional to the importance of the source page w, which the link comes from

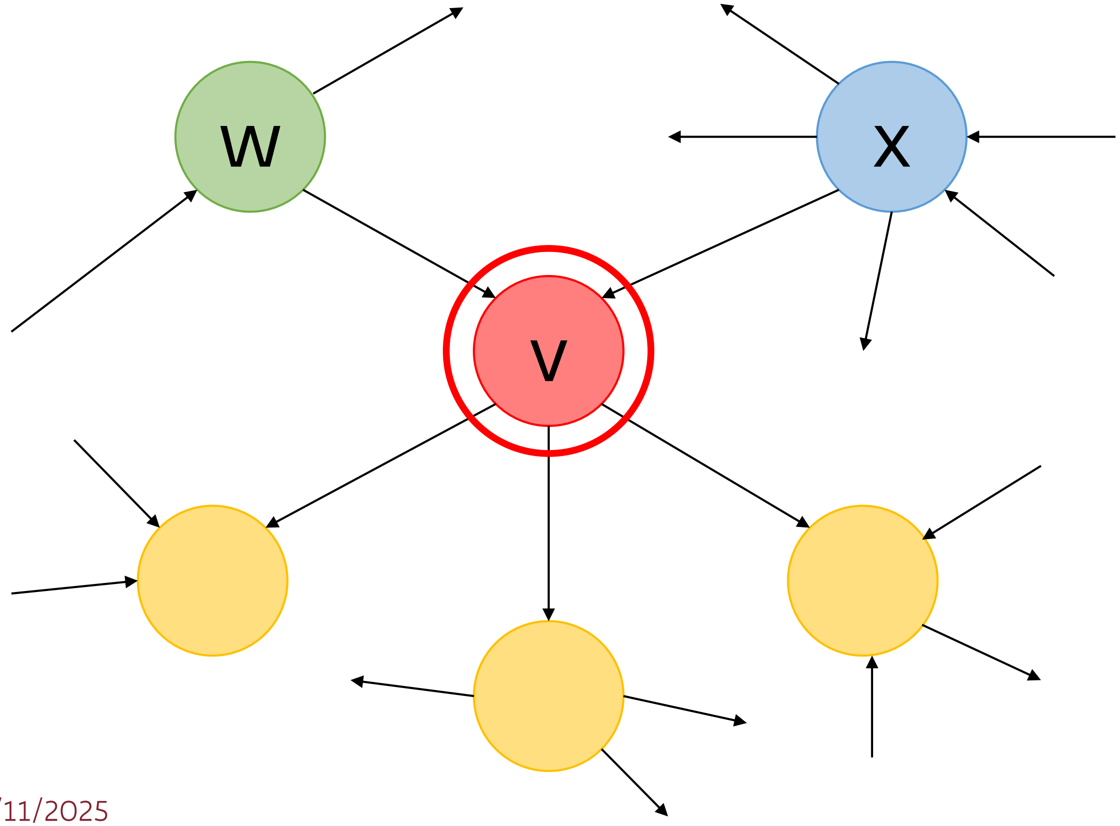
If a page w has importance r_w and out-degree o_w, each out-link will get an equal proportion of the importance, i.e., r_w/o_w

Each page v's importance can be computed just as the sum of votes of all its incoming links (i.e., in-degree)

PageRank: First Simple Recursive Formulation

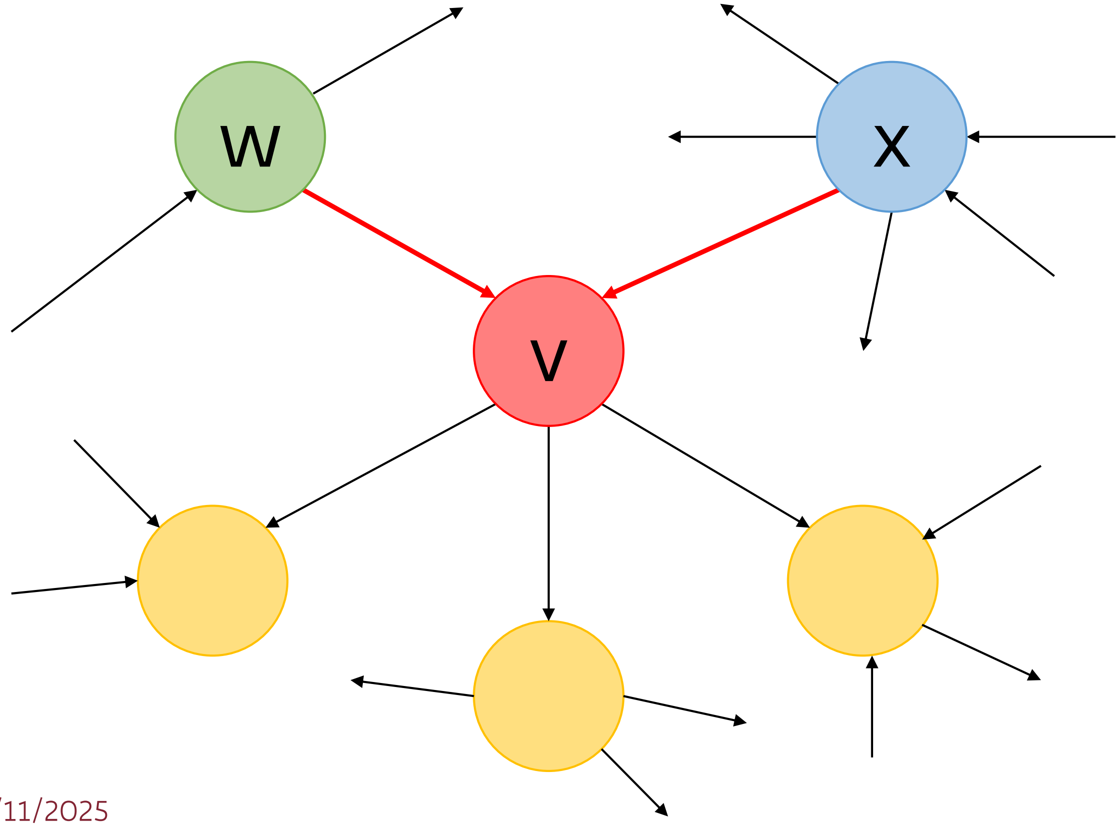


PageRank: First Simple Recursive Formulation



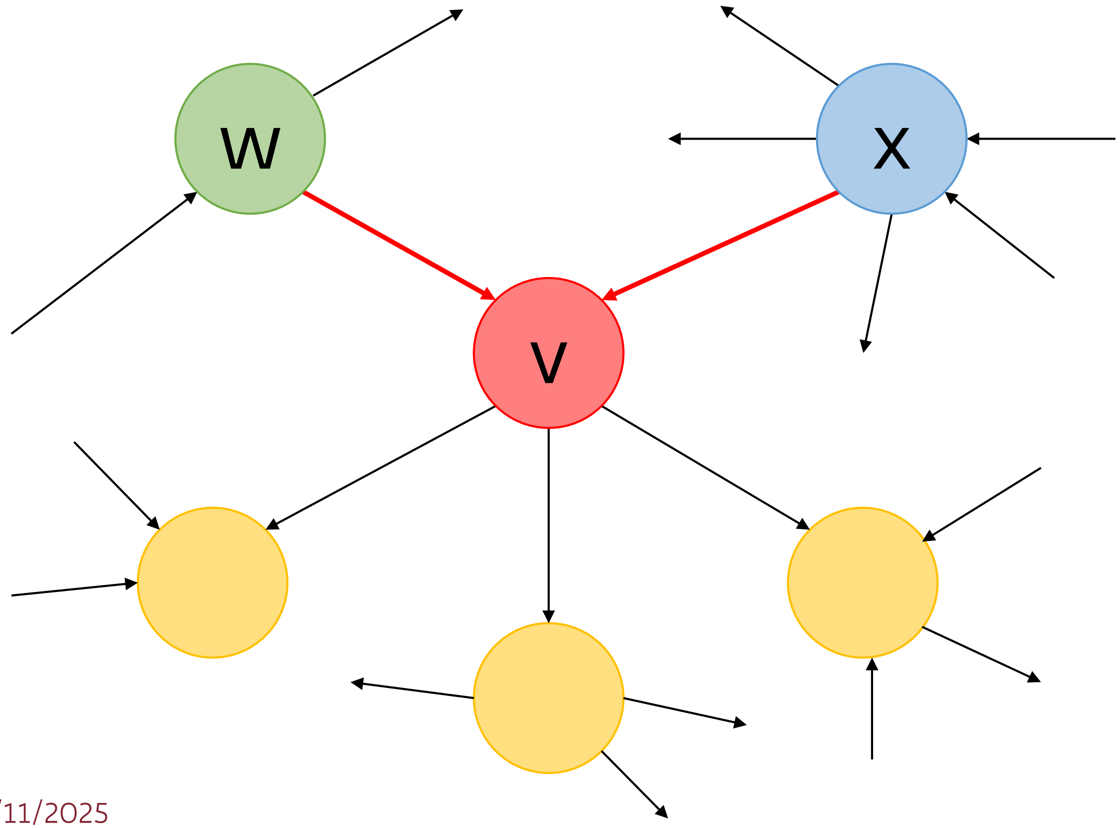
What is r_v?

PageRank: First Simple Recursive Formulation



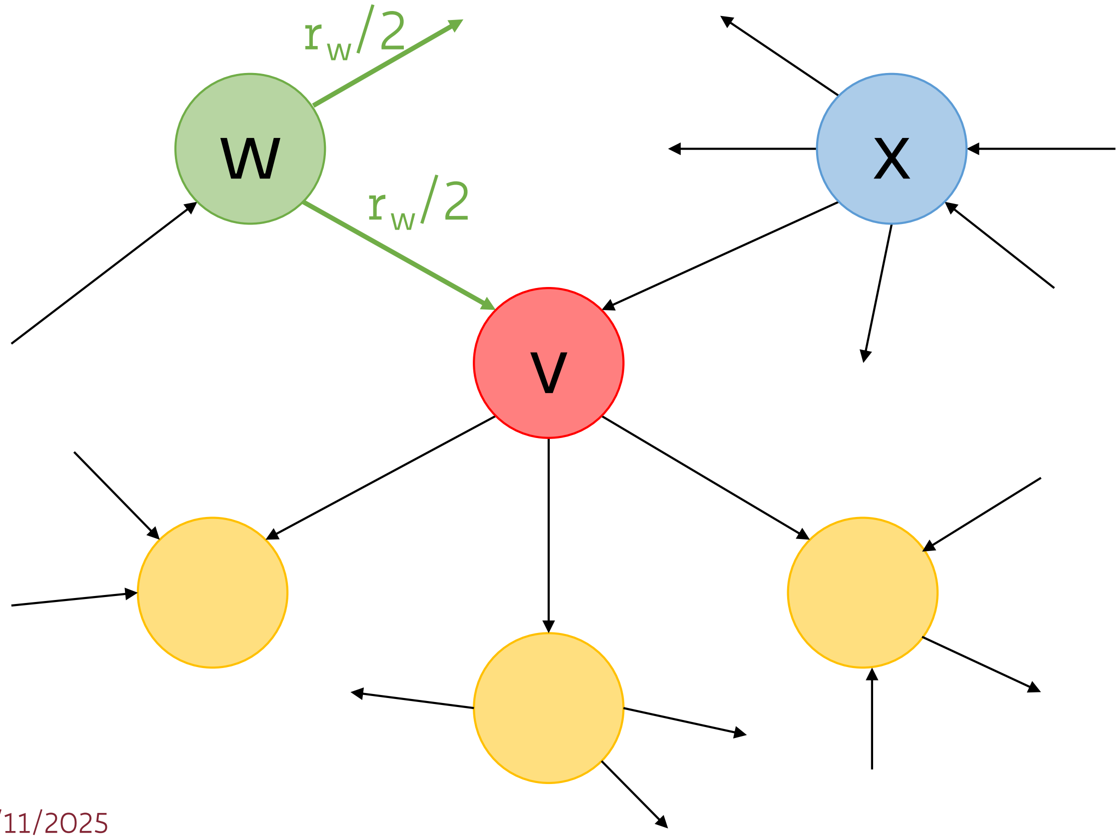
Suppose v has only 2 in-links coming from w and x

PageRank: First Simple Recursive Formulation



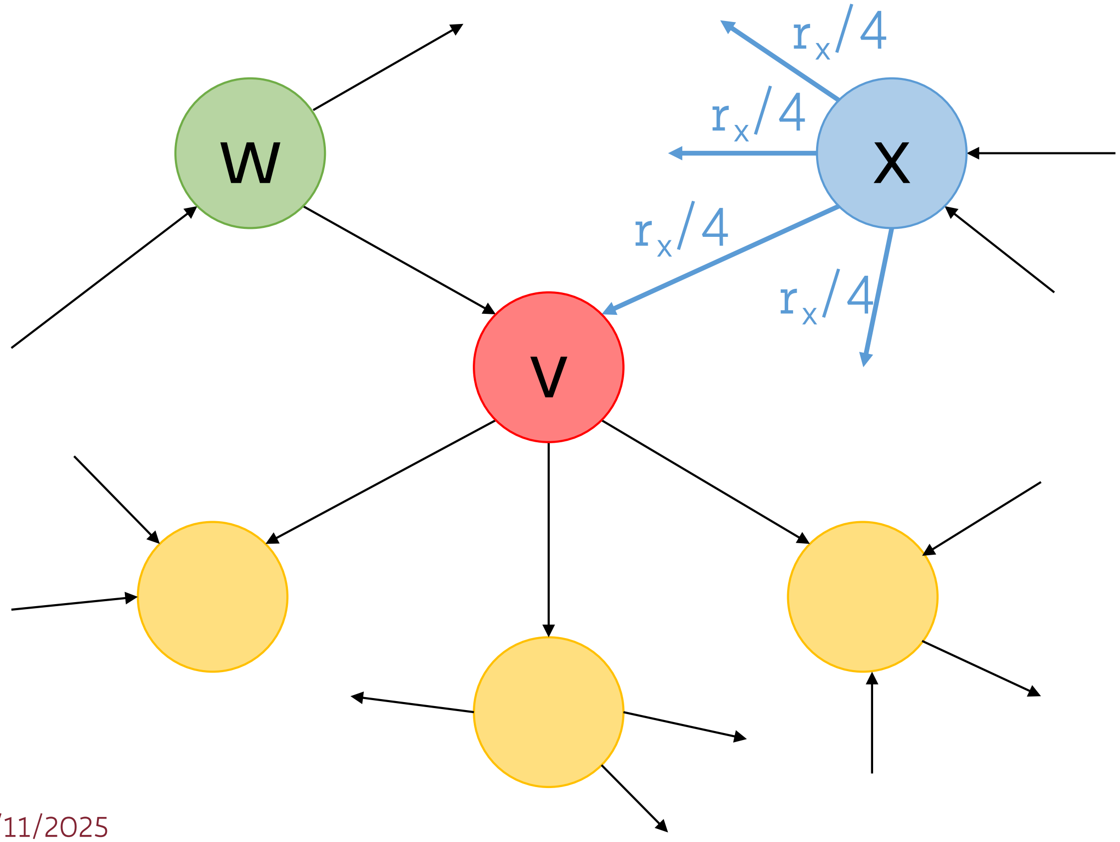
We must compute the in-link's vote from w and from x

PageRank: First Simple Recursive Formulation



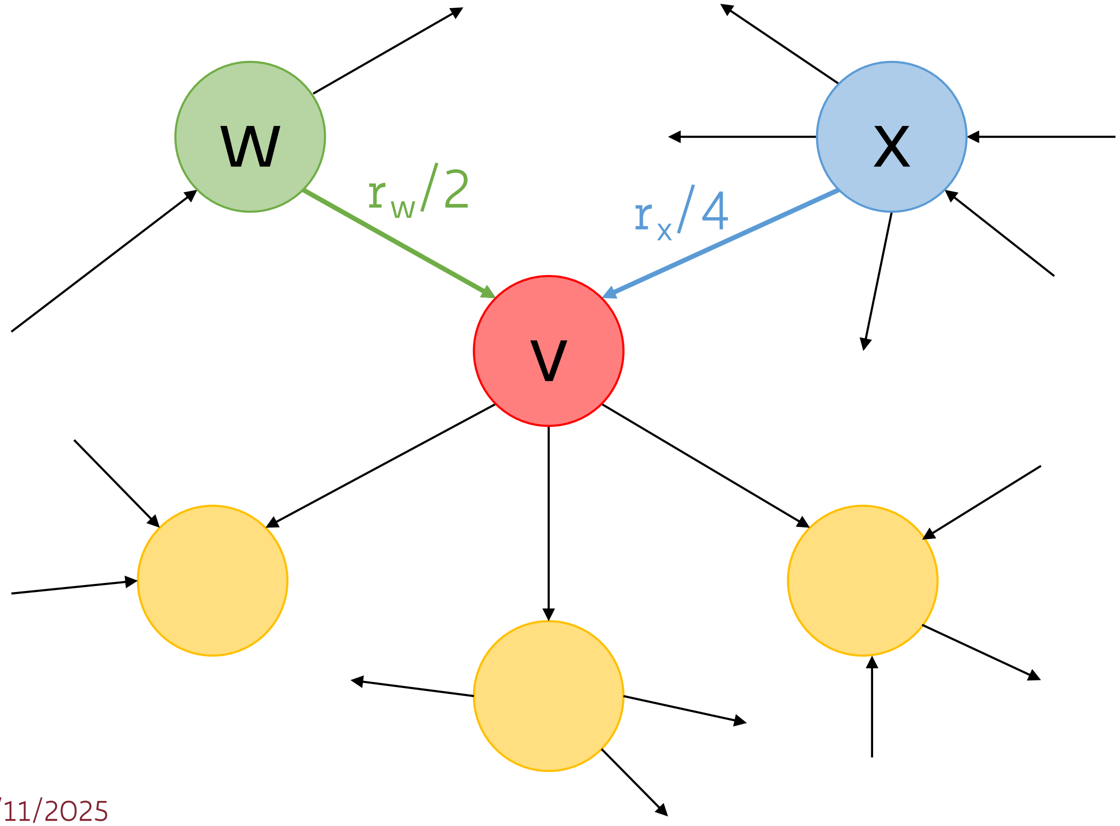
The importance of page w (r_w) is distributed across each of its 2 outgoing links

PageRank: First Simple Recursive Formulation



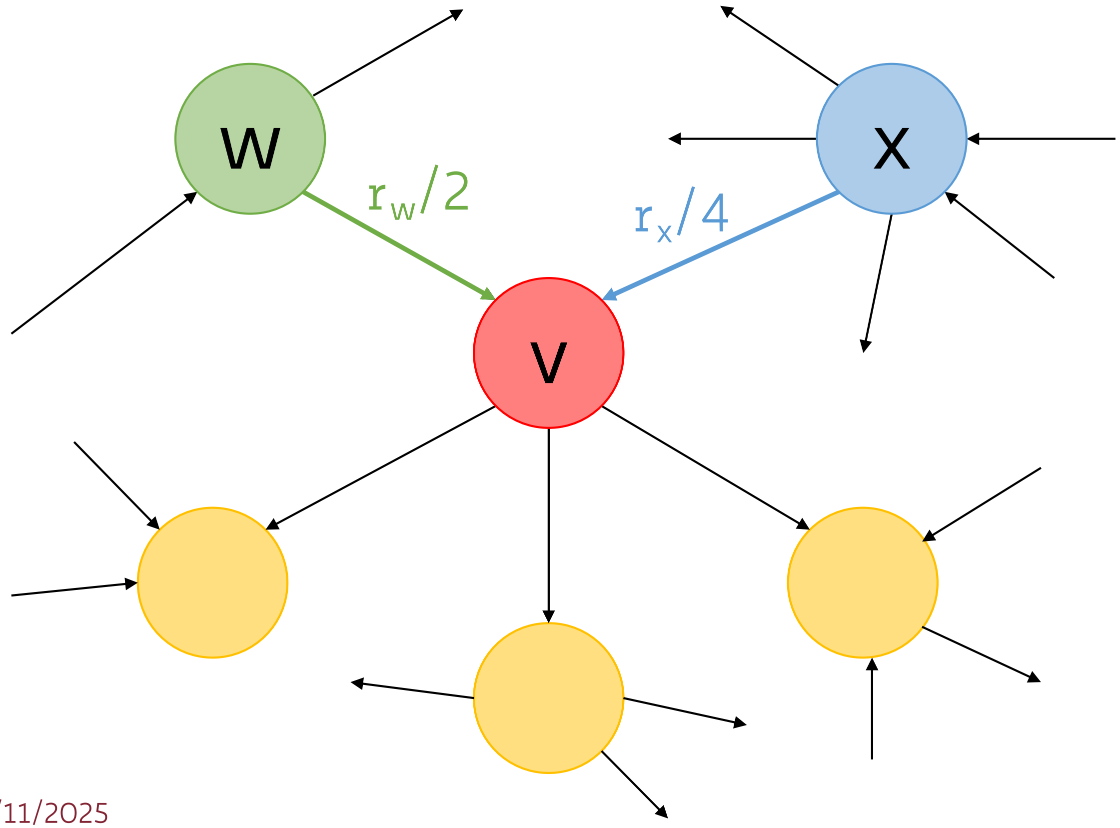
The importance of page x (r_x) is distributed across each of its 4 outgoing links

PageRank: First Simple Recursive Formulation



The importance of page v (r_v) is just the sum of its incoming links' votes

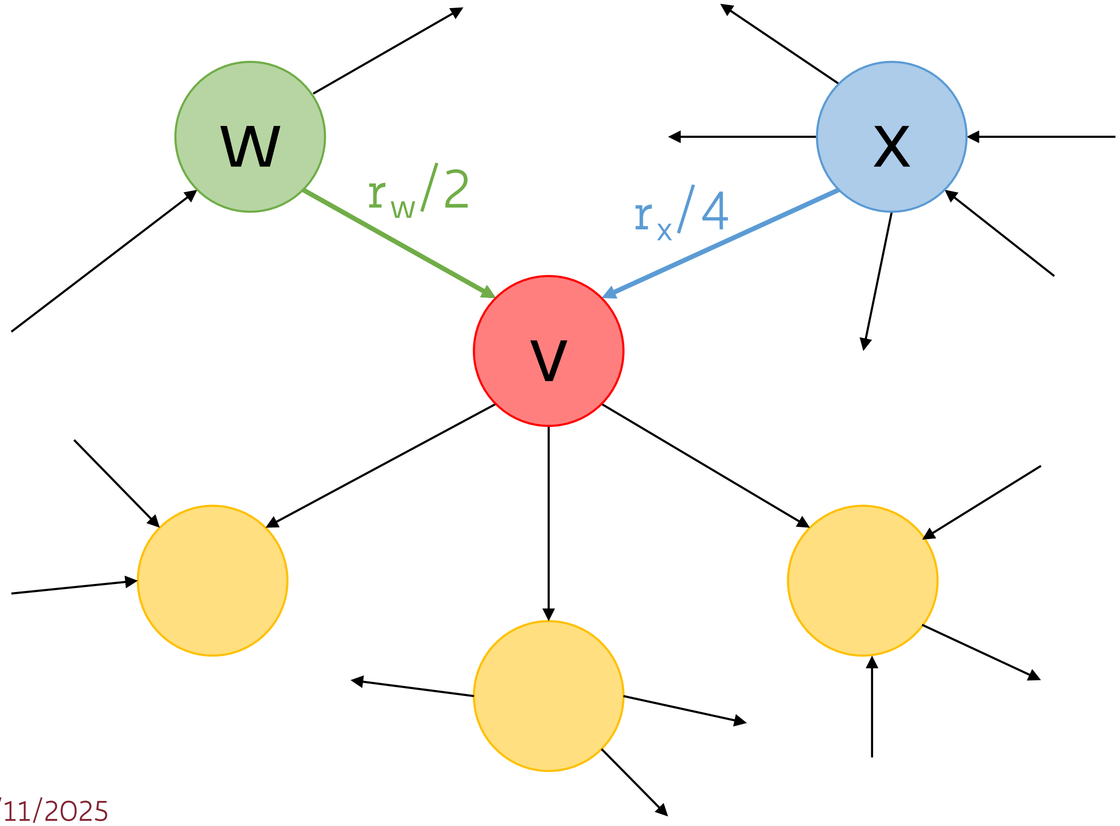
PageRank: First Simple Recursive Formulation



The importance of page v (r_v) is just the sum of its incoming links' votes

$$r_v = r_w/2 + r_x/4$$

PageRank: First Simple Recursive Formulation

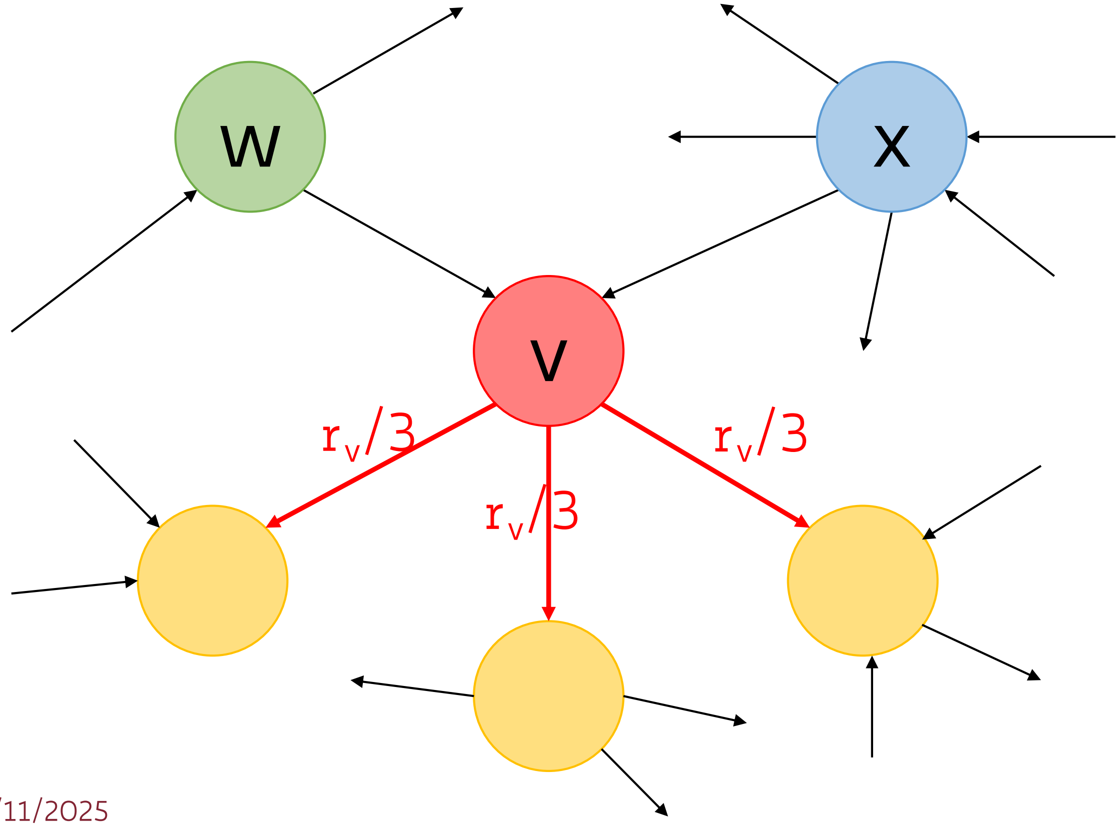


The importance of page v (r_v) is just the sum of its incoming links' votes

$$r_v = r_w/2 + r_x/4$$

$$r_v = \sum_{u \in I_v} \frac{r_u}{o_u}$$

PageRank: First Simple Recursive Formulation



Similarly, page v uniformly distributes its importance r_v to its outgoing links

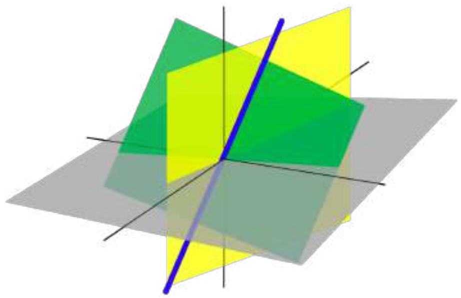
PageRank's Interpretations

$\boxed{\text{2 main perspectives}}$

PageRank's Interpretations

2 main perspectives

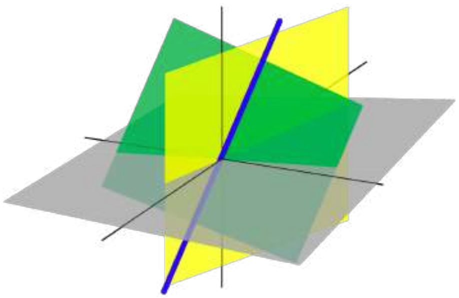
Linear Algebra



PageRank's Interpretations

2 main perspectives

Linear Algebra

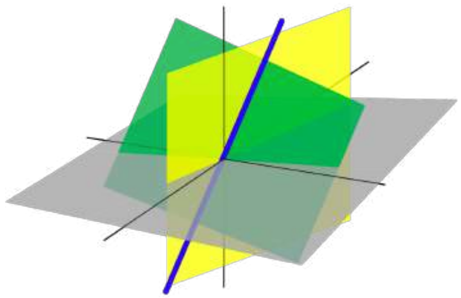


Probabilistic

PageRank's Interpretations

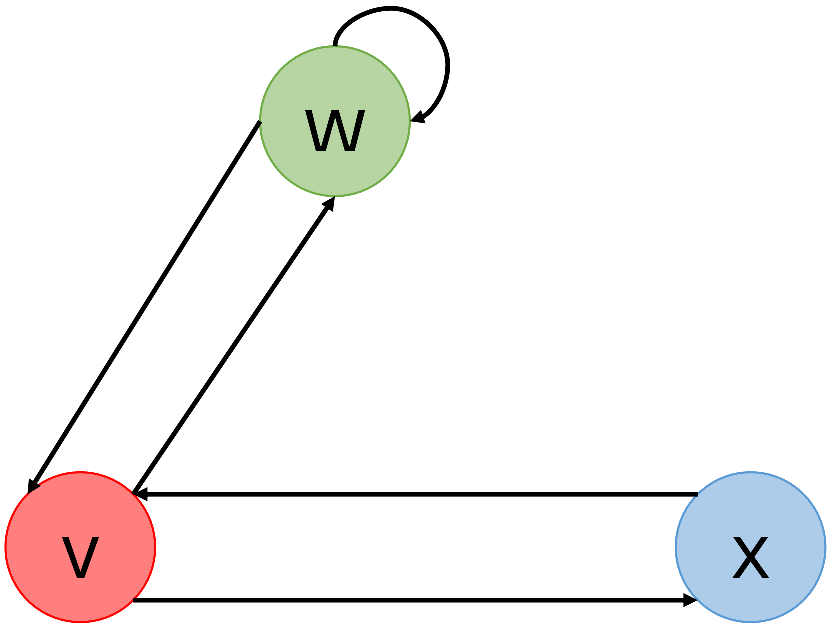
2 main perspectives

Linear Algebra

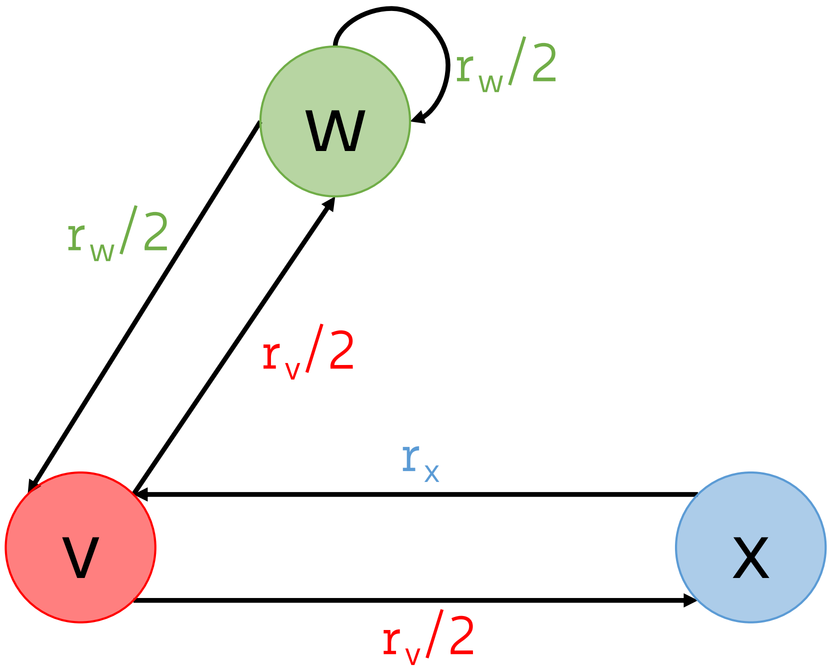


Probabilistic

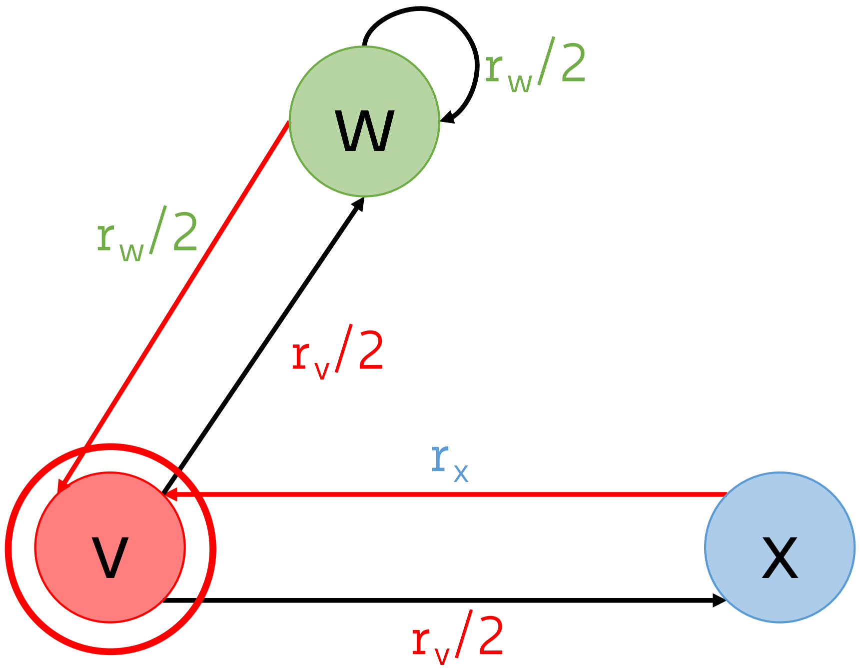
PageRank: The "Flow" Model



PageRank: The "Flow" Model

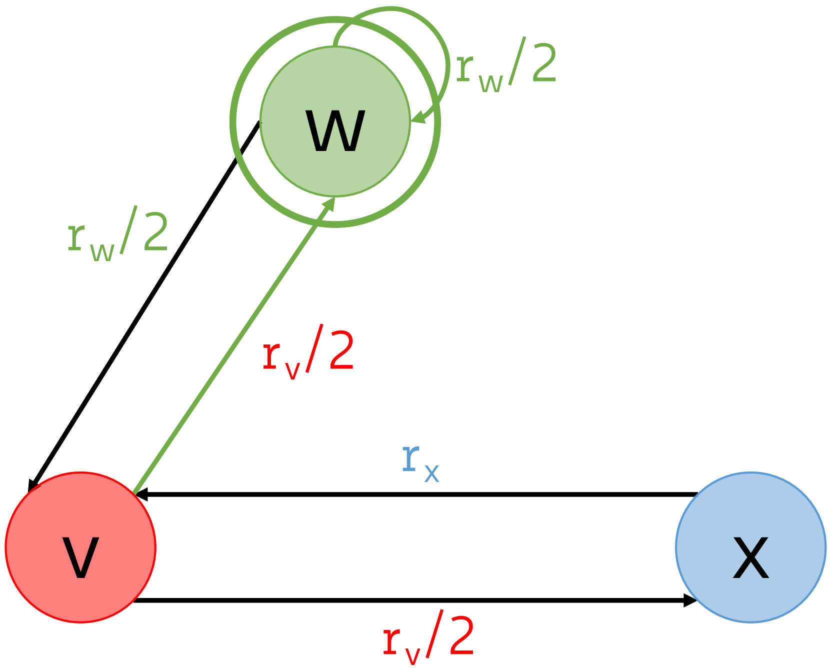


PageRank: The "Flow" Model



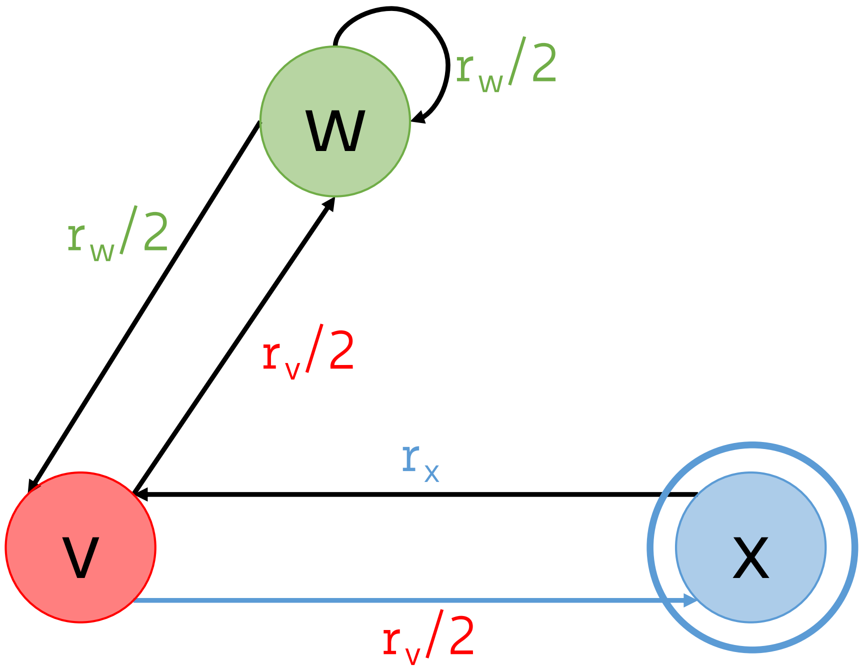
$$\begin{cases} \boxed{r_v = r_w/2 + r_x} \\ \\ \end{cases}$$

PageRank: The "Flow" Model



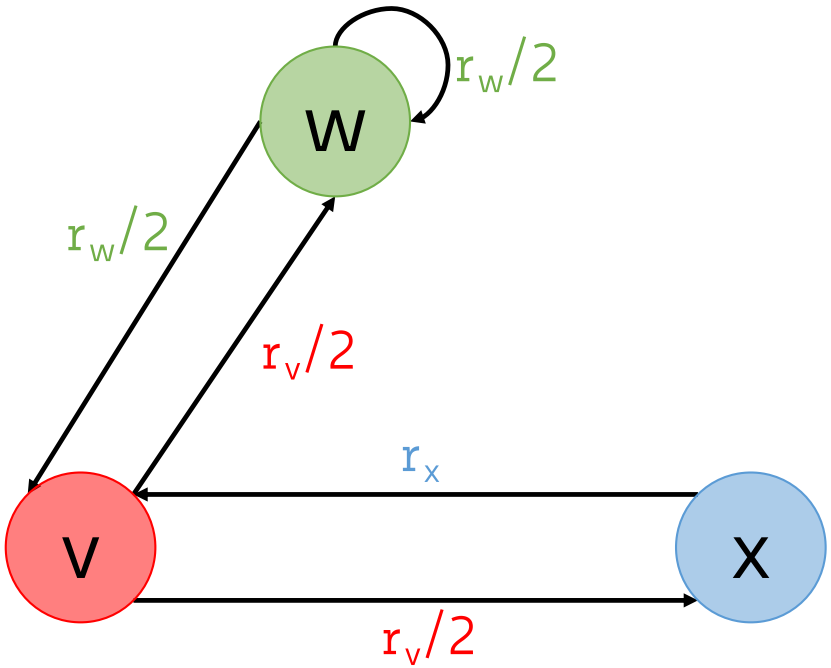
$$\begin{cases} r_v = r_w/2 + r_x \\ \boxed{r_w = r_v/2 + r_w/2} \end{cases}$$

PageRank: The "Flow" Model



$$
\begin{cases}
r_v = r_w/2 + r_x \\
r_w = r_v/2 + r_w/2 \\
\boxed{r_x = r_v/2}
\end{cases}
$$

PageRank: The "Flow" Model



$$
\begin{cases}
r_v = r_w/2 + r_x \\
r_w = r_v/2 + r_w/2 \\
r_x = r_v/2
\end{cases}
$$

"Flow" Equations

Solving the System of "Flow" Equations

$$\begin{cases} r_v = r_w/2 + r_x \\ r_w = r_v/2 + r_w/2 \\ r_x = r_v/2 \end{cases}$$

3 equations with 3 unknowns: r_v, r_w, and r_x

Solving the System of "Flow" Equations

$$\begin{cases} r_v = r_w/2 + r_x \\ r_w = r_v/2 + r_w/2 \\ r_x = r_v/2 \end{cases}$$

3 equations with 3 unknowns: r_v, r_w, and r_x

But the first 2 equations are exactly the same if we substitute r_x

Solving the System of "Flow" Equations

$$\begin{cases} r_v = r_w/2 + r_x \\ r_w = r_v/2 + r_w/2 \\ r_x = r_v/2 \end{cases}$$

3 equations with 3 unknowns: r_v, r_w, and r_x

But the first 2 equations are exactly the same if we substitute r_x

No unique solution!
Infinitely many apart from a constant scale factor

Solving the System of "Flow" Equations

$$
\begin{cases}
r_v = r_w/2 + r_x \\
r_w = r_v/2 + r_w/2 \\
r_x = r_v/2 \\
\boxed{r_v + r_w + r_x = 1}
\end{cases}
$$

Additional constraint (equation) enforces the uniqueness of the solution

Solving the System of "Flow" Equations

$$\begin{cases} r_v = r_w/2 + r_x \\ r_w = r_v/2 + r_w/2 \\ r_x = r_v/2 \\ r_v + r_w + r_x = 1 \end{cases}$$

Additional constraint (equation) enforces the uniqueness of the solution

$$\boxed{r_v = r_w = \frac{2}{5} \quad r_x = \frac{1}{5}}$$

Solving the System of "Flow" Equations

$$\begin{cases} r_v = r_w/2 + r_x \\ r_w = r_v/2 + r_w/2 \\ r_x = r_v/2 \\ r_v + r_w + r_x = 1 \end{cases}$$

Additional constraint (equation) enforces the uniqueness of the solution

$$r_v = r_w = \frac{2}{5} \quad r_x = \frac{1}{5}$$

This may work for very small systems of linear equations
(e.g., using Gaussian elimination)

Solving the System of "Flow" Equations

$$\begin{cases} r_v = r_w/2 + r_x \\ r_w = r_v/2 + r_w/2 \\ r_x = r_v/2 \\ r_v + r_w + r_x = 1 \end{cases}$$

Additional constraint (equation) enforces the uniqueness of the solution

$$r_v = r_w = \frac{2}{5} \quad r_x = \frac{1}{5}$$

In the case of web pages we might have 100s of billions of equations!

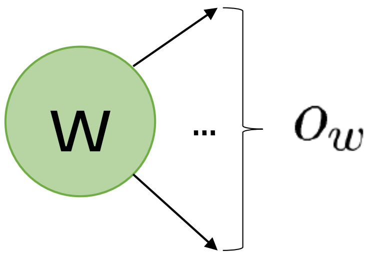
We need a new formulation

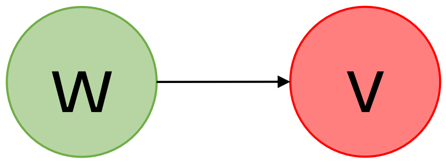
PageRank: The Matrix Formulation

Represent the Web graph of documents G=(V, E) s.t. |V|=N as a column stochastic matrix M of size NxN

PageRank: The Matrix Formulation

Represent the Web graph of documents G=(V, E) s.t. |V|=N
as a column stochastic matrix M of size NxN

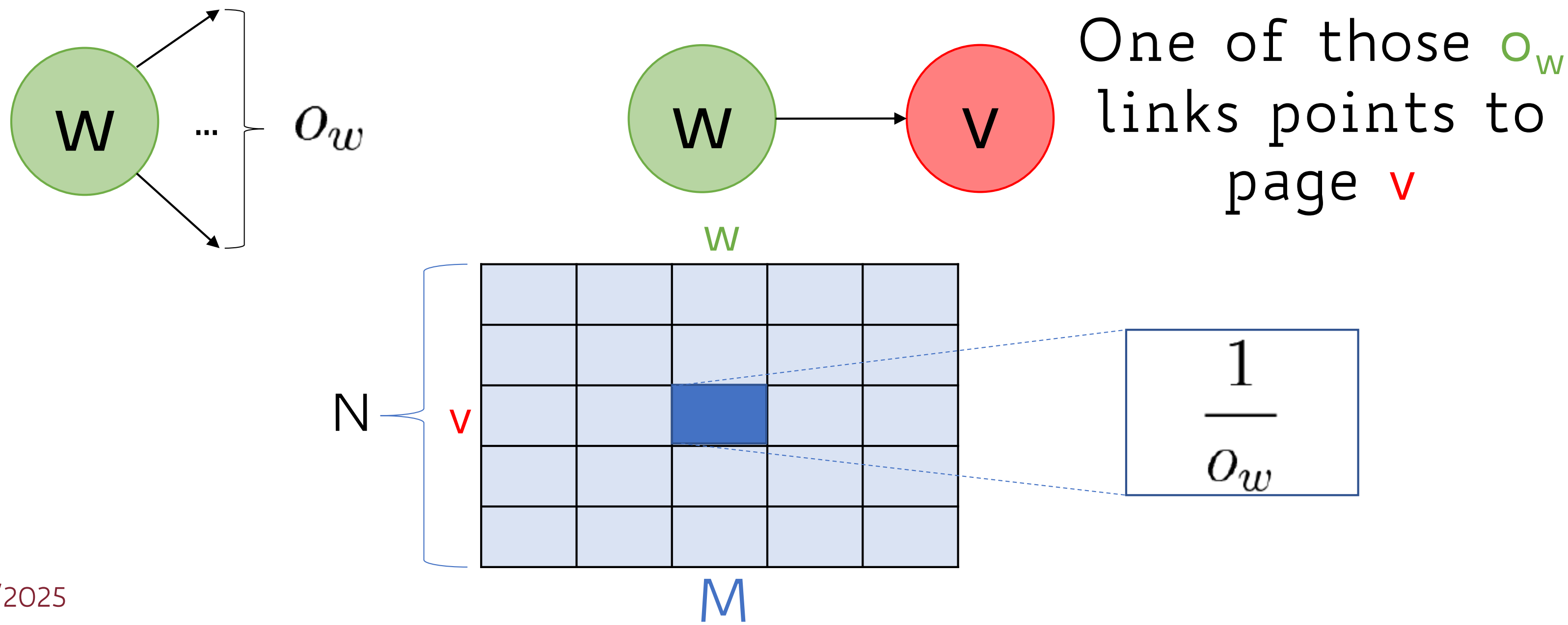




One of those o_w links points to page v

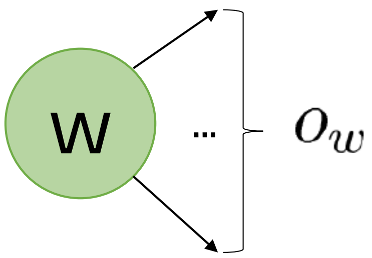
PageRank: The Matrix Formulation

Represent the Web graph of documents G=(V, E) s.t. |V|=N as a column stochastic matrix M of size NxN

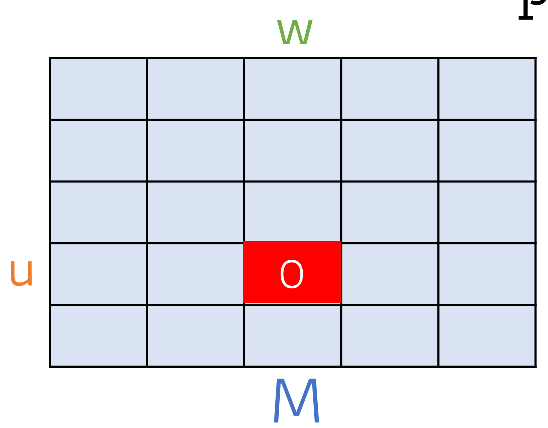


PageRank: The Matrix Formulation

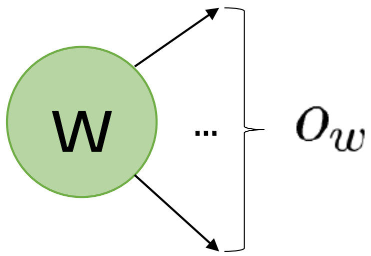
Represent the Web graph of documents G=(V, E) s.t. |V|=N
as a column stochastic matrix M of size NxN



For any other page u which w is not pointing to M[u, w] = 0

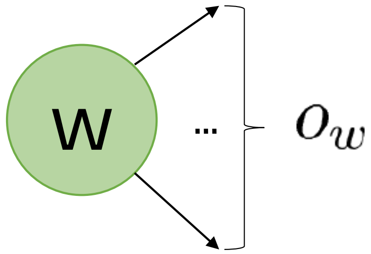


PageRank: The Matrix Formulation

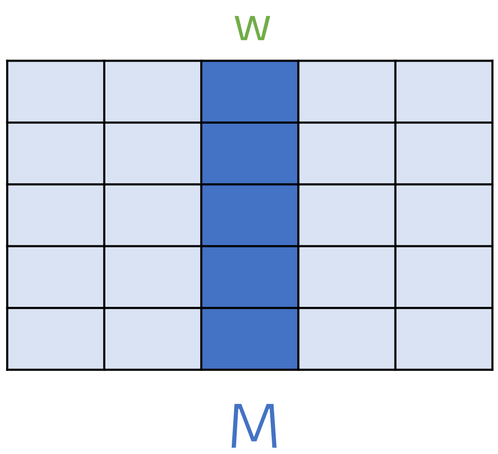


M is column stochastic because, by design, each of its column sums up to 1

PageRank: The Matrix Formulation



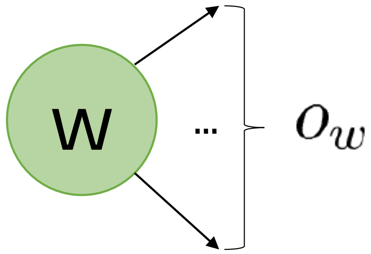
M is column stochastic because, by design, each of its column sums up to 1



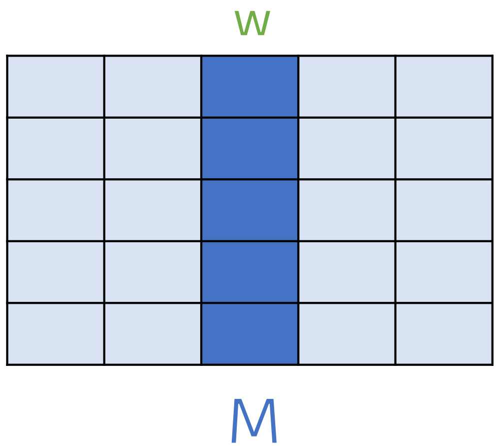
The w-th column will contain o_w <= N non-zero entries, each evaluating to $1/o_w$

$$\sum_{v=1}^{N} m_{v,w} = o_w \times \frac{1}{o_w} = 1$$

PageRank: The Matrix Formulation



M is column stochastic because, by design, each of its column sums up to 1



<u>Note:</u>
We are implicitly assuming there exists at least one outgoing link from each node

A Formal View of the Matrix **M**

$$\mathbf{A}_{N \times N} \quad a_{v,w} = \begin{cases} 1 & \text{if } w \in O_v \\ 0 & \text{otherwise} \end{cases}$$ Traditional adjacency matrix

A Formal View of the Matrix $\mathbf{M}$

$$\mathbf{A}_{N \times N} \quad a_{v,w} = \begin{cases} 1 & \text{if } w \in O_v \\ 0 & \text{otherwise} \end{cases}$$

Traditional adjacency matrix

$$\mathbf{L}_{N \times N} \quad l_{v,w} = \begin{cases} o_v & \text{if } v = w \\ 0 & \text{otherwise} \end{cases}$$

Diagonal matrix of out-degrees

A Formal View of the Matrix M

$$\mathbf{A}_{N\times N} \quad a_{v,w} = \begin{cases} 1 & \text{if } w \in O_v \\ 0 & \text{otherwise} \end{cases}$$

Traditional adjacency matrix

$$\mathbf{L}_{N\times N} \quad l_{v,w} = \begin{cases} o_v & \text{if } v = w \\ 0 & \text{otherwise} \end{cases}$$

Diagonal matrix of out-degrees

$$\mathbf{M}_{N\times N} \quad m_{v,w} = \begin{cases} \frac{1}{o_w} & \text{if } v \in O_w \\ 0 & \text{otherwise} \end{cases}$$

Column stochastic matrix

$$\boxed{\mathbf{M} = (\mathbf{L}^{-1}\mathbf{A})^T}$$

PageRank: The Matrix Formulation

$\mathbf{r}$ Nx1 rank vector with an entry for each page

PageRank: The Matrix Formulation

$\mathbf{r}$ Nx1 rank vector with an entry for each page

r_v Rank score of page v

PageRank: The Matrix Formulation

$\mathbf{r}$ Nx1 rank vector with an entry for each page

r_v

Rank score of page v

$$\sum_{v=1}^{N} r_v = 1$$

All the rank scores must sum up to 1

PageRank: The Matrix Formulation

$\mathbf{r}$ Nx1 rank vector with an entry for each page

r_v  Rank score of page v

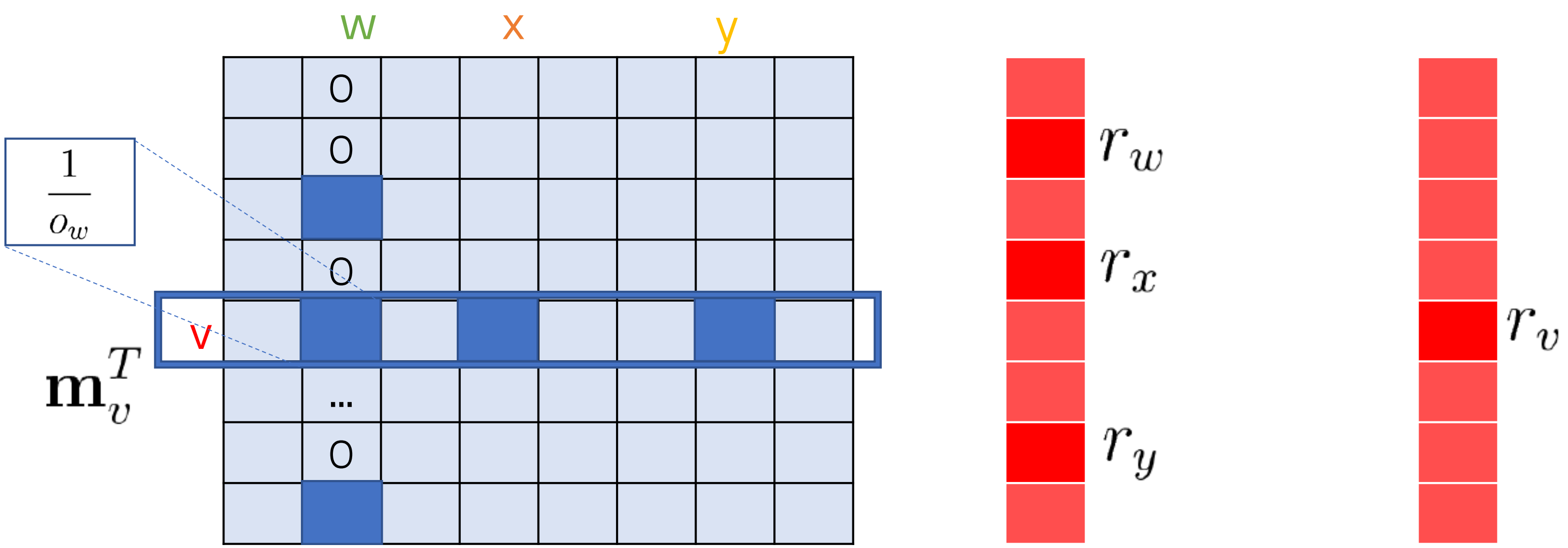
$$\sum_{v=1}^{N} r_v = 1$$

All the rank scores must sum up to 1

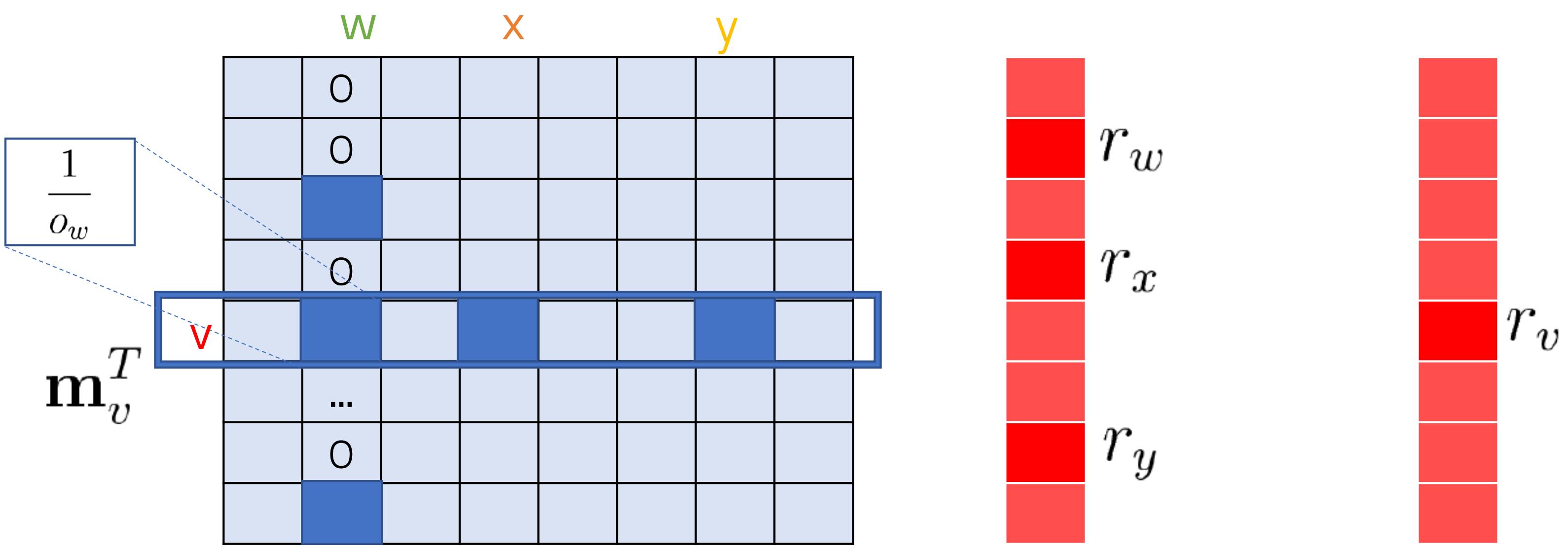
$$r_v = \sum_{w \in I_v} \frac{r_w}{o_w} \quad \Rightarrow \quad \mathbf{r} = \mathbf{M}\mathbf{r}$$

Flow equations in matrix form

PageRank: The Matrix Formulation

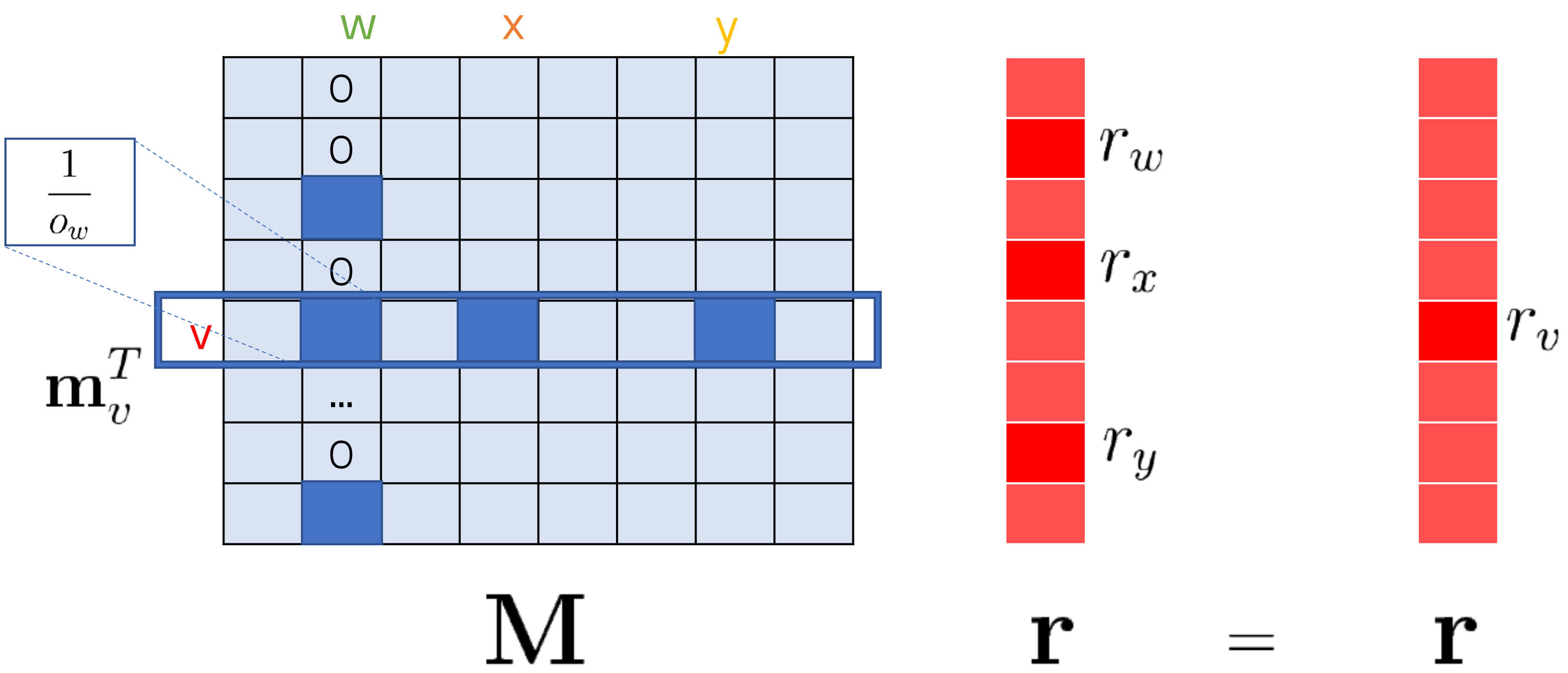


PageRank: The Matrix Formulation



$$r_v = \mathbf{m}_v^T \cdot \mathbf{r} = \sum_{w=1}^{N} m_{v,w} \times r_w = \sum_{w=1}^{N} \frac{1}{o_w} \times r_w = \sum_{w=1}^{N} \frac{r_w}{o_w} = \sum_{w \in I_v} \frac{r_w}{o_w}$$

PageRank: The Matrix Formulation



PageRank: The Eigenvector Formulation

$$\boxed{\mathbf{Mr} = \mathbf{r}}$$

Doesn't it look familiar?

PageRank: The Eigenvector Formulation

$$\boxed{\mathbf{Mr} = \mathbf{r}}$$

Doesn't it look familiar?

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

x is an eigenvector

λ is an eigenvalue

PageRank: The Eigenvector Formulation

$$\boxed{\mathbf{Mr} = \mathbf{r}}$$

Doesn't it look familiar?

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

x is an eigenvector

λ is an eigenvalue

So, the rank vector r is an eigenvector of the matrix M

PageRank: The Eigenvector Formulation

$$\boxed{\mathbf{Mr} = \mathbf{r}}$$

Doesn't it look familiar?

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

x is an eigenvector

λ is an eigenvalue

So, the rank vector r is an eigenvector of the matrix M

In fact, r is the eigenvector corresponding to the eigenvalue λ = 1

PageRank: The Eigenvector Formulation

$$\boxed{\mathbf{Mr} = \mathbf{r}}$$

For a fixed eigenvalue, eigenvectors are just scalar multiples of each other

PageRank: The Eigenvector Formulation

$$\boxed{\mathbf{Mr} = \mathbf{r}}$$

For a fixed eigenvalue, eigenvectors are just scalar multiples of each other

We can choose any of them to be our PageRank vector r

PageRank: The Eigenvector Formulation

$$\boxed{\mathbf{Mr} = \mathbf{r}}$$

For a fixed eigenvalue, eigenvectors are just scalar multiples of each other

We can choose any of them to be our PageRank vector r

Since PageRank should reflect only the relative importance of the nodes, choose $r = r^*$ as the eigenvector whose entries sum up to 1

PageRank: The Eigenvector Formulation

$$\boxed{\mathbf{Mr} = \mathbf{r}}$$

For a fixed eigenvalue, eigenvectors are just scalar multiples of each other

We can choose any of them to be our PageRank vector r

Since PageRank should reflect only the relative importance of the nodes, choose $r = r^*$ as the eigenvector whose entries sum up to 1

This may be referred to as the probabilistic eigenvector corresponding to the eigenvalue $\lambda = 1$

PageRank: The Eigenvector Formulation

$$\boxed{\mathbf{Mr} = \mathbf{r}}$$

We know from linear algebra theory that for any stochastic matrix M its largest eigenvalue is $\lambda = 1$

PageRank: The Eigenvector Formulation

$$\boxed{\mathbf{Mr} = \mathbf{r}}$$

We know from linear algebra theory that for any stochastic matrix M its largest eigenvalue is $\lambda = 1$

Therefore, $r = r^*$ is the principal eigenvector of M
(i.e., the eigenvector associated with the largetst eigenvalue)

PageRank: The Eigenvector Formulation

$$\boxed{\mathbf{Mr} = \mathbf{r}}$$

We know from linear algebra theory that for any stochastic matrix M its largest eigenvalue is $\lambda = 1$

Therefore, $r = r^*$ is the principal eigenvector of M (i.e., the eigenvector associated with the largetst eigenvalue)

Note:

So far, we have assumed that M is (column) stochastic yet this may not be the case for the general Web graph...

PageRank: Quick Recap

We start from "flow" equations

PageRank: Quick Recap

We start from "flow" equations

We reformulate the system of linear equations using linear algebra
(i.e., stochastic matrix M and rank vector r)

PageRank: Quick Recap

We start from "flow" equations

We reformulate the system of linear equations using linear algebra
(i.e., stochastic matrix M and rank vector r)

We reduce to finding the eigenvector of the matrix M

PageRank: Quick Recap

We start from "flow" equations

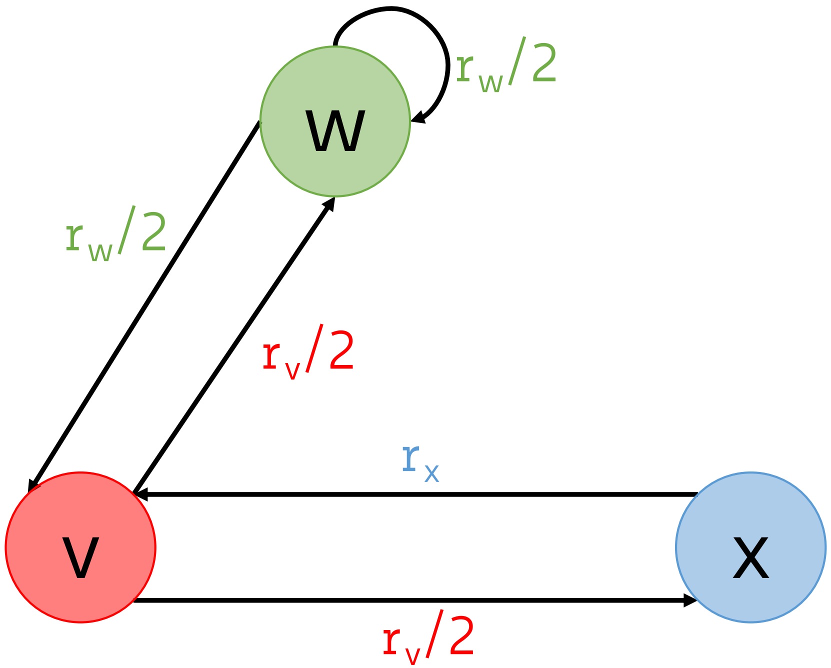
We reformulate the system of linear equations using linear algebra
(i.e., stochastic matrix M and rank vector r)

We reduce to finding the eigenvector of the matrix M

⇩

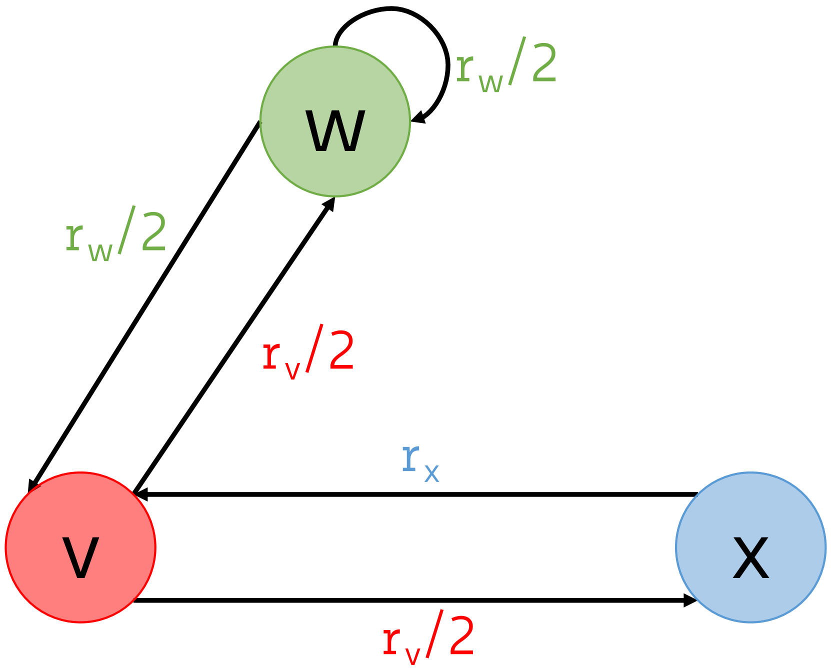
We know how to solve this efficiently using power iteration method

PageRank: The "Flow" Model



$$\begin{cases} r_v = r_w/2 + r_x \\ r_w = r_v/2 + r_w/2 \\ r_x = r_v/2 \end{cases}$$

PageRank: The "Flow" Model



$$\begin{cases} r_v = r_w/2 + r_x \\ r_w = r_v/2 + r_w/2 \\ r_x = r_v/2 \end{cases}$$

0	1/2	1
1/2	1/2	0
1/2	0	0

$$\mathbf{r} = \mathbf{M} \quad \mathbf{r}$$

PageRank: Power Iteration Method

At the beginning, we assume all pages have the same rank score, uniformly distributed across the N pages

$$\textbf{init:}\ \ t = 0; \mathbf{r}(t) = (1/N, 1/N, \ldots, 1/N)^T$$

PageRank: Power Iteration Method

Keep updating the rank vector r until convergence

$$\textbf{init:}\ t = 0; \mathbf{r}(t) = (1/N, 1/N, \ldots, 1/N)^T$$

repeat:

$$\mathbf{r}(t+1) = \mathbf{M}\mathbf{r}(t)$$

$$\textbf{until}\ \delta(\mathbf{r}(t+1), \mathbf{r}(t)) < \epsilon$$

$$\epsilon > 0$$

PageRank: Power Iteration Method

$$\textbf{init: } t = 0; \mathbf{r}(t) = (1/N, 1/N, \ldots, 1/N)^T$$

repeat:

$$\mathbf{r}(t+1) = \mathbf{M}\mathbf{r}(t)$$

$$\textbf{until } \delta(\mathbf{r}(t+1), \mathbf{r}(t)) < \epsilon$$

$$\epsilon > 0$$

$$\left\{ \begin{array}{c} \delta(\mathbf{r}(t+1), \mathbf{r}(t)) = |\mathbf{r}(t+1) - \mathbf{r}(t)| \\ \text{or} \\ \delta(\mathbf{r}(t+1), \mathbf{r}(t)) = ||\mathbf{r}(t+1) - \mathbf{r}(t)|| \end{array} \right.$$

Power Iteration Method: Example

$$\begin{cases} r_v = r_w/2 + r_x \\ r_w = r_v/2 + r_w/2 \\ r_x = r_v/2 \end{cases}$$

0	1/2	1
1/2	1/2	0
1/2	0	0

$$\mathbf{r} = \mathbf{M} \quad \mathbf{r}$$

Power Iteration Method: Example

$$\begin{cases} r_v = r_w/2 + r_x \\ r_w = r_v/2 + r_w/2 \\ r_x = r_v/2 \end{cases}$$

$$\underset{\mathbf{r}}{\begin{bmatrix} \; \\ \; \\ \; \end{bmatrix}} = \underset{\mathbf{M}}{\begin{bmatrix} 0 & 1/2 & 1 \\ 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \end{bmatrix}} \underset{\mathbf{r}}{\begin{bmatrix} \; \\ \; \\ \; \end{bmatrix}}$$

$$\underset{\mathbf{r}(0)}{\begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}}$$

Power Iteration Method: Example

$$\begin{cases} r_v = r_w/2 + r_x \\ r_w = r_v/2 + r_w/2 \\ r_x = r_v/2 \end{cases}$$

$$\mathbf{r} = \mathbf{M} \mathbf{r}, \qquad \mathbf{M} = \begin{bmatrix} 0 & 1/2 & 1 \\ 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \end{bmatrix}$$

$$\mathbf{r}(0) = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \qquad \mathbf{r}(1) = \mathbf{M}\mathbf{r}(0) = \begin{bmatrix} 3/6 \\ 1/3 \\ 1/6 \end{bmatrix}$$

Power Iteration Method: Example

$$\begin{cases} r_v = r_w/2 + r_x \\ r_w = r_v/2 + r_w/2 \\ r_x = r_v/2 \end{cases}$$

$$\mathbf{r} = \mathbf{M} \, \mathbf{r}$$

0	1/2	1
1/2	1/2	0
1/2	0	0

$$\mathbf{r}(0) = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \quad \mathbf{r}(1) = \mathbf{M}\mathbf{r}(0) = \begin{bmatrix} 3/6 \\ 1/3 \\ 1/6 \end{bmatrix} \quad \mathbf{r}(2) = \mathbf{M}\mathbf{r}(1) = \begin{bmatrix} 1/3 \\ 5/12 \\ 3/12 \end{bmatrix}$$

Power Iteration Method: Example

$$\begin{cases} r_v = r_w/2 + r_x \\ r_w = r_v/2 + r_w/2 \\ r_x = r_v/2 \end{cases}$$

$$\underset{\mathbf{r}}{\begin{bmatrix} \ \\ \ \\ \ \end{bmatrix}} = \underset{\mathbf{M}}{\begin{bmatrix} 0 & 1/2 & 1 \\ 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \end{bmatrix}} \underset{\mathbf{r}}{\begin{bmatrix} \ \\ \ \\ \ \end{bmatrix}}$$

$$\underset{\mathbf{r}(0)}{\begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}} \quad \underset{\mathbf{r}(1) = \mathbf{Mr}(0)}{\begin{bmatrix} 3/6 \\ 1/3 \\ 1/6 \end{bmatrix}} \quad \underset{\mathbf{r}(2) = \mathbf{Mr}(1)}{\begin{bmatrix} 1/3 \\ 5/12 \\ 3/12 \end{bmatrix}} \quad \underset{\dots}{\dots} \quad \underset{\mathbf{r}(t+1) = \mathbf{Mr}(t)}{\begin{bmatrix} 6/15 \\ 6/15 \\ 3/15 \end{bmatrix}} \quad \begin{matrix} 2/5 \\ 2/5 \\ 1/5 \end{matrix}$$

Power Iteration Method: Example

$$\begin{cases} r_v = r_w/2 + r_x \\ r_w = r_v/2 + r_w/2 \\ r_x = r_v/2 \end{cases}$$

$$\mathbf{r} = \mathbf{M}\,\mathbf{r}$$

0	1/2	1
1/2	1/2	0
1/2	0	0

$\mathbf{r}(0)$	$\mathbf{r}(1) = \mathbf{M}\mathbf{r}(0)$	$\mathbf{r}(2) = \mathbf{M}\mathbf{r}(1)$	…	$\mathbf{r}(t+1) = \mathbf{M}\mathbf{r}(t)$	
1/3	3/6	1/3		6/15	2/5
1/3	1/3	5/12	…	6/15	2/5
1/3	1/6	3/12		3/15	1/5

We came up with the same set of solutions for r_v, r_w, and r_x without explicitly solving the system of equations

Take-Home Message of Today

- We present an example of **link analysis** algorithm: **PageRank**

Take-Home Message of Today

- We present an example of **link analysis** algorithm: **PageRank**
- <u>Goal:</u> Find an **importance score** for each web page

Take-Home Message of Today

- We present an example of **link analysis** algorithm: **PageRank**
- **Goal:** Find an **importance score** for each web page
- Represent the Web graph as a matrix M, where a link from page w to v is a **vote** from w to v

Take-Home Message of Today

- We present an example of **link analysis** algorithm: **PageRank**
- Goal: Find an **importance score** for each web page
- Represent the Web graph as a matrix M, where a link from page w to v is a **vote** from w to v
- 2 different yet equivalent approaches:
  - Linear Algebra → Matrix eigenvector
  - Probabilistic → ? (More on this next time…)