Seminars in Database

Prerequisites – 1st

F.Ricci

Content

- Linear Algebra
- Matrices
- Eigenvalues and eigenvectors
- Markov chains
- Google PageRank

Matrices

- A **Matrix** is a rectangular array of numbers

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

- a_{ij} is the **element** of matrix **A** in row i and column j
- **A** is said to be a n x m matrix if it has n rows and m columns
- A **square** matrix is a n x n matrix
- The **transpose** $\mathbf{A}^T$ of a matrix **A** is the matrix obtained by exchanging the rows and the columns

$$A^T = \begin{pmatrix} a_{11}^T & a_{12}^T \\ a_{21}^T & a_{22}^T \\ a_{31}^T & a_{32}^T \end{pmatrix} = \begin{pmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \\ a_{13}^T & a_{23}^T \end{pmatrix} = \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix}$$

Matrices

- A square matrix is **diagonal** iff has $a_{ij} = 0$ for all $i \neq j$

$$A = \begin{pmatrix} a_{11} & 0 \\ 0 & a_{22} \end{pmatrix}$$

- The **Identity** matrix **1** is the diagonal matrix with 1´s along the diagonal

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

- A symmetric matrix **A** satisfy the condition $\mathbf{A} = \mathbf{A}^T$

Vectors

- A vector **v** is a one-dimensional array of numbers (is an n x 1 matrix)
- Example:

$$v = \begin{pmatrix} 3 \\ 5 \\ 7 \end{pmatrix}$$

- The standard form of a vector is a column vector
- The transpose of a column vector $\mathbf{v}^{\mathbf{T}}$ =(3 5 7) is a row vector

Operation on matrices

- **Addition:** $\mathbf{A}=(a_{ij})$, $\mathbf{B}=(b_{ij})$, $\mathbf{C}=(c_{ij}) = \mathbf{A}+\mathbf{B}$
  - $c_{ij} = a_{ij} + b_{ij}$
- **Scalar multiplication:** λ is a number, $\lambda \mathbf{A} = (\lambda a_{ij})$
- **Multiplication:** if **A** and **B** are compatible, i.e., the number of *columns* of **A** is equal to the number of *rows* of **B**, then
  - $\mathbf{C}=(c_{ij})= \mathbf{AB}$
  - $c_{ij} = \Sigma_k \, a_{ik} \, b_{kj}$

Examples

$$AB = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix} = \begin{pmatrix} 1*1+2*2+3*3 & 1*4+2*5+3*6 \\ 4*1+5*2+6*3 & 4*4+5*5+6*6 \end{pmatrix} = \begin{pmatrix} 14 & 32 \\ 32 & 77 \end{pmatrix}$$

$$AB = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

- If **AB**=**1**, then **B** is said to be the **inverse** of **A** and is denoted with $\mathbf{A}^{-1}$
- If a matrix has an inverse is called **invertible** or **non singular**

Rank of a Matrix

- The row (column) **rank** of a matrix is the maximum number of rows (columns) that are linearly independent
- The vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ are **linearly independent** iff there is no linear combination $a_1\mathbf{v}_1 + \ldots + a_n\mathbf{v}_n$ (with coefficients a_i not all 0) of the vectors that is equal to 0
- *Example 1:* (1 2 3), (1 4 6), and (0 2 3) are linearly dependent: *show it*
- *Example 2:* (1 2 3) and (1 4 6) are not linearly dependent: *show it*
- The **kernel** of a matrix $\mathbf{A}$ is the subspace of vectors $\mathbf{v}$ such that $\mathbf{Av}=0$

Rank and Determinant

- **Theorem.** A n x n square matrix is nonsingular iff has full rank (i.e. n).
- **Theorem.** A matrix has full column rank iff it does not have a null vector
- **Theorem.** A n x n matrix **A** is singular iff the det(**A**)=0

$$\det(A) = \begin{cases} a_{11} & \text{if } \mathrm{n} = 1 \\ \sum_{j=1}^{n} (-1)^{1+j} a_{1j} \det(A_{[1j]}) & \text{if } n > 1 \end{cases}$$

- $A_{[ij]}$ is the ij minor, i.e., the matrix obtained by deleting the i-th row and the j-th column

Eigenvectors and Eigenvalues

- **Definition**. If **M** is a square matrix, **v** is a nonzero vector and λ is a number such that
  - $\mathbf{M}\ \mathbf{v} = \lambda\ \mathbf{v}$
- then **v** is said to be an *eigenvector* of **A** with *eigenvalue* λ
- If **v** is an eigenvector of **M** with eigenvalue λ, then so is any nonzero multiple of **v**
- Only the direction matters.

Example

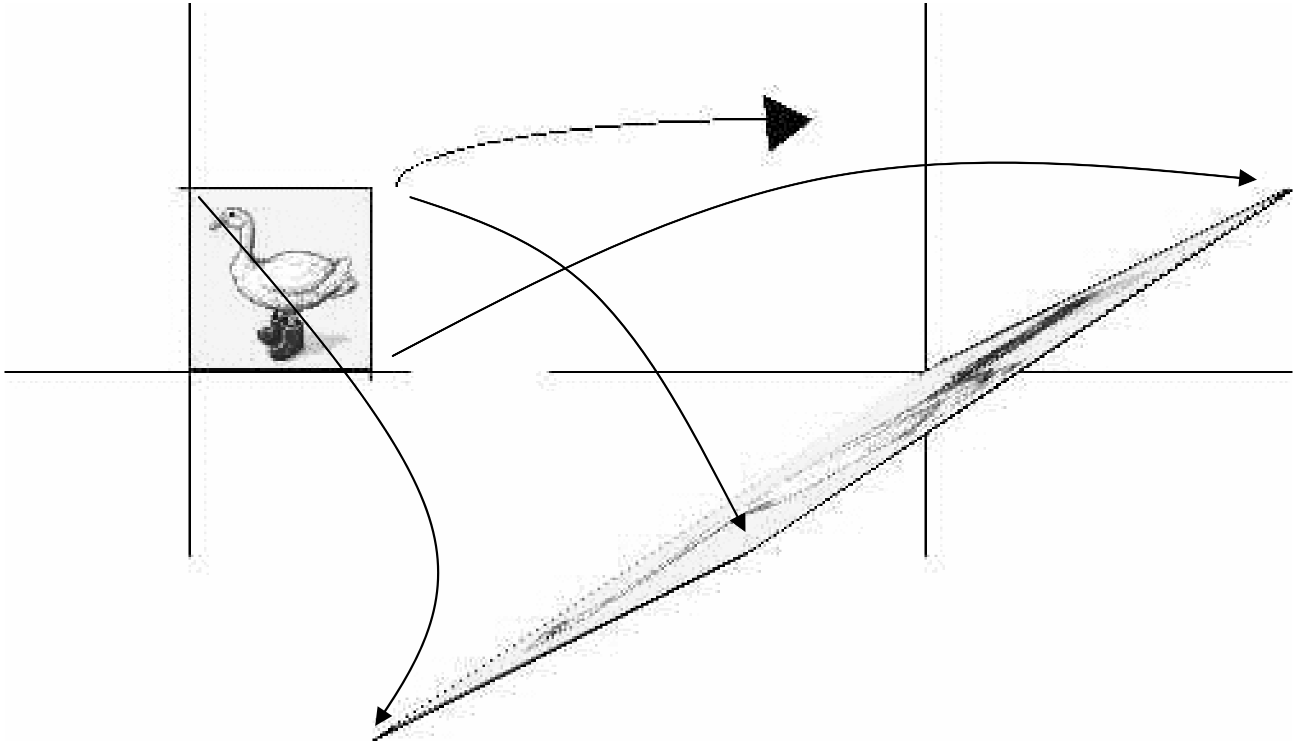
- The matrix

$$M = \begin{pmatrix} 2 & -3 \\ 1 & -2 \end{pmatrix}$$

- Has two eigenvectors:
  - $\mathbf{v_1}$ =(1 1)^t and $\mathbf{v_2}$ = (3 1)^t
- $\mathbf{Mv_1}$ = (-1 -1)^t = -1 $\mathbf{v_1}$
  - The eigenvalue is -1
- $\mathbf{Mv_2}$ = (3 1)^t = 1 $\mathbf{v_2}$
  - The eigenvalue is 1

Transformation

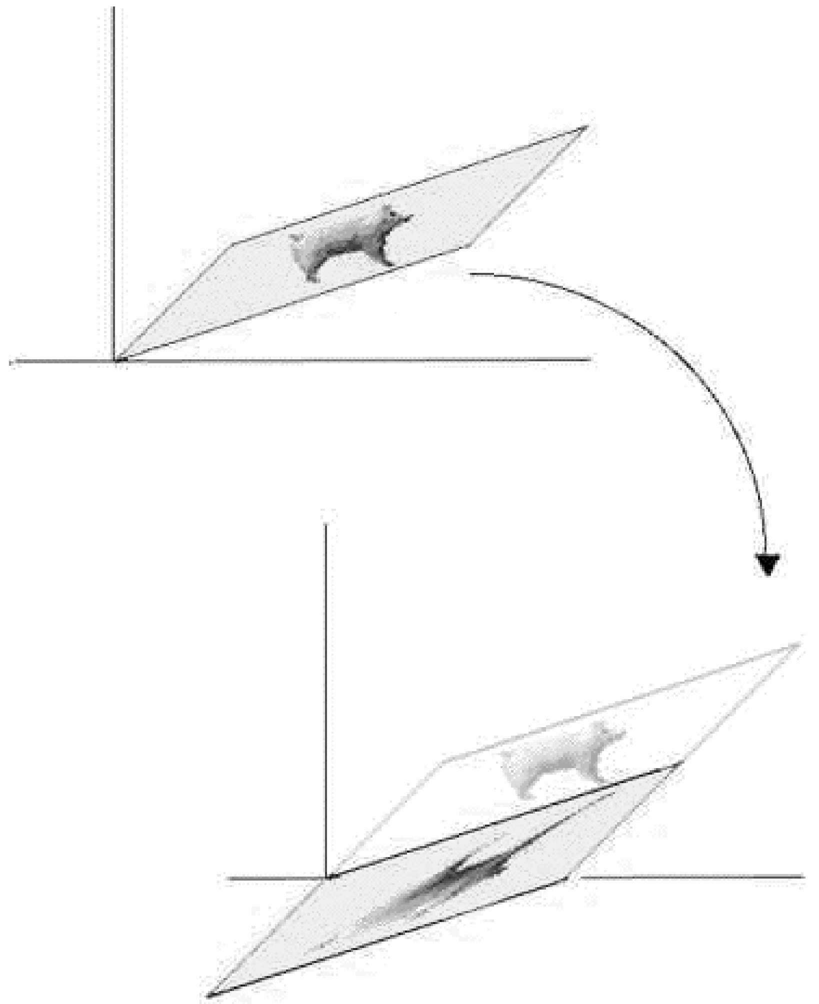
$$M = \begin{pmatrix} 2 & -3 \\ 1 & -2 \end{pmatrix}$$



- There is a lot of distortion in these directions (1 0) (1 1) (0 1)

Transformation along eigenvectors

- There are two independent directions which are not twisted at all by the matrix **M**: (1 1) and (3 1)
- one of them is flipped (1 1)
- We see less distortion if our box is oriented in the two special directions.



Results

- **Theorem:** every square matrix has at least one eigenvector
- The usual situation is that an n x n matrix has n linearly independent eigenvectors
- If there are n of them, they are a useful basis for R^n.
- Unfortunately, it can happen that there are fewer than n of them.

Finding Eigenvectors

- $\mathbf{M}\,\mathbf{v} = \lambda\,\mathbf{v}$
  - $\mathbf{v}$ is an eigenvector and is λ an eigenvalue
- If $\lambda = 0$, then finding eigenvectors is the same as finding nonzero vectors in the null space – iff $\det(\mathbf{M}) = 0$, i.e., the matrix is singular
- If λ != 0, then finding the eigenvectors is equivalent to finding the null space for the matrix $\mathbf{M} - \lambda\mathbf{1}$ ($\mathbf{1}$ is the identity matrix)
- The matrix $\mathbf{M} - \lambda\mathbf{1}$ has a non zero vector in the null space iff $\det(\mathbf{M} - \lambda\mathbf{1}) = 0$
- $\det(\mathbf{M} - \lambda\mathbf{1}) = 0$ is called the **characteristic equation**.

Example

$$M = \begin{pmatrix} 2 & -3 \\ 1 & -2 \end{pmatrix}$$

- det(**M** – λ**1**) = 0
  - (2 - λ)(-2 - λ) + 3 = $\lambda^2 - 1$
- The solutions are +1 and -1
- Now we have to solve the set of linear equations
  - **Mv**^{t}=**v**^{t} (for the first eigenvalue)

$$2x - 3y = x$$

$$x - 2y = y$$

  - Has solution x=3y, (3 1)^t – and all vectors obtained multiplying this with a scalar.

Algorithm

- *To find the eigenvalues and eigenvectors of* **M**:
  - *First find the eigenvalues by solving the characteristic equation*.
    Call the solutions $\lambda_1,\ldots, \lambda_n$. (There is always at least one eigenvalue, and there are at most n of them.)
  - *For each eigenvalue* λ_k*, use row reduction to find a basis of the eigenspace* Ker($\mathbf{M} - \lambda_k \mathbf{1}$)
  - For all λ_k, the existence of a nonzero vector in this null space is guaranteed. Any such vector is an eigenvector.

Diagonalization

- **Basic Modeling Theorem.** *If* $\mathbf{e}_k$ *denotes the unit vector with k-th component* 1, *all other components being* 0, *and if* **A** *and* $\mathbf{e}_k$ *are of compatible sizes so that* $\mathbf{A}\,\mathbf{e}_k$ *makes sense, then* $\mathbf{A}\,\mathbf{e}_k$ *is the k-th column of* **A**.
- Suppose that **A** is an n x n matrix with n linearly independent eigenvectors $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$
- Let be **U** the matrix that has $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ as its columns
- According to the modeling theorem, the matrix **U** converts the standard basis $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ into a basis consisting of the eigenvectors $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$
- **U** must have an inverse $\mathbf{U}^{-1}$, which converts the eigenvectors back into the standard basis.

Diagonalization

- $\mathbf{D} := \mathbf{U}^{-1}\mathbf{A}\mathbf{U}$
- The first column of D is
  - $\mathbf{D}\ \mathbf{e}_1 = \mathbf{U}^{-1}\mathbf{A}\mathbf{U}\ \mathbf{e}_1 = \mathbf{U}^{-1}\mathbf{A}\ \mathbf{v}_1 = \mathbf{U}^{-1}\lambda_1\mathbf{v}_1 = \lambda_1\ \mathbf{e}_1$
- D is a diagonal matrix with the eigenvalues in the diagonal

$$\begin{pmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & \lambda_n \end{pmatrix}$$

- $\mathbf{A} = \mathbf{U}\ \mathbf{D}\ \mathbf{U}^{-1}$, i.e., A is diagonalized

Diagonalization

- In fact if A is non singular then

$$A = \sum_{i=1}^{k} \lambda_i r_i l_i^T$$

- Where r_i are the right eigenvectors and l_i are the left eigenvectors ($\lambda_i l_i = A^T l_i$)
- Attention: $r_i l_i^T$ is a n x n matrix:

$$\begin{pmatrix} r_{i1} \\ \vdots \\ r_{in} \end{pmatrix} \begin{pmatrix} l_{i1} & \dots & l_{in} \end{pmatrix} = \begin{pmatrix} r_{i1} l_{i1} & \cdots & r_{i1} l_{in} \\ \vdots & & \vdots \\ r_{in} l_{i1} & \cdots & r_{in} l_{in} \end{pmatrix}$$

- The idea is that a matrix is decomposed in the sum of other matrices, and the first addend already provides an approximation of the matrix (corresponding to the largest eigenvalue)

Test for diagonalization

- *An n by n matrix* **A** *can be diagonalized if any of the following conditions can be verified:*
  - *if and only if the sum of the dimensions of its eigenspaces is equal to n*
  - if it has *n* distinct eigenvalues
  - *if* **A** *is "symmetric", that is,* $\mathbf{A}^T = \mathbf{A}$; *or*
  - *if* **A** *is "normal", that is,* $\mathbf{A}\,\mathbf{A}^T = \mathbf{A}^T\,\mathbf{A}$
- Where $\mathbf{A}^T$ is the transpose of **A** ($a_{ij} = a^T_{ji}$)

Graphs

- A **directed graphs G** is a pair (V,E), where V is a finite set and E is a binary relations on V
  - V is the **Vertex set** of G: contains the **vertices**
  - E is the **Edge set** of G: contains the **edges**
- In an undirected graphs G=(V,E) the edges consists of unordered pairs of vertices
- The **in-degree** of a vertex v (directed graph) is the number of edges entering in v
- The **out-degree** of a vertex v (directed graph) is the number of edges leaving v.

Markov chains

Ranking web pages

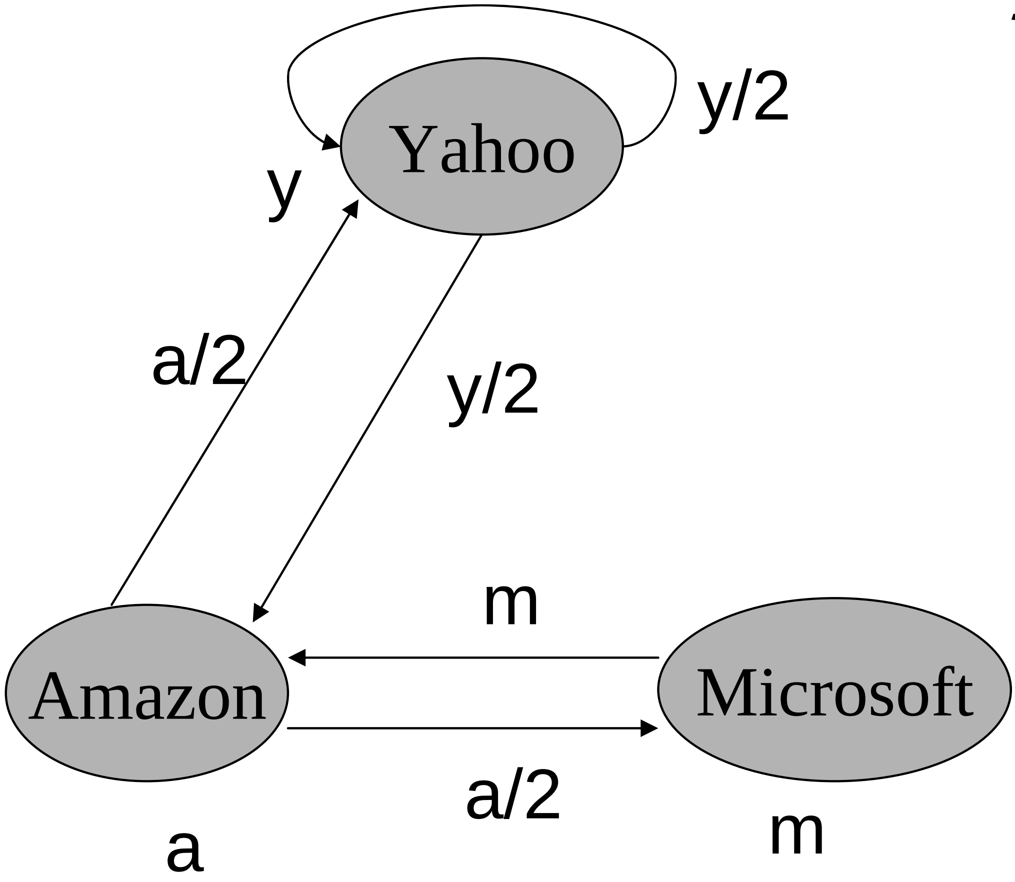
- To count inlinks: `http://siteexplorer.search.yahoo.com`
- Web pages are not equally “important”
  - www.unibz.it vs. www.stanford.edu
  - Inlinks as votes
    - www.stanford.edu has 686,387 inlinks
    - www.unibz.it has 3,903 inlink
- Are all inlinks equal?
  - Recursive question!

Simple recursive formulation

- Each link's vote is proportional to the importance of its source page
- If page **P** with importance **x** has **n** outlinks, each link gets **x / n** votes

Simple "flow" model

The web in 1839



$$y = y/2 + a/2$$
$$a = y/2 + m$$
$$m = a/2$$

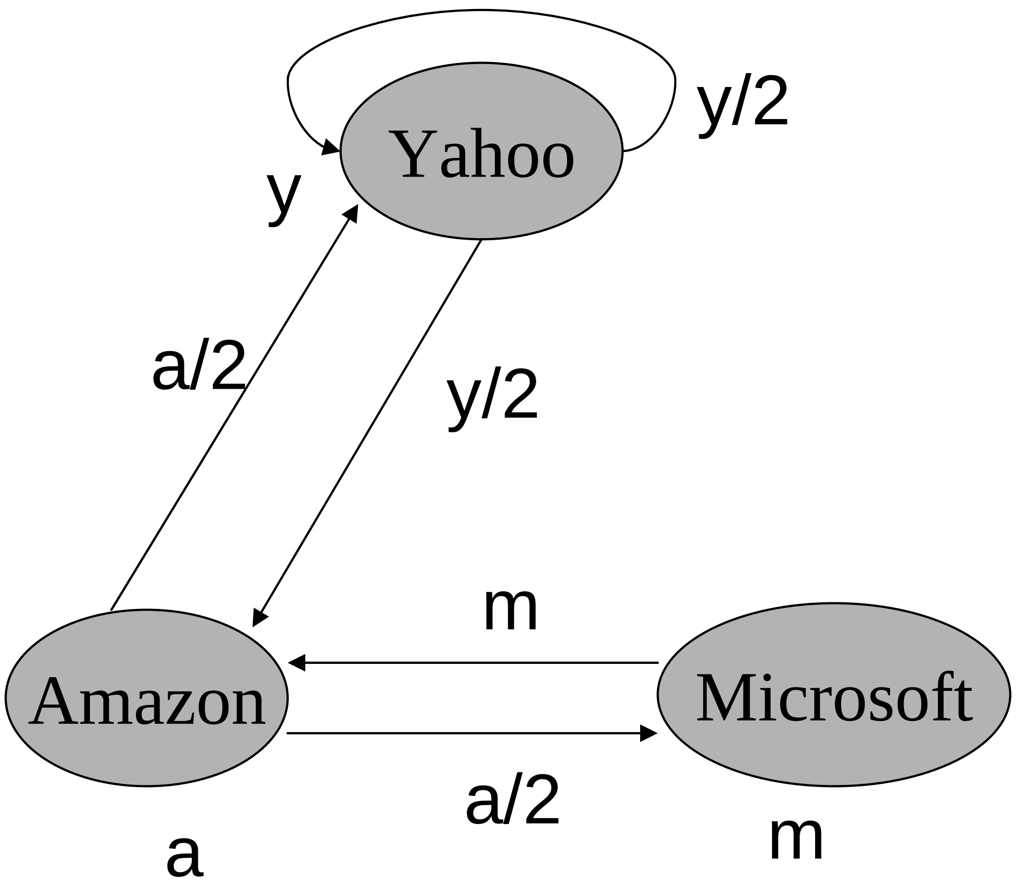
Solving the flow equations

- 3 equations, 3 unknowns, no constants
  - No unique solution
  - All solutions equivalent modulo scale factor
- Additional constraint forces uniqueness
  - $y+a+m = 1$
  - $y = 2/5$, $a = 2/5$, $m = 1/5$
- Gaussian elimination method works for small examples, but we need a better method for large graphs.

Matrix formulation

- Matrix **M** has one row and one column for each web page
- Suppose page i has n outlinks
  - If *i links to j*, then $M_{ij}=1/n$
  - Else $M_{ij}=0$
- **M** is a **row stochastic matrix**
  - Rows sum to 1
- Suppose **r** is a vector with one entry per web page
  - r_i is the importance score of page i
  - Call it the **rank vector**

Example

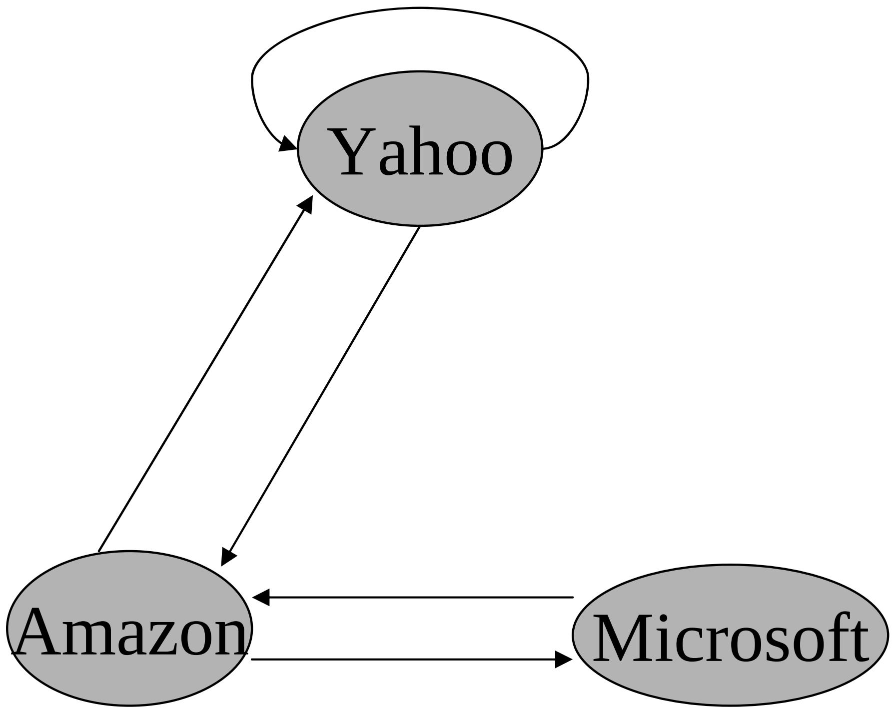


$$\begin{array}{c} \\ y \\ a \\ m \end{array} \begin{array}{c} \begin{array}{ccc} y & a & m \end{array} \\ \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 1 & 0 \end{pmatrix} \end{array}$$

(y a m) = (y a m)**M**

$$y = y/2 + a/2$$
$$a = y/2 + m$$
$$m = a/2$$

Power Iteration Example



$$
\begin{array}{c|ccc|}
 & y & a & m \\
\hline
y & \frac{1}{2} & \frac{1}{2} & 0 \\
a & \frac{1}{2} & 0 & \frac{1}{2} \\
m & 0 & 1 & 0
\end{array}
$$

$$
\begin{array}{c}
y \\ a \\ m
\end{array}
=
\begin{array}{c}
1/3 \\ 1/3 \\ 1/3
\end{array}
\quad
\begin{array}{c}
1/3 \\ 1/2 \\ 1/6
\end{array}
\quad
\begin{array}{c}
5/12 \\ 1/3 \\ 1/4
\end{array}
\quad
\begin{array}{c}
3/8 \\ 11/24 \\ 1/6
\end{array}
\quad \ldots \quad
\begin{array}{c}
2/5 \\ 2/5 \\ 1/5
\end{array}
$$

(y a m) = (y a m)**M** **M** **M** **M** **...** **M**

Example

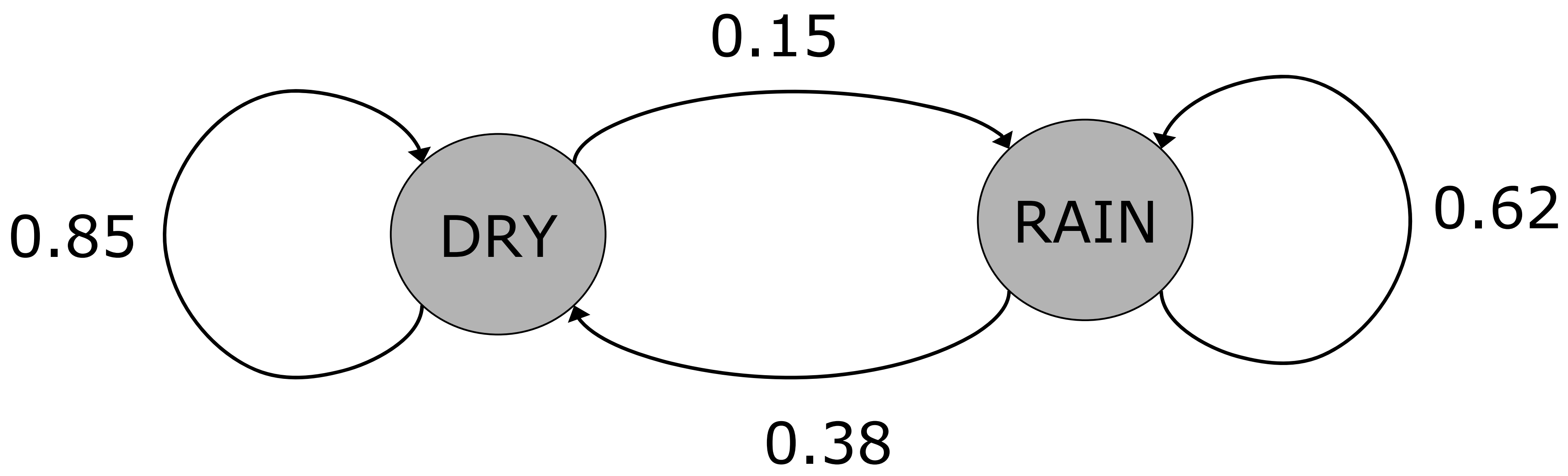
Following 6 hours

First 6 hours	dry	rain	
dry	61	11	72 cases
rain	11	17	28 cases
	72 cases	28 cases	

Following 6 hours

First 6 hours	dry	rain	
dry	.85	.15	72 cases
rain	.38	.62	28 cases
	72 cases	28 cases	

States and probabilities



Composing transitions

$$\begin{array}{ccc} & \otimes & \mathbf{R} \\ \otimes & \multicolumn{2}{c}{\multirow{2}{*}{$\begin{pmatrix} .85 & .15 \\ .38 & .62 \end{pmatrix}$}} \\ \mathbf{R} & & \end{array}$$

$$\begin{array}{c|cc} & \otimes & \mathbf{R} \\ \hline \otimes & .85 & .15 \\ \mathbf{R} & .38 & .62 \end{array}$$

What can we say about the 12 h transition? From 00-06 to 12-18?

Rain > > >		Rain > > >		Rain	
	.62		.62		= .38
Rain > > >		dry > > >		Rain	
	.38		.15		= .06
					= .44

0.44 = 0.38*0.15+0.62*0.62
What kind of operation is on the matrix?

Composing transitions

$$\begin{pmatrix} .85 & .15 \\ .38 & .62 \end{pmatrix}\begin{pmatrix} .85 & .15 \\ .38 & .62 \end{pmatrix} = \begin{pmatrix} .78 & .22 \\ .56 & .44 \end{pmatrix} = A^2$$

- The probabilities of the 12hours transitions are given by squaring the matrix representing the probabilities of the 6hours transitions
  - P(rain-in-12hours|rain-now)= P(rain-in-12hours|rain-in-6hours)*P(rain-in-6hours|rain-now)+P(rain-in-12hours|dry-in-6hours)*P(dry-in-6hours|rain-now)=.62*.62+.15*.38=.44
  - P(dry-in-12hours|rain-now)= P(dry-in-12hours|rain-in-6hours)*P(rain-in-6hours|rain-now)+P(dry-in-12hours|dry-in-6hours)*P(dry-in-6hours|rain-now)=.38*.62+.85*.38=.56

Behavior in the limit

$$A = \begin{pmatrix} .85 & .15 \\ .38 & .62 \end{pmatrix} \quad A^2 = \begin{pmatrix} .78 & .22 \\ .56 & .44 \end{pmatrix} \quad A^3 = \begin{pmatrix} .75 & .25 \\ .64 & .36 \end{pmatrix}$$

$$A^4 = \begin{pmatrix} .73 & .27 \\ .68 & .32 \end{pmatrix} \quad A^5 = \begin{pmatrix} .72 & .28 \\ .70 & .30 \end{pmatrix} \quad A^6 = \begin{pmatrix} .72 & .28 \\ .71 & .29 \end{pmatrix}$$

$$A^7 = \begin{pmatrix} .72 & .28 \\ .71 & .29 \end{pmatrix} \quad A^8 = \begin{pmatrix} .72 & .28 \\ .72 & .28 \end{pmatrix} \quad A^9 = \begin{pmatrix} .72 & .28 \\ .72 & .28 \end{pmatrix}$$

$$A^\infty = \begin{pmatrix} .72 & .28 \\ .72 & .28 \end{pmatrix}$$

Behavior in the limit

- If a,b <=1, and a+b=1, i.e., (a b) is a generic state with a certain probability a to be dry and b=1-a to be rain, then

$$\begin{pmatrix} a & b \end{pmatrix} A^{\infty} = \begin{pmatrix} a & b \end{pmatrix} \begin{pmatrix} .72 & .28 \\ .72 & .28 \end{pmatrix} = \begin{pmatrix} .72 & .28 \end{pmatrix}$$

- In particular (.72 .28)A=(.72 .28), i.e., it is an **eigenvector** with **eigenvalue** 1
- The eigenvector (.72 .28) represents the limit situation starting from a generic situation (a b): it is called the **stationary distribution**.

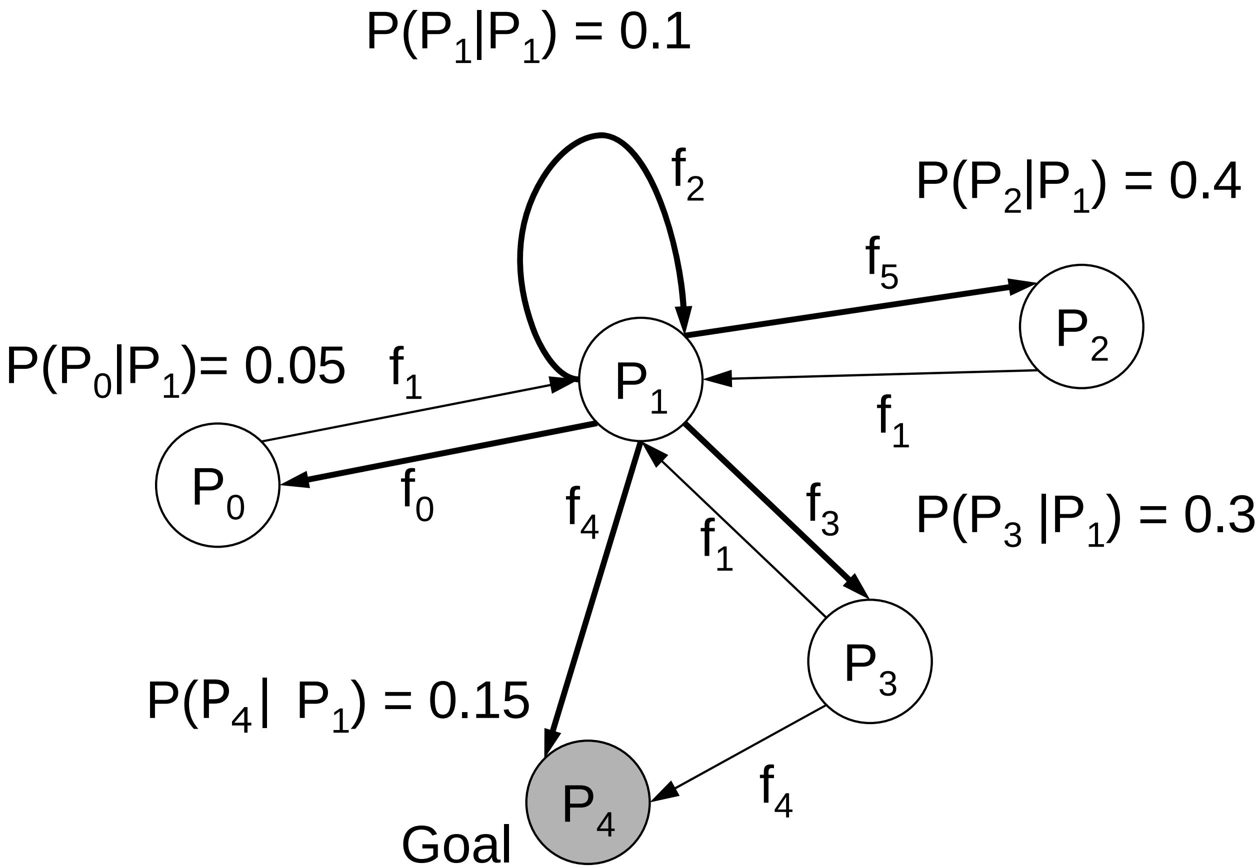
Markov Chain

- A Markov chain is a sequence X_1, X_2, X_3, ... of random variables with the property:
- **Markov property**: the conditional probability distribution of the next future state X_{n+1} given the present and past states is a function of the present state X_n alone

$$\Pr(X_{n+1} = x | X_0 = x_0, X_1 = x_1, \ldots, X_n = x_n) = \Pr(X_{n+1} = x | X_n = x_n).$$

- If the state space is finite then the transition probabilities can be described with a matrix $P_{ij}=P(X_{n+1}= j \mid X_n = i)$, i,j =1, …m
- **Example:**
  - X_t is the page visited by a user at time t; at every time t the user can be in one among m pages
  - We assume that when a user is on page i at time t, then the probability to be in page j at time t+1 depends only on page i, and not on the pages previously visited
  - **Problem:** what is the probability that a user will be (in the limit) in some particular page?

Probabilities



In this example the probability to pass from a page to another is not constant

We'll se later that in Google this is assumed to be constant (1/#of outgoing links from a page)

Examples

- $P_{ij}=P(X_{n+1}= j \mid X_n = i)$, i,j =1, …m
- $(1, 0, 0, \ldots, 0)\, P = (P_{11}, P_{12}, P_{13}, \ldots, P_{1n})$
  - if at time n we are in state 1, then at time n+1 we are in state j with probability P_{1j}, i.e., the first row of P_{ij} gives the probabilities to be in the other states
- $(0.5, 0.5, 0, \ldots, 0)\, P = (P_{11}\cdot 0.5 + P_{21}\cdot 0.5, \ldots, P_{1n}\cdot 0.5 + P_{2n}\cdot 0.5)$
- In the previous example
  - $P(X_{n+1} = 1) = P(X_{n+1} = 1 \mid X_n = 1)P(X_n = 1) + P(X_{n+1} = 1 \mid X_n = 2)P(X_n = 2)$

Stationary distribution

- The **stationary distribution** is a m-dimensional (sum 1) vector which satisfies the equation

$$\pi^T \mathbf{P} = \pi^T,$$

- Where π is a (column) vector and π^T (row vector) is the transpose of π
- If the stationary state exists (it does exists if P is invertible) then

$$\pi^T = \lim_n x^T P^n$$

- Where x is a generic distribution over the m states (i.e., it is an m-dimensional vector whose entries are <=1 and the sum is 1)

Random Walk Interpretation

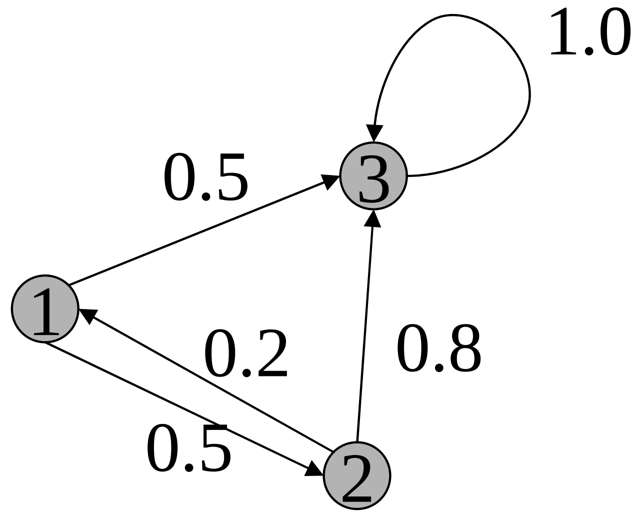
- Imagine a **random web surfer**
  - At any time t, surfer is on some page P
  - At time t+1, the surfer follows an outlink from P uniformly at random
  - Ends up on some page Q linked from P
  - Process repeats indefinitely
- Let **p**(t) be a vector whose i^{th} component is the probability that the surfer is at page i at time t
  - **p**(t) is a probability distribution on pages

The stationary distribution

- Where is the surfer at time t+1?
  - Follows a link uniformly at random
  - $\mathbf{p}(t+1) = \mathbf{p}(t)\mathbf{M}$
- Suppose the random walk reaches a state such that $\mathbf{p}(t+1) = \mathbf{p}(t)\mathbf{M} = \mathbf{p}(t)$
  - Then $\mathbf{p}(t)$ is called a **stationary distribution** for the random walk
- Our rank vector $\mathbf{r}$ satisfies $\mathbf{r} = \mathbf{r}\mathbf{M}$
  - So it is a stationary distribution for the random surfer

Example

$$P = \begin{pmatrix} 0 & 0.5 & 0.5 \\ 0.2 & 0 & 0.8 \\ 0 & 0 & 1 \end{pmatrix}$$



- It is easy to show that the steady state (left eigenvector) is $\pi^T = (0\ 0\ 1)$, $\pi^T P = \pi^T$, i.e., is the state 3
- The user will always reach the state 3 and will stay there (*spider trap*)

Random teleports

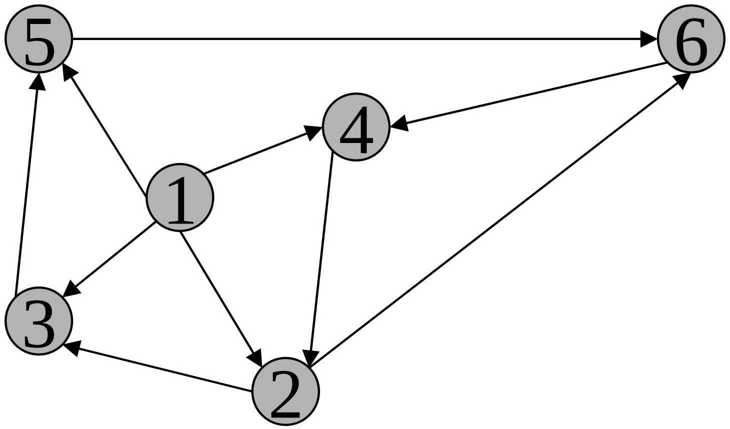
- The Google solution for spider traps
- At each time step, the random surfer has two options:
  - With probability β, follow a link at random
  - With probability $1-\beta$, jump to some page uniformly at random
  - Common values for β are in the range 0.8 to 0.9
- Surfer will teleport out of spider trap within a few time steps

Matrix formulation

- Suppose there are N pages
  - Consider a page i, with set of outlinks O(i)
  - We have
    - $M_{ij} = 1/|O(i)|$ when *i links j*
    - and $M_{ij} = 0$ otherwise
  - The random teleport is equivalent to
    - adding a **teleport link** from i to every other page with probability $(1-\beta)/N$
    - reducing the probability of following each outlink from $1/|O(i)|$ to $\beta/|O(i)|$
    - Equivalent: tax each page a fraction $(1-\beta)$ of its score and redistribute evenly.

Example

- Simple example with 6 pages



- P(5|1)=P(4|1)=P(3|1)=P(2|1)= $\beta/4 + (1-\beta)/6$
- P(1|1)=P(6|1)= $(1-\beta)/6$
- P(*|1) = $4[\beta/4 + (1-\beta)/6] + 2(1-\beta)/6 = 1$

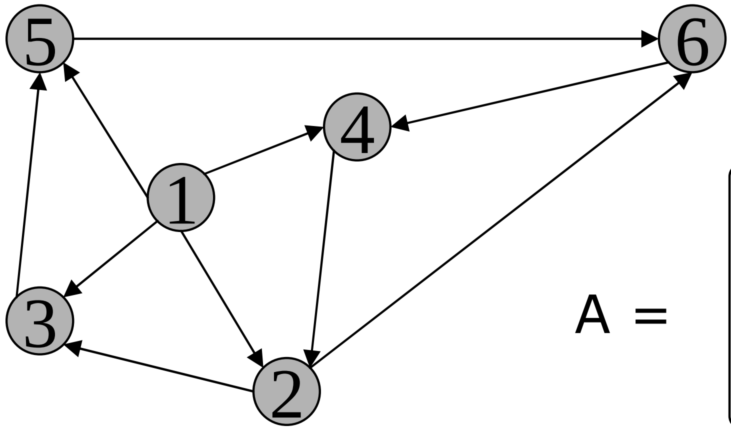
Google Page Rank

- Construct the NxN matrix **A** as follows
  - $A_{ij} = \beta M_{ij} + (1-\beta)/N$
- Verify that **A** is a stochastic matrix
- The **page rank vector r** is the principal eigenvector of this matrix
  - satisfying **r** = **rA**
  - The score of each page r_i is

$$r_i = \beta \left(\sum_{k \in I(i)} \frac{r_k}{|O(k)|} \right) + \frac{(1-\beta)}{N}$$

- Equivalently, **r** is the stationary distribution of the random walk with teleports.

Example



$$A = \begin{pmatrix} 0,03 & 0,24 & 0,24 & 0,24 & 0,24 & 0,03 \\ 0,03 & 0,03 & 0,45 & 0,03 & 0,03 & 0,45 \\ 0,03 & 0,03 & 0,03 & 0,03 & 0,88 & 0,03 \\ 0,03 & 0,88 & 0,03 & 0,03 & 0,03 & 0,03 \\ 0,03 & 0,03 & 0,03 & 0,03 & 0,03 & 0,88 \\ 0,03 & 0,03 & 0,03 & 0,88 & 0,03 & 0,03 \end{pmatrix}$$

$P(4|1)=0.24=0.85/4 + 0.15/6$

$P(6|1)=0.03=0.15/6$

$P(4|6)=0.88=0.85/1 + 0.15/6$

$$A^{30} = \begin{pmatrix} 0,03 & 0,23 & 0,13 & 0,24 & 0,14 & 0,24 \\ 0,03 & 0,23 & 0,13 & 0,24 & 0,14 & 0,24 \\ 0,03 & 0,23 & 0,13 & 0,24 & 0,14 & 0,24 \\ 0,03 & 0,23 & 0,13 & 0,24 & 0,14 & 0,24 \\ 0,03 & 0,23 & 0,13 & 0,24 & 0,14 & 0,24 \\ 0,03 & 0,23 & 0,13 & 0,24 & 0,14 & 0,24 \end{pmatrix}$$

Stationary distribution = (0.03 0.23 0.13 0.24 0.14 0.24)

Dead ends

- Pages with no outlinks are "**dead ends**" for the random surfer
  - Nowhere to go on next step
- When there are dead ends the matrix is no longer stochastic (the sum of the row elements is not 1)

Dealing with dead-ends

- **Teleport**
  - Follow random teleport links with probability 1.0 from dead-ends (i.e., for that pages set $\beta = 0$)
  - Adjust matrix accordingly
- **Prune and propagate**
  - Preprocess the graph to eliminate dead-ends
  - Might require multiple passes (why?)
  - Compute page rank on reduced graph
  - Approximate values for dead ends by propagating values from reduced graph

Computing page rank

- Key step is matrix-vector multiply
  - $\mathbf{r}^{new} = \mathbf{r}^{old}\mathbf{A}$
- Easy if we have enough main memory to hold $\mathbf{A}$, $\mathbf{r}^{old}$, $\mathbf{r}^{new}$
- Say N = 1 billion pages
  - We need 4 bytes for each entry (say)
  - 2 billion entries for vectors $\mathbf{r}^{new}$ and $\mathbf{r}^{old}$, approx 8GB
  - Matrix A has N^2 entries
    - 10^{18} is a large number!

Sparse matrix formulation

- Although **A** is a dense matrix, it is obtained from a sparse matrix **M**
  - 10 links per node, approx 10N entries
- We can restate the page rank equation
  - $\mathbf{r} = \beta\mathbf{r}\mathbf{M} + [(1-\beta)/N]_N$
  - $[(1-\beta)/N]_N$ is an N-vector with all entries $(1-\beta)/N$
- So in each iteration, we need to:
  - Compute $\mathbf{r}^{new} = \beta\mathbf{r}^{old}\mathbf{M}$
  - Add a constant value $(1-\beta)/N$ to each entry in $\mathbf{r}^{new}$

Sparse matrix encoding

- Encode sparse matrix using only nonzero entries
  - Space proportional roughly to number of links
  - say 10N, or 4*10*1 billion = 40GB
  - still won’t fit in memory, but will fit on disk

source node	degree	destination nodes
0	3	1, 5, 7
1	5	17, 64, 113, 117, 245
2	2	13, 23

Basic Algorithm

- Assume we have enough RAM to fit $\mathbf{r}^{new}$, plus some working memory
  - Store $\mathbf{r}^{old}$ and matrix $\mathbf{M}$ on disk

Basic Algorithm:

- Initialize: $\mathbf{r}^{old} = [1/N]_N$
- Iterate:
  - **Update:** Perform a sequential scan of $\mathbf{M}$ and $\mathbf{r}^{old}$ and update $\mathbf{r}^{new}$
  - Write out $\mathbf{r}^{new}$ to disk as $\mathbf{r}^{old}$ for next iteration
  - Every few iterations, compute $|\mathbf{r}^{new}-\mathbf{r}^{old}|$ and stop if it is below threshold
    - Need to read in both vectors into memory

Update step

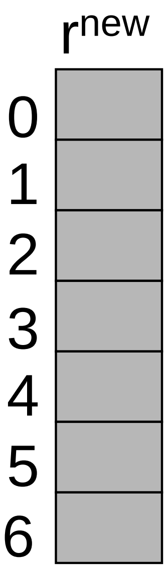
Initialize all entries of $\mathbf{r}^{new}$ to $(1-\beta)/N$

For each page p (out-degree n):

    Read into memory: p, n, $dest_1, \ldots, dest_n$, $r^{old}(p)$

    for j = 1..n:

        $r^{new}(dest_j)$ += $\beta * r^{old}(p)/n$



src	degree	destination
0	3	1, 5, 6
1	4	17, 64, 113, 117
2	2	13, 23