PageRank and Markov Chains

lecture-6

Course on Graph Neural Networks (Winter Term 21/22)

Prof. Dr. Holger Giese (holger.giese@hpi.uni-potsdam.de)

Christian Medeiros Adriano (christian.adriano@hpi.de) - **"Chris"**

Matthias Barkowski (matthias.barkowski@hpi.de)

Breaking news on Learning on Networks

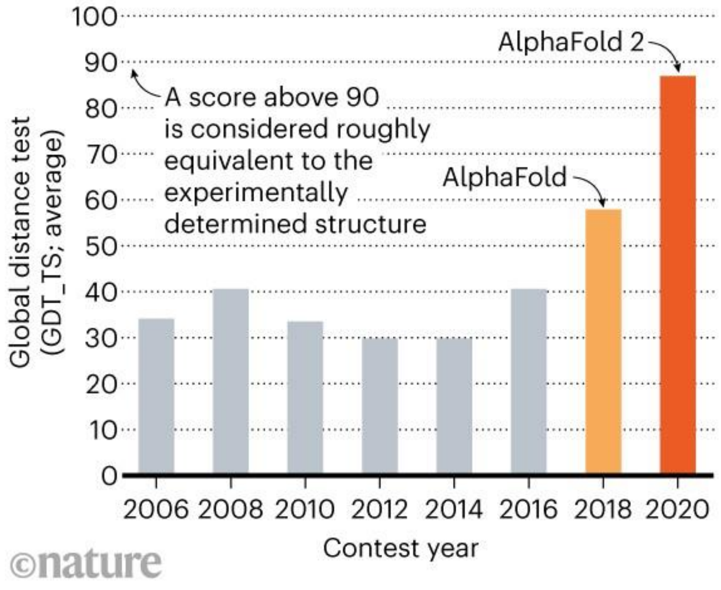
Goal: predict the protein’s constituent parts = a string of different amino acids and map out the many twists and folds of its eventual shape. Poor predictions in 1980s and 1990s.

AphaFold-1 (2018):
1- apply deep learning to structural and genetic data to predict the distance between pairs of amino acids in a protein
2- use this information to build ‘consensus’ model of how the protein should look like

[Nature 2020] It will change everything: DeepMind’s AI makes gigantic leap in solving protein structures - Google’s deep-learning for determining the 3D shapes of proteins https://www.nature.com/articles/d41586-020-03348-4

STRUCTURE SOLVER

DeepMind’s AlphaFold 2 algorithm significantly outperformed other teams at the CASP14 protein-folding contest — and its previous version’s performance at the last CASP.



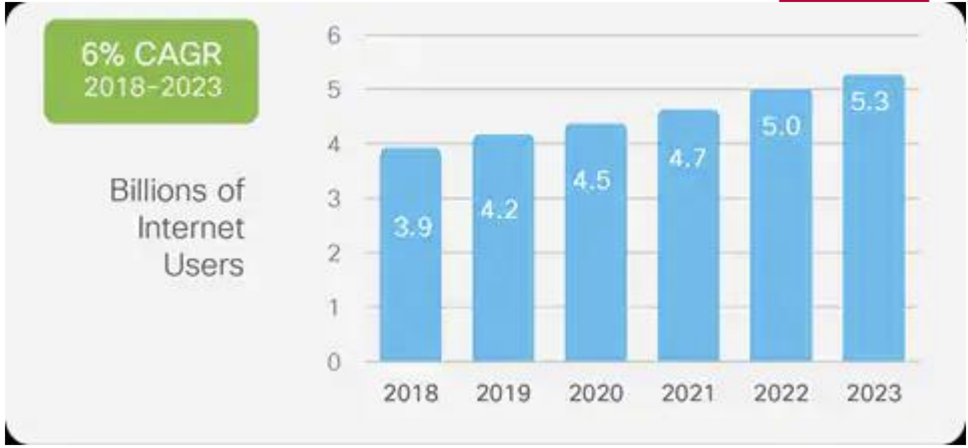
Can a Computer Devise a Theory of Everything?*

- The Feynman Lectures on Physics **->** used them to generate data that was then fed to a neural network. DL was able to recover all 100 formulas
- Data from the Large Hadron Collider **->** the system successfully identified and distinguished between quarks and gluons, without ever knowing what either was.

NY Times - https://www.nytimes.com/2020/11/23/science/artificial-intelligence-ai-physics-theory.html

*It might be possible, physicists say, but not anytime soon. And there’s no guarantee that we humans will understand the result…

Motivation [Cisco Annual Internet report 2020]



Nearly 2/3 of the global population will have Internet access by 2023, i.e., 5.3 billion total Internet users (66% of global population) and up from 3.9 billion (51% of global population) in 2018.

2023:

- 29.3 billion networked devices (compared with 18.4 billion in 2018)
- 14.7 billion IoT connections (33% growth over 2-18)

- Connected **home** apps will have the largest share and connected **cars** will be the fastest growing application type.
- Connected home apps will have nearly half or 48% of IoT share by 2023 and Connected car applications will grow the fastest at 30% over the forecast period (2018–2023).

How to make predictions on networks of hundreds of billions of nodes which are constantly evolving?

1. We need to sample effectively = node ranking
2. This allows better search and more efficient monitoring

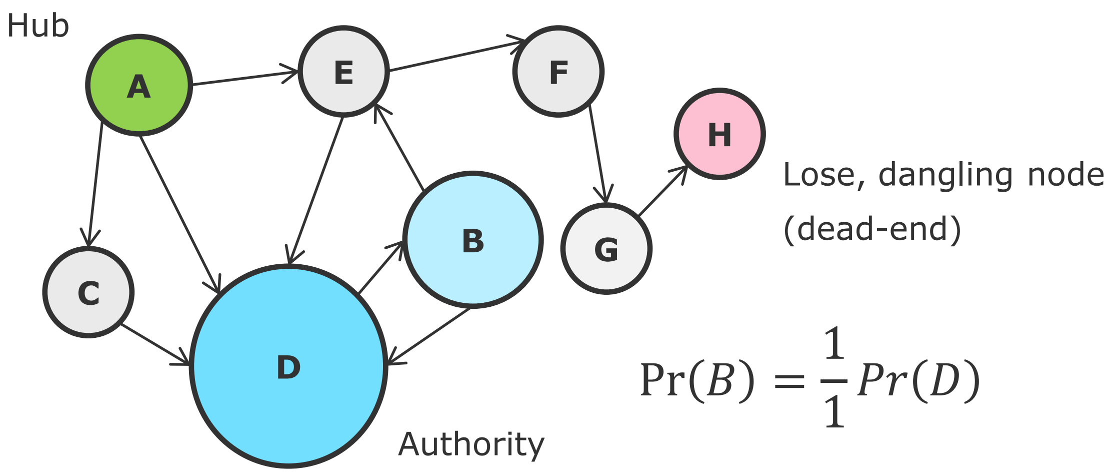
Google **PageRank** takes few weeks for the crawlers to map of the entire web and a few hours to recompute the importance of pages.

[Cisco 2020] Cisco Annual Internet report , https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

What we want

If I spend time randomly surfing the web, assuming equal time between clicks, what percent of time would I be on each page? i.e., how important is each node?

$$\Pr(E) = \frac{1}{3}Pr(A) + \frac{1}{2}Pr(B)$$



$$\Pr(B) = \frac{1}{1}Pr(D)$$

$$\Pr(D) = \frac{1}{3}Pr(A) + \frac{1}{2}Pr\ (\mathrm{E}) + \frac{1}{2}Pr(B) + Pr(C)$$

Caveat: this does not scale well for large graphs
Solution: we need to sample!

<u>Importance of a node (*Pr*):</u>

- In-links are less prone to be manipulated
- Importance is proportional to its neighbors
  - Flow/Message passing way to computing

$\Pr(i) = \sum_{j \in Nr(i)} \frac{1}{D_j} Pr(j)$, where D_i is the out-degree of node i

Add one more constraint $1 = \sum_{i \in N} \Pr(i)$

Solve the system of equations:

$$\begin{bmatrix} \Pr(A) = \dots \\ \Pr(B) = \dots \\ \dots \\ \Pr(H) = \dots \\ 1 = \sum_{i \in N} \Pr(i) \end{bmatrix}$$

Sampling on Networks = Random Walking

Goal: model a random process in which the system transitions from one state to another at discrete time steps.

- States are nodes
- Transitions are edges

At each time,

- there are N states the system could be in.
- we model the system as a vector $\overrightarrow{R_t} \in \mathbb{R}^n$,

$\overrightarrow{\boldsymbol{R_t}}$ represents the probability of being at any given state.

- where, $t = 0,1,2, \dots, T$, where the "initial state" is the vector $\overrightarrow{R_0}$.
- the total probability of being at any given state should be 1, i.e.,$1 = \sum_{i \in N} R_{t_i}$

A sequence of probability vectors $\overrightarrow{R_0}, \overrightarrow{R_1}, \dots, \overrightarrow{R_t}$ is called a **Markov Chain**

Intro to Discrete Time Markov Chains

Discrete Time Markov Chain

- **States**
  - □ Events: infection, failures, rumor spread, toxic contamination, traffic accident, etc.
  - □ Entities: components, people, infrastructure, that have **states** and are affected by **events**
  - □ State (S):
    - – operational (no failure/not infected): So
    - – degraded (performance): Sd
    - – unresponsive (disabled): Su
  - □ Transition (T): change from one state to another state (self-loops included)
- **State Traces**
  - □ A sequence of states that happened within a given time horizon:
    - – $ST_1 = So_1; So_2; So_3; Sd_3; So_3; So_1; Sd_1; Sd_2; Su_3; So_1; So_2; So_3$
  - □ Obtained from system logs, contact tracing (GPS), sensors (traffic, water pipes), etc.

Assumptions

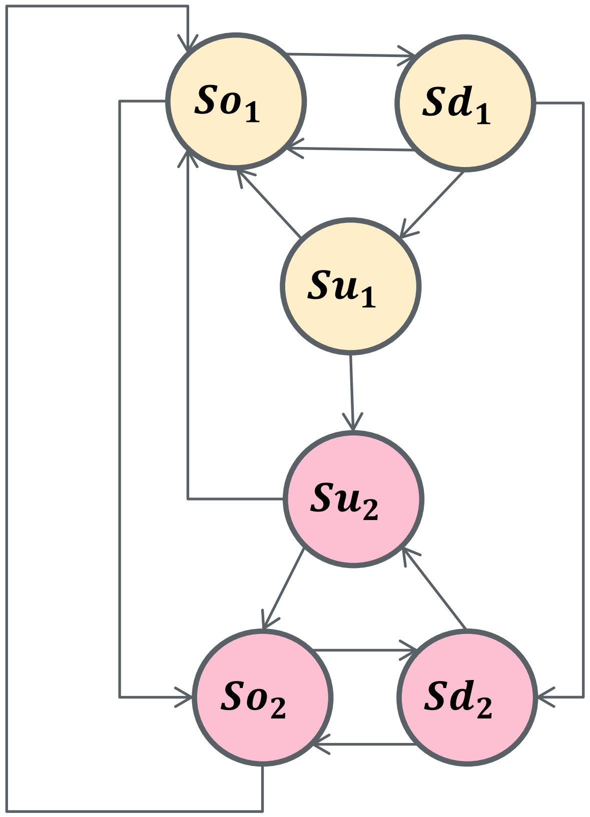
1. Markov property
   - □ $(S_{t+1} \perp S_{t-1})|\, S_t$ or
   - □ $\mathrm{P}(\mathrm{S}_{t+1}, \mathrm{S}_{t-1}) = \mathrm{P}(\mathrm{S}_{t+1}|S_t)\, P(\mathrm{S}_{t-1})$
   - □ Memoryless, we do not keep the information from previous states, but the state is rich enough to estimate the transition probabilities.

2. Events might cause other events

3. Root-causes of events are unknown, but should be able to estimate

4. Transitions might have prior probabilities

Example of Markov Chain



Intentionally not showing the self-loops

Markov matrix M

Also called stochastic matrix or transition matrix)

M is a square matrix whose columns are probability vectors $\overrightarrow{x_t}$.

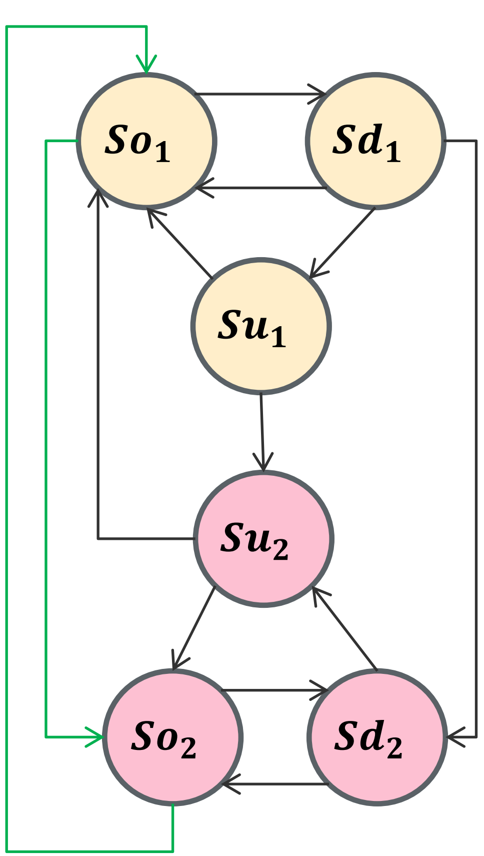
$\overrightarrow{x_t}$

Source states

Target states

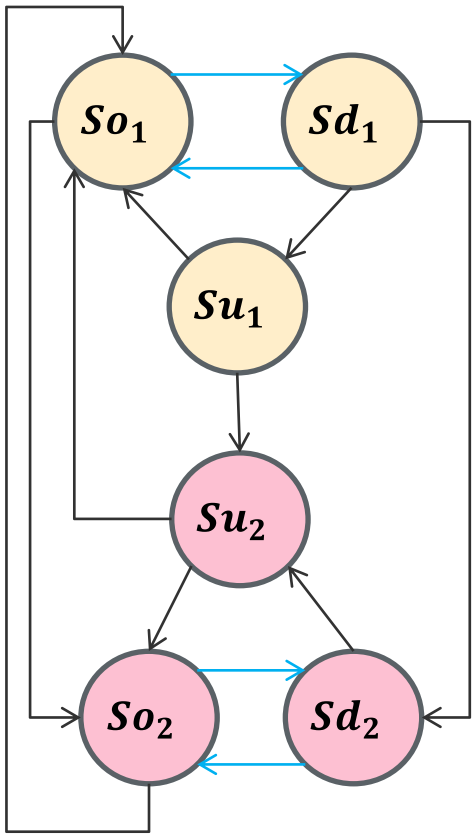
	So_1	Sd_1	Su_1	So_2	Sd_2	Su_2
So_1	0.15	0.50	0.25	0.80	0.0	0.25
Sd_1	0.05	0.25	0.0	0.0	0.0	0.0
Su_1	0.0	0.10	0.50	0.0	0.0	0.0
So_2	0.80	0.0	0.0	0.15	0.50	0.25
Sd_2	0.0	0.15	0.0	0.05	0.25	0.0
Su_2	0.0	0.0	0.25	0.0	0.25	0.50
$\sum$	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**

Types of Traces



Normal operation

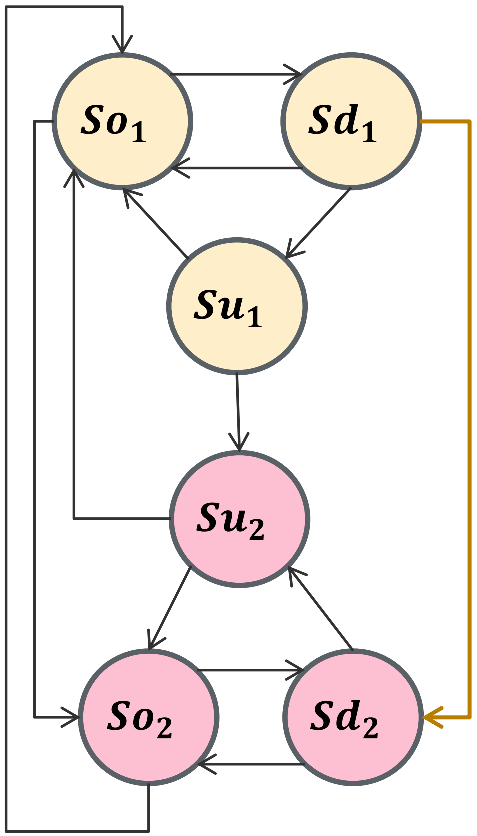
$(So_1; So_2; So_1)$



Intermittent failure

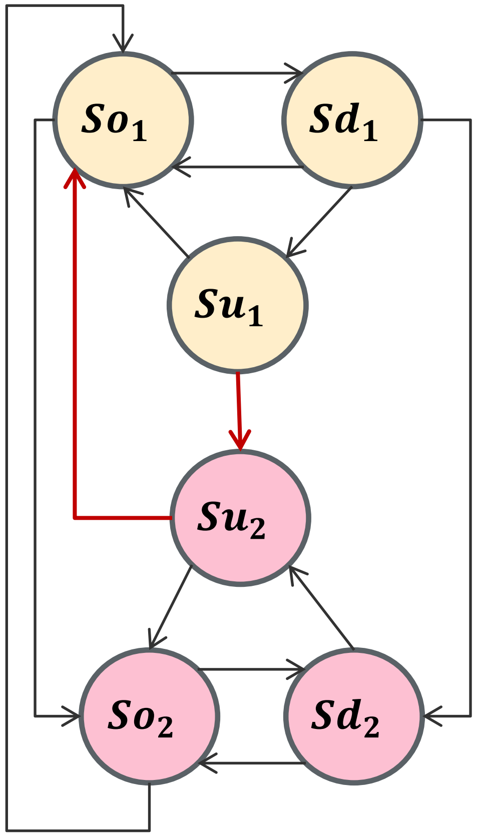
$(So_1 \rightarrow Sd_1 \rightarrow So_1)$

$(So_2 \rightarrow Sd_2 \rightarrow So_2)$



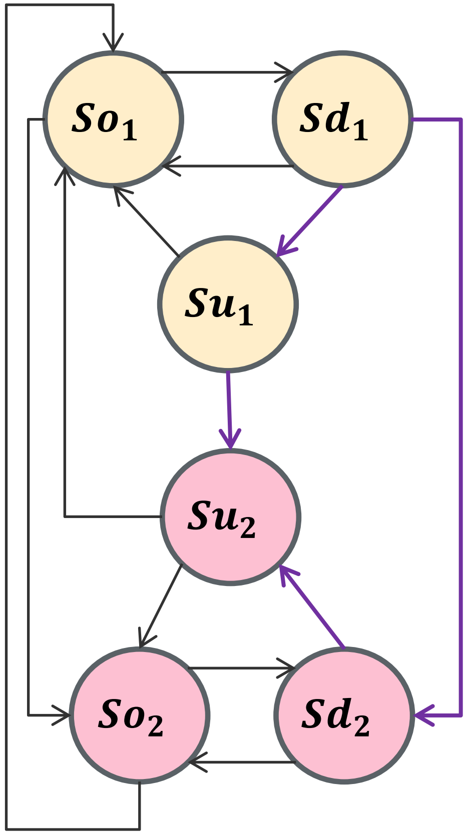
Systemic degradation

$(Sd_1 \rightarrow Sd_2)$



Failure masking

$(Su_1 \rightarrow Su_2 \rightarrow So_1)$

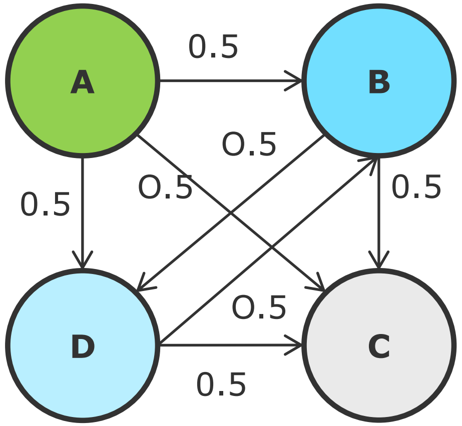


Failure cascade

$(Sd_1 \rightarrow Su_1 \rightarrow So_2)$

$(Sd_1 \rightarrow Sd_2 \rightarrow Su_2)$

Computation using a Markov Chain



Transition Matrix
$\boldsymbol{M}$

Uniform Prior probabilities before surfing
$\boldsymbol{R_o}$

Probabilities after iteration 1

$$\begin{array}{cccc} \textbf{A} & \textbf{B} & \textbf{C} & \textbf{D} \end{array}$$
$$\begin{bmatrix} 0 & 0 & 0.5 & 0.5 \\ 1 & 0 & 0 & 0.5 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 \end{bmatrix} \text{X} \begin{bmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{bmatrix} = \begin{bmatrix} 0 * 0.25 + 0 * 0.25 + 0.5 * 0.5 + 0.5 * 0.5 \\ \vdots \\ . \end{bmatrix} = \begin{bmatrix} 0.25 \\ 0.375 \\ 0.125 \\ 0.25 \end{bmatrix}$$

Next iteration

$$\overrightarrow{R_2} = \begin{bmatrix} 0.185 \\ 0.375 \\ 0.185 \\ 0.250 \end{bmatrix}$$

$$\ldots \lim_{t \to \infty} M^t \overrightarrow{R_0} \begin{array}{l} \begin{bmatrix} 0.217 \\ 0.348 \\ 0.174 \\ 0.261 \end{bmatrix} \end{array} \begin{array}{c} \textbf{A} \\ \textbf{B} \\ \textbf{C} \\ \textbf{D} \end{array}$$

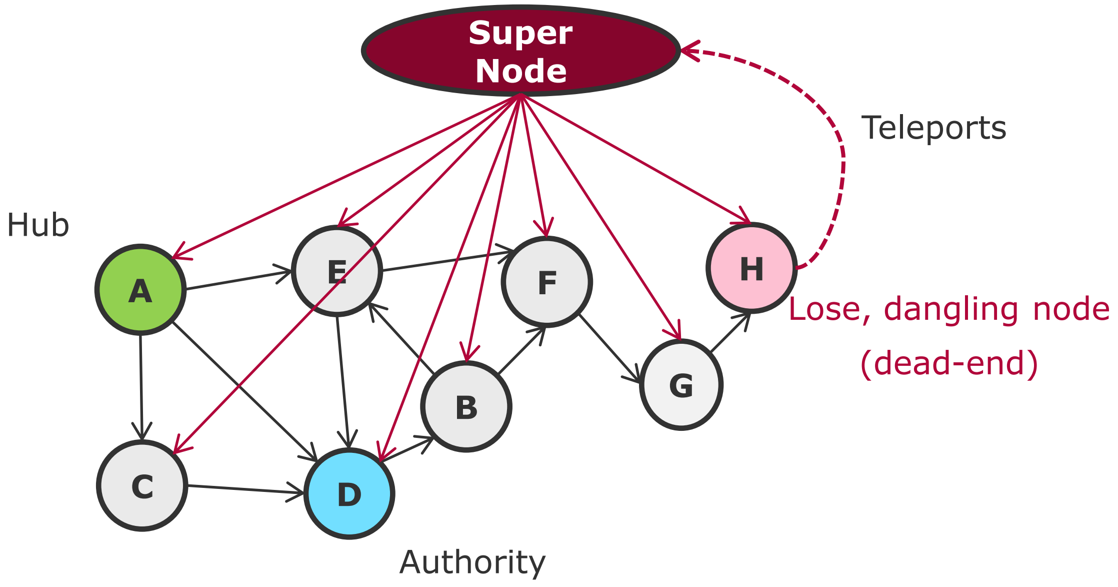
↑
Eigenvalues of M

A Markov chain has a **stationary distribution** if and only if the Markov chain is ergodic. If the Markov chain is ergodic, the stationary distribution is unique.

Markov Chain properties

- **Reducible**: if it is possible to get to any state from any state.
- **Periodicity**: a state in a Markov chain is periodic if the chain can return to the state **only** at multiples of some integer larger than 1. Thus, if we start at state *i*, the chain can return to this state *i* **only** at multiples of the period $\boldsymbol{T > 1}$. Conversely, if state *i* is aperiodic, then if $T = 1$.
- **Transient**: if, given that we start in state *i*, there is a non-zero probability that the chain **will never** return to *i*.
- **Recurrent**: if it is expected to return to state *i* within a finite number of steps.
- **Ergodicity**: a state *i* is ergodic if it is aperiodic and positive recurrent. If all states in an irreducible Markov chain are ergodic, then the chain is said to be ergodic.
- **Absorbing State**: a state *i* is called absorbing if it is impossible to leave this state. Therefore, the state *i* is absorbing if $p_{ii} = 1$ and $p_{ij} = 0\ for\ i \neq j$. If from every state we can reach an absorbing state, then the Markov chain is an absorbing Markov chain.

Dead-Ends [Brin, Page, Motwani & Winograd 99]



Implications of a super node:

- It creates a super connected component
- Which has a unique stationary distribution $\overrightarrow{R_s}$
- Which we can guarantee to reach regardless of initial state distribution $\overrightarrow{R_0}$

$$\lim_{t \to \infty} M^t \overrightarrow{R_o} = \overrightarrow{R_s}$$

At each step:

- With probability α , follow the link in the Markov Chain
- With probability $1 - \alpha$, jump to some random page with probability $\frac{1}{N}$

Hence, the new ranking is computed as follows

$$\Pr(i) = \sum_{j \in Nr(i)} \frac{\alpha}{D_j} Pr(j) + \frac{(1-\alpha)}{N}$$

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). *The PageRank citation ranking: Bringing order to the web*. Stanford InfoLab.

How to derive the Transition matrix M from Adjacency matrix A?

- For every row of A that has no 1, replace each element for $\frac{1}{N}$
- For all the other rows:
  - Divide each 1 occurrence in A by the number of 1's in the row (which is the out-degree)
  - Multiply the resulting matrix by $(1 - \alpha)$
  - Add $\frac{\alpha}{N}$ to every entry of the resulting matrix to obtain transition matrix M.

END