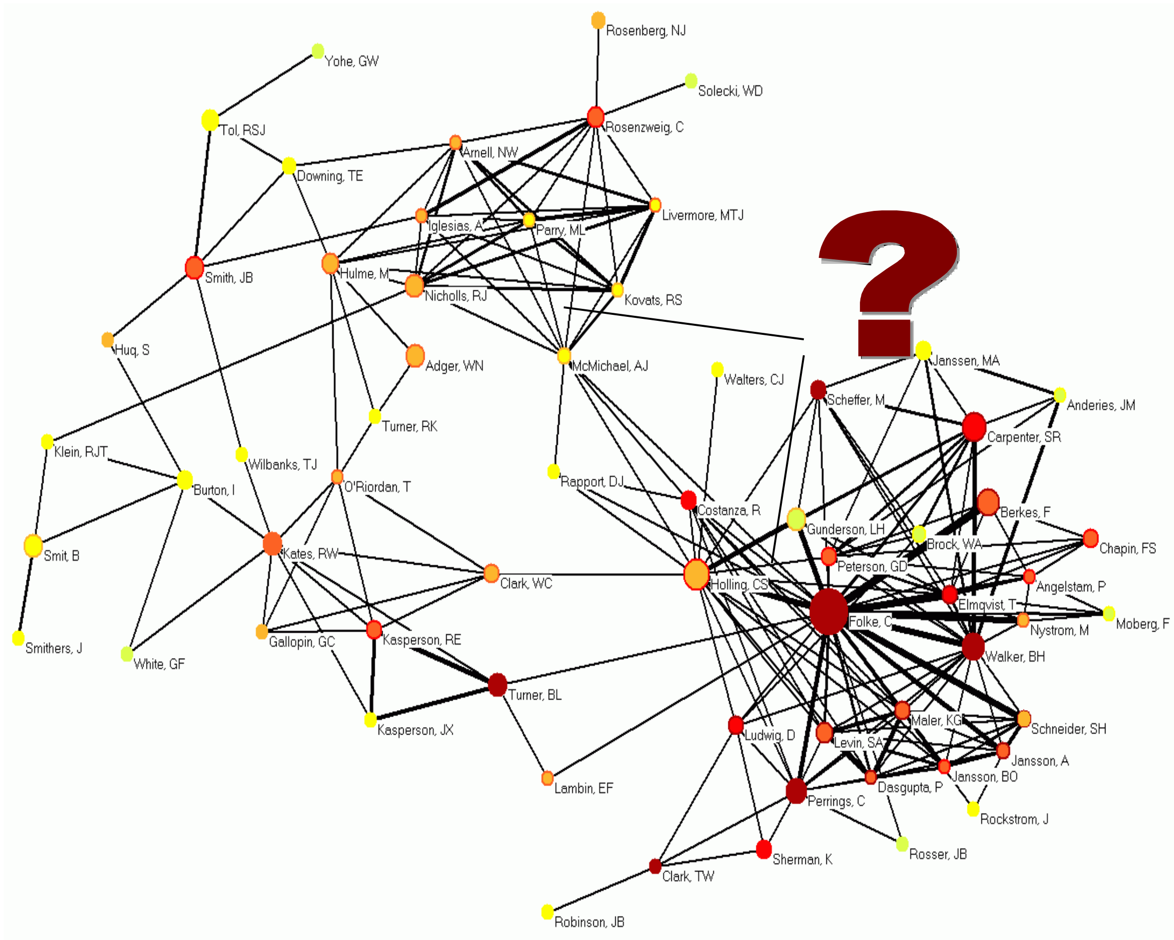
Random Walks on Graphs: An Overview

Purnamrita Sarkar

Motivation: Link prediction in social networks



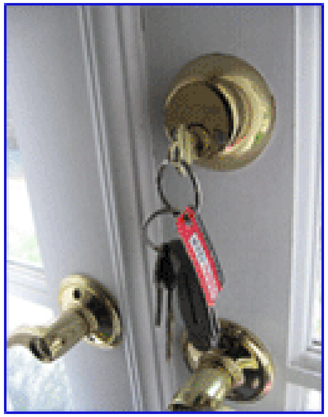
Motivation: Basis for recommendation

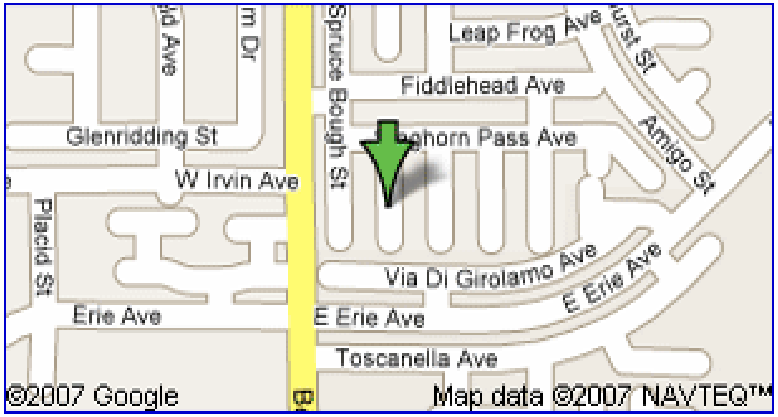
Motivation: Personalized search

Web Images Video News Maps Gmail more ▼

my car keys | Search | Advanced Search | Preferences

Web





In the front door, where you left them last night.

Where Are **My Car Keys**?

Why graphs?

- The underlying data is naturally a graph
  - Papers linked by citation
  - Authors linked by co-authorship
  - Bipartite graph of customers and products
  - Web-graph
  - Friendship networks: who knows whom

What are we looking for

- Rank nodes for a particular query
  - Top k matches for "Random Walks" from Citeseer
  - Who are the most likely co-authors of "Manuel Blum".
  - Top k book recommendations for Purna from Amazon
  - Top k websites matching "Sound of Music"
  - Top k friend recommendations for Purna when she joins "Facebook"

Talk Outline

- Basic definitions
  - Random walks
  - Stationary distributions
- Properties
- Applications

Definitions

- nxn Adjacency matrix A.
  - $A(i,j)$ = weight on edge from *i* to *j*
  - If the graph is undirected $A(i,j)=A(j,i)$, i.e. A is symmetric
- nxn Transition matrix P.
  - P is row stochastic
  - $P(i,j)$ = probability of stepping on node j from node i
    $= A(i,j)/\sum_i A(i,j)$
- nxn Laplacian Matrix L.
  - $L(i,j)=\sum_i A(i,j)-A(i,j)$
  - Symmetric positive semi-definite for undirected graphs
  - Singular

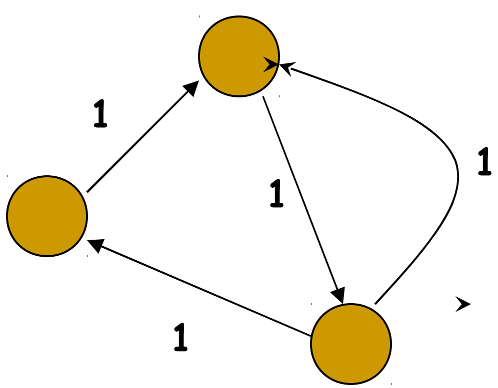
Definitions

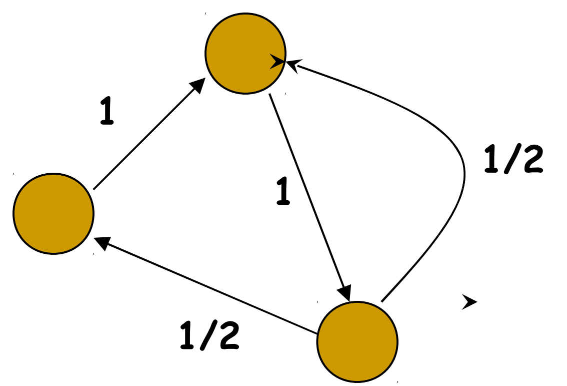
0	1	0
0	0	1
1	1	0

Adjacency matrix A

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

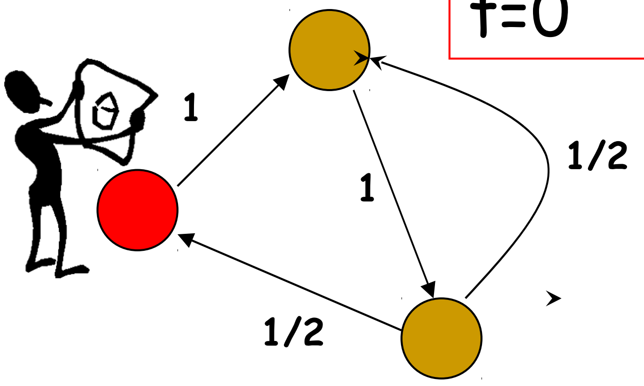
Transition matrix P



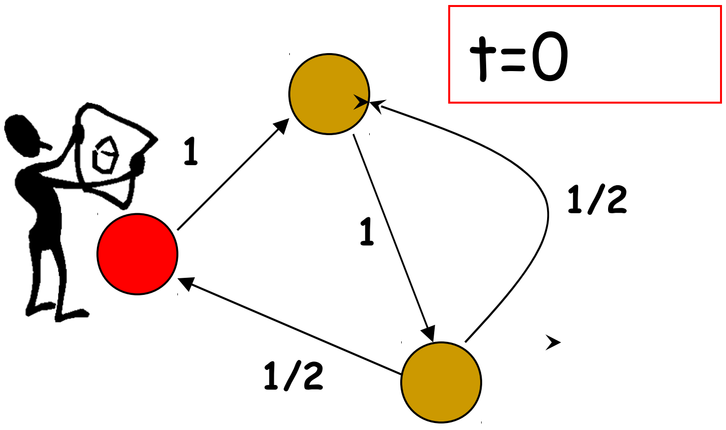


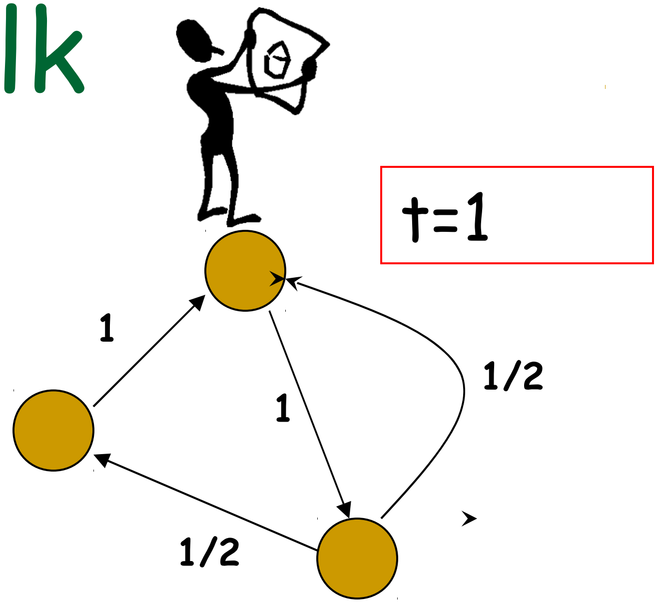
What is a random walk

t=0

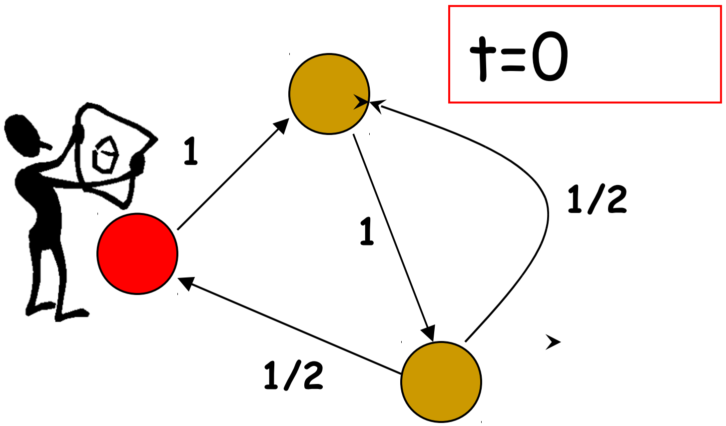


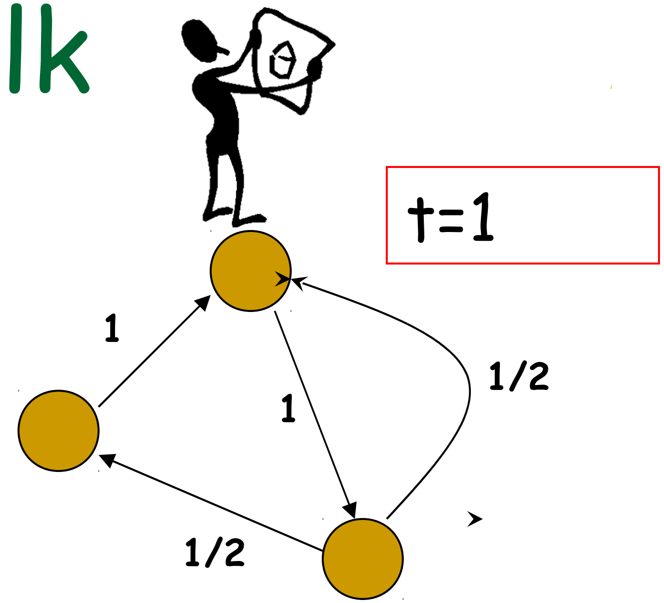
What is a random walk

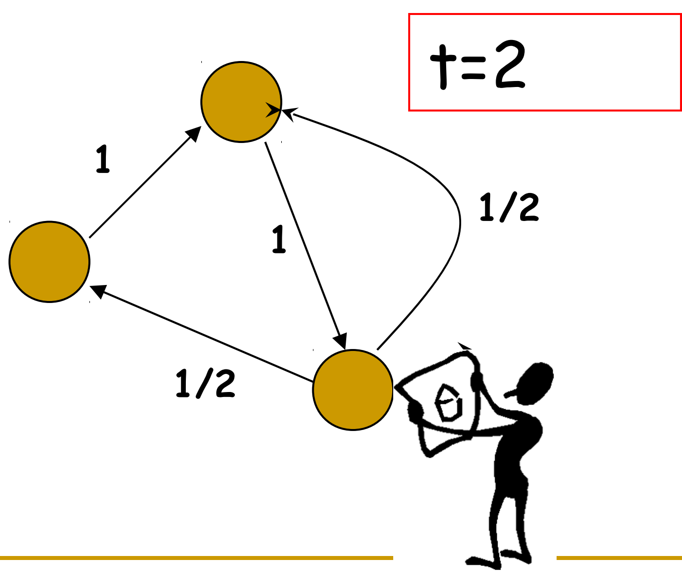




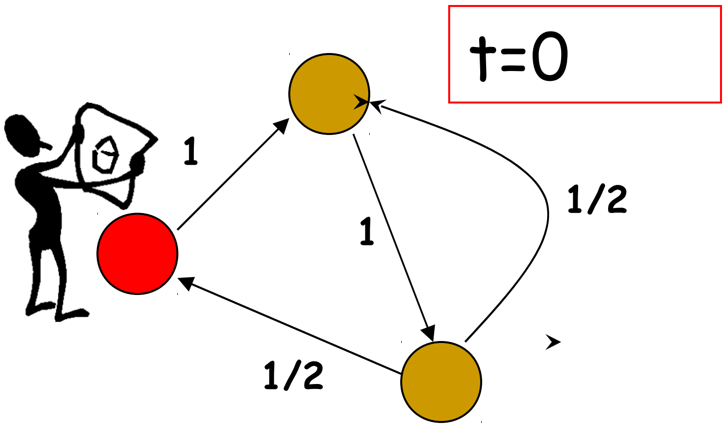
What is a random walk

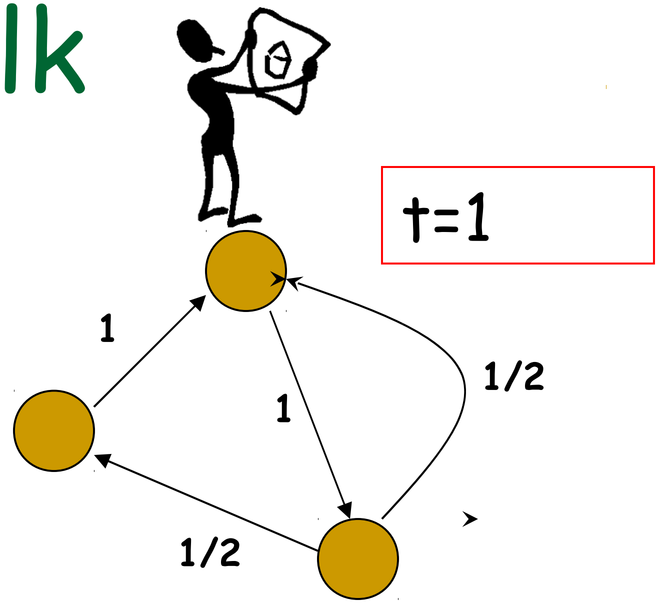


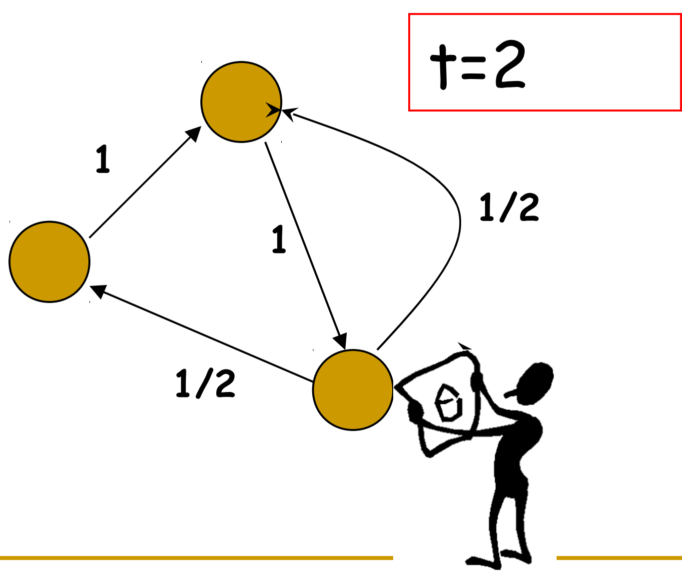


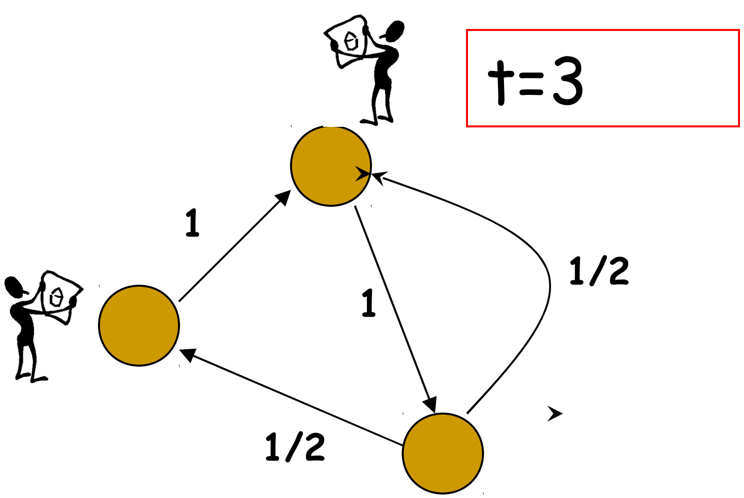


What is a random walk









Probability Distributions

- $x_t(i)$ = probability that the surfer is at node *i* at time *t*
- $x_{t+1}(i) = \sum_j$(Probability of being at node j)*Pr(j->i) $=\sum_j x_t(j)*P(j,i)$
- $x_{t+1} = x_t P = x_{t-1}*P*P = x_{t-2}*P*P*P = \ldots = x_0 P^t$
- What happens when the surfer keeps walking for a long time?

Stationary Distribution

- When the surfer keeps walking for a long time
- When the distribution does not change anymore
  - i.e. $x_{T+1} = x_T$
- For "well-behaved" graphs this does not depend on the start distribution!!

What is a stationary distribution? Intuitively and Mathematically

What is a stationary distribution? Intuitively and Mathematically

- The stationary distribution at a node is related to the amount of time a random walker spends visiting that node.

What is a stationary distribution? Intuitively and Mathematically

- The stationary distribution at a node is related to the amount of time a random walker spends visiting that node.
- Remember that we can write the probability distribution at a node as
  - $x_{t+1} = x_t P$

What is a stationary distribution? Intuitively and Mathematically

- The stationary distribution at a node is related to the amount of time a random walker spends visiting that node.
- Remember that we can write the probability distribution at a node as
  - $x_{t+1} = x_t P$
- For the stationary distribution v_0 we have
  - $v_0 = v_0 P$

What is a stationary distribution? Intuitively and Mathematically

- The stationary distribution at a node is related to the amount of time a random walker spends visiting that node.
- Remember that we can write the probability distribution at a node as
  - $x_{t+1} = x_t P$
- For the stationary distribution v_0 we have
  - $v_0 = v_0 P$
- Whoa! that's just the left eigenvector of the transition matrix !

Talk Outline

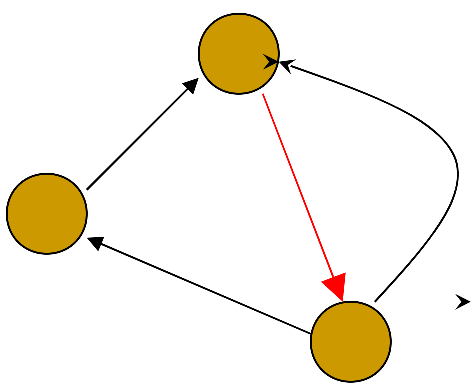
- Basic definitions
  - Random walks
  - Stationary distributions
- Properties
  - Perron frobenius theorem
- Applications

Interesting questions

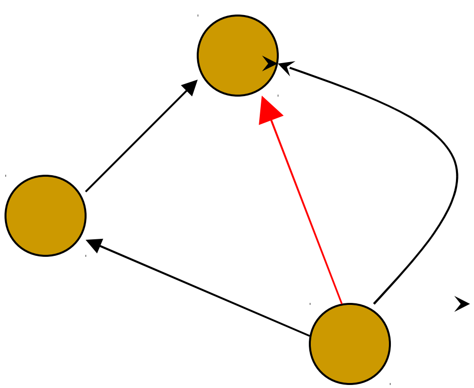
- Does a stationary distribution always exist? Is it unique?
  - Yes, if the graph is "well-behaved".
- What is "well-behaved"?
  - We shall talk about this soon.
- How fast will the random surfer approach this stationary distribution?
  - Mixing Time!

Well behaved graphs

- Irreducible: There is a path from every node to every other node.



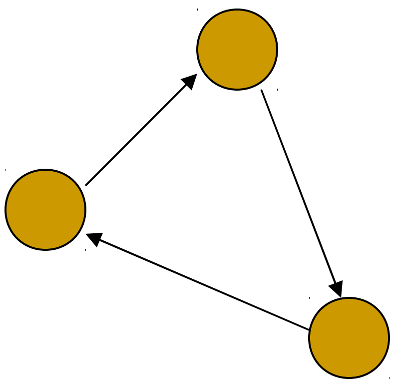
Irreducible



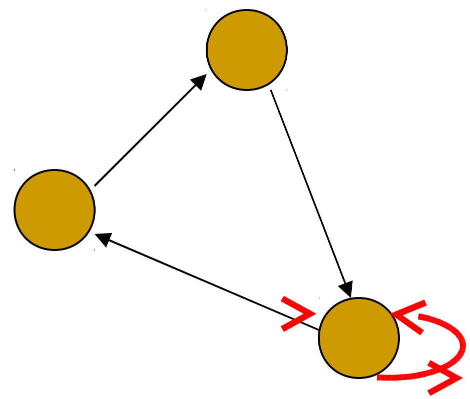
Not irreducible

Well behaved graphs

- Aperiodic: The GCD of all cycle lengths is *1*. The GCD is also called period.



Periodicity is 3



Aperiodic

Implications of the Perron Frobenius Theorem

- If a markov chain is irreducible and aperiodic then the largest eigenvalue of the transition matrix will be equal to **1** and all the other eigenvalues will be strictly less than **1**.
  - Let the eigenvalues of P be $\{\sigma_i \mid i=0:n-1\}$ in non-increasing order of σ_i .
  - $\sigma_0 = 1 > \sigma_1 > \sigma_2 >= \ldots\ldots >= \sigma_n$

Implications of the Perron Frobenius Theorem

- If a markov chain is irreducible and aperiodic then the largest eigenvalue of the transition matrix will be equal to **1** and all the other eigenvalues will be strictly less than **1**.
  - Let the eigenvalues of P be $\{\sigma_i \mid i=0:n-1\}$ in non-increasing order of σ_i .
  - $\sigma_0 = 1 > \sigma_1 > \sigma_2 >= \ldots\ldots >= \sigma_n$

- These results imply that for a well behaved graph there exists an unique stationary distribution.

- More details when we discuss pagerank.

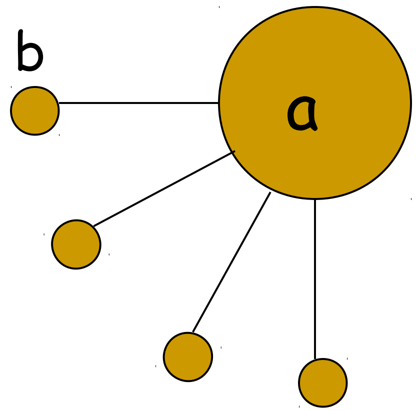
Some fun stuff about undirected graphs

- A connected undirected graph is irreducible
- A connected non-bipartite undirected graph has a stationary distribution proportional to the degree distribution!
- Makes sense, since larger the degree of the node more likely a random walk is to come back to it.

Talk Outline

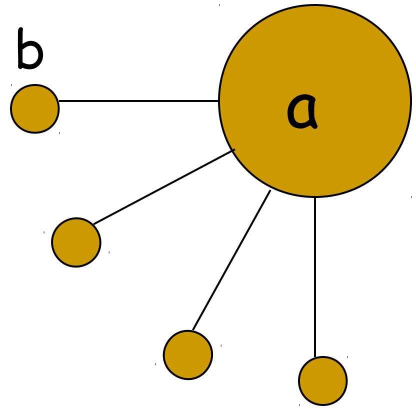
- Basic definitions
  - Random walks
  - Stationary distributions
- Properties
  - Perron frobenius theorem
  - Electrical networks, hitting and commute times
    - Euclidean Embedding
- Applications

Proximity measures from random walks



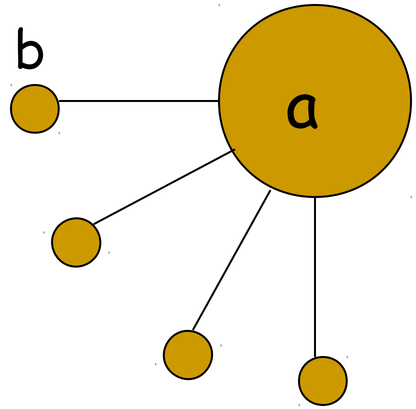
- How long does it take to hit node ***b*** in a random walk starting at node ***a*** ? Hitting time.
- How long does it take to hit node ***b*** and come back to node ***a*** ? Commute time.

Hitting and Commute times



- Hitting time from node *i* to node *j*
  - Expected number of hops to hit node *j* starting at node *i*.
  - Is not symmetric. h(a,b) > h(a,b)
  - $h(i,j) = 1 + \Sigma_{k \in nbs(A)} p(i,k)h(k,j)$

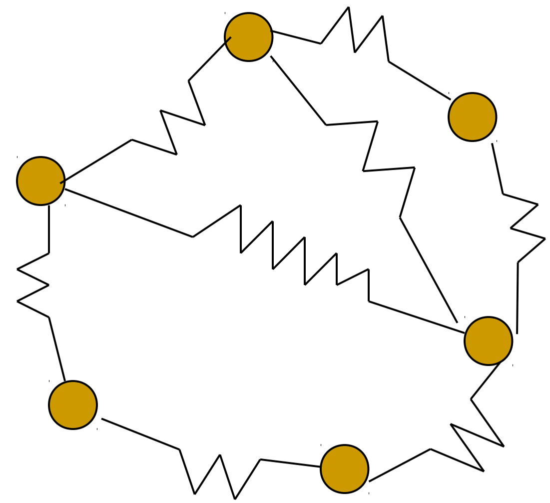
Hitting and Commute times



- Commute time between node **i** and **j**
  - Is expected time to hit node **j** and come back to **i**
  - $c(i,j) = h(i,j) + h(j,i)$
  - Is symmetric. $c(a,b) = c(b,a)$

Relationship with Electrical networks[1,2]

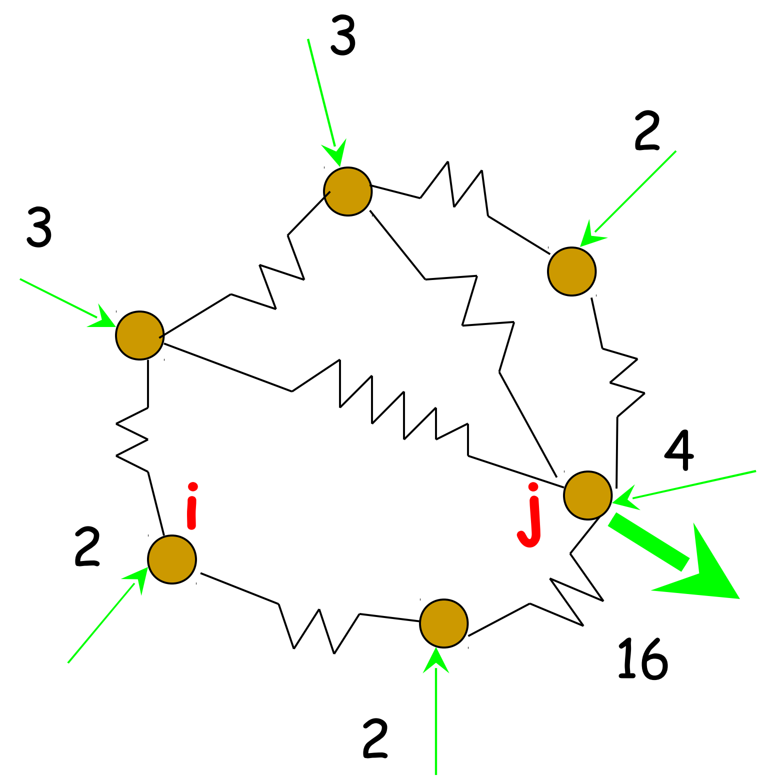
- Consider the graph as a n-node resistive network.
- Each edge is a resistor of 1 Ohm.
- Degree of a node is number of neighbors
- Sum of degrees = 2*m
  - m being the number of edges



1. **Random Walks and Electric Networks** , Doyle and Snell, 1984
2. **The Electrical Resistance Of A Graph Captures Its Commute And Cover Times,** Ashok K. Chandra, Prabhakar Raghavan, Walter L. Ruzzo, Roman Smolensky, Prasoon Tiwari, 1989

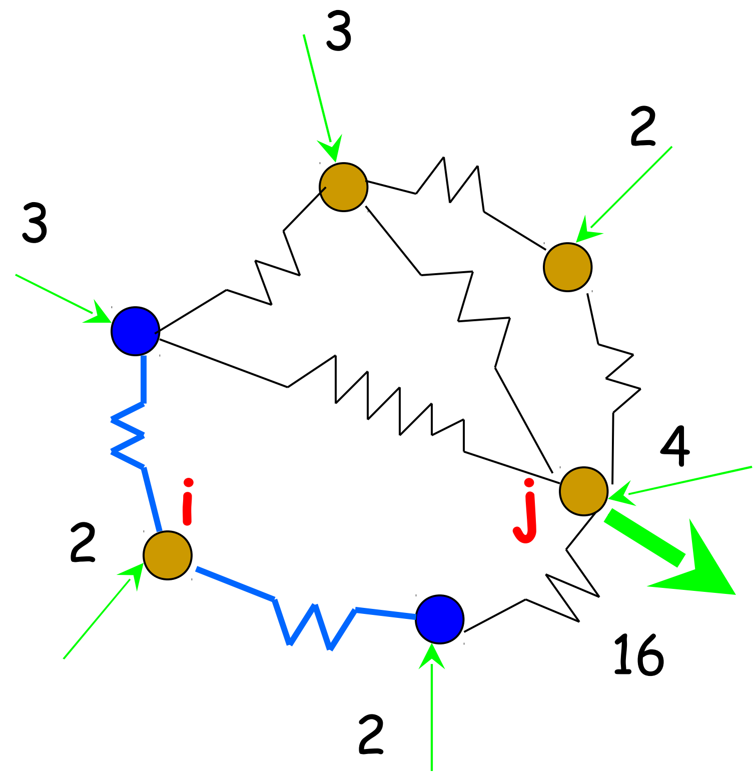
Relationship with Electrical networks

- Inject d(i) amp current in each node
- Extract ***2m*** amp current from node **j**.
- Now what is the voltage difference between **i** and **j** ?



Relationship with Electrical networks

- Whoa!! Hitting time from **i** to **j** is exactly the voltage drop when you inject respective degree amount of current in every node and take out 2*m from **j**!



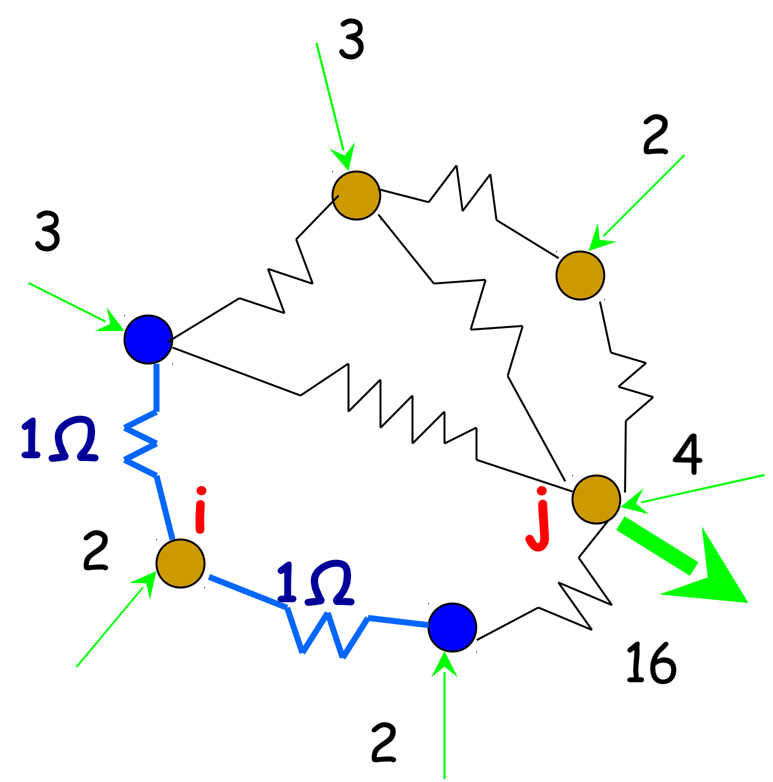
Relationship with Electrical networks

- Consider neighbors of **i** i.e. NBS(**i**)
- Using Kirchhoff's law
  $d(i) = \Sigma_{k \in NBS(A)} \Phi(i,j) - \Phi(k,j)$

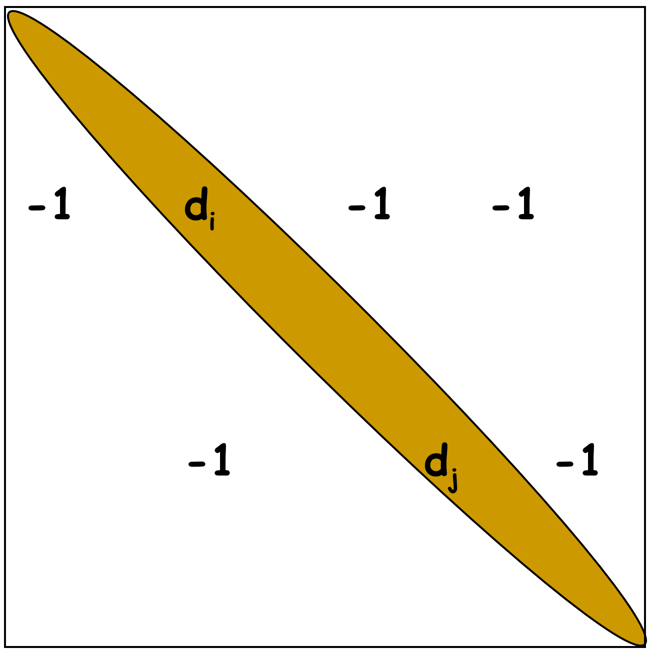
$$\phi(i,j) = 1 + \frac{1}{d(i)} \sum_{k \in NBS(i)} \phi(k,j)$$

- Oh wait, that's also the definition of hitting time from **i** to **j**!

$$h(i,j) = 1 + \sum_{k \in NBS(i)} P(i,k) h(k,j)$$



Hitting times and Laplacians

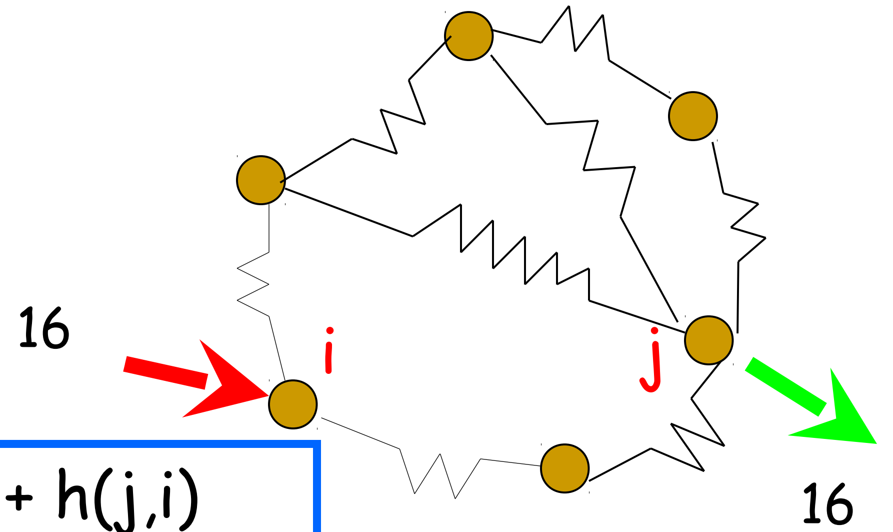


L

$$\begin{pmatrix} \phi_0 \\ . \\ \phi_i \\ . \\ . \\ . \\ \phi_j \\ . \\ \phi_{n-1} \end{pmatrix} = \begin{pmatrix} d_0 \\ . \\ d_i \\ . \\ . \\ . \\ d_j - 2m \\ . \\ d_{n-1} \end{pmatrix}$$

$$h(i,j) = \Phi_i - \Phi_j$$

Relationship with Electrical networks

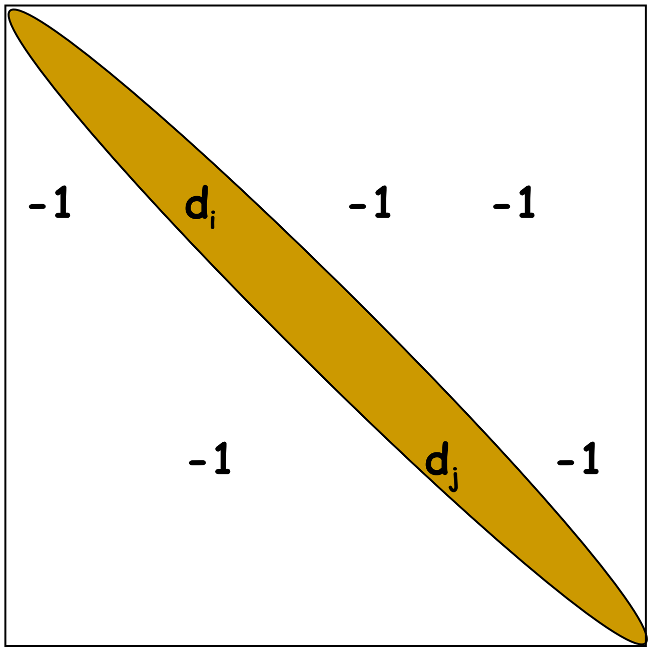


h(i,j) + h(j,i)

$$c(i,j) = h(i,j) + h(j,i) = 2m^*R_{eff}(i,j)$$ [1]

1. **The Electrical Resistance Of i Graph Captures Its Commute And Cover Times,** Ashok K. Chandra, Prabhakar Raghavan, Walter L. Ruzzo, Roman Smolensky, Prasoon Tiwari, 1989

Commute times and Lapacians



L

$$\begin{pmatrix} \phi_0 \\ . \\ \phi_i \\ . \\ . \\ . \\ \phi_j \\ . \\ \phi_{n-1} \end{pmatrix} = \begin{pmatrix} 0 \\ . \\ 2m \\ . \\ . \\ . \\ -2m \\ . \\ 0 \end{pmatrix}$$

- $C(i,j) = \Phi_i - \Phi_j$

  $= 2m\,(e_i - e_j)^T L^+ (e_i - e_j)$

  $= 2m\,(x_i - x_j)^T (x_i - x_j)$

- $x_i = (L^+)^{1/2}\, e_i$

Commute times and Laplacians

- Why is this interesting ?
- Because, this gives a very intuitive definition of embedding the points in some Euclidian space, s.t. the commute times is the squared Euclidian distances in the transformed space.[1]

1. **The Principal Components Analysis of a Graph, and its Relationships to Spectral Clustering** . M. Saerens, et al, ECML '04

L^+: some other interesting measures of similarity[1]

- $L^+_{ij} = x_i^T x_j$ = inner product of the position vectors
- $L^+_{ii} = x_i^T x_i$ = square of length of position vector of *i*
- Cosine similarity $$\frac{l^+_{ij}}{\sqrt{l^+_{ii} l^+_{jj}}}$$

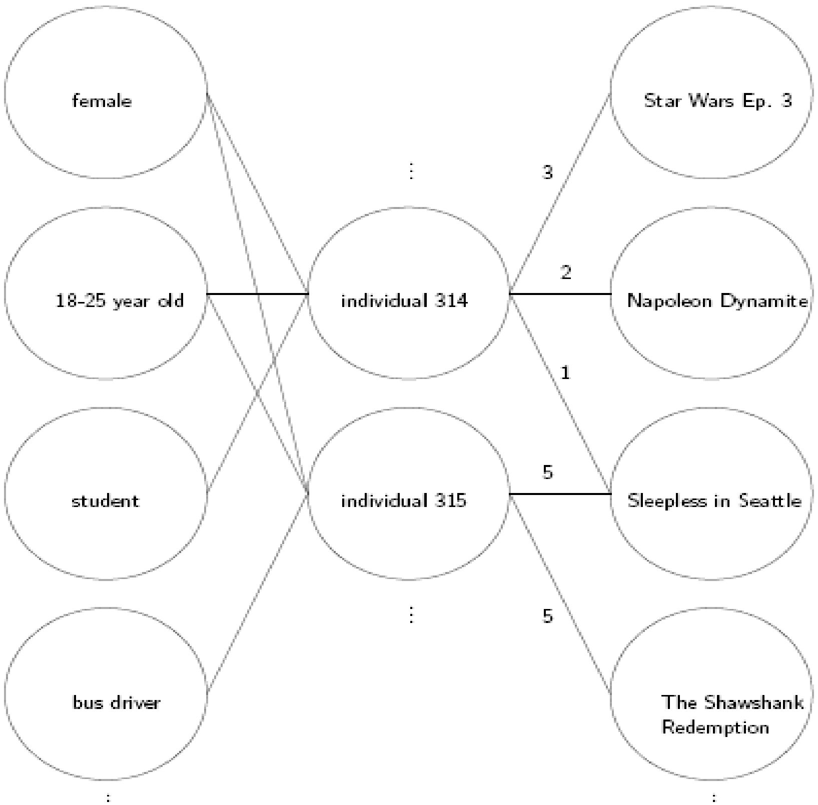
1. **A random walks perspective on maximising satisfaction and profit**. Matthew Brand, SIAM '05

Talk Outline

- Basic definitions
  - Random walks
  - Stationary distributions
- Properties
  - Perron frobenius theorem
  - Electrical networks, hitting and commute times
    - Euclidean Embedding
- Applications
  - Recommender Networks
  - Pagerank
    - Power iteration
    - Convergencce
  - Personalized pagerank

Recommender Networks[1]

An example association graph



1. **A random walks perspective on maximising satisfaction and profit**. Matthew Brand, SIAM '05

Recommender Networks

- For a customer node *i* define similarity as
  - H(i,j)
  - C(i,j)
  - Or the cosine similarity $$\frac{L^+_{ij}}{\sqrt{L^+_{ii} L^+_{jj}}}$$
- Now the question is how to compute these quantities quickly for very large graphs.
  - Fast iterative techniques (Brand 2005)
  - Fast Random Walk with Restart (Tong, Faloutsos 2006)
  - Finding nearest neighbors in graphs (Sarkar, Moore 2007)

Ranking algorithms on the web

- HITS (Kleinberg, 1998) & Pagerank (Page & Brin, 1998)

- We will focus on Pagerank for this talk.
  - An webpage is important if other important pages point to it.
  - Intuitively $v(i) = \sum_{j \rightarrow i} \frac{v(j)}{\deg^{out}(j)}$
  - **v** works out to be the stationary distribution of the markov chain corresponding to the web.

Pagerank & Perron-frobenius

- Perron Frobenius only holds if the graph is irreducible and aperiodic.

- But how can we guarantee that for the web graph?
  - Do it with a small restart probability *c*.

- At any time-step the random surfer
  - jumps (teleport) to any other node with probability ***c***
  - jumps to its direct neighbors with total probability ***1-c***.

$$\tilde{\mathbf{P}} = (1-c)\mathbf{P} + c\mathbf{U}$$

$$\mathbf{U}_{ij} = \frac{1}{n} \forall i, j$$

Power iteration

- Power Iteration is an algorithm for computing the stationary distribution.
  - Start with any distribution x_0
  - Keep computing $x_{t+1} = x_t P$
  - Stop when x_{t+1} and x_t are almost the same.

Power iteration

- Why should this work?
- Write x_0 as a linear combination of the left eigenvectors $\{v_0, v_1, \ldots, v_{n-1}\}$ of P
- Remember that v_0 is the stationary distribution.
- $x_0 = c_0 v_0 + c_1 v_1 + c_2 v_2 + \ldots + c_{n-1} v_{n-1}$

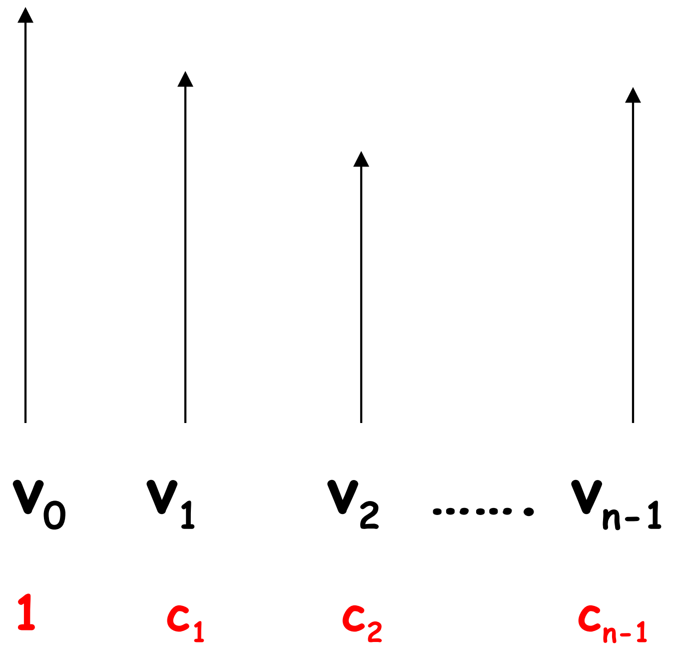
Power iteration

- Why should this work?
- Write x_0 as a linear combination of the left eigenvectors $\{v_0, v_1, \ldots , v_{n-1}\}$ of P
- Remember that v_0 is the stationary distribution.
- $x_0 = c_0v_0 + c_1v_1 + c_2v_2 + \ldots + c_{n-1}v_{n-1}$

$c_0 = 1$. WHY? (slide 71)

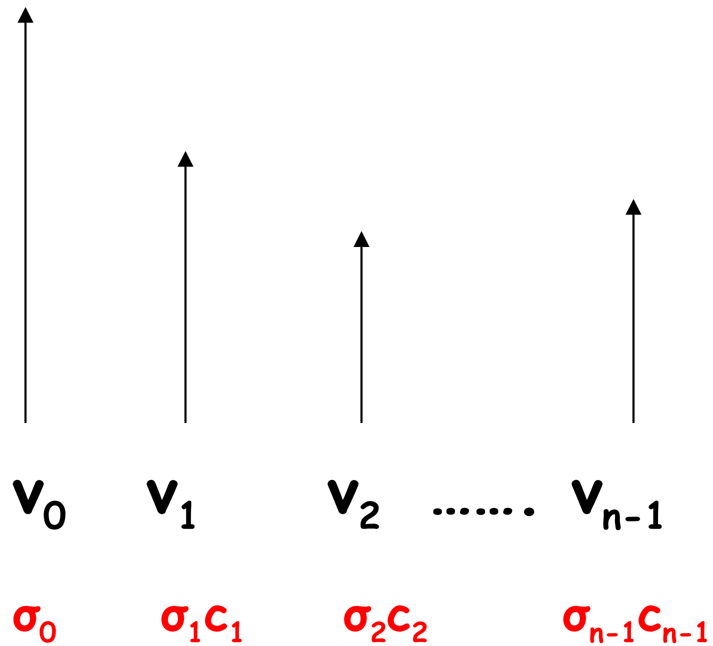
Power iteration

x_0



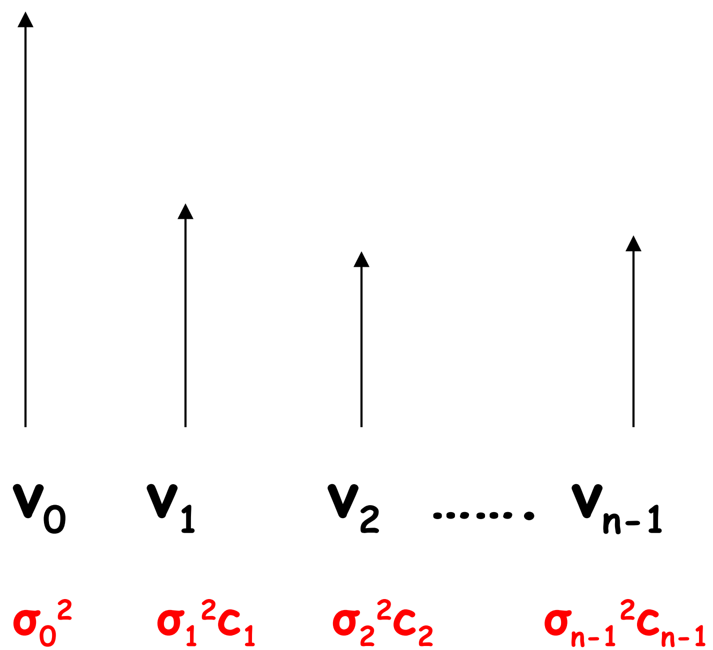
Power iteration

$$x_1 = x_0 \tilde{P}$$



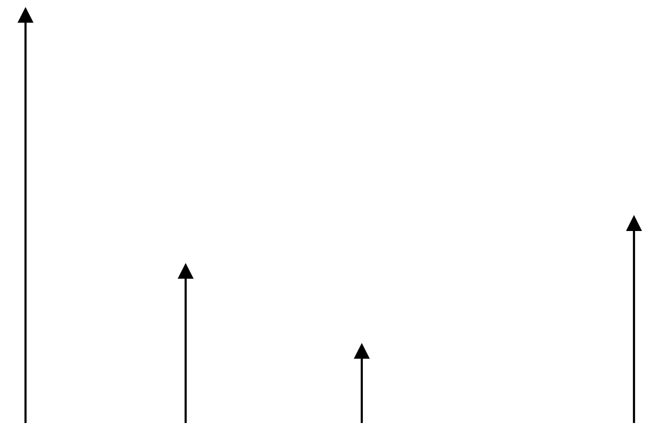
Power iteration

$$x_2 = x_1 \tilde{P} = x_0 \tilde{P}^2$$



Power iteration

$$x_t = x_0 \tilde{P}^t$$



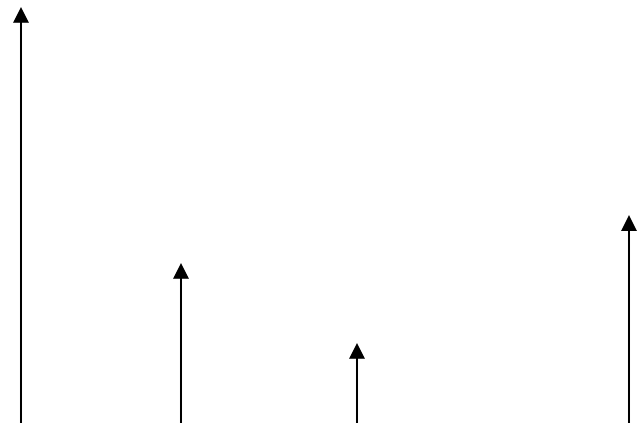
v_0 v_1 v_2 ……. v_{n-1}

σ_0^t $\sigma_1^t c_1$ $\sigma_2^t c_2$ $\sigma_{n-1}^t c_{n-1}$

Power iteration

$$x_t = x_0 \tilde{P}^t$$

$\sigma_0 = 1 > \sigma_1 \geq \ldots \geq \sigma_n$



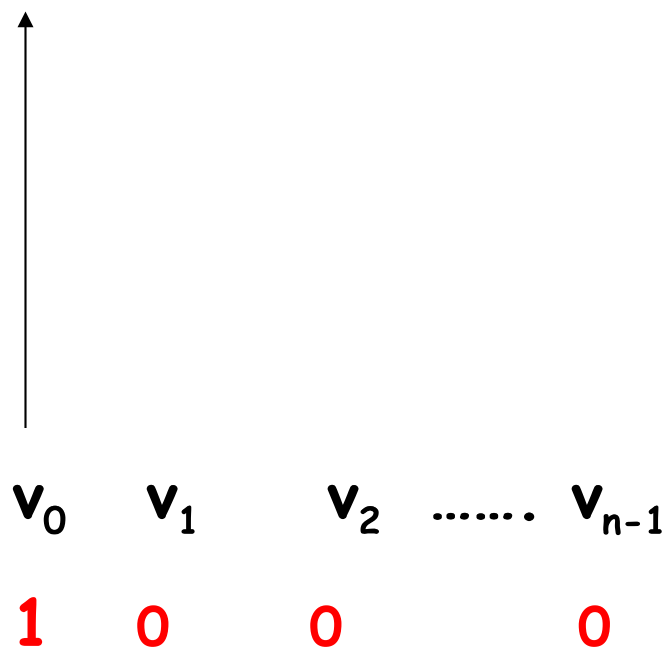
$v_0 \quad v_1 \quad v_2 \quad \ldots\ldots \quad v_{n-1}$

$1 \quad \sigma_1^t c_1 \quad \sigma_2^t c_2 \quad \sigma_{n-1}^t c_{n-1}$

Power iteration

x_∞

$\sigma_0 = 1 > \sigma_1 \geq \ldots \geq \sigma_n$



Convergence Issues

- Formally $||x_0P^t - v_0|| \leq |\lambda|^t$
  - λ is the eigenvalue with second largest magnitude
- The smaller the second largest eigenvalue (in magnitude), the faster the mixing.
- For $\lambda<1$ there exists an unique stationary distribution, namely the first left eigenvector of the transition matrix.

Pagerank and convergence

- The transition matrix pagerank uses really is

$$\tilde{P} = (1 - c)P + cU$$

- The second largest eigenvalue of $\tilde{P}$ can be proven[1] to be ≤ (1-c)
- Nice! This means pagerank computation will converge fast.

1. **The Second Eigenvalue of the Google Matrix,** Taher H. Haveliwala and Sepandar D. Kamvar, Stanford University Technical Report, 2003.

Pagerank

- We are looking for the vector v s.t.

$$\mathbf{v} = (1 - c)\mathbf{vP} + c\mathbf{r}$$

- **r** is a distribution over web-pages.
- If **r** is the uniform distribution we get pagerank.
- What happens if **r** is non-uniform?

Pagerank

- We are looking for the vector v s.t.

$$\mathbf{v} = (1 - c)\mathbf{v}\mathbf{P} + c\mathbf{r}$$

- **r** is a distribution over web-pages.

- If **r** is the uniform distribution we get pagerank.

- What happens if **r** is non-uniform?

Personalization

Personalized Pagerank[1,2,3]

- The only difference is that we use a non-uniform teleportation distribution, i.e. at any time step teleport to a set of webpages.

- In other words we are looking for the vector **v** s.t.

$$\mathbf{v} = (1 - c)\mathbf{v}P + c\mathbf{r}$$

- **r** is a non-uniform **preference** vector specific to an user.

- **v** gives "personalized views" of the web.

1. ***Scaling Personalized Web Search***, Jeh, Widom. 2003

2. ***Topic-sensitive PageRank***, *Haveliwala, 2001*

3. ***Towards scaling fully personalized pagerank***, *D. Fogaras and B. Racz, 2004*

Personalized Pagerank

- **Pre-computation: r** is not known from before
- Computing during query time takes too long
- A crucial observation[1] is that the personalized pagerank vector is linear w.r.t **r**

$$\mathbf{r} = \begin{pmatrix} \alpha \\ 0 \\ 1-\alpha \end{pmatrix} \Rightarrow v(r) = \alpha v(r_0) + (1-\alpha) v(r_2)$$

$$\mathbf{r}_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \mathbf{r}_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Scaling Personalized Web Search, Jeh, Widom. 2003

Topic-sensitive pagerank (Haveliwala'01)

- Divide the webpages into 16 broad categories
- For each category compute the biased personalized pagerank vector by uniformly teleporting to websites under that category.
- At query time the probability of the query being from any of the above classes is computed, and the final page-rank vector is computed by a linear combination of the biased pagerank vectors computed offline.

Personalized Pagerank: Other Approaches

- Scaling Personalized Web Search (Jeh & Widom '03)
- Towards scaling fully personalized pagerank: algorithms, lower bounds and experiments (Fogaras et al, 2004)
- Dynamic personalized pagerank in entity-relation graphs. (Soumen Chakrabarti, 2007)

Personalized Pagerank (Purna's Take)

- But, whats the guarantee that the new transition matrix will still be irreducible?

- Check out
  - The Second Eigenvalue of the Google Matrix, *Taher H. Haveliwala and Sepandar D. Kamvar*, Stanford University Technical Report, 2003.
  - Deeper Inside PageRank, *Amy N. Langville. and Carl D. Meyer.* Internet Mathematics, 2004.

- As long as you are adding any rank one (where the matrix is a repetition of one distinct row) matrix of form ($\mathbf{1}^T\mathbf{r}$) to your transition matrix as shown before,
  - $\lambda \leq 1-c$

Talk Outline

- Basic definitions
  - Random walks
  - Stationary distributions
- Properties
  - Perron frobenius theorem
  - Electrical networks, hitting and commute times
    - Euclidean Embedding
- Applications
  - Recommender Networks
  - Pagerank
    - Power iteration
    - Convergence
  - Personalized pagerank
  - Rank stability

Rank stability

- How does the ranking change when the link structure changes?
- The web-graph is changing continuously.
- How does that affect page-rank?

Rank stability[1] (On the Machine Learning papers from the CORA[2] database)

Rank on the entire database.

Rank on 5 perturbed datasets by deleting 30% of the papers

1	"Genetic Algorithms in Search, Optimization and…", Goldberg	1	1	1	1	1
2	"Learning internal representations by error…", Rumelhart+al	2	2	2	2	2
3	"Adaptation in Natural and Artificial Systems", Holland	3	5	6	4	5
4	"Classification and Regression Trees", Breiman+al	4	3	5	5	4
5	"Probabilistic Reasoning in Intelligent Systems", Pearl	5	6	3	6	3
6	"Genetic Programming: On the Programming of …", Koza	6	4	4	3	6
7	"Learning to Predict by the Methods of Temporal …", Sutton	7	7	7	7	7
8	"Pattern classification and scene analysis", Duda+Hart	8	8	8	8	9
9	"Maximum likelihood from incomplete data via…", Dempster+al	10	9	9	11	8
10	"UCI repository of machine learning databases", Murphy+Aha	9	11	10	9	10
11	"Parallel Distributed Processing", Rumelhart+McClelland	-	-	-	10	-
12	"Introduction to the Theory of Neural Computation", Hertz+al	-	10	-	-	-

1. **Link analysis, eigenvectors, and stability,** Andrew Y. Ng, Alice X. Zheng and Michael Jordan, IJCAI-01
2. **Automating the contruction of Internet portals with machine learning,** A. Mc Callum, K. Nigam, J. Rennie, K. Seymore, In Information Retrieval Journel, 2000

Rank stability

- Ng et al 2001: $\tilde{\mathbf{P}} = (1-c)\mathbf{P} + c\mathbf{U}$
- Theorem: if **v** is the left eigenvector of $\tilde{P}$. Let the pages $\mathbf{i}_1$, $\mathbf{i}_2$,…, $\mathbf{i}_k$ be changed in any way, and let **v'** be the new pagerank. Then

$$\|\mathbf{v} - \mathbf{v}'\|_1 \le \frac{\sum_{j=1}^{k} \mathbf{v}(i_j)}{c}$$

- So if *c* is not too close to *0*, the system would be rank stable and also converge fast!

Conclusion

- Basic definitions
  - Random walks
  - Stationary distributions
- Properties
  - Perron frobenius theorem
  - Electrical networks, hitting and commute times
    - Euclidean Embedding
- Applications
  - Pagerank
    - Power iteration
    - Convergencce
  - Personalized pagerank
  - Rank stability

Thanks!

Please send email to Purna at psarkar@cs.cmu.edu with questions, suggestions, corrections ☺

Acknowledgements

- Andrew Moore
- Gary Miller
  - Check out Gary's Fall 2007 class on "*Spectral Graph Theory, Scientific Computing, and Biomedical Applications*"
  - *http://www.cs.cmu.edu/afs/cs/user/glmiller/public/Scientific-Computing/F*
- Fan Chung Graham's course on
  - *Random Walks on Directed and Undirected Graphs*
  - *http://www.math.ucsd.edu/~phorn/math261/*
- Random Walks on Graphs: A Survey, *Laszlo Lov'asz*
- Reversible Markov Chains and Random Walks on Graphs, *D Aldous, J Fill*
- Random Walks and Electric Networks, *Doyle & Snell*

Convergence Issues[1]

- Lets look at the vectors x for t=1,2,…
- Write x_0 as a linear combination of the eigenvectors of P
- $x_0 = c_0v_0 + c_1v_1 + c_2v_2 + \dots + c_{n-1}v_{n-1}$

$c_0 = 1$. WHY?

Remember that **1** is the right eigenvector of P with eigenvalue 1, since P is stochastic. i.e. $P*\mathbf{1}^T = \mathbf{1}^T$. Hence $v_i\mathbf{1}^T = 0$ if $i \neq 0$.

$1 = x*\mathbf{1}^T = c_0v_0*\mathbf{1}^T = c_0$. Since v_0 and x_0 are both distributions

1. We are assuming that **P** is diagonalizable. The non-diagonalizable case is trickier, you can take a look at Fan Chung Graham's class notes (the link is in the acknowledgements section).