

CSCI 5417
Information Retrieval Systems

Jim Martin

Lecture 20
11/3/2011

Today

- Finish PageRank
- HITs
- Start ML-based ranking

PageRank Sketch

- The pagerank of a page is based on the pagerank of the pages that point at it.
 - Roughly

$$\Pr(P) = \sum_{in \in P} \frac{\Pr(in)}{V(in)}$$

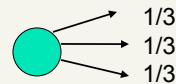
11/11/11

CSCI 5417 - IR

3

PageRank scoring

- Imagine a browser doing a random walk on web pages:
 - Start at a random page
 - At each step, go out of the current page along one of the links on that page, equiprobably
- “In the steady state” each page has a long-term visit rate - use this as the page’s score
 - Pages with low rank are pages rarely visited during a random walk



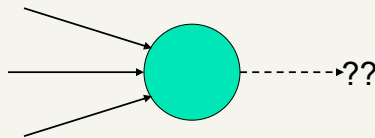
11/11/11

CSCI 5417 - IR

4

Not quite enough

- The web is full of dead-ends. Pages that are pointed to but have no outgoing links
 - Random walk can get stuck in such dead-ends
 - Makes no sense to talk about long-term visit rates in the presence of dead-ends.



11/11/11

CSCI 5417 - IR

5

Teleporting

- At a dead end, jump to a random web page
- At any non-dead end, with probability 10%, jump to a random web page
 - With remaining probability (90%), go out on a random link.
 - 10% - a parameter (call it alpha)

11/11/11

CSCI 5417 - IR

6

Result of teleporting

- Now you can't get stuck locally.
- There is a long-term rate at which any page is visited
- How do we compute this visit rate?
 - Can't directly use the random walk metaphor

11/11/11

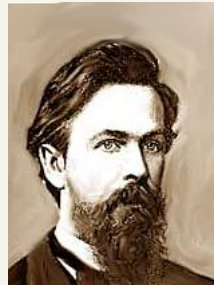
CSCI 5417 - IR

7

State Transition Probabilities

We're going to use the notion of a transition probability. If we're in some particular state, what is the probability of going to some other particular state from there.

If there are n states (pages) then we need an $n \times n$ table of probabilities.



11/11/11

CSCI 5417 - IR

8

Markov Chains

- So if I'm in a particular state (say the start of a random walk)
- And I know the whole $n \times n$ table
- Then I can compute the probability distribution over all the next states I might be in in the next step of the walk...
- And in the step after that
 - And the step after that

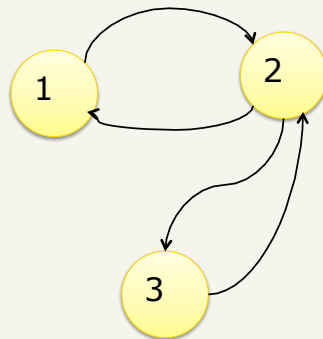
11/11/11

CSCI 5417 - IR

9

Example

- Say $\alpha = .5$



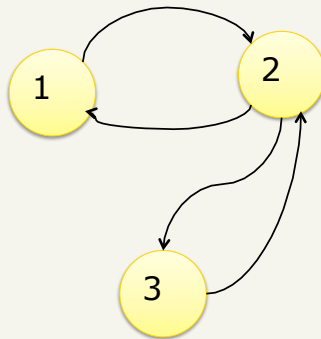
11/11/11

CSCI 5417 - IR

10

Example

- Say $\alpha = .5$



$P(3 \rightarrow 2)$

	?	

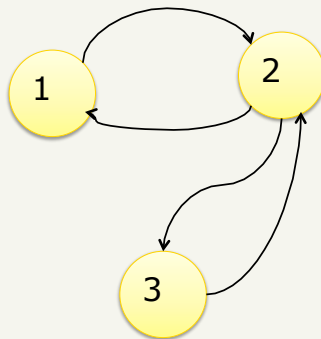
11/11/11

CSCI 5417 - IR

11

Example

- Say $\alpha = .5$



$P(3 \rightarrow 2)$

	2/3	

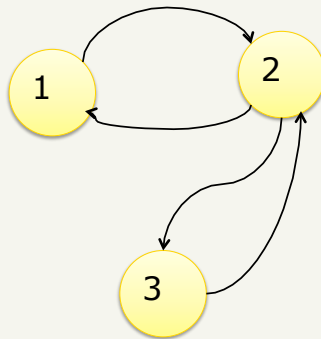
11/11/11

CSCI 5417 - IR

12

Example

- Say $\alpha = .5$



$P(3 \rightarrow^*)$

1/6	2/3	1/6

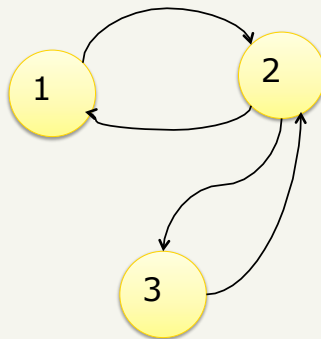
11/11/11

CSCI 5417 - IR

13

Example

- Say $\alpha = .5$



1/6	2/3	1/6
5/12	1/6	5/12
1/6	2/3	1/6

11/11/11

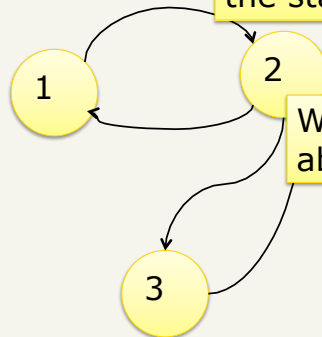
CSCI 5417 - IR

14

Example

- Say alpha = .5

Assume we start a walk in 1 at time T0. Then what should we believe about the state of affairs in T1?



What should we believe about things at T2?

1/6	2/3	1/6
5/12	1/6	5/12
1/6	2/3	1/6

11/11/11

CSCI 5417 - IR

15

Example

\vec{x}_0	1	0	0
\vec{x}_1	1/6	2/3	1/6
\vec{x}_2	1/3	1/3	1/3
\vec{x}_3	1/4	1/2	1/4
\vec{x}_4	7/24	5/12	7/24
...
\vec{x}	5/18	4/9	5/18

PageRank values

11/11/11

CSCI 5417 - IR

16

More Formally

- A probability (row) vector $x = (x_1, \dots, x_N)$ tells us where the random walk is at any point.

(0	0	0	...	1	...	0	0	0)
---	---	---	---	-----	---	-----	---	---	---	---
- Example

	1	2	3	...	i	...	$N-2$	$N-1$	N
							2	1	
- More generally: the random walk is on the page i with probability x_i .
- | | | | | | | | | | | |
|---|-----|-----|-----|-----|-----|-----|-------|-------|-----|---|
| (| 0.0 | 0.0 | 0.0 | ... | 0.2 | ... | 0.0 | 0.0 | 0.0 |) |
| | 5 | 1 | | | | | 1 | 5 | 3 | |
| | 1 | 2 | 3 | ... | i | ... | $N-2$ | $N-1$ | N | |
- $\sum x_i = 1$

17

Change in probability vector

- If the probability vector is $\vec{x} = (x_1, \dots, x_N)$, at this step, what is it at the next step?

18

Change in probability vector

- If the probability vector is $\vec{x} = (x_1, \dots, x_N)$, at this step, what is it at the next step?
- Recall that row i of the transition probability matrix P tells us where we go next from state i .

19

Change in probability vector

- If the probability vector is $\vec{x} = (x_1, \dots, x_N)$, at this step, what is it at the next step?
- Recall that row i of the transition probability matrix P tells us where we go next from state i .
- So from \vec{x} , our next state is distributed as $\vec{x}P$.

20

Steady state in vector notation

21

Steady state in vector notation

- The steady state in vector notation is simply a vector $\vec{\pi} = (\pi_1, \pi_2, \dots, \pi_N)$ of probabilities.

22

Steady state in vector notation

- The steady state in vector notation is simply a vector $\vec{\pi} = (\pi_1, \pi_2, \dots, \pi_N)$ of probabilities.
 - Use $\vec{\pi}$ to distinguish it from the notation for the probability vector \vec{x} .)

23

Steady state in vector notation

- The steady state in vector notation is simply a vector $\vec{\pi} = (\pi_1, \pi_2, \dots, \pi_N)$ of probabilities.
 - Use $\vec{\pi}$ to distinguish it from the notation for the probability vector \vec{x} .)
- π is the long-term visit rate (or PageRank) of page i .

24

Steady state in vector notation

- The steady state in vector notation is simply a vector $\vec{\pi} = (\pi_1, \pi_2, \dots, \pi_N)$ of probabilities.
 - Use $\vec{\pi}$ to distinguish it from the notation for the probability vector x .)
- π is the long-term visit rate (or PageRank) of page i .
- So we can think of PageRank as a very long vector – one entry per page.

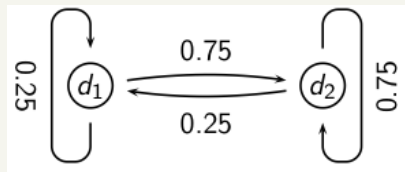
25

Steady-state distribution: Example

26

Steady-state distribution: Example

- What is the PageRank / steady state in this example?



27

Steady-state distribution: Example

28

Steady-state distribution: Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$	
			$P_{11} = 0.25$ $P_{12} = 0.75$ $P_{21} = 0.25$ $P_{22} = 0.75$
t_0	0.25	0.75	
t_1			

$$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$$

$$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$$

29

Steady-state distribution: Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$	
			$P_{11} = 0.25$ $P_{12} = 0.75$ $P_{21} = 0.25$ $P_{22} = 0.75$
t_0	0.25	0.75	0.25 0.75
t_1			

30

Steady-state distribution: Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$		
			$P_{11} = 0.25$	$P_{12} = 0.75$
			$P_{21} = 0.25$	$P_{22} = 0.75$
t_0	0.25	0.75	0.25	0.75
t_1	0.25	0.75		

PageRank vector $\vec{\pi} = \pi = (\pi_1, \pi_2) = (0.25, 0.75)$

$$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$$

$$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$$

31

Steady-state distribution: Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$		
			$P_{11} = 0.25$	$P_{12} = 0.75$
			$P_{21} = 0.25$	$P_{22} = 0.75$
t_0	0.25	0.75	0.25	0.75
t_1	0.25	0.75	(convergence)	

PageRank vector $\vec{\pi} = \pi = (\pi_1, \pi_2) = (0.25, 0.75)$

$$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$$

$$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$$

32

How do we compute the steady state vector?

33

How do we compute the steady state vector?

- In other words: how do we compute PageRank?

34

How do we compute the steady state vector?

- In other words: how do we compute PageRank?
- Recall: $\vec{\pi} = (\pi_1, \pi_2, \dots, \pi_N)$ is the PageRank vector, the vector of steady-state probabilities ...

35

How do we compute the steady state vector?

- In other words: how do we compute PageRank?
- Recall: $\vec{\pi} = (\pi_1, \pi_2, \dots, \pi_N)$ is the PageRank vector, the vector of steady-state probabilities ...
- ... and if the distribution in this step is \vec{x} , then the distribution in the next step is $\vec{x}P$.

36

How do we compute the steady state vector?

- In other words: how do we compute PageRank?
- Recall: $\vec{\pi} = (\pi_1, \pi_2, \dots, \pi_N)$ is the PageRank vector, the vector of steady-state probabilities ...
- ... and if the distribution in this step is \vec{x} , then the distribution in the next step is $\vec{x}P$.
- But $\vec{\pi}$ is the steady state!

37

How do we compute the steady state vector?

- In other words: how do we compute PageRank?
- Recall: $\vec{\pi} = (\pi_1, \pi_2, \dots, \pi_N)$ is the PageRank vector, the vector of steady-state probabilities ...
- ... and if the distribution in this step is \vec{x} , then the distribution in the next step is $\vec{x}P$.
- But $\vec{\pi}$ is the steady state!
- So: $\vec{\pi} = \vec{\pi} P$

38

How do we compute the steady state vector?

- In other words: how do we compute PageRank?
- Recall: $\vec{\pi} = (\pi_1, \pi_2, \dots, \pi_N)$ is the PageRank vector, the vector of steady-state probabilities ...
- ... and if the distribution in this step is \vec{x} , then the distribution in the next step is $\vec{x}P$.
- But $\vec{\pi}$ is the steady state!
- So: $\vec{\pi} = \vec{\pi} P$
- Solving this matrix equation gives us $\vec{\pi}$.

39

How do we compute the steady state vector?

- In other words: how do we compute PageRank?
- Recall: $\vec{\pi} = (\pi_1, \pi_2, \dots, \pi_N)$ is the PageRank vector, the vector of steady-state probabilities ...
- ... and if the distribution in this step is \vec{x} , then the distribution in the next step is $\vec{x}P$.
- But $\vec{\pi}$ is the steady state!
- So: $\vec{\pi} = \vec{\pi} P$
- Solving this matrix equation gives us $\vec{\pi}$.
- $\vec{\pi}$ is the principal left eigenvector for P ...

40

How do we compute the steady state vector?

- In other words: how do we compute PageRank?
- Recall: $\vec{\pi} = (\pi_1, \pi_2, \dots, \pi_N)$ is the PageRank vector, the vector of steady-state probabilities ...
- ... and if the distribution in this step is \vec{x} , then the distribution in the next step is $\vec{x}P$.
- But $\vec{\pi}$ is the steady state!
- So: $\vec{\pi} = \vec{\pi} P$
- Solving this matrix equation gives us $\vec{\pi}$.
- $\vec{\pi}$ is the principal left eigenvector for P ...
 - That is, $\vec{\pi}$ is the left eigenvector with the largest eigenvalue.

41

One way of computing the PageRank $\vec{\pi}$

42

One way of computing the PageRank $\vec{\pi}$

- Start with any distribution \vec{x} , e.g., uniform distribution

43

One way of computing the PageRank $\vec{\pi}$

- Start with any distribution \vec{x} , e.g., uniform distribution
- After one step, we're at $\vec{x}P$.

44

One way of computing the PageRank $\vec{\pi}$

- Start with any distribution \vec{x} , e.g., uniform distribution
- After one step, we're at $\vec{x}P$.
- After two steps, we're at $\vec{x}P^2$.

45

One way of computing the PageRank $\vec{\pi}$

- Start with any distribution \vec{x} , e.g., uniform distribution
- After one step, we're at $\vec{x}P$.
- After two steps, we're at $\vec{x}P^2$.
- After k steps, we're at $\vec{x}P^k$.

46

One way of computing the PageRank $\vec{\pi}$

- Start with any distribution \vec{x} , e.g., uniform distribution
- After one step, we're at $\vec{x}P$.
- After two steps, we're at $\vec{x}P^2$.
- After k steps, we're at $\vec{x}P^k$.
- Algorithm: multiply \vec{x} by increasing powers of P until convergence.

47

One way of computing the PageRank $\vec{\pi}$

- Start with any distribution \vec{x} , e.g., uniform distribution
- After one step, we're at $\vec{x}P$.
- After two steps, we're at $\vec{x}P^2$.
- After k steps, we're at $\vec{x}P^k$.
- Algorithm: multiply \vec{x} by increasing powers of P until convergence.
- This is called the **power method**.

48

One way of computing the PageRank $\vec{\pi}$

- Start with any distribution \vec{x} , e.g., uniform distribution
- After one step, we're at $\vec{x}P$.
- After two steps, we're at $\vec{x}P^2$.
- After k steps, we're at $\vec{x}P^k$.
- Algorithm: multiply \vec{x} by increasing powers of P until convergence.
- This is called the **power method**.
- Recall: regardless of where we start, we eventually reach the steady state $\vec{\pi}$.

49

One way of computing the PageRank $\vec{\pi}$

- Start with any distribution \vec{x} , e.g., uniform distribution
- After one step, we're at $\vec{x}P$.
- After two steps, we're at $\vec{x}P^2$.
- After k steps, we're at $\vec{x}P^k$.
- Algorithm: multiply \vec{x} by increasing powers of P until convergence.
- This is called the **power method**.
- Recall: regardless of where we start, we eventually reach the steady state $\vec{\pi}$.
- Thus: we will eventually reach the steady state.

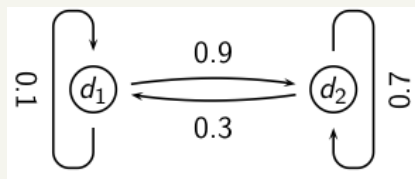
50

Power method: Example

51

Power method: Example

- What is the PageRank / steady state in this example?



52

Computing PageRank: Power Example

53

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$	
			$P_{11} = 0.1$ $P_{12} =$ $P_{21} = 0.3$ 0.9 $P_{22} =$ 0.7
t_0	0	1	\rightarrow
t_1			\rightarrow
t_2			\rightarrow
t_3			\rightarrow
t_4			\rightarrow
t_∞			\rightarrow

$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$
 $P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$

54

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	\rightarrow \rightarrow
t_0	0	1	0.3	0.7	$= xP$
t_1					$= xP^2$
t_2					$= xP^3$
t_3					$= xP^4$
					\dots
t_∞	$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$ $P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$				$= xP^\infty$

55

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	\rightarrow \rightarrow
t_0	0	1	0.3	0.7	$= xP$
t_1	0.3	0.7			$= xP^2$
t_2					$= xP^3$
t_3					$= xP^4$
					\dots
t_∞	$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$ $P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$				$= xP^\infty$

56

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	\rightarrow \rightarrow
t_0	0	1	0.3	0.7	$= xP$
t_1	0.3	0.7	0.24	0.76	$= xP^2$
t_2					$= xP^3$
t_3					$= xP^4$
					\dots
t_∞					$= xP^\infty$
	$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$ $P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$				

57

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	\rightarrow \rightarrow
t_0	0	1	0.3	0.7	$= xP$
t_1	0.3	0.7	0.24	0.76	$= xP^2$
t_2	0.24	0.76			$= xP^3$
t_3					$= xP^4$
					\dots
t_∞					$= xP^\infty$
	$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$ $P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$				

58

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	\rightarrow \rightarrow
t_0	0	1	0.3	0.7	$= xP$
t_1	0.3	0.7	0.24	0.76	$= xP^2$
t_2	0.24	0.76	0.252	0.748	$= xP^3$
t_3					$= xP^4$
					\dots
t_∞					$= xP^\infty$
	$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$ $P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$				

59

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	\rightarrow \rightarrow
t_0	0	1	0.3	0.7	$= xP$
t_1	0.3	0.7	0.24	0.76	$= xP^2$
t_2	0.24	0.76	0.252	0.748	$= xP^3$
t_3	0.252	0.748			$= xP^4$
					\dots
t_∞					$= xP^\infty$
	$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$ $P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$				

60

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	\rightarrow \rightarrow
t_0	0	1	0.3	0.7	$= xP$
t_1	0.3	0.7	0.24	0.76	$= xP^2$
t_2	0.24	0.76	0.252	0.748	$= xP^3$
t_3	0.252	0.748	0.2496	0.7504	$= xP^4$
					\dots
t_∞	$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$ $P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$				$= xP^\infty$

61

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	\rightarrow \rightarrow
t_0	0	1	0.3	0.7	$= xP$
t_1	0.3	0.7	0.24	0.76	$= xP^2$
t_2	0.24	0.76	0.252	0.748	$= xP^3$
t_3	0.252	0.748	0.2496	0.7504	$= xP^4$
					\dots
t_∞	$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$ $P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$				$= xP^\infty$

62

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	\rightarrow \rightarrow
t_0	0	1	0.3	0.7	$= xP$
t_1	0.3	0.7	0.24	0.76	$= xP^2$
t_2	0.24	0.76	0.252	0.748	$= xP^3$
t_3	0.252	0.748	0.2496	0.7504	$= xP^4$
		
t_p	0.25	0.75	0.25	0.75	$= xP^\infty$
	$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$ $P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$				

63

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	\rightarrow \rightarrow
t_0	0	1	0.3	0.7	$= xP$
t_1	0.3	0.7	0.24	0.76	$= xP^2$
t_2	0.24	0.76	0.252	0.748	$= xP^3$
t_3	0.252	0.748	0.2496	0.7504	$= xP^4$
		
t_p	0.25	0.75	0.25	0.75	$= xP^\infty$
	$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$ $P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$				

64

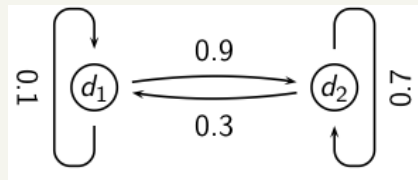
Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	\rightarrow \rightarrow
t_0	0	1	0.3	0.7	$= \vec{x}P$
t_1	0.3	0.7	0.24	0.76	$= \vec{x}P^2$
t_2	0.24	0.76	0.252	0.748	$= \vec{x}P^3$
t_3	0.252	0.748	0.2496	0.7504	$= \vec{x}P^4$
		\rightarrow		\dots	\dots
t_∞	0.25	0.75	0.25	0.75	$= \vec{x}P^\infty$

65

Power method: Example

- What is the PageRank / steady state in this example?



- The steady state distribution (= the PageRanks) in this example are 0.25 for d_1 and 0.75 for d_2 .

66

PageRank summary

- Preprocessing
 - Given graph of links, build matrix P
 - Apply teleportation
 - From modified matrix, compute $\vec{\pi}$
 - $\vec{\pi}_i$ is the PageRank of page i .
- Query processing
 - Retrieve pages satisfying the query
 - Rank them by their PageRank
 - Return reranked list to the user

67

PageRank issues

- Real surfers are not random surfers.
 - Examples of nonrandom surfing: back button, bookmarks, directories, tabs, search, interruptions
 - → Markov model is not a good model of surfing.
 - But it's good enough as a model for our purposes.
- Simple PageRank ranking produces bad results for many pages.

68

How important is PageRank?

- Frequent claim: PageRank is the most important component of Google's web ranking
- The reality:
 - There are several components that are at least as important: e.g., anchor text, phrases, proximity, tiered indexes ...
 - Rumor has it that PageRank in his original form (as presented here) now has a negligible impact on ranking!
 - However, variants of a page's PageRank are still an essential part of ranking.
 - Addressing link spam is difficult and crucial.

69

Break

- Today's colloquium is relevant to the current material

11/11/11

CSCI 5417 - IR

70

Machine Learning for ad hoc IR

- We've looked at methods for ranking documents in IR using factors like
 - Cosine similarity, inverse document frequency, pivoted document length normalization, Pagerank, etc.
- We've looked at methods for classifying documents using supervised machine learning classifiers
 - Naïve Bayes, kNN, SVMs
- Surely we can also use such **machine learning to rank the documents** displayed in search results?

11/11/11

CSCI 5417 - IR

71

Why is There a Need for ML?

- Traditional ranking functions in IR used a very small number of features
 - Term frequency
 - Inverse document frequency
 - Document length
- It was easy to tune weighting coefficients by hand
 - And people did
 - But you saw how "easy" it was on HW1

11/11/11

CSCI 5417 - IR

72

Why is There a Need for ML

- Modern systems – especially on the Web – use a large number of features:
 - Log frequency of query word in anchor text
 - Query term proximity
 - Query word in color on page?
 - # of images on page
 - # of (out) links on page
 - PageRank of page?
 - URL length?
 - URL contains “~”?
 - Page edit recency?
 - Page length?
- The *New York Times* (2008-06-03) quoted Amit Singhal as saying Google was using over 200 such features.

11/11/11

CSCI 5417 - IR

73

Using ML for ad hoc IR

- Well classification seems like a good place to start
 - Take an object and put it in a class
 - With some confidence
 - What do we have to work with in terms of training data?
 - Documents
 - Queries
 - Relevance judgements

11/11/11

CSCI 5417 - IR

74

Using Classification for ad hoc IR

- Collect a training corpus of (q, d, r) triples
 - Relevance r is here binary
 - Documents are represented by a feature vector
 - Say 2 features just to keep it simple
 - Cosine sim score between doc and query
 - Note this hides a bunch of "features" inside the cosine (tf, idf, etc.)
 - Minimum window size around query words in the doc
 - Train a machine learning model to predict the class r of each document-query pair
 - Where class is relevant/non-relevant
 - Then use classifier confidence to generate a ranking

11/11/11

CSCI 5417 - IR

75

Training data

example	docID	query	cosine score	ω	judgment
Φ_1	37	linux operating system	0.032	3	<i>relevant</i>
Φ_2	37	penguin logo	0.02	4	<i>nonrelevant</i>
Φ_3	238	operating system	0.043	2	<i>relevant</i>
Φ_4	238	runtime environment	0.004	2	<i>nonrelevant</i>
Φ_5	1741	kernel layer	0.022	3	<i>relevant</i>
Φ_6	2094	device driver	0.03	2	<i>relevant</i>
Φ_7	3191	device driver	0.027	5	<i>nonrelevant</i>

11/11/11

CSCI 5417 - IR

76

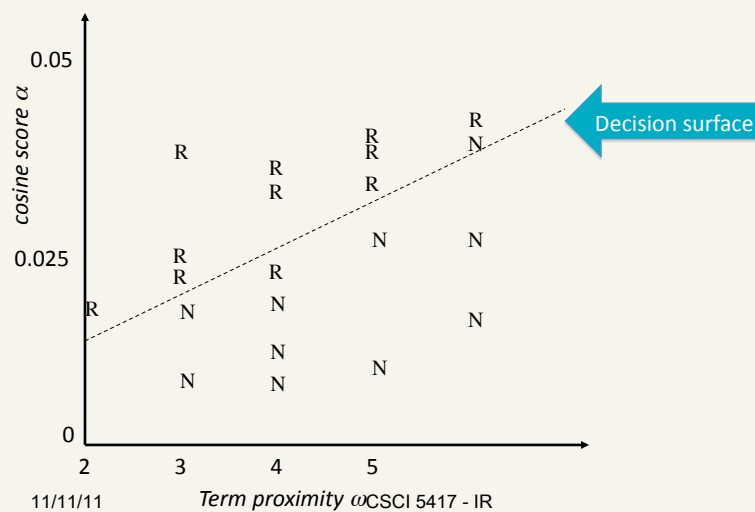
Using classification for ad hoc IR

- A linear scoring function on these two features is then

$$\text{Score}(d, q) = \text{Score}(a, \omega) = aa + b\omega + c$$
- And the linear classifier is

$$\text{Decide relevant if } \text{Score}(d, q) > \theta$$
- ... just like when we were doing text classification

Using classification for ad hoc IR



More Complex Cases

- We can generalize this to classifier functions over more features
- We can use any method we have for learning the linear classifier weights

11/11/11

CSCI 5417 - IR

79

An SVM Classifier for IR [Nallapati 2004]

- Experiments:
 - 4 TREC data sets
 - Comparisons done with Lemur, another state-of-the-art open source IR engine (LM)
 - Linear kernel normally best or almost as good as quadratic kernel
 - 6 features, all variants of tf, idf, and tf.idf scores

11/11/11

CSCI 5417 - IR

80

An SVM Classifier for IR [Nallapati 2004]

Train \ Test		Disk 3	Disk 4-5	WT10G (web)
Disk 3	LM	0.1785	0.2503	0.2666
	SVM	0.1728	0.2432	0.2750
Disk 4-5	LM	0.1773	0.2516	0.2656
	SVM	0.1646	0.2355	0.2675

- At best, the results are about equal to LM
 - Actually a little bit below

11/11/11

CSCI 5417 - IR

81

An SVM Classifier for IR [Nallapati 2004]

- Paper's advertisement: Easy to add more features
- Especially for specialized tasks
 - Homepage finding task on WT10G:
 - Baseline LM 52% success@10, baseline SVM 58%
 - SVM with URL-depth, and in-link features: 78% S@10

11/11/11

CSCI 5417 - IR

82

Problem

- The ranking in this approach is based on the classifier's confidence in its judgment
- It's not clear that that should directly determine a ranking between two documents
 - That is, it gives a ranking of confidence not a ranking of relevance
 - Maybe they correlate, maybe not

11/11/11

CSCI 5417 - IR

83

Learning to Rank

- Maybe classification isn't the right way to think about approaching ad hoc IR via ML
- Background ML
 - Classification problems
 - Map to a discrete unordered set of classes
 - Regression problems
 - Map to a real value
 - Ordinal regression problems
 - Map to an *ordered* set of classes

11/11/11

CSCI 5417 - IR

84

Learning to Rank

- Assume documents can be totally ordered by relevance given a query
 - These are totally ordered: $d_1 < d_2 < \dots < d_j$
 - This is the ordinal regression setup
- Assume training data is available consisting of document-query pairs represented as feature vectors ψ_i and a relevance ranking between them
- Such an ordering can be cast as a set of **pair-wise judgements**, where the input is a pair of results for a single query, and the class is the relevance ordering relationship between them

11/11/11

CSCI 5417 - IR

85

Learning to Rank

- But assuming a total ordering across all docs is a lot to expect
 - Think of all the training data
- So instead assume a smaller number of categories **C** of relevance exist
 - These are totally ordered: $c_1 < c_2 < \dots < c_j$
 - Definitely rel, relevant, partially, not relevant, really really not relevant... Etc.
 - Indifferent to differences within a category
- Assume training data is available consisting of document-query pairs represented as feature vectors ψ_i and relevance ranking based on the categories **C**

11/11/11

CSCI 5417 - IR

86

Experiments

Based on the LETOR test collection (Cao et al)

- An openly available standard test collection with pregenerated features, baselines, and research results for learning to rank
 - OHSUMED, MEDLINE subcollection for IR
 - 350,000 articles
 - 106 queries
 - 16,140 query-document pairs
 - 3 class judgments: Definitely relevant (DR), Partially Relevant (PR), Non-Relevant (NR)

11/11/11

CSCI 5417 - IR

87

Experiments

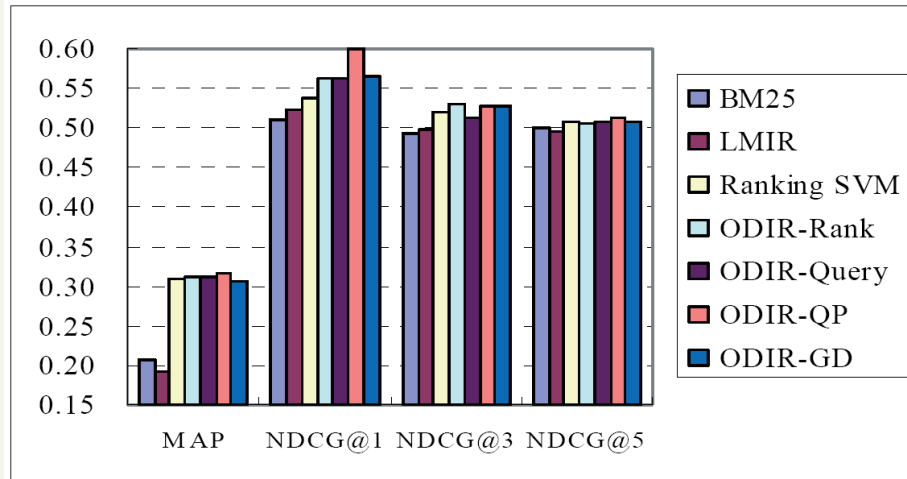
-
- OHSUMED (from LETOR)
 - Features:
 - 6 that represent versions of tf, idf, and tf.idf factors
 - BM25 score (*IIR* sec. 11.4.3)
 - A scoring function derived from a probabilistic approach to IR, which has traditionally done well in TREC evaluations, etc.

11/11/11

CSCI 5417 - IR

88

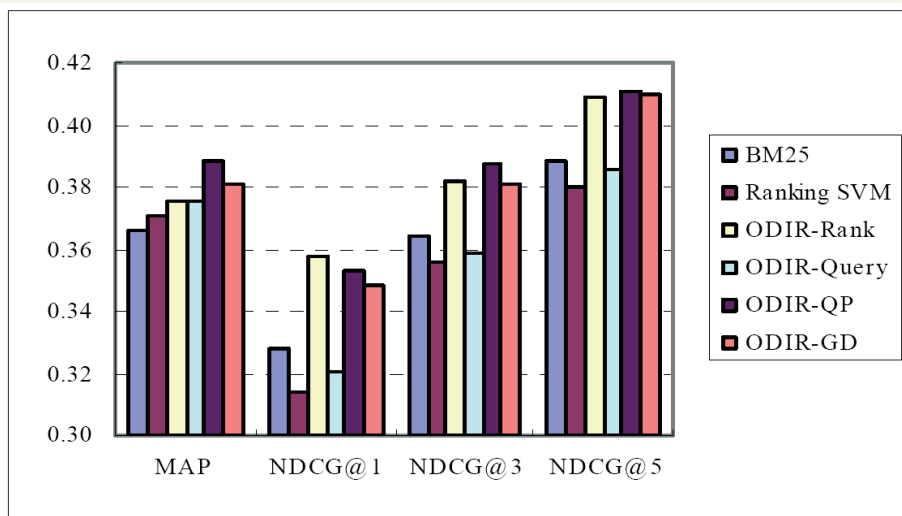
Experimental Results (OHSUMED)



MSN Search

- Second experiment with MSN search
- Collection of 2198 queries
- 6 relevance levels rated:
 - Definitive 8990
 - Excellent 4403
 - Good 3735
 - Fair 20463
 - Bad 36375
 - Detrimental 310

Experimental Results (MSN search)



Limitations of Machine Learning

- Everything that we have looked at (and most work in this area) produces *linear* models of features by weighting different base features
- This contrasts with most of the clever ideas of traditional IR, which are *nonlinear* scalings and combinations of basic measurements
 - log term frequency, idf, pivoted length normalization
- At present, ML is good at weighting features, but not at coming up with nonlinear scalings
 - Designing the basic features that give good signals for ranking remains the domain of human creativity

Summary

- Machine learned ranking over many features now easily beats traditional hand-designed ranking functions in comparative evaluations
- And there is every reason to think that the importance of machine learning in IR will only increase in the future.