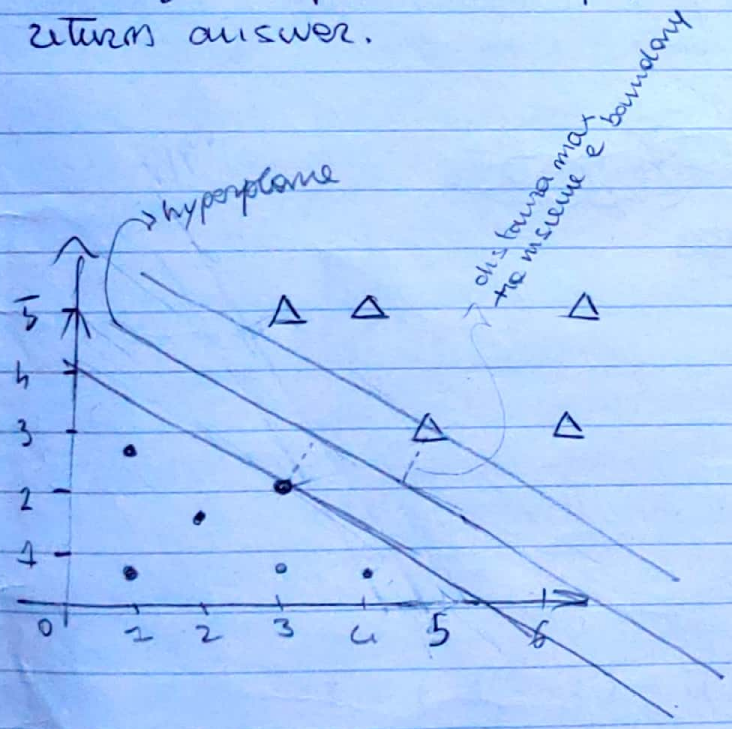
EXAM 11/02/19

1) INTERSECT (p1, p2) postings lists T_1 AND T_2

```
answer ← ()
while p1 ≠ NIL and p2 ≠ NIL
do if docID(p1) = docID(p2)
     then ADD(answer, docID(p1))
          p1 ← next(p1)
          p2 ← next(p2)
     else if docID(p1) < docID(p2)
          then p1 ← next(p1)
          else p2 ← next(p2)
return answer.
```

2)



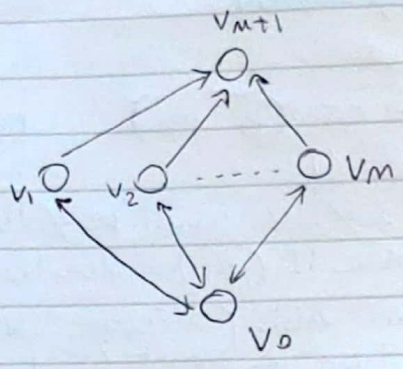
Identify the linear maximum margin classifier. Draw 3 lines: the two boundaries and the hyperplane which of the vectors are support vector?

- Scelgo gli elementi che si trovano più vicini tra loro e creo un iperpiano in modo da massimizzare la distanza.

devo massimizzare la distanza tra 1 punto di una classe e il decision boundary.

I support vector sono (3,3) e (4,5,3)

3) Pagerank equation
teleporting probability α
personalization vector $\frac{1}{n}$



$\sum_{i=0\ldots M+1} \pi_i = 1$

$\pi_0 = \frac{\alpha}{M} + \frac{(1-\alpha)}{2} \sum_{i=1}^{M} \pi_i$

$\pi_i = \frac{\alpha}{M} + \frac{(1-\alpha)}{M} \pi_0 \quad i = 1 \ldots M$

$\pi_{M+1} = \frac{\alpha}{M} + \frac{(1-\alpha)}{2} \sum_{i=1}^{M} \pi_i$

(b) personalize vector $p = (0 \ldots 1)^T$ all components are 0 except for v_{M+1}. Compute pagerank π. Motivate the answer

$\pi = \alpha p_i + (1-\alpha) \sum_i \frac{\pi_i}{d_i}$

$$\begin{cases} \pi_0 = \alpha \cdot p_i + \frac{(1-\alpha)}{2} \sum_{i=1}^{M} \pi_i \\ \pi_i = \alpha \cdot p_i + \frac{(1-\alpha)}{M} \pi_0 \\ \pi_{M+1} = \alpha \cdot p_i + \frac{(1-\alpha)}{M} \sum_{i=1}^{M} \pi_i \end{cases}$$

$$\begin{cases} \pi_0 = \frac{(1-\alpha)}{2} \sum_{i=1}^{M} \pi_i \\ \pi_i = \frac{(1-\alpha)}{2} \pi_0 \\ \pi_{M+1} = \alpha + \frac{(1-\alpha)}{2} \sum_{i=1}^{M} \pi_i \end{cases}$$

formula generale $\pi = (1-\alpha)M + \alpha I_p$ → matrice identità

EXAM 12/04/19

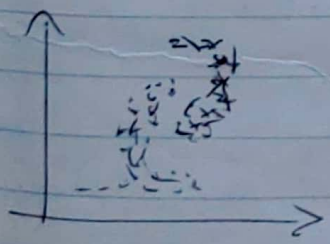
1) UNION (p_1, p_2) postings lists T_1 OR T_2

```
answer ← ()
while p1 ≠ NIL and p2 ≠ NIL
do if docID(p1) != docID(p2)
     if docID(p1) < docID(p2)
        Then ADD(Answer, docID(p1))
        p1 ← next(p1)
     else if docID(p1) > docID(p2)
        Then ADD(answer, docID(p2))
        p2 ← next(p2)
  if docID(p1) = docID(p2)
     Then ADD(answer, docID(p1))
     p1 ← next(p1)
     p2 ← next(p2)

if p1 = NIL
   while (p2 != NIL)
      then ADD(answer, docID(p2))
if p2 = NIL
   while (p1 != NIL)
      then ADD(answer, docID(p1))

return answer
```

nel caso finisse una delle liste

---

2) Better SVM or KNN ?



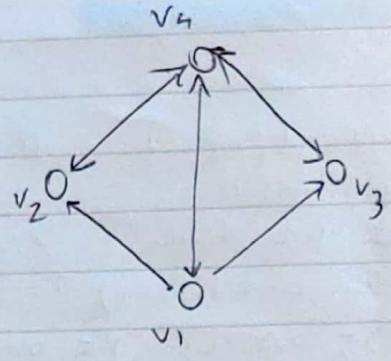
Linear SVM is not good and soft margin technique is not sufficient because dataset is not linearly separable.

So in this case KNN is better.

However we can use kernelization to make this set linearly separable and then use SVM.

3) teleporting prob. α

personalization vector $\{1, 0, 0, 0\}$



$$\sum_i \pi_i = 1 \quad i = 1 \dots 4$$

$\pi_2 = \pi_3$ simmetry

$$\pi_1 = \alpha + \frac{(1-\alpha)}{3}\pi_4$$

$$\pi_2 = \frac{(1-\alpha)}{3}\pi_1 + \frac{(1-\alpha)}{3}\pi_4$$

$$\pi_4 = \frac{(1-\alpha)}{3}\pi_1 + (1-\alpha)\pi_2 + (1-\alpha)\pi_3$$

(b) $\pi_1 \quad p_1 = \{1, 0, 0, 0\}^T$

$\pi_2 \quad p_2 = \{0, 1, 0, 0\}^T$

Explain in details how to calculate the personalized pagerank with respect to the personalization vector $\{0.5, 0.5, 0, 0\}$

$$\pi = (1-\alpha)\sum_{j \to i} \frac{\pi_j}{\text{out}_j} + \alpha \cdot p_i$$

$$\pi_1 = \frac{\alpha}{2} + \frac{(1-\alpha)}{3}\pi_4$$

$$\pi_2 = \frac{\alpha}{2} + \frac{(1-\alpha)}{3}(\pi_1 + \pi_4)$$

$$\pi_4 = \frac{(1-\alpha)}{3}\pi_1 + (1-\alpha)(\pi_2 + \pi_3)$$

$$\pi_2 = \pi_3$$

$p_3 = \{\frac{1}{2}, \frac{1}{2}, 0, 0\}$

$$\pi_1(p_3) = \pi_1(p_1) - \frac{\alpha}{2}$$

$$\pi_2(p_3) = \pi_2(p_2) - \frac{\alpha}{2}$$

$$\pi_3(p_3) = \pi_3(p_1) = \pi_3(p_2)$$

$$\pi_4(p_3) = \pi_4(p_1) = \pi_4(p_1)$$

You have to suppose that you can jump with probability 1 to node v_1 or probability 1 to node v_2

If we have $p = \{\frac{1}{2}, \frac{1}{2}, 0, 0\}$ it means that we have to ~~subtract~~ subtract $\frac{\alpha}{2}$ to π_1 and π_2

$$\pi_1 = \pi_1 - \frac{\alpha}{2}$$

$$\pi_2 = \pi_2 - \frac{\alpha}{2}$$

~~thus~~, we have

$$v_{NB} = \text{argmax}\, P(v_j) \prod P_i(a_i / v_j)$$

assuming that the attribute values are conditionally independent given the target value, it is possible to notice that the probability of observing the conjunction $a_1 \dots a_n$ is just the product of the probability for the individual attribute. Substituting

EXAM 03/11/20

2) → (Yes, Yes, No)

Which class (Fail or Pass) will be assigned to X by Naive Bayes?

$$V_{NB} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_i P(a_i / v_j)$$

(a_i: attributi; $P(v_j)$: probabilità di un evento)

$P(Fail) = 2/5$

$P(Pass) = 3/5$

- probabilità (Yes, Yes, No / P(Fail))
  - $P(Confident = Yes / P(Fail)) = 1/2$
  - $P(Studied = yes / P(Fail)) = 1/2$
  - $P(Sick = No / P(Fail)) = 1/2$

$\Rightarrow P(Fail) \cdot \prod_i P(a_i / Fail) =$

$P(Result = Fail) \cdot P(Confident = Yes / Result = Fail) \cdot P(Studied = Yes / Result = Fail) \cdot P(Sick = No / Result = Fail) =$

$= 2/5 \cdot 1/2 \cdot 1/2 \cdot 1/2 = \frac{1}{20} \approx 0{,}05$

- probabilità (Yes, Yes, No / P(Pass))

  $P(Confident = Yes / Result = Pass) = 2/3$

  $P(Studied = yes / Result = Pass) = 2/3$

  $P(Sick = No / Result = Pass) = 1/3$

$P(Result = Pass) \cdot P(Confident = Yes / Result = Pass) \cdot P(Studied = Yes / Result = Pass) \cdot P(Sick = No / Result = Pass)$

$= 3/5 \cdot 2/3 \cdot 2/3 \cdot 1/3 = \frac{4}{45} \approx 0{,}08$

Quindi dato Yes, Yes, NO, la probabilità più alta è che passi poiché bisogna considerare argmax

(i valori degli attributi sono conditionally independent. In questo modo posso vedere la prob. degli attributi come il prodotto dei singoli attrib.)

3) Indicate where the conditional independence assumption

The Bayesian approach to classify a new instance is to assign the most probable target value v_{map} given the attribute value $a_{1\ldots n}$ that describe the instance. $v_{map} = \operatorname{argmax} P(v_j$

by using the bayes theorem, we can rewrite the previous equation as

$v_{map} = \operatorname{argmax} P(v_j) \cdot P(a_{1\ldots n} / v_j)$

1)

$\frac{N}{df}$ = $\frac{\text{number of docs}}{\text{number of docs in which term occurs}}$

DOC	T	D2	TF = $\frac{\text{numb. words}}{\text{in document}}$	IDF	TF/IDF
D1	aruccia		0	$\frac{N}{df} = \frac{5}{2}$	
	dish		0		
	pork		3		
	rabbit		0		
	recipe		0		
	roast		0		
D2					

T	TF_{D1}	TF_{D2}	TF_{D3}	TF_{D4}	TF_{D5}
aruccia	0	0	3	0	0
dish					
pork					
rabbit					
recipe					
roast					

Vedi foto Daniele

$$Q = \begin{pmatrix} 1 \\ 1/3 \\ 1/3 \\ 1/3 \end{pmatrix}$$

	anna	pork	recipe
D_1	0	$\frac{15}{64}$	0
D_2	$\frac{5}{42}$	$\frac{10}{84}$	0
D_3	0	$\frac{10}{212}$	$\frac{5}{106}$
D_4	0	0	$\frac{5}{30}$
D_5	$\frac{5}{80}$	$\frac{5}{160}$	0

$$D_1 = \begin{pmatrix} 0 \\ 15/64 \\ 0 \end{pmatrix} \quad D_2 = \begin{pmatrix} 5/42 \\ 10/84 \\ 0 \end{pmatrix} \quad D_3 = \begin{pmatrix} 0 \\ 10/212 \\ 5/106 \end{pmatrix} \quad D_4 = \begin{pmatrix} 0 \\ 0 \\ 5/30 \end{pmatrix} \quad D_5 = \begin{pmatrix} 5/80 \\ 5/160 \\ 0 \end{pmatrix}$$

$D_1 = \frac{15}{64}$ $D_2 = \frac{20}{84}$ $D_3 = \frac{10}{212}$ $D_4 = \frac{5}{30}$ $D_5 = \frac{15}{160}$

0,23 0,23 0,09 0,15 0,09

cos. sim (Q, D_1)

$$= \frac{\frac{1}{3}\cdot 0 + \frac{1}{3}\cdot\frac{15}{64} + \frac{1}{3}\cdot 0}{\sqrt{\left(\frac{15}{64}\right)^2}\cdot\sqrt{\left(\frac{1}{3}\right)^2+\left(\frac{1}{3}\right)^2+\left(\frac{1}{3}\right)^2}} = \frac{\frac{5}{64}}{\frac{15}{64}\cdot\frac{\sqrt{3}}{3}} = \frac{5}{64}\cdot\frac{64}{5\sqrt{3}} = \frac{1}{\sqrt{3}} \approx 0{,}36$$

(Q, D_2)

$$= \frac{\frac{1}{3}\cdot\frac{5}{42} + \frac{1}{3}\cdot\frac{10}{84}}{\frac{\sqrt{3}}{3}\cdot\sqrt{\left(\frac{5}{42}\right)^2+\left(\frac{10}{84}\right)^2}} = \frac{\frac{10+10}{252}}{\frac{\sqrt{3}\cdot 0{,}02}{3}} = \frac{20}{252}\cdot\frac{3}{\sqrt{3}\cdot 0{,}02}$$

$$\frac{\sqrt{3}}{3}\sqrt{0{,}01+0{,}01}$$

$0{,}27 \cdot 18{,}4 = 4{,}92$

un grafo diretto é fortemente connesso se esiste un percorso per arrivare in ogni nodo (se ogni nodo del grafo é raggiungibile da ogni altro)

3) Unique stat. dist? ?

(b) Siccome il grafo bipartito e' un grafo con un numero finito di stati, allora esiste almeno ~~uno~~ stationary distribution. Tuttavia, siccome il grafo e non diretto e ~~non connesso~~ non é fortemente connesso, la stationary distrib. non é unica

Non é fortemente connesso perché FGFR3 é collegato a Bladder e Colon ma questi sono solo collegati a FGFR3, quindi non arrivano in nessun altro nodo
(nemmeno se fosse stato orientato)

5) Questa struttura é per hubs e authorities
Quindi prendo alg. di questo genere

HUBS and AUTHORITIES (ESEMPIO INTERNET)

measure of importance similar to pageRank.
Uses both outlinks and incoming links.

↳ PUNTA
HUB: pagine che punta a un'altra pagina
AUTHORITY: pagine puntata da un'altra pagina
↳ È PUNTATO

AUTHORITY SCORE: somma degli hub score delle pagine che puntano alla pagina.

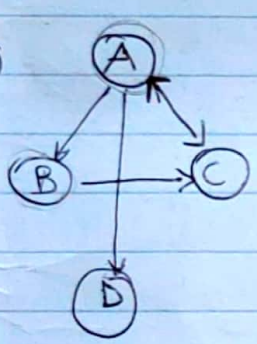
$$x_i^{(k)} = \sum_{j:e_{ji} \in E} y_j^{(k-1)}$$

~~HUB SCORE~~:

HUB SCORE: somma degli authority score delle pagine a cui punta

$$y_i^{(k)} = \sum_{j:e_{ij} \in E} x_j^{(k)}$$

es



A è hub di B D C
B è hub di C
C è hub di A

A è authority di C
B è authority di A
C è authority di A e B
D è authority di A

hub → \ authority ↓	A	B	C	D
A	0	1	1	1
B	0	0	1	0
C	1	0	0	0
D	0	0	0	0

k = 0

$y_i = 1$

k = 1

$$x_i^{(k)} = \sum_{j:e_{ji} \in E} y_j^{(k-1)} \Rightarrow x_i = \sum_{j:e_{ji} \in E} y_j^0$$

(quando quando gli hub conto gli authorities)

authorities

$$x_A = \sum_{C:e_{CA} \in E} y_C^0 = 1$$

auth. x

$$x_B = \sum_{A:e_{AB} \in E} y_A^0 = 1$$

$$x_C = \sum_{AB:e_{AC/BC}} y_j^0 = y_A^0 + y_B^0 = 1+1 = 2$$

$$x_D = \sum_{A:e_{AD} \in E} y_D^0 = 1$$

normalizziamo i valori al massimo valore
$1/2 = 0,5$
$1/2 = 0,5$
$2/2 = 1$
$1/2 = 1,5$

hub y

$$y_i^1 = \sum_{j:e_{ij} \in E} x_j^{(k)}$$

$y_A = x_B' + x_D' + x_C' = 0,5 + 0,5 + 1 = 2$
$y_B = x_C' = 1$
$y_C = x_A' = 0,5$

$2/2 = 1$
$1/2 = 0,5$
$0,5/2 = 0,25$
$0/2 = 0$

$k=2$ $\quad x_i^{(k)} = \sum y_j^{(k-1)} \longrightarrow x_i^{2} = \sum y_j^{1}$

X
$x_A = y_C = 0,25$
$x_B = y_A = 1$
$x_C = y_A + y_B = 1 + 0,5 = 1,5$
$x_D = y_A = 1$

normalizzo
$x_A = \frac{0,25}{1,5} = 0,16$
$x_B = \frac{1}{1,5} = 0,66$
$x_C = \frac{1,5}{1,5} = 1$
$x_D = 1/1,5 = 0,66$

Y
$y_i^{(k)} = \sum x_j^{(k)}$ → inoltre
$y_A = x_B^{(2)} + x_D^{(2)} + x_C^{(2)} = 0,66 + 0,66 + 1 = 2,32$
$y_B = x_C^{(2)} = 1$
$y_C = x_A^{(2)} = 0,16$

$y_A = \frac{1,87}{1,87} = 1$
$y_B = 0,43$
$y_C = 0,06$
$y_D = 0$

→ questa operazione converge quando tutti i valori si trovano sotto un valore fissato inizialmente
(ad ogni step si continua)

hub = pagina che punta a un'altra pagina

authority = pagina che viene puntata

EXAM 23/01/20

3) HITS algorithm con adjacency matrix A
a(t) and h(t) are authorities and hubs

(a) Indichiamo con E l'insieme di tutti i collegamenti diretti nel grafo che rappresenta la rete e chiamiamo e_{ij} il collegamento dal nodo i al nodo j.
A ciascuna pagina sono associati authority score $a_i^{(0)}$ e hub score $h_i^{(0)}$. In particolare $h_i^{(0)}$ viene inizializzato a 1.
Successivamente HITS affina i punteggi calcolando

$$a_i^{(k)} = \sum_{j:e_{ij} \in E} h_j^{(k-1)} \qquad h_i^{(k)} = \sum_{j:e_{ij} \in E} a_j^{(k)} \qquad k = 1, 2, 3 \dots \text{ iterazioni}$$

possiamo scrivere queste formule in forma matriciale con la matrice di adiacenza L

$$L_{ij} = \begin{cases} 1 \text{ se c'è un link dalla pagina } i \text{ a } j \\ 0 \text{ altrimenti} \end{cases}$$

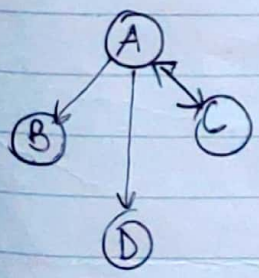
$$a^{(k)} = L^T h^{(k-1)} \qquad h^{(k)} = L \cdot a^{(k)}$$

1) Inizialmente $h^{(0)}$ è un vettore colonna di tutti 1
2) Fino a convergenza ripetere:

$$a^{(k)} = L^T h^{(k-1)}$$
$$h^{(k)} = L \cdot a^{(k)}$$
$$k = k + 1$$

Normalizzazione $a^{(k)}$ $h^{(k)}$ rispetto al valore più alto rispettivamente di $a^{(k)}$ e $h^{(k)}$

(b) esempio network con alcuni vertici che hanno hub score = 0



In questo caso B e D hanno hub score 0 perché non puntano a nessuno

hub → / authorities ↓	A	B	C	D
A	0	1	1	1
B	0	0	0	0
C	1	0	0	0
D	0	0	0	0

L_{ij}

3) Discuss how pagerank could be used as an alternative measure of paper's scientific impact

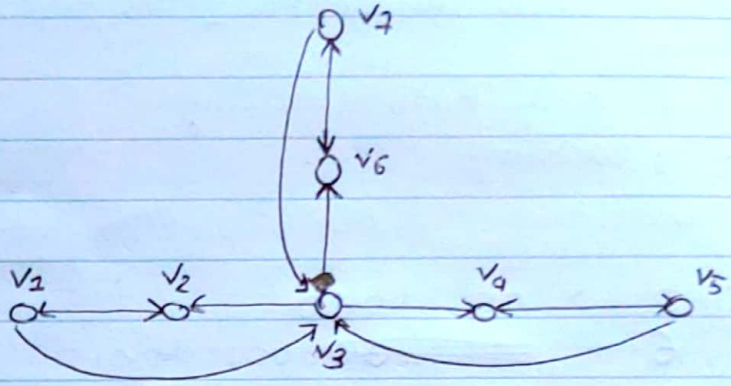
EXAM 11/01/2011

1) general teleporting probability α

2) $\pi_2 = \pi_4$ } for symmetry
$\pi_1 = \pi_5$ } (le probabilità su ogni nodo)
teleporta

$$\pi_3 = \frac{\alpha}{7} + 2\left(\frac{1-\alpha}{2}\pi_1\right) + \frac{(1-\alpha)}{2}\pi_7$$

(links $(1-\alpha)$; 2 perché $\pi_2 = \pi_4$; link uscenti dai nodi vicini)



$$\pi_6 = \frac{\alpha}{7} + \frac{(1-\alpha)}{2}\pi_7 + \frac{(1-\alpha)}{3}\pi_3$$

$$\pi_7 = \frac{\alpha}{7} + \frac{(1-\alpha)}{1}\pi_6$$

$$\pi_2 = \frac{\alpha}{7} + \frac{(1-\alpha)}{2}\pi_1 + \frac{(1-\alpha)}{3}\pi_3 = \pi_4$$

$$\pi_1 = \frac{\alpha}{7} + \frac{(1-\alpha)}{1}\pi_2 = \pi_5$$

$$\sum_i \pi_i = 1 \quad i = 1 \dots 7$$

b) $\alpha = \frac{1}{2}$

$$\begin{cases} \pi_1 = \frac{1}{14} + \left(1 - \frac{1}{2}\right)\pi_2 + \dots \\ \pi_2 = \frac{1}{14} + \left(\frac{1-\frac{1}{2}}{3}\right)\pi_3 + \left(\frac{1-\frac{1}{2}}{2}\right)\pi_1 \\ \pi_7 = \frac{1}{14} + \frac{1}{2}\pi_6 \\ \pi_6 = \frac{1}{14} + \frac{1}{4}\pi_7 + \left(\frac{1-\frac{1}{2}}{3}\right)\pi_3 \\ \pi_3 = \frac{1}{14} + \frac{1}{2}\pi_1 + \frac{1}{2}\pi_7 \end{cases}$$

$\pi_2 = \pi_4$
$\pi_1 = \pi_5$

CALCOLI LUNGHISSIMI NON HO VOGLIA DI FARLI (LASCIA PERDERE PURE TU)

1) What is the importance of teleporting probability with respect to the convergence of pagerank?

We use teleporting probability in pagerank to avoid the interruption of random walk due to spider traps or dead ends. In fact teleporting probability is the possibility to jump to another node in the network, with a certain probability, to exit loops in the graphs.

512 128 64 32 16 8 4 2 1
0 0 0 0 0 0 0 0 0

2) Show how can compress the list [2, 8, 17, 22, 30, 40, 52, 80] using
(a) variable byte encoding → 2, 6, 9, 5, 8, 10, 12, 28
(b) γ encoding.

riscrivo i valori con i gap

(a) 2 → 00000010
6 → 00000110
9 → 00001001
5 → 00000101
8 → 00001000
10 → 00001010
12 → 00001100
28 → 00011100

quindi creo il codice mettendo 0 separando per ultimi 7 bit e mettendoci 1 come primo bit dell'ultimo byte. Mentre come primo bit dei byte prima metto uno 0.

Per decodificare con variable byte code leggiamo una sequenza di byte con continuation bit 0 terminato da un byte con continuation bit 1

→ implica che è l'ultimo byte del numero 2

2 6 9 5
10000010 10000110 10001001 10000101
8 10 12 28
10001000 10001010 10001100 10011100

(3) γ encoding

→ uno il primo 1 per l'offset

2 → 000000 10 offset → 0 → unario = 10 → unario + offset

→ metto un 1 per ogni elemento nell'offset e aggiungo uno finale

6 → bin → offset → 10 unario = 110 → 11010
9 → bin offset → 001 unario = 1110 → 1110001
5 → bin offset → 01 unario = 110 → 11001
8 → bin offset → 000 unario = 1110 → 1110000
10 → bin offset → 010 unario = 1110 → 1110010
12 → bin offset → 100 unario = 1110 → 1110100
28 → bin offset → 1100 unario = 11110 → 111101100

3) (a) give a linear algorithm for web AND Information AND NOT Retrieval

```
INTERSECT (p1, p2, ¬p3)
    resultAND ← ()          // gestisco AND
    answer ← ()             // gestisco AND NOT
    while p1 != NULL and p2 != NULL
        if docID(p1) = docID(p2)
            ADD (p1, resultAND)        (docID)
            p1 ← next(p1)
            p2 ← next(p2)
        if else docID(p1) < docID(p2)
            p1 ← next(p1)
        else (docID(p2) < docID(p1))
            p2 ← next(p2)

    while resultAND != NULL and p3 != NULL
        if docID(resultAND) = docID(p3)
            resultAND ← next(resultAND)
            p3 ← next(p3)
        if else docID(resultAND) < docID(p3)
            ADD (resultAND, answer)        (docID)
            resultAND ← next(resultAND)
        else (docID(resultAND) > docID(p3))
            ADD (...)
            p3 ← next(p3)

    if p3 = NULL
        while resultAND != NULL
            ADD (answer, docID(resultAND))
    return answer
```

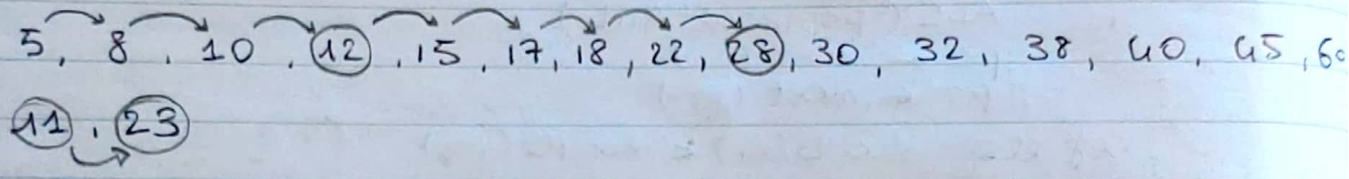
X AND NOT Y
prendo gli elementi di X che non sono in Y

(ex)
P1: 3, 4, 5, 8, 10
P2: 3, 4, 6, 8, 19
resultAND: 3, 4, 8
P3: 6, 8, 9
answer: 3, 4

(b) 2-word query. The postings lists are:
[5, 8, 10, 12, 15, 17, 18, 22, 28, 30, 32, 38, 40, 45, 60]
[11, 23]

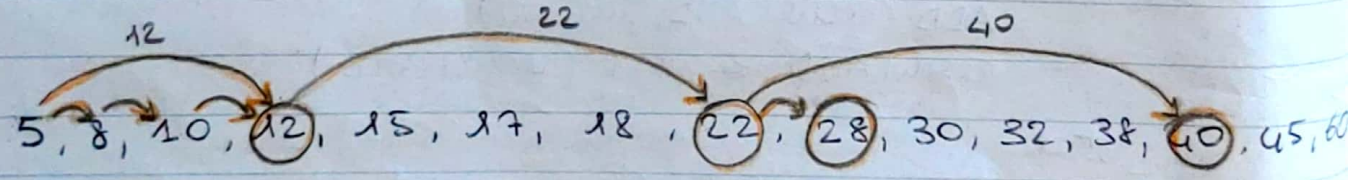
how many comparisons for the intersection and justify the answer:

(a) Using standard postings lists



comincio con 5 e 11, 5 < 11 quindi scorro fino a trovare un numero uguale o > 11.
12 > 11 quindi prendo il successivo di 11 (23) e siccome 12 < 23 scorro i successivi di 12 ecc.
In questo caso l'intersezione restituisce una lista vuota e ho effettuato 10 (12) comparazioni
answer = { }

(b) Using lists stored with skip pointers with skip length of $\sqrt{P}$ where P is the length of the postings list.



answer = { }
Siccome P = 15, $\sqrt{P}$ = 3 e qualcosa (quasi 4), quindi i salti saranno di 3/4 elementi circa
comincio confrontando 5 e 11, 5 < 11 quindi dovrei prendere il successivo, ma siccome ho uno skip, provo a vedere se lo skip è ≤ 11. 12 > 11, quindi potrei avere l'11 tra gli elementi che ho saltato, allora riprendo il confronto da 5. 8 < 11, 10 < 11. Siccome non ho 11 nella prima lista e 12 > 11, allora prendo il successivo di 11, 23.
23 > 12 quindi provo a usare lo skip. 22 < 23, quindi posso saltare i valori compresi tra 12 e 23 e tento lo skip a 40
40 > 23, quindi torno a 22 e confronto i successivi. 28 > 23, quindi termina il confronto
Ho effettuato 8 comparazioni, quindi è stato vantaggioso

STEMMING: vado a considerare solo la radice delle parole
COMPUTER → COMPUT ⇐ ER, ATION

LEMMATIZATION: vado a considerare i lemmi delle parole
SONO, SEI, È → LEMMA ESSERE

EXAM 26/01/2012

~~Front queues and back queues in Mercator's crawler URL frontier scheme? Explain how they work.~~

1) TRUE OR FALSE

1) (a) In a Boolean retrieval sys., stemming never lowers precision

FALSE: stemming can increase the retrieved set without increasing the number of relevant documents

(b) " , stemming never lowers recall

TRUE: stemming can only increase the retrieved set, which means increased or unchanged recall

(c) Stemming increases the size of the vocabulary

FALSE: stemming decreases the size of the vocabulary because I reduce different words to the same radix.

(d) Stemming should be invoked at indexing time but not while processing a query.

FALSE: the same processing should be applied to documents and queries to ensure matching terms.

2) Why skip pointers are not useful for (x OR y)?
because in order to compute the union of the two, you have to run both the lists one element by one.

3) Bivalent index "New York" "York University"
Example of a document returned for a query New York University which is a false positive that should not be returned.

Document = " I like New York. York University is awesome. "

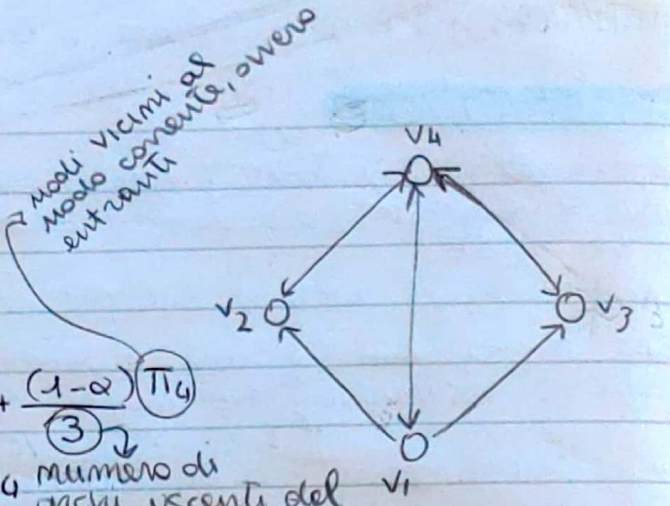
PRECISION = # relevant retrieved / # total retrieved

RECALL: # relevant retrieved / # relevant total

3) teleporting prob α

$\sum_i \pi_i = 1 \qquad i = 1,2,3,4$

$\pi_2 = \pi_3$ simmetria



$$\pi_1 = \frac{\alpha}{4} + \frac{(1-\alpha)}{2}\pi_2 + \frac{(1-\alpha)}{2}\pi_3 + (1-\alpha)\pi_4$$

$$\pi_2 = \frac{\alpha}{4} + \frac{(1-\alpha)}{3}\pi_1 + \frac{(1-\alpha)}{3}\pi_4$$

(3) numero di archi uscenti del vicino

$$\pi_4 = \frac{\alpha}{4} + \frac{(1-\alpha)}{3}\pi_1 + \frac{(1-\alpha)}{1}\pi_2 + \frac{(1-\alpha)}{1}\pi_3$$

$$\pi_3 = \frac{\alpha}{4} + \frac{1-\alpha}{3}\pi_1 + \frac{(1-\alpha)}{3}\pi_4$$

(b) $\alpha = \frac{1}{2}$

$$\pi_1 = \frac{1}{8} + \frac{1}{2}\pi_2 + \frac{1}{2}\pi_3 + \frac{1}{6}\pi_4$$

$$\pi_2 = \frac{1}{8} + \frac{1}{6}\pi_1 + \frac{1}{6}\pi_4$$

$$\pi_4 = \frac{1}{8} + \frac{1}{6}\pi_1 + \frac{1}{2}\pi_2 + \frac{1}{2}\pi_3$$

$$\pi_3 = \pi_2$$

$$\begin{cases} \pi_1 = \frac{1}{8} + \frac{1}{6}\pi_4 \\ \pi_2 = \frac{1}{8} + \frac{1}{6}\pi_1 + \frac{1}{6}\pi_4 \\ \pi_4 = \frac{1}{8} + \frac{1}{6}\pi_1 + \frac{1}{2}\pi_2 + \frac{1}{2}\pi_3 \\ \pi_3 = \frac{1}{4} + \frac{1}{6}\pi_1 + \frac{1}{6}\pi_4 \end{cases}$$

$$\pi_2 = \frac{1}{8} + \frac{1}{6}\left(\frac{1}{8} + \frac{1}{6}\pi_4\right) + \frac{1}{6}\pi_4$$

$$\pi_2 = \frac{1}{8} + \frac{1}{48} + \frac{1}{36}\pi_4 + \frac{1}{6}\pi_4 \longrightarrow \frac{7}{48} + \frac{7}{36}\pi_4$$

$$\pi_4 = \frac{1}{8} + \frac{1}{6}\left(\frac{1}{8} + \frac{1}{6}\pi_4\right) + \pi_2 = \frac{1}{8} + \frac{1}{6}\left(\frac{1}{8} + \frac{1}{6}\pi_4\right) + \frac{7}{48} + \frac{7}{36}\pi_4$$

$$\frac{36-7-1}{36}\pi_4 = \frac{14}{48} \longrightarrow \pi_4 = \frac{14}{48}\cdot\frac{36}{28} = \frac{3}{8}$$

$$\pi_2 = \frac{7}{48} + \frac{7}{36}\cdot\frac{3}{8} = \frac{14+7}{96} = \frac{21}{96} = \frac{7}{32} = \pi_3$$

$$\pi_1 = \frac{1}{8} + \frac{1}{6}\cdot\frac{3}{8} = \frac{3}{16}$$

$$\sum \pi_i = 1 \implies \frac{3}{16} + \frac{7}{32} + \frac{7}{32} + \frac{3}{8} = \frac{32}{32} = 1$$

OK

(c) prove that for any graph the page rank of each node is at least $\frac{\alpha}{N}$

$\forall v: \pi_v \geq \frac{\alpha}{n}$

$\pi_v = \underbrace{\frac{\alpha}{N}}_{\text{teleport}} + \underbrace{(1-\alpha) \sum_{j \to i} \frac{\pi_j}{d_i}}_{\text{following link}}$ where d_i is the number of out links

$\Rightarrow$ always > 0 because the sum of rank in a graph $= 1$

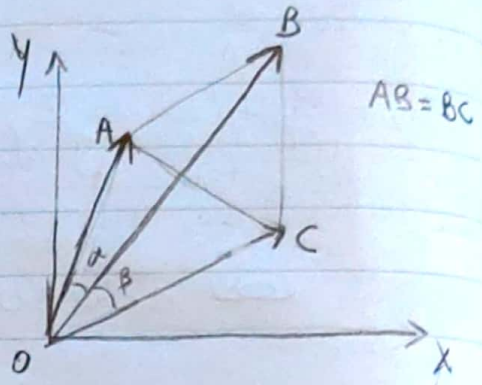
whereas $1 < \alpha < 0$ and $(1-\alpha) \sum \frac{\pi_i}{d_i} > 0$, $\Rightarrow \pi_v \geq \frac{\alpha}{N}$ sempre.

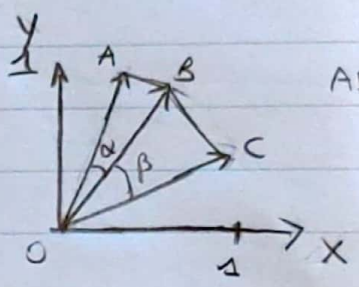
(4) KNN e K-means, | KNN con inverted list |

2) Show that for normalized vectors, Euclidean distance gives the same proximity ordering as the cosine measure

(A) Normalized vectors have module = 1.

Let's consider this example, where $\alpha < \beta$.
We know that OA is nearer to OB because cosine is greater, so is more similar than OC.
However, the triangle $\hat{ABC}$ is equilateral, so the euclidean distance in this case would lead to misleading results because it will consider OA and OC equidistanced from OB.





To solve this problem we normalize vectors, making them of module = 1.
In this way the angle will remain unchanged, while the euclidean distance will be coherent with the cosine similarity because it will show that the two vector with smaller angle have smaller Euclidean distance. (same proximity ordering)

(B) Is this true for non normalized vectors?

The previous statement is not true for non normalized vectors and the proof is in the previous exercise

EXAM ~~2012~~ 02/02/2012

1) (A) for which queries skip pointers are ~~not~~ useful and for which are useless. Briefly explain your answer.

(a) x AND y, where x frequent term and y rare
in this case it could be useful,

(ex) x: 1, 2, 5, 15, 19, 20, 32, 35 (5 → 19 → 32 → 45)
y: 19
facao pochi passaggi

(b) x AND y where both x and y are frequent
USEFUL because it save times

(c) x OR y
(d) x OR y
} USELESS because in OR we have to sum all elements of both lists.

(B) pseudocode merging two postings lists term 1 AND term 2 using SKIP POINTERS

```
MERGE SKIPPOINTERS (p1, p2)
    Answer ← ()
    while p1 != NULL and p2 != NULL
        if docID(p1) = docID(p2)
            ADD(answer, docID(p1))
            p1 ← next(p1)
            p2 ← next(p2)

        if docID(p1) < docID(p2)
            if hasSkip(p1) and docID(skip(p1)) ≤ docID(p2)
                while hasSkip(p1) and docID(skip(p1)) ≤ docID(p2)
                    p1 ← skip(p1)
            else p1 ← next(p1)

        if docID(p2) < docID(p1)
            if hasSkip(p2) and docID(skip(p2)) ≤ docID(p1)
                while hasSkip(p2) and docID(skip(p2)) ≤ docID(p1)
                    p2 ← skip(p2)
            else p2 ← next(p2)
    return ANSWER.
```

2) teleporting prob α

$n = 4$

$\alpha = \frac{1}{2}$

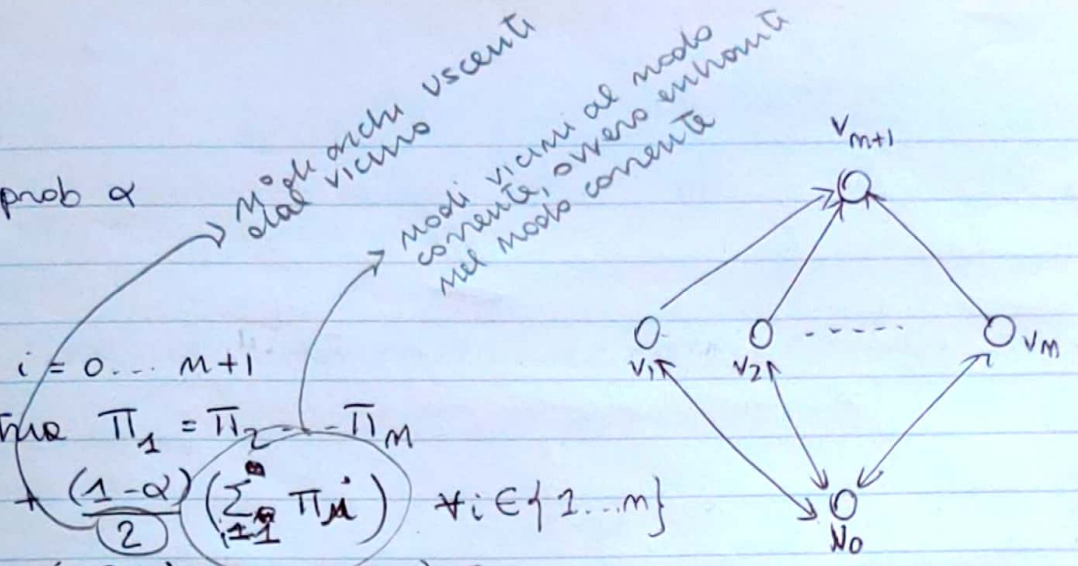
$\sum_{i=0}^{n+1} \pi_i = 1 \quad i = 0 \dots n+1$

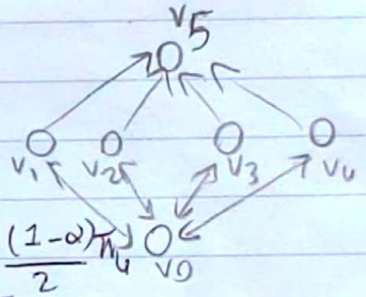
per simmetria $\pi_1 = \pi_2 = \dots = \pi_n$

$\pi_0 = \frac{\alpha}{n+2} + \frac{(1-\alpha)}{2}\left(\sum_{i=1}^{n} \pi_i\right) \quad \forall i \in \{1 \dots n\}$

$\pi_i = \frac{\alpha}{n+2} + \frac{1-\alpha}{n}\pi_0$

$\pi_{n+1} = \frac{\alpha}{n+2} + \frac{(1-\alpha)}{2}\left(\sum_{i=1}^{n} \pi_i\right)$





$$\begin{cases} \pi_0 = \frac{\alpha}{6} + \frac{(1-\alpha)}{2}\pi_1 + \frac{(1-\alpha)}{2}\pi_2 + \frac{(1-\alpha)}{2}\pi_3 + \frac{(1-\alpha)}{2}\pi_4 \\ \pi_5 = \frac{\alpha}{6} + \frac{1-\alpha}{2}\pi_1 + \frac{1-\alpha}{2}\pi_2 + \frac{1-\alpha}{2}\pi_3 + \frac{1-\alpha}{2}\pi_4 \\ \pi_1 = \frac{\alpha}{6} + \frac{1-\alpha}{4}\pi_0 \end{cases}$$

$\pi_1 = \pi_2 = \pi_3 = \pi_4$

$$\begin{cases} \pi_0 = \frac{1}{12} + \frac{1}{4}(\pi_1 + \pi_2 + \pi_3 + \pi_4) = \frac{1}{12} + \frac{4}{4}\left(\frac{1}{12} + \frac{1}{8}\pi_0\right) \\ \pi_0 - \frac{1}{8}\pi_0 = \frac{1}{12} + \frac{1}{12} \rightarrow \left(1 - \frac{1}{8}\right)\pi_0 = \frac{1}{6} \end{cases}$$

$\pi_0 = \frac{1}{6} \cdot \frac{8}{7} = \frac{4}{21}$

$\pi_0 = 0,19$

$\pi_5 = 0,19$

$\pi_1 = \frac{1}{12} + \frac{1}{8}(0,19) = \frac{1}{12} + 0,02375 = 0,107 = \pi_2 = \pi_3 = \pi_4$

$\Sigma = 0,19 + 0,19 + (0,107 \times 4) = 0,38 + 0,428 = 0,81$

dovrebbe essere questo ma non viene 1

N.B

IMPORTANTE PER COSINE SIM e EUCLIDEAN DIST.

un documento viene rappresentato in uno spazio vettoriale come un vettore. Il numero di componenti del vettore sono gli elementi del dizionario che forma il documento (ovvero i termini senza ripetizioni).

Più è lungo il documento, più è lungo il vettore.

Addition β V0

Markov chain

---

(5) interpolation 33%?
Vedo nella stringa a cosa corrisponde il 33% delle recall.
Siccome ho 10 elementi rilevanti totali, il 33% sarà intorno ai 3 elementi (33% di 10). Quindi scorro la lista fino a che non ottengo il 33%.

(es) $\frac{1}{10} = 0,1$ $\frac{2}{10} = 0,2$ $\frac{3}{10} = 0,3$ → siccome questo è il 30% potrei usare questo

precision $= \frac{3}{4} \approx 75\%$

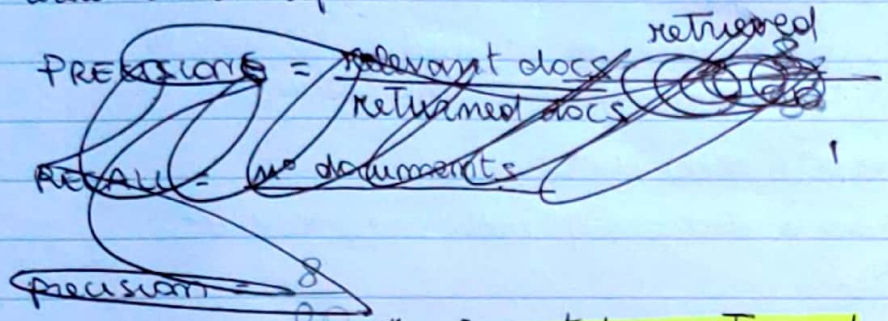
Quindi all 33% di recall corrisponde circa il 75% di precision

④ R e N R = relevant returned
N = non relevant returned

30 documents

RRNRN NRRNN NRNNN NRNNR NNNNN NRNRN

① precision of the system on the top 20?
② recall on the top 20?



$$\text{PRECISION} = \frac{\#\text{ relevant docs retrieved}}{\#\text{ total retrieved}} = \frac{8}{20} = \frac{2}{5} = 0,4$$

$$\text{RECALL} = \frac{\#\text{ relevant docs retrieved}}{\#\text{ total relevant docs}} = \frac{8}{10} = \frac{4}{5} = 0,8$$

③ F_1 measure on top 20?

$$F_1\text{-SCORE} = 2 \cdot \frac{\text{PRECISION} \cdot \text{RECALL}}{\text{PRECISION} + \text{RECALL}} = 2 \cdot \frac{\frac{2}{5} \cdot \frac{4}{5}}{\frac{2}{5} + \frac{4}{5}} = \frac{8}{15}$$

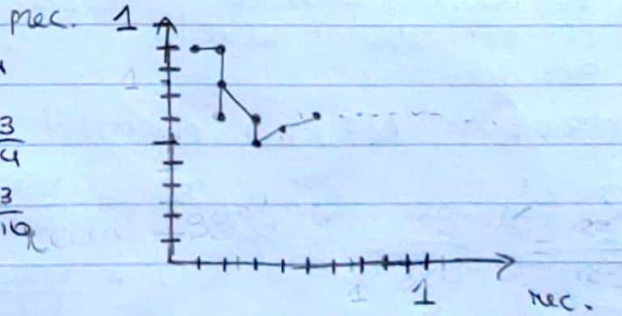
④ draw the precision-recall curve
precision y recall x
faccio prima prec. e recall sul primo elemento, poi sul primo e secondo, poi sul primo secondo e terzo .... ecc.

$pr_1 = \frac{1}{1}$ $pr_2 = \frac{2}{2}$ $pr_3 = \frac{2}{3}$ $pr_4 = \frac{3}{4}$

$rec_1 = \frac{1}{10}$ $rec_2 = \frac{2}{10}$ $rec_3 = \frac{2}{10}$ $rec_4 = \frac{3}{10}$

$pr_5 = \frac{3}{5}$ $pr_6 = \frac{3}{6}$ $pr_7 = \frac{4}{7}$ $pr_8 = \frac{5}{8}$

$rec_5 = \frac{3}{10}$ $rec_6 = \frac{3}{10}$ $pr_7 = \frac{4}{10}$ $rec_8 = \frac{5}{10}$



⑤ interpolation 33% recall

PRECISION : $\frac{\text{relevant documents retrieved}}{\text{retrieved documents}}$

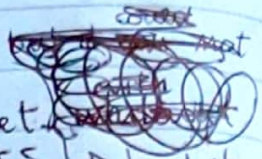
RECALL = $\frac{\text{relevant retrieved documents}}{\text{total relevant}}$

EXAM 20/07/2015

1) True or False ?

(A) (a) stemming always increases precision

FALSE : because stemming increases the retrieved set increasing the number of relevant documents but it could not does not always increase
So stemming ~~decreases~~ precision (it can be unchanged or lower)



(b) stemming increases recall

FALSE : stemming can increase ~~as~~ or unchange recall because it can only increase the retrieved set

(c) Stemming reduces the size of the dictionary

TRUE: because we have a dictionary with only radices

(B) Are skip pointers useful for X AND NOT Y

NO because we have to consider all elements of X ~~that~~ that are not in Y and discard those that are equals. There could be some useful skip pointers ma in general No.

(C) assume a biword index. Example with New York University which is a false positive

Document : "I like New York. York University here is great."

↳ they are both in the same document but not next to each other

2) R = relevant retrieved
N = non relevant retrieved
30 document

NRRRN NRRRN NRNNN NRNNR NNNNN NRNRN

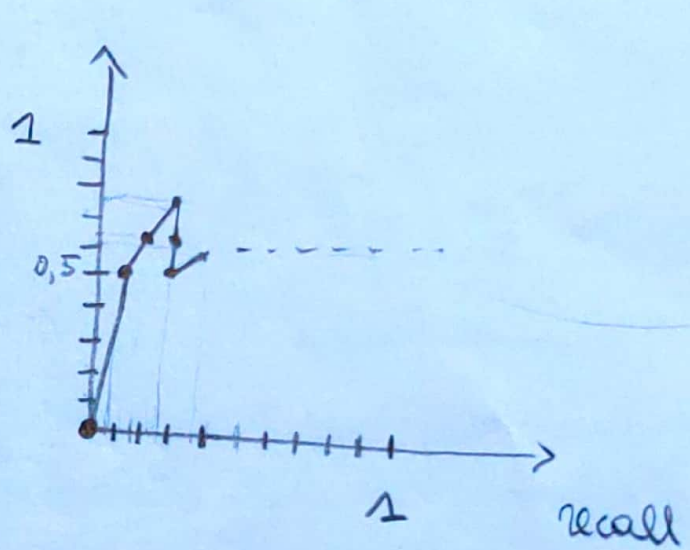
(1) precision. top 20 ? = $\frac{\text{relevant doc retrieved}}{\text{total retrieved}} = \frac{9}{20}$

(2) recall top 20 ? = $\frac{\text{relevant doc retrieved}}{\text{total rel. doc}} = \frac{9}{11}$

(3) $F_1\text{-SCORE} = 2 \cdot \frac{\text{Prec} \cdot \text{rec}}{\text{Prec} + \text{rec}} = 2 \cdot \frac{\frac{9}{20} \cdot \frac{9}{11}}{\frac{9}{20} + \frac{9}{11}} =$

(4) draw precision-recall curve

(4) draw precision and recall curve.
precision on y axis
recall on x axis
from 0 to 1
to calculate it we calculate precision and recall ~~for every~~ jrsily for the first element, then for the first two, then the three and so on....



$prec_1 = \frac{0}{1} = 0$ $\quad prec_{1,2} = \frac{1}{2}$

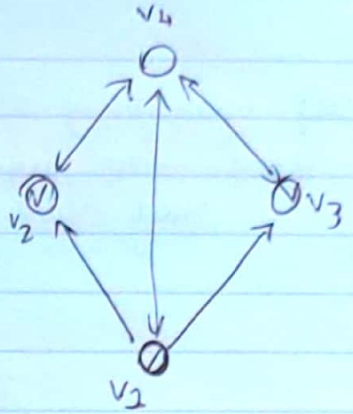
$rec_1 = \frac{0}{11} = 0$ $\quad rec_{1,2} = \frac{1}{11} = 0,09$

$prec_{1,2,3} = \frac{2}{3} = 0,66$ $\quad prec_{1,2,3,4} = \frac{3}{4} = 0,75$ $\quad prec_5 = \frac{3}{5} = 0,6$ $\quad prec_6 = \frac{3}{6} = 0,5$

$rec_{1,2,3} = \frac{2}{11} = 0,18$ $\quad rec_{1,2,3,4} = \frac{3}{11} = 0,27$ $\quad rec_5 = \frac{3}{11} =$ $\quad prec_6 = \frac{3}{11} =$

$prec_7 = \frac{4}{7} = 0,57$

$rec_7 = \frac{4}{11} = 0,36$



3) teleporting probability α.
calculate personalized pagerank
personalization vector $\{1, 0, 0, 0\}$

$\sum \pi_i = 1 \quad i = 1..4$

$\pi_2 = \pi_3$ per simmetria

$\pi_1 = \frac{\alpha}{4} + \frac{(1-\alpha)}{3}\pi_4$ (il vicino è quello entrante nel modo corrente; 3 → numero di nodi uscenti dal vicino)

$\pi_2 = \frac{\alpha}{4} + \frac{(1-\alpha)}{3}\pi_4 + \frac{(1-\alpha)}{3}\pi_1$

$\pi_4 = \frac{\alpha}{4} + \frac{(1-\alpha)}{3}\pi_1 + \frac{(1-\alpha)}{1}\pi_2 + \frac{(1-\alpha)}{1}\pi_3$

(2) $p_1 = (1, 0, 0, 0)$

$\pi_1 = \alpha + \frac{(1-\alpha)}{3}\pi_4$

$\pi_2 = \frac{(1-\alpha)}{3}\pi_4 + \frac{(1-\alpha)}{3}\pi_1 = \pi_3$

$\pi_4 = \frac{(1-\alpha)}{3}\pi_1 + (1-\alpha)(\pi_2 + \pi_3)$

(3) $p_2 = \{0, 1, 0, 0\}$

$\pi_1 = \frac{(1-\alpha)}{3}\pi_4$

$\pi_2 = \alpha + \frac{(1-\alpha)}{3}(\pi_4 + \pi_1)$

$\pi_3 = \frac{(1-\alpha)}{3}(\pi_4 + \pi_1)$

$\pi_4 = \frac{(1-\alpha)}{3}\pi_1 + (1-\alpha)(\pi_1 + \pi_3)$

(4) explain in detail how to calculate the personalized pagerank with respect to the personalization vector $p_3 = \{0,5, 0,5, 0, 0\}$ without solving the system. $p_3 = \{\frac{1}{2}, \frac{1}{2}, 0, 0\}$

we know that the positions in p corresponds to π_1 and π_2, this means that the probability is shared between those two

$\pi_1(p_3) = \pi_1(p_1) - \frac{\alpha}{2}$

$\pi_2(p_3) = \pi_2(p_2) - \frac{\alpha}{2}$

$\pi_3(p_3) = \pi_3(p_1) = \pi_3(p_2)$

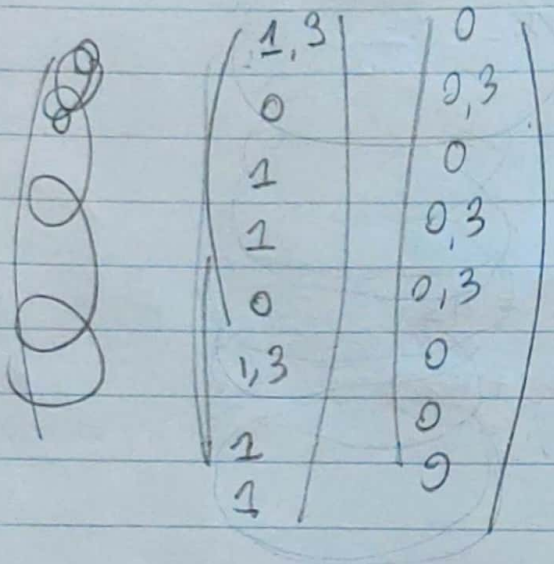
$\pi_4(p_3) = \pi_4(p_1) = \pi_4(p_2)$

4) tf × idf weighting scheme

(a) two docs only frequent words (the, a, an, of etc) in common

(b) " that have no words in common

(c) " " many rare words in common



512 256 128 64 32 16 8 4 2 1
0 0 0 0 0 0 0 0 0 0

EXAM 09/2017

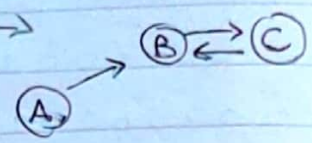
2) (3) compress using byte encoding [5, 7, 18, 19, 28, 40, 52, 80]
↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
5 2 11 1 9 12 12 28

5 → 00000101 ⇒ 10000101
2 → 00000010 ⇒ 10000010
11 → 00001011 ⇒ 10001011
1 → 00000001 ⇒ 10000001
9 → 00001001 ⇒ 10001001
12 → 00001100 ⇒ 10001100
12 → " ⇒ "
28 → 00011100 ⇒ 00011100

3) Importance of teleporting probability with the convergence of pagerank.

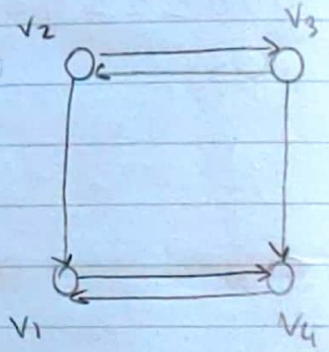
Teleporting probability is important in pagerank because it avoids spider traps which are when the random user can't escape from two nodes.

With teleporting probability α, we can both follow a certain link or we can jump to another node



$$\pi_i = (1-\alpha) \sum_{j \to i} \frac{\pi_j}{d_j} + \alpha p_i$$

where p_i is the personalized vector that shows with what probability you can jump on a node rather than another.



② equations for probability α

$$\sum \pi_i = 1 \qquad i = 1 \dots 4$$

$$\pi_1 = \frac{\alpha}{4} + \frac{(1-\alpha)}{2}\pi_2 + \frac{(1-\alpha)}{1}\pi_4$$

$$\pi_2 = \frac{\alpha}{4} + \frac{(1-\alpha)}{2}\pi_3$$

$$\pi_3 = \frac{\alpha}{4} + \frac{(1-\alpha)}{2}\pi_2$$

$$\pi_4 = \frac{\alpha}{4} + \frac{(1-\alpha)}{1}\pi_1 + \frac{(1-\alpha)}{2}\pi_3$$

$\pi_2 = \pi_3$, $\pi_1 = \pi_4$ } symmetry

③ $\alpha = \frac{1}{2}$

$$\begin{cases} \pi_1 = \frac{1}{8} + \frac{1}{4}\pi_2 + \frac{1}{2}\pi_4 \\ \pi_2 = \frac{1}{8} + \frac{1}{4}\pi_3 \\ \pi_3 = \frac{1}{8} + \frac{1}{4}\pi_2 \\ \pi_4 = \frac{1}{8} + \frac{1}{2}\pi_1 + \frac{1}{2}\pi_3 \end{cases}$$

$$\begin{cases} \pi_2 = \frac{1}{8} + \frac{1}{4}\left(\frac{1}{8} + \frac{1}{4}\pi_2\right) \rightarrow \frac{1}{8} + \frac{1}{32} + \frac{1}{16}\pi_2 \\ \pi_2 = \frac{5}{32} \cdot \frac{16}{15} = \frac{1}{6} \\ \pi_2 = \pi_3 = \frac{1}{6} \end{cases}$$

$$\pi_1 = \frac{1}{8} + \frac{1}{24} + \frac{1}{2}\pi_1 \rightarrow \frac{1}{2}\pi_1 = \frac{4}{24} \rightarrow \pi_1 = \frac{1}{3} = \pi_4$$

$$\sum = 1 \rightarrow \frac{1}{6} + \frac{1}{6} + \frac{1}{3} + \frac{1}{3} = \frac{1}{3} + \frac{2}{3} = 1 \qquad \text{OK}$$

→ Unsupervised learning.

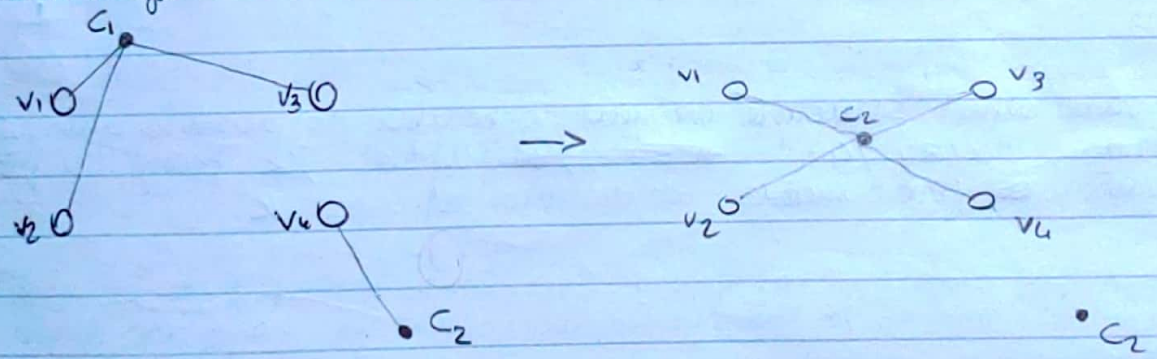
4) Explain how the K-MEANS works. Write the algorithm

① The K-means algorithm is used for classification problems. initially we take centroids ~~as~~ random and we take all points in a cluster that are nearest to the centroid. After that, the centroids is recalculated positioning it in a more central position of the cluster. This procedure is iterated until the convergence is reached and all elements of the clusters are correctly classified.

Algorithm

random position for the centroid
until convergence: → convergence when the old clusters are the same as the new.
do:
assign each point in the distribution to the nearest centroid.
do the mean to recalculate the new centroid positions

② Show that if the initial assignment is unlucky the K-means solution might be bad.



③ Why K-means converges?

K-means converges when the centroids do not update their position so the last position is the less expensive one in terms of costs. It converges because at every iteration the position of the centroids is adjusted until the center of the cluster is reached, that is when is less costly.
It converges because we minimize the distorsion measure through EM algorithm.

misure di distorsione
Cfr vedere su libro machine learning (424)

EXAM 01/2013

1) How to handle X AND NOT Y ? Why naive implementation expensive? Write algorithm merge. → NO POSTINGS LISTS

(a) To handle X AND NOT Y we take element of X that are not in Y (so all elements of X except for those in Y too)

The naive implementation is expensive because we have to run all element in the list X and see if there are elements in Y too and then discard them. We would have to use two cicles so obtain this result so the cost will be $O(n^2)m$)

~~MERGE(p1,p2) in fact we see if the smaller list is in the biggest one~~

answer ←
~~while p1 != nil and~~

in a naive implementation, we can consider an adjacency matrix with one entry for each element of the dictionary of the documents. However, if we have big collections of documents (like thousands) the adjacency matrix will be too big to be computed.
So this is not the best way to proceed and we use posting lists and even skip pointers when possible to obtain ~~operations~~ less costly operations.

(b) describe what structure we need to be able to answer queries such as "X /3 Y /4 Z" ~~where /k~~ with /k being the proximity operator ~~as~~ that means at distance at most k.

To handle this type of query (proximity queries) we use the positional index. In the positional index we have the posting list storing, for each document, the position in which the terms appear.

(ex) t1 → doc1: 1, 42, 45, 80; doc2: 7, 21 ... —

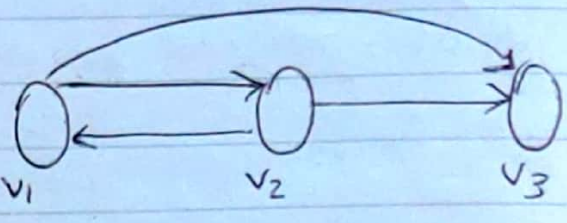
We can use proximity search to find ~~the~~ matching documents

3)

Describe an external memory algorithm for the implementations of the power iteration method for pagerank computation

?

4) teleporting probability α



$\sum \pi_i = 1 \quad i = 1,2,3$

$\pi_1 = \frac{\alpha}{3} + \frac{(1-\alpha)}{2}\pi_2$

$\pi_2 = \frac{\alpha}{3} + \frac{(1-\alpha)}{2}\pi_1$

$\pi_3 = \frac{\alpha}{3} + \frac{(1-\alpha)}{2}\pi_2 + \frac{(1-\alpha)}{2}\pi_1$

$\alpha = \frac{1}{2}$

$\pi_1 = \frac{1}{6} + \frac{1}{4}\pi_2 \rightarrow \pi_1 = \frac{1}{6} + \frac{1}{4}\left(\frac{1}{6} + \frac{1}{2}\pi_1\right) = \frac{1}{6} + \frac{1}{24} + \frac{1}{8}\pi_1$

$\pi_2 = \frac{1}{6} + \frac{1}{2}\pi_1$

$\frac{7}{8}\pi_1 = \frac{5}{24} \rightarrow \pi_1 = \frac{5}{24}\cdot\frac{8}{7} = \frac{5}{21}$

$\pi_3 = \pi_1$

$\alpha = \frac{1}{2}$

$$\begin{cases} \pi_1 = \frac{1}{6} + \frac{1}{4}\pi_2 \\ \pi_2 = \frac{1}{6} + \frac{1}{4}\pi_1 \\ \pi_3 = \frac{1}{6} + \frac{1}{4}\pi_2 + \frac{1}{4}\pi_1 \end{cases}$$

$\pi_2 = \frac{1}{6} + \frac{1}{4}\left(\frac{1}{6} + \frac{1}{4}\pi_2\right) = \frac{1}{6} + \frac{1}{24} + \frac{1}{16}\pi_2$

$\frac{15}{16}\pi_2 = \frac{5}{24} \Rightarrow \pi_2 = \frac{5}{24}\cdot\frac{16}{15} = \frac{2}{9}$

$\pi_1 = \frac{1}{6} + \frac{1}{24}\cdot\frac{2}{9} = \frac{1}{6} + \frac{1}{108} = \frac{19}{108}$

$\pi_3 = \frac{1}{6} + \frac{1}{24}\cdot\frac{2}{9} + \frac{1}{24}\cdot\frac{2}{9} = \frac{1}{6} + \frac{1}{18} + \frac{1}{108} = \frac{23}{108} = \frac{3+1+1}{18} = \frac{5}{18}$

$\pi_2 = \frac{2}{9}$

$\pi_1 = \frac{2}{9}$

$\pi_3 = \frac{5}{18}$

NON VIENE IL CALCOLO

$\frac{2}{9} + \frac{2}{9} + \frac{5}{18} = \frac{4+4+5}{18} =$

$\frac{16}{18}$

$\frac{2}{9} + \frac{5}{21} + \frac{23}{84} = \frac{14+20+23}{84}$

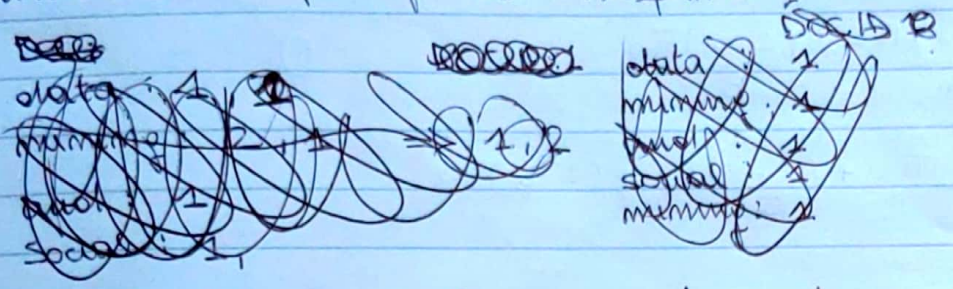
$\frac{2}{9}$

$\frac{1}{6} + \frac{1}{4}\left(\frac{1}{6} + \frac{1}{4}\right)$

EXAM 18-06-18

1) three textual documents (D1, D2, D3)
- D1: data mining and social mining
- D2: social network analysis
- D3: data mining

① write down the posting lists corresponding to the above document



term	DocID
data	1
mining	1
and	1
social	1
mining	1

term	DocId
social	2
network	2
analysis	2

term	DocId
data	3
mining	3

term	doc frequency
data	2
mining	3
and	1
social	2
network	1
analysis	1

⇒

posting list

data : [1], [3]
mining : [1], [3]
and : [1]
social : [1], [2]
network : [2]
analysis : [2]

② write down document frequency (df)
③ write down the term frequency (tf) for document D1
④ write down the formula that relates the tf-idf weigh for a term given its tf and idf.

term	df	tf_{D1}
data	2	1/5
mining	2	2/5
and	1	1/5
social	2	1/5
network	1	1/5
analysis	1	1/5

tf_{D1}: 1, 2, 1, 1 → tf assoluto, non rispetto alla lunghezza del documento

pesata rispetto alla lunghezza del documento

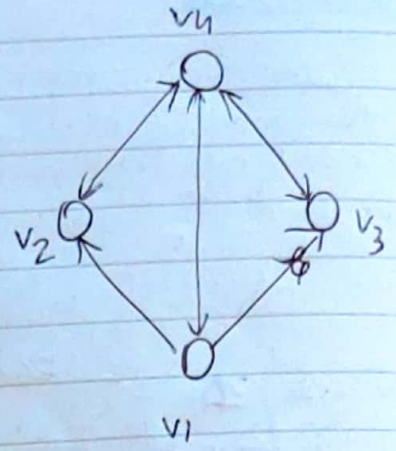
the tf-idf formula is: $tf_{t,d} \times idf_t = (1 + \log_{10} tf_{t,d}) \log_{10} \frac{N}{df_t}$
where N is the number of documents in the collection.

teleportation probability α

3) personalized vectors:

$p_1 = (0, 1, 0, 0)^T$

$p_2 = (0, 0, 1, 0)^T$



4)

$\sum \pi_i = 1 \quad i = 1 \dots 4$

$\pi_1 = \alpha p_i + \frac{(1-\alpha)}{3}\pi_4$

$\pi_2 = \alpha p_i + \frac{(1-\alpha)}{3}\pi_1 + \frac{(1-\alpha)}{3}\pi_4$

$\pi_3 = \pi_2$ simmetry

$\pi_4 = \alpha p_i + \frac{(1-\alpha)}{3}\pi_1 + \frac{(1-\alpha)}{3}\pi_2 + (1-\alpha)\pi_3$

$p_1 = (0, 1, 0, 0)$

$\pi_1 = \frac{1-\alpha}{3}\pi_4$

$\pi_2 = \alpha + \frac{1-\alpha}{3}(\pi_1 + \pi_4)$

$\pi_3 = \frac{1-\alpha}{3}(\pi_1 + \pi_4)$

$\pi_4 = \frac{1-\alpha}{3}\pi_1 + (1-\alpha)(\pi_2 + \pi_3)$

$p_2 = (0, 0, 1, 0)$

$\pi_1 = \frac{(1-\alpha)}{3}\pi_4$

$\pi_2 = \frac{(1-\alpha)}{3}(\pi_1 + \pi_4)$

$\pi_3 = \alpha + \frac{(1-\alpha)}{3}(\pi_1 + \pi_4)$

$\pi_4 = \frac{(1-\alpha)}{3}\pi_1 + (1-\alpha)(\pi_2 + \pi_3)$

2) Show how any Personalized pagerank vector π corresponding to personalization vector $\alpha_1 p_1 + \alpha_2 p_2$ $(\alpha_1 + \alpha_2 = 1)$ can be computed from π_1 and π_2.
If you can rigorously show why this works

$p = \alpha_1 p_1 + \alpha_2 p_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}\alpha_1 + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}\alpha_2 = (0\ \alpha_1\ \alpha_2\ 0)$

with $\alpha_1 + \alpha_2 = 1$

$\pi = \begin{pmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \pi_4 \end{pmatrix} \Rightarrow$

$\pi_1 = (1-\alpha)\frac{\pi_4}{3}$

$\pi_2 = \alpha_1 + (1-\alpha)\frac{(\pi_4 + \pi_1)}{3}$

$\pi_3 = \alpha_2 + (1-\alpha)\frac{\pi_4 + \pi_1}{3}$

$\pi_4 = (1-\alpha)(\pi_2 + \pi_3 + \frac{\pi_1}{3})$

If we have $\alpha_1, \alpha_2, \pi_1, \pi_2$, we can calculate π_4 and π_4
So we will calculate any personalised pagerank vector π.

we know that π_3 and π_2 have similar equations by symmetry.

$$\pi_2 = \alpha_1 + \underbrace{\frac{(1-\alpha)}{3}(\pi_4 + \pi_1)}$$

$$\pi_3 = \alpha_2 + \underbrace{\frac{(1-\alpha)}{3}(\pi_4 + \pi_1)}$$

factors equals

$\pi_2 - \alpha_1 =$ factors
$\pi_3 - \alpha_2 =$ factors $\Rightarrow \pi_2 - \alpha_1 = \pi_3 - \alpha_2$

$$\pi_2 = \pi_3 - \alpha_2 + \alpha_1$$
$$\pi_3 = \alpha_2 - \alpha_1 + \pi_2$$

whereas $\alpha_1 + \alpha_2 = 1 \Rightarrow \pi_3 = \pi_2 + \alpha_2 - 1 + \alpha_2$
$\alpha_1 = 1 - \alpha_2$
$= \pi_2 - 1 + 2\alpha_2$

$$\pi_4 = (1-\alpha)\left[\pi_2 + \underbrace{(\pi_2 - 1 + 2\alpha_2)}_{\pi_3} + \pi_{1/3}\right]$$
$$= (1-\alpha)\left[2\pi_2 - 1 + 2\alpha_2 + \pi_{1/3}\right]$$

So π_3 and π_4 can be expressed in terms of π_1 and π_2
therefore, we can calculate any $\pi = (\pi_1, \pi_2, \pi_3, \pi_4)$

EXAM 20-10-15

1) ① per le query a AND NOT b can we still run intersection in time $O(x+y)$?

yes because we have to run both lists in the worst case, taking only elements in a but not in b.

② same question for a OR NOT b

a OR NOT b → means that I have to take everything except for element not in a but in b.

The cost in this way in the worst case is $O(x+y)$

```
Algorithm a OR NOT b (p1, p2)
    answer ← ()
    while p1 != nil and p2 != nil
        if docID(p1) = docID(p2)
            ADD(answer, docID(p1))
            p1 ← next(p1)
            p2 ← next(p2)
        else if docID(p1) < docID(p2)
            ADD(answer, docID(p1))
            p1 ← next(p1)
        else docID(p2) < docID(p1)
            p2 ← next(p2)
    while p1 != NULL
        if p2 = NULL
            ADD(answer, docID(p1))
            p1 ← next(p1)
    return answer
```

ex) p1 : 1, 3, 4, 7, 9, 12, 15
p2 : 4, 5, 8, 9, 10

answer : 1, 3, 4, 7, 9, 12, 15

prendo tutto tranne gli elementi che stanno solo in p_2

→

a AND NOT b
elementi di a che non sono in b

③ MERGE(p1, p2)
answer ← ()
while p1 != NULL and p2 != NULL
if docID(p1) = docID(p2)
p1 ← next(p1)
p2 ← next(p2)
else if docID(p1) < docID(p2)
ADD(answer, docID(p1))
p1 ← next(p1)
else docID(p1) > docID(p2)
p2 ← next(p2)
while p1 != NULL
ADD(answer, docid(p1))
p1 ← next(p1)
return answer

ex p1: 1, 3, 5, 6, 15
p2: 3, 6, 14, 18
result: 1, 5, 15

SLIDE: CLASSIFICATION NAIVE BAYES

	doc ID	words in document	in c = China ?
training set	1	chim, Beij, chum	yes
	2	chum, chum, Shang.	yes
	3	chim, Macao	yes
	4	Tokyo, Japan, chum	no
Test set	5	chum, chum, chum, Tokyo, Jap	?

FACCIO PRIMA SENZA SENZA SMOOTHING

quindi non devo aggiungere 1 al numeratore e il numero di elementi del vocabolario al denominatore

$P(c = yes) = \frac{3}{4}$ $P(c = no) = \frac{1}{4}$

$P(chum/yes) = \frac{5}{8}$ $P(Beij/yes) = \frac{1}{8}$ $P(Shang/yes) = \frac{1}{8}$

$P(Beij/no) = 0$ $P(shang/no) = 0$

$P(chum/no) = \frac{1}{3}$

$P(Macao/yes) = \frac{1}{8}$

$P(Macao/no) = 0$

$P(Tokyo/yes) = 0$ $P(Jap/yes) = 0$

$P(Tokyo/no) = \frac{1}{3}$ $P(jap/no) = \frac{1}{3}$

adesso calcolo la probabilità del test set (d5) rispetto a yes o no

$P(yes/d5) = P(c=yes) \cdot P(chum/yes)^3 \cdot P(tokyo/yes) \cdot P(jap/yes) =$

$= \frac{3}{4} \cdot \frac{125}{512} \cdot 0 \cdot 0 = 0$

$P(no/d5) = P(c=no) \cdot P(chum/no)^3 \cdot P(tokyo/no) \cdot P(jap/no) =$

$= \frac{1}{4} \cdot \frac{1}{27} \cdot \frac{1}{3} \cdot \frac{1}{3} = 0,001$

quindi senza smoothing d5 è classificato come (no)

lo smoothing serve per evitare che delle probabilità vengano 0 come succede per $P(d5/yes)$ in questo caso.

- FACCIO con SMOOTING : numero di elementi/prob + 1 / numero totale/prop + numero totale dei diversi

$P(chum/yes) = \frac{5+1}{8+6} = \frac{6}{14} \frac{3}{7}$ ↳ numero elementi di zero nullo

$P(tokyo/yes) = \frac{0+1}{8+6} = \frac{1}{14}$

$P(chum/no) = \frac{1+1}{3+6} = \frac{2}{9}$

$P(tokyo/no) = \frac{1+1}{3+6} = \frac{2}{9}$

$P(Jap/yes) = \frac{0+1}{8+6} = \frac{1}{14}$

$P(Jap/No) = \frac{1+1}{3+6} = \frac{2}{9}$

$P(YES/ds) = \frac{3}{4} \cdot \left(\frac{3}{7}\right)^3 \cdot \frac{1}{14} \cdot \frac{1}{14} = 0,0003$

$P(NO/ds) = \frac{1}{4} \cdot \left(\frac{2}{9}\right)^3 \cdot \frac{2}{9} \cdot \frac{2}{9} = 0,0001$

In questo caso viene classificato come yes

EXAM - 2015 - 16

2) RRNRR NRRNNN NRNNN NRNNR NNNNN NRNRN

(a) precision and recall 10

$$\text{precision} = \frac{\#\text{ relevant retrieved documents}}{\#\text{ retrieved documents}} = \frac{6}{10}$$

$$\text{recall} = \frac{\#\text{ relevant retrieved documents}}{\#\text{ relevant documents tot}} = \frac{6}{11}$$

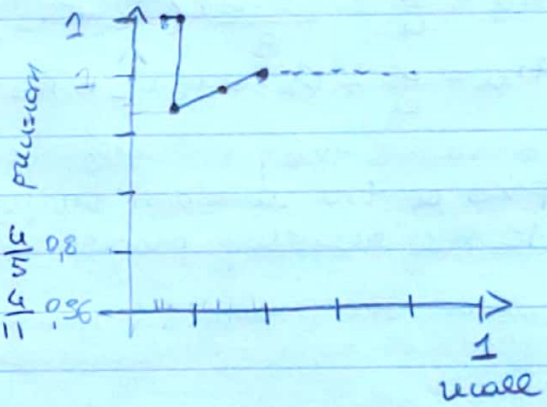
(b) precision-recall curve

$prec_1 = \frac{1}{1} = 1$ $prec_{1,2} = \frac{2}{2} = 1$

$rec_1 = \frac{1}{11} 0,09$ $rec_{1,2} = \frac{2}{11} 0,18$

$prec_{1,2,3} = \frac{2}{3} 0,66$ $prec_4 = \frac{3}{4} 0,75$ $prec_5 = \frac{4}{5} 0,8$

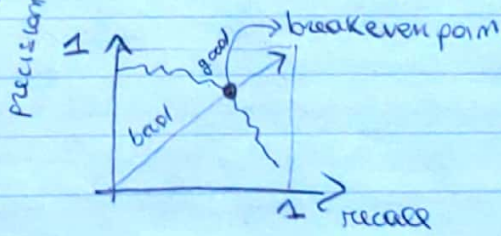
$rec_{1,2,3} = \frac{2}{11}$ $rec_4 = \frac{3}{11} 0,27$ $rec_5 = \frac{4}{11} 0,36$



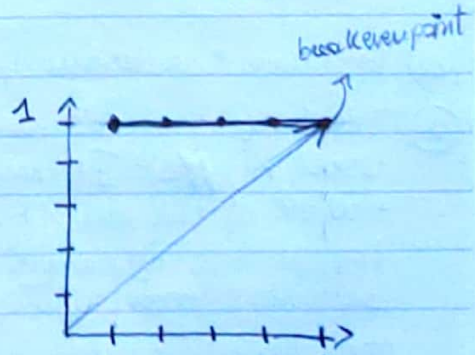
(c) must there always be a break even point between precision and recall? Either show there must be or give a counter-example.

COSA E' IL BREAK EVEN POINT:

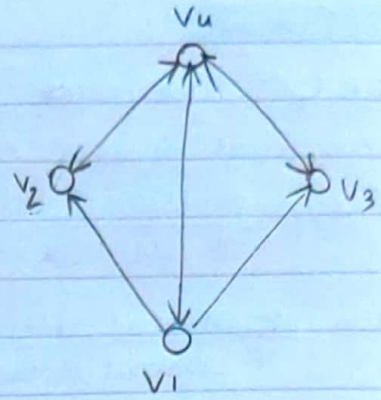
il break even point è il punto che indica che prima ci sono solo risultati peggiori e dopo ci sono solo risultati migliori.
Si ottiene dall'intersezione tra la bisettrice e la curva precision-recall.



To solve the problem, consider a collection RRRRR, then the graph will be this one →
This means that the break even point will coincide with the maximum point ~~of the~~ possible in the graph (1,1).
This means means that there is no improvement in the result after it.
So there no always be a breakeven point.



3) α teleport prob



$\sum_i \pi_i = 1 \quad i = 1 \dots 4$

$\pi_1 = \frac{\alpha}{4} + \frac{(1-\alpha)}{3} \pi_4$

$\pi_2 = \pi_3$ by symmetry

$\pi_2 = \frac{\alpha}{4} + (1-\alpha)\left(\frac{\pi_1}{3} + \frac{\pi_4}{3}\right)$

$\pi_3 = \frac{\alpha}{4} + \frac{(1-\alpha)}{3}(\pi_1 + \pi_4)$

$\pi_4 = \frac{\alpha}{4} + (1-\alpha)\left(\pi_2 + \pi_3 + \frac{\pi_1}{3}\right)$

(b) assume that the teleport prob is modified: at every step, with prob α the random surfer jumps to V_1.
Is the resulting process still a Markov chain? (Motivate the answer)

In a markov chain the probability to go on a certain node depends only on the current state

Quindi è markov chain o no ?

Siccome è una modifica al personalization vector, che passa da $\{\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\}$ a $\{1, 0, 0, 0\}$, quindi significa che la struttura del grafo è rimasta la stessa e non ha violato nessuna proprietà delle markov chain. (perché la probabilità di fermarsi in un nodo continua a essere indipendente dagli stati precedenti ma dipende da quello corrente.

i) Describe the assumption of a Naive Bayes in the bag of words model

(a) The bag of words model is one of the simplest language models used in NLP. It makes a unigram model of the text by keeping track of the number of occurrences of each word.
In the bag of words model you only take individual words into account and give each word a specific subjectivity score
The Naive Bayes model is also based on the bag of words model.
In Naive bayes, the assumption is that the attribute values are conditionally independent.

(b)

(b) compute the coefficient of a boolean classifier without smoothing

(a) browsing tiger safari.	apple
(b) africa video iran	not apple
(c) iran mountain osx.	apple
(d) mountain safari browsing tiger	not apple

Senza smoothing significa che non devo considerare i fattori di correzione a numeratore e denominatore

$P(apple) = \frac{1}{2}$ $P(not\ apple) = \frac{1}{2}$

$P(browsing/apple) = \frac{1}{6}$ $P(tiger/apple) = \frac{1}{6}$ $P(safari/apple) = \frac{1}{6}$

$P(iran/apple) = \frac{1}{6}$ $P(mountain/apple) = \frac{1}{6}$ $P(osx/apple = \frac{1}{6}$

$P(africa/notapple) = \frac{1}{7}$ $P(video/not\ apple) = \frac{1}{7}$ $P(iran/notapple) = \frac{1}{7}$

$P(mountain/not\ apple) = \frac{1}{7}$ $P(safari/notapple) = \frac{1}{7}$ $P(tiger/not\ apple) = \frac{1}{7}$

$P(browsing/not\ apple) = \frac{1}{7}$

(c) classify the query document: iran mountain safari

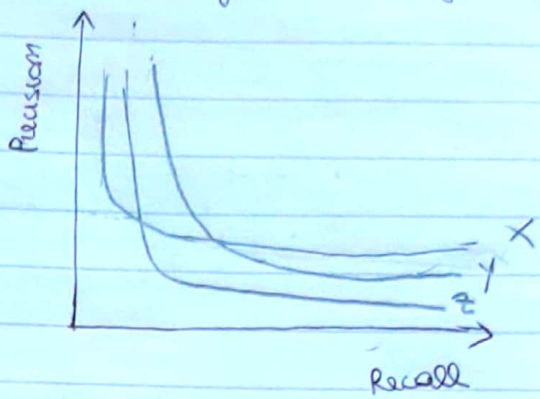
$P(apple/testdocumet) = P(apple) \cdot P(iran/apple) \cdot P(mount/apple) \cdot P(safari/apple) = \frac{1}{2} \cdot \frac{1}{6} \cdot \frac{1}{6} \cdot \frac{1}{6} = 0,027$

$P(notapple/testdocument) = P(notapple) \cdot P(iran/notapp) \cdot P(mount/notapp) \cdot P(safari/notapp) = \frac{1}{2} \cdot \frac{1}{7} \cdot \frac{1}{7} \cdot \frac{1}{7} = \frac{1}{42} = 0,024$

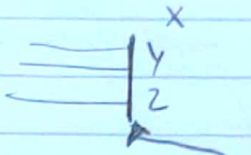
$P(risposte/testdocument) = argmax \Rightarrow 0,027$, quindi la query viene classificata come apple

EXAM 21/04/2021

1) Describe to obtain the picture on the right comparing ~~search~~ engines
X (blue) Y (red) Z (green)



TUTTI I DOCUMENTI RILEVANTI → RECALL.



mi interessa la recall

(2) recall è uguale per tutti, ma bisogna prendere quella con precision più alta per evitare NOISE.
Quindi la scelta giusta è la X

Se vuole prendere all the relevant docs we care about recall.

See of the systems where 100% of relevant document because recall is 1.

mi posizionerò alla fine

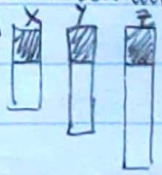
as the others

The difference is that X retrieves all relevant docs but X has retrieved less ~~noise~~ non-relevant documents perché la precision è più alta.

---

(3) providing only relevant documents.
(aumenta la precisione, perché tra quelli rilevanti vuole maggioranza di relevant.

quindi tutti hanno preso tutti i documenti rilevanti ma X ha preso anche pochi non rilevanti (meno NOISE) :)



NOISE: amount of not relevant documents

If you want to retrieved at most 50% of ~~documents~~ relevant docs Y → perché ha precisione alta

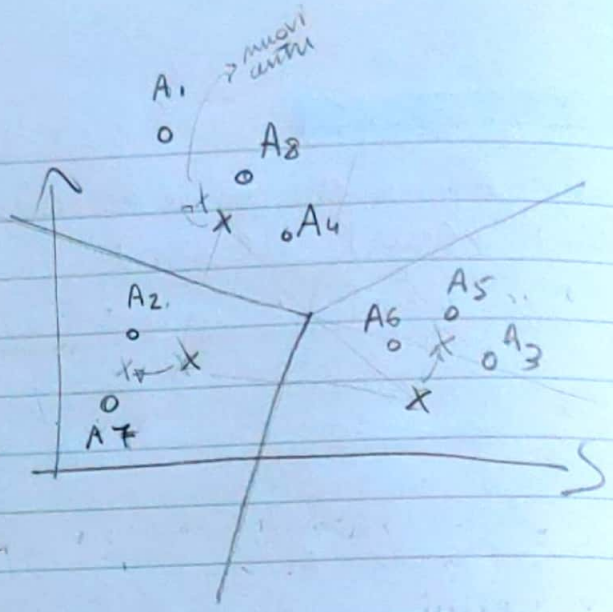
If you want ~~at most 50% X~~ more than 50%, best solution is X because precision is higher.

↳ mi interessa il rapporto tra doc rilevanti e doc totali trovati, quindi uso precisione
(non mi interessa quanti rilevanti prendo ma mi interessa solo che siano rilevanti)

2) K-means per 3 classi

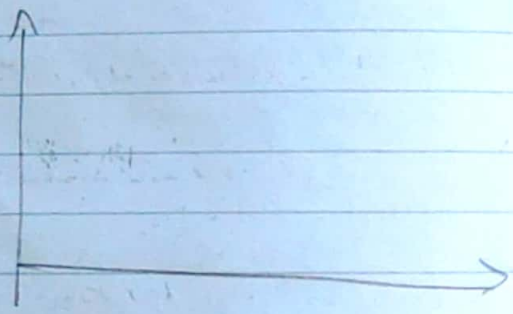
1) ~~2 distanze~~ perpendicolare alla distanza dei 2 centroidi

Initially the centroids are given random, then we consider the distance between them and find the ~~perpendicular bisect~~ perpendicular bisect between them. Then the centroids are repositioned at the center of the ~~new~~ cluster and proceed until convergence



2) A_1, A_2, A_4 rossi
A_5, A_6 blu
2 classi

(In questo caso uso KNN e classifico quelli che stanno in base a quanti elementi di una certa classe ci sono nel raggio dei centroidi)



3) D_1 = he moved from London Ontario to London, England
D_2 = " " " England " " Ontario
D_3 = " " England to ~~Ont~~ London Ontario

1) which docs have the same representation in the bag of words model? ~~Bernoulli~~ model
You represent docs as sets of words.

$\tilde{D}_1$ = {He, moved, from, London ....} we do not count the occurrences

$\tilde{D}_2 = \tilde{D}_3$ NON CONTO LE OCCORRENZE

$\tilde{D}_1 = \tilde{D}_2 = \tilde{D}_3$

(perché con bernoulli voglio solo vedere se un dato appartiene a una certa classe, non quante volte compare)

EXAM 2012-09-10

(2) Vector space Model = multinomial

$\tilde{D}_1 = \{He, moved, from, London\}$ ma abbiamo London con 2

3 modi di scrittura

$\tilde{D}_1 = \{He, moved, from, London, London \dots\}$ CONTA LE OCCORRENZE

$\tilde{D}_1 = \{(He,1),(moved,1),(from,1),(London,2)\dots\} = \tilde{D}_2$

$\tilde{D}_1 = [1, 1, 1, 2, 1 \dots]$

D_2 è uguale

mentre D_3 è diverso perché London compare una volta

(3) suppose D_1, D_2, D_3 all belong to the same class "c"

$P(t/c)$ according to the Bernoulli model

$$P(t/c) = \frac{P(t) \cdot P(c/t)}{P(c)}$$

⇒ BAYES THEOREM

assumptions: all docs belong to same class, quindi $P(c) = 1$

$P(c/t) = 1$ perché so che il termine t compare in tutti i documenti e tutti i docs ∈ c

So $P(t/c) = \frac{P(t) \cdot 1}{1} = P(t)$

in the bag of word model you have to take the word and divide per il numero di elementi del set (perché in Bernoulli non conto le occorrenze)

$P(t/c) = \frac{1}{7}$

e assumo inizialmente

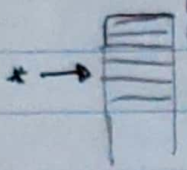
ESERCIZIO PROF

Suppose that you want to answer queries containing wildcards like: R*ome, R*l

Describe an algorithm that will process and solve those wildcards queries.

R*me → controlla quali sono le entryes che possono matchare le parole

considereremo l'unione di tutte le parole che potrebbero andare bene



→ Rome
Rime
⋮

$\Rightarrow Q = \bigcup_i t_i$ (la query finale è l'unione di tutti questi termini)

We have to compute the complexity: $O(|V| \cdot |q|)$

(|V| = cardinalità di vocabolario; |q| = length della query)

—— MENO DISPENDIOSO ——

Rom*

Suppose the vocabulary is not sorted, first of all we sort it to do

binary search $O(|q| \log |V| + \#\text{ occurrences having Rom as prefix})$

~~Suppose the vocabulary is not sorted~~

*img → albero binario inverso

$O(|q| \log |V| + \#\text{occ})$

Rom*img → first Rom*, then *img

S = Rom*, T = *img

$\Rightarrow Q = S \cap T$

$O(|q| \log |V| + \#\text{occ})$

SVM: margine
la più grande distanza
tra le decision boundary e il più
vicino punto dell'insieme

NAIVE BAYES
la chain rule è troppo difficile da
risolvere se ho tanti attributi, quindi
uso la semplificazione del naive bayes