Recuperação de Informação – modelo vectorial

Paulo Trigo Silva

Ideia – cada documento é um vector

- Numa colecção de documentos existem *t* termos distintos
  - identificados depois de construído o dicionário
- Os termos formam um espaço de vectores
  - dimensão do espaço = *t* = | dicionário |
  - ... com 2 termos é bidimensional, ... com n ternos n-dimensional
- A cada termo i do documento j é atribuído um peso
  - que é um valor real ω_{ij}
- Assim, um documento d_j representa-se pelo vector
  - $d_j \equiv < \omega_{1j}, \omega_{2j}, \ldots, \omega_{tj} >$
  - ... que tem *t* dimensões

... cada interrogação também é um vector

- Uma interrogação j pode ser vista como um pequeno documento!
  - ou seja, como um vector de pesos...
  - $< \omega_{1j}, \omega_{2j}, ... , \omega_{tj} >$ (com um peso por cada uma das *t* dimensões)
- ... o que permite reformular a noção de interrogação
  - já não se pergunta “quais os documentos que contêm estes termos?”
- ... então, “dado um documento que outros lhe são semelhantes?”
  - passa a ser a pergunta para a qual se pretende resposta
- O modelo vectorial foi desenvolvido no sistema SMART
  - por Salton, 1970; depois explorado na recuperação de informação Web

Paulo Trigo Silva

Representação gráfica

Exemplo

- 3 termos: T_1, T_2, T_3
- 3 documentos: d_1, d_2, d_3

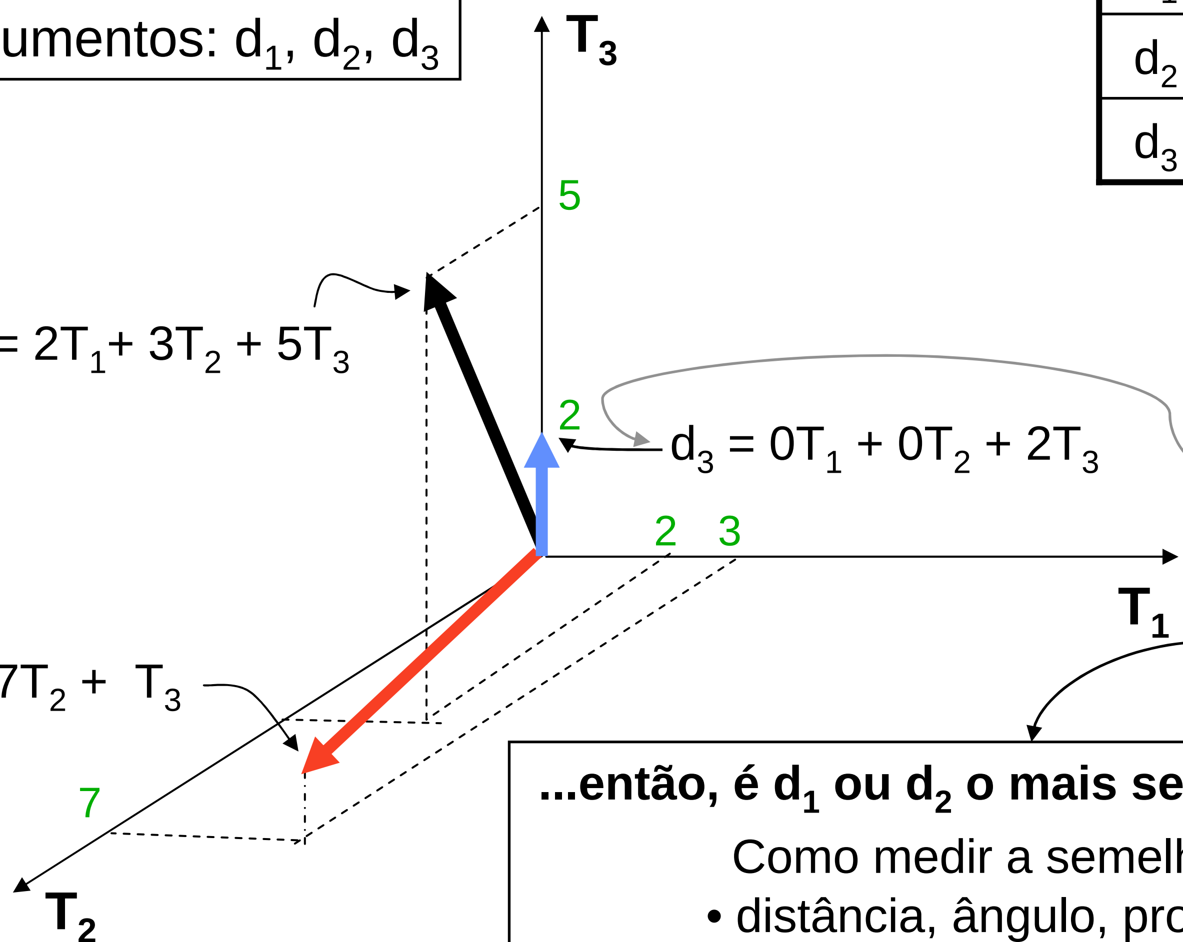
matriz de pesos

ω_{ij}	T_1	T_2	T_3
d_1	2	3	5
d_2	3	7	1
d_3	0	0	2

$d_1 = 2T_1 + 3T_2 + 5T_3$

$d_2 = 3T_1 + 7T_2 + T_3$

$d_3 = 0T_1 + 0T_2 + 2T_3$



d_3 é uma interrogação

?

...então, é d_1 ou d_2 o mais semelhante a d_3?

Como medir a semelhança?

• distância, ângulo, projecção?

Paulo Trigo Silva

Representação da colecção de documentos

- A colecção de documentos representa-se no espaço de vectores
  - como uma matriz de “termos × documentos”
- O “peso do termo no documento”
  - é representado por cada elemento da matriz
  - ... zero indica termo sem significado ou inexistente (no documento)

$$
\begin{array}{c}
 \\ d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_n
\end{array}
\begin{pmatrix}
T_1 & T_2 & \cdots & T_t \\
\omega_{11} & \omega_{21} & \cdots & \omega_{t1} \\
\omega_{12} & \omega_{22} & \cdots & \omega_{t2} \\
\vdots & \vdots & & \vdots \\
\vdots & \vdots & & \vdots \\
\omega_{1n} & \omega_{2n} & \cdots & \omega_{tn}
\end{pmatrix}
$$

Recordar:
matriz do modelo Booleano,
onde $\omega_{ij} \in \{0, 1\}$

Paulo Trigo Silva

Peso de cada termo – perspectiva da frequência

Atenção:

A literatura da recuperação de informação (RI, ou IR) usa o conceito de "frequência" para significar "quantidade".

... i.e. não se divide pelo número total de termos no documento (o que tornaria a "quantidade" numa "frequência")

Assim adoptaremos a ideia de que,

"frequência do termo no documento"

$\equiv$

"número de ocorrências do termo no documento"

Paulo Trigo Silva

Peso de cada termo – simplesmente a sua frequência?

- A forma mais simples de definir o peso de cada termo
  - é considerar que corresponde simplesmente à frequência
- Ou seja, o peso, $\omega_{t,d}$, do termo t no documento d, seria dado por
  - $\boldsymbol{\omega_{t,d} = f_{t,d}}$ (onde $f_{t,d}$ é a frequência do termo t no documento d)
- É importante manter uma perspectiva global do peso de cada termo
  - e para isso é preciso normalizar a frequência
- ... normalizar a frequência do termo corresponde a considerar
  - $\boldsymbol{\omega_{t,d} = tf_{t,d} = f_{t,d} / \max_{d \in C} \{ f_{t,d} \}}$ (onde C é a colecção de documentos)
- ... ou seja, divide-se a frequência do termo no documento pelo valor máximo da frequência daquele termo na colecção de documentos
  - obtendo um valor entre 0 e 1 que é comparável com os restantes

Exemplo – frequência normalizada do termo

Que filme! Um filme sobre como realizar um filme acerca de um gato!

`docA.txt`

"Gato branco, gato preto" é um filme (de Kusturika) com imagens surrealistas.

`docB.txt`

$C \equiv$ colecção de documentos

$$tf_{t,d} = f_{t,d} \;/\; \max_{d \in C} \{\, f_{t,d} \,\}$$

termo t	$f_{t,docA}$	$f_{t,docB}$
`gato`	1	2
`filme`	3	1

Qual é a matriz de pesos $tf_{t,d}$

i.e., qual a matriz com as frequências normalizadas?

Paulo Trigo Silva

Exemplo – frequência normalizada do termo (cont.)

Que filme! Um filme sobre como realizar um filme acerca de um gato!	"Gato branco, gato preto" é um filme (de Kusturika) com imagens surrealistas.
`docA.txt`	`docB.txt`

C ≡ colecção de documentos

frequência máxima do termo na colecção

$$tf_{t,d} = f_{t,d} \,/\, max_{d \in C} \{ f_{t,d} \}$$

termo t	$f_{t,docA}$	$f_{t,docB}$	$\mathbf{max_{d \in C} \{f_{t,d}\}}$
`gato`	1	2	**2**
`filme`	3	1	**3**

termo t	$tf_{t,docA}$	$tf_{t,docB}$
`gato`	**0.5**	**1**
`filme`	**1**	**0.3**

Paulo Trigo Silva

... frequência do termo – limitação e ideia para a "atenuar"

- A principal limitação da perspectiva da frequência do termo é que
  - todos os termos se consideram igualmente importantes
  - ... na avaliação da relevância de um documento face a uma interrogação
- Por exemplo, numa colecção sobre "apólices de seguros"
  - é natural que o termo "seguro" ocorra em todos os documentos
- ... é preciso "atenuar" o efeito dos termos que "surgem demasiado"
  - na avaliação da relevância de um documento

Ideia:

- factor para reduzir $tf_{t,d}$ (frequência do termo t no documento d), e
- redução aumenta quando aumenta o número de termos t na colecção.

Paulo Trigo Silva

... “atenuar” efeito dos termos que “surgem demasiado”

Ideia:

- factor para reduzir $tf_{t,d}$ (frequência do termo t no documento d), e
- redução aumenta quando aumenta o número de termos t na colecção.

Para construir esse factor de redução há duas medidas possíveis:

- $cf_t \equiv$ número de ocorrências do termo t na colecção, e
- $df_t \equiv$ número de documentos que têm o termo t na colecção.

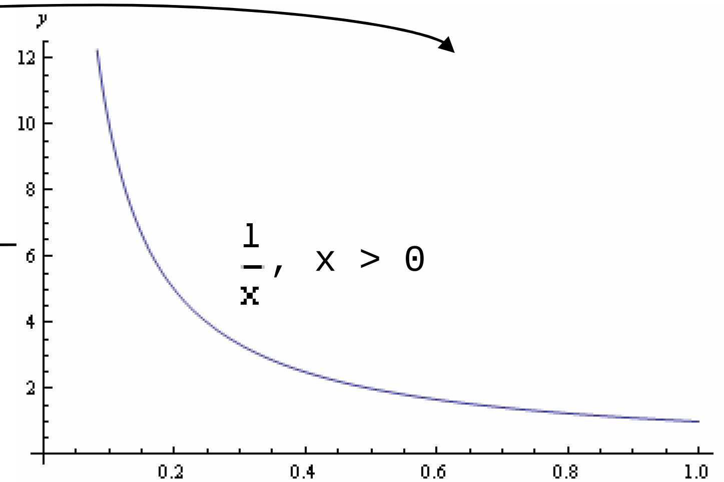
Aquelas medidas são conhecidas como:

- $cf_t \equiv$ frequência na colecção (*collection frequency*), e
- $df_t \equiv$ frequência do documento (*document frequency*).

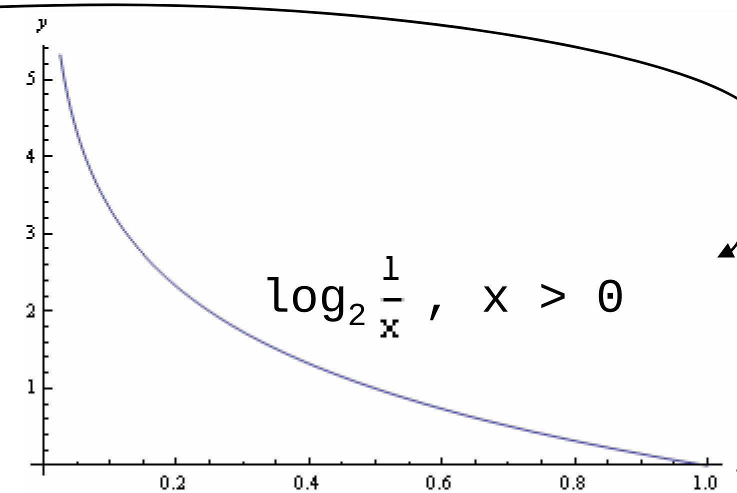
Paulo Trigo Silva

Reduzir frequência do termo

- Pode reduzir-se $tf_{t,d}$ multiplicando-o pelo inverso de uma das medidas
  - $tf_{t,d}$ x 1 / cf_t, ou
  - $tf_{t,d}$ x 1 / df_t



... mas a função 1/x é demasiado abrupta!



O seu logaritmo (de 1/x) permite:

- decrescer de modo "menos abrupto", e
- "melhor comportamento" perto de zero

Paulo Trigo Silva

... mas que medida, cf ou df, usar?

Exemplo ilustrativo

termo t	cf_t	df_t
`ferrari`	**10442**	**23**
`seguro`	**10440**	**3997**

- A frequência da colecção (cf) e a frequência do documento (df)
  - podem ter comportamento bastante diferente (um do outro)
- ... `ferrari` e `seguro` têm valor de **cf** idêntico, mas
  - o valor de **df** é bastante diferente entre eles!
- Intuitivamente, pretendemos que,
  - os poucos documentos com `ferrari` tenham maior peso (numa interrogação sobre `ferrari`) do que os que contém `seguro`.
- ... assim, é usual usar-se **df** como medida para reduzir $\mathbf{tf_{t,d}}$.

Paulo Trigo Silva

Função “inversa da frequência do documento”

Como usar a frequência do documento (**df**) para reduzir o peso de um termo?

Dado uma <u>colecção com **N** documentos</u>, considera-se:

$$\mathbf{idf_t} = \log_2 (N / df_t), \text{ para } 0 < df_t \leq N$$

Recordar:
log (1) = 0

idf$_\mathbf{t}$ ≡ inversa da frequência do documento (*inverse document frequency*)

O logaritmo torna a função inversa menos abrupta e com “crescimento regular” face ao número de documentos.

Paulo Trigo Silva

... o efeito da dimensão da colecção (e papel do logaritmo)

inverse document frequency

$$\mathbf{idf_t} = \log_2 (N / df_t)$$

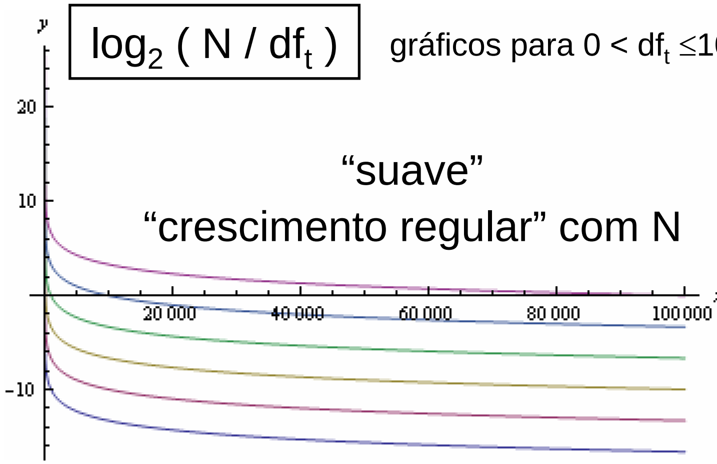
N
100 000
10 000
1 000
100
10
1

$\log_2 (N / df_t)$

gráficos para $0 < df_t \leq 100\ 000$

“suave”

“crescimento regular” com N

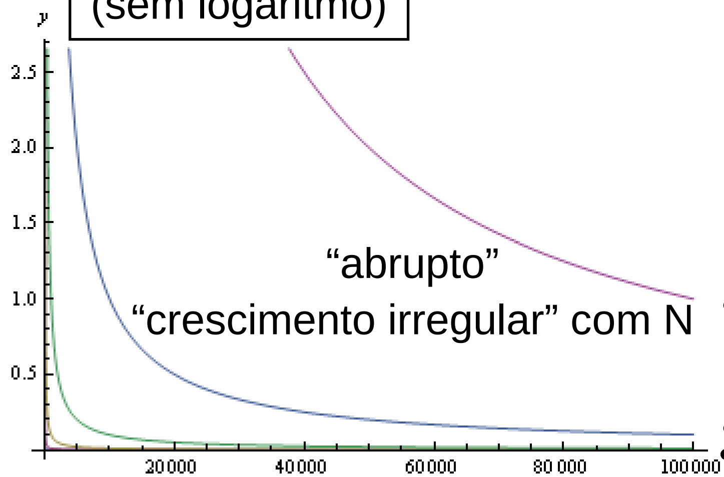


N / df_t

(sem logaritmo)

“abrupto”

“crescimento irregular” com N



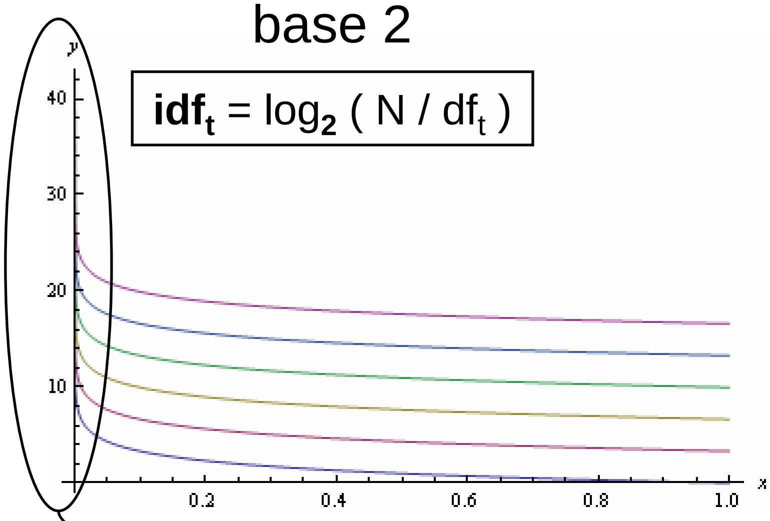
Paulo Trigo Silva

... que base usar?

O logaritmo pode usar qualquer outra base, para além da base 2 (e.g. 10), que não afecta o comportamento global da função.

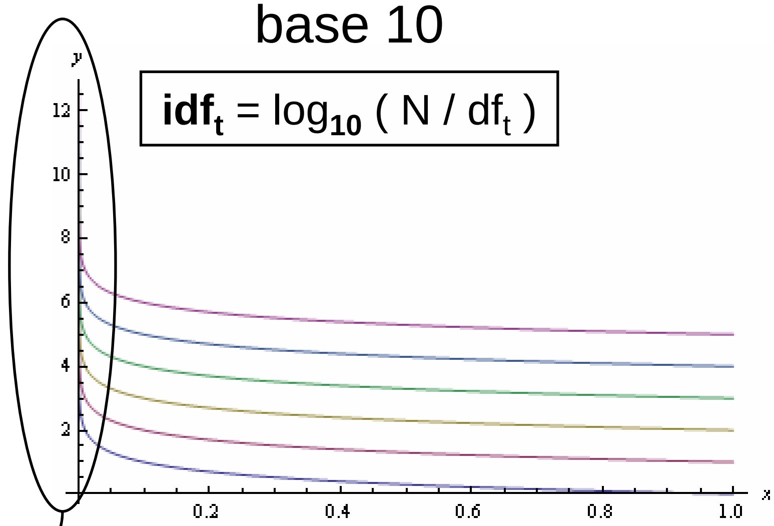
base 2

$\mathbf{idf_t} = \log_{\mathbf{2}} (N / df_t)$



base 10

$\mathbf{idf_t} = \log_{\mathbf{10}} (N / df_t)$



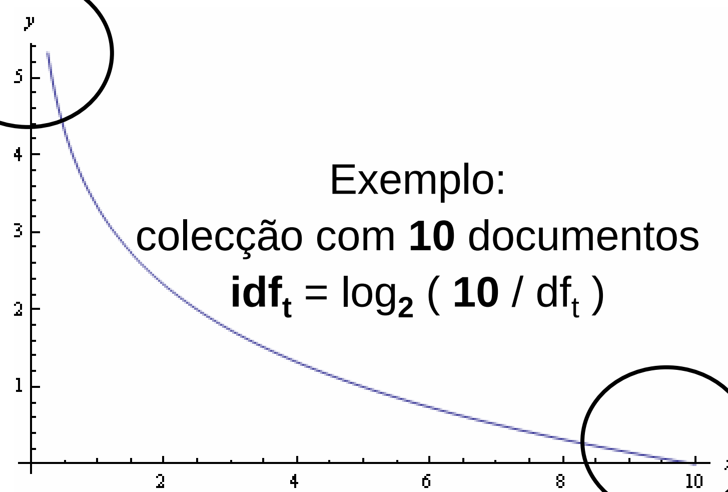
... aumentar a base reduz os valores de **idf** .

Paulo Trigo Silva

O que oferece o “idf” (*inverse document frequency*)?

O **idf_t** dá ideia da capacidade discriminatória do termo t.

É uma medida da raridade (“do quanto de raro”) do termo na colecção.



Exemplo:
colecção com **10** documentos
$\mathbf{idf_t} = \log_{\mathbf{2}} (\mathbf{10} / df_t)$

notar que o máximo valor de df_t é 10

O que é um termo, t, raro?

- é um que ocorre em pouco documentos, i.e. cujo valor de **df_t** é baixo!

O **idf_t** de um termo, t, <u>raro é alto</u>.
O **idf_t** de um termo, t, <u>frequente é baixo</u>.

Paulo Trigo Silva

“atenuar” efeito dos termos que “surgem demasiado”

... recordar a ideia inicial:

- factor para reduzir $tf_{t,d}$ (frequência do termo t no documento d), e
- redução aumenta quando aumenta o número de termos t na colecção.

$$\mathbf{tf\text{-}idf_{t,d}} = tf_{t,d} \times idf_t$$

$$f_{t,d} / \max_{d \in C} \{ f_{t,d} \}$$

$$\log_2 (N / df_t)$$

“frequência do termo t no documento d”
≡
“número de ocorrências do termo t no documento d”

número de documentos que têm o termo t na colecção

Análise teórica em: Kishore Papineni, NAACL 2, 2002.

Paulo Trigo Silva

... em síntese – o peso “tf-idf”

- Atribuir um peso, $\omega_{t,d}$, tf-idf a cada termo, t, em cada documento d

$$\boxed{\omega_{t,d} = tf_{t,d} \times \log_2 (N / df_t)}$$

  - $tf_{t,d} \equiv$ número de ocorrências do termo t no documento d
    - ◊ ...este valor pode estar normalizado
  - $N \equiv$ número total de documentos na colecção
  - $df_t \equiv$ número de documentos que, na colecção, têm o termo t

- <u>Aumenta com o número de ocorrências do termo num documento</u>

- <u>Aumenta com a “raridade” do termo na colecção de documentos</u>

Qual o peso de um termo que ocorre em todos os documentos?

$df_t = N \Rightarrow \omega_{t,d} = 0$

Em quantos documentos deve um termo ocorrer para que o seu peso tenha valor máximo?

$df_t = 1$ (domínio dos naturais) $\Rightarrow \omega_{t,d}$ máximo

Paulo Trigo Silva

Exemplo – peso “tf-idf”

Dado uma colecção com 23456 documentos:
- sejam os termos A, B e C, e
- sejam 2 documentos d1 e d2.

Considere-se que os termos A, B e C ocorrem:
- no documento d1, respectivamente 5, 5 e 2 vezes, e
- no documento d2, respectivamente 15, 1 e 6 vezes.

Considere-se que os termos A, B e C ocorrem na colecção:
- em, respectivamente 2000, 5 e 2 documentos.

Construa a matriz de “termos $\times$ documentos”
(para a informação disponível)

$$
\begin{array}{c}
\\ d_1 \\ d_2 \\ \vdots \\ d_n
\end{array}
\begin{pmatrix}
T_1 & T_2 & \cdots & T_t \\
\omega_{11} & \omega_{21} & \cdots & \omega_{t1} \\
\omega_{12} & \omega_{22} & \cdots & \omega_{t2} \\
\vdots & \vdots & & \vdots \\
\omega_{1n} & \omega_{2n} & \cdots & \omega_{tn}
\end{pmatrix}
$$

... exemplo – peso "tf-idf"

Cálculos

N = 23456

tf	A	B	C
d1	5	5	2
d2	15	1	6

df	2000	5	2
log2 (N/df)	3.55	12.20	13.52

tf x log2 (N/df)	A	B	C
d1	17.76	60.98	27.04
d2	53.28	12.20	81.11

$$\omega_{t,d} = tf_{t,d} \times \log_2 (N / df_t)$$

... não depende de d, pelo que $\omega_{t,d}$ mantém a proporção dos respectivos $tf_{t,d}$.

Por exemplo:

$tf_{C,d2} / tf_{C,d1} = 6 / 2 = \mathbf{3}$

$\omega_{C,d2} / \omega_{C,d1} = 81.11 / 27.04 = \mathbf{3}$

$\omega_{t,d}$	A	B	C
d1	17.76	60.98	27.04
d2	53.28	12.20	81.11
...	...	...	...
d23456	...	...	...

Mas, para um documento, d, altera-se muito a relação entre os pesos, $\omega_{t,d}$, dos seus termos t.

Por exemplo:

$tf_{B,d1} / tf_{A,d1} = 5 / 5 = \mathbf{1}$

$\omega_{B,d1} / \omega_{A,d1} = 60.98 / 17.76 = \mathbf{3.43}$

A noção de proximidade

- Temos técnica para construir uma matriz de pesos
  - calcula-se tf-idf$_{t,d}$ e fica-se com um espaço de vectores
  - ... representados como uma matriz "termos $\times$ documentos"

$$\begin{array}{c|cccc} & T_1 & T_2 & \cdots & T_t \\ \hline d_1 & \omega_{11} & \omega_{21} & \cdots & \omega_{t1} \\ d_2 & \omega_{12} & \omega_{22} & \cdots & \omega_{t2} \\ \vdots & \vdots & \vdots & & \vdots \\ d_n & \omega_{1n} & \omega_{2n} & \cdots & \omega_{tn} \end{array}$$

- Mas, como usar esse espaço para escolher documentos?
  - cada documento é um vector de pesos, portanto
  - ... como comparar um vector (documento) com outro vector?

- Como decidir quanto à <u>proximidade entre vectores</u> (documentos)?
  - baseada na distância de Manhattan?
  - baseada na distância euclidiana?
  - baseada no produto interno entre vectores?
  - baseada no valor do co-seno do ângulo entre vectores?
  - ... as 2 anteriores podem ver-se como uma única (o co-seno pode ver-se como produto interno normalizado)

Paulo Trigo Silva

Abordagem 1 – distância de Manhattan

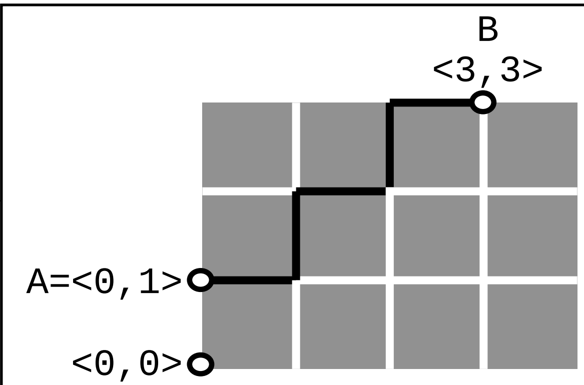
Distância de Manhattan, ou distância de blocos.

Inspira-se na ideia de que as cidades Americanos têm um formato em grelha.

Distância de Manhattan entre

A=<0,1> e B=<3,3>

3 – 0 + 3 – 1 = 3 + 2 = 5



Distância de Manhattan

$$ManhDist(X,Y) = \sum_{i=1}^{n} | x_i - y_i |$$

Calcular distância de Manhattan entre

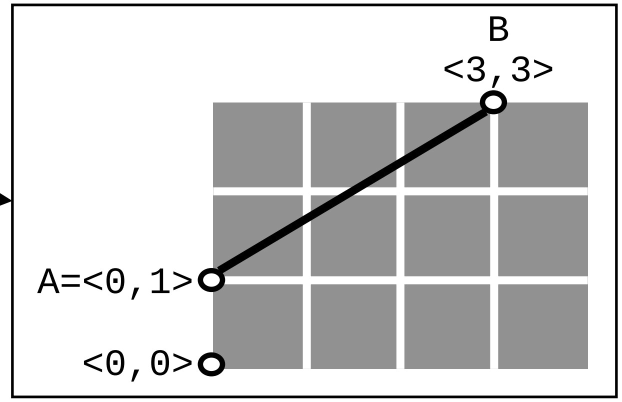
A=<0,3,2,1,10> e B=<2,7,1,0,0>

|0-2|+|3-7|+|2-1|+|1-0|+|10-0| = 18

Abordagem 2 – distância euclidiana

Distância em linha recta entre dois pontos.

Distância euclidiana entre
A=<0,1> e B=<3,3>
[(3 – 0)^2 + (3 – 1)^2]$^{1/2}$ = [9 + 4]$^{1/2}$ = 3.6



Distância euclidiana entre os documentos (vectores) d_j e d_k

$$\left| d_j - d_k \right| = \sqrt{\sum_{i=1}^{n} \left(d_{i,j} - d_{i,k} \right)^2}$$

Calcular distância euclidiana entre
d_1=<a,b,c> e d_2=<x,y,z>

$$\sqrt{\mathrm{Abs}[a-x]^2 + \mathrm{Abs}[b-y]^2 + \mathrm{Abs}[d-z]^2}$$

Paulo Trigo Silva

... limitações – abordagem 1 e 2

- As métricas sobre distância sofrem grande influência da
  - da dimensão do documento
- Documentos pequenos tendem a ser próximos não pelo conteúdo
  - mas pela sua dimensão
- Para comparar distâncias é preciso normalizar

$$\left|\vec{d}_j\right| = \sqrt{\sum_{i=1}^{n} w_{i,j}^2}$$

  - dividindo cada componente pelo módulo (tamanho) do vector
    - ◊ ... i.e. para um documento d e dados k pesos, $\omega_{1,d}$, ..., $\omega_{k,d}$, fazer
    - ◊ $\omega_{i,d} / (\omega_{1,d}^2 + ... + \omega_{k,d}^2)^{1/2}$, para cada $1 \leq i \leq k$
  - ao normalizar fica-se, para cada documento d_j, com
    - ◊ $|d_j| = (\omega_{1,d}^2 + ... + \omega_{k,d}^2)^{1/2} = \mathbf{1}$ (vector projectado em esfera de raio 1)
- ... mas, olhando apenas para os ângulos (em vez das distâncias)
  - é também possível obter uma perspectiva normalizada

Abordagem 3 – produto interno

O produto interno (ou escalar) entre dois vectores dá uma medida do "peso" (ou força) daqueles vectores quando projectados numa mesma direcção.

- A semelhança (similaridade) entre documentos pode medir-se pelo
  - produto interno dos seus vectores
- ... dado um documento d_j e interrogação q (também é documento)

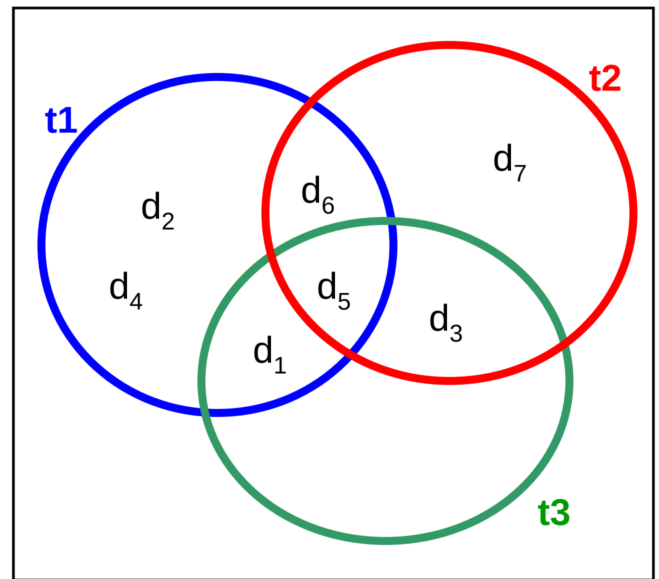
$$\boxed{\mathrm{sim}(d_j,q) = d_j \bullet q = \sum_{i=1}^{t} \omega_{ij} \times \omega_{iq}}$$

  - onde ω_{iq} é o peso do termo i na interrogação q
- Para vectores binários dá o número de termos comuns em d_j e q
  - corresponde à dimensão da intersecção dos conjuntos termos
- Para vectores com pesos reais dá a soma dos produtos dos pesos
  - dos termos que existem em ambos os documentos

Paulo Trigo Silva

... exemplo – produto interno (vectores binários)

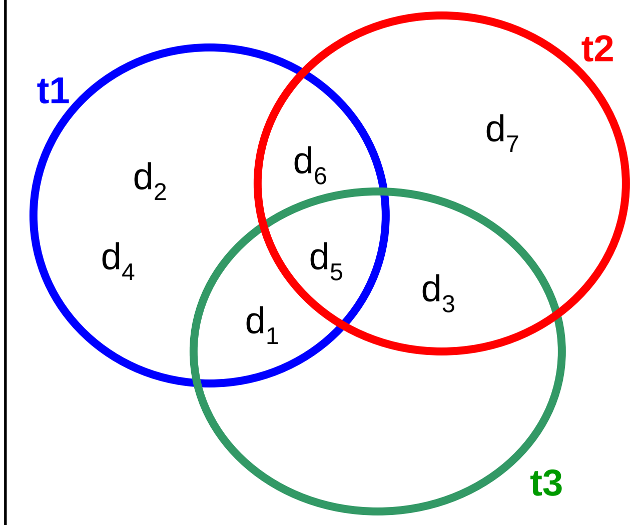
	t1	t2	t3	**q • dj**
d_1	1	0	1	**2**
d_2	1	0	0	**1**
d_3	0	1	1	**2**
d_4	1	0	0	**1**
d_5	1	1	1	**3**
d_6	1	1	0	**2**
d_7	0	1	0	**1**
q	1	1	1	



Paulo Trigo Silva

... exemplo – produto interno (limitações)

	t1	t2	t3	**q • dj**
d_1	1	0	1	**4**
d_2	1	0	0	**1**
d_3	0	1	1	**5**
d_4	1	0	0	**1**
d_5	1	1	1	**6**
d_6	1	1	0	**3**
d_7	0	1	0	**2**
q	1	**2**	**3**	

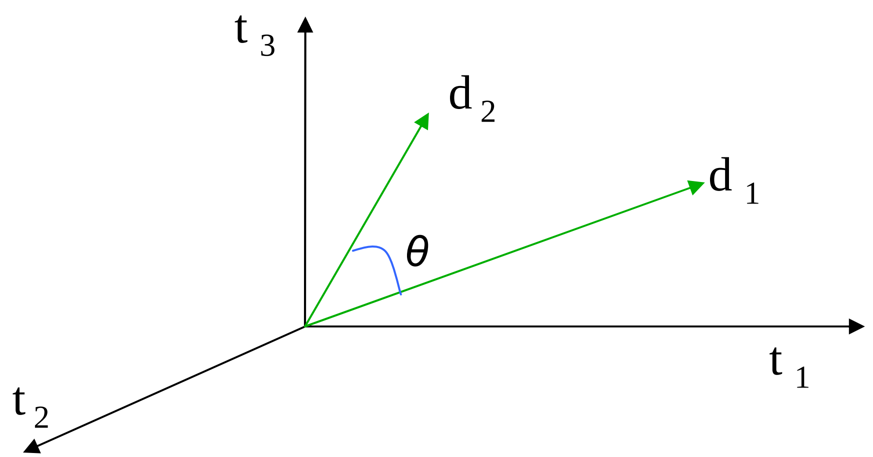


Favorece documentos longos com grande número de termos únicos repetidos.

Favorece o número de termos comuns nos documentos (d_j e q), mas não contempla o número de termos que não são comuns.

Paulo Trigo Silva

.. termos, documentos e ângulos



t_1, t_2 e t_3 são termos;

d_1 e d_2 são documentos e portanto têm um peso para cada termo;

d_1 e d_2 são portanto vectores de pesos,

e.g., $d_1 = 15t_1 + 4t_2 + 0t_3 = <15, 4, 0>$

e.g., $d_2 = 2t_1 + 6t_2 + 3t_3 = <2, 6, 3>$

$d_1 \bullet d_1 = 15 \times 2 + 4 \times 6 + 0 \times 3 = 54$

... o produto interno depende da dimensão dos vectores.

Mas o ângulo formado pelos vectores não depende de nenhuma dimensão!

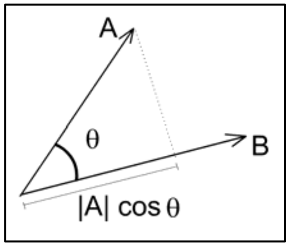
... mais sobre o produto interno

- O produto interno (ou escalar) de vectores usa-se para determinar
  - a componente escalar de um vector numa determinada direcção
  - ... também designado por “projecção” do vector nessa direcção

- O produto interno entre os vectores A e B é dado por

$$\boxed{A \bullet B = |\,A\,|\,|\,B\,| \cos(\theta)}$$

$\theta \equiv$ ângulo entre A e B



- A • B não é um vector; é um escalar
  - $A \bullet B = 0$, para vectores perpendiculares
  - $A \bullet B = |\,A\,|\,|\,B\,|$, para A e B colineares com mesmo sentido ($\cos(0) = 1$)
  - $A \bullet B = -\,|\,A\,|\,|\,B\,|$, para A e B colineares de sentido inverso ($\cos(\pi) = -1$)
  - ... ou seja, o produto interno varia em: $-\,|\,A\,|\,|\,B\,| \leq A \bullet B \leq |\,A\,|\,|\,B\,|$

ângulo nulo → co-seno = 1 → **semelhança máxima**

ângulo recto → co-seno = 0 → **diferença máxima**

Paulo Trigo Silva

Abordagem 4 – ângulo entre vectores

- A partir do valor do produto interno é simples calcular
  - o valor do co-seno do ângulo formado pelos vectores, e.g. A e B

$$A \bullet B = |A|\,|B| \cos(\theta)$$
$$\Leftrightarrow$$
$$\cos(\theta) = (A \bullet B) / (\boxed{|A|}\boxed{|B|})$$

Recordar, módulo do vector $d_j = <\omega_{1j}, ..., \omega_{nj}>$

$$\left|\vec{d}_j\right| = \sqrt{\sum_{i=1}^{n} w_{i,j}^2}$$

- Ou seja, o co-seno pode ver-se como o produto interno normalizado
  - ... A / | A | transforma A num vector de módulo 1 (e o mesmo para B)

- Em síntese, a similaridade (semelhança) entre documentos, d_j e d_k,
  - pode calcular-se como *sim(*d_j*,* d_k*)* sendo o valor do co-seno entre d_j, e d_k,

$$sim(d_j, d_k) = \frac{\vec{d}_j \cdot \vec{d}_k}{\left|\vec{d}_j\right|\left|\vec{d}_k\right|} = \frac{\sum_{i=1}^{n} w_{i,j} w_{i,k}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2}\sqrt{\sum_{i=1}^{n} w_{i,k}^2}}$$

$\omega_{t,d}$ é o peso do termo t no documento d

Paulo Trigo Silva

Exemplo – dois documentos e uma interrogação

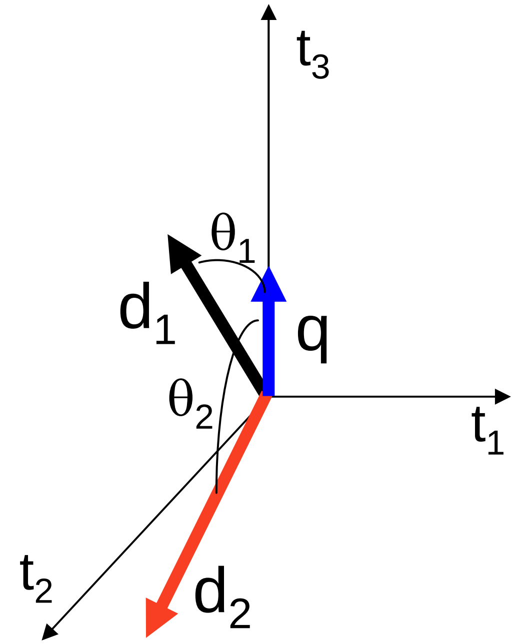
$$sim(\,d_j,\,q\,) = \frac{\vec{d}_j \bullet \vec{q}}{\left|\vec{d}_j\right| \cdot \left|\vec{q}\right|} = \frac{\sum_{i=1}^{t} (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^{t} w_{ij}^{2} \cdot \sum_{i=1}^{t} w_{iq}^{2}}}$$

Exemplo
- 3 termos: t_1, t_2, t_3
- 3 documentos: d_1, d_2, q

$d_1 = 2t_1 + 3t_2 + 5t_3$
$d_2 = 3t_1 + 7t_2 + t_3$
$q = 0t_1 + 0t_2 + 2t_3$

$d_1 \bullet q = 2 \times 0 + 3 \times 0 + 5 \times 2 = \mathbf{10}$
$d_2 \bullet q = 3 \times 0 + 0 \times 0 + 1 \times 2 = \mathbf{2}$

$sim(\,d_1,\,q\,) = 10 / [\,(4 + 9 + 25)\,(0 + 0 + 4)\,]^{1/2} = \mathbf{0.81}$
$sim(\,d_2,\,q\,) = 2 / [\,(9 + 49 + 1)\,(0 + 0 + 4)\,]^{1/2} = \mathbf{0.13}$



Similaridade dos documentos em relação à interrogação q:

- <u>usando co-seno</u>: d_1 é <u>6.23 vezes melhor</u> que d_2
- <u>usando produto interno</u>: d_1 é <u>apenas 5 vezes</u> melhor que d_2

Paulo Trigo Silva

... outros exemplos

Ordenar por ordem crescente de similaridade (medida pelo co-seno):

- (A) dois documentos que só têm palavras usuais em comum
  - e.g. “de”, “para”, “um”, ...
- (B) dois documentos que não têm palavras em comum
- (C) dois documentos que têm muitas palavras raras em comum,
  - e.g. “esferoidal”, “decangular”, ... (tem forma esferóide e tem 10 ângulos)

$$sim(d_j, d_k) = \frac{\vec{d}_j \cdot \vec{d}_k}{\left|\vec{d}_j\right|\left|\vec{d}_k\right|} = \frac{\sum_{i=1}^{n} w_{i,j} w_{i,k}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2}\sqrt{\sum_{i=1}^{n} w_{i,k}^2}}$$

A relação de ordem seria:

• (**C**) numerador cresce com as raras; provavelmente também terá usuais.
• (**A**) numerador só cresce com usuais.
• (**B**) numerador é zero.

Paulo Trigo Silva

Exemplo – co-seno com vectores normalizados

Se os vectores estiverem normalizados, então:

$$\cos(\vec{d}_j, \vec{d}_k) = \vec{d}_j \cdot \vec{d}_k$$

Os pesos, ω_{ij}, são simplesmente $tf_{t,d}$ (número de ocorrências do termo t em d)

		affection	jealous	gossip	| dj |
Sense and Sensibility	d1	115	10	2	115.451
Pride and Prejudice	d2	58	7	0	58.421
Wuthering Heights	d3	20	11	6	23.601

Recordar, módulo do vector $d_j = <\omega_{1j}, ..., \omega_{nj}>$

$$\left|\vec{d}_j\right| = \sqrt{\sum_{i=1}^{n} w_{i,j}^2}$$

		affection / |dj|	jealous / |dj|	gossip / |dj|
Sense and Sensibility	d1	0.996	0.087	0.017
Pride and Prejudice	d2	0.993	0.120	0.000
Wuthering Heights	d3	0.847	0.466	0.254

Recordar, normalizar corresponde a dividir cada componente, ω_{ij}, de d_j pelo módulo de d_j.

cos(**d1, d2**) = .996 × .993 + .087 × .120 + .017 × 0.0 = **0.999**

cos(**d1, d3**) = .996 × .847 + .087 × .466 + .017 × .254 = **0.889**

d1 e d2 ambos de Jane Austin; d3 é de Emile Bronte;

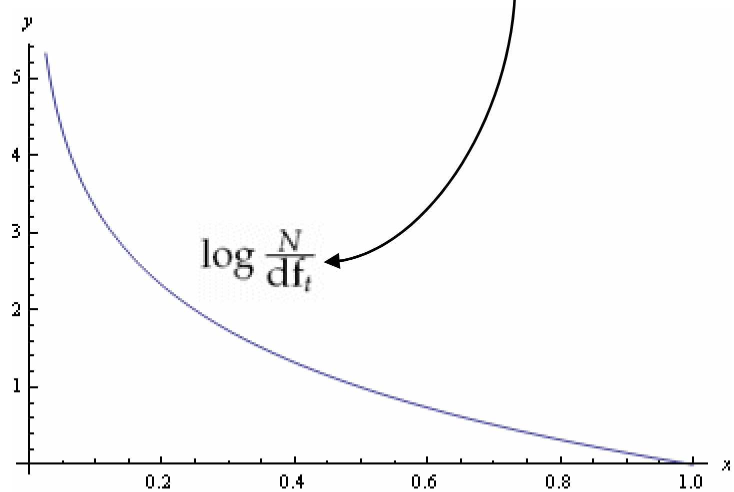
d1 e d2 estão muito próximos!

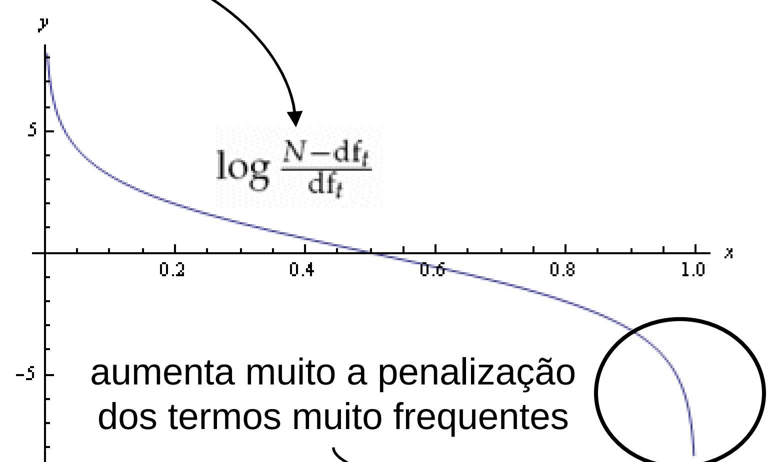
Síntese – “td”, “df”, “tf-idf” e normalização (várias métricas)

$tf_{t,d}$ ≡ número de ocorrências do termo t em d

df_t ≡ número de documentos que, na colecção, têm o termo t

Term frequency		Document frequency		Normalization	
n (natural)	$\mathrm{tf}_{t,d}$	n (no)	1	n (none)	1
l (logarithm)	$1+\log(\mathrm{tf}_{t,d})$	t (idf)	$\log\frac{N}{\mathrm{df}_t}$	c (cosine)	$\frac{1}{\sqrt{w_1^2+w_2^2+\ldots+w_M^2}}$
a (augmented)	$0.5+\frac{0.5\times\mathrm{tf}_{t,d}}{\max_t(\mathrm{tf}_{t,d})}$	p (prob idf)	$\log\frac{N-\mathrm{df}_t}{\mathrm{df}_t}$		
L (log ave)	$\frac{1+\log(\mathrm{tf}_{t,d})}{1+\log(\mathrm{ave}_{t\in d}(\mathrm{tf}_{t,d}))}$			b (byte size)	$1/charLeng^{\alpha},\ \alpha<1$
b (boolean)	$\mathrm{tf}_{t,d}>0$				





aumenta muito a penalização dos termos muito frequentes

Paulo Trigo Silva

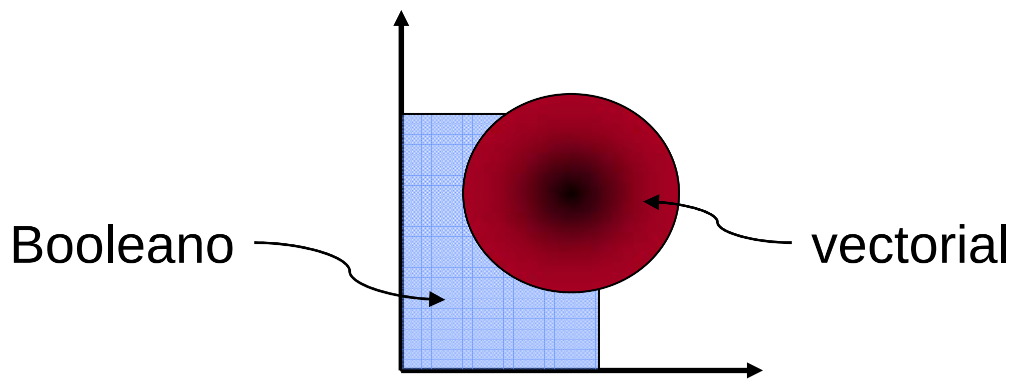
Comentários ao modelo vectorial

- Abordagem baseada em conceitos formais (matemáticos) simples
  - mas com muito adequados ao problema em causa
- Considera perspectiva local (tf) e global (idf) da ocorrência de termos
  - a combinação das perspectivas (tf-idf) reduz limitação de ambas
- Permite obter respostas parciais
  - pode desprezar documentos com proximidade quase nula
- Permite ordenar as respostas
  - de acordo com o valor de proximidade face a determinada interrogação
- ... na prática permite obter bons resultados
  - quanto à cobertura e relevância das respostas (a detalhar mais tarde)

Paulo Trigo Silva

... interrogações no modelo Booleano ‘versus’ vectorial

- Os modelos vectorial e Booleano não funcionam muito bem juntos!
  - a noção de similaridade não “combina bem” com a de presente/ausente
- No espaço dos termos a similaridade de vectores define esferas
  - e.g. “todos os documentos com co-seno $\geq$ 0.5” face à interrogação
- A interrogação Booleana devolve hiper-rectângulos
  - e suas intersecções e uniões
- ... “entalhe rectangular” versus “cavilha redonda”



Paulo Trigo Silva

Implementação do modelo vectorial – na versão ‘naïve’

Input: uma colecção C de documentos e uma interrogação q
Output: uma lista ordenada de documentos

- para cada documento $d_j \in C$
  - converter d_j num vector de pesos tf-idf
- converter a interrogação q num vector de pesos tf-idf
- para cada documento $d \in C$
  - calcular $score_j = sim(d_j, q)$
- ordenar documentos por ordem decrescente de $score_j$
- apresentar, ao utilizador, os documentos de topo

Complexidade temporal: O(|T|·|C|); mau para grandes T & C !

|T| = 10,000; |C| = 100,000; |T|·|C| = 1,000,000,000

Implementação do modelo vectorial – na prática

- Os documentos que não têm qualquer termos da interrogação
  - não afectam o resultado da interrogação!
- Tentar identificar os documentos que contêm
  - pelo menos 1 termo da interrogação
- Usar índices invertidos para suportar a pesquisa

Implementar funções de pré-processamento

- “tokenization” (eliminar pontuações, espaços, etc)
- remoção de “stop words” (“de”, “e”, “com”, etc)
- “steamming” (passar termos para forma canónica)
  - e.g. “casa”, “casinha”, “casarão” → “casa”

Objectivo:

Reduzir o espaço da pesquisa!

Tornar eficiente o cálculo dos co-seno

- Encontrar K documentos na colecção
  - aqueles que estão mais próximos da interrogação
- ... implica calcular os K maiores co-senos entre
  - a interrogação e K documentos
- Problema:
  - calcular, ou garantir a similaridade, do co-seno de forma eficiente
  - escolher os k co-seno de forma eficiente

<u>Objectivo</u>:

Escolher os K maiores co-senos sem ter que os calcular todos!

Paulo Trigo Silva

Garantir similaridade do co-seno de forma eficiente

- Numa interrogação sem termos repetidos
  - o valor normalizado dos termos é igual!
- ... na interrogação pode considerar-se
  - que os seus termos têm valor 1 (i.e. não se normaliza)

q	gato	rato	rei
	1	1	1

|q|
1.73

$\frac{\text{gato}}{\|q\|}$	$\frac{\text{rato}}{\|q\|}$	$\frac{\text{rei}}{\|q\|}$
0.58	0.58	0.58

- Para quaisquer dois documentos d_1 e d_2 tem-se,

$$\boxed{\vec{V}(q)\cdot\vec{v}(d_1) > \vec{V}(q)\cdot\vec{v}(d_2) \Leftrightarrow \vec{v}(q)\cdot\vec{v}(d_1) > \vec{v}(q)\cdot\vec{v}(d_2).}$$

vector de pesos

vector apenas com 1s e 0s

para provar basta dividir ambos os termos da desigualdade por |q| (q / |q| é q normalizado)

- Para qualquer documento d, calcular $\boxed{\vec{V}(q)\cdot\vec{v}(d)}$ corresponde
  - apenas a <u>somar os pesos de d que constam da interrogação q</u>
- Assim, <u>garante-se relação de ordem sem calcular o co-seno</u>!

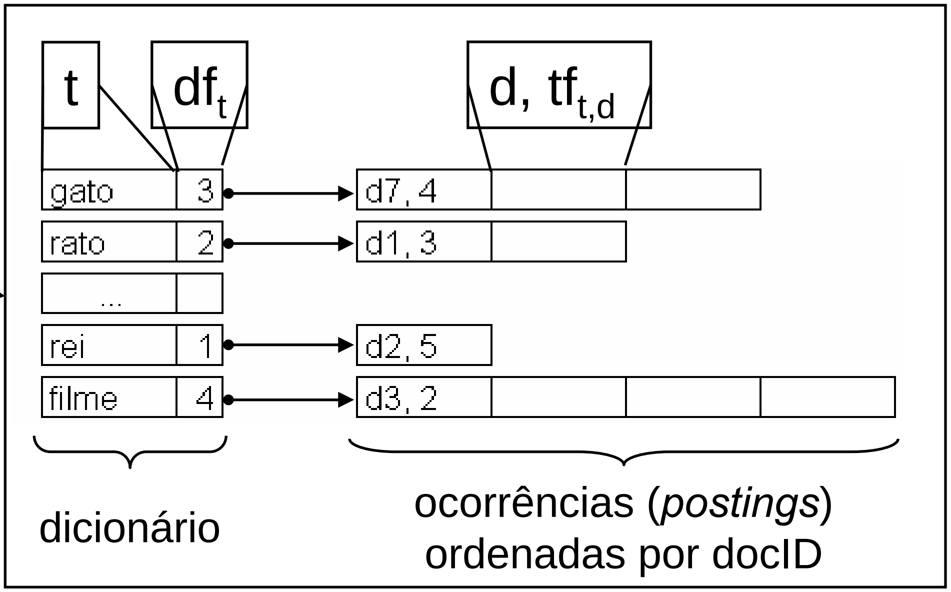
Paulo Trigo Silva

... calcular similaridade de forma eficiente

Recordar:
$\text{tf-idf}_{t,d} = \omega_{t,d} = \text{tf}_{t,d} \times \log_2 (N / \text{df}_t)$

- Manter uma lista de índices invertidos
  - com informação sobre o valor de “df” e “tf”

- Dado uma interrogação q
  - calcular $\boxed{\vec{V}(q) \cdot \vec{v}(d)}$ percorrendo as ocorrências de cada termo, t, em q
  - acumulando, para cada documento d, o valor “tf-idf” de cada termo t



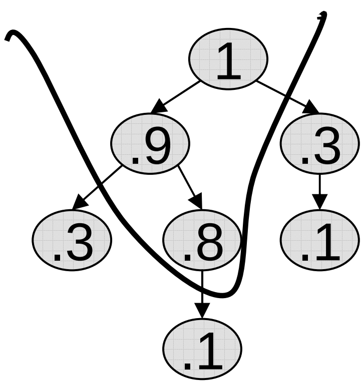
Processo idêntico ao do modelo Booleano!

No entanto, aqui é atribuída uma pontuação (score) a cada documento.

Paulo Trigo Silva

Escolher os K co-seno de forma eficiente

- Uma abordagem consiste em ordenar todos os co-seno
  - e escolher os K maiores
  - ... melhor algoritmo de ordenação tem custo pior caso O(N log N)
- Outra abordagem mais eficiente é a de manter uma estrutura
  - cujo custo de manutenção compense depois na procura dos K co-seno
- Manter uma árvore binária onde
  - o valor de cada nó é maior do que a dos seus filhos (“heap”)
- ... dados N documentos,
  - construir a “heap” tem custo pior caso O(2N)
  - obter os K nós tem custo pior caso O(2 log N)



Paulo Trigo Silva

... comparar ordenação com manutenção de “heap”

Com ordenação:

ordenar: O(N log N)

obter K co-senos: O(1)

Total: (N log N)

Com árvores (“heap”):

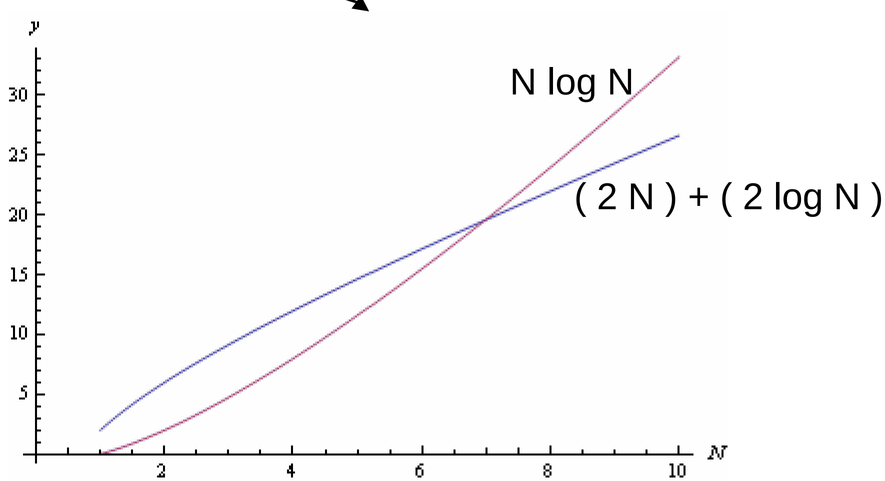
ordenar: O(2 N)

obter K co-senos: O(2 log N)

Total: (2 N) + (2 log N)

(N log N) cresce mais depressa, a partir de certa altura, do que (2 N) + (2 log N)

Para valores grandes de N (dimensão da colecção) compensa manter o “heap”



Paulo Trigo Silva

Os “mais altos” ou os “bons candidatos a mais alto”?

Duas alternativas (face a uma interrogação):

1. procurar os K co-seno mais altos na colecção
   (esta foi a abordagem seguida até agora)

2. procurar K bons candidatos a ter o maior co-seno
   (esta será a próxima abordagem)

Em geral, passar de uma pesquisa global para diversas pesquisas locais:

- reduz complexidade, mas
- não garante que se encontre a solução óptima.
- ... apesar disso as soluções sub-óptimas podem ser suficientemente boas!

Paulo Trigo Silva

Obter os K co-seno “provavelmente” mais altos

- Queremos os melhores K candidatos a ter co-seno mais alto
  - mas, podem surgir “infiltrados”!
  - ... i.e., um candidato que afinal não tem um dos K co-seno mais altos
- O risco tem o objectivo de reduzir custo de computação
  - tentando não afectar grandemente a percepção do utilizador quanto à relevância dos K documentos apresentados
- ... a própria medida de similaridade (co-seno) já é uma estimativa
  - da noção de relevância do utilizador
  - portanto uma estimativa próxima dessa pode ser suficiente
  - ... de facto a “efectiva relevância” é uma medida de cada utilizador!
- Assim, obter os K co-seno “provavelmente” mais altos
  - não é necessariamente uma abordagem pior para o utilizador
  - do que a de obter exactamente os K co-seno mais altos!

Paulo Trigo Silva

Reduzir complexidade

- Ao procurar os K melhores candidatos reduzindo complexidade
  - deixamos de fazer uma pesquisa exaustiva no espaço global
  - passamos a fazer pesquisas em espaços locais mais pequenos
- ... portanto pode acontecer que nas pesquisas locais
  - “escapem” documentos por estarem fora dos espaços pesquisados
  - “surjam” documentos com alto valor local mas baixo valor global
- Algumas das técnicas para reduzir complexidade incluem
  - heurísticas para descartar índices (“index elimination”)
  - lista campeã (“champion list”, ou “fancy list”)
  - corte por agrupamento (“cluster pruning”)

Reduzir complexidade – heurísticas para descartar índices

- Heurística 1 – limiar (“threshold”) ε para valor de “idf”
  - apenas considerar documentos com termos cujo “idf” exceda ε.
  - ... os termos com baixo “idf” passam a ser vistos como “stop words”
  - ... em geral origina grande redução no cálculo dos co-seno
- Heurística 2 – interrogação conjuntiva
  - só considerar documentos que tenham todos (ou maioria) dos termos
  - apenas para esses documentos será efectuado o cálculo do co-seno
  - ... a interrogação passa a ser vista como uma conjunção
- ... na interrogação conjuntiva podem ter que se repetir pesquisas
  - quando há menos que K candidatos para os termos considerados
  - e.g., não há K candidatos para 4 termos, repetir pesquisa para 3 termos

Reduzir complexidade – lista campeã

Técnica

lista campeã, ou "champion list", ou "fancy list".

Pré-Processamento:

1. para cada termo, t, construir o conjunto dos r documentos com maior $tf_{t,d}$;
   o valor de r é escolhido 'a priori';
   aqueles são os documentos da "lista campeã".

Interrogação q:

1. construir conjunto A com a união das listas campeãs para os termos em q.
2. calcular co-seno apenas dos documentos em A.

O valor de r deve ser alto, em relação a K, para se obterem bons resultados.

Não é necessário ter o mesmo valor de r para todos os termos do dicionário.

Paulo Trigo Silva

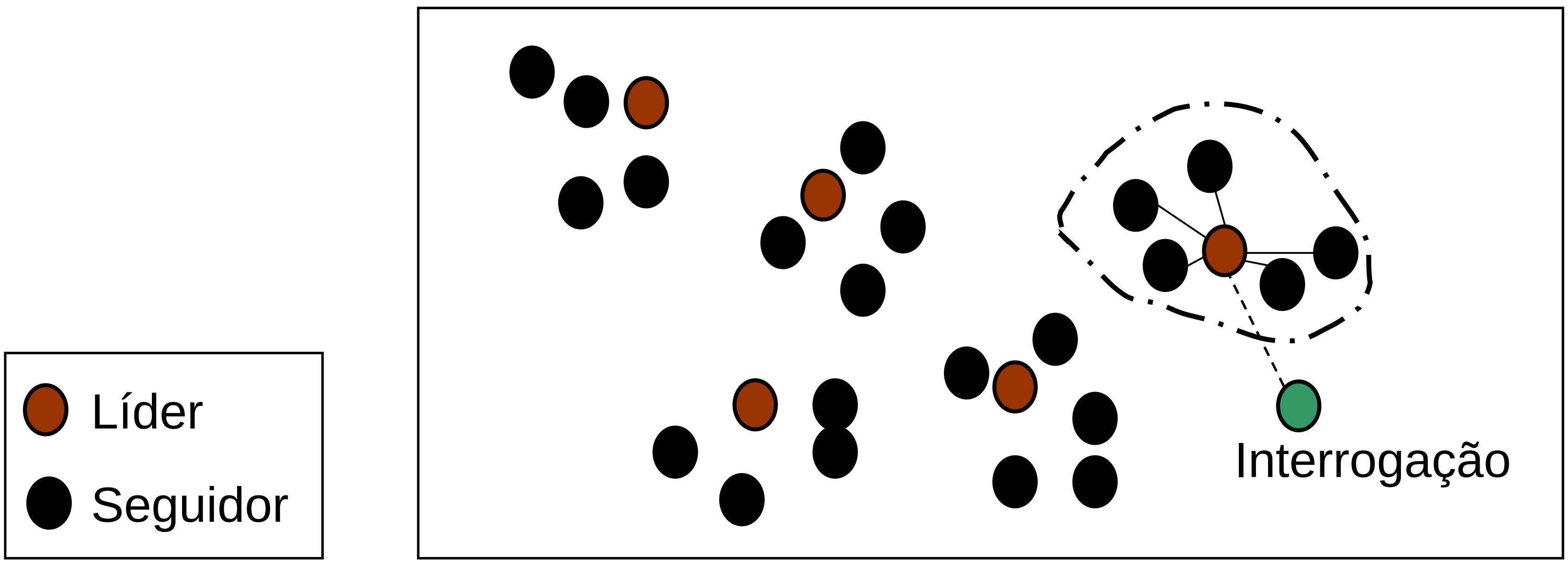
Reduzir complexidade – corte por agrupamento

Técnica

corte por agrupamento, ou "cluster pruning".

Pré-Processamento:

1. escolher aleatoriamente √N documentos da colecção; chamar a cada um desses documentos líder.
2. para cada um dos restantes documentos calcular o seu líder mais próximo chamar a cada uma desses documento seguidor.



Paulo Trigo Silva

... corte por agrupamento (as partições e a interrogação)

<u>Interrogação</u> q:

1. encontrar o líder (documento) L mais próximo de q (calcular $\sqrt{N}$ co-senos).
2. os documentos candidatos são L e os seus seguidores;
   calcular co-seno dos documentos candidatos.

- O espaço de documentos é fraccionado $\sqrt{N}$ em grupos
  - cada grupo é representado pelo seu líder
- ... na partição de seguidores induzida pelo $\sqrt{N}$ líderes
  - o número esperado de seguidores para cada líder será
  - $\approx (N / \sqrt{N}) = \sqrt{N}$

Paulo Trigo Silva

... corte por agrupamento (líderes aleatórios e variação)

- Usar líderes aleatórios é rápido e pode também
  - reflectir a distribuição do espaço de vectores (documentos)
- ... é provável que uma região densa em documentos
  - origine múltiplos líderes e assim partições finas dessa região
- Variação do corte por agrupamento: sejam b_1 e b_2 inteiros
  - pré-processamento:
    - ◊ para cada seguidor identificar os b_1 líderes mais próximos.
  - interrogação q:
    - ◊ considerar os b_2 líderes mais próximos de q.
  - ... note-se que a técnica base corresponde a ter $b_1 = b_2 = 1$
- ... aumentar b1 e b2 aumenta a possibilidade de encontrar os K
  - documentos que de facto têm o valor de co-seno mais alto
  - ... à custa de mais processamento!

Paulo Trigo Silva

Métricas de avaliação do resultado da pesquisa

		avaliação do utilizador (humano)	
		Relevante	Não Relevante
resposta da pesquisa (máquina)	Recuperado	positivo (p)	falso positivo (fp)
	Não Recuperado	falso negativo (fn)	negativo (n)

Métricas:

Precisão ("Precision") ≡ fracção dos documentos recuperados que é relevante

P(Relevante | Recuperado) =

= #(documentos relevantes e recuperados) / #(documentos recuperados)

Cobertura ("Recall") ≡ fracção os documentos relevantes que são recuperados

P(Recuperado | Relevante) =

= #(documentos relevantes e recuperados) / #(documentos relevantes)

... métricas de avaliação – cálculo

		avaliação do utilizador (humano): Relevante	avaliação do utilizador (humano): Não Relevante
resposta da pesquisa (máquina)	Recuperado	positivo (p)	falso positivo (fp)
resposta da pesquisa (máquina)	Não Recuperado	falso negativo (fn)	negativo (n)

Precisão ("Precision") ≡ fracção dos documentos recuperados que é relevante

= #(documentos relevantes e recuperados) / #(documentos <u>recuperados</u>)

= $\boxed{p / (p + fp)}$

Cobertura ("Recall") ≡ fracção os documentos relevantes que são recuperados

P(Recuperado | Relevante) =

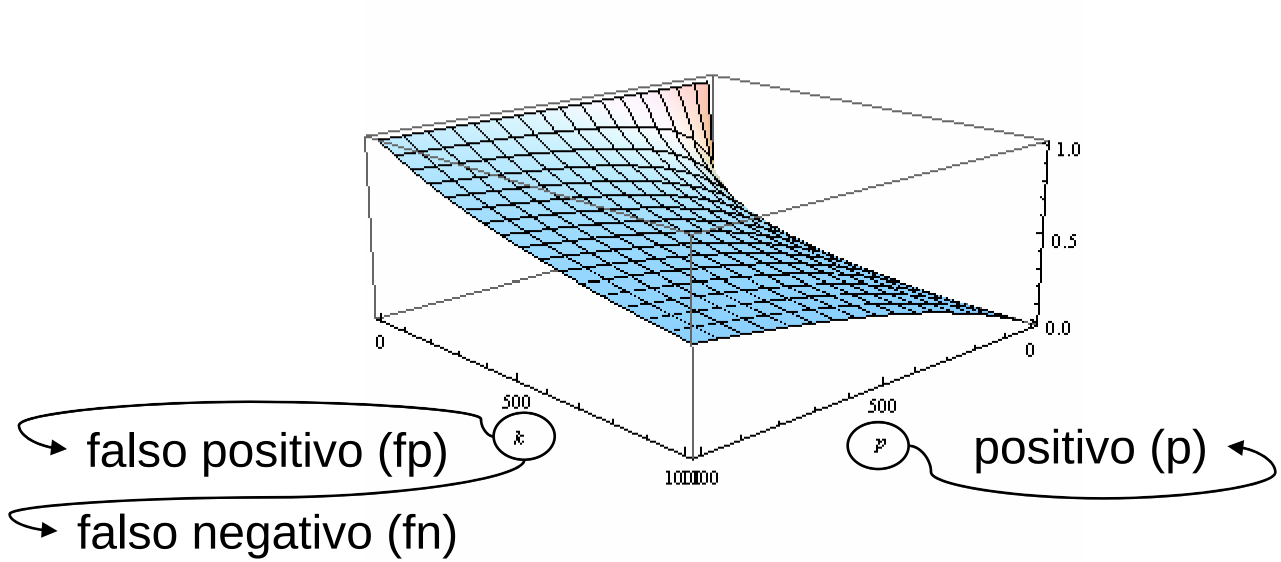
= #(documentos relevantes e recuperados) / #(documentos <u>relevantes</u>)

= $\boxed{p / (p + fn)}$

Perspectiva de variação – precisão e cobertura

Precisão (“Precision”) ≡ p / (p + fp)

Cobertura (“Recall”) ≡ p / (p + fn)



A **precisão** aumenta quando diminuem os falsos positivos.

A **cobertura** aumenta diminuem os falsos negativos.

A **precisão** e a **cobertura** diminuem quando diminuem os positivos.

Paulo Trigo Silva

Paulo Trigo Silva

... precisão e cobertura – “tensão” entre ambas

Para obter óptima cobertura basta recuperar todos os documentos da colecção!

No entanto teria baixa precisão (muitos não seriam relevantes).

A cobertura não decresce com o número de documentos recuperados.

A precisão pode decrescer com o número de documentos recuperados.

A precisão decresce, em sistemas reais (e.g. comerciais) usualmente com:

• o aumento de documentos recuperados, ou

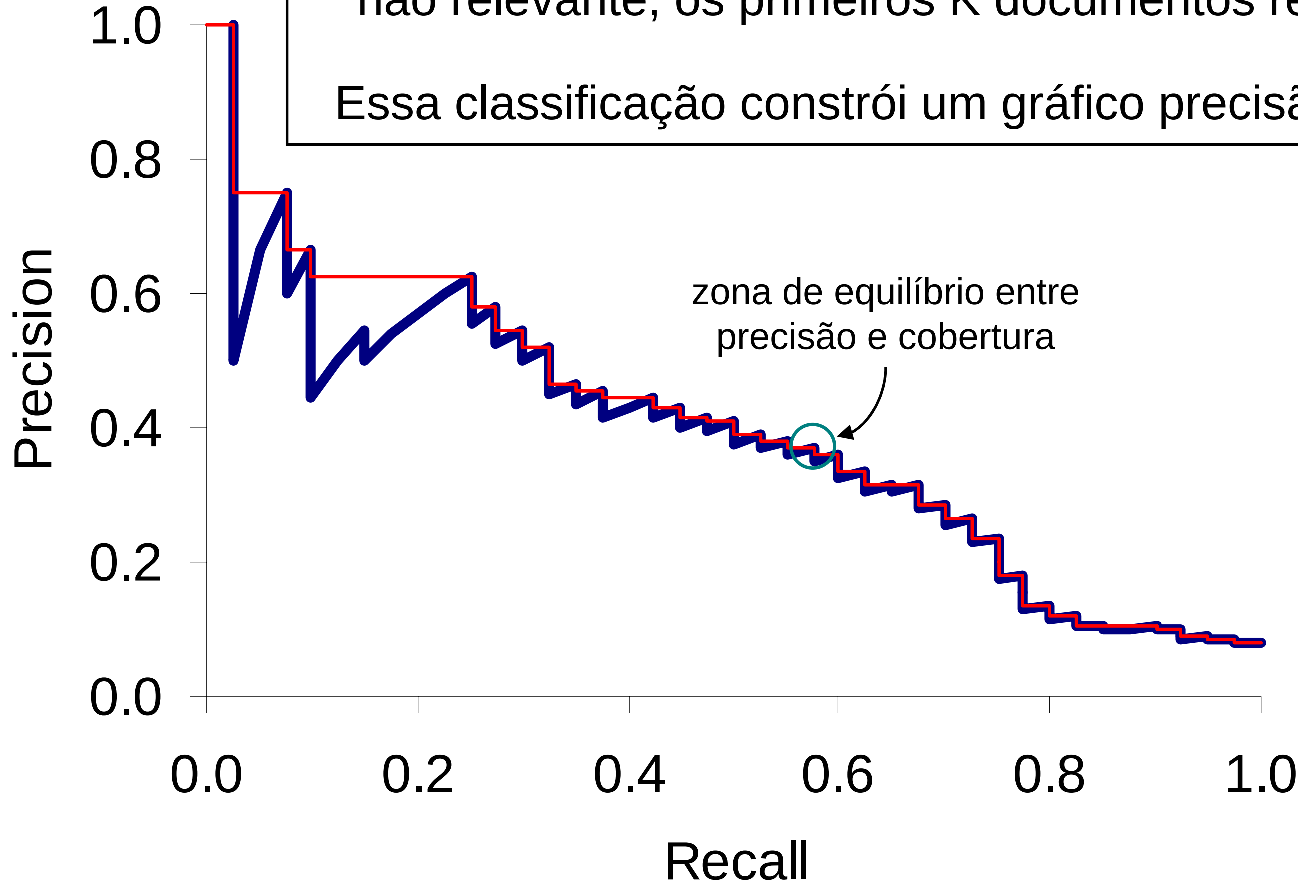
• o aumento da cobertura.

... empiricamente pode comprovar-se num qualquer um motor de busca.

Gráfico precisão-cobertura

Um avaliador (humano) pode classificar como relevante, ou não relevante, os primeiros K documentos recuperados.

Essa classificação constrói um gráfico precisão-cobertura.



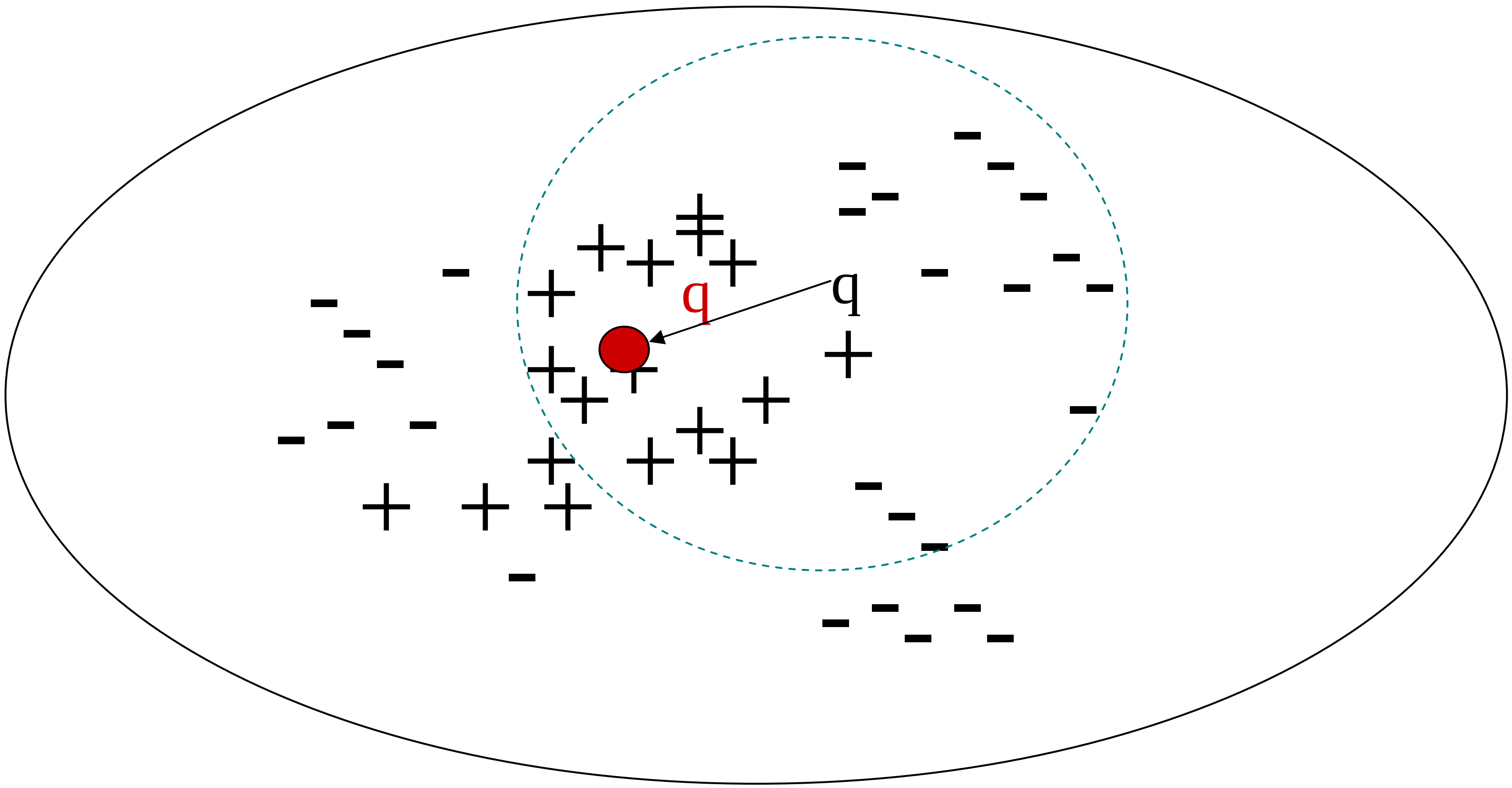
exemplo de caso concreto

Paulo Trigo Silva

Retroacção

- Aprender com exemplos
  - neste caso um exemplo é “a avaliação do utilizador a uma resposta”
- ... os exemplos podem ser positivos ou negativos
  - positivos: documentos considerados relevantes (pelo utilizador)
  - negativos: documentos considerados não relevantes (pelo utilizador)
- Como utilizar os exemplos para melhorar o desempenho?
  - “modificando a interrogação”
  - ... o método mais conhecido é o da “retroacção de Rocchio”
- Como modificar a interrogação?
  - adicionar novos termos
  - ajustar pesos de termos antigos
  - combinar as aproximações anteriores

Retroacção de Rocchio – ideia



Retroacção de Rocchio – formulação

Nova Interrogação

Parâmetros

$$\vec{q}_m = \alpha\,\vec{q} + \frac{\beta}{|D_r|}\sum_{\forall \vec{d}_j \in D_r} \vec{d}_j - \frac{\gamma}{|D_n|}\sum_{\forall \vec{d}_j \in D_n} \vec{d}_j$$

Interrogação Inicial

Documentos Relevantes

Documentos Não Relevantes

Paulo Trigo Silva

Características do Modelo Vectorial

- Bom desempenho
  - obteve melhores resultados na TREC
- Intuitivo e simples de implementar
  - mais estudado e avaliado
  - implementado no sistema SMART; desenvolvido Cornell 1960-1999
  - ... ainda usado
- Assume independência dos termos
- Assume que documentos e interrogações são o mesmo
- Dificuldade em “afinar” os diversos parâmetros

Paulo Trigo Silva

A reter...

- Modelo Vectorial pertence a uma família de modelos heurísticos
  - ... para recuperação de informação
- Normalização dos pesos (tf-idf) conduz a bons resultados
- Retroacção de Rocchio é modelo eficiente de alteração de pesos
- Modelo Vectorial é usado num grande número de aplicações
- Dificuldade em “afinar” os diversos parâmetros
- Generalização do Modelo Vectorial
  - ... tem maior fundamentação teórica mas é pouco usado na prática

Exercício

Documentos:
Austen; *Sense and Sensibility* (SaS), *Pride and Prejudice* (PaP);
Bronte; *Wuthering Heights* (WP)

Calcular a semelhança do documento SaS com os documentos Pap e WH

	SaS	**PaP**	**WH**
affection	**115**	**58**	**20**
jealous	**10**	**7**	**11**
gossip	**2**	**0**	**6**

	SaS	**PaP**	**WH**
affection	**0,996**	**0,993**	**0,847**
jealous	**0,087**	**0,120**	**0,466**
gossip	**0,017**	**0,000**	**0,254**

cos(SaS, PaP) = .996 x .993 + .087 x .120 + .017 x 0.0 = 0.999
cos(SaS, WH) = .996 x .847 + .087 x .466 + .017 x .254 = 0.929