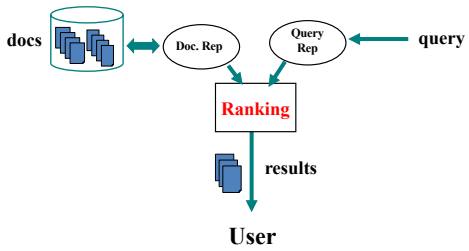
Vector Space Retrieval Models

(Search and Data Mining)

Hui Fang
Department of Electrical and Computer Engineering
University of Delaware

Relevance modeling is important.



The Basic Question

Given a query, how do we know if document A is more relevant than B?

One Possible Answer

If document A uses more query words than document B

(Word usage in document A is more similar to that in query)

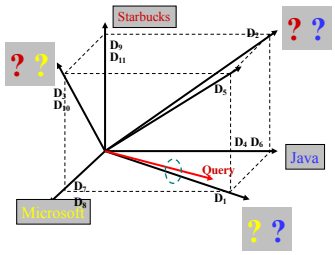
Relevance = Similarity

- Assumptions
  - Query and document are represented similarly
  - A query can be regarded as a "document"
  - Relevance(d,q) $\propto$ similarity(d,q)
- R(q) = {d$\in$C|f(d,q)>θ}, f(q,d)=Δ(Rep(q), Rep(d))
- Key issues
  - How to represent query/document?
  - How to define the similarity measure Δ?

Vector Space Model: Main Idea

- Documents and queries are represented as vectors in a high-dimensional space.
  - Each dimension corresponds to a term.
  - All the terms are assumed to be independent.
  - The value of an element in a vector corresponding to the weight of the term.
    - E.g., d=(x_1,…,x_N), x_i is "importance" of term I
- Relevance can be approximated by the similarity between the query vector and document vector in the vector space

VS Model: illustration



VS Models: Representations

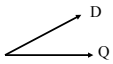
- Query/Document: term vectors

$D = (w_{t_1 D}, ..., w_{t_n D})$ $Q = (w_{t_1 Q}, ..., w_{t_n Q})$

- Relevance between doc and query

→distance between two vectors

$similarity(D,Q) = \sum w_{t_i D} \times w_{t_i Q}$



How to Measure Similarity?

$\bar{D}_i = (w_{i1}, ..., w_{iN})$

$\bar{Q} = (w_{q1}, ..., w_{qN})$

Dot product similarity: $sim(\bar{Q}, \bar{D}_i) = \sum_{j=1}^{N} w_{qj} * w_{ij}$

Cosine: $sim(\bar{Q}, \bar{D}_i) = \frac{\sum_{j=1}^{N} w_{qj} * w_{ij}}{\sqrt{\sum_{j=1}^{N} (w_{qj})^2 * \sum_{j=1}^{N} (w_{ij})^2}}$

$Similarity(D,Q) = \sum_{t_i \in Q, D} w_{t_i D} \cdot w_{t_i Q}$

Q= "a dog"

D1="a dog walk dog animal dog cute"

D2=" a cat walk cat cat dog "

D3=" a book a book"

Assume $w_{t_i D} = 1$ when $t_i \in D$

$w_{t_i D} = 0$ when $t_i \notin D$

	dog	a	walk	cute	animal	book	cat
Q=(							)
D1=(							)
D2=(							)
D3=(							)

Challenge: How to assign the weights to each term?

Term Frequency (TF)

The relevance score of a document is related to the occurrence of a query term.

dog

The **cat** is a small carnivorous mammal. The **cat** is known to hunt over 1,000 species for food. The **cat** can be trained to obey simple commands. **Cats**, like **dogs**, are digitigrades. **Cats** are capable of walking very precisely.

Dogs were domesticated from wolves. **Dogs** were first domesticated in East Asia. **Dogs**, like humans, are highly social animals. This similarity has earned **dogs** a unique position in the realm of interspecies relationships.

$Similarity(D,Q) = \sum_{t_i \in Q, D} w_{t_i D} \cdot w_{t_i Q}$

Q= "a dog"

D1="a dog walk dog animal dog cute"

D2=" a cat walk cat cat dog "

D3=" a book a book"

$w_{t_i D} = TF(t_i, D) = c(t_i, D)$

	dog	a	walk	cute	animal	book	cat
Q=(	1,	1,	0,	0,	0,	0,	0)
D1=(	3,	1,	1,	1,	1,	0,	0)
D2=(	1,	1,	1,	0,	0,	0,	3)
D3=(	0,	2,	0,	0,	0,	2,	0)

Inverse Document Frequency (IDF)

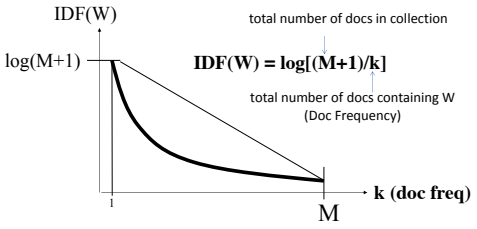
The importance of a term is related to document frequency of the term.

the dog

The earliest **trees** were **tree** ferns and horsetails, which grow in vast forests in **the** Carboniferous Period. **Tree** ferns still survive, but **the** only surviving horsetails are not of tree form. **The** listing below gives examples of many **trees**.

Dogs were domesticated from wolves. **Dogs** were first domesticated in East Asia. **Dogs**, like humans, are highly social animals. This similarity has earned **dogs** a unique position in the realm of interspecies relationships.

IDF Weighting: Penalizing Popular Terms



$$Similarity(D,Q) = \sum_{w \in Q,D} w_{w,D} \cdot w_{w,Q}$$

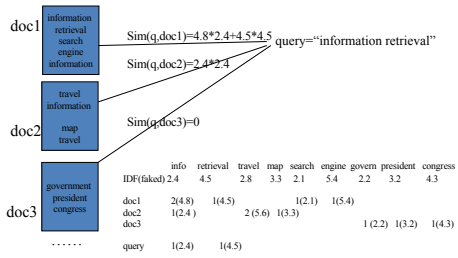
Q= "a dog"
D1="a dog walk dog animal dog cute"
D2=" a cat walk cat cat dog "
D3=" a book a book"

$$w_{t,D} = TF(t,D) \cdot IDF(t) = TF(t,D) \cdot \ln\left(\frac{N+1}{df}\right)$$

N=3, df(a)=3, df(dog)=df(walk)=2, df(cat)=df(book)=df(cute)=df(animal)=1

	dog	a	walk	cute	animal	book	cat	
Q=(	log2,	log4/3,	0,	0,	0,	0,	0	)
D1=(	3*log2,	log(4/3),	log2,	log4,	log4,	0,	0	)
D2=(	log2,	log(4/3),	log2,	0,	0,	0,	3*log4	)
D3=(	0,	2*log(4/3),	0,	0,	0,	2*log4,	0	)

VS Example: Raw TF & Dot Product



	info	retrieval	travel	map	search	engine	govern	president	congress
IDF(faked)	2.4	4.5	2.8	3.3	2.1	5.4	2.2	3.2	4.3
doc1	2(4.8)	1(4.5)				1(2.1)	1(5.4)		
doc2	1(2.4)		2 (5.6)	1(3.3)					
doc3								1 (2.2)	1(3.2) 1(4.3)
query	1(2.4)		1(4.5)						

Pivoted Normalization Formula

$$S(Q,D) = \sum_{t \in Q,D} \frac{1+\ln(1+\ln(c(t,D)))}{(1-b)+b\frac{|D|}{avdl}} \cdot c(t,Q) \cdot \ln\frac{N+1}{df(t)}$$

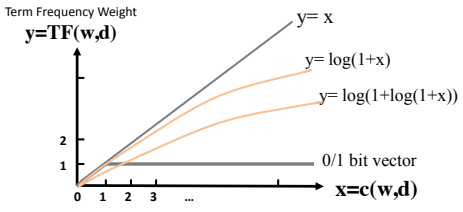
IDF weighting

Pivoted Normalization Formula

$$S(Q,D) = \sum_{t \in Q,D} \frac{1+\ln(1+\ln(c(t,D)))}{(1-b)+b\frac{|D|}{avdl}} \cdot c(t,Q) \cdot \ln\frac{N+1}{df(t)}$$

TF weighting

TF Transformation



Pivoted Normalization Formula

$$S(Q,D)=\sum_{t\in Q,D}\frac{1+\ln(1+\ln(c(t,D)))}{(1-b)+b\frac{|D|}{avdl}}\cdot c(t,Q)\cdot\ln\frac{N+1}{df(t)}$$

Length normalization

Document Length Normalization (LN)

The relevance score of a document is related to the document length.

dog

The **cat** is a small carnivorous mammal. It is valued by humans for its companionship and its ability to hunt vermin, and has been associated with humans for at least 9,500 years. A skilled predator, the **cat** is known to hunt over 1,000 species for food. The **cat** is intelligent and can be trained to obey simple command. **Cats**, like **dogs**, walk directly on their toes. **Cats** are capable of walking very precisely, because **cats** ….

Dogs, like humans, are highly social animals and this similarity in their overall behavioral pattern accounts for their trainability.

Document Length Normalization

- Penalize a long document with a document length normalizer.
  - Long documents have a better chance to match any query.
  - Need to avoid over-penalization.
- A document is long because
  - It uses more words→ more penalization
  - It has more content → less penalization

Pivoted Length Normalization
- Use average document length as pivot

$$normalizer = 1-b+b\frac{|d|}{avdl}$$

$b \in [0,1]$