UFMG UNIVERSIDADE FEDERAL DE MINAS GERAIS

Ranking Models

Vector Space Models

Rodrygo L. T. Santos
rodrygo@dcc.ufmg.br

The ranking problem

Given
- Some evidence of the user's need

Produce
- A list of matching information items
- In decreasing order of relevance

The ranking problem

Given
- Some ~~evidence of the user's need~~ ***query***

Produce
- A list of matching ~~information items~~ ***documents***
- In decreasing order of relevance

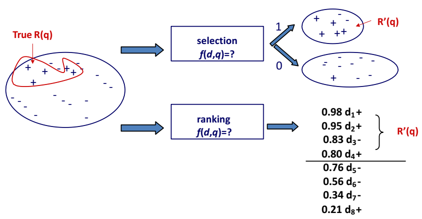
The ranking problem

$$f(q,d)$$

Why rank?

Couldn't $f(q,d)$ be just an indicator function?

Document selection vs. ranking



Why not select?

The classifier is unlikely accurate
- Over-constrained: no relevants returned
- Under-constrained: too many relevants returned
- Hard to find an appropriate threshold

Not all relevant documents are equally relevant!
- Prioritization is needed

Probability Ranking Principle (PRP)

> *Ranking documents by decreasing probability of relevance results in optimal effectiveness, provided that probabilities are estimated (1) with certainty and (2) independently.*
> - Robertson, 1977

Ranking effectiveness

Effectiveness is about doing the right thing; it's about finding documents that are relevant to the user

Relevance is influenced by many factors
- Topical relevance vs. user relevance
- Task, context, novelty, style

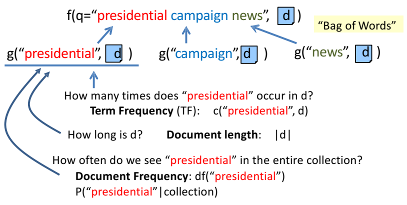
Ranking models define ***a view of*** relevance

Ranking models

Provide a mathematical framework for ranking
- Each model builds upon different assumptions

Progress in ranking models has corresponded with improvements in effectiveness
- An effective model should score relevant documents higher than non-relevant documents

Fundamental elements



Many classical models

Similarity-based models: $f(q,d) = \mathrm{sim}(q,d)$
- Vector space models

Probabilistic models: $f(d,q) = p(R=1|d,q)$
- Classic probabilistic models
- Language models
- Information-theoretic models

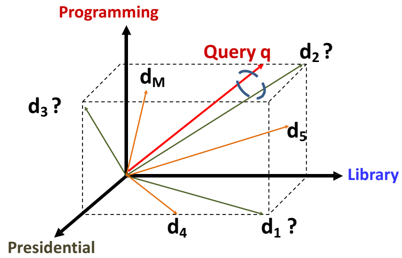
Many extended models

Structural models
- Beyond bags-of-words

Semantic models
- Beyond lexical matching

Contextual models
- Beyond queries

Vector Space Model (VSM)



VSM is a framework

Queries and documents as term vectors
- Term as the basic concept (e.g., word or phrase)

A vocabulary V defines a $|V|$-dimensional space
- Vector components as real-valued term weights

Relevance estimated as $f(q,d) = \mathrm{sim}(q,d)$
- $q = (x_1, \ldots, x_{|V|})$ and $d = (y_1, \ldots, y_{|V|})$

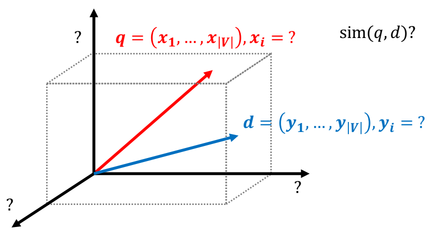
What VSM doesn't say

How to define vector dimensions
- Concepts are assumed to be orthogonal

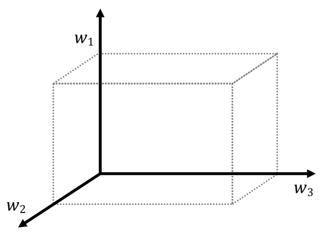
How to place vectors in the space
- Term weight in query indicates importance of term
- Term weight in document indicates topicality

How to define the similarity measure

What VSM doesn't say

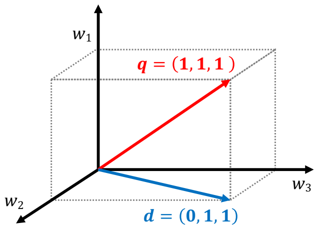


Dimensions as a bag of words (BOW)



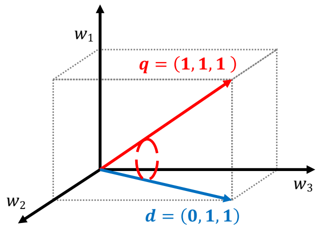
Vocabulary
$V = (w_1, \ldots, w_{|V|})$

Vectors placed as bit vectors



$x_i, y_i \in \{0,1\}$
1: word w_i is present
0: word w_i is absent

Similarity as dot product



$$\begin{aligned}\mathrm{sim}(q, d) &= q \cdot d \\ &= x_1 y_1 + \cdots + x_{|V|} y_{|V|} \\ &= \sum_{i=1}^{|V|} x_i y_i\end{aligned}$$

Simplest VSM = BOW + bit vectors + dot

$q = (x_1, \dots, x_{|V|})$

$d = (y_1, \dots, y_{|V|})$

$x_i, y_i \in \{0,1\}$
1: word w_i is present
0: word w_i is absent

$$\begin{aligned}\mathrm{sim}(q, d) &= q \cdot d \\ &= x_1 y_1 + \cdots + x_{|V|} y_{|V|} \\ &= \sum_{i=1}^{|V|} x_i y_i\end{aligned}$$

What does this ranking function intuitively capture?
Is this a good ranking function?

How would you rank these documents?

q = [news about presidential campaign]

		ideal
d_1	… **news about** …	d_4 +
d_2	… **news about** organic food **campaign**…	d_3 +
d_3	… **news** of **presidential campaign** …	d_1 −
d_4	… **news** of **presidential campaign** … … **presidential** candidate …	d_2 −
d_5	… **news** of organic food **campaign**… **campaign**…**campaign**…**campaign**…	d_5 −

Ranking using the simplest VSM

q = [news about presidential campaign]

d_1	… **news about** …
d_3	… **news** of **presidential campaign** …

V = { news, about, presidential, campaign, food, … }

$q = (1, 1, 1, 1, 0, \dots)$
$d_1 = (1, 1, 0, 0, 0, \dots)$ $\quad \mathrm{sim}(q, d_1) = 2$
$d_3 = (1, 0, 1, 1, 0, \dots)$ $\quad \mathrm{sim}(q, d_3) = 3$

Is it effective?

q = [news about presidential campaign]

		f(*q*,*d*)	ranking	ideal
d_1	… **news about** …	2	d_2	d_4 +
d_2	… **news about** organic food **campaign**…	3	d_3	d_3 +
d_3	… **news** of **presidential campaign** …	3	d_4	d_1 −
d_4	… **news** of **presidential campaign** … … **presidential** candidate …	3	d_1	d_2 −
d_5	… **news** of organic food **campaign**… **campaign**…**campaign**…**campaign**…	2	d_5	d_5 −

What's wrong with it?

q = [news about presidential campaign]	f(q,d)	ranking	ideal
		d_2	d_4 +
		d_3	d_3 +
d_3 … **news** of **presidential campaign** …	3	d_4	d_1 −
d_4 … **news** of **presidential campaign** … … **presidential** candidate …	3	d_1	d_2 −
		d_5	d_5 −

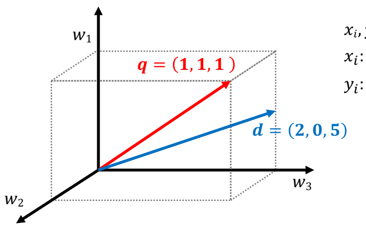
Matching "presidential" ***more times*** *deserves more credit!*

What's wrong with it?

q = [news about presidential campaign]	f(q,d)	ranking	ideal
		d_2	d_4 +
d_2 … **news about** organic food **campaign**…	3	d_3	d_3 +
d_3 … **news** of **presidential campaign** …	3	d_4	d_1 −
		d_1	d_2 −
		d_5	d_5 −

Matching "presidential" is ***more important*** *than matching "about"!*

Vectors placed as tf vectors



$x_i, y_i \in \mathbb{N}$
x_i: $\text{tf}_{w_i,q}$
y_i: $\text{tf}_{w_i,d}$

Ranking using VSM with tf vectors

q = [news about presidential campaign]

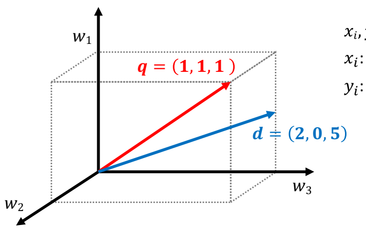
d_3 … **news** of **presidential campaign** …

d_4 … **news** of **presidential campaign** … … **presidential** candidate …

V = { news, about, presidential, campaign, food, … }

$q = (1, 1, 1, 1, 0, \dots)$
$d_3 = (1, 0, 1, 1, 0, \dots)$ $\quad \text{sim}(q, d_3) = 2$
$d_4 = (1, 0, 2, 1, 0, \dots)$ $\quad \text{sim}(q, d_4) = 4$

Vectors placed as tf-idf vectors



$x_i, y_i \in \mathbb{R}$
x_i: $\text{tf}_{w_i,q}\ \text{idf}_{w_i}$
y_i: $\text{tf}_{w_i,d}\ \text{idf}_{w_i}$

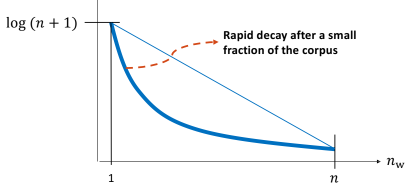
Inverse document frequency (idf)

$$\text{idf}_w = \log \frac{n+1}{n_w}$$

- n: number of documents in the corpus
- n_w: number of documents where w appears

Why a log-based penalization?

$$\mathrm{idf}_w = \log \frac{n+1}{n_w}$$



Ranking using VSM with tf-idf vectors

q = [news about presidential campaign]

d_2 … **news about** organic food **campaign**…

d_3 … **news** of **presidential campaign** …

V = { news, about, presidential, campaign, food, … }
idf = (1.5, 1.0, 2.5, 3.1, 1.8, …)

$q = (1, 1, 1, 1, 0, \dots)$

$d_2 = (1*1.5, 1*1.0, 0, 1*3.1, 0, \dots)$ $\quad \mathrm{sim}(q, d_2) = 5.6$

$d_3 = (1*1.5, 0, 1*2.5, 1*3.1, 0, \dots)$ $\quad \mathrm{sim}(q, d_3) = 7.1$

Is it effective?

q = [news about presidential campaign]

	f(*q*,*d*)	ranking	ideal
d_1 … **news about** …	2.5	d_5	d_4 +
d_2 … **news about** organic food **campaign**…	5.6	d_4	d_3 +
d_3 … **news** of **presidential campaign** …	7.1	d_3	d_1 −
d_4 … **news** of **presidential campaign** … … **presidential** candidate …	9.6	d_2	d_2 −
d_5 … **news** of organic food **campaign**… **campaign**…**campaign**…**campaign**…	13.9	d_1	d_5 −

Is it effective?

q = [news about presidential campaign]

	f(*q*,*d*)	ranking	ideal
	2.5	d_5	d_4 +
	5.6	d_4	d_3 +
	7.1	d_3	d_1 −
d_4 … **news** of **presidential campaign** … … **presidential** candidate …	9.6	d_2	d_2 −
d_5 … **news** of organic food **campaign**… **campaign**…**campaign**…**campaign**…	13.9	d_1	d_5 −

Ranking using VSM with tf-idf vectors

q = [news about presidential campaign]

d_4 … **news** of **presidential campaign** … … **presidential** candidate …

d_5 … **news** of organic food **campaign**… **campaign**…**campaign**…**campaign**…

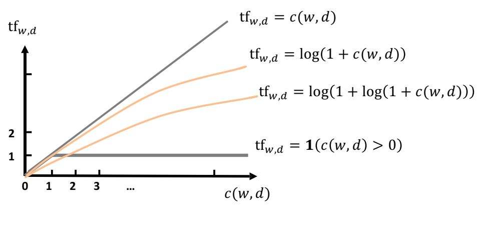
V = { news, about, presidential, campaign, food, … }
idf = (1.5, 1.0, 2.5, 3.1, 1.8, …)

$q = (1, 1, 1, 1, 0, \dots)$

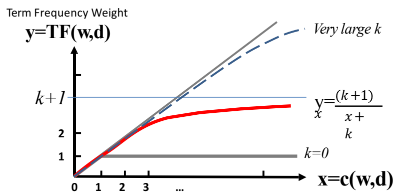
$d_4 = (1*1.5, 0, 2*2.5, 1*3.1, 0, \dots)$ $\quad \mathrm{sim}(q, d_4) = 9.6$

$d_5 = (1*1.5, 0, 0, 4*3.1, 1*1.8, \dots)$ $\quad \mathrm{sim}(q, d_5) = 13.9$

Transforming tf

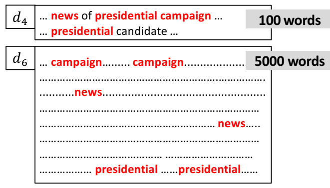


TF Transformation: BM25 Transformation



What about document length?

q = [news about presidential campaign]



$f(q,d_6) > f(q,d_4)$?

Document length normalization

Penalize long documents

- Avoid matching by chance
- Must also avoid over-penalization

A document is long because

- It uses more words → more penalization
- It has more content → less penalization

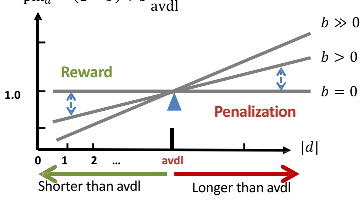
Pivoted length normalization (pln)

$$\text{pln}_d = (1-b) + b\frac{|d|}{\text{avdl}}$$

- $|d|$: document length in tokens
- avdl: average document length in the corpus
- $b \in [0,1]$: parameter

Pivoted length normalization (pln)

$\text{pln}_d = (1-b) + b\frac{|d|}{\text{avdl}}$



State-of-the-art VSM ranking

Pivoted length normalization VSM [Singhal et al. 1996]

- $f(q,d) = \sum_{w\in q} c(w,q)\frac{\ln(1+\ln(1+c(w,d)))}{(1-b)+b\frac{|d|}{avdl}}\log\frac{n+1}{n_w}$

Okapi/BM25 [Robertson and Walker, 1994]

- $f(q,d) = \sum_{w\in q} c(w,q)\frac{(k_1+1)\,c(w,d)}{c(w,d)+k_1\left((1-b)+b\frac{|d|}{avdl}\right)}\log\frac{n+1}{n_w}$

Summary

Fundamental ranking components
- Term and document frequency
- Document length

VSM is a framework
- Components as term and document weights
- Relevance as query-document similarity

Summary

Lack of theoretical justification
- Axiomatic approaches, probabilistic approaches

Room for further improvement
- Structure, semantics, feedback, context
- Feature-based models

References

Text Data Management: A Practical Introduction to Information Retrieval and Text Mining, Ch. 6
Zhai and Massung, 2016

Search Engines: Information Retrieval in Practice, Ch. 7
Croft et al., 2009

References

Pivoted document length normalization
Singhal et al., SIGIR 1996

Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval
Robertson and Walker, SIGIR 1994

The probability ranking principle in IR
Robertson, J. Doc. 1977