Information Retrieval, and the Vector Space Model

Art B. Owen
Stanford University
owen@stat.stanford.edu

Search Engines

Goal: Find documents relevant to a query

Examples:

1. Boolean query:
   Monte Carlo AND (importance OR stratification) AND NOT Chevrolet
2. Natural language query:
   Is it raining in Topanga?
3. List of words:
   Efron bootstrap resample

Word counts

Most engines use word counts in documents

Most use other things too

- links
- titles
- position of word in document
- sponsorship
- present and past user feedback

Term Document Matrix

$f_{ij} \equiv$ number of times term T_i is in document D_j

Documents

1. web page
2. article
3. section
4. paragraph
5. sentence

Terms

1. word e.g. “airplane”
2. n-gram e.g. “airp”, “irpl”, “rpla”, “plan”, “lane”
3. collocation e.g. “white house” or “New York”

Term-document matrices are huge and sparse

Further processing

Stop words Ignore very common words

"the" "and" "what"

Stemming Strip words to root

reformation reformative reformed reforming

$\rightarrow$ reform

tf-idf

Term frequency, inverse document frequency

$$f_{ij} \rightarrow w_{ij} = (1 + \log(f_{ij})) \times \left(1 + \log\left(\frac{N}{f_{i+}}\right)\right)$$

where

$$N = \text{Number of documents}$$
$$f_{i+} = \text{Number of documents with at least one } t_i$$

There are many variations

Example

Upper left 10x10 corner

0	0	0	0	3.08	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	2.67	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	4.39	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Example ctd

f_{i+}	f_{ij}	Term
24	29	TOMORROW
28	31	SPENT
7	12	FACTS
38	63	EXPLOSIVES
24	29	LEADING
36	45	NATIONS
9	18	0
58	91	1
44	85	2
23	31	OPPORTUNITY
74	136	GENERAL
8	10	TEARS
11	13	VIDEOTAPE
17	32	DEVICES
37	43	FACE
13	14	ALONE
33	35	ALONG
27	37	HAVEN
86	137	FACT

Vector space

Each document is a vector $D_j = (w_{1j}, \ldots, w_{Tj})'$ of transformed counts

Document similarity could be

$$D_j' D_k \quad \text{or} \quad \frac{D_j' D_k}{\|D_j\| \|D_k\|}$$

A query Q is a (very short) document

Precision-recall

Given Q rank N documents in order of relevance

Suppose there are R truly relevant documents

Precision $=$

% of first n ranked documents that are relevant

Recall $=$

% of R relevant documents among first n ranked documents

Transposing it

A document has a weighted list of words

A word has a weighted list of documents

Query with a list of documents:

1. Todays documents...word NASDAQ is hot
2. All documents in bovine set
3. All documents in dental set

Also

"Words are known by the company they keep"

Do "boat" queries find "ship" docs?

Maybe we should "cluster" the terms

Let $W = (w_{ij})$

Clustering: approximate by

$\widehat{W}_{ij} = \sum_{k=1}^{K} \theta_{ik}\mu_{kj}$

μ_k is k'th cluster mean

θ_{ik} is 1 if term i in cluster k, zero else

Latent semantic indexing

SVD:

$$W_{ij} = \sum_{k=1}^{\min(N,T)} \lambda_k u_{ik} v'_{jk}$$

$$\widehat{W}_{ij} = \sum_{k=1}^{K} \lambda_k u_{ik} v'_{jk}$$

May find a nautical singular vector u_k with "boat" and "ship" and "starboard" etc.

Run queries on $\widehat{W}$ with $K \ll N$

- SVD looks a bit like clusters
- First few singular values have "less noise"
- $\widehat{W}'Q$ much faster than $W'Q$
- Less storage too

References

1. Manning and Shutze (1999) "Foundations of Statistical Natural Language Processing", MIT Press
   - Grammar and Parsing
   - Statistical models of word frequency
2. Witten, Moffat and Bell (1999) "Managing Gigabytes" 2nd Edition. Morgan Kaufman.
   - Compression and index construction
   - Searching compressed data
3. Berry and Browne (1999) "Understanding Search Engines" SIAM
   - Describe a course project building an engine
4. Berry, Dumais, O'Brien (1995) "Using Linear Algebra for Intelligent Information Retrieval", SIAM Review v37 N4 pp573–595
   - Emphasize SVD updates