Informatics 1: Data & Analysis

Lecture 16: Vector Spaces for Information Retrieval

Ian Stark

School of Informatics
The University of Edinburgh

Tuesday 19 March 2013
Semester 2 Week 9

http://www.inf.ed.ac.uk/teaching/courses/inf1/da

Coursework Submission

The coursework assignment has now been online for some time.. This runs alongside your usual tutorial exercises; ask tutors for help with it where you have specific questions.

The assignment is a Inf1-DA examination paper from 2011. Your tutor will give you marks and feedback on your work in the last tutorial of semester.

How to submit your work

Submit your solutions on paper to the labelled box outside the ITO office on level 4 of Appleton Tower by **4pm Thursday 21 March 2013**. Please ensure that all sheets you submit are firmly stapled together, and on the first page write your name, matriculation number, tutor name and tutorial group.

Late Coursework and Extension Requests

There is a web page with general information about coursework, assessment and feedback in the School of Informatics. Please read it.

http://www.inf.ed.ac.uk/teaching/coursework.html

This also links to the School policy on late coursework and extension requests. Please read that too.

Late Submissions

Normally, you will not be allowed to submit coursework late. Coursework submitted after the deadline set will receive a mark of 0%.

If you have a good reason to need to submit late, you must do the following:

- Read the extension requests web page carefully.
- Request an extension identifying the affected course and assignment.
- Submit the request via the ITO contact form.

Unstructured Data

Data Retrieval

- The information retrieval problem
- The vector space model for retrieving and ranking

Statistical Analysis of Data

- Data scales and summary statistics
- Hypothesis testing and correlation
- χ^2 tests and collocations

also *chi-squared*, pronounced "kye-squared"

Unstructured Data

Data Retrieval

- The information retrieval problem
- The vector space model for retrieving and ranking

Statistical Analysis of Data

- Data scales and summary statistics
- Hypothesis testing and correlation
- χ^2 tests and collocations — also *chi-squared*, pronounced "kye-squared"

Possible Query Types for Information Retrieval

We shall consider simple keyword queries, where we ask an IR system to:

- Find documents containing one or more of word_1, word_2, …, word_n

More sophisticated systems might support queries like:

- Find documents containing all of word_1, word_2, …, word_n;
- Find documents containing as many of word_1, word_2, …, word_n as possible.

Other systems go beyond these forms to more complex queries: using boolean operations, searching for whole phrases, regular expression matches, etc.

Models for Information Retrieval

If we look for all documents containing some words of the query then this may result in a large number of documents of widely varying relevance.

At this point we might want to refine retrieval beyond simple selection/rejection and introduce some notion of *ranking*.

Introducing more refined queries, and in particular ranking the results, requires a model of the documents being retrieved.

There are many such models. We focus on the *vector space model*.

This model is the basis of many IR applications; it originated in the work of Gerard Salton and others in the 1970's, and is still actively developed.

The Vector Space Model

Treat documents as vectors in a high-dimensional space, with one dimension for every distinct word.

Applying this to ranking of retrieved documents:

- Each document is a vector;
- Treat the query (a very short document) as a vector too;
- Match documents to the query by the *angle* between the vectors.
- Rank higher those documents which point in the same direction as the query.

Operating the model does not, in fact, require a strong understanding of higher-dimensional vector spaces: all we do is manipulate fixed-length lists of integers.

Various programming languages provide a **vector** datatype for fixed-length homogeneous sequences

The Vector for a Document

Suppose that $w_1, w_2, \ldots, w_n$ are all the different words occurring in a collection of documents $D_1, D_2, \ldots, D_k$.

We model each document D_i by an n-dimensional vector

$$(c_{i1}, c_{i2,}, \ldots, c_{ij}, \ldots, c_{in})$$

where c_{ij} is the number of times word w_j occurs in document D_i.

In the same way we model the query as a vector $(q_1, \ldots, q_n)$ by considering it as a document itself: q_j counts how many times word w_j occurs in the query.

Example

Consider a small document containing only the phrase

Sun, sun, sun, here it comes

from a document collection which contains only the words "comes", "here", "it", "sun" and "today".

The vector for the document is $(1, 1, 1, 3, 0)$:

comes	here	it	sun	today
1	1	1	3	0

The vector for the query "sun today" is $(0, 0, 0, 1, 1)$:

comes	here	it	sun	today
0	0	0	1	1

Document Matrix

For an information retrieval system based on the vector space model, frequency information for words in a document collection is usually precompiled into a *document matrix*:

- Each column represents a word that appears the document collection;
- Each row represents a single document in the collection;
- Each entry in the matrix gives the frequency of that word in that document.

This is a model in that it captures some aspects of the documents in the collection — enough to carry out certain queries or comparisons — but ignores others.

Example Document Matrix

	w_1	w_2	w_3	$\dots$	w_n
D_1	14	6	1	$\dots$	0
D_2	0	1	3	$\dots$	1
D_3	0	1	0	$\dots$	2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
D_K	4	7	0	$\dots$	5

Note that each row of the document matrix is the appropriate vector for the corresponding document.

Origins of the Vector Space Model

The following paper was never written.

G. Salton.
A Vector Space Model for Information Retrieval.
Communications of the ACM, 1975.
OR: Journal of the American Society for Information Science, 1975.
OR: None of the above.

This paper explains the story.

D. Dubin.
The most influential paper Gerard Salton never wrote.
Library Trends 52(4):748–764, 2004

Similarity of Vectors

Now that we have documents modelled as vectors, we can rank them by how closely they align with the query, also modelled as a vector.

A simple measure of how well these match is the angle between them as (high-dimensional) vectors: smaller angle means more similarity.

Using angle makes this measure independent of document size.

It turns out to be computationally simpler to calculate the cosine of that angle; this is more efficient, and gives exactly the same ranking.

Cosines (Some Things You Already Know)

The *cosine* of an angle A is

$$\cos(A) = \frac{\text{adjacent}}{\text{hypotenuse}}$$

for a right-angled triangle with angle A.

Some particular values of cosine:

$$\cos(0) = 1 \qquad \cos(90^\circ) = 0 \qquad \cos(180^\circ) = -1$$

The cosine of the angle between two vectors will be 1 if they are parallel, 0 if they are orthogonal, and -1 if they are antiparallel.

Scalar Product of Vectors

Suppose we have two n-dimensional vectors $\vec{x}$ and $\vec{y}$:

$$\vec{x} = (x_1, \ldots, x_n) \qquad\qquad \vec{y} = (y_1, \ldots, y_n)$$

We can calculate the cosine of the angle between them as follows:

$$\cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}||\vec{y}|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

Here $\vec{x} \cdot \vec{y}$ is the *scalar product* or *dot product* of the vectors $\vec{x}$ and $\vec{y}$, with $|\vec{x}|$ and $|\vec{y}|$ the length or *norm* of vectors $\vec{x}$ and $\vec{y}$, respectively.

Example

Matching the document "Sun, sun, sun, here it comes" against the query "sun today" we have:

$$\vec{x} = (1, 1, 1, 3, 0) \qquad\qquad \vec{y} = (0, 0, 0, 1, 1)$$

For this we can calculate:

$$\begin{aligned} \vec{x} \cdot \vec{y} &= 0 + 0 + 0 + 3 + 0 = 3 \\ |\vec{x}| &= \sqrt{1 + 1 + 1 + 9 + 0} = \sqrt{12} \\ |\vec{y}| &= \sqrt{0 + 0 + 0 + 1 + 1} = \sqrt{2} \\ \cos(\vec{x}, \vec{y}) &= \frac{3}{\sqrt{12} \times \sqrt{2}} = \frac{3}{\sqrt{24}} = 0.61 \end{aligned}$$

to two significant figures. (The actual angle between the vectors is 52°.)

Ranking Documents

Suppose $\vec{q}$ is a query vector, with document vectors $\vec{D_1}, \vec{D_2}, \ldots, \vec{D_K}$ making up the document matrix.

We calculate the K cosine similarity values:

$$\cos(\vec{q}, \vec{D_1}) \quad \cos(\vec{q}, \vec{D_2}) \quad \ldots \quad \cos(\vec{q}, \vec{D_K})$$

We can then sort these: rating documents with the highest cosine against $\vec{q}$ as the best match, and those with the lowest cosine values the least suitable.

Because all document vectors are positive — no word occurs a negative number of times — the cosine similarity values will all be between 0 and 1.

Discussion

The cosine similarity measure, as presented here, has some evident limitations.

- It only uses the frequency of individual words, not their position or ordering in relation to each other.
- It treats equally all words in the document collection, including both very common "stop" words and very uncommon words unrelated to the search.
- It does not make any connection between closely related words, like "sun", "sunny" and "sunshine".
- It takes no account of document size.

Nonetheless, more refined variations of cosine and other similar measurements based on the vector space model continue to be popular and effective in information retrieval, text mining, and clustering analysis.

Some Other Issues Around Information Retrieval

Precision and recall, as defined in this course, only evaluate a fixed set of documents returned, without taking into account any ranking. More sophisticated measures such as *precision at a cutoff* address this.

We have not considered the efficient implementation of the search for documents matching a query (or, indeed, any implementation). One method is an *inverted index* which indexes documents in a collection using all individual words as keys.

Information retrieval and ranking methods may also make use of information beyond the document itself. This might be metadata on the source and history of a document, or how other documents reference it (citations). For example, Google's *pagerank* algorithm selects and ranks web pages based on their own content and the content (and ranking) of all pages which link to them.