CHAPTER 21

Information Retrieval

This chapter covers advanced querying techniques for databases and information retrieval. Advanced querying techniques include decision support systems, online analytical processing, including SQL:1999 support for OLAP, and data mining.

Although information retrieval has been considered as a separate field from databases in the research community, there are strong connections. Distributed information retrieval is growing in importance with the explosion of documents on the world wide web and the resultant importance of web search techniques.

Considering the growing importance of all the topics covered in this chapter, some of the sections of the chapter can be assigned as supplementary reading material, even in an introductory course. These could include OLAP, some parts of data mining, and some parts of information retrieval. The material in the chapter is also suitable for laying the groundwork for an advanced course, or for professionals to keep in touch with recent developments.

Exercises

21.6 Using a simple definition of term frequency as the number of occurrences of the term in a document, give the TF-IDF scores of each term in the set of documents consisting of this and the next exercise.

Answer:

Term frequency TF(d, t) = log(1 + n(d, t)/n(d))
where n(d, t) denotes the number of occurences of term t in the document d and n(d) denotes the number of terms in the document.

using - log(1 + 1/75)
a - log(1 + 5/75)
simple - log(1 + 1/75)
definition - log(1 + 1/75)
of - log(1 + 6/75)
term - log(1 + 3/75)
frequency - log(1 + 1/75)

as - **log**(1 + 1/75)
the - **log**(1 + 1/75)
number - **log**(1 + 7/75)
occurences - **log**(1 + 1/75)
in - **log**(1 + 2/75)
document - **log**(1 + 1/75)
give - **log**(1 + 1/75)
TFIDF - **log**(1 + 1/75)
scores - **log**(1 + 1/75)
each - **log**(1 + 3/75)
set - **log**(1 + 1/75)
documents - **log**(1 + 3/75)
consisting - **log**(1 + 1/75)
this - **log**(1 + 1/75)
and - **log**(1 + 2/75)
next - **log**(1 + 1/75)
exercise - **log**(1 + 1/75)
create - **log**(1 + 1/75)
small - **log**(1 + 2/75)
example - **log**(1 + 1/75)
4 - **log**(1 + 1/75)
with - **log**(1 + 1/75)
PageRank - **log**(1 + 3/75)
inverted - **log**(1 + 1/75)
lists - **log**(1 + 1/75)
sorted - **log**(1 + 1/75)
by - **log**(1 + 1/75)
you - **log**(1 + 1/75)
do - **log**(1 + 1/75)
not - **log**(1 + 1/75)
need - **log**(1 + 1/75)
to - **log**(1 + 1/75)
compute - **log**(1 + 1/75)
just - **log**(1 + 1/75)
assume - **log**(1 + 1/75)
some - **log**(1 + 1/75)
values - **log**(1 + 1/75)
page - **log**(1 + 1/75)

21.7 Create a small example of four small documents, each with a PageRank, and create inverted lists for the documents sorted by the PageRank. You do not need to compute PageRank, just assume some values for each page.

Answer: Given 4 documents - A, B, C, D where the PageRanks are decreasing in that order, which means A has the highest PageRank and D has the lowest PageRank. We have, pages that are pointed to from more

web pages have higher PageRank. Similarly, pages pointed to by Web pages with a high PageRank will also have a higher PageRank.
One way of creating an inverted list is:

a. B, C, D all point to A. $A <- B$, $A <- C$, $A <- D$.

b. A points to B. $B <- A$.

c. B points to C. $C <- B$.

d. C points to D, $D <- C$.

21.8 Suppose you wish to perform keyword querying on a set of tuples in a database, where each tuple has only a few attributes, each containing only a few words. Does the concept of term frequency make sense in this context? And that of inverse document frequency? Explain your answer. Also suggest how you can define the similarity of two tuples using TF-IDF concepts.

Answer: Term frequency is the logarithm of the number of occurences of the term divided by the number of terms in the document. When it comes to small databases with few attributes, each containing only a few words, the concept of term frequency may not make sense. The relevance of a term may not depend on the number of occurences of the term, and also when the domain is very small the logarithmic increase we used in the term frequency many not be a good indicator.
The inverse document frequency which is the inverse of the number of documents that contain this term may not also be very relavant in this case. For example, the primary key value and some other key may be having the inverse document frequency of 1, but we cant assume their weights to be equal.
The similarity of the two tuples can be measured by the *cosinesimilarity* metric. But one major difference is only values that belong to the same attribute should be considered. Two different attributes from two tuples may be having the same value, but that doesnt increase the similarity factor.

21.9 Web sites that want to get some publicity can join a Web ring, where they create links to other sites in the ring, in exchange for other sites in the ring creating links to their site. What is the effect of such rings on popularity ranking techniques such as PageRank?

Answer: PageRank is a measure of popularity of a page based on the popularity of the pages that link to the page. It may be noted that the pages that are pointed to from many Web pages are more likely tobe visited, and thus will have a higer PageRank. Similarly, pages pointed by Web pages with a high Pagerank will also have a higher probability of being visited, and thus will have a higher PageRank. In the given scenario where Web sites join a Web ring and create links to other sites, the PageRank of all the pages increases. The number of links referencing to each page increases, which only increases the PageRank.

21.10 The Google search engine provides a feature whereby Web sites can display advertisements supplied by Google. The advertisements supplied are based on the contents of the page. Suggest how Google might choose which advertisements to supply for a page, given the page contents.

Answer: Google might use the concepts in similarity based retrieval. Here, they can give the system a document A and the set of advertisements B, and ask the system to retrieve adverstisements that are similar to A. One appraoch is to find k terms in A with highest values of TF(A,t) * IDF(t), and to use to these k terms as a query to find relevance of other documents. The metric '$cosinesimilarity$' can also be used to determine which advertisements to supply for a page, given the page contents.

21.11 One way to create a keyword-specific version of PageRank is to modify the random jump such that a jump is only possible to pages containing the keyword. Thus pages that do not contain the keyword but are close (in terms of links) to pages that contain the keyword also get a nonzero rank for that keyword.

a. Give equations defining such a keyword-specific version of PageRank.

b. Give a formula for computing the relevance of a page to a query containing multiple keywords.

Answer:

a. Give equations defining such a keyword-specific version of PageRank.

$$P[j] = \delta/N_i + (1 - \delta) * \sum_{i=1}^{N} (T[i, j] * P]i])$$

where δ is a constant between 0 and 1, N is the number of pages, N_i is the number of pages containing the keyword; δ represents the probability of a step in the walk being a jump to a page containing the keyword.

b. Give a formula for computing the relevance of a page to a query containing multiple keywords.
The relevance of a document of a document to a query comtaining multiple keywords is estimated by combining by adding the relevance measures of the page to each keyword. Some weights also might be considered.

$$r(d, Q) = \sum_{t=1}^{n} TF(d, t) * IDF(t)$$

where Q is the set of keywords of size n, TF is the $termfrequency$ and IDf is the $inversedocumentfrequency$. This measure can be further

refined if the user is permitted to specify weights $w(t)$ for terms in the query, in which case the user specified weights are also taken into account by multiplying $TF(d, t)$ by $w(t)$ in the above formula.

21.12 The idea of popularity ranking using hyperlinks can be extended to relational and XML data, using foreign key and IDREF edges in place of hyperlinks. Suggest how such a ranking scheme may be of value in the following applications:

a. A bibliographic database that has links from articles to authors of the articles and links from each article to every article that it references.

b. A sales database that has links from each sales record to the items that were sold.

Also suggest why prestige ranking can give less than meaningful results in a movie database that records which actor has acted in which movies.

Answer:

a. A bibliographic database, which has links from articles to authors of the articles and links from each article to every article that it references.
This helps us in ranking articles according to their popularity. If an article is referenced by many articles, then its more popular, so each article has a rank associated with it. and we could also find the authors for those articles.

b. A sales database which has links from each sales record to the items that were sold.
This helps us in determining wich items are the most popular. If the item is referenced by many sales records, then its more popular. So, ranking the database helps in determining the popularity of the items that were sold.

A movie which has many actors associated with it is deemed to be more popular when prestige ranking is taken into account. Same goes with the actor as well. But that may not be true in real life, the popularity of a movie or that of an actor cannot be determined by ranking the movie lists which map the actors to the movies.

21.13 What is the difference between a false positive and a false drop? If it is essential that no relevant information be missed by an information retrieval query, is it acceptable to have either false positives or false drops? Why?

Answer: False drop - A few relevant documents may not be retrieved. False positive - A few irrelevant documents may be retrieved. It is acceptable to have false positives but not any false drops when it is essential that no relevant information is to missed because by permitting false positive; the system can later filter the results away later by looking at the keywords than they actually contain, but by permitting false drops, some

relvant information is missed out. By allowing false positives and not allowing false drops, no relevant information is missed out.