Max-Planck-Institut für Informatik

Databases and Information Systems Group (AG 5)

Prof. Dr.-Ing. G. Weikum

Dipl.-Inform. Hanglin Pan
Dipl.-Inform. Sergej Sizov
Dipl.-Inform. Stefan Siersdorfer

Selected Topics in Web Information Retrieval and Mining (WS 03/04)

Assignment 6

Handout on: Thursday, December 18, 2003

Due on: solutions will be discussed on Friday, Jan 09, 2004

Exercise 6.1:

Consider a setting for top-k queries where the total score is based on a cosine or scalar product measure with query and document component values being arbitrary real numbers (including unbounded negative numbers). Discuss to what extent the Fagin family of algorithms is applicable, or to what extent these algorithms need to be modified.

// hint: first think about monotonicity of score aggregation, but even with monotonic scoring the fact that s_i values are not normalized

// to lie in [0, 1] may require modifications to the algorithms

Exercise 6.2:

Discuss specific choices for the score aggregation function, or special data instances, and corresponding algorithms that can outperform (a) FA and/or (b) TA , in terms of their total execution cost.

// hint: constant score, just pick the first k docs. max, could find top k with m*k sorted accesses, see Fagin's PODS'99 paper

Exercise 6.3:

Consider the version of Fagin's threshold algorithm that uses only sorted access. This algorithm correctly computes the top-k results, but does not necessarily return their total scores. For which score aggregation functions would it still be possible to return also the total scores? How do you have to modify the algorithm?

// hint: max is a very simple example that does not require modification of the algorithm, median is sophisticated example that requires some modification.

// see, e.g., Fagin et al.: Sigmod Record 2002

Exercise 6.4:

Consider a setting where random access to index lists, in addition to sorted access, is possible but fairly expensive. How would you modify Fagin's threshold algorithm (or the version that uses only sorted access) to take this into account?

// hint: CA algorithm, see Fagin et al.: Sigmod Record 2002

Exercise 6.5:

Consider a peer-to-peer Web search system. Every peer maintains index lists for terms, where each list contains the documents that the peer knows about and contains the corresponding term. Each list is assumed to be sorted by a term-specific score s_i in descending order (e.g., by tf*idf weight). Assume for simplicity that the scores are normalized into the range [0, 1] and are globally unified. The latter means that if two peers knew the same document they would agree on all s_i scores for this document. (This could be achieved by making all peers exchange their local statistics and agree on global tf values.) How would you perform top-k keyword queries in such a setting (e.g., employing and adapting or generalizing Fagin's threshold algorithm)?

Exercise 6.6:

An m-ary aggregation function f: [0,1]^m -> [0,1] is said to be strict if

$f(x_1, ..., x_m) = 1 <=> x_1=1$ and ... and $x_m=1$.

f is monotone if

$(x_1<=x_1'$ and ... and $x_m<=x_m') => (f(x_1, ..., x_m) <= f(x_1',..., x_m'))$.

A binary aggregation function f: [0, 1] x [0, 1] -> [0, 1] is called a triangular norm if the following four properties hold:

1) $f(0,0)=0$ and $f(x,1)=f(1,x)=x$
2) $(x_1<=x_1'$ and $x_2<=x_2') => (f(x_1,x_2) <= f(x_1',x_2'))$
3) $f(x_1,x_2) = f(x_2,x_1)$
4) $f(f(x_1,x_2),x_3) = f(x_1,f(x_2,x_3))$

Which of the following aggregation functions are monotone?
Which ones are strict? Which ones correspond to triangular norms?

a) max	
b) min	
c) bounded sum:	$f(x_1, x_2) = \min(1, x_1+ x_2)$
d) algebraic sum:	$f(x_1, x_2) = x_1 + x_2 - x_1 * x_2$
e) weighted arithmetic mean:	$f(x_1, x_2) = (w_1* x_1 + w_2* x_2) / (w_1+w_2)$
and	$f(x_1, ..., x_m) = (w_1* x_1 + ...+ w_m* x_m) / (w_1+...+w_m)$
f) geometric mean:	$f(x_1, x_2) = \mathrm{sqrt}(x_1* x_2)$
and	$f(x_1, ..., x_m) = (x_1*...* x_m) ^(1/m)$

Exercise 6.7:

Consider a monotone and strict score aggregation function, and assume that all datasets have the property that in every index list L_i all scores $s_i(d)$ are pair wise different (i.e., scores are unique within each L_i). Prove that under these assumptions Fagin's threshold algorithm is instance optimal among all algorithms including those that make "wild guesses".