AN ANALYSIS OF FRAME SLICED SIGNATURE FILES

Bülent TEZCAN Gökhan TÜR Fazlı CAN

BİLKENT UNIVERSITY

Department of Computer Engineering
and
Information Science

Technical Report BU-CEIS-94-26

AN ANALYSIS OF FRAME SLICED SIGNATURE FILES

Bülent TEZCAN Gökhan TÜR Fazlı CAN*

Department of Computer Engineering
and Information Science
Bilkent University
Bilkent, Ankara, 06533, Turkey

July 1994

Abstract

Signature files provide an efficient access tool for information retrieval. They reflect the contents of data objects in terms of bit patterns and act as a filter during query processing. In this study, an exact formula for the false drop probability estimation of multiterm queries for Frame Sliced Signature Files (FSSF) is given. The experimental results, which are obtained using the INSPEC database of 12,684 documents, are provided and compared with the theoretical results.

1. INTRODUCTION

An information retrieval system deals with various types of data, such as text, image, graphics, sound, etc. It tries to find the data items that are relevant to the submitted user queries [Salton 1989].

Signature approach is an abstraction which can be applied to formatted and unformatted data. The main idea of most signature file access methods is to represent the essence of data objects in terms of superimposed bit patterns called *signatures.* Other signature generation schemes, such as word signatures and compressed bit strings are also available in the literature [Aktug & Can 1993]. A term signature of size F bits is obtained by setting m (m << F) bits using a hash function. Object

* On sabbatical leave from the Department of Systems Analysis, Miami University, Oxford, OH 45056, USA, e-mail: fc74sanf@miamiu.acs.muohio.edu

signatures are obtained by superimposing (ORing) the signatures belonging to the terms of the object. Query signatures are obtained in a similar manner.

A characteristic of the signature file access method is that it produces false drops. The signature file may indicate that corresponding object contains a particular term, while in fact it does not. A false drop resolution is needed as a second step in the query evaluation. Figure 1. gives an example about signatures where m is 2. For retrieval efficiency various file organization methods are proposed and available in the literature [Aktug & Can 1993, Zezula, Tiberio & Rabitti 1991, Tharp 1988].

Terms	Term Signatures	
signature	0000 1100 0000	
file	1010 0000 0000	
method	0000 0110 0000	
	1010 1110 0000	<== Block Signature

Queries	Query Signatures	
signature	0000 1100 0000	True Match
hashing	1100 0000 0000	No Match
dynamic	0000 1010 0000	False Drop

Figure 1. Signature extraction and query processing.

In this paper, we propose an exact formula for the false drop probability calculation for multiterm queries in frame sliced signature files (FSSF), and theoretically and experimentally analyze the performance of FSSF in comparison with a recent paper [Lin & Faloutsos 1992]. In our presentation, we assume that the database is a collection of documents and the terminology will be accordingly.

2. FRAME SLICED SIGNATURE FILES (FSSF)

The main idea of FSSF is to minimize the random disk accesses by dividing the signature into k frames of s bits each. For each word in the document or the query, one of the k frames will be chosen by a hash function, and then another hash function is used to set distinct m bits in that frame. For each of N documents, one signature is used. Figure 2 demonstrates this method. With the sequentiality assumption, since only one frame needs to be retrieved for a single word query, i.e. only one random disk access is required, at most c frames have to be scanned for a c word query [Lin & Faloutsos 1992].

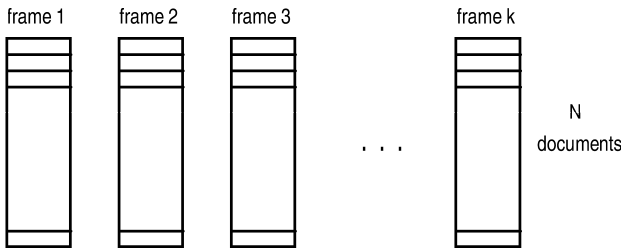

Figure 2. Frame Sliced Signature File.

3. THEORETICAL ANALYSIS OF FSSF

In FSSF, the space overhead, O_v, is defined as the signature file size divided by document size which can be formulated as follows [Lin & Faloutsos 1992].

$$O_v = ((F/8)*N + p*N)/(L*N) = ((F/8)+p)/L$$

where in this formula F indicates the signature size in bits, N indicates the number of documents, p is the pointer size in bytes and L is the average document size in bytes.

VARIABLE	MEANING
c	Number of query terms
k	Number of frames (bits)
m	Number of distinct bits to be set per term
p	Pointer size (bytes)
s	Frame size (bits)
b	Block size (bytes)
D	Number of distinct terms in a document
F	Signature size (bits)
L	Average length of a document (bytes)
N	Number of documents
F_d	False drop probability
O_v	Space overhead
T_{scan}	Average CPU time scanning 1 block
T_{seek}	Average disk seek time
T_{trans}	Average sequential disk transfer time
T_{rs}	Response time

Table 1. Important variables and their meanings.

For easy reference, the meanings of the symbols used in this paper are provided in Table 1.

3.1. Single Term Queries

The response time, T_{rs}, of FSSF for single term queries is the sum of reading the signature file, retrieving the pointers and retrieving the false drop documents, which is formulated below [Lin & Faloutsos 1992]:

$$\begin{aligned} T_{rs} &= (1 * T_{seek} + 1/(8b) * N * s * (T_{trans} + T_{scan})) \\ &+ (F_d * N * T_{seek} + F_d * N * p/b * (T_{trans} + T_{scan})) \\ &+ (F_d * N * T_{seek} + 1/b * F_d * N * L * (T_{trans} + T_{scan})) \end{aligned}$$

where T_{seek} is the average disk seek time, T_{scan} is the time to scan one disk block, T_{trans} is the average sequential disk transfer time for one block and b is block size (in bytes). F_d is the false drop probability, and by definition, the false drop probability is

$$F_d = \text{false drops} / (N - \text{actual drops}).$$

For FSSF, F_d is defined as follows [Lin & Faloutsos 1992].

$$\begin{aligned} F_d &= \sum_{t=0}^{D} B(D,t,1/k) * P_{set}(t,m) \quad \text{where} \\ B(D,t,1/k) &= \text{Prob(t words being hashed into one frame)} \\ &= \binom{D}{t} * (1/k)^t * (1-1/k)^{D-t} \\ P_{set}(t,m) &= \text{Prob(m bits found set / t words in that frame)} \\ &= (1-(1-m/s)^t)^m \end{aligned}$$

Here, we want to note that in [Lin & Faloutsos 1992], the optimum parameters of response time for FSSF are not used. A hill climbing algorithm finds better frame sizes (s) than 64 which is used by [Lin & Faloutsos 1992].

Overhead (O_v)	Signature Size (F bits)	Frame Size (s bits)	Response Time (T_{rs})
%15.6	1280	64	308.9
%15.6	1280	183	163.7
%18.8	1536	64	187.4
%18.8	1536	153	124.1

Table 2. A theoretical response time comparison of FSSF.

3.2. Multiterm Queries

The response time of FSSF for multiterm queries is given by [Lin & Faloutsos 1992] as follows.

$$\begin{aligned} T_{rs} &= C(c)*(1*T_{seek}+1/(8b)*N*s*(T_{trans}+T_{scan})) \\ &+(F_d*N*T_{seek}+F_d*N*p/b*(T_{trans}+T_{scan})) \\ &+(F_d*N*T_{seek}+1/b*F_d*N*L*(T_{trans}+T_{scan})) \end{aligned}$$

$$\begin{aligned} C(c) &= \text{Average number of distinct frames selected} \\ &= k*(1-(1-1/k)^c) \end{aligned}$$

[Lin & Faloutsos 1992] rounds the F_d of multiterm queries as the c'th power of F_d of single term queries which is very rough. A recent paper derives a formula such that if there are c words in a query, then the probability that the weight of the query, W, is equal to w is defined as follows [Grandi 1994]:

$$Pr[W=w]=\binom{s}{w}\sum_{j=0}^{w}(-1)^j\binom{w}{j}\prod_{i=1}^{D}\binom{w-j}{m}\Big/\binom{s}{m}$$

where weight of a query indicates the number of bits set in the query signature.

Here we propose an exact formula for the false drop probability:

$$F_d=\sum P(i_1,i_2,\ldots,i_q)*F_d(i_1,i_2,\ldots,i_q)$$

where $1\le i_q\le c$ and $1\le q\le \min(k,c)$ and $\sum_{j=1}^{q} i_j = c.$

In the formula that appears on the previous page

$$F_d(i_1,i_2,\ldots,i_q)=\prod_{j=1}^{q}F_d(x) \text{ where}$$

$$F_d(x)=\text{False drop probability if x terms to be found in that frame}$$

$$=\sum_{t=0}^{D}(B(D,t,1/k)*\sum_{w=m}^{x*m}Pr[W=w]*(1-(1-m/s)^t)^w)$$

$$P(i_1,i_2,\ldots,i_q)=\text{Prob}(i_1 \text{ terms in one frame, } i_2 \text{ terms in another frame, etc.})$$

$$=\left(k*\binom{c}{i_1}*(k-1)*\binom{c-i_1}{i_2}\ldots(k-q+1)*\binom{c-\sum_{j=1}^{q-1}i_j}{i_q}\right)\Bigg/\left(k^c*\prod_{j=1}^{c}n_j!\right)$$

$$=((k)_q*c!)/(k^c*\prod_{j=1}^{q}i_j!\prod_{j=1}^{c}n_j!)$$

where n_j is the number of occurences of term j in $i_1..i_q$.

To make the notation clear, an example is provided. Let the number of query terms, c, is 4. Then the false drop probability is defined as:

$$F_d=P(4)*F_d(4)+P(3,1)*F_d(3,1)+P(2,2)*F_d(2,2)$$
$$+P(2,1,1)*F_d(2,1,1)+P(1,1,1,1)*F_d(1,1,1,1).$$

In this example, P(2,1,1) denotes the probability that 2 of the 4 terms are in one frame, two others are in two other distinct frames, and $F_d(2,1,1)$ is the corresponding false drop probability, which is the product of 3 false drop probabilities, $F_d(2)$, $F_d(1)$, $F_d(1)$.

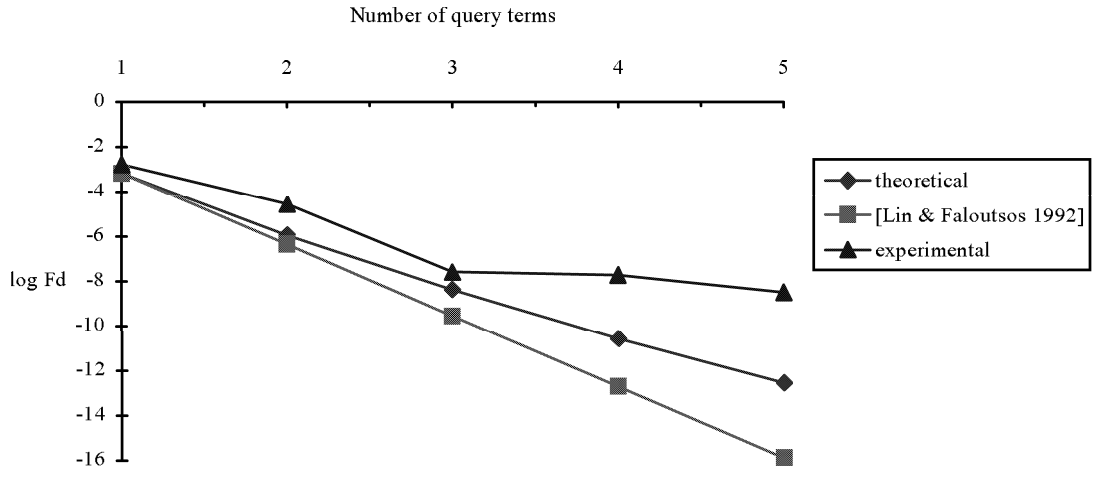

Figure 3. The false drop probability: experimental and theoretical results.

4. EXPERIMENTAL EVALUATION

The experimental analysis is performed in UNIX environment on SUN workstations by using the C language and INSPEC document database of 12,684 documents each containing 32.5 terms on the average. The average size of each document is 426.8 bytes. In order to get the false drop probability, 100,000 random queries are asked for each query size (number of query terms).

The response time of Frame Sliced Signature Files (FSSF) is minimized by using the XMaple software. When the space overhead is fixed, by using the response time formula the corresponding optimum m, k, and s values can be found. These optimum parameters are used in the theoretical and experimental analysis.

Figure 3 gives the false drop probability versus the number of query terms. The overhead is fixed to 20%, i.e. the frame size is 650 bits and there are 5 frames, which are found to be optimum. The curve labeled as "experimental" denotes the experimental false drop probability found as a result of our implementation. The curve labeled as "theoretical" is the demonstration of above formula, the third curve is the rough false drop probability of [Lin & Faloutsos 1992].

The experimental results are closer to our theoretical results, but they are not exactly the same, this is because in our experiments a logical block is a document vector and document vectors have different number of terms.

5. CONCLUSION

In [Lin & Faloutsos 1992], although they mentioned the usage of optimum parameters for FSSF, it turns out that this was not the actual case. Using the XMaple software, we conclude that better response time can be achieved with the parameters in [Lin & Faloutsos 1992]. Also the multiterm query analysis had been made with a rough false drop probability formula in [Lin & Faloutsos 1992]. We developed an exact formula for the false drop probability for multiterm queries. This formula can also be used in the generalized version of FSSF, which is called Generalized FSSF (GFSSF).

REFERENCES

AKTUG, D. and CAN F. 1993. Signature files: An integrated access method for formatted and unformatted databases. Submitted for publication.

GRANDI, F. 1993. On the signature weight in "multiple" m signature files. Submitted for publication.

LIN, Z. and FALOUTSOS, C. 1992. Frame-sliced signature files. *IEEE Transactions on Knowledge and Data Engineering*, 4, 3, 281 - 289.

SALTON, S. 1989. *Automatic Text Processing: The Transformation Analysis, and Retrieval of Information in Computer.* Addison Wesley, Reading, Massachusetts.

THARP, A. L. 1988. *File Organization and Processing*, New York: Wiley.

ZEZULA, P. and TIBERIO, P. and RABITTI, F. 1991. Dynamic partitioning of signature files. *ACM Transactions on Information Systems,* 9, 4, 336 - 369.