ELSEVIER

Information Processing Letters 65 (1998) 101–106

Information Processing Letters

Comparison of signature file models with superimposed coding

D. Dervos [a,b,1], Y. Manolopoulos [a,*], P. Linardis [a]

[a] *Department of Informatics, Aristotle University, 540 06 Thessaloniki, Greece*

[b] *Department of Informatics, Technology Educational Institute, 541 01 Thessaloniki, Greece*

Received 26 January 1996; revised 16 June 1997

Communicated by F. Dehne

Abstract

The current study considers the superimposed coding variation of the signature file method (SC-SF). A number of issues which are not clearly specified in the literature are addressed by adopting a unified approach which makes possible the comparison of two SC-SF models. Heuristic analytic derivations of earlier studies are shown to be valid by analysis and calculation. The study identifies a clear advantage of one model over the other, in a way which is quite the opposite to what has been suggested in the past. © 1998 Elsevier Science B.V.

Keywords: Signature file; Superimposed coding; Analysis of algorithms

1. Introduction

Today, information retrieval implementations utilize one, or more, of the following techniques: full text scanning, inversion, and the signature file. Full text scanning introduces zero space overhead, but involves long response times. In the cases of inversion and the signature file, an intermediary representation structure (index) is utilized providing direct links to relevant data.

Concentrating on the inverted index and the signature file, the former excels in query processing efficiency, whereas the latter involves a simpler structure and utilizes significantly less secondary storage [9]. Inverted index structures are usually implemented as B+tree variants, whereby each real text block address is stored more than once. The scheme needs to frequently undergo re-organization under intensive information insertion/updating procedures. The method is also reported to perform poorly for multiple term user queries [13].

The superimposed coding variation of the signature file (SC-SF) was originally introduced as a text indexing methodology [7]. Today, it is used in a wide range of applications, for example in office filing [4,23] and hypertext systems [8], relational, object oriented databases and extensible databases [2,12,20,24], as well as in data mining [1]. In addition, there exist hybrid schemes which combine the benefits of the signature and inverted index structures [13,14].

We consider the special case where SC-SF indexes text, although this is not restrictive in view of the general case outlined in the previous. SC-SF is a sequential index structure. To each block of text corresponds a record in the signature file. The record registers the block's address, plus a binary pattern in place of the

* Corresponding author. Email: manolopo@athena.auth.gr.

[1] Email: ddervos@athena.auth.gr.

PII S0020-0190(97)00210-X

free	0010	0011
text	1100	0010
extracted signature	1110	0011

Fig. 1. Signature extraction for $F = 8$, $D = 2$ and $m = 3$.

corresponding text: the *block signature*. The latter is of a fixed size (F bits).

By construction, the scheme does not register search key values and each text block address is stored only once. Compared to the inverted index, SC-SF is more efficient in handling new insertions and queries on parts of words. However, the scheme introduces information loss. More specifically, its output usually involves a number of false matches. The latter may only be identified by means of a full text scanning operation on every text block short-listed in the output. Also, for each query processed, the entire signature file needs to be scanned. Consequently, SC-SC involves high processing and I/O cost. In this respect, the basic SC-SF configuration is better only next to full text scanning. The query processing efficiency of SC-SF is improved by partitioning the signature file, as well as by exploiting parallel computer architectures [5,14,21].

In this study, we apply some housekeeping to the basic SC-SF methodology, the word signature construction algorithm in particular. As it will become clear in the sequel, there exists an ambiguity with regard to signature file construction.

During SC-SF index creation, each word is processed separately by a hashing function. The latter sets a constant number (m) of 1s in the $[1..F]$ range. The resulting binary pattern is termed the *word signature*. Yet one more design parameter is the blocking factor (D); text is seen to consist of fixed size logical blocks, each block involving a constant number D of non-common, distinct words. The D word signatures of a block are superimposed (bit OR-ed) to produce a single F-bit pattern, which is the block signature stored in the SC-SF record.

Fig. 1 demonstrates the SC-SF principle for $F = 8$, $D = 2$ and $m = 3$. To the right of each word appears the corresponding word signature. The two word signatures (i.e. those of *free* and *text*) are superimposed to produce the "11100011" block signature. At the query processing stage, word signatures are checked against the block signatures in the SC-SF index. When a 1 in the search word signature corresponds to a 0 in the block signature, the word is known not to be present in the block (e.g. the word signature 10010010 in Fig. refgr1). When all the 1s in the word signature match 1s in the block signature, the case may correspond either to a *hit*, or to a *false match*. For the example in Fig. refgr1, the word signature 00100011 (i.e. the word *free*) gives a hit, whereas the word signature 10000011 produces a false match.

The rate at which false matches occur is reflected by the False Drop Probability (*FDP*). *FDP* along with the storage utilized by the signature file comprise important performance parameters for SC-SF. An increase in storage utilization decreases the *FDP* rate and vice-versa. It has been shown that the performance of SC-SF is optimized when the average block signature involves an equal number of 1s and 0s [3]. This is along the lines of the entropy (or information content) maximization rule in [18]. It has also been proved that when the block signature is half-populated with 1s and half-populated with 0s, the three design parameters m, F and D satisfy Eq. (1) [3].

$$F \times \ln 2 = m \times D. \tag{1}$$

According to Christodoulakis and Faloutsos, *FDP* is calculated by Eq. (2) on the assumption that the probability distribution of the number of 1s in the $[1..F]$ range, $PR(M)$, is known. However, they admit that Eq. (2) is inconvenient for optimization [3].

$$FDP = \sum_{M=1}^{F} \left(\frac{M}{F}\right)^{m} PR(M). \tag{2}$$

In the absence of a closed formula for *FDP*, the probability distribution of the number of 1s in the average block signature, $PR(M)$, is replaced by its mean value $\bar{M}$. Some studies calculate $\bar{M}$ via the heuristic formula in Eq. (3) (e.g. [7,12,15,16]), whereas others use Eq. (4) (e.g. [10,19,22]).

$$\bar{M} = F\left(1 - \left(1 - \frac{1}{F}\right)^{mD}\right), \tag{3}$$

$$\bar{M} = F\left(1 - \left(1 - \frac{m}{F}\right)^{D}\right), \tag{4}$$

It is noted that Eq. (4) may be obtained by applying Newton's binomial expansion to the innermost term in Eq. (3), retaining the first two of the resulting component terms:

$$\left(1-\frac{1}{F}\right)^{m} \approx 1-\frac{m}{F}. \tag{5}$$

The approximation in Eq. (5) holds true since m is a positive integer and $1/F$ is a real number in the $(0..1]$ range.

A number of issues are seen to remain open. Let the case in Eq. (3) be labeled as "A1", and the case in Eq. (4) be labeled as "A2". In the absence of a closed formula expression for $PR(M)$, FDP has been calculated indirectly via a $\bar{M}$. Also, the heuristics behind the expressions in Eqs. (3) and (4) allow for questioning the validity of each approach. Lastly, assuming validity for both A1 and A2: why should one prefer one over the other? To apply some housekeeping to SC-SF, the present study establishes a unified modeling approach for the method.

2. Two SC-SF models

Christodoulakis and Faloutsos suggest in [3] that the m word signature bits need not be distinct: let their approach be labeled *Model One*, M1. On the other hand, there exist instances where exactly the opposite has been suggested. Murphry and Aktug state that SC-SF performs better when the 1s are not allowed to overlap in the word signature [17]. Their approach is herewith labeled *Model Two*, M2. To better understand the difference between the two models, we consider the word signature creation algorithm.

Letter triplets have been shown to be the best choice for information carrying text segments in the construction of the word signature [3]. For words with fewer than m letter triplets, a pseudo-random number generator is used to calculate the missing numbers. Let *warray* symbolize an array of m integers, each reflecting the corresponding letter triplet in the typical word: i.e. 1st, 2nd, 3rd, ..., mth. The pseudo-code which follows assumes the *warray*$[1..m]$ values to have been calculated already. Each letter triplet is mapped on a number in the $[1..F]$ range by utilizing a hashing function. For model M1, the algorithm for word signature creation has as follows:

```
S[1..F]:=0 \\ initialization phase
for i:=1 to m
   begin
      x:=hash(warray[i])
      S[x]:=1
   end
```

Quite analogously, the algorithm for model M2 has as follows:

```
S[1..F]:=0 \\ initialization phase
i:=1
repeat
   flag:=0
   x[i]:=hash(warray[i])
   for j:=1 to i-1 do
      if x[j]=x[i] then flag:=1
   if flag=0 then
      begin
         S[x[i]]:=1
         i:=i+1
      end
   until i=m+1
```

Evidently, M2 involves a more complex signature extraction algorithm. By intuition, for an average block signature which is half-full with 1s, M1 is expected to relate to a higher FDP rate: fewer 1s in the word signature imply higher probability for a complete match with 1s in the block signature, i.e. high FDP. Quite interestingly, this is shown not to apply in the sequel.

3. Analysis

Considering M1, the scheme has a classical probabilistic analogue where mD balls are uniformly distributed into F urns, with replacement. It is assumed that each urn can accommodate more than one ball. The theorem which follows has been proved by Eastman and Trueblood who have addressed the equivalent problem of estimating the number of block accesses in database environments [6].

Theorem 1. *Assuming replacement, the probability distribution function $PR(M)$ for the number of bit positions which register a 1 value is:*

$$PR(M) = \binom{F}{M}\sum_{i=0}^{M-1}(-1)^{i}\left(\frac{M-i}{F}\right)^{mD}\binom{M}{M-i}, \tag{6}$$

where $M \in [1..\min(mD, F)]$.

Proof. The probability for M urns to receive all mD balls equals $(M/F)^{mD}$. The event includes cases where the mD balls end up in fewer than M urns. The scheme calls for the application of the inclusion-exclusion principle. The $\binom{M}{M-i}$ term in Eq. (8) reflects the number of times $M-1$, $M-2$, ..., urns may be selected from a total of M. Lastly, the $\binom{F}{M}$ term considers the number of times M, out of F, urns are selected.

Given the closed formula for $PR(M)$, the expected value $\bar{M}$ is calculated [6]:

$$\bar{M} = \sum_{i=1}^{\min(mD,F)} iPR(i) = F\left(1 - \left(1 - \frac{1}{F}\right)^{mD}\right). \quad (7)$$

Eq. (7) is identical to Eq. (3), which means that the A1 approach in fact corresponds to the M1 case. □

Model Two (M2) considers each word signature to involve m distinct 1s in the $[1..F]$ range. Different words may be mapped on to signatures which may have some (or even all) of their 1s in common, of course.

Theorem 2. *Assuming non-replacement, the probability distribution function $PR(M)$ for the number of bit positions which register a 1 value is:*

$$PR(M) = \frac{1}{\binom{F}{m}^{D-1}} \sum_{j=0}^{M-m} (-1)^{M-m+j} \binom{F-m}{j} \times \binom{F-m-j}{M-m-j} \binom{m+j}{m}^{mD}, \quad (8)$$

where $M \in [1..\min(mD, F)]$.

Proof. By the end of stage-k, i.e. after the kth word signature has been superimposed on to the block signature, let $PR_k(n)$ symbolize the probability for having n 1s. Obviously, $PR_1(m)=1$. By the end of stage-2, there will be two extreme cases: (a) all the m 1s of word-2 coincide with the 1s of word-1, and (b) $2m$ discrete bit positions register 1s in the block signature. The distribution of 1s for a D-word block signature is given by the recursive expression:

$$PR(M) = PR_D(M) = \frac{1}{\binom{F}{m}} \sum_{j=0}^{m} PR_{D-1}(M-j) \times \binom{F-m-j}{j} \binom{M-j}{m-j}. \quad (9)$$

Murphry and Aktug apply linear algebra techniques and replace the corresponding Markov chain by the closed formula expression in Eq. (8) [17]. □

Grandi applies his γ-transform methodology and proves that Eq. (10) is equivalent to Eq. (8) [11].

$$PR(M) = \frac{\binom{F}{M}}{\binom{F}{M}^{D}} \sum_{j=0}^{M} (-1)^j \binom{M}{j} \binom{M-j}{m}^{D}. \quad (10)$$

One advantage Eq. (10) has over Eq. (9) is that it makes easier the calculation of the $\bar{M}$ value [11]:

$$\bar{M} = \sum_{i=1}^{\min(mD,F)} iPR(i) = F\left(1 - \left(1 - \frac{m}{F}\right)^{D}\right). \quad (11)$$

Eq. (11) is identical to Eq. (4), which means that the A2 approach in fact corresponds to the M2 case.

4. Calculation results

Calculations relating to the two SC-SF models and the expressions of Eqs. (3), (4), (6) and (8), were carried out in a Unix/ANSI C programming environment. Overflow and underflow errors occurred while calculating intermediary results for the expressions in Eqs. (6) and (8). The problem was dealt with by writing special code. More specifically, decimal numbers were registered in long arrays, each with up to 4,000 elements. This was done in order to overcome the restriction imposed by the double floating number data type in C which stores only up to sixteen decimal digits for a number. Thirty five different SC-SF configurations were considered and some seventy calculations were conducted.

The results obtained have shown M1 to almost be identical to M2 for as long as m obtains values in the $[1..10]$ range (Fig. 2). The two start to deviate significantly for $m \geqslant 10$ (Fig. 3). M1 and M2 were

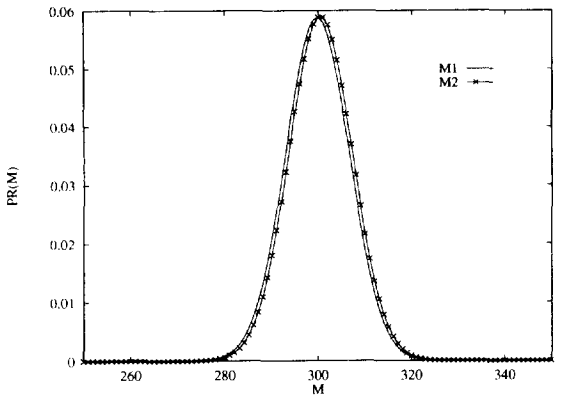

Fig. 2. $PR(M)$ distribution for $F = 600$, $m = 5$, and $D = 83$.

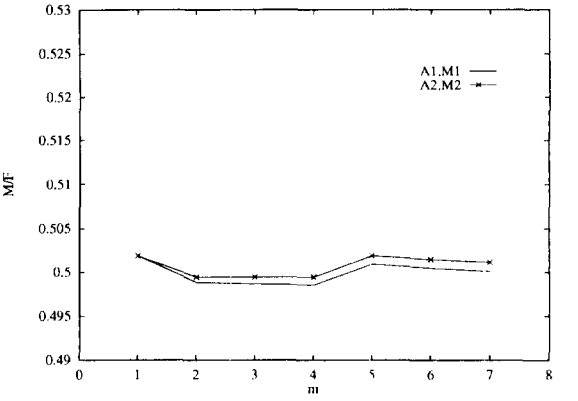

Fig. 4. $\bar{M}/F$ as a function of m for $D = 100$ and variable F.

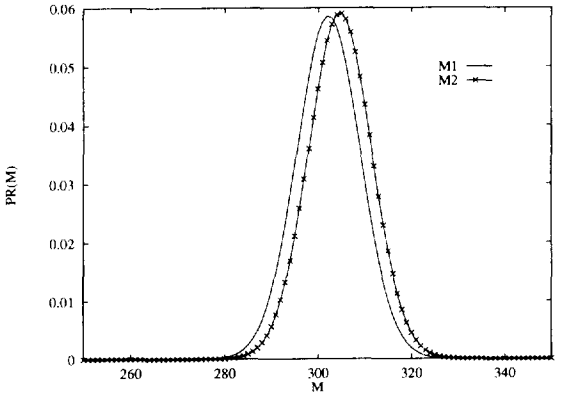

Fig. 3. $PR(M)$ distribution for $F = 600$, $m = 15$, and $D = 28$.

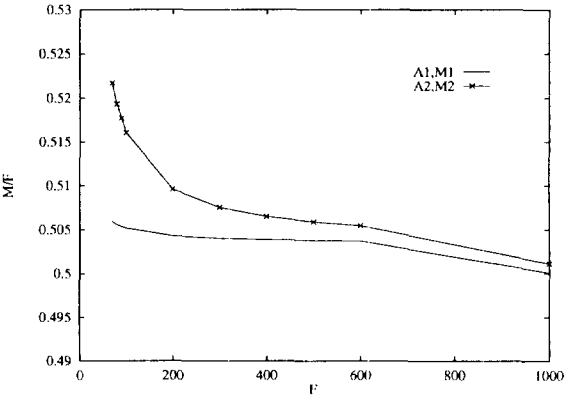

Fig. 5. $\bar{M}/F$ as a function of F for $m = 7$ and variable D.

measured to involve practically the same variance of the $PR(M)$ distribution around the mean value M. More specifically, M1 has its peak remain closer to the $M = F/2$ value which, as stated in Section 1, optimizes the performance of SC-SF.

By considering a number of SC-SF configurations whereby F, m, D satisfy Eq. (1), $\bar{M}/F$ was plotted as a function of m when D is kept constant (Fig. 4), as well as a function of F when m is kept constant and D varies accordingly (Fig. 5). Figs. 4 and 5 show M1 and M2 to coincide with A1 and A2 respectively. This is in tune with the analysis in Section 3. As stated in Section 1, SC-SF performs optimally when $\bar{M}/F$=0.5. Fig. 4 shows M1 and M2 to give almost identical $\bar{M}/F$ values and remain close enough to the optimal 0.5 value. However, M2 in Fig. 5 deviates significantly from the optimal $\bar{M}/F = 0.5$ line, the moment when M1 remains closer to it, especially for small F values. M1 is seen to behave clearly better than M2. It is also noted that the curves in Figs. 4 and 5 indicate that Eq. (1) alone does not suffice in telling the "optimality" of an SC-SF configuration; quite possibly, there exists room for further improvement in this respect.

5. Epilogue

The present study has been motivated by an ambiguity which exists in the SC-SF bibliography with respect to the word signature extraction algorithm. Two SC-SF models were considered:

one whereby the word signature involves m distinct numbers in the $[1..F]$ range (M2), and one whereby the numbers are not necessarily distinct (M1). Closed, analytic formulae for the calculation of the $PR(M)$ and FDP values are provided, together with the corresponding calculation results.

The contribution made may be summarized as follows:

(1) The A1 approach is valid and relates to model M1.
(2) Analogously, A2 is also a valid approach and relates to model M2.
(3) M1 remains closer to the optimal SC-SF configuration when compared to M2. This is opposite to what is being suggested in some of the relevant literature (e.g. [17]).
(4) Calculation results suggest that $F \ln 2 = mD$ may not by itself comprise a single, absolute, quality indicating formula for SC-SF; there may exist room for further improvement in this respect.

References

[1] H. Andre-Jönsson, D. Badal, Using signature files for querying time-series data, in: Proc. 1st European Symp. on Principles of Data Mining and Knowledge Discovery, 1997.
[2] W.W. Chang, H.J. Schek, A signature access method for the STARBURST database system, in: Proc. 19th VLDB Conf., 1989, pp. 145–153.
[3] S. Christodoulakis, C. Faloutsos, Design considerations for a message file server, IEEE Trans. Software Engineering 10 (2) (1984) 201–210.
[4] S. Christodoulakis, M. Theodoridou, F. Ho, M. Papa, A. Pathria, Multimedia document presentation, information extraction and document formation in MINOS – A model and a system, ACM Trans. Office Inform. Systems 4 (4) (1986) 345–383.
[5] P. Ciaccia, P. Zezula, Declustering of key-based partitioned signature files, ACM Trans. Database Systems 21 (3) (1996) 295–338.
[6] C.M. Eastman, R.P. Trueblood, Occupancy models for the estimation of block accesses, Computer J. 35 (6) (1992) 654–658.
[7] C. Faloutsos, Access methods for text, ACM Comput. Surveys 17 (1) (1985) 49–74.
[8] C. Faloutsos, R. Lee, C. Plaisant, B. Shneiderman, Incorporating string search in a hypertext system: User interface and signature file design issues, HyperMedia 2 (3) (1990) 183–200.
[9] W. Frakes, R. Baeza-Yates (Eds.), Information Retrieval: Data Structures and Algorithms, Prentice-Hall, Englewood Cliffs, NJ, 1992.
[10] F. Grandi, P. Tiberio, P. Zezula, Frame-sliced partitioned parallel signature files, in: Proc. 1992 ACM SIGIR Conf., 1992, pp. 286–297.
[11] F. Grandi, On the signature weight in multiple m signature files, ACM SIGIR Forum 29 (1) (1995) 20–25.
[12] Y. Ishikawa, H. Kitigawa, N. Ohbo, Evaluation of signature files as set access facilities in OODBs, in: Proc. 1993 ACM SIGMOD Conf., 1993, pp. 247–256.
[13] H. Jagadish, C. Faloutsos, Hybrid index organizations for text databases, in: Proc. 1992 Conf. on Extending Database Technology (EDBT), 1992, pp. 310–327.
[14] D.L. Lee, Massive parallelism on the hybrid text-retrieval machine, Inform. Process. Management 31 (6) (1995) 815–30.
[15] Z. Lin, C. Faloutsos, Frame-sliced signature files, IEEE Trans. Knowledge and Data Engineering 4 (3) (1992) 281–289.
[16] Z. Lin, Concurrent frame signature files, Distributed and Parallel Databases 1 (3) (1993) 231–249.
[17] E.S. Murphry, D. Aktug, A Markov chain model for the query weight problem, J. Appl. Statist. Sci. 2 (4) (1995) 387–396.
[18] E. Reingold, W. Hansen, Data structures in Pascal, Little Brown, 1986, pp. 410–413.
[19] R. Sacks-Davis, K. Ramamohanarao, A. Kent, Multikey access methods based on superimposed coding techniques, ACM Trans. Database Systems 12 (4) (1987) 655–696.
[20] R. Sacks-Davis, A. Kent, K. Ramamohanarao, J. Thom, J. Zobel, Atlas: A nested relational database system for text applications, IEEE Trans. Knowledge and Data Engineering 7 (3) (1995) 454–70.
[21] C. Stanfill, B. Kahle, Parallel free-text search on the connection machine system, Comm. ACM 29 (12) (1986) 1229–1239.
[22] S.Y. Sung, Performance analysis of superimposing-coded signature files, in: Proc. 1993 Conf. on Foundations On Data Organization (FODO), 1993.
[23] D. Tsichritzis, S. Christodoulakis, P. Economopoulos, C. Faloutsos, A. Lee, D. Lee, J. Vanderbroek, C. Woo, A multimedia office filing system, in: Proc. 9th VLDB Conf. 1983.
[24] H.S. Yong, S. Lee, H.J. Kim, Applying signatures for forward traversal query processing in object-oriented databases, in: Proc. 10th IEEE Internat. Conf. on Data Engineering, 1994, pp. 518–525.