Optimal Weight Assignment for Signature Generation

Chun-wu Roger Leng Dik Lun Lee

Department of Computer and Information Science
The Ohio State University
2036, Neil Ave
Columbus, OHIO 43210-1277

April 24, 1995

Abstract

Previous work on superimposed coding are characterized by two aspects. First, it is in general assumed that signatures are generated from *logical* text blocks of the same size. That is, each block contains the same number of unique terms after stopword and duplicate removal. We call this approach the fixed size block (FSB) method, since each text block has the same size as measured by the number of unique terms contained in it. Second, with only a few exceptions [7,8,9,16], most previous work has assumed that each term in the text contributes the same number of 1's to the signature (i.e., the weight of the term signatures is fixed).

The main objective of this paper is to derive an optimal weight assignment which assigns weights to document terms according to their occurrence and query frequencies to minimize the false drop probability. The optimal scheme can account for both uniform and nonuniform occurrence and query frequencies, and the signature generation method is still based on hashing rather than on table lookup.

Further, a new way of generating signatures, the fixed weight block (FWB) method, is introduced. FWB controls the weight of *every* signature to a constant, whereas in FSB, only the *expected* signature weight is constant. We have shown that FWB has a lower false drop probability than that of the FSB method but its storage overhead is slightly higher. Other advantages of FWB are that the optimal weight assignment can be obtained analytically without making unrealistic assumptions and that the formula for computing the term signature weights is simple and efficient.

Categories and Subject Descriptors: H.2.2 **[Database management]:** Physical Design - *access methods*; H.3.3 **[Information Storage and Retrieval]:** Information Search and Retrieval - *retrieval models*; H.3.6 **[Information Storage and Retrieval]:** Library Automation; H.4.1 **[Information Systems Applications]:** Office Automation; I.7.0 **[Text Processing]:** Text Editing

General Terms: Design, Performance

Additional Keywords and Phrases: Document retrieval, information retrieval, text retrieval, superimposed coding, signature file, optimization, access method, coding methods

[1]This research has been supported by the Applied Information Technologies Research Center (AITRC), Columbus, Ohio, from 4/1/88 to 6/30/90.

1 Introduction

Many applications such as library systems and office automation systems must access to a large amount of textual data. Since traditional database management systems are mainly for handling formatted records, they have limited facilities for retrieving textual data. One approach which has been widely studied as an efficient access method for text retrieval is the *signature file method* [6,9,12]. A signature file has a storage overhead much smaller than that of an inverted file. Its simple file structure facilitates database maintenance (e.g., fast insertion and deletion) as well as parallel search on special hardware [10] and general-purpose parallel computers [19]. Also, it can support both text retrieval and multiattribute retrieval in relational databases. However, signature files suffer from two drawbacks: (1) They are slow compared to inverted files, and (2) they produce false drops; that is, the signature file may identify a block as one satisfying the query but in fact it is not.

The first problem stems from the fact that the size of the signature file is proportional to the size of the database. Thus, a straightforward, exhaustive, search algorithm on the signature file will have a linear time complexity with respect to the database size, and the performance will become a problem for large databases.

Many techniques have been proposed to improve the search performance of the signature file. Storing the file as a transposed file [10,15] allows the system to retrieve only the bit slices which are necessary to evaluate the query. Sophisticated file structures such as the two-level and multi-level superimposed coding methods [2,3,11,16,17], the indexed descriptor file method [14], and the partitioned signature file methods [12,13] have been proposed to further improve the search speed. The general idea of these methods is to avoid searching the whole signature file by using a nonsequential file structure, typically using tree search or hashing techniques.

False drops introduced by the signature file also seriously degrade the system performance. The problem is particularly acute for large databases, since the number of false drops produced by a query is proportional to the number of unqualified documents. Thus, when the size of the database is large and the query is highly selective (in the extreme case, an unsuccessful search), the number of false drops will be very large even though the false drop probability is small. Therefore, a substantial amount of efforts has been devoted to the study of optimal algorithms for generating signatures from the text and to the estimation of false drop probability [5,6].

In previous work on superimposed coding, it has generally been assumed that signatures are generated from *logical* text blocks of the same size. That is, every logical text block contains the same number of distinct terms after stopword removal. We call this approach the fixed size block (FSB) method. Also, with only a few exceptions [7,8,9,16], most previous work assumed that every term signature has the same weight (i.e., containing the same number of 1's). In previous work, the condition that every signature is generated from the same number of distinct terms is necessary for obtaining design parameters and evaluating the false drop probabilities. The uniform weight assumption is clearly a simplification, and there have been investigations on weight assignment methods that assign weights to terms based on occurrence and query frequencies [8].

The main objective of this paper is to derive an optimal way of assigning weights to document terms based on their occurrence and query frequencies to minimize the false drop probability. Our optimal scheme can account for both uniform and nonuniform occurrence and query frequencies. The signature generation method is still based on hashing rather than table lookup [7]. We evaluate the performance of the optimal weight assignment against the uniform weight assignment as well as two other assignment schemes.

Further, we introduce a new way of generating signatures, namely, the fixed weight block (FWB) method, in which the weight of *every* signature is controlled to a constant. In the FSB method, only the *expected* signature weight is constant. Thus, the FWB method is intuitively better than the FSB method, since there is a nonzero probability in the FSB method for signatures with a large number of ones to be generated, thus increasing the number of false drops. The FWB method completely eliminates this situation. We have shown by analysis and simulation that the false drop probability of the FWB method is indeed smaller than that of the FSB method, although its storage overhead is slightly higher. Another advantage of the FWB method is that it allows an optimal weight assignment to be obtained analytically without making unrealistic assumptions, mainly because of the condition that every signature in FWB has the same weight.

In Section 2, we review the traditional signature file method and the term weight assignment methods proposed in the literature. Then, a new blocking approach, the Fixed Weight Block (FWB) method, is introduced. In Section 3, we obtain the optimal term weight assignment based on the FWB method. The false drop probability and the storage overhead of the optimal method is evaluated and compared to three other weight assignment strategies. In Section 4, the performances of the FWB and FSB methods are evaluated and compared. Both analytical and simulation results are presented in Section 3 and 4.

2 Overview

A signature file is basically a compressed version of a text. Thus, searching the signature file is substantially faster than searching the text itself. However, the signature file approach is different from traditional compression techniques (e.g., Huffman coding) in two major aspects. First, the signature file typically allows information loss (i.e., the compression is non-reversible) in order to achieve a good compression rate. Second, search on the signature file must be efficient. In particular, a search must be able to start at any point in the signature file. Traditional compression techniques, however, are reversible and emphasize on compression rate rather than search time. Thus, the signature file method has in itself many interesting research issues involving the organization and search techniques of the signature file and the tradeoff between storage overhead and information loss.

There are many signature file techniques proposed in the literature [4,5]. In this section, we provide an overview of the superimposed coding method, which is pertinent to the understanding of this paper. A summary of important symbols frequently used in this paper can be found in Table 1.

The typical steps involved in generating a signature is illustrated in Figure 1. The first step is to hash each noncommon term (word) in the text into a fixed length bit string called a *term signature*. Then, a number of term signatures are superimposed together to form a *block signature* S_i [20,21]. When a query is received, a *query signature* S_Q is generated in the same way and compared to every signature in the signature file. Figure 1 depicts the three possible outcomes of a comparison: (1) a (true) match in which the text indeed contains the query term; (2) a no-match indicating the text does not contain the query term; (3) a false-drop match in which the block is identified as containing the query term but in fact it is not. It can be observed that there are three steps involved in signature generation: (1) to determine the number of 1's to set in a term signature; (2) to determine the positions to be set; and (3) to determine the number of term signatures to superimpose into a block signature. All these factors have influence on the false drop probability of the signature file. Previous work typically assume that the weight (i.e., the number of 1's) of every term signature is the same. Second, the bit positions are chosen randomly once the term signature weight is determined. Finally, the number of unique terms in a block is fixed,

B_i	ith block in FWB approach
$\hat{B}_i$	ith block in FSB approach
m	length of a signature
f_i	number of blocks containing term i
p_i	number of blocks which do not contain term i
$P(Q_i)$	probability that Q_i is selected as a query
S_i	ith signature
S_Q	query signature
T	no. of distinct terms in a block
V	the vocabulary size
w_{B_i}	number of ones (weight) of ith FWB block signature
$w^+_{B_i}$	summation of terms' weights in the ith FWB block
$w_{B_{avg}}$	average weight of all the FWB blocks in database
$w^+_{B_{avg}}$	average value of all the $w^+_{B_i}$ in database
w_Q	weight of query signature

Table 1: List of Symbols.

thus determining the number of term signatures to be superimposed into a block signature. Methods in which block signatures are generated from logical text blocks containing the same number of distinct terms are called *fixed size block* (FSB) methods.

Since false drops can seriously degrade the performance of a signature file, a lot of the recent work has focused on reducing the false drop probability. Signature schemes which have lower false drop probabilities than superimposed coding have been proposed [5,6].

Figure 1: Generating signatures from a text block.

Using information theory, Faloutsos and Christodoulakis have established the relationship between the information loss when signature extraction is performed and the false drop probability of the signature file [7]. They also attempted to find an optimal signature extraction method for assigning term signatures to document terms. For uniform occurrence and query frequencies, they demonstrated that the optimal way was to assign the same number of distinct terms to each signature. However, there are a few problems with their results:

1. The optimal solution is based on a lookup table which maps terms in the vocabulary to a pool of term signatures. An algorithm based on hashing is preferable.
2. For nonuniform occurrence and query frequencies, they only obtained a suboptimal algorithm for mapping terms to term signatures.
3. They assumed that each text block contains only one term. Thus, they avoided tackling the problem of false drops introduced when term signatures were combined into block signatures (e.g., using superimposition). In other words, they only consider the step of term signature assignment without considering the effect of combining term signatures into block signatures.

Thus, although their study forms a very important first step toward a theoretical investigation of term signature assignment method, much still needs to be done. In a different report, Faloutsos and Christodoulakis proposed another assignment strategy which accounts for nonuniform occurrence and query frequencies [8]. In their method, the set S of all possible terms in the text file is partitioned into n classes S_1, S_2, $\cdots$, S_n, such that S_i's are disjoint and $S = S_1 \cup S_2 \cup ... \cup S_n$. Terms with occurrence frequencies falling into the same range are assigned to the same class and to the same weight. The expected number of terms in a block that are assigned to class j, D_j, is assumed given. Thus, $D_1, D_2, ..., D_n$ are known and the expected number of distinct terms in a block is $D = \sum_{i=1}^{n} D_i$. Given the query frequencies, the length of the signatures, and $D_1, D_2, ...D_n$, an optimal weight assignment strategy for minimizing the false drop probability is obtained. However, a careful study of their analysis reveals several undesirable assumptions:

1. The optimal weight assignment is obtained based on the *expected value* of D_j's without considering the distribution of the D_j values. Thus, the result is optimal only for "well-behaved" distributions (e.g., when the value of D_j for a specific class j is the same for

all block i). For real databases, in which the distribution may not be well-behaved, the assignment strategy may deviate from the optimal point. It is necessary to include the effect of the distribution in the analysis. Further, if D_j's are only expected values, the sizes of the text blocks may be different from each other. This is contradictory to the general condition that every block has a fixed size.

2. Their analysis requires the assumption that the weight for each block signature is constant but in fact only the expected weight is fixed.

We can see that the problem mainly stems from the fact that the analysis is based on expected values. When the input parameters are only expected values, there is no guarantee that the result of the analysis can predict the general behaviors of a real system.

In this paper, we obtain an optimal weight assignment based on a new signature generation technique in which the weights of the block signatures are controlled to a constant. We call our method the *fixed weight block* (*FWB*) method. The FWB method can be implemented by generating term signatures from the text incrementally and superimposing them into the block signature as they are generated until the block signature weight reaches the pre-defined value. The terms used in generating the signature then form a logical block of the document. An informal algorithm is given below:

1. Initialize *count* to zero and set the block signature to zeros;
2. Repeat the following steps until *count* $\geq$ the pre-defined constant:
   (a) Get next term from the document;
   (b) Generate term signature and superimpose it into the block signature;
   (c) Increment *count* by the number of *new* bit positions set in the block signature;
3. Output the block signature and logical text block; goto (1).

Observe that the final weight of the block signature may exceed the pre-defined constant, but the deviation should not be large. Also duplicates are automatically eliminated. Note that both the FSB and FWB methods yield variable-length physical text blocks. For the FSB method, since only the number of unique terms in a block is fixed, the actual block length is almost guaranteed to vary since a block may contain multiple occurrences of the same term and common terms removed by a stop-word list. Hence, the FWB method is not inferior than the FSB method in this aspect.

As discussed before, the FWB method intuitively produces fewer false drops than the FSB method. This is because the FSB method may produce signatures with large weights although the expected weight of all signatures remains fixed. The heavy signatures have a higher chance to be retrieved as false drops, and thus increasing the overall false drop probability. This can be easily observed by considering two signatures, one containing all zeros and the other all ones. The false drop probability of these two signatures will be much higher than that of two signatures with exactly one-half of the bits set. In the FWB method, this situation is completely eliminated. Further, since the block signature weights are constant in the FWB method, the analysis for obtaining the optimal weight assignment for term signatures can be simplified, and the analysis is not dependent on expect values of the parameters.

3 Optimal Weight Assignment

A good weight assignment scheme must satisfy the following three criteria:

- Since the occurrence frequencies in the database is nonuniform (e.g., natural-language text follows the Zipf's law [18]), the assignment should take into account the occurrence and query frequencies. It must be able to handle both uniform and nonuniform occurrences and query frequencies, in such a way that the uniform case can be treated as a special case in the formulation.
- The scheme must be optimal in minimizing the false drop probability.
- Lookup tables for direct mapping of terms into their signatures should be avoided.

For the last criterion, it must be noted that lookup tables are unavoidable for finding the occurrence and query frequencies of a term (unless an approximation based on some distribution function such as the Zipf's law must be used). If lookup tables are used to directly obtain optimal signature assignments for the terms (as in [7]), the storage overhead would be tremendous for any realistic environment, and lookup tables are inflexible for changes. Thus, it is desirable to obtain signatures algorithmically once the occurrence and query frequencies are known.

The problem of signature generation can be separated into two parts: (1) to determine the weight of each term; and (2) to choose the positions to set in a signature after its weight has been decided. In this section, we address the problem of weight assignment. Once the weight is determined, the bit positions are chosen randomly. This avoids the use of lookup table, and the optimal weight assignment can still be determined.

3.1 False Drop Probability

Our optimal solution is essentially different from the optimal solutions reported by Roberts [15] and Faloutsos and Christodoulakis [8]. Their optimal solutions are based on the condition that each block contains a fixed number of unique terms. For databases containing the same number of unique terms, the size of the signature files (thus the overheads) are the same. The optimization problem is to determine the term weight to minimize the false drop probability. An optimal point exists since when the weight of a term signature increases, there are two conflicting effects on the false drop probability. First, the weights of the block signatures to which this term belongs will increase and thus increasing the false drop probability; second, the false drop probability will decrease when the term is selected as a query since more bits are set in the query signature.. Hence, an optimal point can be obtained at which the false drop probability is minimized. It turns out that the optimal point is when the expected numbers of ones and zeros in the block signatures are the same. Based on this condition, the term weight (assuming the same for every term) can be found easily.

In our optimization method, we first determine what parameters must be fixed. To this end, we define a parameter as *monotone* if and only if the false drop probability increases or decreases monotonously with respect to this parameter. For example, the average term weight is not monotone in the FSB approach since, as mentioned before, the false drop probability could be increasing or decreasing with respect to it when the block size is fixed. However, it is monotone in the FWB approach, because when the average term weight increases the average query weight also increases, resulting in a smaller false drop probability. However, increasing the average term signature weight would not increase the block signature weight in FWB by definition. It only means fewer terms are included in a signature. This will, of course, increase the storage overhead, which is a trade-off to be determined at design time. Another monotone parameter is the signature length m. It is monotone in both the FSB and FWB methods, since in both cases the false drop probability decreases monotonously when m increases. It is obvious that all monotone parameters must be fixed since there is no bounded optimal values for these parameters.

In this section, we obtain the optimal weight assignment based on the FWB approach. The first theorem introduces a weight assignment to achieve an optimal false drop probability when the monotone parameters are fixed. Moreover, our new optimal assignment is more accurate than Faloutsos and Christodoulakis' method [8] in that it assigns each individual term with a specific weight depending on its occurrence frequency and the query probability, instead of assigning every term in a class with the same weight.

Let the query probability of term j be denoted by $P(Q_j)$, then the expected false drop probability is a product of the number of false drops under query Q_j and the value $P(Q_j)$. Consequently, the overall performance, measured by the expected false drop probability Ex, with respect to all the possible query terms can be expressed as

$$Ex = \sum_{j=1}^{V} \sum_{i=1}^{p_j} \frac{\binom{w_{B_{i,j}}}{w_{Q_j}}}{\binom{m}{w_{Q_j}}} P(Q_j),$$

where $\{B_{i,j} | i = 1, 2, ..., p_j\}$ is the set of blocks which don't contain Q_j, and V is the vocabulary size. It may be argued that the value V should have been replaced with $V - x$, where x is the number of terms that are too common to be used in a query. We use this formulation later in Theorem 1. The equation can be simplified further to:

$$\begin{aligned} Ex &= \sum_{j=1}^{V} \sum_{i=1}^{p_j} \frac{w_{B_{i,j}}(w_{B_{i,j}} - 1) \cdots (w_{B_{i,j}} - w_{Q_j} + 1)}{m(m-1) \cdots (m - w_{Q_j} + 1)} P(Q_j) \\ &\approx \sum_{j=1}^{V} \sum_{i=1}^{p_j} \left(\frac{w_{B_{i,j}}}{m} \right)^{w_{Q_j}} P(Q_j) \end{aligned}$$

Since $w_{B_{1,j}} = w_{B_{2,j}} = ... = w_{B_{p_j,j}} = w_{B_{avg}}$ for all $j = 1, ..., V$,

$$Ex = \sum_{j=1}^{V} p_j \left(\frac{w_{B_{avg}}}{m} \right)^{w_{Q_j}} P(Q_j) \tag{1}$$

We have observed that both m and $w_{B_{avg}}$ are pre-defined constants for every i in the FWB approach, since they are monotone parameters. Thus if we assume that the vocabulary size is a constant, then the only parameters which could affect Ex are the query term weight w_{Q_j},[2] the query probability $P(Q_j)$, and the number of blocks not containing Q_j. That is, the problem becomes one of deciding the value of w_{Q_j}, in terms of $P(Q_j)$ and p_j, such that Ex is optimal. Of course, it is obvious that Ex becomes better when w_{Q_j} increases. However, a larger w_{Q_j} will also imply a larger term weight and in turn a larger storage overhead in the FWB method. Hence, the optimal solution should take into consideration the fact that the storage overhead should be fixed (or the storage overhead is a pre-defined constant) when comparing it with other weight assignment methods.

In the next theorem, an optimal solution is introduced under two restrictions. The first restriction fixes the storage overhead of a weight assignment method and the second restriction forces the sum of $P(Q_j)$ to be 1. We represent these two restrictions with two mathematical equations,

1. $\sum_{j=1}^{V} f_j w_{Q_j} = C$,
2. $\sum_{j=1}^{V} P(Q_j) = 1$.

C is in fact the total number of ones produced by all of the terms (multiple occurrences of a term are counted in the formula). It will be shown in Theorem 2 that the summation of the first equation is the storage overhead of a weight assignment method; that is, the storage overhead of a weight assignment method is fixed if its C value is fixed. Like other signature methods, the storage overhead is always a design parameter in the sense that the number of false drops will decrease if a larger storage overhead can be afforded. Thus, C is a design parameter to be supplied by the designer.

Theorem 1 *In the FWB method, if the storage overhead is fixed, then the optimal false drop probability can be obtained when the weight of each term is related to the query probability, the term frequency, and the number of blocks not containing the query as follows:*

$$w_{Q_j} = \begin{cases} 0 & \textit{when } P(Q_j) = 0 \textit{ or } p_j = 0 \\ m & \textit{when } f_j = 0 \\ \frac{C}{\sum_{j=1}^{t} f_j} + \sum_{j=1}^{t} \frac{f_j}{\sum_{j=1}^{t} f_j} \log_d \frac{p_j P(Q_j)}{f_j} + \log_d \frac{f_j}{p_j P(Q_j)} & \textit{otherwise} \end{cases}$$

[2]The set of query terms are the same as the set of document terms (except for some common terms for which $P(Q_j)$ is zero). Since the weight of a term is the same as its query weight, we will obtain the term weights by finding the query weights.

where $t = V - x$ is the vocabulary size excluding terms with $P(Q_j) = 0$ or $p_j = 0$, $C = \sum_{j=1}^{V} f_j w_{Q_j}$, and $d = w_{B_{avg}}/m$.

Proof: In the FWB method, $w_{B_{avg}} = w_{B_i}$ is a constant for all i and is strictly less than m, or $d = w_{B_{avg}}/m < 1$.

First, we consider the first two cases. That $P(Q_j) = 0$ means Q_j will never be used in a query. Further, when a term belongs to every block (i.e., $p_j = 0$), all blocks will qualify if the term is used in the query. In either case, the term is useless and should not be encoded in the signature file in order not to increase the false drop probability and the storage overhead. Thus, terms in this class are assigned with zero weight. A term having $f_j = 0$ means it is not contained in any block. Since it won't have any effect on the signature file, it should be assigned a weight as large as possible (i.e., m) to reduce the false drop probability when it is used in a query.

In order to find the optimal solution of Equation (1) under the following restrictions:

1. $\sum_{j=1}^{t} f_j w_{Q_j} - C = 0$
2. $\sum_{j=1}^{t} P(Q_j) - 1 = 0$

we use the *Lagrange's method* [1] to solve the optimization problem. First, we consider $w_{Q_1}, w_{Q_2}, \ldots, w_{Q_t}$ as t variables formulated in a new function, Φ, composed of Equation (1) and the above two restrictions:

$$\begin{aligned}&\Phi(w_{Q_1}, w_{Q_2}, \ldots, w_{Q_t})\\ &= \sum_{j=1}^{t} p_j (d)^{w_{Q_j}} P(Q_j) + \lambda_1 \cdot \left(\sum_{j=1}^{t} f_j w_{Q_j} - C\right) + \lambda_2 \left(\sum_{j=1}^{t} P(Q_j) - 1\right)\end{aligned}$$

It is obvious that the function Φ is first-order differentiable with respect to each of the t variables. Our goal is to construct $t+2$ equations to obtain the values of the $t+2$ variables, $w_{Q_1}, w_{Q_2}, \ldots, w_{Q_t}, \lambda_1, \text{and} \lambda_2$, as well as the relationship between w_{Q_j}, p_j, f_j and $P(Q_j)$.

Using the Lagrange's method, there are t partial derivatives $D_j\Phi = 0$ with respect to the t variables w_{Q_j} and the two restrictions:

$$\begin{array}{lcccl} D_1\Phi & = & p_1 \ln d (d)^{w_{Q_1}} P(Q_1) + \lambda_1 f_1 & = & 0 \\ D_2\Phi & = & p_2 \ln d (d)^{w_{Q_2}} P(Q_2) + \lambda_1 f_2 & = & 0 \\ \vdots & & & & \\ D_t\Phi & = & p_t \ln d (d)^{w_{Q_t}} P(Q_t) + \lambda_1 f_t & = & 0 \\ & & \sum_{j=1}^{t} f_j w_{Q_j} - C & = & 0 \\ & & \sum_{j=1}^{t} P(Q_j) - 1 & = & 0 \end{array}$$

The first t equations yield:

$$w_{Q_j} = \log_d \frac{-\lambda_1 f_j}{p_j \ln d P(Q_j)} = \log_d(-\lambda_1) - \log_d(\ln d) + \log_d \frac{f_j}{p_j \cdot P(Q_j)} \tag{2}$$

Substituting w_{Q_j} in the first restriction, we can obtain the value of $\log_d(-\lambda_1)$:

$$\begin{aligned}
\sum_{j=1}^{t} f_j w_{Q_j} - C &= 0 \\
\sum_{j=1}^{t} f_j \log_d(-\lambda_1) - f_j \log_d(\ln d) + f_j \log_d \frac{f_j}{p_j P(Q_j)} - C &= 0 \\
\log_d(-\lambda_1) &= \frac{C}{\sum_{j=1}^{t} f_j} + \frac{\sum_{j=1}^{t} f_j \log_d(\ln d)}{\sum_{j=1}^{t} f_j} - \\
&\quad \frac{\sum_{j=1}^{t} f_j \log_d \frac{f_j}{p_j P(Q_j)}}{\sum_{j=1}^{t} f_j}
\end{aligned} \tag{3}$$

Combining Equation (2) and (3), the proof can be obtained:

$$w_{Q_j} = \frac{C}{\sum_{i=1}^{t} f_i} + \sum_{i=1}^{t} \left(\frac{f_i}{\sum_{k=1}^{t} f_k} \log_d \frac{p_i P(Q_i)}{f_i} \right) + \log_d \frac{f_j}{p_j P(Q_j)} \tag{4}$$

$\square$

Note that the second restriction (or λ_2) is not used in the proof because it does not contain any variable w_{Q_j}. Since p_j and f_j depend on the occurrence frequency, the formula accounts for both occurrence and query frequencies. The frequencies are usually obtained from real databases or estimated from empirical laws. In particular, the Zipf's law can be used to estimate the occurrence frequencies.

Theorem 1 not only develops a weight assignment that accounts for both uniform and nonuniform occurrence and query frequencies but also proves that it is an optimal solution. In addition, the term weights are computed from the occurrence and query frequencies rather than lookup from a static table, and the computation is efficient since the first and second terms in the formula can be pre-computed. Therefore, the weight assignment satisfies the three criteria described at the beginning of this section.

Simulation. The false drop probability of the optimal weight assignment is compared with the three methods below:

1. *Linear Weight*: All of the distinct terms are listed in descending order according to their occurrence frequencies, and are equally partitioned into groups. The group containing terms with highest occurrence frequencies is assigned with the largest weight, the group with the second highest occurrence frequencies is assigned with the second largest weight, and so on. The term weights are chosen to generate the desire signature weight.

2. *Uniform Weight*: Each term is assigned with the same weight. This strategy typically has been assumed in previous work on superimposed coding.

3. *Inverse Linear Weight*: This strategy is the same as the linear weight assignment, except that the least frequent group is assigned with the largest weight while the most frequent group is assigned with the smallest weight.

In the comparison, the value of $\sum_{j=1}^{V} f_j w_{Q_j}$ is set to the same value for all four methods. Also, the query frequencies are assumed uniform in the simulation. The signature length m is 1024 and the block signature weight is set to 312, 412, 512 612, and 712. Figure 2 shows the results of the simulation. It is clear from the figure that the linear weight assignment method is the worst and that our optimal method is almost an order of magnitude better than the uniform weight assignment method.

Consider Equation (4) again. Intuitively, the first term, $\frac{C}{\sum_{j=1}^{t} f_j}$ can be considered as the *weighted* average term weight and is constant for every term. A frequent term, Q_j, will have a large number of blocks containing it (i.e., small p_j and large f_j). When $P(Q_j)$ is uniform for all j, the last term in (4) yields negative values for frequent terms and positive values for infrequent terms. Therefore, the optimal weight assignment is an inverse function of the occurrence frequency. However, since the last term is a logarithm term, the inverse function is not linear and should be closer to the uniform function than the inverse linear function. When the query frequencies are nonuniform, the optimal weight assignment would deviate further towards the inverse linear function when frequent terms are queried less frequently, and move towards the uniform function when frequent terms are queried more frequently.

3.2 Storage Overhead

The primary result of Theorem 1 is that the optimal weight assignment is the best in terms of false drop probability. However, false drop probabilities cannot be compared without

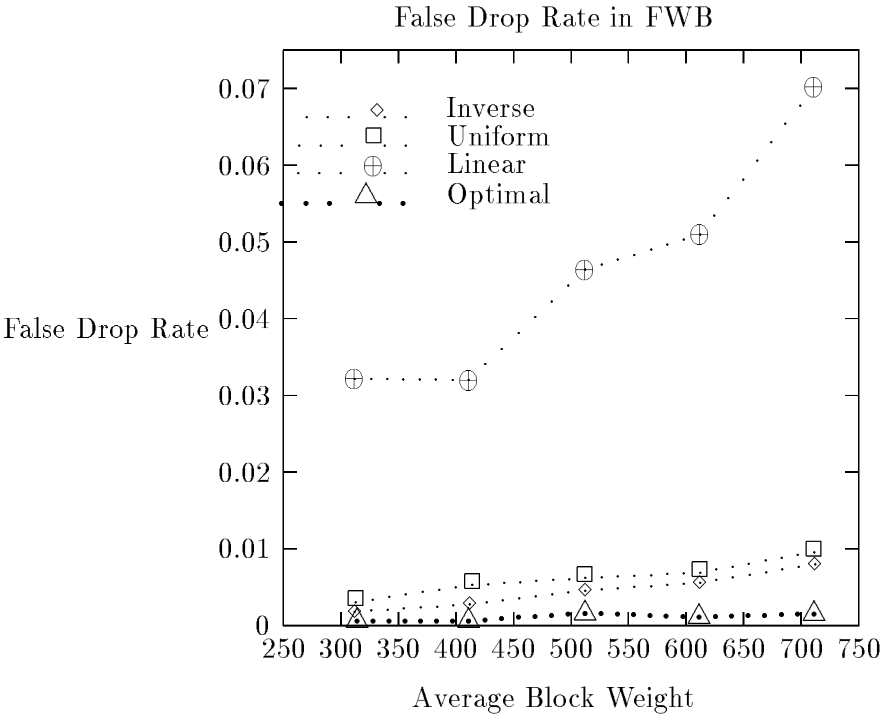

Figure 2: False drop probability of the FWB approach.

reference to the storage overhead. Therefore, it is necessary to study the storage overhead of this new method. In this section, we evaluate the storage overheads of the three different weight assignments: (1) high occurrence terms are assigned with large weights, (2) uniform weight assignment, and (3) low occurrence terms are assigned with large weights. These three methods are similar to those in Section 3.1, but the first and third methods are not necessarily linear.

Before introducing the theorem, some important notations are reviewed.

- w_{B_j} is the weight (number of 1's) of block signature S_j which is superimposed from the term signatures in the block.
- $w_{B_{avg}}$ is defined as the average weight of all of the block signatures in the database.
- $w^+_{B_j} = \sum_{i=1}^{T} w_{i,j}$, where $w_{i,j}$ is the weight of the ith term in the block B_j, and T stands for the block size (number of distinct terms).
- $w^+_{B_{avg}}$ represents the average value of $w^+_{B_j}$ for all of the B_j.

For the example in Figure 1, the weight of block B_i, which contains two terms *free* and *text*, w_{B_i} is 7 while the value of $w^+_{B_i}$ is 4+4=8. Consequently, we need to study the relationship of two quantities, w_{B_j} and $w^+_{B_j}$, to explore the distribution of 1's in each block. The problem can be stated in the following lemma.

Lemma 1 *Given a block B_j, the ratio of $w^+_{B_j}$ to w_{B_j}, denoted by μ, depends only on w_{B_j}, when the following conditions are true:*

1. *bit positions to be set are chosen randomly;*
2. *the length of the block signature is fixed, namely, m;*
3. *the size of the text block B_j is T, which is the same as the number of distinct words in the block; and*
4. *$w_{i,j}$, which represents the ith term's weight in block B_j, is given for all $i \in [1, T]$;*

Proof: It is obvious that $1 - w_{i,j}/m$ is the probability for a bit to be '0' in term i's signature. Thus,

$$\begin{aligned} & 1 - [(1 - \frac{w_{1,j}}{m})(1 - \frac{w_{2,j}}{m}) \cdots (1 - \frac{w_{T,j}}{m})] \\ \approx \; & 1 - (1 - \frac{1}{m})^{w_{1,j} + w_{2,j} + \cdots + w_{T,j}} \end{aligned}$$

represents the probability of a bit to be '1' within the *block* signature as long as $m \gg w_{i,j}$. Consequently, the expected weight for the block signature can be formulated as:

$$\begin{aligned} w_{B_j} &= m \left[1 - (1 - \frac{1}{m})^{w_{1,j} + w_{2,j} + \cdots + w_{T,j}} \right] \\ \frac{w_{B_j}}{m} &= 1 - (1 - \frac{1}{m})^{w^+_{B_j}} \end{aligned}$$

Finally, we may simplify the previous equation into:

$$\begin{aligned} w^+_{B_j} &= \frac{\ln(1 - w_{B_j}/m)}{\ln(1 - 1/m)} \\ \frac{w^+_{B_j}}{w_{B_j}} &= \frac{\ln(1 - w_{B_j}/m)}{w_{B_j} \ln(1 - 1/m)} \\ &= \frac{\ln(1 - w_{B_j}/m)}{\ln(1 - 1/m)^{w_{B_j}}} \\ &= \mu \end{aligned}$$

Therefore, the ratio μ is a constant as long as the block signature weight and the length of the signature are fixed (i.e., FWB approach). More precisely, the ratio μ can be formulated as a function of w_{B_j}, denoted by $\mu(w_{B_j})$. In particular, this ratio is independent of the number of unique terms in a block, and the ratio approaches 1 when the length of a signature is much greater than the expected weight of the block signature. $\square$

The main purpose for this lemma is to show that each FWB block signature contains the same number of 1's (the same $w^+_{B_j}$ value), since w_{B_j} is pre-defined and fixed for each block. That is, the value of $w^+_{B_j}$ is a function of w_{B_j}; it is fixed as long as w_{B_j} is fixed. Therefore, the storage overhead can be evaluated based on this condition.

Theorem 2 *In the FWB method, each text block signature has the same weight. Let* $wt_i^{(1)}$ *and* $wt_i^{(2)}$ *be two weight assignments for a term* i *where* $1 \leq i \leq V$. *Then the storage overhead for both weight assignments will be the same if and only if the following equation holds:*

$$\sum_{i=1}^{V} f_i wt_i^{(1)} = \sum_{i=1}^{V} f_i wt_i^{(2)}$$

Proof: During the proof of this theorem, an important assumption that each physical block contains a term at most once must be observed (alternatively, multiple occurrences of the same term are counted as one occurrence). Consequently, the term frequency f_i can be defined as the number of blocks in which the term appears:

$$f_i = |\{k \mid t_i \in B_k, \forall k = 1, 2, \ldots\}|$$

Then the number of ones that a term contributes to the database will be the product of term frequency f_i and its weight. Hence, the total number of ones in the database will be $\sum_{i=1}^{V} f_i wt_i$.

According to Lemma 1, each block B_i requires the same number of individual ones, $w^+_{B_i}$, to yield the specific weight w_{B_i} for all i. For the FWB approach, every $w_{B_i} = w_{B_{avg}}$ and, hence, $w^+_{B_i} = w^+_{B_{avg}}$. Therefore, the number of blocks can be obtained by dividing the total number of ones in the database by $w^+_{B_{avg}}$. That is,

$$\text{Number of FWB blocks} = \frac{\sum_{i=1}^{V} f_i wt_i}{w^+_{B_{avg}}} \tag{5}$$

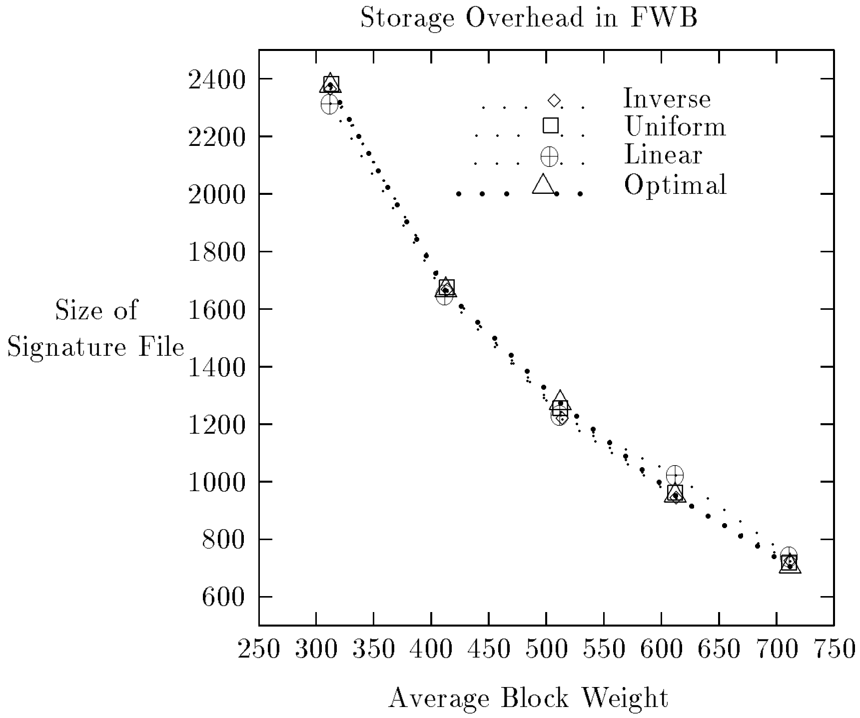

Figure 3: Storage overhead of the FWB approach.

Then the storage overhead in the FWB method of two different weight assignments will be the same, if $\sum_{i=1}^{V} f_i wt_i^{(1)} = \sum_{i=1}^{V} f_i wt_i^{(2)}$. $\square$

Figure 3 illustrates the storage overheads of the four methods in our simulation, obtained with the same set of parameters as in Figure 2. The simulation results demonstrate that the optimal weight assignment has much better performance in false drop probability than the other three strategies when their storage overheads in the signature file are all the same.

4 Performance of the FWB and the FSB Methods

In this section, we analyze the FWB and FSB approaches in terms of their false drop probabilities and storage overheads. We will demonstrate that the optimal weight assignment developed for the FWB environment still maintains the lowest false drop probability compared to other weight assignment methods in the FSB approach. The main idea of the FWB approach is to fix the weight of each block signature instead of fixing the number of unique terms in a block as in the FSB method. The rationale behind this is that the weight of each block may vary when it is superimposed from a fixed number of different term signatures, especially when the term signatures have non-uniform weights. A widely dispersed distribution of the block signature weight results in a higher false drop probability but lower storage overhead than a narrow distribution.

We first introduce two lemmas to aid in the proof of our main theorem in this section.

Lemma 2 *If $a_1 \geq a_2 \geq \cdots \geq a_p$ and $b_1 \geq b_2 \geq \cdots \geq b_p$, then*

$$\left(\sum_{k=1}^{p} a_k\right) \left(\sum_{k=1}^{p} b_k\right) \leq p \sum_{k=1}^{p} a_k b_k$$

Proof: Since $a_j \geq a_k$ and $b_j \geq b_k$ for every $j \leq k$, then it is obvious that

$$\sum_{1 \leq j \leq k \leq p} (a_k - a_j)(b_k - b_j) \geq 0$$

After expanding and rearranging the left-hand side of the inequality, we obtain:

$$\sum_{k=1}^{p} \sum_{j=1}^{k} a_j b_j + \sum_{k=1}^{p} \sum_{j=1}^{k} a_k b_k \geq \left(\sum_{k=1}^{p} a_k\right) \left(\sum_{k=1}^{p} b_k\right) + \sum_{k=1}^{p} a_k b_k$$

In addition, the term $\sum_{k=1}^{p} \sum_{j=1}^{k} a_j b_j$ equals to $\sum_{k=1}^{p} (p-k+1) a_k b_k$ when j is replaced with k. Hence, the inequality can be expressed as:

$$\sum_{k=1}^{p} (p-k+1) a_k b_k + \sum_{k=1}^{p} k a_k b_k \geq \left(\sum_{k=1}^{p} a_k\right) \left(\sum_{k=1}^{p} b_k\right) + \sum_{k=1}^{p} a_k b_k$$

and after simplification:

$$p \sum_{k=1}^{p} a_k b_k \geq \left(\sum_{k=1}^{p} a_k\right) \left(\sum_{k=1}^{p} b_k\right)$$

□

The next lemma introduces another inequality that also involves two monotone sequences of integers.

Lemma 3 *Let $a_1, a_2, \cdots, a_p$ be positive integers and $a_1 \geq a_2 \geq \cdots \geq a_p$. In addition, if $b_1, b_2, \cdots, b_p$ are also integers with $b_1 \leq b_2 \leq \cdots \leq b_p$ and $\sum_{i=1}^{p} b_i = 0$, then $\sum_{i=1}^{p} a_i b_i \leq 0$.*

Proof: Two cases should be considered:

case 1: If $b_1 = b_2 = b_3 = ... = b_p = 0$, then it is obvious that $\sum_{i=0}^{p} a_i b_i \leq 0$.

case 2: If $b_1 \leq b_2 \leq \cdots \leq b_p$ and $\sum_{i=1}^{p} b_i = 0$, then there exists an integer $j \in [1, p)$ such that $b_1 \leq b_2 \leq \cdots \leq b_j < 0 \leq b_{j+1} \leq b_{j+2} \leq \cdots \leq b_p$ and $\sum_{i=j+1}^{p} b_i = -\sum_{i=1}^{j} b_i$. Hence, a new inequality can be obtained by introducing a decreasing sequence a_i into the latter equality,

$$\sum_{i=j+1}^{p} b_i a_i \leq \sum_{i=j+1}^{p} b_i a_{j+1} \leq -\sum_{i=1}^{j} b_i a_j \leq -\sum_{i=1}^{j} b_i a_i$$

since each a_i is positive and $a_i \geq a_j \geq a_{j+1} \geq a_k$ for every $i \in [1, j], k \in [j+1, p]$. Then it follows that:

$$\sum_{i=1}^{p} a_i b_i \leq 0$$

$\square$

Theorem 3 *The false drop probability for the FWB method is smaller than that of the FSB method when the average block signature weights of the signatures in the two approaches are the same.*

Proof: Consider a set of blocks $\{B_i | 1 \leq i \leq p\}$ in the FWB method and a set of blocks $\{\hat{B}_i | 1 \leq i \leq p\}$ in the FSB method. Given a query Q not contained in any B_i or $\hat{B}_i$, then the average number of false drops for FWB is the summation of the probabilities P_i, where P_i is the probability that a query signature S_Q is qualified for block signature S_{B_i}. That is, the average number of false drops can be expressed as:

$$\sum_{i=1}^{p} P_i = \sum_{i=1}^{p} \frac{\binom{w_{B_i}}{w_Q}}{\binom{m}{w_Q}}$$

and this is the same for the FSB approach.

However, each block in FWB has the same weight i.e., $w_{B_1} = w_{B_2} = \cdots = w_{B_p} = w_{B_{avg}}$, then the theorem can be proved by proving the following inequality:

$$p \frac{\binom{w_{B_{avg}}}{w_Q}}{\binom{m}{w_Q}} \leq \sum_{i=1}^{p} \frac{\binom{w_{\hat{B}_i}}{w_Q}}{\binom{m}{w_Q}} \tag{6}$$

where $w_{B_{avg}} = \sum_{i=1}^{p} w_{B_i}/p$ and $w_{\hat{B}_{avg}} = \sum_{i=1}^{p} w_{\hat{B}_i}/p$ are the average block weights for FWB and FSB approaches, respectively. Since m and w_Q are fixed, (6), which we try to prove, can be simplified as:

$$p\binom{w_{B_{avg}}}{w_Q} \leq \sum_{i=1}^{p} \binom{w_{\hat{B}_i}}{w_Q}$$

or

$$pw_{B_{avg}}(w_{B_{avg}} - 1)\cdots(w_{B_{avg}} - w_Q + 1) \leq \sum_{i=1}^{p} w_{\hat{B}_i}(w_{\hat{B}_i} - 1)\cdots(w_{\hat{B}_i} - w_Q + 1) \tag{7}$$

(7) can be proved by induction. Considered $w_Q = 2$, the proof makes use of Lemma 2 and the assumption that the average weights of the signatures in the two methods are the same (i.e. $pw_{B_{avg}} = pw_{\hat{B}_{avg}} = \sum_{i=1}^{p} w_{\hat{B}_i}$). Without loss of generality, it is assumed that $w_{\hat{B}_1} \geq w_{\hat{B}_2} \geq \cdots w_{\hat{B}_p}$ to satisfy the condition in Lemma 2. Hence, by Lemma 2,

$$\left(\sum_{i=1}^{p} w_{\hat{B}_i}\right)\left(\sum_{i=1}^{p} w_{\hat{B}_i}\right) \leq p\sum_{i=1}^{p} w_{\hat{B}_i} w_{\hat{B}_i}$$

and then we may obtain a new inequality when using $pw_{B_{avg}} = \sum_{i=1}^{p} w_{\hat{B}_i}$.

$$\begin{aligned}
(pw_{B_{avg}})(pw_{B_{avg}}) &\leq p\sum_{i=1}^{p} w_{\hat{B}_i}^2 \\
pw_{B_{avg}}^2 - pw_{B_{avg}} &\leq \sum_{i=1}^{p} w_{\hat{B}_i}^2 - \sum_{i=1}^{p} w_{\hat{B}_i} \\
pw_{B_{avg}}(w_{B_{avg}} - 1) &\leq \sum_{i=1}^{p} w_{\hat{B}_i}(w_{\hat{B}_i} - 1)
\end{aligned} \tag{8}$$

Thus, (7) is valid for $w_Q = 2$. Next, if $w_Q = j$ is valid, that is,

$$pw_{B_{avg}}(w_{B_{avg}} - 1)\cdots(w_{B_{avg}} - j + 1) \leq \sum_{i=1}^{p} w_{\hat{B}_i}(w_{\hat{B}_i} - 1)\cdots(w_{\hat{B}_i} - j + 1)$$

then

$$\begin{aligned}
& pw_{B_{avg}}(w_{B_{avg}} - 1)\cdots(w_{B_{avg}} - j + 1)(w_{B_{avg}} - j) \\
\leq\ & \left[\sum_{i=1}^{p} w_{\hat{B}_i}(w_{\hat{B}_i} - 1)\cdots(w_{\hat{B}_i} - j + 1)\right](w_{B_{avg}} - j) \\
=\ & \sum_{i=1}^{p} w_{\hat{B}_i}(w_{\hat{B}_i} - 1)\cdots(w_{\hat{B}_i} - j) \\
& + \sum_{i=1}^{p} w_{\hat{B}_i}(w_{\hat{B}_i} - 1)\cdots(w_{\hat{B}_i} - j + 1)(w_{B_{avg}} - w_{\hat{B}_i})
\end{aligned} \tag{9}$$

According to our assumption, $w_{\hat{B}_1} \geq \cdots \geq w_{\hat{B}_p}$. Therefore,

$$(w_{B_{avg}} - w_{\hat{B}_1}) \leq \cdots \leq (w_{B_{avg}} - w_{\hat{B}_p})$$

and $\sum_{i=1}^{p}(w_{B_{avg}} - w_{\hat{B}_i}) = 0$. Using a sequence of decreasing positive integers $\{w_{\hat{B}_i}(w_{\hat{B}_i} - j + 1) |$ for every i from 1 to p $\}$ as the sequence $\{a_i\}$ in Lemma 3, we may obtain another inequality by Lemma 3

$$\sum_{i=1}^{p} w_{\hat{B}_i}(w_{\hat{B}_i} - 1)\cdots(w_{\hat{B}_i} - j + 1)(w_{B_{avg}} - w_{\hat{B}_i}) \leq 0 \tag{10}$$

From Equation (9) and (10), we can verify that it is also valid when $w_Q = j + 1$:

$$p w_{B_{avg}}(w_{B_{avg}} - 1)\cdots(w_{B_{avg}} - j + 1) \leq \sum_{i=1}^{p} w_{\hat{B}_i}(w_{\hat{B}_i} - 1)\cdots(w_{\hat{B}_i} - j + 1)$$

By the principle of mathematical induction, the theorem is proved. □

The primary result of Theorem 3 is that the traditional FSB technique is worse than the FWB method in terms of false drop probability. Furthermore, Theorem 3 can be generalized to show that a widely dispersed distribution in block signature weights results in a higher false drop probability.

Next, we consider the storage overhead of the two methods. We have observed that the ratio $\mu = \frac{w^+_{B_j}}{w_{B_j}}$ in Lemma 1 can be treated as a function of the block signature weight w_{B_j} when the signature length m is a constant. Hence, the value $w^+_{B_j}$ can also be defined as a function of w_{B_j}.

$$w^+_{B_j} = \frac{\ln(1 - w_{B_j}/m)}{\ln(1 - 1/m)} \tag{11}$$

In order to know how the value $w^+_{B_j}$ is growing, the first and second derivatives of the function $w^+_{B_j}$ with respect to w_{B_j} are obtained as follows.

$$\frac{d\ w^+_{B_j}}{d\ w_{B_j}} = \frac{-1}{\ln(1 - 1/m)} \frac{1}{m - w_{B_j}} \tag{12}$$

$$\frac{d^2\ w^+_{B_j}}{d\ (w_{B_j})^2} = \frac{-1}{\ln(1 - 1/m)} \frac{1}{(m - w_{B_j})^2} \tag{13}$$

Both derivatives are positive, when $w_{B_j} \in [0, m)$, since the value of $-1/\ln(1 - 1/m)$ is always positive with a positive m. Then the value $w^+_{B_j}$ is not only an increasing function but also a function growing more rapidly than any linear function. It means that, for instance, the total term weight added to a block in order to increase the block signature weight from 200 to 300 is smaller than what is necessary to increase the block signature weight from weight 700 to 800. Therefore, the following theorem can be easily proved.

Theorem 4 *The storage overhead for the FWB method is larger than that of the FSB method when the average block signature weights of the signatures in the two approaches are the same.*

Proof: The main idea of this proof is to try to verify that, given two sets of the same size, representing FWB and FSB blocks respectively, the total number of ones superimposed in the blocks of the FWB set is less than that of the FSB set (i.e., blocks in FSB set will absorb more ones than those in FWB). Therefore, for a given database, it is obviously that the storage overhead of FWB will be greater than that of FSB.

Let $\{B_i | 1 \leq i \leq p\}$ and $\{\hat{B}_i | 1 \leq i \leq p\}$ be two sets of FWB blocks and FSB blocks respectively. From Equation (5), it can be generalized to satisfy both approaches.

$$\text{Number of FWB blocks} = \frac{\sum_{i=1}^{V} wt_i f_i}{w^+_{B_{avg}}} \tag{14}$$

$$\text{Number of FSB blocks} = \frac{\sum_{i=1}^{V} wt_i f_i}{w^+_{\hat{B}_{avg}}} \tag{15}$$

Hence the storage overhead in Equation (14) is larger than that in Equation (15), if and only if $w^+_{B_{avg}}$ is less than $w^+_{\hat{B}_{avg}}$. Alternatively,

$$\sum_{i=1}^{p} w^+_{B_i} \leq \sum_{i=1}^{p} w^+_{\hat{B}_i} \tag{16}$$

According to the definitions of the two approaches and the assumption in this theorem, we observe that

$$\frac{\sum_{i=1}^{p} w_{\hat{B}_i}}{p} = \frac{\sum_{i=1}^{p} w_{B_i}}{p} = w_{B_i} \quad \forall i \in [1, p].$$

Without loss of generality, the sequence of blocks in the FSB set can be ordered into $w_{\hat{B}_1} \leq w_{\hat{B}_2} \leq \cdots \leq w_{\hat{B}_p}$, and then there exists an index $j \in [1, p]$ such that

$$w_{\hat{B}_1} \leq \cdots \leq w_{\hat{B}_j} \leq w_{B_{avg}} \leq w_{\hat{B}_{j+1}} \leq \cdots \leq w_{\hat{B}_p}$$

and

$$\sum_{i=1}^{j} w_{B_{avg}} - w_{\hat{B}_i} = \sum_{i=j+1}^{p} w_{\hat{B}_i} - w_{B_{avg}} \quad \text{where } w_{B_{avg}} = w_{B_i} \forall i \in [1, p]. \tag{17}$$

Then Equation (16) and, hence, the theorem can be proved when the inequality expressed in Equation (18) is valid.

$$\sum_{i=1}^{j} (w^{+}_{B_{avg}} - w^{+}_{\hat{B}_i}) \leq \sum_{i=j+1}^{p} (w^{+}_{\hat{B}_i} - w^{+}_{B_{avg}}) \tag{18}$$

The detail proof for Equation (18) will be described in Appendix A. □

Simulation results of both theorems are illustrated in Figure 4 and 5. The text generated for the simulations follows the Zipf's Law. The database contains 90,000 terms, and the number of distinct terms V is 10,000. Two weight assignments were used, namely, the inverse linear and the uniform assignments, to generate two distinct distributions of the block signature weights for the FSB and FWB approaches; the optimal assignment method cannot be used since an optimal assignment is not available for the FSB method. Figure 6 compares the distributions of the block signature weights in the FSB and FWB methods when the two weight assignments are used.

According to Figure 4(a) and 5(a), FWB is 13% to 45% better than FSB in false drop probability while incurring only 6% to 8% more storage overhead when the inverse linear weight assignment method is employed. For the uniform weight assignment, however, both the false drop probabilities and the storage overheads are the same in both method, since the distributions of the block signature weights in the two methods are about the same. It can be concluded from the analytical and simulation results that the FWB approach is at least as good as the FSB method in false drop probability at the expense of a slightly higher storage overhead. The difference depends on how flat the distribution in FSB is. It must be pointed out that for a large block size and a "well-behaved" distribution (such as a Zipf distribution), the signatures generated from the FSB and FWB methods are practically identical under the uniform weight assignment. The FWB method excels when there are "ill-behaved" text blocks and under non-uniform weight assignment strategies. Since the optimal weight assignment has been shown to be better than the inverse linear method, its performance gain over the FSB method should be even better.

Figure 4: False drop probabilities of FSB and FWB: (a) Inverse proportional; (b) Uniform weight assignments.

Figure 5: Storage overheads of FSB and FWB: (a) Inverse proportional; (b) Uniform weight assignments.

Figure 6: Distribution of block signature weights in FSB and FWB: (a) Inverse proportional; (b) Uniform weight assignments.

5 Conclusion

The objective of this paper is to obtain the optimal weight assignment for a term as a function of the occurrence and query frequencies of the term. Once the weight is obtained, the term signature can be generated by setting the specified number of bits randomly. Thus, the optimal signature can be generated algorithmically. This is in contrast with previous methods where a lookup table is used to map a word directly to the word signature [7]. Further, unlike the traditional blocking scheme, the fixed size block (FSB) method, which includes a fixed number of distinct terms in each block, we propose a new way, the fixed weight block (FWB) method, to force each block signature to have a constant weight. The FWB method provides two important advantages over the FSB approach. First, the FSB method, with a constant block signature weight, simplifies the process of obtaining the optimal weight assignment, since the block signature weight is directly involved in the expression for determining the false drop probability. This allows us to obtain the optimal weight assignment without making unrealistic assumptions and, more importantly, the expression for computing the term signature weight is simple and efficient. The second advantage is that the FWB approach has a lower false drop probability than the traditional FSB method due to its constant block weight. This eliminates the situation where some signatures are generated with abnormally high weights, which is possible (though with a small probability) in the FSB method.

From the simulation results, the false drop probability of the optimal weight assignment is close to 10 times better than that of the commonly used uniform weight assignment. The comparison is done at the same storage overhead and under the FWB method. When comparing the FWB method with the FSB method, it has been shown that FWB has a better false drop probability than that of the FSB method, but the storage overhead is slightly higher. The differences depend on the degree of dispersion in the distribution of the signature weights.

From an implementation point of view, the optimal solution can be used directly for computing the term weights. Statistics about the document terms and query terms can be kept in a hash table, which can be updated periodically, if necessary, to reflect updates to the document base and changes in usage pattern. Given the size of main memory typically available on computers, the hash table can easily fit into main memory. Thus, the terms weights can be computed efficiently. This is impossible if a lookup table is used to map terms into word signatures, since the table would be too large. Of course, the system designer also has the option of further simplifying the formula (e.g., set query frequencies to a constant value) or using an expression (e.g., Zipf distribution) to approximate the occurrence frequencies. However, the optimal solution remains the best, and feasible, choice if minimizing false drops is the primary concern.

The combination of the new blocking scheme and the optimal weight assignment definitely outperforms the traditional method in terms of false drop probability and storage overhead. The technique developed here can work with other signature file structures, such as the multi-level signature file [11,16] and the partition signature file [12]. Since many of these search techniques have a better search performance when more 1's are specified in the query signature (see, for example, Lee and Leng [12]), our optimal weight assignment, which assigns higher weight to frequently queried terms, not only reduces the false drop probability, but also improves the search performance of these search methods.

Acknowledgment

The authors would like to thank the Applied Information Technology Research Center, a Thomas Edison program funded by the State of Ohio, for their financial support, and to the anonymous referees whose comments have helped us to improve the contents and presentation of this paper.

Appendix I

We have shown in Equation (11) that the value of $w^+_{B_i}$ is a function of w_{B_i}. Additionally, in Equation (12) and (13), both the first and the second derivatives of $w^+_{B_i}$ with respect to w_{B_i} are positive when $w_{B_i} \in [0, m)$. In order to make $w^+_{B_i}$ more expressive, a function f is defined as:

$$f(w_{B_i}) = w^+_{B_i} = \frac{\ln(1 - w_{B_j}/m)}{\ln(1 - 1/m)}$$

Then the inequality expression in Equation (18), which Theorem 4 attempts to prove, becomes:

$$\sum_{i=1}^{j}[f(w_{B_{avg}}) - f(w_{\hat{B}_i})] \leq \sum_{i=j+1}^{p}[f(w_{\hat{B}_i}) - f(w_{B_{avg}})] \qquad (19)$$

when Equation (17) is valid. For every $i \in [1, j]$,

$$f(w_{B_{avg}}) - f(w_{\hat{B}_i}) = \sum_{k=0}^{w_{B_{avg}} - w_{\hat{B}_i} - 1} f(w_{B_{avg}} - k) - f(w_{B_{avg}} - k - 1)$$

$$= \sum_{k=0}^{w_{B_{avg}} - w_{\hat{B}_i} - 1} \frac{f(w_{B_{avg}} - k) - f(w_{B_{avg}} - k - 1)}{(w_{B_{avg}} - k) - (w_{B_{avg}} - k - 1)}$$

Then, according to the Mean-Value Theorem, we can translate the above summation into

$$\sum_{k=0}^{w_{B_{avg}} - w_{\hat{B}_i} - 1} f^{'}(c_k)$$

where $c_k \in [w_{B_{avg}} - k - 1, w_{B_{avg}} - k] \; \forall \; k$. Because $f^{''}$ is positive within $[0, m)$, $f^{'}$ must be increasing with respect to the value $w_{\hat{B}_i}$. Therefore,

$$\sum_{k=0}^{w_{B_{avg}} - w_{\hat{B}_i} - 1} f^{'}(c_k) \leq (w_{B_{avg}} - w_{\hat{B}_i}) f^{'}(w_{B_{avg}})$$

or

$$f(w_{B_{avg}}) - f(w_{\hat{B}_i}) \leq (w_{B_{avg}} - w_{\hat{B}_i}) f^{'}(w_{B_{avg}})$$

is satisfied for every $i \in [1, j]$. That is, we can reach the inequality:

$$\sum_{i=1}^{j} f(w_{B_{avg}}) - f(w_{\hat{B}_i}) \leq \sum_{i=1}^{j} (w_{B_{avg}} - w_{\hat{B}_i}) f^{'}(w_{B_{avg}}). \tag{20}$$

By the same arguments, we may obtain another similar inequality:

$$\sum_{i=j+1}^{p} (w_{\hat{B}_i} - w_{B_{avg}}) f^{'}(w_{B_{avg}}) \leq \sum_{i=j+1}^{p} f(w_{\hat{B}_i}) - f(w_{B_{avg}}) \tag{21}$$

Combining Equations (17), (20) and (21), we can verify Equation (19):

$$\sum_{i=1}^{j} [f(w_{B_{avg}}) - f(w_{\hat{B}_i})] \leq \sum_{i=j+1}^{p} [f(w_{\hat{B}_i}) - f(w_{B_{avg}})]$$

and also Equation (18):

$$\sum_{i=1}^{j} [w^{+}_{B_{avg}} - w^{+}_{\hat{B}_i}] \leq \sum_{i=j+1}^{p} [w^{+}_{\hat{B}_i} - w^{+}_{B_{avg}}]$$

REFERENCES

[1] T.M. Apostol, *Mathematical Analysis*. Addison-Wesley Publishing Company, Inc., Reading, MA, 1975.

[2] W.W. Chang, and H.J. Schek, "A signature access method for the Starburst database system," *Proceedings of the Fifteenth International Conference on Very Large Data Bases*, Amsterdam, The Netherlands, Aug. 1989, 145–153.

[3] U. Deppisch, "S-Tree: A dynamic balanced signature index for office retrieval," *Proceedings of the ACM Conference on Research and Development in Information Retrieval*, Pisa, Italy, Sept. 1986, 77–87.

[4] C. Faloutsos, "Signature-based text retrieval methods: A survey," *Data Engineering Bulletin*, Vol. 13, No. 1, Mar. 1990, 25–32.

[5] C. Faloutsos, and S. Christodoulakis, "Signature files: an access method for documents and its analytical performance evaluation," *ACM Transactions on Office Information Systems*, Vol. 2, No. 4, Oct. 1984, 267–288.

[6] C. Faloutsos, and S. Christodoulakis, "Description and performance analysis of signature file methods for office filing," *ACM Transactions on Office Information Systems*, Vol. 5, No. 3, July 1987, 237–257.

[7] C. Faloutsos, and S. Christodoulakis, "Optimal signature extraction and information loss," *ACM Transactions on Database Systems*, Vol. 12, No. 3, Sept. 1987, 395–428.

[8] C. Faloutsos, and S. Christodulakis, "Design of a signature file method that accounts for non-uniform occurrence and query frequencies," *Proceedings of the Eleventh International Conference on Very Large Data Bases*, Stockholm, Sweden, Aug. 1985, 165–170 .

[9] A. Kent, K. Ramamohanarao, and R. Sacks-Davis, "A signature file scheme based on multiple organizations for indexing very large databases," *Journal of American Society of Information Science*, Vol. 41, No. 7, 1990, 508–534.

[10] D.L. Lee, "A word-parallel, bit-serial signature processor for superimposed coding," *Proceedings of the Second International Conference on Data Engineering*, Los Angeles, Feb. 1986, 352–359.

[11] D.L. Lee, Y.M. Kim, and G. Patel, "Efficient signature file methods for text retrieval.," *To appear in IEEE Transactions on Data and Knowledge Engineering*, 1994.

[12] D.L. Lee, and C.-W. Leng, "Partitioned signature files: Design issues and performance evaluation," *ACM Transactions on Information Systems*, Vol. 7, No. 2, Apr. 1989, 158–180.

[13] D.L. Lee, and C.-W. Leng, "A partitioned signature file structure for multiattribute and text retrieval," *Proceedings of the Sixth International Conference on Data Engineering*, Los Angeles, Feb. 1990, 389–397.

[14] J.L. Pfaltz, W.J. Berman, and E.M. Cagley, "Partial match retrieval using indexed descriptor files," *Communications of ACM*, Vol. 23, No. 9, Sept. 1980, 522–528.

[15] C.S. Roberts, "Partial-match retrieval via method of superimposed codes," *Proc. IEEE*, Vol. 67, No. 12, Dec. 1979, 1624–1642.

[16] R. Sacks-Davis, A. Kent, and K. Ramamohanarao, "Multikey access methods based on superimposed coding techniques," *ACM Transactions on Database Systems*, Vol. 12, No. 4, Dec. 1987, 655–696.

[17] R. Sacks-Davis, and K. Ramamohanarao, "A two level superimposed coding scheme for partial match retrieval," *Information Systems*, Vol. 8, No. 4, 1983, 273–280.

[18] G. Salton, and M.J. McGill, *Introduction to Modern Information Retrieval.* McGraw-Hill, Inc., New York, N.J., 1983.

[19] C. Stanfill, and B. Kahle, "Parallel free-text search on the connection machine system," *Communications of ACM*, Vol. 29, No. 12, Dec. 1986, 1229–1239.

[20] S. Stiassny, "Mathematical analysis of various superimposed coding methods," *American Documentation*, Vol. 11, No. 2, February 1960, 155–169.

[21] D. Tsichritzis, and S. Christodulakis, "Message files," *ACM Trans. Office Info. Syst.*, Vol. 1, No. 1, January 1983, 88–98.