PERGAMON

Information Processing and Management 37 (2001) 39–51

INFORMATION PROCESSING & MANAGEMENT

www.elsevier.com/locate/infoproman

Document ranking for variable bit-block compression signatures

Robert W.P. Luk*, C.M. Chen

Department of Computing, Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

Received 20 June 1999; accepted 1 March 2000

Abstract

Variable bit-block compression (VBC) signature is extended for document ranking. Two different extensions were experimented: the weighted VBC (WVBC) scheme and the aggregate VBC (AVBC) scheme. For both, analytical bounds of the additional storage for the term frequencies were derived. The upper and lower bounds of WVBC signatures were better than the corresponding bounds for AVBC signatures. In general, these bounds are functions of the word size (in bits) of the term frequencies. Therefore, term frequencies were scaled to reduce the word size. Experiments showed that the additional storage cost is closer to the lower than the upper bound for both WVBC and AVBC signatures. In addition, WVBC signatures were better than AVBC signatures in terms of storage and retrieval speed. Logarithmic scaling was found to be significantly better than linear scaling, in measuring the agreement of document ranking against the case without scaling, using the Kendall rank-order correlation. If a 75% ranking performance is acceptable, then the additional storage of the term frequencies is only 3.4% of all the indexed documents.

Keywords: Information retrieval; Signature file; Compression; Indexing and document ranking

1. Introduction

Searching office documents is becoming an indispensable tool as computers can access the Internet and intranets. For large companies, many office documents are generated every day. Manually searching for previous related documents is unrewarding and time consuming. For

* Corresponding author. Tel.: +852-2766-5143; fax: +852-2774-0842.
E-mail address: csrluk@comp.polyu.edu.hk (R.W.P. Luk).

0306-4573/00/$ - see front matter
PII: S0306-4573(00)00020-0

multi-national companies, office documents are generated in different geographical locations; a central search engine can reduce the manual effort needed to examine documents available from each location. For companies striving for quality, the timely retrieval of the relevant documents is important, since one part of the quality metric is a punctual response to changes and feedback.

Inverted file is the common indexing technique for archival (textual) databases. However, in a dynamic environment, like the office or the Internet, new documents are frequently generated and they have to be added to the index. Typically, adding new documents to the inverted file requires merging the old inverted file with the new information to form a new inverted file. This re-indexing process requires at least double the storage demand of the original inverted file. In addition, the time for re-indexing the inverted file is too large for frequent updates, and the time cost is usually amortized over a period of time. In between updates, the inverted file may be considered out-dated, and this may be undesirable in an office environment or in the Internet environment.

Signatures are the alternative to inverted files when the update cost and keeping a timely index are the main concern. Amongst various signatures, Faloutsos and Christodoulakis have concluded that variable bit-block compression (VBC) signatures are the best for "office use" (Faloutsos & Christodoulakis, 1987) where office documents are generated frequently and their size varies substantially (cf the size of a memo and a technical report).

Compared to signatures using run-length encoding (Golomb, 1966), VBC signatures are searched faster because many unqualified documents can be skipped without decoding. Compared to bit-block compression signatures, VBC signatures adapt their compression performance for each document, incurring less storage cost. This becomes important in an office environment, since the compression performance depends on the document size and the size of office documents varies substantially.

Compared with the more common signatures by superimposed coding (Mooers, 1949), VBC signatures do not have false drop probabilities due to the disjunction of word signatures in the text block. Faloutsos and Christodoulakis (1987) found that the total false drop probability of VBC signatures was only 10% of the false drop probability of superimposed coding signatures. Recent work in VBC signatures (Chan, Wong & Luk, 1998) for multi-lingual indexing has also shown that the time for indexing (about 20 min for 4000 documents) is shorter than superimposed coding (i.e. done overnight). The storage requirement of VBC signatures is smaller than that of superimposed coding signatures (cf 23% and 9–12% of the original text size, respectively). However, superimposed coding signatures can be extended to support document ranking (Croft & Savino, 1998; Wong & Lee, 1990; Lee & Ren, 1996), whereas VBC signatures cannot.

In this paper, we describe our implementation to extend VBC signatures for document ranking. The rest of this paper is organized as follows. In the next section, the problem of supporting document ranking for signatures is discussed. An extension of signatures for document ranking is reviewed. It was argued that the extension is particularly suitable for superimposed coding, instead of VBC signatures. In Section 3, VBC signatures are defined and two different extensions of VBC signatures for document ranking are discussed in detail. The upper and lower bounds of the additional storage for the term frequencies are derived. Compression of term frequencies by scaling is also examined. In Section 4, the two different

extensions of VBC signatures for document ranking are evaluated in terms of storage, retrieval speed and retrieval effectiveness. Finally, we conclude that VBC signatures with document ranking are a viable alternative to weight partitioned (superimposed coding) signatures for document ranking (Table 1).

2. Document ranking for signatures

Ordering retrieved documents according to their relevance to the query is called document ranking. The relevance is usually considered as the similarity (or distance) between the query and the document, which can be measured in various ways (e.g. cosine measure and inner products). These similarity measures are determined by aggregating the term weights of the document. A common term weight $w_{i,j}$ (Salton & Buckley, 1988) is defined as:

$$w_{i,\,j} = f_{i,\,j} \times d_j$$

for the ith document and for the jth term, where $f_{i,j}$ is the jth term occurrence frequency in document i and d_j is the inverse document frequency of term j.

To support document ranking, the inverse document frequency can be stored in the (term) dictionary because it is only a function of term j. However, term frequencies are a function of both the document i and the term j, and there are many more term frequencies to store than

Table 1
Description of symbols used

Symbol descriptions	
$\lvert a_i \rvert$	Aggregate VBC signature size for document i
b	Bit-block size
b_{opt}	Optimal bit-block size
c	Constant which is approximately 1.91
d_j	Inverse document frequency of term j
$\lvert e_i \rvert$	Weighted VBC signature size for document i
$f_{i,j}$	Occurrence frequency of term j in document i
m	Word size (in bits) of the scaled down integer
n_i	Number of terms in document i
$P(1)$	Probability of a position in the dictionary bit-vector is one
R_a	Relative additional storage of aggregate VBC compared with the VBC signatures
R_e	Relative additional storage of weighted VBC compared with the VBC signatures
s	Word size of the (scaled or not scaled) term frequency stored in the signature
$\lvert t \rvert$	Number of unique terms in the dictionary
T	Kendall rank-order correlation
$w_{i,j}$	Term weight of term j in document i
x_i	Number of one-bits in the binary representation of all the non-zero term frequencies in document i

the inverse document frequencies. Therefore, the main problem for signatures to support document ranking is to efficiently encode the term frequencies.

Croft and Savino (1988) examined various approaches to approximately represent term frequencies for signatures. Wong and Lee (1990) used a set of term-frequency (or weight) partitioned files to store the term frequency information. Conceptually, each file contains all the terms with the same frequency. These files are indexed by another signature file for efficient searching. Lee and Ren (1996) found that indexing these signature files for term frequencies should also be optimally configured for each term frequency file to reduce false drop to a minimal. For the asymptotic precision performance, the storage overhead of these signature files compared with the original document was found to be (approximately) 25% after stemming and common word removal.

Superimposed coding was used for the weight-partitioned signature because the relevant text block within a document can be identified instead of the entire document. For VBC signatures, searching would be slower than superimposed coding in this case, because they are searched at the document level instead of the text block level. In addition, the storage cost of VBC signatures would be high for files storing low-term frequencies, because there are much more low than high-frequency terms by Zipf law. If the number of terms in a file is more than or equal to 26% of the vocabulary size (Chan et al., 1998), then VBC signatures have a higher storage cost than without VBC. Therefore, a different approach is needed for VBC signatures to support document ranking.

3. Variable bit-block compression signature

A VBC signature is a compressed dictionary bit-vector. The kth dimension or position in the original bit vector corresponds to the presence or absence of the kth dictionary term in the document. To carry out VBC, the original bit vector is divided into bit blocks (of fixed size b). The bit-block properties define the VBC signature, which is divided into three bit strings or parts: Part I, II and III.

Part I is a bit string where one bit corresponds to one bit-block. If the corresponding bit-block of a bit in Part I has no one-bits, then the corresponding bit is set to zero. Otherwise, that bit is set to one. Conceptually, Part II counts the number of one-bits in those bit-blocks with at least a single one-bit. The number of one-bits is expressed by the number of binary digits in Part II. For instance, if there are n (>0) one-bits in the bit-block, then there are $n-1$ one-bits followed by a zero-bit, which indicates the end of the current bit-block being counted. For Part III, the ceiling of $\log_2 b$ bits is used to define the position (in binary) where the bit value is one. Table 2 shows an example of a VBC signature for a vector of bit-block size four.

3.1. Storing term weights

To support document ranking, term weights have to be stored in the VBC signature. Two schemes are explored here: the weighted VBC (WVBC) and the aggregate VBC (AVBC). Upper and lower bounds of the relative additional storage for both schemes are derived. An important parameter for the bounds of both schemes is the word size (or the number of binary

digits) of the term frequencies. Two encoding methods to reduce the word size are also discussed here: a linear and a logarithmic scaling.

3.1.1. Weighted VBC

Weighted VBC (WVBC) builds a Part IV for the VBC signature. This part is simply a bit string, which is obtained by concatenating the fixed-size binary numbers that represent the non-zero term frequencies. For example, if there is a document that has a VBC signature as in Table 2, where four bits are used to represent a term frequency and the three terms in the document have frequencies one, two and four, then Part IV of the VBC signature is the concatenation of 0001_2, 0010_2 and 0100_2.

The optimal bit-block size of the WVBC signature is the same as for the VBC signature, because the storage cost of Part IV is not a function of the bit-block size b. The storage cost of the WVBC signature for document i is:

$$| e_i | = \frac{| t |}{b} + n_i + n_i \log_2 b + sn_i$$

where s is the word size (in bits) of the term frequencies, n_i is the number of different terms found in the document and $|t|$ is the vocabulary size. Substituting the optimal bit-block size $b_{\text{opt}} = \ln 2\ |t|/n_i$, the above equation becomes:

$$| e_i | = n_i\left(c + s - \log_2 \frac{n_i}{| t |}\right)$$

where $c = 1/\ln 2 + 1 + \log_2 \ln 2 \sim 1.91$. The relative storage increase R_e can be expressed by dividing the above with the storage of VBC for the optimal bit-block size:

$$\boldsymbol{R}_e = \frac{| \boldsymbol{e}_i |}{| \boldsymbol{v}_i |} - 1 = \frac{\boldsymbol{s}}{\boldsymbol{c} - \log_2 \frac{\boldsymbol{n}_i}{| \boldsymbol{t} |}}$$

Using the condition that $1/|t| \le p(1) = n_i/|t| \le 1$, R_e is bounded by:

$$\frac{s}{c + \log_2 | t |} \le R_e \le 0.52s$$

Table 2
A VBC signature is divided into three-bit strings: Part I, II and III. The bit-block size is four. For Part III, two binary digits are used to encode the positions of the one-bits in the bit-block

Bit vector	Bit blocks			
	0000	1010	0000	0001
Part I	0	1	0	1
Part II	Undefined	10	Undefined	0
Part III		0010		11

assuming that there is at least one term found in the document. If $s = 16$ bits and $|t| = 2^{16}$ terms, then the additional storage due to Part IV is between 1.9 and 9.3 times the storage of the VBC without Part IV. In this case, the lower bound of R_e is quite large. If the term frequencies can be encoded to use a smaller word size, then R_e can be reduced proportionally with the word size reduction.

3.1.2. Aggregate VBC

An alternative to store term frequencies of the VBC signature is to append the binary representation of the term frequencies as bit strings to the (dictionary) bit vector. The VBC signature is generated for the bit vector, which is an aggregate of the dictionary bit-vector and the binary representations of the non-zero term frequencies. The optimal bit-block size would be determined using the aggregate bit vector instead of the original (dictionary) bit vector. We refer this alternative as the aggregate VBC (AVBC).

For AVBC, the bit vector size is increased from $|t|$ to $|t| + sn_i$ and the number of one bits for the binary representation of all the non-zero term frequencies is denoted as x_i. The storage cost of AVBC is the sum of the three parts, i.e.:

$$| a_i | = \frac{| t | + sn_i}{b} + n_i + x_i + (n_i + x_i) \log_2 b$$

The optimal bit-block size for AVBC is found to be:

$$b_{\text{opt}} = \ln 2 \frac{| t | + sn_i}{n_i + x_i}$$

Using the optimal bit-block size, the AVBC storage cost is:

$$| a_i | = (n_i + x_i)\left(c + \log_2 \frac{| t | + sn_i}{n_i + x_i}\right)$$

The additional, relative storage cost R_a is:

$$\boldsymbol{R}_a = \frac{| \boldsymbol{a}_i |}{| \boldsymbol{v}_i |} - 1 = \frac{\boldsymbol{x}_i\left(\boldsymbol{c} + \log_2 \frac{| \boldsymbol{t} | + \boldsymbol{sn}_i}{\boldsymbol{n}_i + \boldsymbol{x}_i}\right) + \boldsymbol{n}_i\left(\log_2\left(\frac{| \boldsymbol{t} | + \boldsymbol{sn}_i}{\boldsymbol{n}_i + \boldsymbol{x}_i} \frac{\boldsymbol{n}_i}{| \boldsymbol{t} |}\right)\right)}{\boldsymbol{n}_i\left(\boldsymbol{c} + \log_2 \frac{| \boldsymbol{t} |}{\boldsymbol{n}_i}\right)}$$

Since $n_i \le x_i \le s \times n_i$, R_a is bounded by (see the Appendix for the derivation):

$$1 \le R_a \le s + 0.52(1 + s) \log_2(1 + s)$$

Clearly, the upper bound of R_a is always larger than the upper bound of R_e. However, the lower bound of R_a can be smaller than R_e depending on s. Although the WVBC signature is more preferable than the AVBC signature, by considering the analytical bounds of the additional storage increase, the actual additional storage increase for WVBC signatures could be larger than that of the AVBC signature, since the lower bound of R_a can be smaller than

the upper bound of R_e, depending on the word size s. Experiments have to confirm whether WVBC or AVBC signatures have a lower storage cost in practice.

3.2. Term frequency scaling

The storage costs of both WVBC and AVBC signatures are functions of the word size s of the non-zero term frequencies. Without any compression, an (unsigned) integer is an m-bit word (which is usually 4 bytes long). To reduce the amount of bits to represent the term frequencies, it is possible to scale the m-bit values down to s bits. The largest value 2^s of an s-bit integer would correspond to the largest value 2^m of an m-bit integer. The smallest values of all different word size integers would be the same (i.e. 0).

Using the largest and smallest values as the boundary conditions, two methods can scale the values between an s-bit integer and the other m-bit integer. The first method is to scale the m-bit integer z to the s-bit integer y linearly, i.e.:

$$y = \frac{2^s - 1}{2^m - 1} z$$

The second method scales down the integer z by a logarithmic transformation:

$$y = (2^s - 1)\frac{\log(z+1)}{m \ \log \ 2}$$

The logarithmic transformation has a *commanding* effect, which magnifies small and decreases large values. This effect also maintains a constant, relative quantization error for different values of z.

Intuitively, linear scaling seems to be a suitable choice because terms with low frequencies in the document are not important, since the ultimate weight is derived from the term frequencies. On the other hand, the inverse document frequency can scale up the quantization error of the term frequencies, so that the document ranking becomes distorted. An experiment is performed to resolve which scaling method is preferred.

4. Evaluation

For evaluation, we collected 4276 documents, which occupied about 9.94 M bytes and contained 183,869 unique terms (i.e. $|t| = 183{,}869$). Altogether 30 queries were used to determine the speed and storage, and 24 for the ranking performance of the retrieval system. The relevance score for document ranking is determined on the basis of the mixed minimum and maximum (MMM) model developed by Fox and Sharat (1986). The MMM model is an improved model of relevance computation based on fuzzy concepts.

4.1. Signature scheme comparison

Table 3 shows performance of different signature schemes: VBC, WVBC and AVBC. The

VBC signature file size is about 12% of the document size and WVBC signature file size a slightly higher percentage of 20%. Although AVBC signature file size is significantly higher than that of VBC, it remains less than 50%. The additional storage increase of WVBC signature is found to be $R_e = 0.55$, which is closer to the lower bound of R_e (i.e. 0.40) than the upper bound (i.e. 4.2). Likewise, $R_a = 2.19$ is closer to the lower bound (i.e. 1) than the upper bound (i.e. 22.8). Clearly, the storage performance of WVBC signatures is better than AVBC signatures.

The time spent to perform indexing is about the same for all three signature schemes (i.e. between 18 and 19 min), but the VBC signature retrieves the documents fastest. Although WVBC signatures are about twice as slow as VBC signatures to retrieve documents, AVBC signatures are too slow to retrieve documents for operational deployment. The significant time cost is mainly in decoding AVBC signatures, because the term frequencies have many bits with a value of one. The time cost for scaling the term frequencies is almost negligible since the indexing speed between the three different signature schemes is about the same.

Without document ranking, the average recall was only 29%. With document ranking, using the MMM model, the average recall is increased to 98%, but this is achieved at the expense of a drop in the average precision from 89 to 7%. Fig. 1 shows the recall and precision trade-off graph, which is typical to many retrieval systems with document ranking.

4.2. Term frequency scaling comparison

Term frequency scaling introduces quantization errors, which cause errors in document ranking. To evaluate the scaling effects, 24 queries were used. Two were discarded because they did not retrieve any document from the collection.

To evaluate document ranking, the ranks of the documents retrieved using term frequencies without scaling are used as a baseline for comparison with ranks after linear and logarithmic scaling. Ranking deviations from the baseline would be due to quantization errors only, since we do not interpret whether the ranking by the baseline is correct or not. A measure of the compatibility or agreement of the two sets of document ranks is the Kendall (1970) rank-order correlation T, which is non-parametric. If T is one, then the ranks are in agreement, or in the

Table 3
The different storage, speed and retrieval effectiveness performances for different VBC signature schemes[a]

Signature scheme	Storage (bytes)		Speed		Effectiveness	
	Signature file size	Additional storage	Indexing speed	Average retrieval speed	Average recall (%)	Average precision (%)
VBC	1.218 M (12%)	0 (0%)	18 (100%)	3.9 (100%)	29	89
WVBC	1.884 M (20%)	0.666 M (55%)	19 (106%)	9.1 (233%)	98	7
AVBC	3.895 M (39%)	2.667 M (219%)	19 (106%)	76.8 (1970%)	98	7

[a] WVBC for weighted VBC and AVBC for the aggregate VBC.

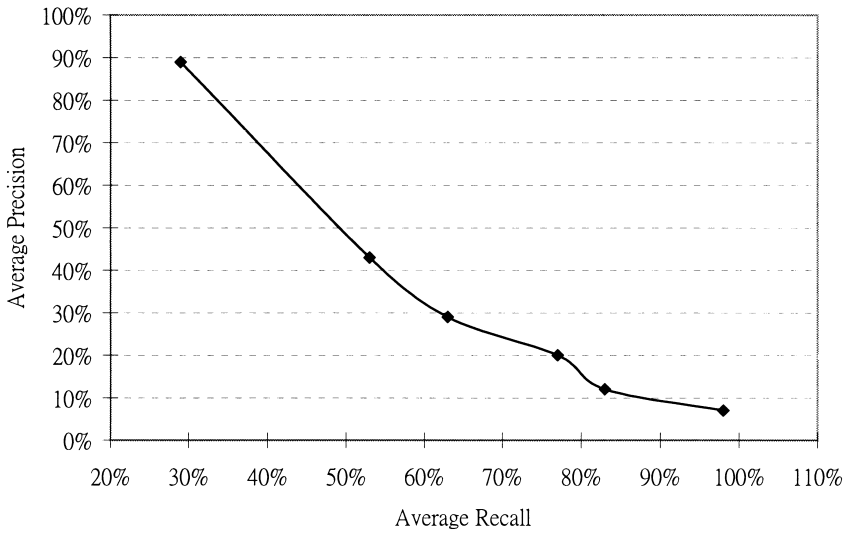

Fig. 1. Precision and recall trade-off for a set of (30) queries.

same order sequence. If T is -1, then the ranks are in agreement in the reverse order. If T is 0, then the ranks have no agreement in the ordering.

Table 4 shows the average values of T for the 22 valid queries. Clearly, logarithmic scaling is better than linear scaling because its correlation (i.e. 0.81) is larger than that for linear scaling (i.e. 0.34). To test for statistical significance, a (query) paired sign-test is carried out, in which the sign of the difference between the T score for logarithmic and linear scaling is counted for the queries. A pair test can reduce effects of variations caused by differences in the queries. There are five queries with the same correlation using logarithmic and linear scaling. The remaining queries (i.e. 17) have a higher correlation for logarithmic scaling than linear scaling (i.e. positive signs). Therefore, logarithmic scaling is statistically significantly better than linear scaling. Effectively, the correlation T for logarithmic scaling is always better or the same as the correlation T for linear scaling.

4.3. Word size against performance

Experiments are carried out to evaluate how word size s affects storage and ranking

Table 4
Ranking performance of linear and logarithmic scaling for 22 valid queries[a]

Average T		Pair difference in T between linear and logarithmic scaling		
Linear scaling	Logarithmic scaling	Average difference	Positive signs	Negative signs
0.34	0.81	0.47	17	0

[a] The measure T is the Kendall rank-order correlation coefficient.

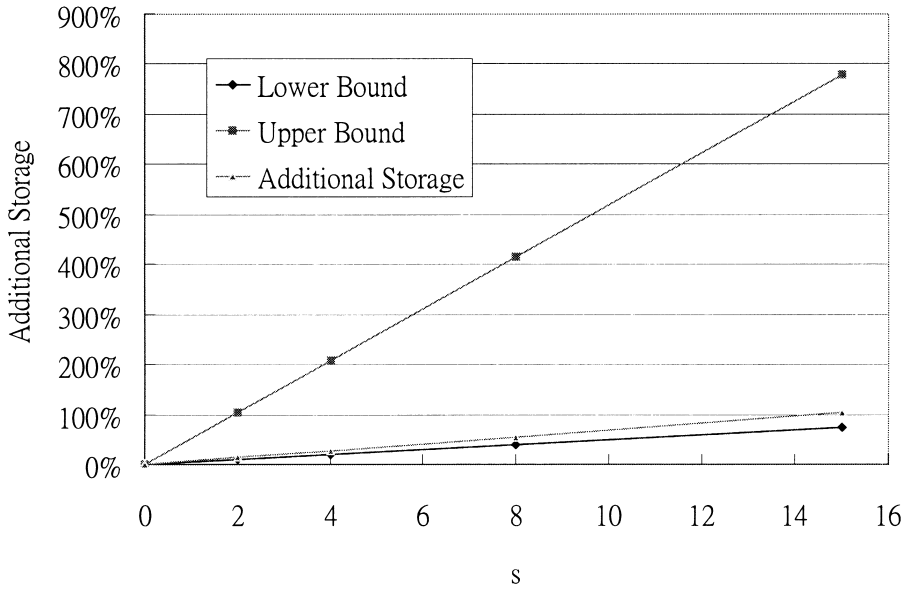

Fig. 2. Variation of additional storage R_e performance against word size s.

performance. We focus on the weighted VBC and logarithmic scaling because they were found to be the best.

Fig. 2 shows how the additional storage R_e performance varies with the word size s. The upper and lower bound of R_e are also plotted. Clearly, R_e varies linearly with s and is much closer to the lower than the upper bound. This is expected since $p(1) \sim 0 \ll 0.26$. R_e is found to

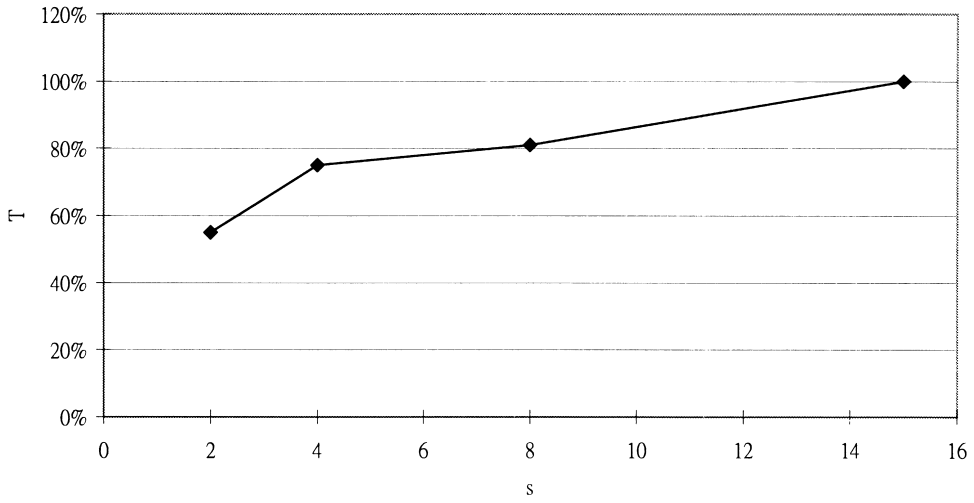

Fig. 3. Variation of Kendall rank-order correlation with different word sizes.

be consistently about 1.36 times larger than the lower bound for the various different word sizes.

Fig. 3 shows the sacrifice of ranking performance with the gain in storage performance by using a smaller word size. The difference in ranking performance between four and eight bits is small (i.e. 6%), but the additional storage has dropped from 55% for eight-bit word size to only 27% for four-bit word size (i.e. the percentage has halved). However, when the word size is below four bits, the ranking performance dropped relatively dramatically to 55%.

5. Conclusion

We have presented two extensions to the VBC signature for document ranking. The weighted VBC (WVBC) scheme has better storage and retrieval speed than the aggregate VBC scheme in terms of both the analytical storage bounds and the performance in practice. Logarithmic scaling of term frequencies is found to be significantly better than linear scaling, in terms of both the magnitude (cf 82 and 34%) and the statistical significance (above p level of 0.01).

The WVBC signature is a viable alternative to the weight-partitioned signature file method. Although the weight-partitioned signature can achieve a storage cost of about 12.5% without stemming and word removal, WVBC achieves an additional storage cost of about 6.7% for a good ranking performance (i.e. 82%). Taking variations into account, WVBC has a similar performance to the weighted-partition signature file using superimposed coding.

For WVBC signatures, if the ranking performance can be further sacrificed, better storage performance can be secured. Since only the ranking is affected by the quantization errors, the precision calculated for all the retrieved documents is unaffected. However, the precision-recall graph would be changed. By contrast, the weight-partitioned signature file makes a trade-off between storage and the false drop probability, instead of the ranking performance. Effectively, the precision is lowered because many irrelevant documents are qualified with a high false drop probability. Thus, we consider that trade-off of storage with ranking performance is preferred over the trade-off of storage with the false drop probability. If the ranking performance of 75% is acceptable, then the WVBC storage performance (i.e. about 3.4% of the document size) is very competitive with the weight-partitioned signature file technique (i.e. about 12.5% of the document size).

Acknowledgements

We are grateful to Prof. D.L. Lee, Department of Computer Science, the Hong Kong University of Science and Technology, for clarifying issues related to the weight-partitioned signature file technique.

Appendix A. Bounds of R_a

Since $x_i \le sn_i$, R_a is upper bounded by

$$R_a \le \frac{sn_i\left(c + \log_2\frac{|t| + sn_i}{n_i + x_i}\right) + n_i\left(\log_2\left(\frac{|t| + sn_i}{n_i + x_i}\,\frac{n_i}{|t|}\right)\right)}{n_i\left(c + \log_2\frac{|t|}{n_i}\right)}$$

$$\le \frac{s\left(c + \log_2\frac{|t| + sn_i}{n_i + x_i}\right) + \log_2\left(\frac{|t| + sn_i}{n_i + x_i}\,\frac{n_i}{|t|}\right)}{\left(c + \log_2\frac{|t|}{n_i}\right)}$$

Using the following inequalities

$$\log_2\left(\frac{|t| + sn_i}{n_i + x_i}\,\frac{n_i}{|t|}\right) \le \log_2\left(\frac{|t| + sn_i}{n_i}\,\frac{n_i}{|t|}\right) = \log_2\left(1 + \frac{sn_i}{|t|}\right) \le \log_2(1 + s)$$

$$\log_2\left(\frac{|t| + sn_i}{n_i + x_i}\right) \le \log_2\left(\frac{|t| + sn_i}{n_i}\right) \le \log_2\left((1 + s)\frac{|t|}{n_i}\right) = \log_2(1 + s) + \log_2\frac{|t|}{n_i}$$

R_a can be bounded by:

$$R_a \le \frac{s\left(c + \log_2\frac{|t|}{n_i}\right) + (1 + s)\log_2(1 + s)}{\left(c + \log_2\frac{|t|}{n_i}\right)} \le s + 0.52(1 + s)\log_2(1 + s)$$

For a lower bound of R_a, since every binary representation must have at least one bit with a value of one for the non-zero term frequencies, $x_i \ge n_i$.

$$R_a = \frac{|a_i|}{|v_i|} - 1 \ge \frac{n_i\left(c + \log_2\frac{|t| + sn_i}{n_i + x_i}\right) + n_i\left(\log_2\left(\frac{|t| + sn_i}{n_i + x_i}\,\frac{n_i}{|t|}\right)\right)}{n_i\left(c + \log_2\frac{|t|}{n_i}\right)}$$

$$\ge \frac{2c + 2\ \log_2\frac{|t| + sn_i}{n_i + x_i} - c + \log_2\frac{n_i}{|t|}}{\left(c + \log_2\frac{|t|}{n_i}\right)} \ge \frac{2\left(c + \log_2\frac{|t| + sn_i}{n_i + x_i}\right)}{\left(c + \log_2\frac{|t|}{n_i}\right)} - 1$$

By observing the following inequality:

$$\log_2\frac{|t| + sn_i}{n_i + x_i} \le \log_2\frac{|t| + sn_i}{n_i + sn_i} \le \log_2\left(\frac{|t|(1 + s)}{(1 + s)}\right) \le \log_2\ |t|$$

the lower bound of R_a becomes:

$$R_a \geq \frac{2\left(c + \log_2 \frac{|t|}{n_i} + \log_2 n_i\right)}{\left(c + \log_2 \frac{|t|}{n_i}\right)} - 1 \geq 1 + \frac{2 \log_2 n_i}{c + \log_2 \frac{|t|}{n_i}} \geq 1$$

for $n_i \geq 1$.

References

Chan, S. K., Wong, Y. C., & Luk, R. W. P. (1998). Variable bit-block compression for English–Chinese information retrieval. In *Proceedings of the 3rd Information Retrieval of Asian Languages. KRDL, National University of Singapore* (pp. 61–66).

Croft, W. B., & Savino, P. (1988). Implementing ranking strategies using text signatures. *ACM Transactions on Office Information Systems*, *6*(1), 42–62.

Faloutsos, C., & Christodoulakis, S. (1987). Description and performance analysis of signature file methods. *ACM Transactions on Office Information Systems*, *5*(3), 237–257.

Fox, E. A., & Sharat, S. (1986). A comparison of two methods for soft boolean interpretation in information retrieval. Technical Report TR-86-1, Department of Computer Science, Virginia Tech.

Golomb, S. W. (1966). Run length encodings. *IEEE Transactions on Information Theory*, *IT-12*, 399–401.

Kendall, M. G. (1970). *Rank correlation methods* (4th ed.). London: Griffin.

Lee, D. L., & Ren, L. (1996). Document ranking on weight-partitioned signature files. *ACM Transactions on Information Systems*, *14*(2), 109–137.

Mooers, C. (1949). Application of random codes to the gathering of statistical information. In *Bulletin 31*. Cambridge, MA: Zator Co.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, *24*(5), 513–523.

Wong, W. Y. P., & Lee, D. L. (1990). Signature file methods for implementing a ranking strategy. *Information Processing and Management*, *26*(5), 641–653.