Universal Hashing of Databases Having Dense Structure

Vladimir B. Balakirsky

Euler Institute for Discrete Mathematics
and its Applications (EIDMA)
c/o Eindhoven University of Technology
P.O.Box 513, 5600 MB Eindhoven
The Netherlands
Email : v.b.balakirsky@tue.nl

1 Introduction

Main memories of a computer usually have random access time. Since their size is limited with the cost requirements, databases are usually stored on external memory with a rather slow access. The typical searching problem is as follows : given a pattern, find addresses of all records that are close to the pattern in some sense. We will understand records and a pattern as binary vectors of the same length and assume that two vectors are close to each other if the Hamming distance between them does not exceed a given value T. The problem can be also introduced as list decoding for a code whose codewords are records when the pattern is the received word.

The designer of a searching procedure usually has only some partial information about the database. In particular, records can consist of several "dense" subsets of vectors, where each subset contains records that are close to each other (these subsets are called clusters). A common approach to solving the problem is known as external hashing [1]. This approach is based on the use of some available part of the working memory filled in advance with the values of the hash–function applied to records. Using these values, we first reject some records that cannot be close to the pattern. At the second step, we access only those records, which remain after the first step, compute the Hamming distances between them and the pattern and output the result.

We consider hashing as a problem of information and coding theory where we assign two functions, the encoding and the decoding, to minimize the expected cardinality of the set of records that should be transmitted to the working memory. The expectation is computed in the probabilistic ensemble of databases introduced in such a way that relatively few positions of records have different contents, while all the others coincide. We develop approaches described in [2], [3] which are based on the use of the metric properties of the Hamming space.

2 Mathematical model for hashing

We address the problem of searching for a pattern in databases (DBs), where the DB is understood as a binary $M \times L$ matrix $\mathbf{X}$ consisting of M records $\mathbf{x}_1, \ldots, \mathbf{x}_M \in \{0,1\}^L$. The pattern is a vector $\mathbf{x} \in \{0,1\}^L$, and we have to construct the set

$$\mathcal{J}(\mathbf{x}, \mathbf{X}) \triangleq \Big\{ j \in \{1, \ldots, M\} : d_{\mathrm{H}}(\mathbf{x}, \mathbf{x}_j) \leq T \Big\}, \tag{1}$$

where $T \geq 0$ is a given threshold and $d_{\mathrm{H}}(\mathbf{x}, \mathbf{x}_j)$ denotes the Hamming distance between $\mathbf{x}$ and $\mathbf{x}_j$.

Suppose that $\ell \in \{1, \ldots, L-1\}$ is given and address the following problem :

⊙ *assign two functions*

$$\begin{array}{rcrcccl} f & : & \mathbf{x}_j & \in & \{0,1\}^L & \longrightarrow & \{0,1\}^\ell \\ \varphi_T & : & (\mathbf{x}, f(\mathbf{x}_j)) & \in & \{0,1\}^L \times \{0,1\}^\ell & \longrightarrow & \{0,1\} \end{array} \tag{2}$$

in such a way that

$$d_{\mathrm{H}}(\mathbf{x}, \mathbf{x}_j) \leq T \implies \varphi_T(\mathbf{x}, f(\mathbf{x}_j)) = 1, \ \textit{for all } \mathbf{x}, \mathbf{x}_j \in \{0,1\}^L. \tag{3}$$

Hashing will be understood as assignment of the particular pair of functions (f, φ_T) satisfying (2), (3). The functions f and φ_T will be referred to as the encoding and the decoding, respectively.

Hashing allows us to construct the set $\mathcal{J}_T(\mathbf{x}, \mathbf{X})$ in two steps :

1) compute $\varphi_T(\mathbf{x}, f(\mathbf{x}_1)), \ldots, \varphi_T(\mathbf{x}, f(\mathbf{x}_M))$ and construct the set $\mathcal{J}(\mathbf{x}, \mathbf{X}|f, \varphi_T)$;

2) sequentially check records $\mathbf{x}_j, j \in \mathcal{J}(\mathbf{x}, \mathbf{X}|f, \varphi_T)$, and construct $\mathcal{J}_T(\mathbf{x}, \mathbf{X})$ as the set

$$\Big\{ j \in \mathcal{J}(\mathbf{x}, \mathbf{X}|f, \varphi_T) : d_{\mathrm{H}}(\mathbf{x}, \mathbf{x}_j) \leq T \Big\}.$$

We suggest the following probabilistic model for the DBs and for the patterns to be assigned. Suppose that a vector $\mathbf{x}_0 \in \{0,1\}^L$ and an integer $N \in \{0, \ldots, L\}$ are given. Furthermore, let $\mathcal{I}$ be a given N–element subset of the set $\{1, \ldots, L\}$,

$$\mathcal{I} \subseteq \{1, \ldots, L\}, \ \ |\mathcal{I}| = N. \tag{4}$$

Let us introduce the probability that a fixed matrix $\mathbf{X}$ is the DB is as the conditional probability

$$\begin{aligned} & P(\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_M)^{\mathrm{T}} | \mathbf{x}_0, \mathcal{I}) \\ \triangleq \ & 2^{-NM} \chi \Big\{ x_j^{(i)} = x_0^{(i)}, \text{ for all } i \notin \mathcal{I} \text{ and } j = \overline{1, M} \Big\}, \end{aligned}$$

where χ stands for the indicator function : $\chi\{S\} = 1$ if the statement S is true and $\chi\{S\} = 0$ otherwise. Let

$$G_{\mathbf{x}_0, \mathcal{I}}(\mathbf{x}|f, \varphi_T) \triangleq \sum_{\mathbf{x}_1, \ldots, \mathbf{x}_M} P(\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_M)^{\mathrm{T}} | \mathbf{x}_0, \mathcal{I}) \cdot \frac{\Big| \mathcal{J}(\mathbf{x}, \mathbf{X}|f, \varphi_T) \Big|}{M}$$

denote the expectation of the relative number of records included in the set $\mathcal{J}(\mathbf{x}, \mathbf{X}|f, \varphi_T)$. Notice that $G_{\mathbf{x}_0,\mathcal{I}}(\mathbf{x}|f, \varphi_T) \in [0,1]$ and denote

$$G(f, \varphi_T) \triangleq \max_{\mathbf{x},\mathbf{x}_0} \max_{\substack{\mathcal{I}\subseteq\{1,\ldots,L\}\\ |\mathcal{I}|=N}} G_{\mathbf{x}_0,\mathcal{I}}(\mathbf{x}|f, \varphi_T), \tag{5}$$

$$\overline{G}(f, \varphi_T) \triangleq \max_{\mathbf{x},\mathbf{x}_0} \frac{1}{\binom{L}{N}} \sum_{\substack{\mathcal{I}\subseteq\{1,\ldots,L\}\\ |\mathcal{I}|=N}} G_{\mathbf{x}_0,\mathcal{I}}(\mathbf{x}|f, \varphi_T). \tag{6}$$

The function $G(f, \varphi_T)$ corresponds to "the worst" assignment for $(\mathbf{x}, \mathbf{x}_0, \mathcal{I})$, while the function $\overline{G}(f, \varphi_T)$ corresponds to "the worst" assignment for $(\mathbf{x}, \mathbf{x}_0)$ when $\mathcal{I}$ is uniformly chosen from the collection of sets satisfying (4). The values of $G(f, \varphi_T)$ and $\overline{G}(f, \varphi_T)$ can be accepted as criteria of the quality of hashing, and we represent our results as possible solutions to the optimization problems of minimizing these values over the domains that consist of all pairs (f, φ_T) satisfying (2), (3).

Numerical example.

Suppose that the DB consists of M records of length $L = 256$. Each record should be encoded to a vector of length $\ell = 12$. For a given pattern of length 256 and a codeword of length 12 corresponding to some record, we want to answer the question : *Is it possible that the Hamming distance between the pattern and the record does not exceed T ?* Let the DB be determined by "an opponent" after the encoding and the decoding are fixed. The opponent proceeds as follows. He selects $N = 64$ positions and puts the same bit $x_0^{(i)}$ to the i–th position of all records if i does not belong to the set of selected positions. Otherwise, if i belongs to the set of selected positions, then the i–th bits of each record are generated by independent flippings of a fare coin, and the opponent does not know outcomes of this random experiment. The opponent also specifies all L bits of the pattern.

"A direct" approach to hashing could be as follows. Assign

$$f'(\mathbf{x}_j) \triangleq (x_j^{(1)}, \ldots, x_j^{(\ell)}), \text{ for all } \mathbf{x}_j \in \{0,1\}^L, \tag{7}$$

$$\varphi'_T(\mathbf{x}, f'(\mathbf{x}_j)) \triangleq \chi\Big\{ d_{\mathrm{H}}(f'(\mathbf{x}), f'(\mathbf{x}_j)) \le T \Big\}, \text{ for all } \mathbf{x}, \mathbf{x}_j \in \{0,1\}^L. \tag{8}$$

Some values of $G(f, \varphi_T)$ and $\overline{G}(f, \varphi_T)$ for $(f, \varphi_T) = (f', \varphi'_T), (f_K^{(q)}, \varphi_{T,K}^{(q)})$, $(q, K) = (8, 4)$ and $T = 0, \ldots, 12$ are given in Table 1, where the functions $f_K^{(q)}$ and $\varphi_{T,K}^{(q)}$ will be introduced below in a general case.

The situation described above can be considered as a very difficult one, because the compression factor 256/12 is big and the designer has no information about the content of the DB. Nevertheless, if $T = 0$ and hashing is specified by the pair of functions $(f_K^{(q)}, \varphi_{T,K}^{(q)})$, then we only leave about 0.14 percents of the total number of records in the list of records that can coincide with the pattern in the average case (the function $\overline{G}$) and leave about 12.65 percents of the total number of records in the worst case (the function G). These values increase with T, but much slower than similar values for the pair (f', φ'_T). Notice also that the value of the function $\overline{G}$ is less than 1 even for $T = \ell = 12$.

3 Hashing algorithm

We will use the following notations. The number of 1's in the vector $\mathbf{x}$, known as the Hamming weight of $\mathbf{x}$, will be denoted by $\mathrm{wt}(\mathbf{x})$. Given integers a and $q \ge 2$, let $a^{[q]}$

Table 1: The values of $G(f, \varphi_T)$ and $\overline{G}(f, \varphi_T)$, where $(f, \varphi_T) = (f', \varphi'_T), (f_K^{(q)}, \varphi_{T,K}^{(q)})$, for $(L, \ell, N) = (256, 12, 64)$ and $(q, K) = (8, 4)$; the pair (f', φ'_T) is defined in (7), (8), and the pair $(f_K^{(q)}, \varphi_{T,K}^{(q)})$ is introduced in the Definition.

T	$G(f', \varphi'_T)$	$\overline{G}(f', \varphi'_T)$	$G(f_4^{(8)}, \varphi_{T,4}^{(8)})$	$\overline{G}(f_4^{(8)}, \varphi_{T,4}^{(8)})$
0	1	0.1982	0.1265	0.0014
1	1	0.5453	0.3788	0.0118
2	1	0.8196	0.6288	0.0469
3	1	0.9895	0.8765	0.1239
4	1	0.9984	1	0.2491
5	1	0.9998	1	0.4111
6	1	0.9999	1	0.5834
7	1	0.9999	1	0.7371
8	1	0.9999	1	0.8536
9	1	0.9999	1	0.9288
10	1	0.9999	1	0.9702
11	1	0.9999	1	0.9895
12	1	1	1	0.9969

denote "a modulo q", and we write $a^{[q]} \triangleq a, \mod q$. For all integers a and b, let

$$d_{\text{Lee}}^{(q)}(a, b) \triangleq \min_{s \in \{0, \pm 1, \pm 2, \ldots\}} |\, a - b + sq \,|$$

stand for the Lee distance between a and b in the lattice $0, \pm q, \pm 2q, \ldots$

Definition. *Suppose that $L = Kr$ and $\ell = L \log q$, where K and q are integers. Represent each record and each pattern as concatenations of K subblocks of length r, i.e., write $\mathbf{x}_j = (\boldsymbol{x}_{j1}, \ldots, \boldsymbol{x}_{jK})$ and $\mathbf{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K)$, where $\boldsymbol{x}_{jk}, \boldsymbol{x}_k \in \{0, 1\}^r$ for all $k = \overline{1, K}$. Define $f_K^{(q)}(\mathbf{x}_j)$ as the concatenation of the standard binary representations of integers*

$$\text{wt}^{[q]}(\boldsymbol{x}_{j1}), \ldots, \text{wt}^{[q]}(\boldsymbol{x}_{jK}) \in \{0, \ldots, q-1\},$$

where each of K entries is specified by $\lceil \log q \rceil$ bits. Define the decoding as

$$\varphi_{T,K}^{(q)}(\mathbf{x}, f_K^{(q)}(\mathbf{x}_j)) \triangleq \chi\Big\{ \sum_{k=1}^{K} d_{\text{Lee}}^{(q)}\Big(\text{wt}(\boldsymbol{x}_k), \text{wt}^{[q]}(\boldsymbol{x}_{jk}) \Big) \le T \Big\}. \qquad (9)$$

4 Evaluation of the performance of the hashing algorithm

The performance of the hashing algorithm for a fixed pair (q, K) can be analyzed by the generating functions technique with the exact values of the entries of the probability distributions

$$\Theta_n^{(q)} = \Big(\sum_{d=0}^{n} \binom{n}{d} 2^{-n} \chi\{ d^{[q]} = \delta \}, \, \delta \in \{0, \dots, q-1\} \Big), \quad n = \overline{1, N}.$$

In particular, this technique was used to compute data in Table 1.

The following lower bound is derived from the observation that the uniform partitioning of the DB in 2^ℓ groups is the best possible, $G(f, \varphi_T) \geq (T+1)2^{-\ell}$. If the value of the parameter N is big enough, then this bound can be attained with the coefficient at most 4.

Theorem 1. *If*

$$\ell \leq \log \sqrt{N} + \log \sqrt{\pi/2} \tag{10}$$

and $q = 2^\ell$, *then*

$$\overline{G}(f_1^{(q)}, \varphi_{T,1}^{(q)}) \leq G(f_1^{(q)}, \varphi_{T,1}^{(q)}) \leq 2(2T+1)2^{-\ell}. \tag{11}$$

Notice that inequality (10) also implies $\ell < \log L$, since $N \leq L$. If (10) is not satisfied, but

$$N \sim L, \quad \ell \sim \log L, \tag{12}$$

then the possibility of partitioning the records in K blocks of length L/K allows us to adopt given values of the parameters to the data required for a good performance of the hashing algorithm. Denote $\mathcal{I}_k \triangleq \Big\{(k-1)r+1, \dots, kr\Big\} \bigcap \mathcal{I}$ for all $k = \overline{1, K}$ and restrict considerations to the following approximations of the functions $G(f_K^{(q)}, \varphi_{T,K}^{(q)})$ and $\overline{G}(f_K^{(q)}, \varphi_{T,K}^{(q)})$ defined in (5), (6) when $f = f_K^{(q)}$ and $\varphi = \varphi_{T,K}^{(q)}$.

1). Suppose that N is divisible by r and that there are $N/r = KN/L$ blocks where all bits of M records are chosen at random. All bits of the records whose indices do not belong to the set of selected blocks coincide with corresponding bits of the vector $\mathbf{x}_0$. Thus, there is a set $\{k_1, \dots, k_{N/r}\} \subseteq \{1, \dots, K\}$ of cardinality N/r such that $|\, \mathcal{I}_k \,| = r$ if $k \in \{k_1, \dots, k_{N/r}\}$ and $|\, \mathcal{I}_k \,| = 0$ otherwise, where $k = \overline{1, K}$. If the maximum on $\mathcal{I}$ in (5) is replaced with the maximum on $k_1, \dots, k_{N/r}$, then we write $\hat{G}(f_K^{(q)}, \varphi_{T,K}^{(q)})$ instead of $G(f_K^{(q)}, \varphi_{T,K}^{(q)})$.

2). Suppose that N is divisible by K and that each block has exactly N/K positions where bits of the records are chosen at random. Thus, $|\, \mathcal{I}_k \,| = N/K$ for all $k = \overline{1, K}$. If the sum on $\mathcal{I}$ in (6) is replaced with the sum on the sets having this property and the normalizing factor $\binom{L}{N}^{-1}$ is replaced with the factor $\binom{L/K}{N/K}^{-K}$, then we write $\hat{\overline{G}}(f_K^{(q)}, \varphi_{T,K}^{(q)})$ instead of $\overline{G}(f_K^{(q)}, \varphi_{T,K}^{(q)})$.

The reason for introducing the approximations above is the point that an investigation of the functions G and $\overline{G}$ requires rather complicated formalism. Nevertheless, if we proceed with the analysis, then conclusions about the behavour of the functions G and $\overline{G}$ are essentially the same as the conclusions presented below for the functions $\hat{G}$ and $\hat{\overline{G}}$.

Theorem 2.

1. If the parameter K is chosen in such a way that $\ell \leq K \log \sqrt{L/K} + K \log \sqrt{\pi/2}$, *then*

$$\hat{G}(f_K^{(q)}, \varphi_{T,K}^{(q)})\Big|_{q=2^{\ell/K}} \leq \Big(4^K (T+1)^K 2^{-\ell}\Big)^{N/L}.$$

2. If the parameter K is chosen in such a way that $\ell \le K \log \sqrt{N/K} + K \log \sqrt{\pi/2}$, then

$$\overline{\hat{G}}(f_K^{(q)}, \varphi_{T,K}^{(q)})\Big|_{q=2^{\ell/K}} \le 4^K (T+1)^K 2^{-\ell}.$$

Thus, if (12) holds, K is a constant and $T \sim L^\alpha$, $\alpha \in (0,1)$, then upper bounds on $\hat{G}$ and $\overline{\hat{G}}$ decrease with L as a function $1/L^{1-\alpha}$; the value of T can be certainly greater than ℓ. The ratio N/L can be viewed as the probability that the bit of a record located at some fixed position is chosen at random according to a uniform distribution. This probability affects only the upper bound on the function $\hat{G}$ that estimates "the worst case performance" of the hashing algorithm and does not affect the upper bound on the function $\overline{\hat{G}}$ that estimates "the average case performance" of the algorithm.

The results of Theorems 1,2 confirm that our hashing scheme allows one to attain a good performance of hashing for a wide class of DBs.

References

[1] G. H. Gonnet and P.–A. Larson, "External hashing with limited internal storage," *Journal of ACM*, vol. 35, pp. 161–184, 1988.

[2] V. N. Koshelev, "Computational aspects of the triangle axiom," *Cybernetics and Computer Technique*, vol. 4, Moscow : Nauka, pp. 204–216, 1988. (In Russian).

[3] V. B. Balakirsky, "Hashing of databases based on indirect observations of Hamming distances," *IEEE Trans. Inform. Theory*, vol. 42, pp. 664–671, 1995.