f-Fractional Bit Minwise Hashing

Xinpan YUAN, Jun LONG*, Zuping ZHANG, Yueyi LUO, Hao Zhang, Weihua Gui
School of Information Science and Engineering ,Central South University ,Changsha 410083 ,China
Email: dragone7968@163.com

***Abstract*—In information retrieval, minwise hashing algorithm is often used to estimate similarities among documents. *b*-bit minwise hashing is capable of gaining substantial advantages in terms of computational efficiency and storage space by only storing the lowest *b* bits of each (minwise) hashed value (e.g., *b*=1 or 2). In this paper, we propose a fractional bit hashing method, which extends the existing *b*-bit Minwise hashing. It is shown theoretically that the fractional bit hashing has a wider range of selectivity for accuracy and storage space requirements. Theoretical analysis and experimental results demonstrate the effectiveness of this method.**

***Index Terms*—similarity, hashing, fractional bit**

I. INTRODUCTION

The explosive expansion of information sources present on the World Wide Web has resulted in more than 20% redundant web documents and thus created a serious problem for internet search engines [1]. Duplicate document detection in intellectual property protection and information retrieval has important applications, and plays an important role in efficient information assessment. Duplicate document detection is to determine similarity of document content so as to detect plagiarism. Plagiarism does not just mean intact copy, but also includes the original shift change, synonym replacement and changes in repeat such claims.

Research on duplicate document detection started from 1970. Primeval copy detection technology only uses a simple program counting test to detect similarity. The natural semantics of copy detection technology appear until 1990s. In 1993, Manber proposed a sif tool[2] for large-scale file system to find documents with similar content. In 1995, Brin, Garcia-Molina and others in the "digital library" project first proposed the text copy detection mechanism COPS (copy protection system)system[3] and the corresponding algorithm. Garcia-Molina and Shivakumar proposed the SCAM (Stanford copy analysis method) prototypes[4, 5] to improve the COPS system used to detect IP conflict. SCAM used the vector space model and word frequency statistics to measure text similarity [6]. Later, Garcia-Molina and Shivakumar proposed dSCAM model[7]. In 2000, Monostori established a MDR (match detect reveal) prototype, which uses the suffix tree to search for the largest substring between the strings [8-11]. Monostori and others have also made use of the suffix vector (suffix vector) storage [12]. Another interesting work is done by SONG Qin, who developed CDSDG (copying detection system of digital goods) system to detect illegal copying of digital goods and their proliferations [13].

In this paper, our duplicate detection system is designed for a research fund committee, who receives a large number of applications every year. The purpose of this system is to avoid duplicate submissions (e.g., a financed proposal might be resubmitted without little modification). With the rapid increase in the number of financed projects, the size of project collection will reach millions. We use Minwise hashing[14] for text similarity detection in this system. Minwise hashing converts the set intersection problem into the probability of occurrence of an event to estimate the similarity between the documents, through a large number k of experiments.

As a general technique for estimating set similarity, Minwise hashing has been applied to a wide range of applications, for example, duplicate web pages removal[15-17], wireless sensor networks[18], community extraction and classification in the Web graph[19], text reuse in the Web[20], Web graph compression[21] and many more. Since then, there have been considerable theoretical and methodological developments [22-26].

By only storing the lowest *b* bits of each (minwise) hashed value (e.g., $b = 1$ or 2), *b*-bit minwise hashing[27] can gain substantial advantages in terms of computational efficiency and storage space. Based on the *b*-bit Minwise hashing method, the theoretical framework of fractional bit hashing method is established, which has a wide range of selectivity for accuracy and storage space requirements.

II. RELATED DEFINITION AND ALGORITHM

A. Minwise Hashing

Computing the size of set intersections is a fundamental problem in information retrieval, databases, and machine learning. Given two shingles sets $S_1, S_2 \subseteq \Omega$ where $\Omega = \{0,1,...,D-1\}$, by shingling[16] document D_1, D_2.

Resemblance $R(1,2)$ of document S_1 and S_2:

$$R(1,2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} = \frac{a}{f_1 + f_2 - a}$$

where $f_1 = |S_1|, f_2 = |S_2|, a = |S_1 \cap S_2|$

Manuscript received May 15, 2011; revised June 7, 2011; accepted June 23, 2011.

*Corresponding author: Jun LONG
Email addresses: jlong@mail.csu.edu.cn

doi:10.4304/jsw.7.1.228-236

After k minwise independent random permutations, denoted by π_1, π_2, ..., π_k, one can estimate R without bias, as a binomial probability, i.e.,

$$\pi .\Omega \rightarrow \Omega , \Omega = \{0, 1, \ldots, D-1\}$$

An elementary probability argument[14] can show

$$\Pr(\min\{\pi(S_1)\} = \min\{\pi(S_2)\}) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} = R(1,2) \quad (1)$$

Hence, by choosing k independent random permutation π_1, π_2, ..., π_k, each document D is converted into the list:

$$\overline{S_D} = (\min\{\pi_1(S_D)\}, \min\{\pi_2(S_D)\}, ..., \min\{\pi_k(S_D)\})$$

An unbiased estimator of R denoted by $\widehat{R}_M$:

$$\hat{R}_M = \frac{1}{k}\sum_{j=1}^{k} 1\{\min(\pi_j(S_1)) = \min(\pi_j(S_2))\} \quad (2)$$

$$Var(\hat{R}_M) = \frac{1}{k}R(1-R) \quad (3)$$

In minwise hashing, a sample is a hashed value, e.g., $\min(\pi_j(S_i))$, which may require e.g., 64 bits to store, depending on the universal size D. The total storage for each set would be bk bits, where $b = 64$ is possible.

B b-bit Minwise Hashing

By only storing the lowest b bits of each (minwise) hashed value, b-bit Minwish hashing greatly enhances the computational efficiency and reduces the required storage space,

Define the minimum values under π to be z_1,z_2:

$$z_1 = \min\{\pi(S_1)\}, \qquad z_2 = \min\{\pi(S_2)\}$$

Define $e_{1,i}$ = ith lowest bit of z_1, and $e_{2,i}$ = ith lowest bit of z_2. Derives the analytical expression for E_b [27]:

$$E_b = \Pr(\prod_{i=1}^{b} 1\{e_{1,i} = e_{2,i}\} = 1) = C_{1,b} + (1 - C_{2,b})R \quad (4)$$

where

$$r_1 = \frac{f_1}{D}, r_2 = \frac{f_2}{D} \quad (5)$$

$$C_{1,b} = A_{1,b}\frac{r_2}{r_1 + r_2} + A_{2,b}\frac{r_1}{r_1 + r_2} \quad (6)$$

$$C_{2,b} = A_{1,b}\frac{r_1}{r_1 + r_2} + A_{2,b}\frac{r_2}{r_1 + r_2} \quad (7)$$

$$A_{1,b} = \frac{r_1[1-r_1]^{2^b-1}}{1-[1-r_1]^{2^b}} \quad (8)$$

$$A_{2,b} = \frac{r_2[1-r_2]^{2^b-1}}{1-[1-r_2]^{2^b}} \quad (9)$$

An unbiased estimator of R, denoted by $\widehat{R}_b$:

$$\hat{R}_b = \frac{\hat{E}_b - C_{1,b}}{1 - C_{2,b}} \quad (10)$$

$$\hat{E}_b = \frac{1}{k}\sum_{j=1}^{k}(\prod_{i=1}^{b} 1\{e_{1,i,\pi_j} = e_{2,i,\pi_j}\} = 1) \quad (11)$$

Following property of binomial distribution, the variance of $\widehat{R}_b$:

$$\begin{aligned} Var(\hat{R}_b) &= \frac{Var(\hat{E}_b)}{[1-C_{2,b}]^2} = \frac{1}{k}\frac{E_b(1-E_b)}{[1-C_{2,b}]^2} \\ &= \frac{1}{k}\frac{[C_{1,b} + (1-C_{2,b})R][1 - C_{1,b} - (1-C_{2,b})R]}{[1-C_{2,b}]^2} \end{aligned} \quad (12)$$

For large b ($A_{1,b}, A_{2,b} \rightarrow 0, C_{1,b}, C_{2,b} \rightarrow 0$), $Var(\hat{R}_b)$ converges to $Var(\hat{R}_M)$.

III. F-FRACTIONAL BIT MINWISE HASHING

One deficiency of the b-bit hash algorithm is that b can only be integers, meaning it could not achieve finer storage space and precision trade-offs. We extend this algorithm to fractional cases by proposing the so called *f*-fractional bit hashing to approximate different precision, accuracy and storage space other in the integer locations.

A The Unbiased Estimator of f-fractional bit

Let b_1, b_2 be integer bits, for example, b_1=1 b_2=2。

Let w_1 be proportion of b=b_1, w_2 be proportion of b= b_2 ($w_1 = \frac{k_1}{k}$, $w_2 = \frac{k_2}{k}$), if the total number of samples k is 1000 and w_1=1/2 w_2=1/2, then k_1=500, k_2 =500.

Define f:

$$f = w_1 b_1 + w_2 b_2 \ , \ w_1 + w_2 = 1 \ , \ b_1 \neq b_2 \quad (13)$$

Let $y_1 = e_{1,1}e_{1,2}...e_{1,b_x}, y_2 = e_{2,1}e_{2,2}...e_{2,b_x} (b_x = b_1 or b_2)$

$$\begin{aligned} E_f &= \Pr(y_1 = y_2) \\ &= \Pr(\prod_{i=1}^{b_x} 1\{e_{1,i} = e_{2,i}\} = 1), b_x = b_1 \ or b_2 \\ &= \Pr(b_x = b_1)\Pr(\prod_{i=1}^{b_1} 1\{e_{1,i} = e_{2,i}\} = 1) \\ &+ \Pr(b_x = b_2)\Pr(\prod_{i=1}^{b_2} 1\{e_{1,i} = e_{2,i}\} = 1) \\ &= w_1 \Pr(\prod_{i=1}^{b_1} 1\{e_{1,i} = e_{2,i}\} = 1) + w_2 \Pr(\prod_{i=1}^{b_2} 1\{e_{1,i} = e_{2,i}\} = 1) \\ &= w_1(C_{1,b_1} + (1 - C_{2,b_1})R) + w_2(C_{1,b_2} + (1 - C_{2,b_2})R) \end{aligned} \quad (14)$$

An unbiased estimator of R denoted by $\widehat{R}_f$:

$$\hat{R}_f = \frac{\hat{E}_f - (w_1 C_{1,b_1} + w_2 C_{1,b_2})}{1 - (w_1 C_{2,b_1} + w_2 C_{2,b_2})} \quad (15)$$

$$\hat{E}_f = \frac{1}{k}\sum_{j=1}^{k}\{\prod_{i=1}^{b_x} 1\{e_{1,i,\pi_j} = e_{2,i,\pi_j}\} = 1\} \quad (16)$$

where e_{1,i,π_j} (e_{2,i,π_j}) denotes the ith lowest bit of z_1 (z_2), under the permutation π_j.

when w_1=1, w_2=0 (w_1=0, w_2=1), (14) Simplified to (10).

The variance of $\widehat{R}_f$:

$$Var(\hat{R}_f)=\frac{Var(\hat{E}_f)}{[1-(w_1C_{2,b_1}+w_2C_{2,b_2})]^2}=\frac{Var(w_1\hat{E}_{b_1}+w_2\hat{E}_{b_1})}{[1-(w_1C_{2,b_1}+w_2C_{2,b_2})]^2}$$
$$=\frac{1}{k}\frac{w_1^2Var(\hat{E}_{b_1})+w_2^2Var(\hat{E}_{b_2})+2w_1w_2Cov(\hat{E}_{b_1},\hat{E}_{b_2})}{[1-(w_1C_{2,b_1}+w_2C_{2,b_2})]^2} \quad (17)$$
$$=\frac{1}{k}\frac{w_1^2E_{b_1}(1-E_{b_1})+w_2^2E_{b_2}(1-E_{b_2})}{[1-(w_1C_{2,b_1}+w_2C_{2,b_2})]^2}$$

If k is very large, then $\hat{E}_{b_1}\approx E_{b_1}$, $\hat{E}_{b_2}\approx E_{b_2}$. And the covariance of $\hat{E}_{b_1}\bullet\hat{E}_{b_2}$:

$$Cov(\hat{E}_{b_1},\hat{E}_{b_2})=E(\hat{E}_{b_1}\bullet\hat{E}_{b_2})-E(\hat{E}_{b_1})\bullet E(\hat{E}_{b_2})$$
$$\approx E\{[C_{1,b_1}+(1-C_{2,b_1})R]\bullet[C_{1,b_2}+(1-C_{2,b_2})R]\}$$
$$-E(C_{1,b_1}+(1-C_{2,b_1})R)\bullet E(C_{1,b_2}+(1-C_{2,b_2})R)$$
$$=[C_{1,b_1}+(1-C_{2,b_1})R]\bullet[C_{1,b_2}+(1-C_{2,b_2})R]$$
$$-[C_{1,b_1}+(1-C_{2,b_1})R]\bullet[C_{1,b_2}+(1-C_{2,b_2})R]$$
$$=0$$

where

$$E_{b_1}=C_{1,b_1}+(1-C_{2,b_1})R$$
$$E_{b_2}=C_{1,b_2}+(1-C_{2,b_2})R \quad (18)$$

Fig.1 plots $Var(f)$ for the whole range of $R\in(0, 1)$ and four selected $r_1 = r_2$ values (from 10^{-10} to 0.9). Fig.1 demonstrate that the variance increases with the smaller f-bit which could be defined with w_1, w_2 combination of different values to approximate different precision, accuracy and storage space to meet the kinds of needs.

B The Variance-Space Trade-off

As we decrease f, the space needed for storing each "sample" will be smaller; the estimation variance (17) at the same sample size k, however, will increase. This variance-space trade-off[27] of b-bit Minwise Hashing can be precisely quantified by the storage factor B (b; R, r_1, r_2):

$$B(b;R,r_1,r_2)=b\times Var(\hat{R}_b)\times k$$
$$=\frac{b[C_{1,b}+(1-C_{2,b})R][1-C_{1,b}-(1-C_{2,b})R]}{[1-C_{2,b}]^2} \quad (19)$$

This variance-space trade-off of f-fractional minwise hashing can be precisely quantified by the storage factor F (w_1, w_2, b_1, b_2; R, r_1, r_2):

$$F(w_1,w_2,b_1,b_2;R,r_1,r_2)=f\times Var(\hat{R}_f)\times k$$
$$=(w_1b_1+w_2b_2)\times\frac{w_1^2E_{b_1}(1-E_{b_1})+w_2^2E_{b_2}(1-E_{b_2})}{[1-(w_1C_{2,b_1}+w_2C_{2,b_2})]^2} \quad (20)$$

Fig.2 plots $F(f)$ for the whole range of $R\in(0, 1)$ and four selected $r_1 = r_2$ values (from 10^{-10} to 0.9). Fig.2 shows that when the ratios, r_1 and r_2, are close to 1, it is always desirable to use f=1, almost for the whole range of R. However, when r_1 and r_2 are close to 0, using f=1 has the advantage when about R≥0.4. But it is worth noting that storage factor only shows improved level, but can not show accuracy. Fig.1 shows that $Var(1)$ is larger and $Var(2)$ is more suitable for practical applications, However amount of storage is twice. For the actual system, we can choose 1.25, 1.5, 1.75 other fractional bits to select the appropriate accuracy and storage space.

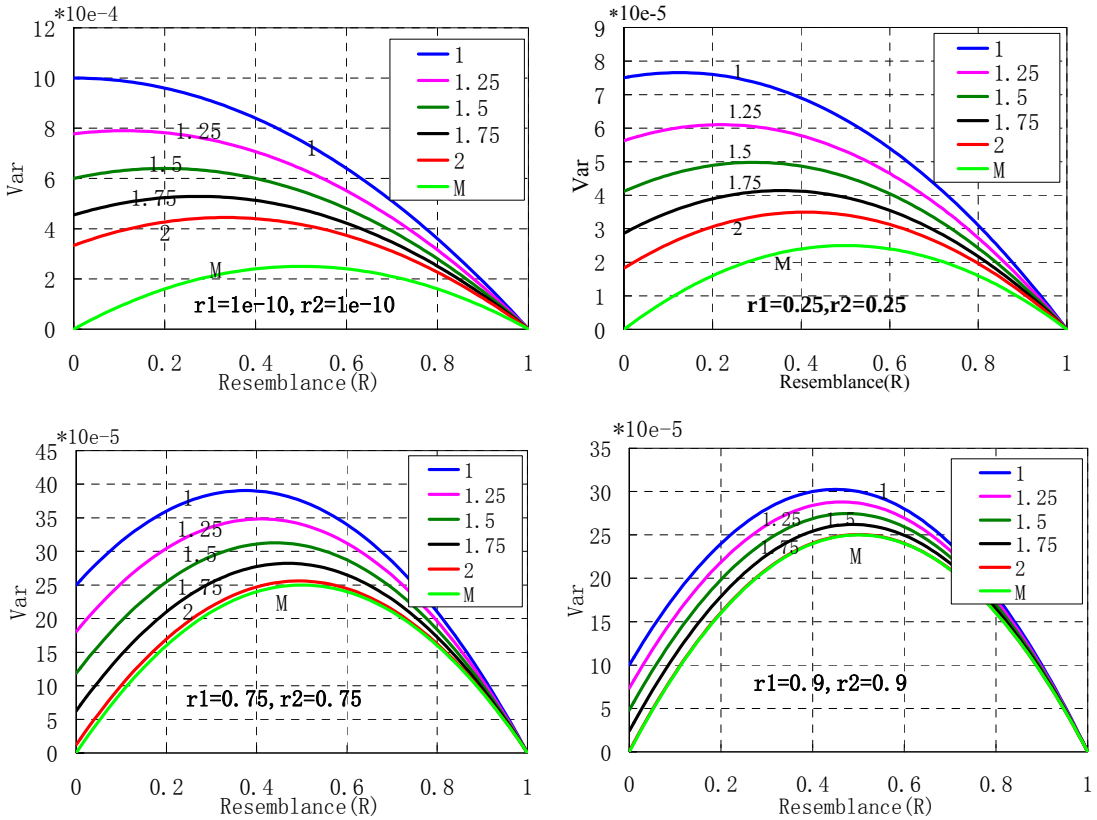


Figure 1. f-fractional bit R-$Var(f)$

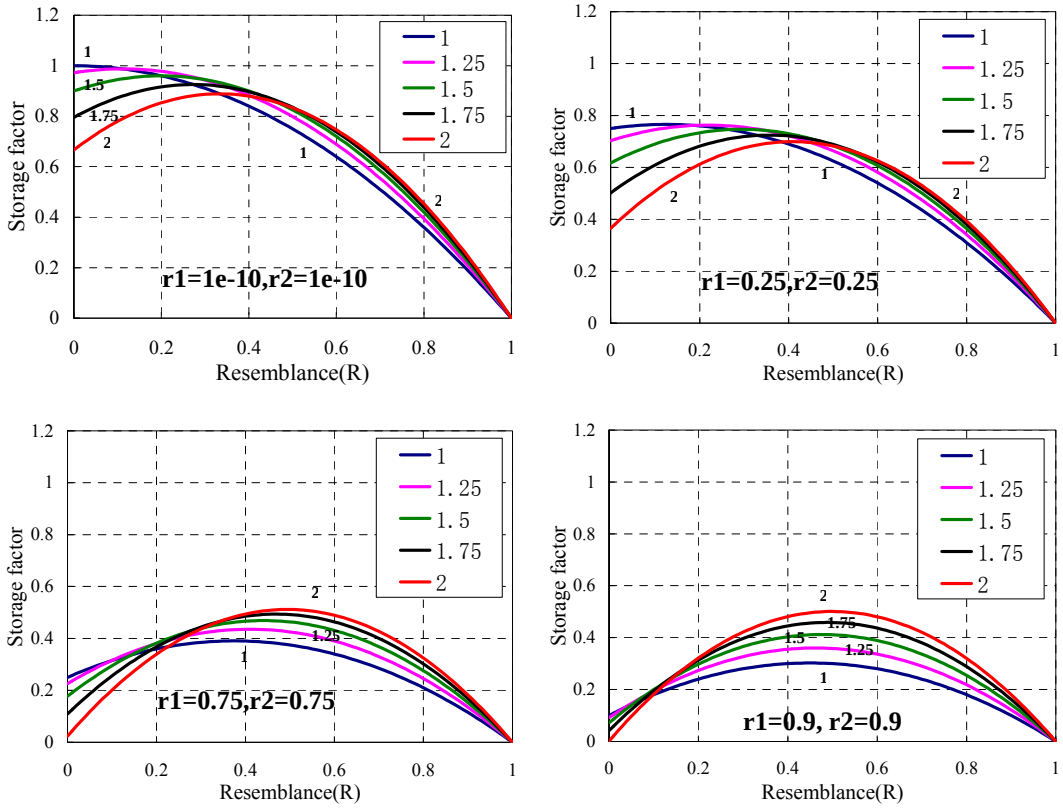


Figure 2. *f*-fractional bit *R-Storage factor*

IV F-FRACTIONAL BIT LESS THAN 1

A Unbiased Estimator of 1/2 bit

One easy method to implement $f<1$ is to combine the integer bit from two permutations. Formula (4) has proved that:

$$E_1 = \Pr(e_{1,1,\pi} = e_{2,1,\pi}) = C_{1,1} + (1-C_{2,1})R \qquad (21)$$

Define variable x_1, x_2:

$$x_1 = XOR(e_{1,1,\pi_1}, e_{1,1,\pi_2}), x_2 = XOR(e_{2,1,\pi_1}, e_{2,1,\pi_2}) \qquad (22)$$

Then x_1= x$_2$ either when $e_{1,1,\pi_1} = e_{2,1,\pi_2}$ and $e_{1,1,\pi_1} = e_{2,1,\pi_2}$ or, when $e_{1,1,\pi_1} \neq e_{2,1,\pi_2}$ and $e_{1,1,\pi_1} \neq e_{2,1,\pi_2}$. Thus

$$T_{1/2} = \Pr(x_1 = x_2) = E_1^2 + (1-E_1)^2 \qquad (23)$$

$$R = \frac{\sqrt{2T_{1/2}-1}+1-2C_{1,1}}{2-2C_{2,1}} \qquad (24)$$

We use $\hat{R}_{1/2}$ to indicate that two bits are combined into one.

$$\hat{R}_{1/2} = \frac{\sqrt{\max\{2\widehat{T_{1/2}}-1,0\}}+1-2C_{1,1}}{2-2C_{2,1}} \qquad (25)$$

The asymptotic variance of $\hat{R}_{1/2}$ can be derived using the "delta method" in statistics[28]:

$$\widehat{Var}(\phi) = \left(\frac{\partial\hat{\phi}}{\partial\hat{\theta}}\right)^2 \bullet \widehat{Var}(\theta) \qquad \text{(delta method)}$$

The variance of $\hat{R}_{1/2}$:

$$Var(\hat{R}_{1/2}) = \frac{1}{k}\frac{T_{1/2}(1-T_{1/2})}{4(1-C_{2,1})^2(2T_{1/2}-1)} + O(\frac{1}{k^2}) \qquad (26)$$

B Unbiased Estimator of 1/4,1/8,…,1/2^n

Define variable x_3, x_4:

$$x_3 = XOR(e_{1,1,\pi_3}, e_{1,1,\pi_4}), x_4 = XOR(e_{2,1,\pi_3}, e_{2,1,\pi_4})$$

Define variable m_1, m_2:

$$m_1 = XOR(x_1, x_3), m_2 = XOR(x_2, x_4)$$

Then $m_1 = m_2$ either when $x_1 = x_2$ and $x_3 = x_4$ or, when $x_1 \neq x_2$ and $x_3 \neq x_4$. Thus

$$T_{1/4} = \Pr(m_1 = m_2) = T_{1/2}^2 + (1-T_{1/2})^2 \qquad (27)$$

$$T_{1/2} = \frac{\sqrt{2T_{1/4}-1}+1}{2} \qquad (28)$$

$$R=\frac{\sqrt{2T_{1/2}-1}+1-2C_{1,1}}{2-2C_{2,1}}=\frac{\sqrt{\sqrt{2T_{1/4}-1}}+1-C_{1,1}}{2-2C_{2,1}} \quad (29)$$

We will recommend the following estimator. $\hat{R}_{1/4}$:

$$\hat{R}_{1/4}=\frac{\sqrt{\sqrt{\max\{2\widehat{T_{1/4}}-1,0\}}+1-2C_{1,1}}}{2-2C_{2,1}} \quad (30)$$

$Var(\hat{R}_{1/4})$ could be derived using the "delta method" in statistics:

$$Var(\hat{R}_{1/4})=\frac{1}{k}\frac{T_{1/4}(1-T_{1/4})}{4(1-C_{2,1})^2\times 4(2T_{1/4}-1)^{\frac{3}{2}}}+O(\frac{1}{k^2}) \quad (31)$$

Similarity, $R_{1/2^n}, Var(R_{1/2^n})$ can be derived :

$$\hat{R}_{1/2^n}=\frac{[\max\{2\widehat{T_{1/2^n}}-1,0\}]^{\frac{1}{2^n}}+1-2C_{1,1}}{2-2C_{2,1}} \quad (32)$$

$$\begin{aligned}&Var(\hat{R}_{1/2^n})\\&=\frac{1}{k}\frac{\frac{1}{2^{2(n-1)}}\left(2T_{1/2^n}-1\right)^{\frac{1-2^n}{2^{n-1}}}T_{1/2^n}(1-T_{1/2^n})}{4(1-C_{2,1})^2}+O(\frac{1}{k^2})\\&=\frac{1}{k}\frac{T_{1/2^n}(1-T_{1/2^n})}{2^{2n}(1-C_{2,1})^2\left(2T_{1/2^n}-1\right)^{\frac{2^n-1}{2^{n-1}}}}+O(\frac{1}{k^2})\end{aligned} \quad (33)$$

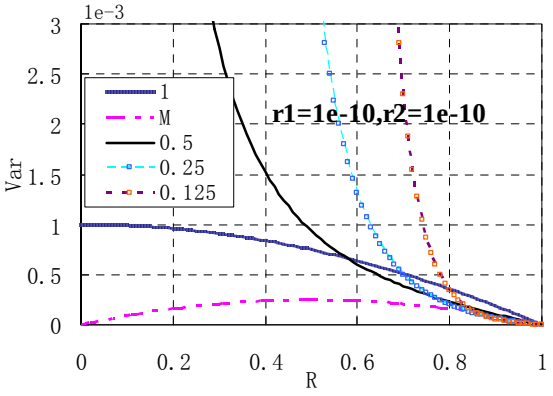


Figure 3. *f*-fractional bit (f<1) *R-Var*

Fig.3 plots *Var*(*f*) for the whole range of $R\in(0, 1)$, $f<1$ and selected $r_1= r_2 =10^{-10}$. Fig.5 shows that $\hat{R}_{1/2}, \hat{R}_{1/4}, \hat{R}_{1/8}$ is not accurate, almost for the $R<0.8$. However, when R is close to 1, $\hat{R}_{1/2}, \hat{R}_{1/4}, \hat{R}_{1/8}$ is better than $\hat{R}_1$. Common application (for example Duplicated Web pages Removal) focus on the situation when R is big other than R is small.

What's more interesting, when $R\to 1$, $Var(\hat{R}_1)$ is twice of $Var(\hat{R}_{1/2})$.(proved in appendix 1):

$$\begin{aligned}&\lim_{R\to 1}\frac{Var(\hat{R}_1)}{Var(\hat{R}_{1/2^n})}\\&=\lim_{R\to 1}\frac{Var(\hat{R}_1)}{Var(\hat{R}_{1/2})}\frac{Var(\hat{R}_{1/2})}{Var(\hat{R}_{1/4})}\cdots\frac{Var(\hat{R}_{1/2^{n-1}})}{Var(\hat{R}_{1/2^n})}\\&=\underbrace{2\times 2\times 2\ldots\times 2}_{n}=2^n\end{aligned} \quad (34)$$

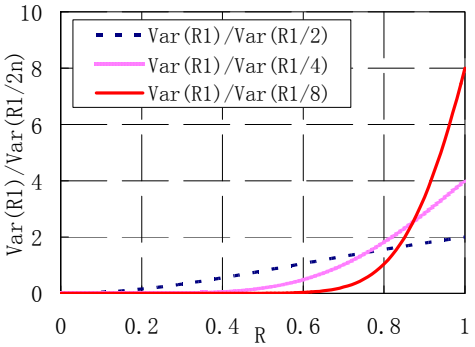


Figure 4. *R-Var*(*b*=1)/*Var*($1/2^n$)

Fig.4 shows that when $R\to 1$, $Var(\hat{R}_1)/Var(\hat{R}_{1/2})\to 2$, $Var(\hat{R}_1)/Var(\hat{R}_{1/4})\to 4$, $Var(\hat{R}_1)/Var(\hat{R}_{1/8})\to 8$, verifying formula (31).

C Unbiased Estimator of 3/4 bit

Set $w_1=1/2, w_2=1/2, b_1=1/2, b_2=1$

Define variable v_1, v_2:

$$v_1=\begin{cases}x_1 & \text{when } b_x=b_1\\ e_{1,1} & \text{when } b_x=b_2\end{cases}$$
$$v_2=\begin{cases}x_2 & \text{when } b_x=b_1\\ e_{2,1} & \text{when } b_x=b_2\end{cases} \quad (35)$$

Thus we can derive:

$$\begin{aligned}T_{3/4}&=\Pr(v_1=v_2)\\&=\Pr(b_x=b_1)T_{1/2}+\Pr(b_x=b_2)E_1\\&=w_1T_{1/2}+w_2E_1\\&=\frac{1}{2}(E_1^2+(1-E_1)^2+E_1)\\&=\frac{1}{2}(2E_1^2+1-E_1)\end{aligned} \quad (36)$$

(36) is equal to (37).

$$E_1=\frac{\sqrt{16T_{3/4}-7}+1}{4} \quad (37)$$

$$\hat{R}_{3/4}=\frac{\sqrt{16\widehat{T_{3/4}}-7}+1-4C_{1,1}}{4-4C_{2,1}} \quad (38)$$

V Expanded Fraction Bit Hashing

The nature of fractional bit hashing provides a wide range of selectivity for accuracy and storage space complexity. Through deployment of w_1, w_2, ..., w_n, we

construct w_1, w_2 expand to w_1, w_2, ..., w_n to get the expanded fractional bit hashing algorithm.

Let $w_1 = \frac{k_1}{k}$, $w_2 = \frac{k_2}{k}$, ... , $w_n = \frac{k_n}{k}$

Define Extended fraction f:

$$f = w_1 b_1 + w_2 b_2 + ... + w_n b_n \tag{39}$$

where

$w_1 + w_2 + ... + w_n = 1$, $b_i \neq b_j$, $i \neq j$, $i, j \in \{1, 2, ...n\}$

$$E_f = \sum_{i=1}^{n} w_i (C_{1,b_i} + (1 - C_{2,b_i})R) \tag{40}$$

An unbiased estimator of R denoted by $\hat{R}_f$:

$$\hat{R}_f = \frac{\hat{E}_f - \sum_{i=1}^{n} w_i C_{1,b_i}}{1 - \sum_{i=1}^{n} w_i C_{2,b_i}} \tag{41}$$

$$Var(\hat{R}_f) = \frac{1}{k} \frac{\sum_{i=1}^{n} w_i^2 E_{b_i}(1 - E_{b_i})}{[1 - \sum_{i=1}^{n} w_i C_{2,b_i}]^2} \tag{42}$$

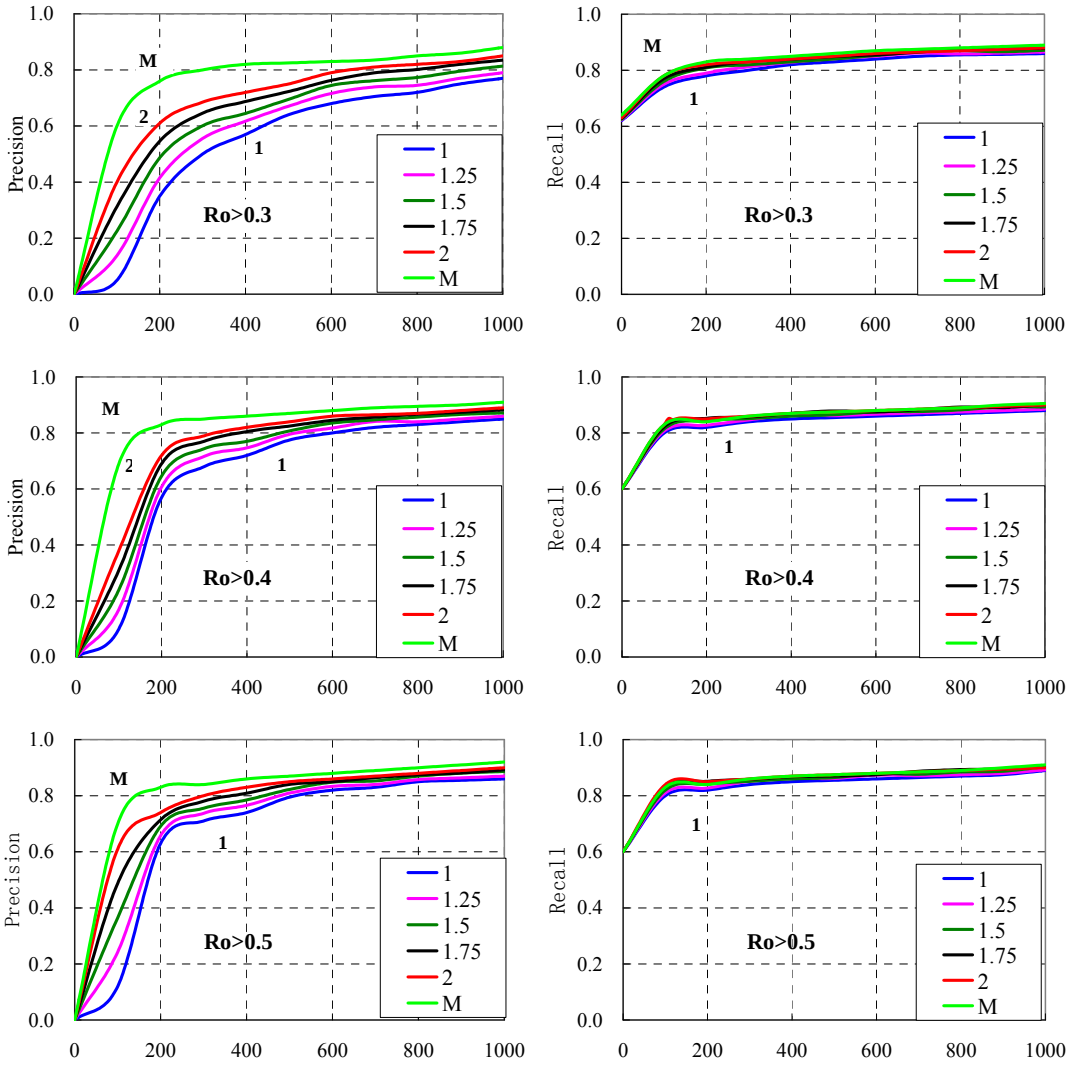
where

$$E_{b_i} = C_{1,b_i} + (1 - C_{2,b_i})R \qquad i \in \{1, 2, ..., n\} \tag{43}$$

VI Experiment Results and Analysis

Taking some funds applied projects as data source, we conducted f-fractional bit similarity measurements for 36889 project text of WORD format. In this section, we conduct experiments to calculate precision and recall for pairs whose resemblance values$\geq R_0$. We estimate the resemblances using $\hat{R}_f$ (f=1.25, 1.5, 1.75), $\hat{R}_b$ (b = 1, 2) and the original minwise hashing $\hat{R}_M$ (using 32 bits). Fig.5 presents the precision and recall curves at different values of thresholds R_0 and sample sizes k.

The fig.5 shows the experiment results of fractional bit hashing algorithm. We can see the precision and the recall of f-fraction bit minwise hashing is between b=1 and b=2. This experiment confirms the effectiveness of the proposed f-fractional hashing algorithm.



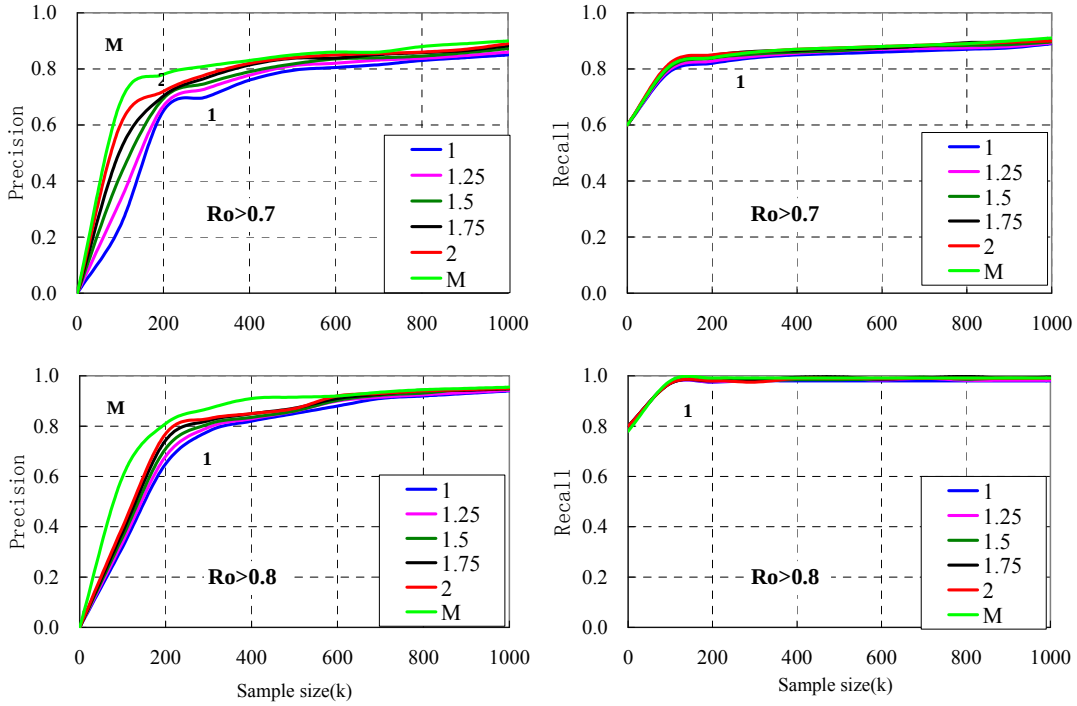


Figure 5. The precision and recall of valuation

VII Conclusion

The min-wise hashing algorithm is widely used in the information retrieval of mass data. The Pingli's *b*-bit Minwise hashing algorithm reduces *b*-bit from 64(or 32 bit) to 1 bit and saves storage space and computing time. We theoretically analyze the feasibility of *f*-fractional bit hashing algorithm, give the unbiased estimator of *f*-fractional and compute the storage factor of *f*-fractional bit to demonstrate variance-space trade-off. Addtionally, the precision and recall of *f*-fractional hashing are measured. The results show that precision and recall of *f*-fractional bit Minwise hashing is between those values achieved by two adjacent integer bits. This verifies the effectiveness of f-fractional Minwise hashing algorithm.

The proposed method is easy to implement and could satisfy a wider range of precision and storage space trade-offs.

Appendix A Proof of $\lim_{R\to 1}\frac{Var(\hat{R}_1)}{Var(\hat{R}_{1/2^n})}=2^n$

$$\frac{Var(\hat{R}_{1/2^{n-1}})}{Var(\hat{R}_{1/2^n})}=\frac{\frac{1}{k}\frac{T_{1/2^{n-1}}(1-T_{1/2^{n-1}})}{2^{2n-2}(1-C_{2,1})^2\left(2T_{1/2^{n-1}}-1\right)^{\frac{2^{n-1}-1}{2^{n-2}}}}}{\frac{1}{k}\frac{T_{1/2^n}(1-T_{1/2^n})}{2^{2n}(1-C_{2,1})^2\left(2T_{1/2^n}-1\right)^{\frac{2^n-1}{2^{n-1}}}}}$$

$$=\frac{4\frac{T_{1/2^{n-1}}(1-T_{1/2^{n-1}})}{\left(2T_{1/2^{n-1}}-1\right)^{\frac{2^{n-1}-1}{2^{n-2}}}}}{\frac{T_{1/2^n}(1-T_{1/2^n})}{\left(2T_{1/2^n}-1\right)^{\frac{2^n-1}{2^{n-1}}}}}$$

Let $t=T_{1/2^{n-1}}$

Then we can drived:

$$\begin{aligned}T_{1/2^n}&=T_{1/2^{n-1}}{}^2+(1-T_{1/2^{n-1}})^2\\&=2T_{1/2^{n-1}}{}^2-2T_{1/2^{n-1}}+1\\&=2t^2-2t+1\end{aligned}$$

$\lim_{R\to 1}\frac{Var(\hat{R}_{1/2^{n-1}})}{Var(\hat{R}_{1/2^n})}$ could be demonstrated by *t*:

$$\lim_{R\to 1}\frac{Var(\hat{R}_{1/2^{n-1}})}{Var(\hat{R}_{1/2^n})}=\lim_{t\to 1}\frac{4\frac{t(1-t)}{(2t-1)^{\frac{2^{n-1}-1}{2^{n-2}}}}}{\frac{(2t^2-2t+1)(1-(2t^2-2t+1))}{\left(2(2t^2-2t+1)-1\right)^{\frac{2^n-1}{2^{n-1}}}}}$$

$$=\lim_{t\to 1}\frac{\frac{2}{(2t-1)^{\frac{2^{n-1}-1}{2^{n-2}}}}}{\frac{(2t^2-2t+1)}{\left(2(2t^2-2t+1)-1\right)^{\frac{2^n-1}{2^{n-1}}}}}=2$$

Therefore, we can obtain the desired results:

$$\lim_{R\to 1}\frac{Var(\hat{R}_1)}{Var(\hat{R}_{1/2^n})}=\lim_{R\to 1}\frac{Var(\hat{R}_1)}{Var(\hat{R}_{1/2})}\frac{Var(\hat{R}_{1/2})}{Var(\hat{R}_{1/4})}\cdots\frac{Var(\hat{R}_{1/2^{n-1}})}{Var(\hat{R}_{1/2^n})}$$

$$=\underbrace{2\times 2\times 2\ldots\times 2}_{n}=2^n$$

ACKNOWLEDGEMENTS

We are grateful to the support of the National Natural Science Foundation of China (Grant No. M0921005, 60873081, 61003033, 61073105 and No. 60970095).
We would like to thank the anonymous referees for their helpful comments and suggestions.

REFERENCES

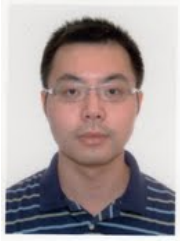
[1] T. B. Lee, *et al.*, "The World-Wide Web," *Commun. ACM,* vol. 37, pp. 76-82, 1994.
[2] U. Manber, "Finding similar files in a large file system," in *Proceedings of the Winter USENIX Conference*, pp. 1~10, 1994.
[3] S. Brin, *et al.*, "Copy detection mechanisms for digital documents," in *Proceedings of the ACM SIGMOD Annual Conference*, 1995.
[4] N. Shivakumar and H. Garcia-Molina, "SCAM: A copy detection mechanism for digital documents," in *Proceedings of the 2nd International Conference in Theory and Practice of Digital Libraries (DL'95)*, 1995.
[5] N. Shivakumar and H. Garcia-Molina, "Building a scalable and accurate copy detection mechanism," in *Proceedings of the 1st ACM Conference on Digital Libraries (DL'96)*, 1996.
[6] G. Salton, "The state of retrieval system evaluation," *Information processing & management,* vol. 28, pp. 441-449, 1992.
[7] H. Garcia-Molina, *et al.*, "dSCAM: Finding document copies across multiple databases," in *Proceedings of the 4th International Conference on Parallel and Distributed Systems (PDIS'96)*, 1996.
[8] K. Monostori, *et al.*, "Parallel and distributed overlap detection on the Web," in *Proceedings of the Workshop on Applied Parallel Computing (PARA2000)*, 2000.
[9] K. Monostori, *et al.*, "Document overlap detection system for distributed digital libraries," in *Proceedings of the ACM Digital Libraries 2000 (DL2000)*, 2000.
[10] K. Monostori, *et al.*, "MatchDetectReveal: Finding overlapping and similar digital documents," in *Proceedings of the Information Resources Management Association International Conference (IRMA2000)*, 2000.
[11] K. Monostori, *et al.*, "Parallel overlap and similarity detection in semi- structured document collections " in *Parallel overlap and similarity detection in semi-structured document collections* 1999.
[12] K. Monostori, *et al.*, "Suffix vector: A space-efficient suffix tree representation," *Algorithms and Computation,* pp. 707-718, 2001.
[13] Bao JP, Shen JY, Liu XD, *et al.*, " A survey on natural language text copy detection.," *Journal of Software,* vol. 14, pp. 1753-1760, 2003.
[14] A. Z. Broder, *et al.*, "Min-wise independent permutations," *Journal of Computer and System Sciences,* vol. 60, pp. 327-336, 1998.
[15] A. Z. Broder, "Identifying and filtering near-duplicate documents," in *11th Annual Symposium on Combinatorial Pattern Matching (CPM 2000)*, pp. 1-10, 2000.
[16] A. Z. Broder, "On the resemblance and containment of documents," in *In the Compression and Complexity of Sequences*, pp. 21-29, 1997.
[17] A. Z. Broder, *et al.*, "Syntactic clustering of the web," *Computer Networks and ISDN Systems,* vol. 29, pp. 1157-1166, 1997.
[18] K. Kalpakis and S. Tang, "Collaborative data gathering in wireless sensor networks using measurement co-occurrence," *Computer Communications,* vol. 31, pp. 1979-1992, 2008.
[19] Y. Dourisboure, *et al.*, "Extraction and classification of dense implicit communities in the Web graph," *ACM Transactions on the Web (TWEB),* vol. 3, pp. 1-36, 2009.
[20] M. Bendersky and W. B. Croft, "Finding text reuse on the web," in *Proceedings of WSDM '09 Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pp. 262-271,2009.
[21] G. Buehrer and K. Chellapilla, "A scalable pattern mining approach to web graph compression with communities," in *WSDM*, pp. 95-106,2008.
[22] P. Indyk, "A small approximately min-wise independent family of hash functions," in *Proceeding of SODA '99 Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 454-456,1999.
[23] M. S. Charikar, "Similarity estimation techniques from rounding algorithms,"in *Proceeding STOC '02 Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pp. 380-388, 2002.
[24] T. Itoh, *et al.*, "On the sample size of k-restricted min-wise independent permutations and other k-wise distributions," in *STOC*, San Diego, CA, 2003, pp. 710-719.
[25] E. Kaplan, *et al.*, "Derandomized constructions of k-wise (almost) independent permutations," *Algorithmica,* vol. 55, pp. 113-133, 2009.
[26] L. Ping, *et al.*, "One sketch for all: theory and application of conditional random sampling," in *Advances in Neural Information Processing Systems*, Vancouver, BC, Canada, 2009.
[27] L. Ping and C. Konig, "b-Bit minwise hashing," in *Proceedings of the 19th international conference on World wide web*, pp. 671-680, 2010.
[28] A. C. Davison, *Statistical models*: Cambridge Univ Pr, 2003.

Xinpan Yuan received the M.E. degree in Information Science and Engineering, Central South University in 2008, And he is pursuing his Ph.D degree in School of Information Science and Engineering, Central South University, ChangSha, China. His current research interests include information retrieval.

Jun Long received the M.E. degree in Information Science and Engineering, Central South University in 2005, he is now associate professor in School of Information Science and Engineering, Central South University, ChangSha, China, His research interests focus on network resource management and service.

Zuping Zhang received the Ph.D. degree in Information Science and Engineering, Central South University in 2005, He is now a Professor in School of Information Science and Engineering, Central South University, ChangSha, China. His current research interests include image processing, visual perception.

Yueyi Luo received the B.S. in Maths Science Central South University in 2005. He is pursuing the M.E. degree in School of Information Science and Engineering, Central South University, ChangSha, China. His current research interests include data mining.



Hao Zhang received the Ph.D. degree in electrical and computer engineering from Polytechnic University, New York, USA, in 2006. He is currently an associate professor in the School of Information Science and Technology at Central South University, ChangSha, China. His current research interests include evaluation system analysis and video coding.

Weihua Gui professor, PhD supervisor. His research interests focus on intelligent control of metallurgical processes.