JCSE Journal of Computing Science and Engineering

Regular Paper
Journal of Computing Science and Engineering,
Vol. 13, No. 2, June 2019, pp. 78-88

Near-Duplication Document Detection Using Weight One Permutation Hashing

Xinpan Yuan, Songlin Wang, and Xiaojun Deng*
School of Computer Science, Hunan University of Technology, Zhuzhou, China
xpyuan@hut.edu.cn, 2561838940@qq.com, WangSL110@163.com

Abstract

As a standard algorithm for efficiently calculating set similarity, Minwise hashing is widely used to detect text similarity. The major drawback associated with Minwise hashing is expensive preprocessing. One permutation hashing (OPH) is proposed in order to reduce the number of random permutations. OPH divides the space Ω evenly into k bins, and selects the smallest nonzero value in each bin to re-index the selected elements. We propose a weight one permutation hashing (WOPH) by dividing the entire space Ω into k_1 and k_2 bins and sampling k_1 and k_2 in proportion to form a weighted k_w. WOPH has a wider range of precision by expanding the proportion of w_1 and w_2 to different accuracy levels of the user. The variance of WOPH can be rapidly decreased first and then slowly decreased, although the final variance is the same as OPH with the same k. We combined the dynamic double filter with WOPH to reduce the calculation time by eliminating unnecessary comparison in advance. For example, for a large number of real data with low similarity accompanied by high threshold queries, the filter reduces the comparison of WOPH by 85%.

Category: Databases / Data Mining

Keywords: One Permutation Hashing; WOPH; Bin; Non-uniform partition; Weighting; Dynamic double filter

I. INTRODUCTION

With the development of computer and Internet technologies, as well as the arrival of big data, information has been increasingly digitized and electronic, which facilitates communication. However, it also increases the risk of copying, plagiarism and duplicating others' academic achievements. The use of illegal means to plagiarize academic results of others has seriously damaged the intellectual property rights of experts, and casts a shadow over the fairness and justice of the academic community. Text similarity detection technology is an effective means to protect the intellectual property rights of digital products. It is widely used in search engines [1-3], anti-spam [4, 5], anti-academic [6] results in plagiarism [7], digital libraries [8-10], and so on.

Traditionally, when comparing two texts for similarities, most of them are converted into a feature vector of the texts to determine the similarity after text segmentation. The commonly used text similarity measurements utilize Euclidean distance [11-13], editing distance [14], cosine similarity [15], and Jaccard coefficient [16-19]. These algorithms are inefficient and the accuracy of detection is not high. Therefore, they are only appropriate for short text or a relatively small amount of data, and cannot be extended to similarity detection of massive data and long text. In the face of similarity measurement of massive text data, most scholars generate k hash codes or fingerprints

Open Access http://dx.doi.org/10.5626/JCSE.2019.13.2.78 http://jcse.kiise.org

Received 22 April 2019; **Accepted** 04 June 2019
*Corresponding Author

 pISSN: 1976-4677 eISSN: 2093-8020

from k independent sample outputs, and estimate the similarity between texts by calculating an equal number of fingerprints. This type of algorithm is collectively referred to as hash similarity measurement.

A. Minwise Hashing

Minwise hashing [20] (or Minhash) is a locally sensitive hash [21] and is considered to be the most popular similarity estimation method [22]. Minhash algorithm is mainly used to estimate the similarity of two sets. It is a standard technology used to calculate the similarity of sets. The Jaccard similarity coefficient is used as the theoretical value for similarity calculation. Minhash algorithm is characterized by rapid calculation and simple fingerprint generation. It is widely used in the fields of web page duplication [23-25], text similarity detection [26], wireless sensor networks [27], network community classification [28], text reuse [29-31], connection graph compression [32], and so on. Therefore, the algorithm also involves a considerable number of theoretical and experimental methods of innovation and development [33-35]. When Minhash algorithm detects the similarity of two document sets, it generates k feature values (fingerprints) via k times of random permutation and then compares the equal number of feature values, and finally estimates the similarity of the two document sets.

The Minhash algorithm is calculated as follows: Let the full set $\Omega = \{0, 1, ..., D\text{-}1\}$ determine the related shingles set S_d by shingling the document d. Given the shingles sets S_1 and S_2 corresponding to the two documents d_1 and d_2, the similarity $R(d_1, d_2)$ of documents d_1 and d_2 is defined as $R(d_1, d_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} = \frac{a}{f_1 + f_2 - a}$, $f_1 = |S_1|$, $f_2 = |S_2|$, $a = |S_1 \cap S_2|$. Calculation of the similarity of two documents is essential for the calculation of the intersection of two shingles sets. Suppose a random independent permutation group on Ω:

$\pi : \Omega \rightarrow \Omega$, $\Omega = \{0, 1, ..., D\text{-}1\}$, the estimation formula of $R\ (d_1, d_2)$ is as follows:

$$\Pr(\min\{\pi(S_1)\} = \min\{\pi(S_2)\}) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} = R(d_1,\ d_2) \tag{1}$$

Based on k random independent permutation groups π_1, π_2, ..., π_n, the shingles set of any document d is transformed into the following equation:

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

The unbiased estimate of R for Minhash is as follows:

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

The variance is obtained as follows:

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

where k represents the number of experiments (or sample size).

In order to achieve high accuracy of text similarity, the number of fingerprints k must be sufficient, generally assuming $k = 500$. Normally, the Minhash algorithm can only produce a single feature value [5] at a time, that is, when the number of fingerprints is $k \geq 500$, k times of random permutation is needed. Thus, when the similarity of massive documents is detected, we spend a lot of time on the random replacement. For example, when the number of documents to be detected is 1 million and the number of fingerprints is $k = 500$, the number of random permutations throughout the detection process is 500 million.

The b-bit Minwise hashing method [36] provides a simple solution by storing only the lowest b-bit of each hashed data [37]. In this way, the dimension of the extended data matrix from the hashed data is only $2^b \times k$. The b-bit Minwise hashing is widely used in sublinear time near-neighbor [38] and linear learning processes [39]. The major drawback of Minhash and b-bit Minwise hashing methods is that they require expensive preprocessing involving k (e.g., 200 to 500) permutations of the entire dataset [37].

B. One Permutation Hashing

Intuitively, Minhash's standard practice should be very "wasteful" because all non-zero elements in a group are scanned (replaced), but only the smallest elements are used [40]. In order to reduce the number of random permutations of the Minhash algorithm, Li et al. [40] proposed the one permutation hashing algorithm, referred to as OPH. The algorithm can generate k fingerprint values with only one random permutation, and the similarity of the document set can be estimated using these k fingerprint values. OPH reduces the number of traditional Minhash permutations from $k = 500$ to 1, which greatly reduces the time consumption of Minhash algorithm in random permutation, and at the same time ensures that the accuracy is basically unchanged or even slightly better. The specific algorithm process is as follows:

Suppose that the random permutation sequences generated by two sets S_1, S_2 after a random permutation are $\pi(S_1)$ and $\pi(S_2)$, respectively. Examples of the specific forms of random permutation sequences $\pi(S_1)$ and $\pi(S_2)$ are provided in Table 1.

The space Ω is evenly divided into k bins, and a minimum non-zero element is selected in each region as the fingerprint generated by sampling if a bin is empty, that is, if there is no non-zero element in the region, the *

Table 1. Examples of $\pi(S_1)$ and $\pi(S_2)$

	1				2				3				4			
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\pi(S_1)$	0	0	1	0	1	0	0	1	0	0	0	0	0	1	0	0
$\pi(S_2)$	1	0	0	1	0	0	1	0	0	0	0	0	1	0	0	0

is used as the fingerprint generated by a sampling. For example, let the random sequences generated by a random permutation of sets S_1 and S_2 be $\pi(S_1)$ = {2,4,7,13} and $\pi(S_2)$ = {0,3,6,12}, respectively. If $k = 4$, select a minimum non-zero element in each bin as the hash value generated by a sample. Therefore, the fingerprints produced by S_1 and S_2 by OPH algorithm are [2-4×0, 4-4×1, *, 13-4×3] = [2, 0, *, 1] and [0-4×0, 6-4×1, *, 12-4×3] = [0, 2, *, 0], respectively.

The OPH defines N_{emp} for the number of bins that are empty in both sets, and N_{mat} for the number of bins that are not empty and have equal fingerprint values in both sets, as follows:

$$N_{emp} = \sum_{i=1}^{K} I_{emp,\,j}\,,\ N_{mat} = \sum_{i=1}^{K} I_{mat,\,j}\,, \tag{4}$$

where I_i represents the i^{th} bin, $I_{mat,i}$ and $I_{emp,i}$ are defined as follows:

$$I_{mat,\,j} = \begin{cases} 1,\ if \min(\pi(S_1)) = \min(\pi(S_2)) \neq *in\ the\ j^{th}\ bin \\ 0,\ otherwise \end{cases}, \tag{5}$$

$$I_{emp,\,j} = \begin{cases} 1,\ if\ \pi(S_1) = \pi(S_2) = *in\ the\ j^{th}\ bin \\ 0,\ otherwise \end{cases} \tag{6}$$

The unbiased estimator of OPH is obtained as follows:

$$\hat{R} = \frac{N_{mat}}{K - N_{emp}}\,. \tag{7}$$

The variance is derived as follows:

$$Var(R) = R(1-R)\left(E\left(\frac{1}{K-N_{emp}}\right)\left(1+\frac{1}{f-1}\right)-\frac{1}{f-1}\right). \tag{8}$$

Compared with Minhash, the number of random permutations k of OPH are greatly reduced when the same number of fingerprints are generated, which greatly reduces the time of the random replacement and improves the efficiency of the sampling algorithm. However, when measuring text similarity, OPH must perform a complete eigenvalue comparison. When the text is large, the complete comparison of the feature value of the entire text will entail significant computational cost.

C. Our Proposal: Weight One Permutation Hashing

The main idea of weight one permutation hashing (WOPH) is to adopt non-uniform partition space Ω to form a weighted k_w. Specific practices are as follows: The entire space Ω is evenly divided into k_1 and k_2 bins in advance, and k_1 and k_2 sampled in proportion to form a weighted k_w. The values of $N_{mat}w$ and $N_{emp}w$ are counted in k_w, and the similarity R_w is finally calculated. Changes in WOPH show a wide range of accuracy, thus, combining the dynamic double filter with WOPH to reduce the calculation time by terminating unnecessary comparison in advance.

In this paper, our main contributions are as follows:

1) In this paper, we innovatively propose WOPH by adopting non-uniform partitioning space to form a weighted k_w.
2) Under the premise that the experimental results prove that the WOPH algorithm can achieve a wide range of precision and the accuracy of calculation is almost consistent with OPH, the WOPH can improve the efficiency of calculation by setting the appropriate threshold during similarity comparison.

The rest of the paper is organized as follows: Section II discusses the theoretical derivation of WOPH similarity calculation. Section III describes the steps to calculate the similarity of WOPH. Section IV discusses and analyzes the variance measurement experiments of WOPH and OPH involving 9 pairs of documents, and the variance changes of WOPH. Section V mainly suggests that WOPH greatly improves the computational efficiency in practical applications by combining dynamic double filter. Finally, Section VI provides conclusions.

II. THEORETICAL ANALYSIS OF WEIGHT ONE PERMUTATION HASHING

Suppose that the random sequence of the set $S \subseteq \Omega$ produced by a random permutation is $\pi(S)$, OPH divides the space evenly into k_1 and k_2 bins. Considering the bins with the ratio t_1 $(0 < t_1 < 1)$ from k_1; assuming the bins with the ratio t_2 $(0 < t_2 < 1)$ from k_2, where $t_1 + t_2 = 1$, the following equation can be generated:

$$kw = t_1 \cdot k_1 + t_2 \cdot k_2 \tag{9}$$

The new bin is defined as k_w based on proportional sampling from k_1 and k_2, where the weights w_1 and w_2 are as follows:

$$w_1 = \frac{t_1 \cdot k_1}{kw},\ w_2 = \frac{t_2 \cdot k_2}{kw},\ w_1 + w_2 = 1 \tag{10}$$

The construction diagram of the k_w is as follows:

Lemma 1. Estimators of weight one permutation hashing.

$$\hat{R}w = \frac{N_{mat}w}{kw - N_{emp}w} \tag{11}$$

The proof process is as follows:

According to probability theory, the values of N_{mat1} and N_{mat2} can be obtained when $k = k_1$ and $k = k_2$.

$$Pr(I_{mat,j} = 1, j \in [1, kw]) = w_1 Pr(I_{mat,j} = 1, j \in [1, t_1 k_1])$$
$$+ w_2 Pr(I_{mat,j} = 1, j \in [t_1 k_1 + 1, t_1 k_1 + t_2 k_2])$$

where $Pr(I_{mat,j} = 1, j \in [1, kw]) = \frac{N_{mat}w}{kw}$ represents the probability of fingerprint matching in the entire III in Fig. 1.

$Pr(I_{mat,j} = 1, j \in [1, t_1 k_1]) = \frac{N_{mat1}}{k_1}$ represents the probability of matching the left portion of the fingerprint of the III in Fig. 1.

$Pr(I_{mat,j} = 1, j \in [t_1 k_1 + 1, t_1 k_1 + t_2 k_2]) = \frac{N_{mat2}}{k_2}$ represents the probability of matching the right portion of the fingerprint of the III in Fig. 1.

Therefore, it can be concluded as follows: $\frac{N_{mat}w}{kw} = w_1 \frac{N_{mat1}}{k_1} + w_2 \frac{N_{mat2}}{k_2}$.

Substituting w_1 and w_2 into the above formula, we obtain $\frac{N_{mat}w}{kw} = \frac{t_1 \cdot k_1}{kw} \cdot \frac{N_{mat1}}{k_1} + \frac{t_2 \cdot k_2}{kw} \cdot \frac{N_{mat2}}{k_2}$.

The value of $N_{mat}w$ can be calculated according to the previously set ratios t_1 and t_2, wherein,

$$N_{mat}w = t_1 \cdot N_{mat1} + t_2 \cdot N_{mat2} \tag{12}$$

Similarly, the value of $N_{emp}w$ can also be obtained according to the previously set ratios t_1 and t_2, wherein

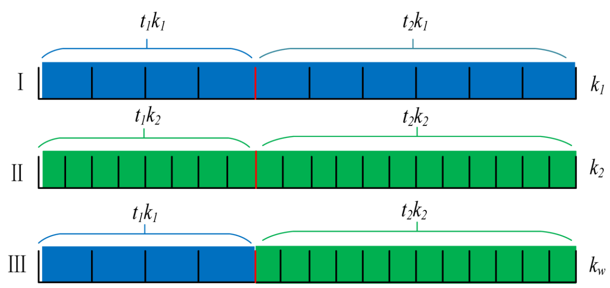

Fig. 1. Constitute a weighted bin k_w by the proportional sampling of k_1 and k_2.

$$N_{emp}w = t_1 \cdot N_{emp1} + t_2 \cdot N_{emp2} \tag{13}$$

Obtaining N_{mat1} and N_{mat2} from Eq. (7), $N_{mat1} = R(k_1 - N_{emp1})$, $N_{mat2} = R(k_2 - N_{emp2})$.

Combined with the formula (12), the following equation can be obtained:

$$R = \frac{N_{mat}w}{kw - (t_1 \cdot N_{emp1} + t_2 \cdot N_{emp2})}.$$

Combined with the formula (13), the unbiased estimator of R_w is as follows: $\hat{R}w = \frac{N_{mat}w}{kw - N_{emp}w}$, Lemma 1 is proved.

Based on the variance formula (8) of OPH, the variance of WOPH can be obtained as follows:

$$Var(\hat{R}_{mat}w) = R(1-R)\left(E\left(\frac{1}{N_{mat}w}\right)\left(1 + \frac{1}{f-1}\right) - \frac{1}{f-1}\right) \tag{14}$$

Lemma 2. If $k_1 \geq k_2$, has $N_{mat1} \geq N_{mat2}$.

Proof is as follows:

If $k_1 \geq k_2$, assume $k_1 = c \cdot k_2$; obviously, there is $c \geq 1$; at the same time, let $N_{mat1} = d \cdot N_{mat2}$, $d > 0$.

If $R = \frac{N_{mat1}}{k_1 - N_{emp1}} = \frac{N_{mat2}}{k_2 - N_{emp2}}$, the following equation was obtained:

$$N_{mat1} \cdot k_2 - N_{mat1} \cdot N_{emp2} = N_{mat2} \cdot k_1 - N_{mat2} \cdot N_{emp1},$$

that is:

$$N_{mat1} \cdot k_2 - N_{mat2} \cdot k_1 = N_{mat1} \cdot N_{emp2} - N_{mat2} \cdot N_{emp1}.$$

Substituting $k_1 = c \cdot k_2$ and $N_{emp1} = d \cdot N_{emp2}$ into the above equation yields the following:

$$N_{mat1} \cdot k_2 - N_{mat2} \cdot c \cdot k_2 = N_{mat1} \cdot N_{emp2} - N_{mat2} \cdot d \cdot N_{emp2}.$$

The following can be obtained by deformation of the above formula:

$k_2 \cdot (N_{mat1} - N_{mat2} \cdot c) = N_{emp2} \cdot (N_{mat1} - N_{mat2} \cdot d)$ and $\frac{N_{mat1} - N_{mat2} \cdot c}{N_{mat1} - N_{mat2} \cdot d} = \frac{N_{emp2}}{k_2}$.

If $k_2 \geq N_{emp2}$, there is $\frac{N_{emp2}}{k_2} \leq 1$.

That is, $\frac{N_{mat1} - N_{mat2} \cdot c}{N_{mat1} - N_{mat2} \cdot d} \leq 1$.

Assuming $c = x \cdot d$ and substituting it into the above formula to get: $\frac{N_{mat1} - N_{mat2} \cdot x \cdot d}{N_{mat1} - N_{mat2} \cdot d} \leq 1$ and $x \geq 1$.

If $R = \frac{N_{mat1}}{k_1 - N_{emp1}} = \frac{N_{mat2}}{k_2 - N_{emp2}}$, $\frac{N_{mat1}}{N_{mat2}} = \frac{k_1 - N_{emp1}}{k_2 - N_{emp2}}$.

And substituting $k_1 = c \cdot k_2$ and $N_{emp1} = d \cdot N_{emp2}$ into the above formula, we derived the following result:

$\frac{N_{mat1}}{N_{mat2}} = \frac{c \cdot k_2 - d \cdot N_{emp2}}{k_2 - N_{emp2}}$, in addition, $c = x \cdot d$.

Finally, $\frac{N_{mat1}}{N_{mat2}} = \frac{c \cdot k_2 - d \cdot N_{emp2}}{k_2 - N_{emp2}} = \frac{d \cdot (x \cdot k_2 - N_{emp2})}{k_2 - N_{emp2}}$, at the same time by $x \geq 1$, there is $\frac{N_{mat1}}{N_{mat2}} \geq d$.

However, if $d = \frac{N_{emp1}}{N_{emp2}}$, and it is known by OPH's Lemma 8:

$E(N_{emp}) = k \cdot \left(1 - \frac{1}{k}\right)^{f_1 + f_2 - a} = k \cdot \left(1 - \frac{1}{k}\right)^{f}$, $f >> 1$. Obviously, when $k_1 \geq k_2$, there is $N_{emp1} \geq N_{emp2}$ and $d \geq 1$, so there is $N_{mat1} \geq N_{mat2}$, thus Lemma 2 is proved.

Lemma 3. Let $k_1 \geq k_2$, there is $Var(R_{mat1}) \leq Var(R_{mat}w) \leq Var(R_{mat2})$.

The proof is as follows:

Formulas (8) and (11) are known to yield the following:

$$Var(\hat{R_{mat}w}) = R(1-R)\left(E\left(\frac{1}{N_{mat}w}\right)\left(1 + \frac{1}{f-1}\right) - \frac{1}{f-1}\right),$$

when $k_1 \geq k_2$, because $kw = t_1 \cdot k_1 + t_2 \cdot k_2$, $t_1 \cdot k_1 + t_2 \cdot k_2 = 1$, so $k_1 \geq kw \geq k_2$.

Based on Eq. (12), $N_{mat}w = t_1 \cdot N_{mat1} + t_2 \cdot N_{mat2}$ is known. According to Lemma 2, when $k_1 \geq k_2$, because of $N_{mat1} \geq N_{mat2}$, there is $N_{mat1} \geq N_{mat}w \geq N_{mat2}$.

Based on the derivation of the above formula, the $\frac{1}{N_{mat1}} \leq \frac{1}{N_{mat}w} \leq \frac{1}{N_{mat2}}$ is established, that is, $E\left(\frac{1}{N_{mat1}}\right) \leq E\left(\frac{1}{N_{mat}w}\right) \leq E\left(\frac{1}{N_{mat2}}\right)$, therefore, $Var(R_{mat1}) \leq Var(R_{mat}w) \leq Var(R_{mat2})$, and the proof of Lemma 3 is complete.

III. SPECIFIC CALCULATION OF WEIGHT ONE PERMUTATION HASHIN

When calculating the similarity between the sets S_1 and S_2, the WOPH algorithm first divides the whole set Ω into t_1 to t_m in proportion, and evenly divides into k_i bins of t_i each, as shown in Fig. 2.

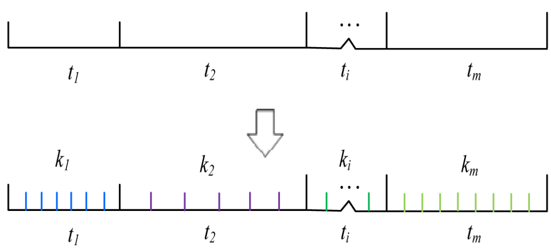

Fig. 2. Bin partition diagram.

The $N_{mat}w$ and $N_{emp}w$ values of the random permutation sequences $\pi(S_1)$ and $\pi(S_2)$ are respectively counted in each bin, and the value of R_w is calculated according to formula (8). The specific division is shown in Fig. 2.

WOPH is shown as in Algorithm 1.

Algorithm 1. Weight one permutation hashing

Input: The set S_1 and S_2, $S_1, S_2 \in \Omega = \{0, 1, ..., D-1\}$, $t = \{t_1, t_2, ..., t_m\}$ and $k = \{k_1, k_2, ..., k_m\}$.

Output: R_w

1: Generate a random permutation function π: $\Omega \to \Omega$;

2: According to the permutation function π, the random permutation sequences of the sets S_1 and S_2 are respectively $\pi(S_1)$, $\pi(S_2)$;

3: Divide the whole set Ω into t_1 to t_m parts in proportion, and then evenly divide into k_i bins in each t_i;

4: Count $N_{mat}w$ and $N_{emp}w$ values corresponding to $\pi(S_1)$ and $\pi(S_2)$ in each bin;

5: Estimate the similarity between S_1 and S_2 based on

$Rw = \frac{N_{mat}w}{kw - N_{emp}w}$, where $kw = \sum_{i=0}^{m} k_i$;

VI. EXPERIMENTAL RESULTS AND ANALYSIS

Corresponding to different k values, the variance of R_{mat} is obviously different and decreases with increasing k. In this part, we mainly carry out the following two experiments.

1) Section B is mainly designed to prove the Lemma 3: Experimental results prove that different values of k_w can be obtained by sampling different proportions of k_1 and k_2; if $k_1 \geq k_2$, there is $Var(R_{mat1}) \leq Var(R_{mat}w) \leq Var(R_{mat2})$.

2) Section C discusses the varying steepness of the decline in the variance of OPH and WOPH and treats k comparisons of hash values as a process. Section C demonstrates that when $k_w = k$, WOPH(k_w) and OPH(k) have the same variance after the similarity calculation ends, the curves showing variance decline of WOPH and OPH differ with increased k.

A. Experimental Datasets

We selected the 9 pairs of documents in the experimental data set in [37] to form the data set of this experiment. The document pairs were arranged into 9 groups according to the similarity from high to low, and a pair of words were randomly selected in each document pair to represent the pair of documents. The experimental data are presented in Table 2.

B. Variance Measure of WOPH

Obviously, in the OPH algorithm, if the number of bins

Table 2. Experimental datasets

Group No.	Word1	Word2	f_1	f_2	$f = f_1 + f_2 - a$	R
1	RIGHTS	RESERVED	12234	11272	12526	0.877
2	OF	AND	37339	36289	41572	0.771
3	ALL	MORE	26668	17909	31638	0.409
4	CONTACT	INFORMATION	16836	16339	24974	0.328
5	MAY	ONLY	2999	2697	4433	0.285
6	TOP	BUSINESS	9151	8284	14992	0.163
7	TIME	JOB	12386	3263	13874	0.128
8	REVIEW	PAPER	3197	1944	4769	0.078
9	A	TEST	39063	2278	2060	0.052

k divided by the entire space Ω is larger, that is, the smaller the width of each bin, the higher is the precision of calculation, and smaller is the variance. Once the number of bins k of the OPH is determined, the bin width is fixed. If the user needs to improve the accuracy, the value of k needs to be increased, that is, the entire space needs to be re-divided into more bins, which results in complete re-creation of the hash value of all documents.

In this experiment, three different WOPHs were constructed using k_1 = 1000 and k_2 = 10000, and the variances of OPH(k_1), OPH(k_2) and three WOPHs were measured separately. The WOPH(750:2500) indicates that the k_w is composed of k_1 = 1000 and k_2 = 10000 in proportion to t_1:t_2=3:1. The WOPH(500:5000) indicates that its k_w is composed of k_1 = 1000 and k_2 = 10000 in proportion to t_1:t_2=1:1. The WOPH (250:7500) indicates that its k_w is composed of k_1 = 1000 and k_2 = 10000 in proportion to t_1:t_2=1:3. A schematic diagram of the construction of WOPH(750:2500), WOPH(750:2500), and WOPH(750:2500) is shown in Fig. 3.

Using the above 9 pairs of documentation, the estimated $Var(R_{mat})$ for OPH(1000), OPH(10000), WOPH(750:2500), WOPH(500:5000), and WOPH(250:7500) was tested.

Fig. 3. Different kinds of WOPH are formed by two different types of OPH.

The experimental results are shown in Fig. 4, and the conclusions are as follows:

1) There is no doubt that the variance of OPH decreases with the increase of k. For example, the variance of OPH(10000) is smaller than that of OPH(1000).
2) The WOPH is composed of OPH(1000) and OPH(10000) in different weight proportions. With increased k, the variance of WOPH also decreases. For example, WOPH(250:7500) has the largest variance, followed by WOPH(500:5000), and WOPH(750:2500) has the smallest variance.
3) In all experimental datasets, the variances of OPH(1000), WOPH(750:2500), WOPH(750:2500), WOPH(750:2500), and OPH(10000) are in descending order. Thus, $Var(R_{mat1}) \leq Var(R_{mat}w) \leq Var(R_{mat2})$ is proved.

The experimental conclusion is that if users want to change the calculation precision, OPH is necessary to re-partition the bin, and cannot use the previous division. However, WOPH only needs two reusable bins of k_1 and k_2 to satisfy all types of user accuracy requirements.

C. Variance Change in The Process of Comparison

If the user's demand for k is fixed, but the variance of WOPH and OPH is the same under the same k, then what does WOPH suggest in this case? We consider the comparison in stages, that is, to determine the change in accuracy based on comparison time. Undoubtedly, as the number of comparisons increases, that is, k increases, the variance decreases. Therefore, we attempted to find the difference between the slopes of the variance curves of WOPH and OPH in the similarity comparison process.

Therefore, we choose to construct WOPH(500:5000), which is composed of k_1 = 1000 and k_2 = 10000 according to the proportion of t_1:t_2=1:1. WOPH(5000:500) is composed of k_2 = 10000 and k_1 = 1000 according to the

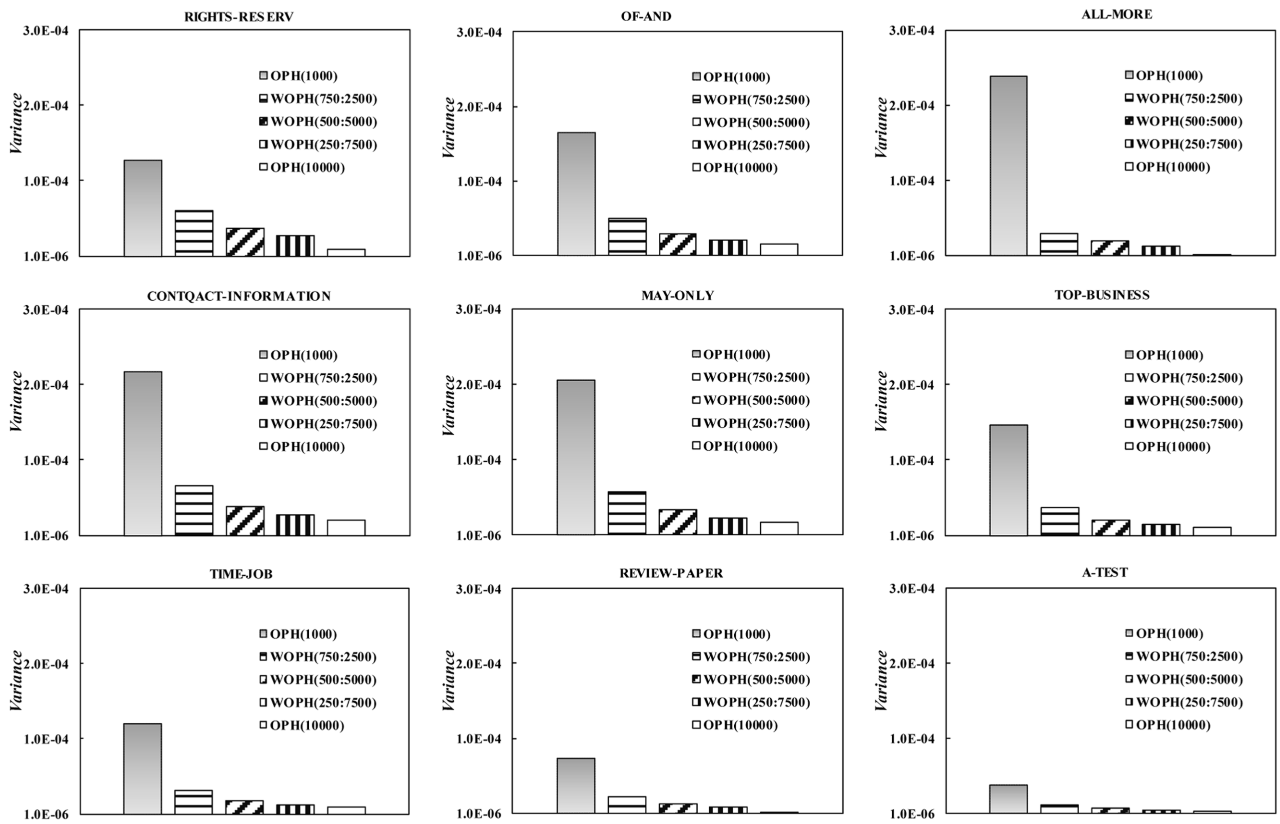

Fig. 4. Variance measurement curve of R_{mat}.

Fig. 5. Configuration scheme of WOPH(500:5000), WOPH(5000:500), and OPH(5500).

proportion of t_2:t_1=1:1. The larger the space length, the smaller the variance and the higher is the calculation accuracy. A schematic diagram of WOPH is shown in Fig. 5.

As shown in Fig. 6, the experimental results demonstrate the following conclusions:

1) At the final comparison point k = 500, WOPH and OPH show the same variance.
2) As shown in Fig. 5, at the same comparison point k = 500, WOPH(500:5000) covers the maximum length of space Ω; therefore, WOPH(500:5000) has the minimum variance of three curves, as shown in Fig. 6.
3) OPH represents a linear downward trend, and WOPH(500:5000) falls sharply first and then slowly. WOPH(5000:500) slowly decreases initially and sharply thereafter.

WOPH can quickly and flexibly form a variety of *kw* to meet different requirements of variance and calculation accuracy. At the same time, the variance curve can be quickly decreased and slowly decreased in similarity comparison. Therefore, WOPH can result in accurate changes in similarity comparison; however, the final precision and variance are the same as OPH.

Therefore, we combine the dynamic double filter with WOPH to obtain the results in advance without the need for complete similarity comparison.

V. APPLICATIONS

A. Document Clustering Pairs

The main function of document clustering is to form document pairs that may have a high degree of similarity.

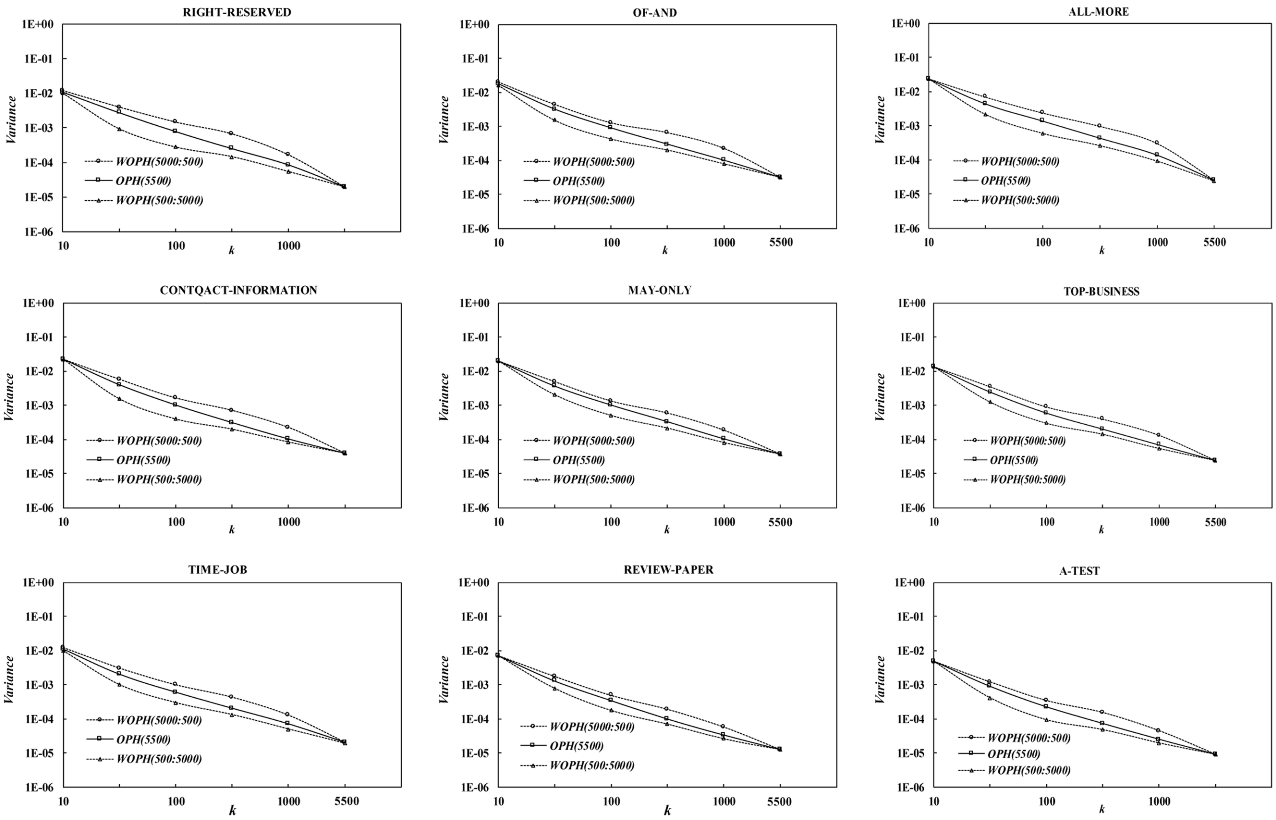

Fig. 6. Variance curve in the process of similarity comparison.

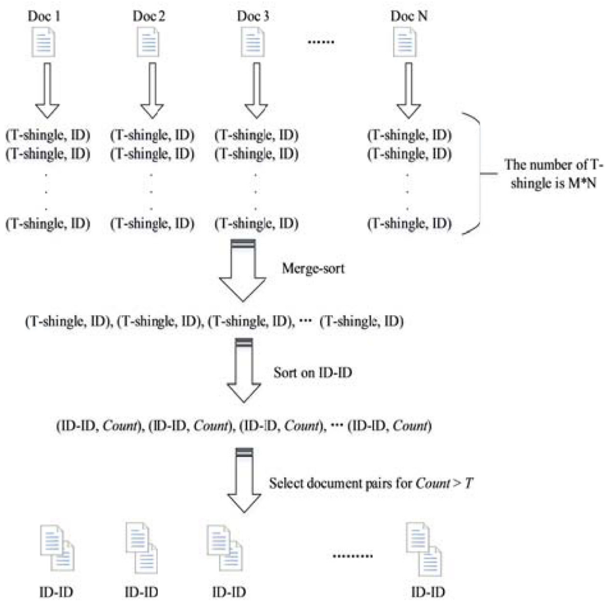

Fig. 7. The process of document clustering.

The document clustering process is shown in Fig. 7 and the main steps of document clustering are as follows:

Step 1. The title and keyword of each document are partitioned into shingles that differ from the body of the shingles and are named T-shingle. This step produces an $m \times n$ binary group (T-shingle, ID), where n is the total number of documents and m is the average T-shingle number of the document.

Step 2. The $m \times n$ binary group (T-shingle, ID) is sorted so that documents with the same T-shingle are clustered together.

Step 3. The sorted (T-shingle, ID) lists are scanned to extract the ID with the same T-shingle to form an (ID-ID, *Count*), where the *Count* represents the number of similar T-shingles in two documents.

Step 4. If the *Count* in (ID-ID, *Count*) is greater than a certain threshold, the document pair corresponding to the ID-ID is extracted to form a document pair to be similarly detected.

B. Threshold Filtering Strategy of Document Pairs

Because the estimator of WOPH accords with a binomial distribution, the calculation speed can be improved by combining the dynamic double-filtering threshold proposed in [41] during eigenvalue comparison. In the eigenvalue comparison, if k = 100, 200, ... the comparison point is

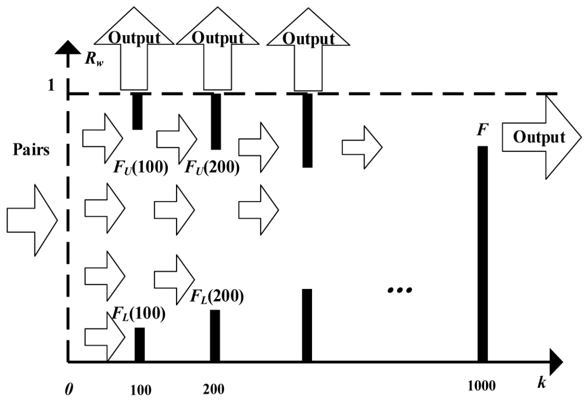

Fig. 8. Dynamic double threshold filtering strategy.

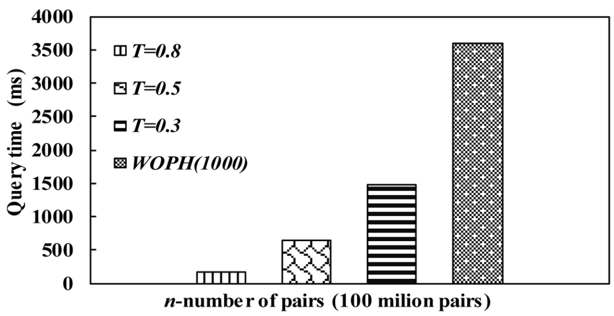

Fig. 9. Comparison of time consumption.

set, and the lower boundary threshold $F_L(k)$ and the upper boundary threshold $F_U(k)$ are defined at each comparison point, and if the following rules are met: if $\hat{R} > F_U(k)$, the output document pair is (S_1, S_2); if $\hat{R}_w < F_L(k)$, the (S_1, S_2) is filtered out, and the remaining filtered data can be used at subsequent observation points (for example, k = 200, 400, 600, etc.). The overall recall rate is 100%. The dynamic double threshold filtering strategy is shown in Fig. 8.

As long as the selected small probability is small enough (for example, 10^{-10}), the probability that the upper and lower thresholds of each comparison point lead to an error is small; in the case where the selection of a small probability does not lead to error filtering, it is possible to select a larger probability to increase the filtering rate.

C. Results

Based on the practical application, 100 million data pairs were calculated by WOPH, accounting for almost $100\times10^6\times k$ comparisons in total. The set small probability value is 10^{-10}. According to Theorems 1 and 2 in [41], the upper bound threshold $F_U(k)$ and the lower bound threshold $F_L(k)$ can be determined. Setting the threshold T_0 to 0.8, 0.5, 0.3 in WOPH(1000) for time testing, the document pairs with similarity less than T_0 were outputted. The experimental results are shown in Fig. 9.

As shown in Fig. 8, the performance of the calculation can be significantly improved by setting the filter. In the actual document set, similar documents are few in number, that is, documents with a similarity of less than 0.8 are dominant. Therefore, only a small number of comparisons can filter most of the document pairs, thereby reducing the amount of comparison time. For a large number of real data with low similarity, accompanied by high threshold queries, the filter reduces the comparison by 85%, compared with the original WOPH.

VI. CONCLUSION

In this paper, we propose a WOPH. The WOPH can flexibly and quickly change the size of the partition number *kw* according to the different levels of accuracy required. Since k_w is composed of pre-divided k_1 and k_2 in proportion, the time for division of the partition is saved compared with OPH. The variance of WOPH can be decreased, rapidly first and then slowly, and the final calculation accuracy is the same as OPH with same k. In applications, we combined the dynamic double filter with WOPH to reduce the calculation time by terminating unnecessary comparisons in advance. For example, for a large number of real data with a low similarity accompanied by high threshold queries, the filter reduces the comparison of WOPH by 85%.

AKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China (No. 61402165), the Key Research Program of Hunan Province (No. 2016JC2018), the Natural Science Foundation of Hunan Province (No. 2018JJ2099, 2019JJ60054).

REFERENCES

1. S. Yumusak, E. Dogdu, H. Kodaz, A. Kamilaris, and P. Y. Vandenbussche, "SpEnD: linked data SPARQL endpoints discovery using search engines," *IEICE Transactions on Information and Systems*, vol. 100, no. 4, pp. 758-767, 2017.
2. J. Xu, L. Xia, Y. Lan, J. Guo, and X. Cheng, "Directly optimize diversity evaluation measures: a new approach to search result diversification," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, no. 3, article no. 41, 2017.
3. R. Mehrotra, A. Anderson, F. Diaz, A. Sharma, H. Wallach, and E. Yilmaz, "Auditing search engines for differential satisfaction across demographics," in *Proceedings of the 26th*

International Conference on World Wide Web Companion, Perth, Australia, 2017, pp. 626-633.
4. N. Perez-Diaz, D. Ruano-Ordas, F. Fdez-Riverola, and J. R. Mendez, “Boosting accuracy of classical machine learning antispam classifiers in real scenarios by applying rough set theory,” *Scientific Programming*, vol. 2016, article ID. 5945192, 2016.
5. M. V. Aragao, E. P. Frigieri, C. A. Ynoguti, and A. P. Paiva, “Factorial design analysis applied to the performance of SMS anti-spam filtering systems,” *Expert Systems with Applications*, vol. 64, pp. 589-604, 2016.
6. C. Calvert, “Book Review: Priests of Our Democracy: The Supreme Court, Academic Freedom, and the Anti-Communist Purge, by Marjorie Heins,” *Journalism & Mass Communication Quarterly*, vol. 90, no. 3, pp. 619-620, 2013.
7. F. J. Newton, J. D. Wright, and J. D. Newton, “Skills training to avoid inadvertent plagiarism: results from a randomised control study,” *Higher Education Research & Development*, vol. 33, no. 6, pp. 1180-1193, 2014.
8. S. Ray Choudhury and C. L. Giles, “An architecture for information extraction from figures in digital libraries,” in *Proceedings of the 24th International Conference on World Wide Web*, Florence, Italy, 2015, pp. 667-672.
9. F. Zhang, Z. Gao, and Q. Ye, “Construction of cloud platform for personalized information services in digital library based on cloud computing data processing technology,” *Automatic Control and Computer Sciences*, vol. 49, no. 6, pp. 373-379, 2015.
10. A. Hinze, D. Bainbridge, S. J. Cunningham, C. Taube-Schock, R. Matamua, J. S. Downie, and E. Rasmussen, “Capisco: low-cost concept-based access to digital libraries,” *International Journal on Digital Libraries*, pp. 1-28, 2018. https://doi.org/10.1007/s00799-018-0232-3
11. J. Draisma, E. Horobet, G. Ottaviani, B. Sturmfels, and R. R. Thomas, “The Euclidean distance degree of an algebraic variety,” *Foundations of Computational Mathematics*, vol. 16, no. 1, pp. 99-149, 2016.
12. L. H. Lee, C. H. Wan, R. Rajkumar, and D. Isa, “An enhanced Support Vector Machine classification framework by using Euclidean distance function for text document categorization,” *Applied Intelligence*, vol. 37, no. 1, pp. 80-99, 2012.
13. Y. Mishchenko, “A fast algorithm for computation of discrete Euclidean distance transform in three or more dimensions on vector processing architectures,” *Signal, Image and Video Processing*, vol. 9, no. 1, pp. 19-27, 2015.
14. D. Chakraborty, E. Goldenberg, and M. Koucky, “Streaming algorithms for embedding and computing edit distance in the low distance regime,” in *Proceedings of the 48th Annual ACM Symposium on Theory of Computing*, Cambridge, MA, 2016, pp. 712-725.
15. C. Luo, J. Zhan, X. Xue, L. Wang, R. Ren, and Q. Yang, “Cosine normalization: using cosine similarity instead of dot product in neural networks,” in *Artificial Neural Networks and Machine Learning.* Cham: Springer, 2018, pp. 382-391.
16. C. Wu and B. Wang, “Extracting topics based on Word2Vec and improved Jaccard similarity coefficient,” in *Proceedings of 2017 IEEE 2nd International Conference on Data Science in Cyberspace (DSC)*, Shenzhen, China, 2017, pp. 389-397.
17. S. M. Tyar and T. Win, “Jaccard coefficient-based word sense disambiguation using hybrid knowledge resources,” in *Proceedings of 2015 7th International Conference on Information Technology and Electrical Engineering (ICITEE)*, Chiang Mai, Thailand, 2015, pp. 147-151.
18. J. Santisteban and J. Tejada-Cárcamo, “Unilateral weighted Jaccard coefficient for NLP,” in *Proceedings of 2015 14th Mexican International Conference on Artificial Intelligence (MICAI)*, Cuernavaca, Mexico, 2015, pp. 14-20.
19. A. K. Gupta and N. Sardana, “Significance of clustering coefficient over Jaccard index,” in *Proceedings of 2015 8th International Conference on Contemporary Computing (IC3)*, Noida, India, 2015, pp. 463-466.
20. A. Shrivastava, “Optimal densification for fast and accurate minwise hashing,” in *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia, 2017, pp. 3154-3163.
21. D. Ravichandran, P. Pantel, and E. Hovy, “Randomized algorithms and NLP: using locality sensitive hash functions for high speed noun clustering,” in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05)*, Ann Arbor, MI, 2005, pp. 622-629.
22. A. Shrivastava and P. Li, “Asymmetric minwise hashing for indexing binary inner products and set containment,” in *Proceedings of the 24th International Conference on World Wide Web*, Florence, Italy, 2015, pp. 981-991.
23. J. P. Kumar and P. Govindarajulu, “Near-duplicate web page detection: an efficient approach using clustering, sentence feature and fingerprinting,” *International Journal of Computational Intelligence Systems*, vol. 6, no. 1, pp. 1-13, 2013.
24. V. A. Narayana, P. Premchand, and A. Govardhan, “A novel and efficient approach for near duplicate page detection in web crawling,” in *Proceedings of 2009 IEEE International Advance Computing Conference*, Patiala, India, 2009, pp. 1492-1496.
25. A. Wilson, M. Cox, D. Elsborg, D. Lindholm, and T. Traver, “A semantically enabled metadata repository for scientific data,” *Earth Science Informatics*, vol. 8, no. 3, pp. 649-661, 2015.
26. A. Abdul Rahman, G. Roe, M. Olsen, C. Gladstone, R. Whaling, N. Cronk, R. Morrissey, and M. Chen, “Constructive visual analytics for text similarity detection,” *Computer Graphics Forum*, vol. 36, no. 1, pp. 237-248, 2017.
27. P. G. V. Naranjo, M. Shojafar, H. Mostafaei, Z. Pooranian, and E. Baccarelli, “P-SEP: a prolong stable election routing algorithm for energy-limited heterogeneous fog-supported wireless sensor networks,” *The Journal of Supercomputing*, vol. 73, no. 2, pp. 733-755, 2017.
28. K. Fang, B. Sivakumar, and F. M. Woldemeskel, “Complex networks, community structure, and catchment classification in a large-scale river basin,” *Journal of Hydrology*, vol. 545, pp. 478-493, 2017.
29. R. S. Renu and G. Mocko, “Computing similarity of text-based assembly processes for knowledge retrieval and reuse,” *Journal of Manufacturing Systems*, vol. 39, pp. 101-110, 2016.
30. Y. Huang, Z. Jiang, C. He, J. Liu, B. Song, and L. Liu, “A semantic-based visualised wiki system (SVWkS) for lesson-learned knowledge reuse situated in product design,”

International Journal of Production Research, vol. 53, no. 8, pp. 2524-2541, 2015.
31. K. Drame, G. Diallo, F. Delva, J. F. Dartigues, E. Mouillet, R. Salamon, and F. Mougin, "Reuse of termino-ontological resources and text corpora for building a multilingual domain ontology: an application to Alzheimer's disease," *Journal of Biomedical Informatics*, vol. 48, pp. 171-182, 2014.
32. D. Thanou, P. A. Chou, and P. Frossard, "Graph-based compression of dynamic 3D point cloud sequences," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1765-1778, 2016.
33. P. Indyk, "A small approximately min-wise independent family of hash functions," in *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms*, Baltimore, MD, 1999, pp. 454-456.
34. T. Itoh, Y. Takei, and J. Tarui, "On the sample size of k-restricted min-wise independent permutations and other k-wise distributions," in *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, San Diego, CA, 2003, pp. 710-719.
35. P. Li, K. W. Church, and T. J. Hastie, "One sketch for all: Theory and application of conditional random sampling," *Advances in Neural Information Processing Systems*, vol. 21, pp. 953-960, 2009.
36. P. Li and C. Konig, "b-Bit minwise hashing," in *Proceedings of the 19th International Conference on World Wide Web*, Raleigh, NC, 2010, pp. 671-680.
37. P. Li, A. Owen, and C. H. Zhang, "One permutation hashing for efficient search and learning," 2012; https://arxiv.org/abs/1208.1259.
38. A. Czumaj, O. Goldreich, D. Ron, C. Seshadhri, A. Shapira, and C. Sohler, "Finding cycles and trees in sublinear time," *Random Structures & Algorithms*, vol. 45, no. 2, pp. 139-184, 2014.
39. M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Machine learning of linear differential equations using Gaussian processes," *Journal of Computational Physics*, vol. 348, pp. 683-693, 2017.
40. P. Li, A. Owen, and C. H. Zhang, "One permutation hashing," *Advances in Neural Information Processing Systems*, vol. 25, pp. 3113-3121, 2012.
41. X. Yuan, Y. Cao, J. Long, G. Zhao, "Dynamic double-threshold filter for minwise hash," *Journal of Shanghai Jiaotong University*, vol. 50, no. 7, pp. 1075-1081, 2016.

Xinpan Yuan

Xinpan Yuan was born in Hunan Province, China. He received Ph.D. degree in computer science from the Central South University in 2012. Currently, he is a lecturer in School of Computer Science of Hunan University of Technology, China. His research interests include information retrieval, data mining, and NLP.

Songlin Wang

Songlin Wang was born in Henan Province, China. He received B.S. degree in computer science from Hunan University of Technology in 2017. Currently, he is a master's student in School of Computer Science of Hunan University of Technology, China. His current research interests include information retrieval and NLP.

Xiaojun Deng

Xiaojun Deng is a professor in the School of Computer, Hunan University of Technology. In 2004, he received the master's degree in computer science and technology from National University of Defense Technology, Changsha, China. His research interests include computer vision, digital image processing, and computer network.