Improving the Sensitivity of MinHash Through Hash-Value Analysis

Gregory Kucherov
LIGM, CNRS/Université Gustave Eiffel, Marne-la-Vallée, France

Steven Skiena
Dept. of Computer Science, Stony Brook University, Stony Brook, NY, USA

Abstract

MinHash sketching is an important algorithm for efficient document retrieval and bioinformatics. We show that the value of the matching MinHash codes convey additional information about the Jaccard similarity of S and T over and above the fact that the MinHash codes agree. This observation holds the potential to increase the sensitivity of minhash-based retrieval systems. We analyze the expected Jaccard similarity of two sets as a function of observing a matching MinHash value a under a reasonable prior distribution on intersection set sizes, and present a practical approach to using MinHash values to improve the sensitivity of traditional Jaccard similarity estimation, based on the Kolmogorov-Smirnov statistical test for sample distributions. Experiments over a wide range of hash function counts and set similarities show a small but consistent improvement over chance at predicting over/under-estimation, yielding an average accuracy of 61% over the range of experiments.

2012 ACM Subject Classification Theory of computation → Bloom filters and hashing

Keywords and phrases MinHash sketching, sequence similarity, hashing

Digital Object Identifier 10.4230/LIPIcs.CPM.2023.20

Funding *Steven Skiena*: Partially supported by a travel grant from Université Paris-Est, and NSF grant IIS-1927227.

1 Introduction

MinHash sketching is an important algorithm for efficient document retrieval. It reduces a set S of size n to a smaller representation of size $m \ll n$ by applying m distinct hash functions $h_1, \ldots h_m$ to each of the n elements of S, and identifies the smallest hash code for each h_i. This vector of minimum hash codes serves a sketch for the larger set S. A classical result [2, 3] shows that the probability that smallest hash codes of two sets S and T are equal is identical to $J(S,T)$, the Jaccard similarity of S and T. Thus the fraction of matching MinHash codes represents an unbiased estimator of $J(S,T)$.

Hash code values, by definition, are not supposed to mean anything. They represent mappings of an item x to a pseudorandom integer $h(x)$ for purpose of fast identity matching and retrieval. The relative values of $h(x)$ and $h(y)$ for items x and y have no special properties beyond that of $h(x) = h(y)$ likely implies that $x = y$ for the conventional hash functions as employed in algorithms such as MinHash.

But in this paper, we report a curious observation associated with MinHash. Suppose the MinHash values for sets $S = \{s_1, \ldots, s_n\}$ and $T = \{t_1, \ldots, t_n\}$ equal both a particular value, namely:

$$a = \min_{j=1}^{n} h_i(s_j) = \min_{j=1}^{n} h_i(t_j)$$

We shall show that the value of this matching MinHash value a conveys additional information about the Jaccard similarity of S and T over and above the fact that the MinHash values agree.

This observation holds the potential to increase the sensitivity of minhash-based retrieval systems. Our main results in this paper are:

- We explain why observing a larger matching MinHash value a increases the expectation of high similarity between S and T. Specifically, the expected value of a common MinHash value a for two n-element sets with intersection size i is $N/(2n-i+1)$, presuming the underlying hash function selects an integer from $[0...N]$ uniformly at random.
- We analyze the expected Jaccard similarity of two sets as a function of observing a matching MinHash value a under a reasonable prior distribution on intersection set sizes, specifically the case where pairs of n-element sets have equal probability of intersection size i for $1 \leq i \leq n$. Experimental results confirm a modest increase in the sensitivity of our hash-code weighted variant of MinHash over the original, over a range of set similarities and number of hash codes.
- We present a practical approach to using MinHash values to improve the sensitivity of traditional Jaccard similarity estimation, based on the Kolmogorov-Smirnov statistical test for sample distributions. Our techniques provide a supplemental signal suggesting whether the fraction of matching MinHashes is more likely to over-estimate or underestimate the true Jaccard similarity between two sets. Experiments over a wide range of hash counts (k) and set similarities show a small but consistent improvement over chance, specifically an average accuracy of 61% over the range of experiments.

We believe that this orthogonal view of measuring Jaccard similarity through the value of matching MinHash codes is novel, and will inspire further interest. Although the practical improvement we have demonstrated is not large, we believe that better interpretations of the underlying statistics may yield better results.

This paper is organized as follows. We begin with a survey of the literature of MinHash and related techniques in Section 2. We provide intuition as to why the value of the matching MinHash value matters through a thought experiment in Section 3. We present our analysis of the expected intersection size as a function of MinHash value for an appropriate prior distribution in Section 4, and ways to combine this information into an estimate of Jaccard similarity in Section 5. An alternate approach to interpret the values of MinHash codes using the Kolmogorov-Smirnoff statistical test is presented in Section 6. Finally, we conclude with some open problems raised by our work.

2 Prior Work

Broder [2, 3] developed MinHash as a solution to identifying similar documents (represented as sets of shingles or substrings) within a large text corpus while avoiding the quadratic-time costs of explicitly comparing each pair of documents. A function $h(x)$ is applied to each set element, mapping each element x to a pseudo-random integer. Each set S is represented by the minimum hash value among all its elements.

The *Jaccard similarity* $J(S,T)$ of two sets S and T is defined as

$$J(S,T) = \frac{|S \cup T|}{|S \cap T|}.$$

For identical sets, $J(S,T) = 1$ while for disjoint sets, $J(S,T) = 0$. Broder [2, 3] observed that the probability that two sets S and T generate the same MinHash value exactly equals the Jaccard similarity of the two sets, $J(S,T)$. The fraction of matching minimum hashes over k independent hash functions provides an unbiased estimator of $J(S,T)$, with the variance of this estimate equal to $J(S,T)(1-J(S,T))/k$ [4]. Surveys of MinHash sketching include Cohen [9].

MinHash is one of the most important algorithms for web search and duplicate detection [12, 15, 17], social networks [8], and machine learning [7, 19]. More recently, it has been successfully applied to bioinformatics for sketching large DNA sequence datasets, starting from the seminal Mash software [21] and followed by related tools [5, 28, 1, 20]. Such applications motivate our efforts in this paper to increase the sensitivity of minimum hashing-based similarity measures.

Locality Sensitive Hashing (LSH) techniques seek to map similar data objects to the same hash codes with a higher probability than the dissimilar ones by adopting a family of randomized hash functions. Indyk et al. [16, 14, 10] introduced the notion of locality-sensitive hashing in the context of nearest-neighbor search and string similarity. MinHash can be viewed an instance of locality sensitive hashing. An extended survey of locality-sensitive hashing can be found in [24].

SimHash [6] is a LSH-based method which provides an unbiased estimator of the similarity between two vectors. Specifically, the probability that two vectors u and v generate the same SimHash value equals the Cosine similarity of u and v. Henzinger [15] performed a large-scale comparison of MinHash and SimHash on detecting similar web pages, finding that a hybrid of the two approaches yielded the best results. Srivastava and Li [22] present analysis and experiments to suggests that MinHash is more sensitive than SimHash in regions of high similarity.

Each subset element is granted equal weight in traditional MinHash schemes, but this can be generalized in weighted MinHash, perhaps to reflect the TD-IDF values of each word. Weighted MinHash algorithms are surveyed in [25].

Finally, we mention another related sketching scheme named HyperLogLog [13] primarily designed for the task of estimating the number of distinct items in a stream, but also capable of estimating the cardinality of the union of two sets and therefore their Jaccard similarity. Several works proposed unifying combinations of the two sketches [11, 27]. Note that HyperLogLog has some relationship with our work, as it employs the idea of estimating the cardinality of a random set from its minimum value. However, a direct application of this idea to MinHash has not been made, to our knowledge.

3 Thought Experiment: Why MinHash Values Matter

We present the following thought experiment to illustrate how the actual value of matching MinHash codes provides information about Jaccard similarity. For a set S, let $M_h(S)$ denote its MinHash value under a given hash function h, i.e. $M_h(S) = \min_{s \in S}\{h(s)\}$. We use the notation $M(S)$ when h is irrelevant (but assumed fixed across sets).

Now consider following two "extreme" situations involving pairs of sets S and T, each with n elements:

1. S and T are identical. Hence by definition, the minimum hash values must match, so $a_1 = M_h(S) = M_h(T)$,
2. S and T intersect in only one element x, which happens to be the minimum value of both under h, so $a_2 = h(x) = M_h(S) = M_h(S)$.

Now, given just the two unlabeled values for a_1 and a_2, can we correctly assign these codes to the appropriate case above with probability greater than 1/2?

Assume hash function h selects an integer from the range $[0 \ldots N]$ uniformly at random. Now suppose that two n-element sets with intersection size i share a common MinHash value m. Then m is the smallest of the $2n - i$ values in the union. Since the expected minimum of ℓ numbers drawn uniformly at random from $[0..N]$ is $N/(\ell + 1)$, the expected value of m is

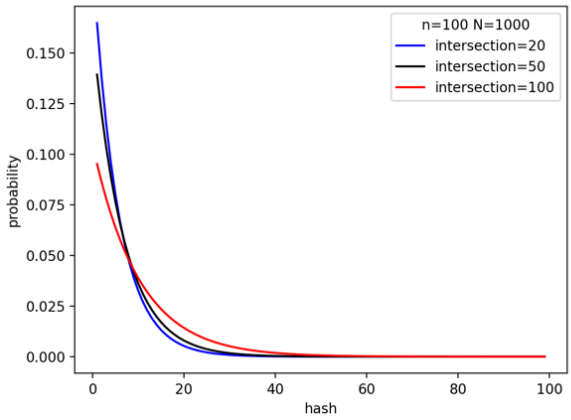

Figure 1 Probability distributions that two sets of size 100 share a common MinHash value, as a function of the size of their intersection (respectively 20, 50, and 100). The probability of small matching MinHash values are increase for relatively dissimilar sets.

$N/(2n-i+1)$. In the first case above, $i=n$, so $\mathbb{E}[a]=N/(n+1)$, while for the second case $i=1$ and $\mathbb{E}[b]=N/(2n)$. Thus it is more likely that $\min(a_1,a_2)$ corresponds to case (1) and $\max(a_1,a_2)$ to case (2).

The situation is illustrated in Figure 1, which shows the probability of observing a given MinHash value a for three different intersection sizes. The probability of observing a MinHash value of 0 with a possible range $[0\ldots1000]$ is almost twice as high for two 100-element sets with a 20-element intersection than when the sets are identical. More similar pairs of sets, with larger intersection sizes, have greater probability of large matching MinHash values.

4 Expected Intersection Size as a Function of MinHash Value

In this section, we analyze the expected intersection size of two sets based on observing a particular matching MinHash value. For two n-element sets S and T where $|S\cap T|=i$, $J(S,T)=i/(2n-i)$. Thus analyzing the intersection size of S and T provides a result which can be alternately interpreted in the context of the Jaccard similarity of S and T for n-element sets.

Let S and T be two sets each of size n. We limit our attention to the case where S and T are non-disjoint, which is necessary for MinHash values to legitimately collide, so $S\cap T\neq\emptyset$. Further, we assume that range of h from $[0\ldots N]$ is sufficiently large relative to n that we can discount the possibility of spurious collisions, namely that there does not exist $s\in S$ and $t\in T$ where $h(s)=h(t)$ despite $s\neq t$.

4.1 Prior Distribution

Determining the expected intersection size as a function of matching hash values requires knowledge of a prior distribution on the value of the intersection size. In the analysis below, we base our analysis on a uniform prior distribution, that all intersection sizes between S and T are equally likely. Thus for every $i\in[1..n]$, $\mathbb{P}[|S\cap T|=i]=1/n$.

The uniform distribution appears most natural to us as a general prior, which is why we analyze this case below. That said, the true prior distribution differs with application, particularly as to whether pairs of randomly selected sets are likely to have large or small intersection sizes. The analysis below can be repeated for any particular well specified prior distribution in an analogous fashion.

4.2 Analysis

The probability of two sets sharing the MinHash value equals the Jaccard similarity index, that is

$$\mathbb{P}[M(S) = M(T) \mid i = |S \cap T|] = \frac{i}{2n - i}. \tag{1}$$

Because all intersections are equiprobable under our prior distribution, we have

$$\mathbb{P}[|S \cap T| = i \mid M(S) = M(T)] = \frac{\frac{i}{2n-i}}{\sum_{j=1}^{n} \frac{j}{2n-j}} \tag{2}$$

Note that given ℓ random numbers $x_1, \ldots, x_\ell$ uniformly drawn from $[1..N]$, for $a \in [1..N]$, we have

$$\mathbb{P}[\min\{x_1, \ldots, x_\ell\} \leq a] = 1 - \mathbb{P}[x_1 > a \ \& \ \ldots \ \& \ x_\ell > a] = 1 - \left(1 - \frac{a}{N}\right)^\ell. \tag{3}$$

Then, the probability the MinHash is exactly a is given by

$$\mathbb{P}[\min\{x_1, \ldots, x_\ell\} = a] = \left(1 - \frac{a-1}{N}\right)^\ell - \left(1 - \frac{a}{N}\right)^\ell. \tag{4}$$

We now estimate the conditional probability $\mathbb{P}[|S \cap T| = i \mid M(S) = M(T) = a]$. We have

$$\begin{aligned}
\mathbb{P}[|S \cap T| = i \mid M(S) = M(T) = a] &= \frac{\mathbb{P}[(|S \cap T| = i) \wedge (M(S) = M(T)) \wedge (a = M(S \cup T))]}{\mathbb{P}[(M(S) = M(T)) \wedge (a = M(S \cup T))]} \\
&= \frac{\mathbb{P}[|S \cap T| = i] \cdot \mathbb{P}[a = M(S \cup T) \mid |S \cap T| = i] \cdot \mathbb{P}[M(S) = M(T) \mid (|S \cap T| = i) \wedge (a = M(S \cup T)]}{\sum_{i=1}^{n} (\mathbb{P}[(M(S) = M(T)) \wedge (a = M(S \cup T)) \mid |S \cap T| = i] \cdot \mathbb{P}[|S \cap T| = i])} \\
&= \frac{\mathbb{P}[(a = M(S \cup T)) \mid (|S \cap T| = i)] \cdot \mathbb{P}[(M(S) = M(T)) \mid (|S \cap T| = i) \wedge (a = M(S \cup T))]}{\sum_{i=1}^{n} \mathbb{P}[(M(S) = M(T)) \wedge (a = M(S \cup T)) \mid |S \cap T| = i]}
\end{aligned} \tag{5}$$

The last rewrite follows because $\mathbb{P}[|S \cap T| = i] = 1/n$ is the same for all i. To further simplify Eqn. 5, observe that

$$\mathbb{P}[M(S) = M(T) \mid (|S \cap T| = i) \wedge (a = M(S \cup T))] = \mathbb{P}[M(S) = M(T) \mid |S \cap T| = i]$$

because the event of sharing common MinHash ($M(S) = M(T)$) is independent of its value (a) for a fixed intersection size. For the same reason, in the denominator,

$$\begin{aligned}
&\mathbb{P}[(M(S) = M(T)) \wedge (a = M(S \cup T)) \mid |S \cap T| = i] = \\
&\qquad \mathbb{P}[M(S) = M(T) \mid |S \cap T| = i] \cdot \mathbb{P}[a = M(S \cup T) \mid |S \cap T| = i]
\end{aligned}$$

Eqn. 5 then rewrites to

$$\frac{\mathbb{P}[a = M(S \cup T) \mid |S \cap T| = i] \cdot \mathbb{P}[M(S) = M(T) \mid |S \cap T| = i]}{\sum_{i=1}^{n} (\mathbb{P}[M(S) = M(T) \mid |S \cap T| = i] \cdot \mathbb{P}[a = M(S \cup T) \mid |S \cap T| = i])}. \tag{6}$$

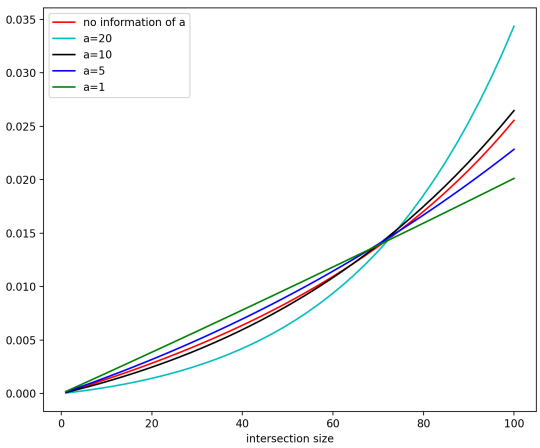

Figure 2 The probability of intersection size of two sets of size 100 sharing a common MinHash value. The red curve shows the probability of having a given intersection size (formula (2)). The other curves show the same probability conditioned on the value a of common MinHash (formula (7)), where the hash space is [1..1000]. Larger values of a favor larger intersection sizes.

Using (4), (1), we obtain

$$\mathbb{P}[|S \cap T| = i \mid M(S) = M(T) = a] = \frac{\frac{i}{2n-i}\left(\left(1-\frac{a-1}{N}\right)^{2n-i} - \left(1-\frac{a}{N}\right)^{2n-i}\right)}{\sum_{j=1}^{n}\frac{j}{2n-j}\left(\left(1-\frac{a-1}{N}\right)^{2n-j} - \left(1-\frac{a}{N}\right)^{2n-j}\right)} \tag{7}$$

Using (7), we can compute the expected intersection size as a function of the shared MinHash value:

$$\mathbb{E}[|S \cap T| \mid M(S) = M(T) = a] = \sum_{i=1}^{n} i \cdot \mathbb{P}[|S \cap T| = i \mid M(S) = M(T) = a] \tag{8}$$

As an illustration, Figure 2 shows probability distributions of intersection sizes without taking into account the common MinHash value (formula (2)) and knowing the common MinHash value (formula (7)). The figure demonstrates that larger common MinHash values provide an evidence for larger intersection sizes.

5 Hash Scoring for Sketch Similarity

In the classical MinHash scheme, the probability that two sets have matching MinHash is equal to the Jaccard similarity between the two sets. Thus, the fraction of matches taken over a number of trials provides an unbiased estimator of the Jaccard similarity. We have shown that the values of these matching MinHashes provides an orthogonal measure of similarity. The question is what the best way to combine these measures is.

We propose the following initial strategy. Traditional MinHash can be interpreted as averaging the values of 0/1 indicator variables, where a match of hash codes is represented by 1 and a mismatch by 0. We will replace the value associated with matching hashes by real values that over/underweight based on the value of shared MinHash. More specifically, a shared MinHash value will contribute with weight

$$\frac{\mathbb{E}[|S \cap T| \mid M(S) = M(T) = a]}{\mathbb{E}[|S \cap T|]}, \tag{9}$$

Table 1 Improvement (in terms of the average reduction of absolute error) in estimating set intersection size by summing hash-weighted counts vs. equal weighting to estimate Jaccard similarity for different numbers of hash functions (rows) and set intersection sizes (columns). Bolded entries represent improvement over traditional MinHash estimation, representing 116/140 = 82.9% of the non-trivial cells in the table.

k / int	1	2	3	5	10	15	19	20
1	**.0068**	**.0140**	**.0205**	**.0375**	**.0837**	**.1150**	**.1550**	-.1650
2	**.0067**	**.0138**	**.0269**	**.0395**	**.0583**	**.0626**	**.1020**	-.1120
3	**.0092**	**.0250**	**.0258**	**.0466**	-.0914	**.0369**	**.0824**	-.0833
4	**.0128**	**.0250**	**.0308**	**.0496**	**.0042**	**.0093**	**.0648**	-.0615
5	**.0161**	**.0240**	**.0312**	**.0493**	**.0280**	-.0799	**.0684**	-.0532
6	**.0163**	**.0251**	**.0377**	**.0537**	-.0471	**.0012**	**.0550**	-.0489
7	**.0116**	**.0301**	**.0368**	-.0277	-.0008	**.0071**	**.0636**	-.0384
8	**.0143**	**.0257**	**.0356**	**.0119**	**.0282**	-.0072	**.0615**	-.0347
9	**.0159**	**.0201**	**.0350**	**.0200**	-.0377	**.0018**	**.0298**	-.0325
10	**.0139**	**.0283**	**.0416**	**.0233**	**.0029**	-.0411	-.0146	-.0290
11	**.0155**	**.0266**	**.0476**	**.0266**	**.0156**	**.0028**	-.0166	-.0244
12	**.0147**	**.0262**	**.0218**	**.0350**	-.0172	-.0023	-.0004	-.0231
13	**.0165**	**.0353**	**.0016**	**.0403**	**.0133**	-.0044	**.0035**	-.0213
14	**.0176**	**.0283**	**.0157**	-.0133	**.0162**	-.0035	**.0070**	-.0193
15	**.0168**	**.0280**	**.0125**	**.0077**	-.0207	-.0226	**.0148**	-.0172
16	**.0164**	**.0291**	**.0142**	**.0173**	**.0081**	-.0003	**.0059**	-.0155
17	**.0189**	**.0328**	**.0196**	**.0115**	**.0113**	-.0045	**.0084**	-.0144
18	**.0163**	**.0293**	**.0193**	**.0207**	-.0157	**.0071**	**.0058**	-.0144
19	**.0152**	-.0060	**.0208**	**.0304**	**.0043**	**.0036**	**.0092**	-.0118
20	**.0146**	**.0094**	**.0213**	**.0289**	**.0059**	-.0193	-.0122	-.0115

where the numerator is defined by Eqn 8, and

$$\mathbb{E}[|S \cap T|] = \sum_{i=1}^{n} \mathbb{P}(|S \cap T| = i \mid M(S) = M(T)) \tag{10}$$

is the expected intersection size independent of MinHash value, i.e. implied by the prior distribution of intersection sizes.

5.1 Experimental Results

We performed a modest experiment to evaluate the performance of this technique, with the results reported in Table 1. We limited the experiment to small sets ($n = 20$), but consider a broad range of hash function counts ($1 \leq k \leq 20 = n$) and set similarities defined by intersection sizes from $1 \leq i \leq 20 = n$. Each cell represents the average difference in absolute error in estimating intersection size between MinHash with over/underweighting and traditional 0/1 counts, where each cell is averaged over 1,000 independent random trials.

We note that the rightmost column in Table 1 (intersection size 20 out of a possible 20) corresponds to identical sets, where the traditional Jaccard (and intersection size) estimate is always correct, leaving our proposed method with no room for possible improvement. But 116/140 = 82.9% of the non-trivial cells show improvement over the traditional MinHash baseline.

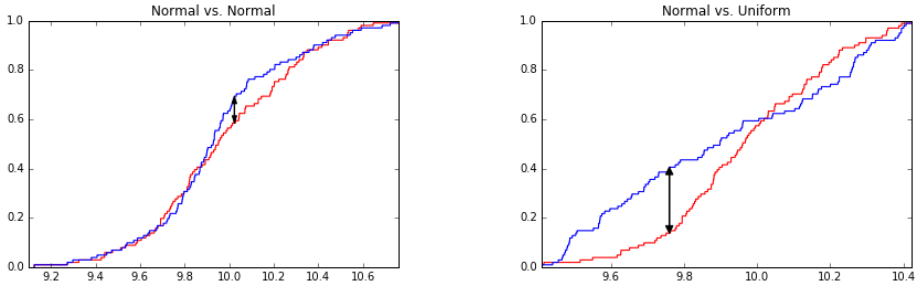

Figure 3 The Kolmogorov-Smirnov test quantifies the difference between two probability distributions by the maximum y-distance gap between the two cumulative distribution functions. On the left, two samples from the same normal distribution. On the right, comparison of samples from uniform and normal distributions drawn over the same x-range..

6 Sketch Evaluation using the Kolmogorov-Smirnov Test

We now propose an alternate approach to improve the Jaccard similarity estimate offered by the classical MinHash approach, namely the fraction of matching MinHash values in k trials. We seek to improve this estimate by analyzing the distribution of the values of the matching hashes from these trials to decide whether it is more likely to be over or under estimating the actual similarity. A key advantage of this approach over of that of Section 4 is that does not require a prior distribution on the actual intersection sizes.

Our approach is based on the *Kolmogorov-Smirnov* (KS) statistical test [18, 23], which compares empirical cumulative distribution functions (CDFs) to assess whether two samples are drawn from the same underlying distribution. We will use it to compare the observed distribution of matching MinHash values against the theoretical distribution for the classical Jaccard estimate. The direction of the largest deviation suggests whether it is more likely an over or under estimate.

6.1 The Kolmogorov-Smirnov Test

In the KS-test, the empirical cumulative distribution function (CDFs) of the two different samples are plotted on the same chart. If the two samples are drawn from the same distribution, the ranges of x values should largely overlap. An empirical CDF $\hat{F}(x)$ of a sample is defined as the fraction of the sample $\leq x$.

We seek to identify the value of x for which the associated values of the two CDFs differ by as much as possible. The distance $D(\hat{F}, \hat{G})$ between two empirical CDFs $\hat{F}$ and $\hat{G}$ is the difference of the y values at this critical x, formally stated as

$$D(\hat{F}, \hat{G}) = \max_x |\hat{F}(x) - \hat{G}(x)|$$

The more substantially that two samples differ in this fashion, the more likely it is that they were drawn from different distributions. Figure 3 (left) shows two independent samples from the same normal distribution. In contrast, Figure 3 (right) compares a sample drawn from a normal distribution against one drawn from the uniform distribution. The big gaps near the tails provide evidence that the two samples are drawn from different distributions.

The KS-test compares the value of $D(\hat{F}, \hat{G})$ against a particular target, declaring that two distributions differ at the significance level of α when:

$$D(\hat{F}, \hat{G}) > c(\alpha)\sqrt{\frac{n_1 + n_2}{n_1 n_2}}$$

where $c(\alpha)$ is a distribution-independent constant to look up in a table. In this paper, we use the ideas behind the KS-test for qualitative evaluation instead of precisely measuring statistical significance, and so will be interested in the direction of the deviation without this associated constant.

6.2 Application to MinHash Analysis

As explained above, the distribution of matching MinHash values differs as a function of the intersection size or (equivalently) Jaccard similarity between two sets of size n. Recall that for ℓ random numbers $x_1, \ldots, x_\ell$ uniformly drawn from $[1..N]$, for $a \in [1..N]$, we have

$$F_\ell(x) = \mathbb{P}[\min\{x_1, \ldots, x_\ell\} \leq x] = 1 - \mathbb{P}[x_1 > x \;\&\; \ldots \;\&\; x_\ell > x] = 1 - \left(1 - \frac{x}{N}\right)^\ell \approx 1 - e^{-x\ell/N}. \quad (11)$$

This defines the CDF on matching MinHash values. Comparing two sets A and B, both of cardinality n with an intersection of size i, any common MinHash value represents the smallest of $\ell = 2n - i$ random values. Thus the distribution of matching MinHash values is defined by Eqn. 11, given an estimate for the union size ℓ. An important observation for us is that CDFs F_ℓ are majorating one another, that is if $\ell_1 > \ell_2$, then $F_{\ell_1}(x) > F_{\ell_2}(x)$ for any x.

Estimates for the union size ℓ and intersection size i follow from classical MinHash analysis. If m matching MinHash values are observed in k trials, m/k is an unbiased estimator of the Jaccard index $\frac{i}{\ell} = \frac{i}{2n-i} = \frac{2n-\ell}{\ell}$. Therefore, i and ℓ are estimated respectively by

$$\hat{i} = \frac{2nm}{k+m}, \quad \hat{\ell} = \frac{2nk}{k+m}.$$

We can now employ the idea underlying the KS-test to evaluate how well the m observed MinHash values match the estimated distribution $F_{\hat{\ell}}(x)$. In doing that, we analyze the sign of the critical deviation

$$D(F_{\hat{\ell}}, \hat{F}) = F_{\hat{\ell}}(\tilde{x}) - \hat{F}(\tilde{x}) \text{ for } \tilde{x} = \operatorname{argmax} |F_{\hat{\ell}}(x) - \hat{F}(x)|,$$

where $\hat{F}$ is the empirical CDF obtained from the sample of matching MinHash values. When D is positive, this suggests that the regular MinHash estimate $\hat{\ell}$ is an overestimate and therefore $\hat{i}$ is an underestimate for the true intersection size. Conversely, a negative D provides an evidence that $\hat{i}$ is an overestimate for the true intersection size. This reasoning is supported by the above-mentioned majorating property of CDFs, as it guarantees that the sign of D correctly defines whether the estimate $\hat{\ell}$ should be increased or decreased for the KS-test statistic to be reduced and therefore for the estimated CDF to fit better the observed MinHash values. We thus propose the sign of D as a secondary signal to improve the accuracy of $\hat{i}$ as an estimator for intersection size.

The running time of this test is $O(m \log m)$ because we must sort the observed matching hash values to compute the CDF. It is only necessary to compare the distributions at the m sample points to identify the extremal points, with each comparison efficiently done using the exponential form of Eqn. 11. The magnitude of the deviation directly maps to a confidence value in the direction of change, with p-values obtainable using tables of $c(\alpha)$ values from the standard KS-test. However, in the experiments below we propose to estimate correction direction from the sign of D independent of its magnitude.

Table 2 Performance (in terms of the fraction of correct direction predictions) of the KS-test-based over/under correction, as a function of the number of hash functions k (shown in left column), and the true Jaccard similarity/intersection size (shown in first/second row). Generally speaking, the improvement is greatest at extreme values of similarity (either high or low), and with smaller numbers of hash functions.

Jaccard	.961	.869	.786	.739	.667	.538	.429	.333	.258	.176	.111	.081	
intersect	980	930	880	850	800	700	600	500	410	300	200	150	Avg
1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2	.816	.696	.596	.534	.457	.339	.988	.994	.996	.998	.999	.998	.784
3	.764	.618	.485	.419	.275	.552	.445	.340	.983	.992	.994	.998	.655
4	.711	.556	.440	.651	.561	.388	.570	.475	.361	.982	.989	.993	.640
5	.683	.519	.632	.572	.478	.551	.394	.551	.450	.974	.985	.993	.649
10	.591	.577	.590	.537	.566	.505	.474	.452	.483	.561	.399	.966	.558
20	.502	.540	.565	.573	.545	.564	.535	.528	.456	.460	.395	.514	.515
50	.506	.550	.539	.567	.550	.560	.549	.544	.541	.520	.474	.396	.525
75	.518	.535	.562	.552	.508	.553	.550	.492	.515	.506	.487	.431	.517
100	.512	.552	.556	.561	.553	.578	.552	.552	.542	.525	.468	.454	.534
200	.525	.549	.550	.557	.563	.568	.555	.557	.543	.527	.505	.491	.541
300	.522	.551	.558	.556	.535	.561	.550	.530	.543	.536	.515	.506	.539
500	.529	.550	.550	.559	.556	.574	.558	.553	.543	.534	.521	.511	.545
1000	.520	.552	.556	.567	.561	.565	.557	.548	.550	.540	.512	.512	.545
Average	.621	.596	.584	.586	.550	.561	.591	.580	.608	.690	.660	.697	.610

6.3 Experimental Results

Table 2 summarizes the performance of our KS-based correction strategy over a wide range of hash counts (from $k = 1$ to $k = 1000$) and true Jaccard similarity (from 0.081 to 0.961). For each Jaccard similarity level, we constructed 10,000 pairs of 1000-element sets, each pair constructed to the appropriate level of similarity. We then constructed k independent hash functions of these sets, and determined the number of matching MinHashes for these trials. We then performed the KS-test on the matching values to propose whether the actual Jaccard estimate should be higher or lower than the observed fraction of matches. We chose parameters of our tests (intersection size) so that to avoid the situation when the MinHash estimate exactly equals the true Jaccard similarity, making each case a fair binary trial.

Of the $14 \times 12 = 172$ entries in Table 2, 144 of them (83.7%) are greater than 0.5, meaning the adjustment breaks in the correct direction more often than not. The average accuracy ratio taken over all trials is 61.0%, substantially better than the baseline of 50%.

When employing large numbers of hash functions $k \geq 100$, our technique improves the estimate on average in 57 of 60 (95%) entries, and proves most beneficial in the middle regions where the Jaccard similarity is ≈ 0.5. This is curious, because larger k provides greater resolution on the fraction of matching hash values, thus reducing the quantization error of classical MinHash. But the KS-analysis also improves with more samples as k increases, and continues to refine the similarity estimate even as $k = 1000$. Presumably in the limit as k grows, the improvement over baseline will disappear, but it seems durable over the range of k that appear in general applications.

That our best (and worst) performance occurs for very small k reflects issues of quantization: for an intersection size of $n/2$ and $k = 5$, the best possible estimate must be wrong by at least 10%. As a statistical significance test, the KS-test was designed to be used with a meaningful number of samples per observed distribution. There are likely other statistical tests to do better with small (and maybe even large) values of k.

7 Conclusions

We have demonstrated that the value of matching MinHash values provides additional information on the degree of similarity between pairs of sets. Our wins are small, but they are real. We believe that there exist better methods of integration to synthesize the mix of the number of matching hashes and their values into a more accurate measure of similarity and believe that this is a research direction worth pursuing. We note that even careful analysis of the values of the matching hash codes will be substantially less computationally expensive than that of obtaining the MinHash codes themselves, so these improvements will come at a little computational cost.

The MinHash values that *do not* match also contain some degree of signal concerning the similarity of two sets. Suppose the smallest hash values of two sets do not match, but are both unusually large, say a substantial fraction of the total range N. These large MinHash values signify that both sets exclude the same large fraction of possible elements from the universe, implying they must both be constructed from just a relatively small set of non-excluded elements. This conditioning increases the expected Jaccard similarity, despite the fact that the hash values do not match. We believe this signal to be very weak except in extreme cases, but its analysis may be part of a complete solution.

The theoretical success of MinHash depends strongly upon the elements of the sets being distinct. If multiplicity of elements should be taken into account, one should resort to the weighted variant of MinHash [26]. Extending our ideas to Weighted MinHash is another interesting direction of study for the future.

References

1 Daniel N. Baker and Ben Langmead. Dashing: fast and accurate genomic distances with HyperLogLog. *Genome Biology*, 20:265, 2019. doi:10.1186/s13059-019-1875-0.

2 Andrei Broder. On the resemblance and containment of documents. In *Proceedings of the Compression and Complexity of Sequences 1997*, SEQUENCES '97, pages 21–, Washington, DC, USA, 1997. IEEE Computer Society. URL: http://dl.acm.org/citation.cfm?id=829502.830043.

3 Andrei Z. Broder. Identifying and filtering near-duplicate documents. In *Combinatorial Pattern Matching, 11th Annual Symposium, CPM 2000, Montreal, Canada, June 21-23, 2000, Proceedings*, pages 1–10, 2000. doi:10.1007/3-540-45123-4_1.

4 Andrei Z Broder, Moses Charikar, Alan M Frieze, and Michael Mitzenmacher. Min-wise independent permutations. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 327–336, 1998.

5 C. Titus Brown and Luiz Irber. Sourmash: a library for minhash sketching of dna. *Journal of Open Source Software*, 1(5):27, 2016. doi:10.21105/joss.00027.

6 Moses S Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 380–388, 2002.

7 Lianhua Chi, Bin Li, and Xingquan Zhu. Context-preserving hashing for fast text classification. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 100–108. SIAM, 2014.

8 Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, Michael Mitzenmacher, Alessandro Panconesi, and Prabhakar Raghavan. On compressing social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 219–228, 2009.

9 Edith Cohen. Min-hash sketches: A brief survey, 2016. URL: http://www.cohenwang.com/edith/Surveys/minhash.pdf.

10 Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 253–262, 2004.

11 Otmar Ertl. Setsketch: Filling the gap between minhash and hyperloglog. *Proc. VLDB Endow.*, 14(11):2244–2257, 2021. doi:10.14778/3476249.3476276.

12 Dennis Fetterly, Mark Manasse, Marc Najork, and Janet Wiener. A large-scale study of the evolution of web pages. In *Proceedings of the 12th international conference on World Wide Web*, pages 669–678, 2003.

13 Philippe Flajolet, Éric Fusy, Olivier Gandouet, and Frédéric Meunier. HyperLogLog: the analysis of a near-optimal cardinality estimation algorithm. *Discrete Mathematics & Theoretical Computer Science*, DMTCS Proceedings vol. AH, 2007 Conference on Analysis of Algorithms (AofA 07), January 2007. doi:10.46298/dmtcs.3545.

14 Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. Similarity search in high dimensions via hashing. In *Vldb*, volume 99, pages 518–529, 1999.

15 Monika Henzinger. Finding near-duplicate web pages: a large-scale evaluation of algorithms. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 284–291, 2006.

16 Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613, 1998.

17 Nitin Jindal and Bing Liu. Opinion spam and analysis. In *Proceedings of the 2008 international conference on web search and data mining*, pages 219–230, 2008.

18 Andrey Kolmogorov. Sulla determinazione empirica di una lgge di distribuzione. *Inst. Ital. Attuari, Giorn.*, 4:83–91, 1933.

19 Ping Li, Anshumali Shrivastava, Joshua Moore, and Arnd König. Hashing algorithms for large-scale learning. *Advances in neural information processing systems*, 24, 2011.

20 Brian D. Ondov, Gabriel Starrett, Anna Sappington, Aleksandra Kostic, Sergey Koren, Christopher B. Buck, and Adam M. Phillippy. Mash screen: high-throughput sequence containment estimation for genome discovery. *Genome Biology*, 20:232, 2019. doi:10.1186/s13059-019-1841-x.

21 Brian D Ondov, Todd J Treangen, Páll Melsted, Adam B Mallonee, Nicholas H Bergman, Sergey Koren, and Adam M Phillippy. Mash: fast genome and metagenome distance estimation using minhash. *Genome Biology*, 17(1):1–14, 2016.

22 Anshumali Shrivastava and Ping Li. In defense of MinHash over SimHash. In *Artificial Intelligence and Statistics*, pages 886–894. PMLR, 2014.

23 Nickolay Smirnov. Table for estimating the goodness of fit of empirical distributions. *The annals of mathematical statistics*, 19(2):279–281, 1948.

24 Jingdong Wang, Heng Tao Shen, Jingkuan Song, and Jianqiu Ji. Hashing for similarity search: A survey. *CoRR*, abs/1408.2927, 2014. arXiv:1408.2927.

25 Wei Wu, Bin Li, Ling Chen, Junbin Gao, and Chengqi Zhang. A review for weighted MinHash algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 34(6):2553–2573, 2020.

26 Wei Wu, Bin Li, Ling Chen, Junbin Gao, and Chengqi Zhang. A review for weighted minhash algorithms. *IEEE Trans. Knowl. Data Eng.*, 34(6):2553–2573, 2022. doi:10.1109/TKDE.2020.3021067.

27 Yun William Yu and Griffin M. Weber. Hyperminhash: Minhash in loglog space. *IEEE Trans. Knowl. Data Eng.*, 34(1):328–339, 2022. doi:10.1109/TKDE.2020.2981311.

28 XiaoFei Zhao. Bindash, software for fast genome distance estimation on a typical personal laptop. *Bioinformatics*, 35(4):671–673, 2019.