

Minwise and Maxwise Hashing

The core of the clustering algorithm in this paper will be based on the concepts of minwise hashing as described in [?]. Minwise hashing has repeatedly proven a powerful tool when comparing large sets of strings rapidly, especially for duplicate detection of long articles. The use of minwise hashing for rRNA sequences has already been done in [?], however the method of this paper will be extended by applying two methods of maxwise hashing as described in [?].

Introduction to Minwise Hashing

Let there be two sets A and B . To find the similarity between the two sets, minwise hashing uses the Jaccard similarity measure, which is defined as

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (1)$$

To increase the speed of calculating the Jaccard similarity, it however uses hash functions to find the value. Let us observe how the hash functions are used in this case.

Min-wise Independency

Let $H : U \rightarrow [r]$ be a class of hashfunctions. Then for any set $X \subseteq [r]$ and any $x \in X$ and let $h \in H$ be chosen uniformly at random, it is considered minwise independent if

$$\Pr(h_{\min}(X) = h(x)) = \frac{1}{|X|} \quad (2)$$

where

$$h_{\min}(X) = \min\{\forall x \in X, h(x)\}$$

Meaning that all elements in X must have an equal probability of having the minimum value going through h . As seen in Eq. ??, this probability is reachable using universal hash functions, which is a great increase in speed over perfect hashing functions.

Min-wise sketch

For two sets A and B it has been proven in [?] that Eq. 2 can be linked to the Jaccard similarity in Eq. 1 as

$$\Pr(h_{\min}(A) = h_{\min}(B)) = \frac{|A \cap B|}{|A \cup B|} \quad (3)$$

For a random set S_1 , we may create a table of random $h_{\min,i}, i = 1, \dots, k$ such that

$$\hat{S}_1 = \{h_{\min,1}(S_1), h_{\min,2}(S_1), \dots, h_{\min,k}(S_1)\}$$

We may then compute the similarity of two sets \hat{S}_1 and \hat{S}_2 defined by the above equation as

$$J(A, B) = \frac{1}{k} \cdot \sum_{i=1}^k (h_{\min,i}(S_1) = h_{\min,i}(S_2)) \quad (4)$$

where

$$(h_{\min,i}(S_1) = h_{\min,i}(S_2)) = \begin{cases} 1, & h_{\min,i}(S_1) = h_{\min,i}(S_2) \\ 0, & \text{otherwise} \end{cases}$$

which is called the **minwise sketch**. As we see from Eq. ??, there will be a slight error in the calculation of h_{min} . Therefore we must see what influence the size of k will have on the error. A proof of the error using Chernoff Bounds¹ is found in [?], shows that the relation between k and ϵ , the error, is

$$k = O\left(\log \frac{1}{\epsilon}\right)$$

Thus, k influences the error inversely exponentially, meaning that $k \approx 100$ should guarantee very small error.

Max-wise hashing

The aforementioned modification is one inspired by the method in the paper [?]. It is an extension of the minwise sketch where in addition to using the min-wise independent sets, we add the maxwise independent set too. Very literally, this means that instead of using the minimum hashvalue, we use the maximal hashvalue such that a set X is said to be maxwise independent if

$$\Pr(h_{\max}(X) = h(x)) = \frac{1}{|X|}, h_{\max} = \max\{\forall x \in X, h(x)\} \quad (5)$$

for any $x \in X$. The Jaccard similarity measure for two sets A and B is

$$\Pr(h_{\max}(A) = h_{\max}(B)) = \frac{|A \cap B|}{|A \cup B|} \quad (6)$$

and finally for a random set S_1 we may create a table of random $h_{\max,i}, i = 1, \dots, k$ such that

$$\tilde{S}_1 = \{h_{\max,1}(S_1), h_{\max,2}(S_1), \dots, h_{\max,k}(S_1)\}$$

This sketch will function almost like the minwise sketch. It is first when combining the two sketches that they have interesting properties.

Combining Max-wise and Min-wise

There are two ways of combining the max-wise and the min-wise algorithm. One is the method in [?], where they halve the amount of hashfunctions, so that for $i = 1, \dots, k/2$

$$J(A, B) = \frac{1}{K} \sum_{i=1}^{K/2} (h_{\min,i}(A) = h_{\min,i}(B) + h_{\max,i}(A) = h_{\max,i}(B)) \quad (7)$$

¹A probabilistic method to find the exponentially decreasing bounds between two independent variates.

Let this method be called **Max-minwise halved sketch** (abbr. $\mathbf{Mm}_{\frac{1}{2}}$). This method has been proven to be double as quick as the min-wise hashing, without loss of precision[?]. It is also shown in Lemma 2 in [?] that for $i = 1, \dots, k/2$

$$\Pr(h_{\min,i}(A) = h_{\min,i}(B) | h_{\max,i}(A) = h_{\max,i}(B)) = \frac{|A \cap B| - 1}{|A \cup B| - 1}$$

Meaning that a collision between h_{\min} and h_{\max} is very unlikely.

Another method, which was developed in the course of this paper uses the following combination

$$J(A, B) = \frac{1}{k} \sum_{i=1}^k (h_{\min,i}(A) = h_{\min,i}(B) | h_{\max,i}(A) = h_{\max,i}(B)) \quad (8)$$

where

$$h_{\min,i}(A) = h_{\min,i}(B) | h_{\max,i}(A) = h_{\max,i}(B) = \begin{cases} 1, & h_{\min,i}(S_1) = h_{\min,i}(S_2) \\ 1, & h_{\max,i}(S_1) = h_{\max,i}(S_2) \\ 0, & \text{otherwise} \end{cases}$$

Let this method be called **Max-minwise sketch** (abbr. \mathbf{Mm}). This also has finds the Jaccard similarity by the following proof:

$$\begin{aligned} \frac{1}{k} \sum_{i=1}^k (h_{\min,i}(A) = h_{\min,i}(B) | h_{\max,i}(A) = h_{\max,i}(B)) = \\ \frac{1}{k} \sum_{i=1}^k (J(A, B) | J(A, B)) = J(A, B) | J(A, B) = J(A, B) \end{aligned} \quad (9)$$

the final three steps follow from Eq. 4 and Eq. 6. Therefore we see that this method also finds the jaccard similarity.

As one may have noted, the difference between $\mathbf{Mm}_{\frac{1}{2}}$ and \mathbf{Mm} is that the first runs only half as many times as the second for each comparison between two sets.