Minwise hashing is a standard technique in the context of search. Numerous practical problems (e.g., duplicate detection, query optimization) can be formulated as set intersection/similarity problems. Minwise hashing is an efficient algorithm for approximating set similarities/intersections.

- Conventional minwise hashing stores each hashed value using 64 bits.
- We suggest to store only the lowest \( b \)-bits, e.g., \( b = 2 \).
- Our method often achieves about 20-fold improvement in storage space and similar improvement in computation, to attain the same accuracy.
- Intuitive idea and practical method, but the analysis is technical.
**Shingling, Resemblance, Minwise Hashing**

**Shingling:** A document (Web page) can be represented by a set of \( w \)-shingles \((w \text{ contiguous words, } w \geq 5)\). The set space is often assumed to be \(|\Omega| = 2^{64}|\).

Document similarity \(\Rightarrow\) Set intersection in ultra-high dimensions.

**Resemblance \((R)\):**
\[
R = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}, \quad S_1, S_2 \in \Omega = \{0, 1, \ldots, 2^{64} - 1\}.
\]

**Minwise Hashing:** Apply a random permutation \(\pi : \Omega \rightarrow \Omega\) on \(S_1, S_2\).

Then, the two minimums are equal with a probability \(R\).

\[
\Pr (\min(\pi(S_1)) = \min(\pi(S_2))) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} = R.
\]
**b-Bit Minwise Hashing (WWW’10)**

Define: $z_1 = \min(\pi(S_1))$, $z_2 = \min(\pi(S_2))$. i.e., $\Pr(z_1 = z_2) = R$

**Theorem:**

\[
\Pr(\text{Lowest } b \text{ bits of } z_1 = \text{Lowest } b \text{ bits of } z_2) = C_{1,b} + (1 - C_{2,b}) R
\]

\[
C_{1,b} = A_{1,b} \frac{|S_2|}{|S_1| + |S_2|} + A_{2,b} \frac{|S_1|}{|S_1| + |S_2|}, \quad C_{2,b} = A_{1,b} \frac{|S_1|}{|S_1| + |S_2|} + A_{2,b} \frac{|S_2|}{|S_1| + |S_2|}.
\]

$A_{1,b} = \text{fun}(|S_1|, b)$ and $A_{2,b} = \text{fun}(|S_2|, b)$ are pre-computed.

**Consequence:**

1. Most favorable case: using $b = 1$ achieves $64$-fold improvement in storage.
2. Least favorable case: using $b = 1$ achieves $21.3$-fold improvement in storage if resemblance $R \geq 0.5$, the common threshold used in practice.
3. Similar improvements in computation time.
b-Bit Minwise Hashing for Estimating Three-Way Similarities

Many problems are multi-way: Word associations/co-occurrences (e.g., search engine query “Machine, Learning, NIPS”); Learning beyond pairwise relations; Tensor algebra, ....

Three-Way Resemblance: \[ R_3 = \frac{|S_1 \cap S_2 \cap S_3|}{|S_1 \cup S_2 \cup S_3|} \]

\textit{b-bit Hashing for estimating } R_3

- The collision probability formula is extremely complicated.
- However, the probability formula is greatly simplified by using data sparsity.
- One must use \( b \geq 2 \) bits for \( R_3 \). (Using \( b = 1 \) leads to \( \infty \) variance.)
- Still significant (e.g., 20-fold) improvement over using 64 bits.