Maximally Consistent Sampling and the Jaccard Index of Probability Distributions

Anonymous
Anonymous

***Abstract*—We introduce simple, efficient algorithms for computing a MinHash of a probability distribution, suitable for both sparse and dense data, with equivalent running times to the state of the art for both cases. The collision probability of these algorithms is a new measure of the similarity of positive vectors which we investigate in detail. We describe the sense in which this collision probability is optimal for any Locality Sensitive Hash based on sampling. We argue that this similarity measure is more useful for probability distributions than the similarity pursued by other algorithms for weighted MinHash, and is the natural generalization of the Jaccard index.**

***Index Terms*—Locality Sensitive Hashing, Retrieval, MinHash, Jaccard index, Jensen-Shannon divergence**

I. Introduction

MinHashing[1] is a popular Locality Sensitive Hashing algorithm for clustering and retrieval on large datasets. Its extreme simplicity and efficiency, as well as its natural pairing with MapReduce and key-value datastores, have made it a basic building block in many domains, particularly document clustering[1][2] and graph clustering[3][4].

Given a finite set U, and a uniformly random permutation π, the map $X \mapsto \arg\min_{i \in X} \pi(i)$ provides a representation of any subset X of U that is stable under small changes to X. If X, Y are both subsets of U the well-known Jaccard index[5] is defined as

$$\mathrm{J}(X,Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

Stability of $\arg\min_{i \in X} \pi(i)$ with respect to X is quantified by

$$\Pr\left[\arg\min_{i \in X} \pi(i) = \arg\min_{i \in Y} \pi(i)\right] = \mathrm{J}(X,Y)$$

Practically, this random permutation is generated by applying some hash function to each i with a fixed random seed, hence "MinHashing."

In order to hash objects other than sets, Chum et al.[6] introduced two algorithms for incorporating weights in the computation of MinHashes. The first algorithm associates constant global weights with the set members, suitable for idf weighting. The collision probability that results is $\frac{\sum_{i \in X \cap Y} w_i}{\sum_{i \in X \cup Y} w_i}$. The second algorithm computes MinHashes of vectors of positive integers, yielding a collision probability of

$$\mathrm{J}_{\mathcal{W}}(x,y) = \frac{\sum_i \min(x_i, y_i)}{\sum_i \max(x_i, y_i)}$$

Subsequent work[7][8][9] has improved the efficiency of the second algorithm and extended it to arbitrary positive weights, while still achieving $\mathrm{J}_{\mathcal{W}}$ as the collision probability.

$\mathrm{J}_{\mathcal{W}}$ is one of several generalizations of the Jaccard index to non-negative vectors. It is useful because it is monotonic with respect to the L_1 distance between x and y when they are both L_1 normalized, but it is unnatural in many ways for probability distributions.

If we convert sets to binary vectors x, y, with $x_i, y_i \in \{0, 1\}$, then $\mathrm{J}_{\mathcal{W}}(x,y) = \mathrm{J}(X,Y)$. But if we convert these vectors to probability distributions by normalizing them so that $x_i \in \left\{0, \frac{1}{|X|}\right\}$, $y_i \in \left\{0, \frac{1}{|Y|}\right\}$, then $\mathrm{J}_{\mathcal{W}}(x,y) \neq \mathrm{J}(X,Y)$ when $|X| \neq |Y|$. The correspondence breaks and it no longer generalizes the Jaccard index. Instead, for $|X| > |Y|$,

$$\mathrm{J}_{\mathcal{W}}(x,y) = \frac{|X \cap Y|}{|X \setminus Y| + |X|} < \mathrm{J}(X,Y). \tag{1}$$

As a consequence, switching a system from an unweighted MinHash to a MinHash based on $\mathrm{J}_{\mathcal{W}}$ will generally decrease the collision probabilities.

Furthermore, $\mathrm{J}_{\mathcal{W}}$ is insensitive to important differences between probability distributions. It counts all differences on an element in the same linear scale regardless of the mass the distributions share on that element. For instance, $\mathrm{J}_{\mathcal{W}}((a,b,c,0),(a,b,0,c)) = \mathrm{J}_{\mathcal{W}}((a+c,b),(a,b+c))$. This makes it a poor choice when the ultimate goal is to measure similarity using an expression based in information-theory or likelihood where having differing support typically results in the worst possible score.

For a drop-in replacement for the Jaccard index that treats its input as a probability distribution, we'd like it to have the following properties.

1) Scale invariant.
2) Not lower than the Jaccard Index when applied to discrete uniform distributions.

Algorithm 1: $\mathcal{P}$-MinHash. We require only a single exponentially distributed hash per nonzero element.

input : Vector $\mathbf{x}$, Seed $\mathbf{s}$
output: Stable sample from $\mathbf{x}$
foreach i **where** $x_i > 0$ **do**
  $e_i \leftarrow \frac{-\log(\text{UniformNonZeroFloat}(i,s))}{x_i}$
end
return $\arg\min_i e_i$

3) Sensitive to changes in support, in a similar way to information-based measures.
4) Easily achievable as a collision probability.

$\mathrm{J}_{\mathcal{W}}$ fails all but the last.

1) It isn't scale invariant, $\mathrm{J}_{\mathcal{W}}(\alpha x, y) \neq \mathrm{J}_{\mathcal{W}}(x, y)$.
2) If the vectors are normalized to make it scale invariant, the values drop below the corresponding Jaccard index (equation 1.)
3) It is insensitive to changes in support.
4) Good algorithms exist, but they are non-trivial.

Existing work has thoroughly explored improvements to Chum et al.'s second algorithm, while leaving their first untouched. In this work we instead take their first algorithm as a starting point. We extend it to arbitrary positive vectors (rather than sets with constant global weights) and analyze the result. In doing so, we find that the collision probability is a new generalization of the Jaccard Index to positive vectors, which we here call $\mathrm{J}_{\mathcal{P}}$.

$$\mathrm{J}_{\mathcal{P}}(x, y) = \sum_{\{i \,:\, x_i, y_i > 0\}} \frac{1}{\sum_j \max\left(\frac{x_j}{x_i}, \frac{y_j}{y_i}\right)}. \quad (2)$$

The names used here, $\mathrm{J}_{\mathcal{W}}$ and $\mathrm{J}_{\mathcal{P}}$, are chosen to reflect how each function interprets x and y, and the conditions under which they match the original Jaccard index. $\mathrm{J}_{\mathcal{W}}$ treats a difference in magnitude the same as any other difference, so treats vectors as "weighted sets." $\mathrm{J}_{\mathcal{P}}$ is scale invariant, so any input is treated the same as a corresponding probability distribution.

The primary contribution of this work is to derive and analyze $\mathrm{J}_{\mathcal{P}}$, and to show that in many situations where the objects being hashed are probability distributions, $\mathrm{J}_{\mathcal{P}}$ is a more useful collision probability than $\mathrm{J}_{\mathcal{W}}$.

We will describe the sense in which $\mathrm{J}_{\mathcal{P}}$ is an optimal collision probability for any LSH based on sampling. We will prove that if the collision probability of a sampling algorithm exceeds $\mathrm{J}_{\mathcal{P}}$ on one pair, it must sacrifice collisions on a pair that has higher $\mathrm{J}_{\mathcal{P}}$.

We will motivate $\mathrm{J}_{\mathcal{P}}$'s utility by showing experimentally that it has a tighter relationship to the Jensen-Shannon divergence than $\mathrm{J}_{\mathcal{W}}$, and is more closely centered around the Jaccard index than $\mathrm{J}_{\mathcal{W}}$. We will even show empirically that in some circumstances, it is better for retrieving documents that are similar under $\mathrm{J}_{\mathcal{W}}$ than $\mathrm{J}_{\mathcal{W}}$ itself (and consequently, sometimes better for retrieving based on L_1-distance.)

II. $\mathcal{P}$-MinHash and its Match Probability

Let $h : [n] \to (0, 1]$ be a pseudo-random hash mapping every element $1 \leq i \leq n$ to an independent uniform random value in $(0, 1]$. Over a non-negative vector x, define

$$H(x) \coloneqq \arg\min_i \frac{-\log h(i)}{x_i}$$

For brevity, we will be using the extended real number system in which $1/\infty \coloneqq 0$. Each term is an exponentially distributed random variable with rate x_i,

$$\Pr\left[\frac{-\log h(i)}{x_i} < y\right] = 1 - e^{-x_i y},$$

so it follows that

$$\Pr\left[H(x) = i\right] = \frac{x_i}{\sum_i x_i}.$$

This well known and beautiful property of exponential random variables derives from the fact that $\Pr\left[\min(x, y) > \alpha\right] = \Pr\left[x > \alpha\right] \Pr\left[y > \alpha\right]$.

Theorem II.1. *For any two nonnegative vectors $x, y \in \mathbb{R}^n_+$,*

$$\Pr\left[H(x) = H(y)\right] = J_{\mathcal{P}}(x, y).$$

Proof. Any monotonic transform on $\frac{-\log h(i)}{x_i}$ will not change the arg min, so multiplying each x_i by a positive α won't either. $H(\alpha x) = H(x)$. Thus for $x_i, y_i > 0$

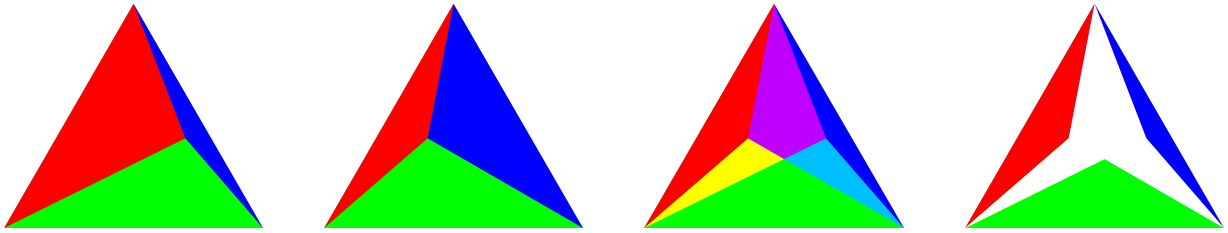
$$\Pr\left[i = H(x) = H(y)\right] = \Pr\left[i = H\left(\frac{x}{x_i}\right) = H\left(\frac{y}{y_i}\right)\right]$$

By definition, $H\left(\frac{x}{x_i}\right) = H\left(\frac{y}{y_i}\right) = i$ means $\frac{-\log h(i)}{x_i/x_i} \leq \frac{-\log h(j)}{x_j/x_i}$ and $\frac{-\log h(i)}{y_i/y_i} \leq \frac{-\log h(j)}{y_j/y_i}$, for $j \neq i$, or equivalently,

$$\begin{aligned} -\log h(i) &\leq \min\left(\frac{-\log h(j)}{x_j/x_i}, \frac{-\log h(j)}{y_j/y_i}\right) \\ &\leq \frac{-\log h(j)}{\max\left(\frac{x_j}{x_i}, \frac{y_j}{y_i}\right)}. \end{aligned}$$

Now we desire a new vector z^i such that $H(z^i) = i$ if and only if $H(x/x_i) = H(y/y_i) = i$. This requires that $\frac{-\log h(j)}{z^i_j/z^i_i} = \frac{-\log h(j)}{\max\left(\frac{x_j}{x_i}, \frac{y_j}{y_i}\right)}$. Thus for fixed i, $z^i_j = \frac{\max\left(\frac{x_j}{x_i}, \frac{y_j}{y_i}\right)}{\sum_k \max\left(\frac{x_k}{x_i}, \frac{y_k}{y_i}\right)}$. Consequently,

$$\Pr\left[H(z^i) = i\right] = \frac{1}{\sum_j \max\left(\frac{x_j}{x_i}, \frac{y_j}{y_i}\right)}$$



(a) $\Pr[H(x) = i]$ (b) $\Pr[H(y) = i]$ (c) $\Pr[H(x), H(y) = i, j]$ (d) $\Pr[H(x) = H(y) = i]$

Fig. 1: The $\mathcal{P}$-MinHash process can be interpreted geometrically as dividing the unit simplex into smaller simplexes proportional to the mass of each term, then selecting the same point in each simplex as its representative. Pictured here are $x = (0.5, 0.4, 0.1), y = (0.2, 0.4, 0.4)$. Colors are assigned to unique values of i, or (i, j) in the case of the joint distributions. The sum of the remaining colored areas in (d) is proportional to $\mathrm{J}_{\mathcal{P}}(x, y)$.

Repeating this process for all i in the intersection yields

$$\Pr[H(x) = H(y)] = \sum_i \frac{1}{\sum_j \max\left(\frac{x_j}{x_i}, \frac{y_j}{y_i}\right)} \quad \square$$

Continuing the notational convention, we will refer to hashing algorithms that achieve $\mathrm{J}_{\mathcal{P}}$ as their pair collision probability as $\mathcal{P}$-MinHashes, and algorithms that achieve $\mathrm{J}_{\mathcal{W}}$ as $\mathcal{W}$-MinHashes.

III. Intuition about $\mathrm{J}_{\mathcal{P}}$

While $\mathrm{J}_{\mathcal{P}}$'s expression is superficially awkward, we can aid intuition by representing it in other ways. The simplest interpretation is to view it as a variant of $\mathrm{J}_{\mathcal{W}}$. We rewrite $\mathrm{J}_{\mathcal{W}}$ allowing one input to be rescaled before computing each term,

$$\mathrm{J}_{\mathcal{W}}(x, y, \alpha) = \sum_i \frac{\min(\alpha_i x_i, y_i)}{\sum_j \max(\alpha_i x_j, y_j)}$$

and choose the vector α to maximize this generalized $\mathrm{J}_{\mathcal{W}}$. If $\alpha_i x_i > y_i$, increasing α_i raises only the denominator. If $\alpha_i x_i < y_i$, increasing α_i raises the numerator more than the denominator. So the optimal α sets $\alpha_i x_i = y_i$, and results in $\mathrm{J}_{\mathcal{P}}$.

We can derive a more powerful representation by viewing the $\mathcal{P}$-MinHash algorithm itself geometrically. A vector of $k + 1$ exponential random variables, when normalized to sum to 1, is a uniformly distributed random point on the unit k-simplex. Every point in the unit k-simplex is also a probability distribution over $k+1$ elements. Using these two facts we can construct the $\mathcal{P}$-MinHash as a function of the simplex as illustrated in Figure 1.

For a probability distribution x, mark the point on the unit simplex corresponding to x, $(x_1, \ldots, x_{k+1})$, and connect it to each of the corners of the simplex. These edges divide it into $k + 1$ smaller simplices that fill the unit simplex. Each internal simplex has volume proportional to the coordinate of x opposite to its unique exterior face. As a result, a uniformly chosen point on the unit simplex will land in one of the sub-simplices with probability given by x. $\mathcal{P}$-MinHashing is equivalent to sampling in this fashion, but holding the chosen point constant when sampling from each new distribution. The match probability is then proportional to the sum of the intersections of simplices that share an external face. This representation makes several properties obvious on small examples that we prove generally in the next section.

IV. Optimality of $\mathrm{J}_{\mathcal{P}}$

When MinHashing is used with a key-value store, high collision probabilities are generally more efficient than low collision probabilities, because as we discuss in section VI, it is much cheaper to lower them than to raise them. For this reason, we are interested in the question of the highest collision probability that can be achieved through sampling. The constraint that the samples follow each distribution forces the collision probability to remain discriminative, but given that constraint, we would like to make it as high as possible to maximize flexibility and efficiency.

Suppose for two distributions, x and y, we want to choose a joint distribution that maximizes $\Pr[H(x) = H(y)]$. If we were concerned with only these two particular distributions in isolation, the upper bound of $\Pr[H(x) = H(y)]$ is given by the Total Variation distance, or equivalently $1 - \mathrm{L}_1(x, y)/2$. Meeting this bound requires the probability mass where x exceeds y to be perfectly coupled with the mass where y exceeds x. Both the mass they share and the mass they do not must be perfectly aligned.

Rather than just two, we want to create a joint distribution (or coupling) of all possible distributions on a given space where the collision probability for any pair is as high as possible. It is always possible to increase the collision probability of one pair at the expense of

another so long as the chosen pair has not hit the Total Variation limit, so the kind of optimality we are aiming for is Pareto optimality. This requires that no collision probability be able to exceed ours everywhere; any gain on one pair must have a corresponding loss on another.

This by itself would not be a very consequential bound for its retrieval performance. We really only desire high collision probabilities for items that are similar, and we would happily lower the collision probability of a dissimilar pair to increase it for a similar pair.

However we are able to prove something stronger by examining the pair whose collisions must be sacrificed. To increase the collision probability for one pair above its $\mathrm{J}_{\mathcal{P}}$, you must always sacrifice collisions on a pair with even higher $\mathrm{J}_{\mathcal{P}}$. To get better recall on one pair, you must always give up recall on an even better pair.

The Jaccard index itself is optimal on uniform distributions, and the short proof is a model for the general case.

Theorem IV.1. *No method of sampling from uniform discrete distributions can have collision probabilities dominating the Jaccard index of their supports. The Jaccard index is Pareto optimal.*

Proof. Let Z be the symmetric difference of X and Y, $Z = X \cup Y - X \cap Y$, and let z, x, y be the corresponding discrete uniform distributions. The three intersections, $X \cap Y,\ X \setminus Y,\ Y \setminus X$, are disjoint, so the three collision events, $H(x) = H(y)$, $H(z) = H(x)$, and $H(z) = H(y)$, are also disjoint. As disjoint events, their probabilities must sum to at most 1. When the three probabilities are given by the Jaccard index of the corresponding sets, they already sum to 1, respectively, $\frac{|X \cap Y|}{|X \cup Y|}$, $\frac{|X \setminus Y|}{|X \cup Y|}$, $\frac{|Y \setminus X|}{|X \cup Y|}$, so the Jaccard index is Pareto optimal. □

To prove the same claim on $\mathrm{J}_{\mathcal{P}}$ for all distributions, we need a few tools. We can rearrange $\mathrm{J}_{\mathcal{P}}$ to separate the two iteration indices within the max.

$$\mathrm{J}_{\mathcal{P}}(x, y) = \sum_i \frac{x_i}{\sum_j y_j \max\left(\frac{x_j}{y_j}, \frac{x_i}{y_i}\right)} \tag{3}$$

This lets us characterize the $\arg\max$ choices in the denominator as functions of a sorted list of x_i/y_i where $\frac{x_i}{y_i} \geq \frac{x_{i+1}}{y_{i+1}}$. If we reindex i according to this sorted list, then we can describe each $\mathrm{J}_{\mathcal{P}}(x, y)_i$ knowing that i terms choose the left side of the max, and $n-i$ terms choose the right. (This also lets us compute $\mathrm{J}_{\mathcal{P}}$ in $O(n \log n)$ rather than $O(n^2)$ time by sorting first and keeping running sums of each choice of the max.)

To analyze $\mathrm{J}_{\mathcal{P}}$ we will need to refer to each term in its outer sum via subscript. $\mathrm{J}_{\mathcal{P}}(x, y)_i := \frac{1}{\sum_j \max\left(\frac{x_j}{x_i}, \frac{y_j}{y_i}\right)} = z_i^i$. We will also use quantifiers in this subscript to indicate a partial sum, i.e. $\mathrm{J}_{\mathcal{P}}(x, y)_{i>a} := \sum_{\{i:i>a\}} \mathrm{J}_{\mathcal{P}}(x, y)_i$.

Lemma IV.2. *Useful tools from sorting $J_{\mathcal{P}}$'s indices:*

1) *For at least two values of i, and distributions x, y, $J_{\mathcal{P}}(x, y)_i = \min(x_i, y_i)$.*
2) *Let $w_1 ... w_n$ be distributions with disjoint support. Consider two linear combinations of these distributions with coefficients α and β. $J_{\mathcal{P}}(\alpha \cdot w, \beta \cdot w) = J_{\mathcal{P}}(\alpha, \beta)$*

Proof. For a given i, if every $\max$ chooses the same side, $\mathrm{J}_{\mathcal{P}}(x, y)_i = \min(x_i, y_i)$. x_i/y_i has both an $\arg\max$ and $\arg\min$ for which this is true. This gives us part 1.

Let $x_k/y_k = x_l/y_l$.

$$\sum_{i \in \{k,l\}} \frac{x_i}{\sum_j y_j \max\left(\frac{x_j}{y_j}, \frac{x_i}{y_i}\right)} = \frac{x_k + x_l}{\sum_j y_j \max\left(\frac{x_j}{y_j}, \frac{x_k}{y_k}\right)}$$

$$\sum_{j \in \{k,l\}} y_j \max\left(\frac{x_j}{y_j}, \frac{x_i}{y_i}\right) = (y_k + y_l) \max\left(\frac{x_k}{y_k}, \frac{x_i}{y_i}\right)$$

Thus, we can form x', y' by merging the mass of x_k, x_l into one element and y_k, y_l into one element and have $\mathrm{J}_{\mathcal{P}}(x, y) = \mathrm{J}_{\mathcal{P}}(x', y')$ Let $w_1 ... w_n$ be distributions with disjoint support, and consider $\mathrm{J}_{\mathcal{P}}(\alpha \cdot w, \beta \cdot w)$. Repeat the merging process until all elements of each w_i are merged into one. This gives us (2). □

This ordering also lets us work more effectively with the z distributions we constructed in theorem II.1. This lemma contains all the algebra needed for the main proof.

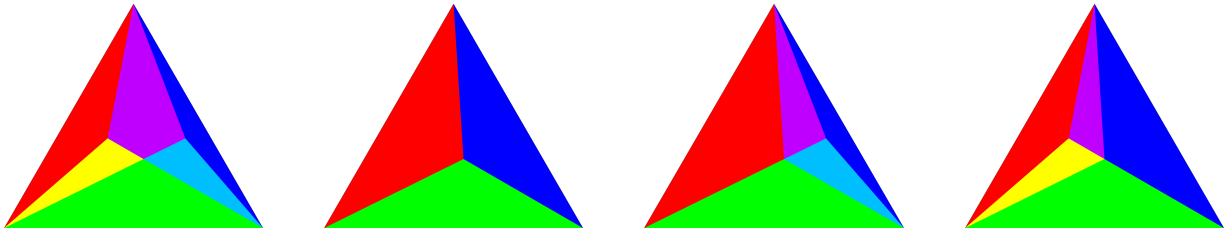
Lemma IV.3. *For fixed a, let z^a, x, y be probability distributions where $z_i^a \propto \max\left(\frac{x_i}{x_a}, \frac{y_i}{y_a}\right)$. Reorder i according to the sorting of (3) such that $\frac{x_i}{y_i} \geq \frac{x_{i+1}}{y_{i+1}}$. The following are true:*

1) $\forall i \leq a,\ J_{\mathcal{P}}(x, z^a)_i = z_i^a = \min(x_i, z_i^a) \geq z_i^i$.
2) $\forall i \geq a,\ J_{\mathcal{P}}(x, z^a)_i = z_i^i$.
3) $J_{\mathcal{P}}(x, y)_i = \min(x_i, y_i) \implies J_{\mathcal{P}}(x, z^a)_i = \min(x_i, z_i^a)$.
4) $J_{\mathcal{P}}(x, z^a) \geq J_{\mathcal{P}}(x, y)$ *and* $J_{\mathcal{P}}(y, z^a) \geq J_{\mathcal{P}}(x, y)$

Proof. For $i \leq a$, $\frac{x_i}{y_i} \geq \frac{x_a}{y_a} \Rightarrow \frac{x_i}{x_a} \geq \frac{y_i}{y_a}$, hence $z_i^a \propto \frac{x_i}{x_a}$. Similarly, for $i \geq a$, $z_i^a \propto \frac{y_i}{y_a}$. Working first with the lower group of indices,

$$\forall i \leq a,\ \mathrm{J}_{\mathcal{P}}(x, z^a)_i = \frac{1}{\sum_j \max\left(\frac{x_j}{x_i}, \frac{\max\left(\frac{x_j}{x_a}, \frac{y_j}{y_a}\right)}{x_i/x_a}\right)}$$

$$= \frac{x_i/x_a}{\sum_j \max\left(\frac{x_j}{x_a}, \frac{y_j}{y_a}\right)} = z_i^a$$



(a) $\Pr[H(x), H(y) = i, j]$ (b) $\Pr[H(z^{\text{g}}) = i]$ (c) $\Pr[H(x), H(z^{\text{g}}) = i, j]$ (d) $\Pr[H(y), H(z^{\text{g}}) = i, j]$

Fig. 2: A visual proof of theorem IV.4 on 3 element disributions. Using the same distributions as in Figure 1, the green regions of x and y could be shifted to overlap more and improve the collision probability of this pair, but any modification that achieves that would worsen at least one of collision probabilities between x or y and z^{green} (both of which have higher collision probability than the (x, y) pair.)

And since $\frac{x_i/x_a}{\sum_j \max(\frac{x_j}{x_a}, \frac{y_j}{y_a})} \leq \frac{x_i/x_a}{\sum_j \frac{x_j}{x_a}}$ we also know that $\forall i \leq a,\ z_i^a = \min(x_i, z_i^a)$. Furthermore, $\forall i \leq a$,

$$z_i^a = \frac{1}{\sum_j \max\left(\frac{x_j/x_a}{\max(\frac{x_i}{x_a}, \frac{y_i}{y_a})}, \frac{y_j/y_a}{\max(\frac{x_i}{x_a}, \frac{y_i}{y_a})}\right)}$$
$$= \frac{1}{\sum_j \max\left(\min\left(\frac{x_j}{x_i}, \frac{x_j y_a}{x_a y_i}\right), \min\left(\frac{y_j}{y_i}, \frac{y_j x_a}{y_a x_i}\right)\right)} \geq z_i^i$$

which gives us part 1. Now, continuing on to the upper group of indices,

$$\forall i \geq a,\ \mathrm{J}_{\mathcal{P}}(x, z^a)_i = \frac{1}{\sum_j \max\left(\frac{x_j}{x_i}, \frac{\max(\frac{x_j}{x_a}, \frac{y_j}{y_a})}{y_i/y_a}\right)}$$
$$= \frac{1}{\sum_j \max\left(\frac{x_j}{x_i}, x_j \frac{y_a}{x_a, y_i}, \frac{y_j}{y_i}\right)}$$

Since $\forall i \geq a,\ \frac{x_a y_i}{y_a} \geq x_i$ we can simplify further to conclude part 2. $\forall i \geq a,\ \mathrm{J}_{\mathcal{P}}(x, z^a)_i = z_i^i$. By noting that the choices within the max are preserved, we conclude part 3. Finally, having bounded all indices, part 4:

$$\forall i,\ \mathrm{J}_{\mathcal{P}}(x, z^a)_i \geq \mathrm{J}_{\mathcal{P}}(x, y)_i$$
$$\mathrm{J}_{\mathcal{P}}(x, z^a) \geq \mathrm{J}_{\mathcal{P}}(x, y). \quad \square$$

We now have the tools to prove the optimality of $\mathrm{J}_{\mathcal{P}}$.

Theorem IV.4. *Let G be a sampling method. If $\Pr[G(x) = G(y) = i] > J_{\mathcal{P}}(x, y)_i$, there exists a z such that $\Pr[G(x) = G(z)] < J_{\mathcal{P}}(x, z)$ or $\Pr[G(y) = G(z)] < J_{\mathcal{P}}(y, z)$. This implies that no method of sampling from discrete distributions can have collision probabilities dominating $J_{\mathcal{P}}$. $J_{\mathcal{P}}$ is Pareto optimal.*

Furthermore, $J_{\mathcal{P}}(x, z) \geq J_{\mathcal{P}}(x, y)$ and $J_{\mathcal{P}}(y, z) \geq J_{\mathcal{P}}(x, y)$. To exceed $J_{\mathcal{P}}(x, y)$, G must sacrifice at least one pair that is closer under $J_{\mathcal{P}}$ than (x, y).

Proof. Let m be the number of elements i for which $\mathrm{J}_{\mathcal{P}}(x, y)_i < \min(x_i, y_i)$.

In the base case where $m = 0$, $\mathrm{J}_{\mathcal{P}}(x, y) = \sum_i \min(x_i, y_i)$ which cannot be improved.

Assume the proposition to be proved is true $\forall x, \forall y$, and $\forall p < m$. By IV.2.1 we know that $m \leq n - 2$, since at least the two endpoints of the sorted list have reached their upper bound. We proceed by induction on m.

As in theorem II.1, for each a where $\mathrm{J}_{\mathcal{P}}(x, y)_a < \min(x_a, y_a)$, construct a distribution z^a, where $z_i^a \propto \max\left(\frac{x_i}{x_a}, \frac{y_i}{y_a}\right)$. Reorder i according to the sorting of (3) such that $\frac{x_i}{y_i} \geq \frac{x_{i+1}}{y_{i+1}}$. From lemma IV.3 we know $\forall i \leq a,\ \mathrm{J}_{\mathcal{P}}(x, z^a)_i = z_i^a = \min(x_i, z_i^a)$ and $\forall i \geq a,\ \mathrm{J}_{\mathcal{P}}(y, z^a)_i = z_i^a = \min(y_i, z_i^a)$.

Now consider a new sampling method, G. The following events are pairwise disjoint by inspection.

$$E_a^< := \{G(x) = G(z^a) < a\}$$
$$E_a^= := \{G(x) = G(y) = a\}$$
$$E_a^> := \{G(y) = G(z^a) > a\}$$

So their probabilities are constrained. $\Pr[E_a^=] + \Pr[E_a^<] + \Pr[E_a^>] \leq 1$. When these probabilities are given by $\mathrm{J}_{\mathcal{P}}$ they already sum to 1. $\Pr[E_a^=] + \Pr[E_a^<] + \Pr[E_a^>] = z_a^a + \sum_{i<a} z_i^a + \sum_{i>a} z_i^a = 1$.

From this bound, if $\Pr[G(x) = G(y) = a] > \mathrm{J}_{\mathcal{P}}(x, y)_a$, at least one of

$$\Pr[G(x) = G(z^a) < a] < \mathrm{J}_{\mathcal{P}}(x, z^a)_{i<a}$$
$$\Pr[G(y) = G(z^a) > a] < \mathrm{J}_{\mathcal{P}}(y, z^a)_{i>a}$$

must be true. Since the two cases are symmetric, we will assume the first one: $\Pr[G(x) = G(z^a) < a] < \mathrm{J}_{\mathcal{P}}(x, z^a)_{i<a}$.

If $\Pr[G(x) = G(z^a)] < \mathrm{J}_{\mathcal{P}}(x, z^a)$ then our z is z^a and we are done. Otherwise, $\Pr[G(x) = G(z^a)] \geq \mathrm{J}_{\mathcal{P}}(x, z^a)$, and the terms $i \geq a$ must compensate for the loss on the terms $i < a$, so $\Pr[G(x) = G(z^a) \geq a] > \mathrm{J}_{\mathcal{P}}(x, z^a)_{i \geq a}$.

However, $\mathrm{J}_{\mathcal{P}}(x, z^a)_a = \mathrm{J}_{\mathcal{P}}(y, z^a)_a = z_a^a$, so these terms have exhausted z^a and cannot be increased. Using

IV.3.1 and IV.3 we know that this adds at least one additional term that is fully consumed, i.e. the size of $\{i : \mathrm{J}_{\mathcal{P}}(x, z^a)_i < \min(x_i, z_i^a)\}$ is less than m. So by induction hypothesis, if $\Pr[G(x) = (z^a) < a] > \mathrm{J}_{\mathcal{P}}(x, z^a)_{i<a}$ there is a new z for which $\Pr[G(x) = G(z)] < \mathrm{J}_{\mathcal{P}}(x, z)$ or $\Pr[G(z^a) = G(z)] < \mathrm{J}_{\mathcal{P}}(z^a, z)$.

By IV.3 we know that $\mathrm{J}_{\mathcal{P}}(x, z^a) \geq \mathrm{J}_{\mathcal{P}}(x, y)$ and by induction we know $\mathrm{J}_{\mathcal{P}}(x, z) \geq \mathrm{J}_{\mathcal{P}}(x, z^a)$, so we conclude $\mathrm{J}_{\mathcal{P}}(x, z) \geq \mathrm{J}_{\mathcal{P}}(x, y)$ and symmetrically $\mathrm{J}_{\mathcal{P}}(y, z) \geq \mathrm{J}_{\mathcal{P}}(x, y)$. □

Figure 2 shows the mechanism of the proof intuitively. On three element distributions, two of the elements are fully constrained, in this case the blue and red terms, so we construct our adversarial z around green. On three elements, no induction is necessary and the diagram itself proves the relationship.

Since $\mathrm{J}_{\mathcal{P}}$ is only Pareto optimal, we should be able to find a sampling method that exceeds it for some pairs but is below it on others. We can generalize our algorithm to construct such a method. Consider arranging the elements of the state space as the leaves of a tree. Internal nodes in the tree are given the weight of the sum of their children, and each is assigned its own exponential hash. Perform $\mathcal{P}$-MinHash among the children of the root node. If the selected node is a leaf, emit it as the sample. If it is an internal node, recurse and repeat. In this generalization, our original algorithm is represented by making all elements direct children of the root. We can prioritize collisions on an index i by placing it closer to the root node than all others; in particular, the tree $(i, (1, \ldots, n))$. Since i and its internal-node sibling form a two element distribution, by lemma 3 the probability of a collision on i will be $\min(x_i, y_i)$ for all x, y.

What of $\mathrm{J}_{\mathcal{W}}$ then? Is it another Pareto optimum? It is not, it is dominated by $\mathrm{J}_{\mathcal{P}}$.

Theorem IV.5. *If $J_{\mathcal{W}}(x, y) = \frac{1-p}{1+p}$, then $\frac{1-p}{1+p} \leq J_{\mathcal{P}}(x, y) \leq 1-p$, and there are distributions that achieve both bounds on $J_{\mathcal{P}}$ for any value of $J_{\mathcal{W}}$.*

Proof. The lower bound becomes clear by rewriting $\mathrm{J}_{\mathcal{W}}$ in a similar form to $\mathrm{J}_{\mathcal{P}}$.

$$\mathrm{J}_{\mathcal{W}}(x, y) = \sum_i \frac{1}{\sum_j \frac{\max(x_j, y_j)}{\min(x_i, y_i)}}$$

$$\sum_i \frac{1}{\sum_j \max\left(\frac{x_j}{x_i}, \frac{y_j}{y_i}\right)} \geq \sum_i \frac{1}{\sum_j \max\left(\frac{x_j}{x_i}, \frac{y_j}{y_i}, \frac{x_j}{y_i}, \frac{y_j}{x_i}\right)}$$

$$\mathrm{J}_{\mathcal{P}}(x, y) \geq \mathrm{J}_{\mathcal{W}}(x, y)$$

To achieve this lower bound, we can transform the distributions by moving the "excess" mass to new elements.

$$x'_{2i} = y'_{2i} = \min(x_i, y_i)$$
$$x'_{2i+1} = \max(x_i - y_i, 0), \quad y'_{2i+1} = \max(y_i - x_i, 0)$$

Shifting the mass in this way has no effect on $\mathrm{J}_{\mathcal{W}}$, but it decreases $\mathrm{J}_{\mathcal{P}}$ to equal $\mathrm{J}_{\mathcal{W}}$. x' and y' can be expressed as linear combinations over the three sets of indices, so using lemma IV.2.2, $\mathrm{J}_{\mathcal{P}}(x', y') = \mathrm{J}_{\mathcal{P}}((0, 1-p, p), (p, 1-p, 0)) = \frac{1-p}{1+p}$

To achieve the upper bound, $1 - p$, consider inverting this transformation, reallocating the p extra mass to maximize $\mathrm{J}_{\mathcal{P}}(x'', y'')$ while holding $\mathrm{J}_{\mathcal{W}}(x'', y'')$ constant. To avoid increasing $\mathrm{J}_{\mathcal{W}}$, we must add the mass to disjoint elements, so divide the indices into two sets, X, Y. We find that if we distribute the mass proportional to the original value, $\mathrm{J}_{\mathcal{P}}$ reaches the total variation limit regardless of the choice of X, Y. Let $|X| \coloneqq \sum_{i \in X} \min(x_i, y_i)$.

$$\forall i \in X, \;\; x''_i = \min(x_i, y_i) \frac{|X| + p}{|X|}, \;\; y''_i = \min(x_i, y_i)$$
$$\forall i \in Y, \;\; y''_i = \min(x_i, y_i) \frac{|Y| + p}{|Y|}, \;\; x''_i = \min(x_i, y_i)$$

We can express this as a linear combination of two distributions with disjoint support. $\mathrm{J}_{\mathcal{P}}(x'', y'') = \mathrm{J}_{\mathcal{P}}((|X| + p, |Y|), (|X|, |Y| + p)) = 1 - p$. Since p is the total variation distance of x and y, $1 - p$ is the maximum collision probability that is possible between two distributions in any context, so it is the upper bound here as well. □

This gives us some insight into how $\mathrm{J}_{\mathcal{P}}$ and $\mathrm{J}_{\mathcal{W}}$ differ. $\mathrm{J}_{\mathcal{P}}$ ranks distributions as more similar than $\mathrm{J}_{\mathcal{W}}$ if their extra mass is on elements that both distributions share.

Like $1 - \mathrm{J}_{\mathcal{W}}$, $1 - \mathrm{J}_{\mathcal{P}}$ is a metric on probability distributions.

Theorem IV.6. *$1 - J_{\mathcal{P}}$ is a proper metric on $\mathcal{P}$ where $\mathcal{P}(\Omega)$ is the space of probability distributions over a finite set Ω.*

Proof. Symmetry is obvious. Non-degeneracy over $\mathcal{P}$ follows from $\frac{1}{\sum_{j \in \Omega} \max\left(\frac{x_j}{x_i}, \frac{y_j}{y_i}\right)} \leq \min(x_i, y_i)$. The triangle inequality follows from being a collision probability, $1 - \mathrm{J}_{\mathcal{P}}(x, y) = \Pr[H(x) \neq H(y)]$. Therefore,

$$\Pr[H(x) = H(y)] \geq \Pr[H(x) = H(z) \wedge H(y) = H(z)]$$
$$\Pr[H(x) \neq H(y)] \leq \Pr[H(x) \neq H(z) \vee H(y) \neq H(z)]$$

for any distribution z. But by the union bound,

$$\Pr\left[H(x) \neq H(z) \vee H(y) \neq H(z)\right] \leq \Pr\left[H(x) \neq H(z)\right] + \Pr\left[H(y) \neq H(z)\right] \quad \square$$

V. Hashing on Dense and Continuous Data

The algorithm we've presented so far is suitable for sparse data such as documents or graphs. It is linear in the number of non-zeros, equivalent to Ioffe 2010 [7]. On dense data (such a image feature histograms[8]) there's significant overlap in the supports of each distribution, so rehashing each element for every distribution wastes

Algorithm 2: Dense and Continuous $\mathcal{P}$-MinHash. "Global-Bound" A* Sampling [10] with a fixed seed.

input : sample space Ω,
sigma-finite measure μ,
proposal sigma-finite measure λ,
finite upper bound, $B \coloneqq \max(\mu(i)/\lambda(i))$
shared random seed s
output: Stable sample from $(\Omega, \mathcal{F}, \mu)$
$(M, X^*, k, e_{-1}) \leftarrow (\inf, \text{null}, 0, 0)$
do
  $e_k \leftarrow -\log(\text{UniformNonZeroFloat}(k, s)) + e_{k-1}$
  $X_k \leftarrow \text{Sample}(\lambda(\Omega), s)$
  $M_k \leftarrow e_k \lambda(X_k)/\mu(X_k)$
  if $M_k < M$ **then**
    $M \leftarrow M_k$
    $X^* \leftarrow X_k$
  end
  $\Omega \leftarrow \Omega \setminus X_k$
  $k \leftarrow k + 1$
while $M > e_k/B$
return X^*

work. With a shared stream of sorted hashes, we expect the hash we select for each distribution to be biased towards the beginning of the stream, and closer to the beginning when the data is denser. Therefore one might expect that we could improve performance by searching only some prefix of the stream to find our sample.

A* Sampling (Maddison, Tarlow, and Minka 2014[10]) explores this idea thoroughly, and we lean on it heavily in this section. In particular, we use their "Global Bound" algorithm, and essentially just run it with a fixed random seed (algorithm 2.) We leave the proof of running time and of correctness as a sampling method to that work, and limit our discussion to the proof of the resulting collision probability. (Their derivation uses Gumbel variables and maxima. We use exponential variables and minima to make the continuity with the rest of our work clear, which is achieved by a simple change of variables.)

The key insight of A* Sampling is that when a (possibly infinite) stream of independent exponential random variables is ordered, the exact marginal distributions of each variable can be computed as a function of the rank and the previous variables. If the vector of sorted exponential variables is e with corresponding parameter vector x, then once $e_1, \ldots, e_{k-1}$ are all known, the distribution of e_k is a truncated exponential with rate $|x \setminus \bigcup_{i=1}^{k} x_i|$ truncated from below at e_{k-1}. The "statelessness" of exponential variables makes this truncation easy to accomplish. Simply generate the desired exponential, and add e_{k-1} to shift it.

To change the parameters of those exponentials and find the new minimum element, only a small prefix of the list must be examined. The running time of finding the new minimum is a function of the difference between the two vectors of parameters, and has equivalent running time to rejection sampling. This gives algorithm 2 equivalent running time to the state of the art for computing a MinHash of dense data, Shrivastava 2016[8]. Because algorithm 2 admits an unbounded list of random variables, it is also applicable to continuous distributions, as the paper[10] describes in detail.

Let's first show that algorithm 2 gives $\mathrm{J}_{\mathcal{P}}(\mu, \nu)$ when Ω is *finite*. Indeed, this construction is simply an alternative way of finding the minimum $-\log U_i/\mu_i$. A* sampling merely reads off the minimum of $-\log U_i/\lambda_i * \lambda_i/\mu_i = -\log U_i/\mu_i$, and similar for ν.

To prove the general case for infinite Ω, we need to first define what we mean by $\mathrm{J}_{\mathcal{P}}(\mu, \nu)$ in that setting. One option is to replace all the summation by integrals in the formula (2). This runs into two difficulties however:

1) A probability space Ω may not be a subset of $\mathbb{R}^n$.
2) Either μ or ν could be singular.

Instead, we define it as a limit over increasingly finer finite partitions of Ω. More formally,

Definition V.1. *Assume $J_{\mathcal{P}}(\mu, \nu)$ is defined as before when $|\Omega| < \infty$, we define*

$$J_{\mathcal{P}}(\mu, \nu) = \inf_{\mathcal{F} \vdash \Omega} J_{\mathcal{P}}(\mu_{\mathcal{F}}, \nu_{\mathcal{F}}), \tag{4}$$

where $\mathcal{F}$ ranges over finite partitions of the space Ω, and $\mu_{\mathcal{F}}$ denotes the push-forward of μ with respect to the map $\pi : \Omega \to \mathcal{F}$, $\pi(x) = Q \in \mathcal{F}$ iff $x \in Q$. ($\mu_{\mathcal{F}}$ is simply a coarsified probability measure on the finite space $\mathcal{F}$ where it (tautologically) assigns probability $\mu(Q)$ to the element $Q \in \mathcal{F}$.)

First we verify that the definition above coincides with $\mathrm{J}_{\mathcal{P}}$ when Ω is finite.

Lemma V.2. *Let $|\Omega'| + 1 = |\Omega| = n$, $\mu, \nu \in \mathcal{P}(\Omega)$, and $\mu', \nu' \in \mathcal{P}(\Omega')$ obtained by merging the last two elements of Ω into a single element. Then $J_{\mathcal{P}}(\mu, \nu) \leq J_{\mathcal{P}}(\mu', \nu')$, with strict inequality if both μ, ν have nonzero masses on those two elements.*

Proof. By considering $\mathrm{J}_{\mathcal{P}}(\mu, \nu)$ as the probability that the argmin's of two lists of independent exponentials land on the same index $1 \leq i_* \leq n$, and using the fact that the minimum of two independent exponentials is an exponential with the sum of the parameters, we can couple the four argmin's arising from μ, ν, μ', ν' and conclude by inspection. □

The lemma shows that any partition of Ω will lead to a $\mathrm{J}_{\mathcal{P}}$ that's greater than or equal to the original $\mathrm{J}_{\mathcal{P}}$. So the infimum is achieved with the most refined partition, namely Ω itself.

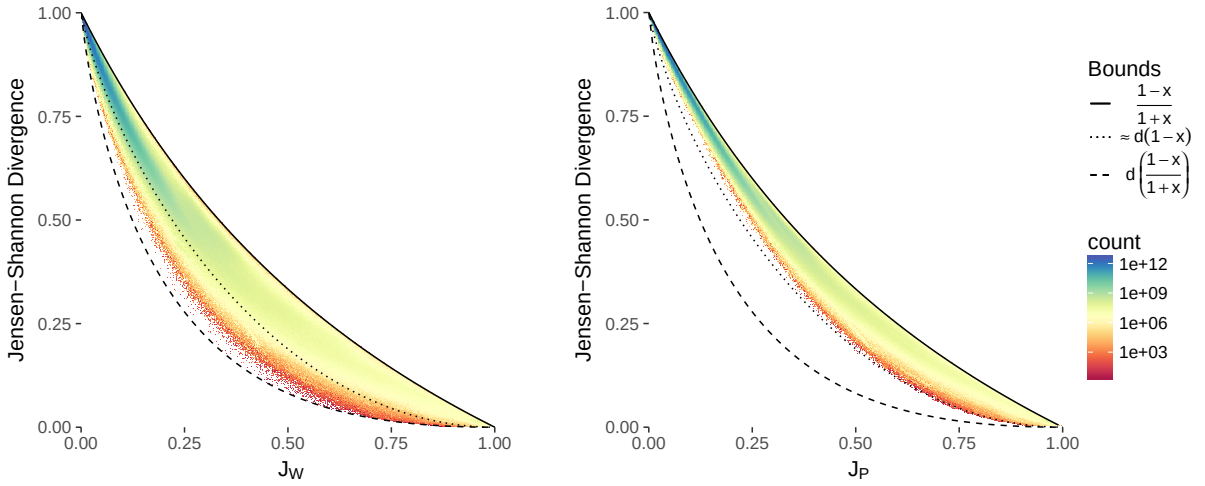


Fig. 3: The Jensen-Shannon divergence compared with $J_{\mathcal{P}}$ and $J_{\mathcal{W}}$ on pairs of normalized unigram term vectors of random web documents. JSD has a much tighter relationship with $J_{\mathcal{P}}$ than $J_{\mathcal{W}}$. We show exact bounds for $J_{\mathcal{W}}$ against JSD and approximate bounds for $J_{\mathcal{P}}$ against JSD where $d(x) = \frac{(1-x)}{2}\log_2(1-x) + \frac{(1+x)}{2}\log_2(1+x)$. The curve that appears to lower bound JSD against $J_{\mathcal{P}}$ is violated on 10^{-7} of the pairs.

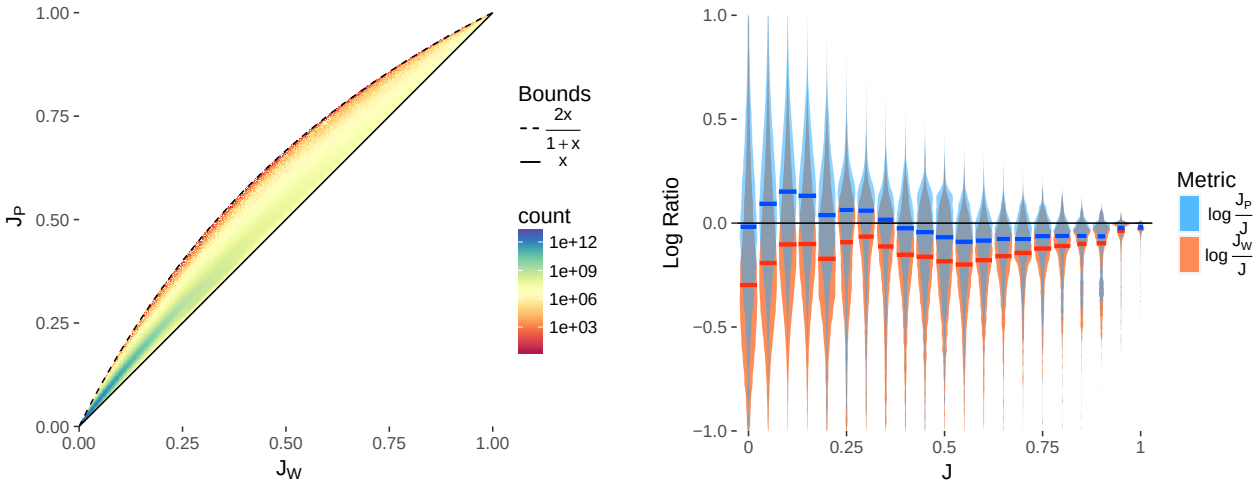


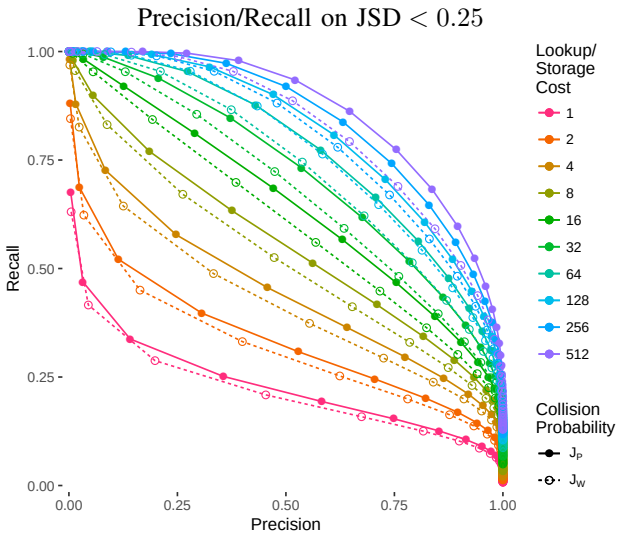
Fig. 4: Comparison of Jaccard measures on normalized unigram term vectors of pairs of random web documents. The left graph shows the joint distribution of $J_{\mathcal{P}}$ and $J_{\mathcal{W}}$ and the bounds we prove. The right graph shows the conditional distributions of $J_{\mathcal{P}}$ and $J_{\mathcal{W}}$ against the (set) Jaccard index of the terms. We show the distribution of the log of their ratios against the Jaccard index and highlight the median. $J_{\mathcal{P}}$ is generally centered around the Jaccard index, while $J_{\mathcal{W}}$ is consistently centered below, as predicted by their behavior on uniform distributions.

Finally we show that A* sampling applied to μ and ν simultaneously has a collision probability equal to $J_{\mathcal{P}}(\mu, \nu)$ as defined above.

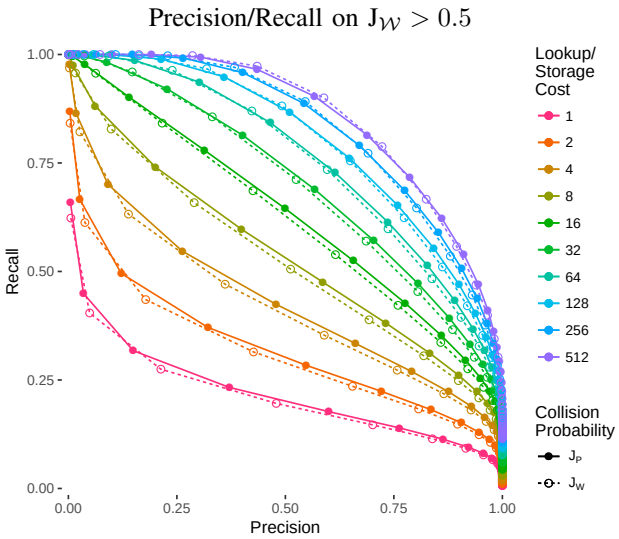
Theorem V.3. *Given two probability measures μ and ν on an arbitrary Polish space Ω, both absolutely continuous with respect to a common third measure λ, it is possible to apply A* sampling with base distribution λ, either in-order or with a hierarchical partition of Ω, to sample from μ and ν simultaneously. Further, the probability of the procedure terminating at the same point $p \in \Omega$ for both μ and ν is exactly $J_{\mathcal{P}}(\mu, \nu)$.*

Proof. The first statement follow from the procedural definition of A* sampling described in [10]. For the second statement, since in-order A* is proven equivalent to hierarchical partition A* in [10], we are free to choose any partition to our convenience. The natural choice is then the partition used in the definition of $J_{\mathcal{P}}(\mu, \nu)$.

More precisely, we know there is a finite partition $\mathscr{F} \vdash \Omega$ such that $J_{\mathcal{P}}(\mu, \nu) \leq J_{\mathcal{P}}(\mu_{\mathscr{F}}, \nu_{\mathscr{F}}) \leq J_{\mathcal{P}}(\mu, \nu) + \epsilon$, for any $\epsilon > 0$. On the other hand, for finite partition like $\mathscr{F}$, the exponential variables attached to the



(a) Performance on retrieving pairs with Jensen-Shannon divergence less than 0.25. $J_{\mathcal{P}}$ performs better both by having a tighter relationship with JSD, and by having higher match probabilities overall, making high recall cheaper to achieve. $\mathcal{P}$-MinHash achieves the same performance with 64 hashes that a $\mathcal{W}$-MinHash achieves with 128.



(b) Performance on retrieving pairs with $J_{\mathcal{W}}$ greater than 0.5. The fact that $\mathcal{P}$-MinHash slightly outperforms $\mathcal{W}$-MinHashes illustrates the benefit of higher match probabilities. $J_{\mathcal{P}}$ is higher than $J_{\mathcal{W}}$ for pairs that are similar under $J_{\mathcal{W}}$, which makes high recall cheaper. As the cost increases, $J_{\mathcal{W}}$ overtakes $J_{\mathcal{P}}$ as the difference between the two scores becomes the larger factor.

Fig. 5: Precision/recall curves illustrating the typical case of retrieval using a key-value store. Each point represents outputting o independent sums of a hashes each, for a collision probability of $1-(1-p^a)^o$. The cost in storage and CPU is dominated by o, so we connect these points to show the trade-offs possible at similar cost.

representative of each part $Q \in \mathscr{F}$ are jointly distributed as exponentials with rate $\lambda(Q)$ with common seed. Let U, V be the two coupled A* processes restricted to $\mathscr{F}$. Either one of them does not terminate, or they both terminate and collide conditionally with probability $J_{\mathcal{P}}(\mu_{\mathscr{F}}, \nu_{\mathscr{F}})$. In other words, letting $\mathbb{P}(T)$ be the probability that both terminate at $\mathscr{F}$ level, and $AC(U,V;\mathscr{F})$ be the collision probability of U, V restricted to $\mathscr{F}$, then $AC(U,V;\mathscr{F}) = \mathbb{P}(T)J_{\mathcal{P}}(\mu_{\mathscr{F}}, \nu_{\mathscr{F}})$. Thus $J_{\mathcal{P}}(\mu,\nu)\mathbb{P}(T) \leq AC(U,V;\mathscr{F}) \leq J_{\mathcal{P}}(\mu_{\mathscr{F}}, \nu_{\mathscr{F}}) \leq J_{\mathcal{P}}(\mu,\nu) + \epsilon$.

So $AC(U,V;\mathscr{F})$ is squeezed between $J_{\mathcal{P}}(\mu,\nu)\mathbb{P}(T)$ and $J_{\mathcal{P}}(\mu,\nu) + \epsilon$. Since $\mathbb{P}(T) \to 1$ and $\epsilon \to 0$ under a refinement sequence $\mathscr{F}$, we get in the limit

$$AC(U,V) := AC(U,V;\mathscr{F}_\infty) = J_{\mathcal{P}}(\mu,\nu). \quad \square$$

VI. Utility of $J_{\mathcal{P}}$ on Pairs of Web Documents

To determine whether the difference between $J_{\mathcal{P}}$ and $J_{\mathcal{W}}$ matters in practice and whether achieving $J_{\mathcal{P}}$ as a collision probability is a useful goal, we computed both for a large sample of pairs of unigram term vectors of web documents. From an index of 6.6 billion documents, we selected pairs using a sum of 2 $\mathcal{W}$-MinHashes to perform importance sampling. We computed several similarity scores for 100 million pairs of normalized unigram term vectors, and weighted them by the inverse of their sampling probability to simulate an unbiased sample of all non-zero pairs.

The Jensen-Shannon divergence (JSD) defines the information loss that results from representing two distributions using a model that is an equal mixture of them, and as such is the ideal criterion to form information-preserving clusters of items of equal importance. Like both $J_{\mathcal{W}}$ and $J_{\mathcal{P}}$ it is bounded, symmetric, and monotonic in a metric distance. Due to these properties, as well as its popularity, we use it here as a basis for comparison.

$J_{\mathcal{P}}$ has a much tighter relationship with the Jensen-Shannon divergence than $J_{\mathcal{W}}$ as shown in figure 3. Tight bounds on JSD as a function of $J_{\mathcal{W}}$ are given by $J_{\mathcal{W}}$'s monotonic relationship with Total Variation, as described by [11]. Let p be the total variation distance, and $d(p) = \frac{(1-p)}{2}\log_2(1-p) + \frac{(1+p)}{2}\log_2(1+p)$. $d(p) \geq \text{JSD}(x,y) \geq p$. Substituting $\frac{1-J_{\mathcal{W}}}{1+J_{\mathcal{W}}} = p$ extends these to $J_{\mathcal{W}}$. These same bounds apply to $J_{\mathcal{P}}$ as well, but $J_{\mathcal{P}}$ has a much tighter relationship with what appears to be a much higher lower bound.

We have approached finding this lower bound with large differential equations that we have only solved numerically, but small examples form good approximate bounds. On 2 element distributions, JSD has a direct relationship with $J_{\mathcal{P}}$, and only 1×10^{-7} of the pairs fall below the resulting curve, $d(1-J_{\mathcal{P}})$. No pairs in our

sample had JSD more than 0.0077 below it. In contrast, $J_{\mathcal{W}}$ puts 7×10^{-3} of the pairs below this curve, with the farthest point 0.16 below.

We also compare both $J_{\mathcal{P}}$ and $J_{\mathcal{W}}$ to the Jaccard index of the set of terms, and compute a kernel density estimate of the log of their ratios. $J_{\mathcal{P}}$ is generally centered around the Jaccard index, while $J_{\mathcal{W}}$ is consistently centered below, as predicted by their behavior on uniform distributions. This makes $\mathcal{P}$-MinHash less disruptive as a drop-in replacement for an unweighted MinHash. Parameters of the system such as the number of hashes or the length of concatenated hashes are likely to continue to function well.

In the typical case of retrieval using a key-value store, performance is characterized by cheap ANDs and expensive ORs.[12] To reduce the collision probability we can sum multiple hashes to form keys, but to raise the collision probability, we must output multiple independent keys. This lets us apply an asymmetric sigmoid to the collision probabilities, $1 - (1 - p^a)^o$ with o independent outputs of a summed hashes each. Assuming that the cost of looking up hashes dominates the cost of generating them, ANDs are essentially free, while CPU and storage cost are both linear in the number of ORs. Furthermore, as a increases linearly, o must increase exponentially to keep the inflection point of the sigmoid in the same place. For instance, if the sigmoid passes through $(0.5, 0.5)$, then $o \approx \log(2)2^a$. This gives a significant performance advantage to algorithms with higher collision probabilities, and thus to $J_{\mathcal{P}}$ over $J_{\mathcal{W}}$. Lowering the probability is much cheaper than raising it.

The effect of this is demonstrated in Figure 5. Unsurprisingly from the tightness of the joint distribution, $\mathcal{P}$-MinHash achieves better precision and recall retrieving low JSD documents for a given cost. More surprising is that it also achieves slightly better precision and recall on retrieving high $J_{\mathcal{W}}$ documents when the cost is low, even though this is the task $\mathcal{W}$-MinHashes are designed for. The reason for this can be seen from the upper bound, $J_{\mathcal{P}} \leq 2J_{\mathcal{W}}/(1 + J_{\mathcal{W}})$. On items that achieve this bound, the collision probability when summing two hashes, $(2x/(1+x))^2$, is similar to $4x^2$ near 0 and similar to x near 1. This in effect gives it the recall of 1 hash with the precision of 2 hashes on this subset of items, and thus a better precision/recall trade-off overall.

VII. Conclusion

We've described a new generalization of the Jaccard index, and shown several qualities that motivate it as the natural extension to probability distributions. In particular, we proved that it is optimal on all distributions in the same sense that the Jaccard index is optimal on uniform distributions. We've demonstrated its utility by showing $J_{\mathcal{P}}$'s similarity in practice to the Jensen-Shannon divergence, a popular clustering criterion. We've described two MinHashing algorithms that achieve this as their collision probability with equivalent running time to the state of the art on both sparse and dense data.

References

[1] A. Z. Broder, "On the resemblance and containment of documents," in *Compression and Complexity of Sequences 1997. Proceedings*, IEEE, 1997, pp. 21–29.

[2] ——, "Filtering near-duplicate documents," in *Proc. FUN 98*, Citeseer, 1998.

[3] D. Gibson, R. Kumar, and A. Tomkins, "Discovering large dense subgraphs in massive graphs," in *Proceedings of the 31st international conference on Very large data bases*, VLDB Endowment, 2005, pp. 721–732.

[4] S.-J. Lee, S. Kang, H. Kim, and J.-K. Min, "An efficient parallel graph clustering technique using pregel," in *Big Data and Smart Computing (BigComp), 2016 International Conference on*, IEEE, 2016, pp. 370–373.

[5] P. Jaccard, "Distribution de la flore alpine dans le bassin des dranses et dans quelques regions voisines," in *Bulletin de la Socit Vaudoise des Sciences Naturelles 37*, 1901, pp. 241–272.

[6] O. Chum, J. Philbin, A. Zisserman, *et al.*, "Near duplicate image detection: Min-hash and tf-idf weighting.," in *BMVC*, vol. 810, 2008, pp. 812–815.

[7] S. Ioffe, "Improved consistent sampling, weighted minhash and l1 sketching," in *ICDM*, 2010.

[8] A. Shrivastava, "Simple and efficient weighted minwise hashing," in *Advances in Neural Information Processing Systems*, 2016, pp. 1498–1506.

[9] M. Manasse, F. McSherry, and K. Talwar, "Consistent weighted sampling," Tech. Rep., Jun. 2010. [Online]. Available: https://www.microsoft.com/en-us/research/publication/consistent-weighted-sampling/.

[10] C. J. Maddison, D. Tarlow, and T. Minka, "A* sampling," in *Advances in Neural Information Processing Systems*, 2014, pp. 3086–3094.

[11] I. Sason, "Tight bounds for symmetric divergence measures and a refined bound for lossless source coding," *IEEE Transactions on Information Theory*, vol. 61, no. 2, pp. 701–707, 2015.

[12] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proceedings of the twentieth annual symposium on Computational geometry*, ACM, 2004, pp. 253–262.