In Defense of MinHash over SimHash

Anshumali Shrivastava

Dept. of Computer Science

Cornell University

anshu@cs.cornell.edu

Ping Li

Dept. of Statistics & Biostatistics

Dept. of Computer Science

Rutgers University

pingli@stat.rutgers.edu

Sparse Binary High Dimensional Data Everywhere

- Wide adoption of the "Bag of Words" (BoW) representations for documents and images.
- When using higher shingles, most of the shingles only occur at most once.
- Most information in the sparsity structure rather than the magnitude.
- Modern "Big data" systems use binary data matrix $n \times D$, with both n and D easily running into billions and even trillions (e.g SIBYL).

Notation

A binary (0/1) vector $\Longleftrightarrow$ a set (locations of nonzeros).

Consider two sets $W_1, W_2 \subseteq \Omega = \{0, 1, 2, ..., D-1\}$ (e.g., $D = 2^{64}$)

$f_1 = |W_1|, \qquad f_2 = |W_2|, \qquad a = |W_1 \cap W_2|.$

The **resemblance** $\mathcal{R}$ and **cosine similarity** $\mathcal{S}$ are two popular measures adopted in practice.

$$\mathcal{R} = \frac{|W_1 \cap W_2|}{|W_1 \cup W_2|} = \frac{a}{f_1 + f_2 - a}.$$

$$\mathcal{S} = \frac{|W_1 \cap W_2|}{\sqrt{|W_1||W_2|}} = \frac{a}{\sqrt{f_1 f_2}}.$$

LSH and Sub-linear Near Neighbor Search

Locality Sensitive Hashing (LSH) function families $\mathcal{H}$, satisfies $Pr_{h\in\mathcal{H}}(h(x) = h(y)) = F(sim(x, y))$, where F is a monotonically increasing function and $sim(x, y)$ is the similarity of interest between x and y.

Sub-Linear Near Neighbor Bucketing Algorithm

- For each point x, generate a hash key by concatenating K hash signatures $g(x)$ = $\{h_1(x), h_2(x), ..., h_K(x)\}$, where each $h_i(x)$ drawn independently, and store data point x in a hashtable at location $g(x)$
- For a given query point q, retrieve elements from the bucket $g(q)$.
- Repeat L times independently. Smart choices of L, K lead to worst case approximate query time of O(n^ρ) where $\rho < 1$. (Adoni-Indyk 08)
- ρ a property of H, the smaller the better

The Two Popular LSH in Practice

MinHash for resemblance Suppose a random permutation π is performed on Ω, i.e., $\pi : \Omega \longrightarrow \Omega$ An elementary probability argument shows that

$$\mathbf{Pr}\left(\text{min}(\pi(W_1)) = \text{min}(\pi(W_2))\right) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} = \mathcal{R}.$$

SimHash for cosine similarity,

$$h_r(x) = \begin{cases} 1 & \text{if } r^T x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

where $r \in R^d$ drawn independently from $N(0, \mathcal{I})$ The seminal work of Geomens-Williamson showed that $Pr(h(x) = h(y)) = 1 - \frac{1}{\pi}\cos^{-1}(\mathcal{S})$

The Main Questions

- Which among the two hash functions, MinHash or SimHash, should be preferred for modern web datasets which are binary and sparse ?
- The two hash function are in the context of different similarity measures, is it even possible to compare them theoretically ?

Our Answers

For binary sparse datasets MinHash is provably a better hash function than SimHash even when the desired similarity measure is cosine similarity!!.

- Yes, it turns out that we can compare the two hash functions theoretically even though they are meant for different similarity measures.
- For binary datasets, the preferred choice of hash function is MinHash, and it is independent of whether the similarity measure is resemblance or cosine similarity.

Key Connection: For binary data resemblance and cosine similarity are distortions of each other.

Worst Case Analysis

Worst Case Distortion: $$\mathcal{S}^2 \leq \mathcal{R} \leq \frac{\mathcal{S}}{2-\mathcal{S}}$$

The bounds are tight over continuous functions. MinHash can be shown as a provable LSH for cosine similarity. MinHash and SimHash can be compared !!. We compare their ρ values for retrieving with cosine similarity.

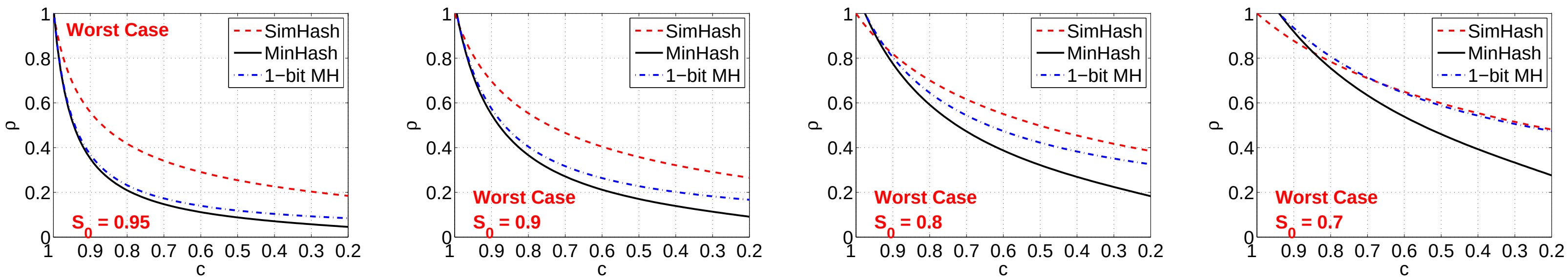


Figure 1: Worst case ρ values of different hash functions; lower is better.

Real Datasets

$$z = z(r) = \sqrt{r} + \frac{1}{\sqrt{r}} \geq 2, \qquad r = \frac{f_2}{f_1} \geq 0$$

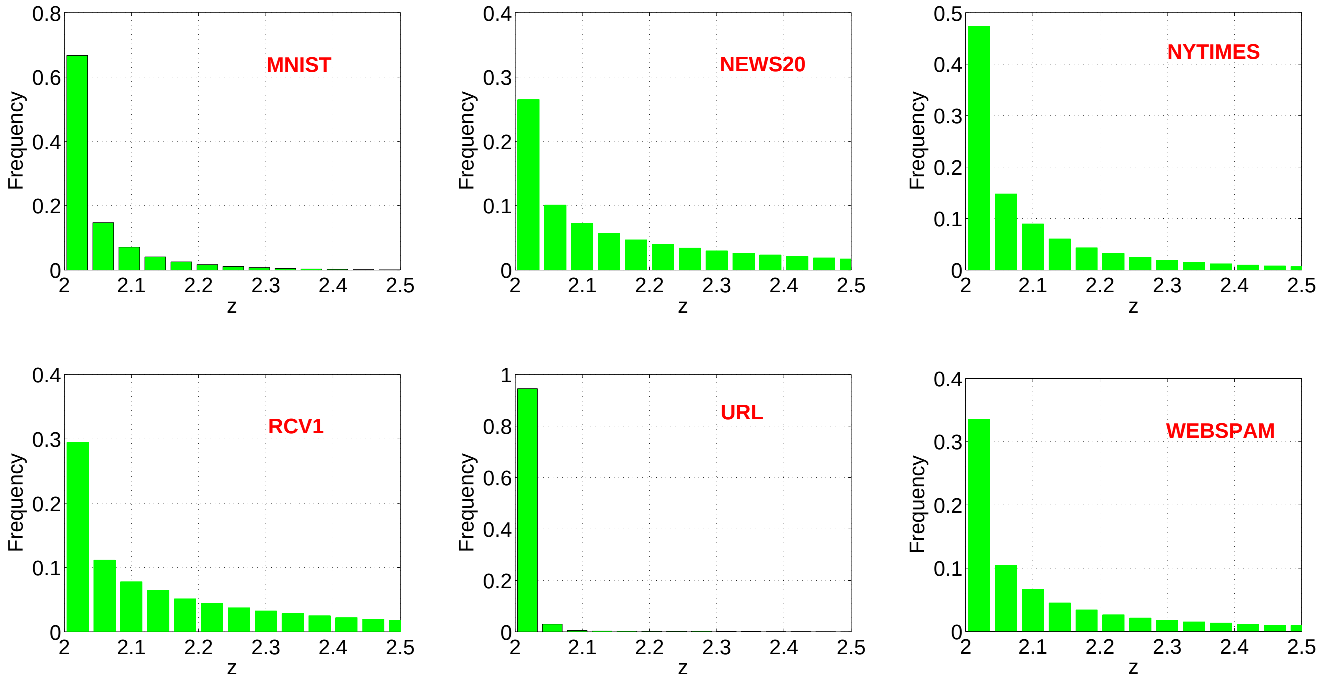


Figure 2: Frequencies of the z values for the six real datasets used in paper

Restricted Worst Case Analysis

Distortion in Practice: $$\frac{\mathcal{S}}{z-\mathcal{S}} \leq \mathcal{R} \leq \frac{\mathcal{S}}{2-\mathcal{S}}$$

z lies roughly between 2 and 2.3. Even for low similarity regions, we observe superior ρ values with MinHash compared to SimHash.

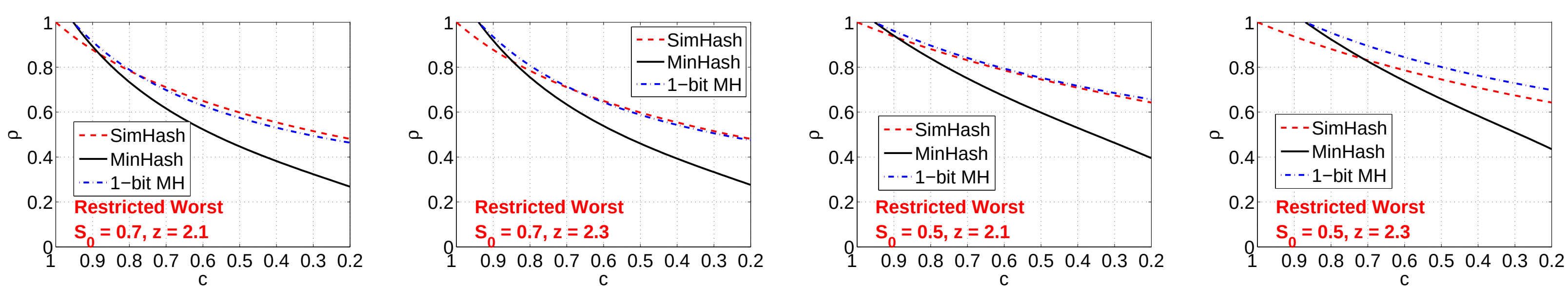
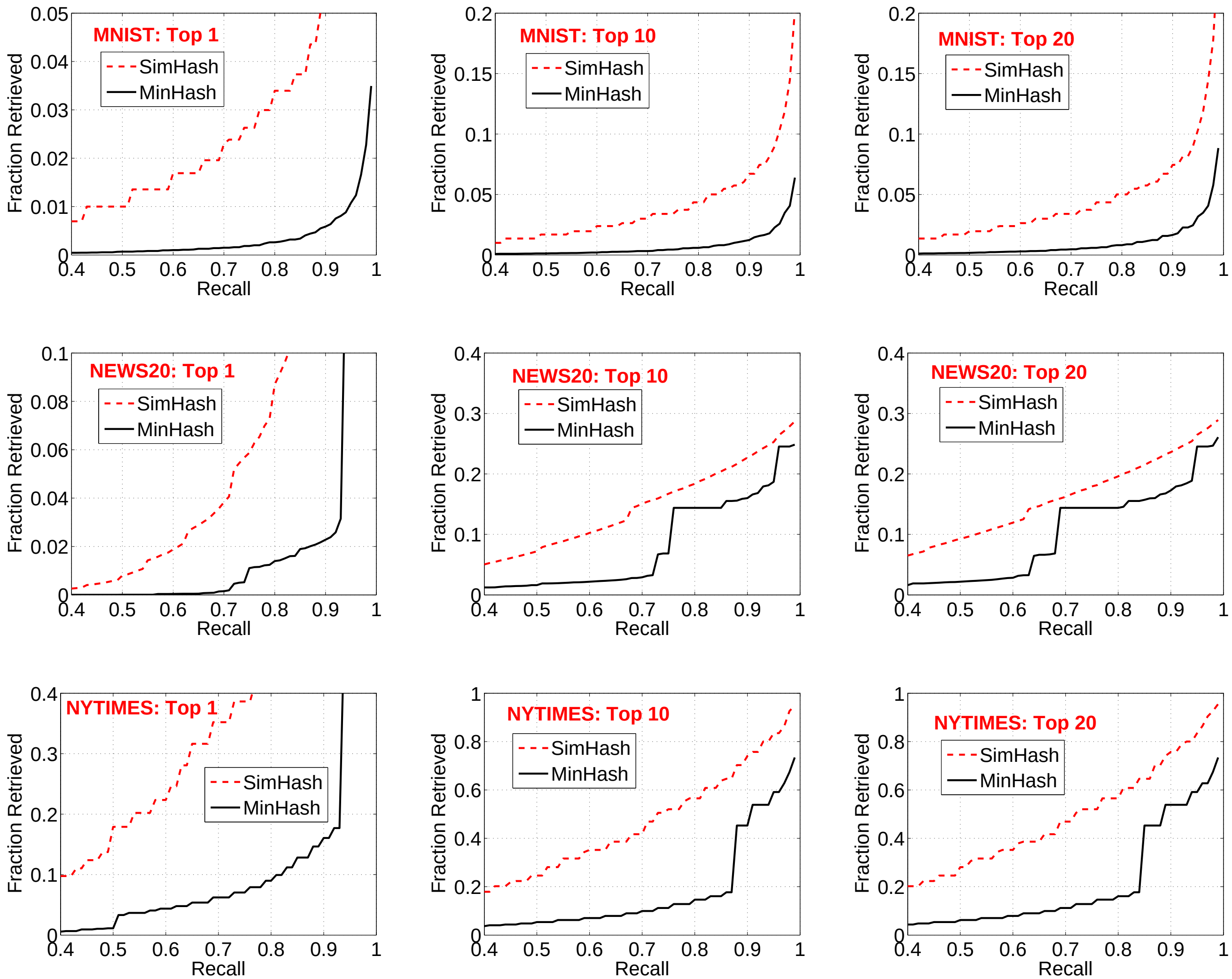


Figure 3: Restricted worst case ρ values of different hash functions; lower is better.

Performance on Near Neighbor Retrieval Task



Performance on Near Neighbor Retrieval Task