Quantization Methods for Inner Product Sketch for Binary Data

Chenyi Li*
cl7201@nyu.edu
New York University

Yurong Liu*
yl6624@nyu.edu
New York University

*Equal contribution

Abstract

In this study, we investigate sketching and quantization techniques for estimating the inner product of binary vectors. We examine established methods such as the Johnson-Lindenstrauss Projection [1], MinHash Sketch [2], and Priority Sampling [3], applying them in a quantization context. Our key contribution is the introduction of 1-bit MinHash for binary vector inner product estimation, which surpasses the performance of traditional quantization approaches like SimHash [4] both theoretically and empirically. Furthermore, we explore the cross-correlation problem, leveraging Locality-Sensitive Hashing to enhance computational efficiency. Our code is available at https://github.com/lyrain2001/sketching.

1 Introduction

In this project, we examine and propose various sketching techniques incorporating quantization for estimating the inner product of vectors. Specifically, we focus on binary vectors. We delve into the problem as defined below:

Definition 1. *Given two binary vectors* $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$*, our goal is to compute their compact sketches, denoted as* $\mathcal{S}(\mathbf{a})$ *and* $\mathcal{S}(\mathbf{b})$*. These sketches should require less than* n *unit space for storage. The expected value of* W *should be equal to the inner product of* $\mathbf{a}$ *and* $\mathbf{b}$ *i.e.,*

$$\mathbb{E}[W] = \langle \mathbf{a}, \mathbf{b} \rangle,$$

where W *is derived from* $\mathcal{S}(\mathbf{a})$ *and* $\mathcal{S}(\mathbf{b})$*.*

Current methodologies include linear sketching, of which an example is Johnson-Lindenstrauss Projection as discussed in [1], as well as sampling methods including Weighted MinHash Sketch [2] and Priority Sampling [3].

While existing variants of linear sketching, such as SimHash [4] and quantized JL[5] methods, do compress sketches by thresholding or rounding, none of the works mentioned above truly utilize quantization strategies. This project, therefore, aims to critically evaluate the most effective quantization approaches to further minimize the sketch size across MinHash Sketch.

Additionally, to expand the application of sketching techniques to binary vectors, we delve into the computation of cross-correlation. While inner product estimators may not be suitable for this purpose, our investigation focuses on the utilization of Locality-Sensitive Hashing (LSH) in conjunction with SimHash and 1-bit MinHash. The objective is to explore the potential of these methods in enhancing computational efficiency in the context of cross-correlation.

To summarize, the key contributions of our work are as follows:

- A comprehensive evaluation of the JL, SimHash and Uniform Priority Sampling methods in estimating the inner product of binary vectors (§ 2).

- The development, analysis, and implementation of a 1-bit MinHash Sketch approach for binary vector inner product estimation, demonstrating superior performance compared to both SimHash and Uniform Priority Sampling (§ 3, 4).
- Application of SimHash LSH and MinHash LSH methods aimed at streamlining cross-correlation computations in Neuropixels data. Preliminary findings suggest these methods, while promising, exhibit limitations in accuracy and efficiency, underscoring the need for further methodological refinement. (§ 5).

These contributions aim to push the boundaries of current sketching techniques for inner product estimation and to provide practical solutions for handling high-dimensional data in various fields.

2 Prior Works

In the subsequent sections, we provide a summary of existing works that serve as the foundation for our project. Note that all the proofs presented are independently written by ourselves.

2.1 Johnson-Lindenstrauss Projection

The Johnson-Lindenstrauss (JL) Projection [1] is a pivotal technique in dimensionality reduction. This method, grounded in the famous Johnson-Lindenstrauss lemma, operates by projecting high-dimensional data vectors into a lower-dimensional space while preserving the distances between these vectors with high probability. Details of JL for inner product estimation can be found in Algorithm 1. A theoretical accuracy guarantee for inner product estimation based on JL is:

Theorem 1. *Let $\Pi \in \mathbb{R}^{m\times n}$ be a random matrix with each entry set independently to $+\sqrt{1/m}$ or $-\sqrt{1/m}$ with equal probability. Let $\mathcal{S}(\mathbf{a}) = \Pi\mathbf{a}$ and $\mathcal{S}(\mathbf{b})$ for vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$. Then if we choose $k = O\left(\frac{\log(1/\delta)}{\epsilon^2}\right)$, we have with probability $1-\delta$,*

$$|\langle \mathcal{S}(\mathbf{a}), \mathcal{S}(\mathbf{b})\rangle - \langle \mathbf{a}, \mathbf{b}\rangle| \leq \epsilon ||\mathbf{a}|| ||\mathbf{b}||,$$

where $\epsilon, \delta \in (0, 1)$.

Algorithm 1 JL

Input: Length n vector a, Length n vector b, random seed s, target sketch size m.
Output: Estimate of $\langle \mathbf{a}, \mathbf{b}\rangle$

1: Use random seed s to generate a random matrix $\Pi \in \{-\mathbf{1}, \mathbf{1}\}^{m\times n}$
2: $\Pi \leftarrow \sqrt{\frac{1}{m}}\Pi$
3: $\mathcal{S}(\mathbf{a}) \leftarrow \Pi(\mathbf{a})$, $\mathcal{S}(\mathbf{b}) \leftarrow \Pi(\mathbf{b})$
4: **return** $\langle \mathcal{S}(\mathbf{a}), \mathcal{S}(\mathbf{b})\rangle$

Recent research has concentrated on examining the effects of quantization on the distortion introduced by the Johnson-Lindenstrauss (JL) transform, aiming to minimize the bit requirement for encoding the transformed data [5]. Initial investigations predominantly addressed binary mappings, exemplified by 1-bit Compressed Sensing [6, 7]. However, contemporary studies have shifted towards a broader spectrum of uniform quantization [5], transcending

Algorithm 2 SimHash

Input: Length n vector a, Length n vector b, random seed s, target sketch size m.
Output: Estimate of $\langle \mathbf{a}, \mathbf{b} \rangle$

1: Use random seed s to generate a random matrix $\Phi \sim \mathcal{N}^{m \times n}(0, 1)$.
2: $\mathcal{S}(\mathbf{a}) \leftarrow \text{sign}(\Phi \mathbf{a})$, $\mathcal{S}(\mathbf{b}) \leftarrow \text{sign}(\Phi \mathbf{b})$.
3: $d_{\mathcal{H}} \leftarrow \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}[S(\mathbf{a})_i \neq S(\mathbf{b})_i]$.
4: **return** $\cos(\pi d_{\mathcal{H}}) \cdot ||\mathbf{a}||_2 \cdot ||\mathbf{b}||_2$.

mere sign-based projection to encompass a more detailed encoding, thus offering a refined representation of the original data.

A critical deviation in this area from the traditional JL Lemma involves the formulation of novel distance metrics, particularly highlighting the introduction of an additive error in relation to angular distance. This advancement marks a pivotal evolution in dimensionality reduction strategies, reflecting the field's adaptation to emerging computational demands and the increasing complexity of datasets.

2.2 SimHash

The use of random hyperplane-based hash functions for vector similarity estimation, pioneered by Charikar [4], forms a foundational basis for quantized hashing techniques.

This approach aims to estimate the angle between two vectors to facilitate inner product estimation. It is accomplished by projecting the vectors onto a randomly chosen Gaussian vector and noting the sign of the resultant inner product.

Let $\Phi \sim \mathcal{N}(0, 1)^{m \times n}$ be a Guassian random matrix, then we define $S : \mathbb{R}^n \to \{\pm 1\}^m$ as

$$S(\mathbf{a}) = \text{sign}(\Phi(\mathbf{a})). \tag{1}$$

The normalized Hamming distance between binary vectors $\mathbf{a}, \mathbf{b} \in \{\pm 1\}^m$ is defined as

$$d_{\mathcal{H}} = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}[S(\mathbf{a})_i \neq S(\mathbf{b})_i], \tag{2}$$

which represents the normalized number of different bits between two binary vectors, where we use the notation $\mathbb{1}[S(\mathbf{u})_i \neq S(\vec{)}_i]$ to denote the indicator random variable that $S(\mathbf{u})_i \neq S(\vec{)}_i$. Now to suitably match the distance $d_{\mathcal{H}}(S(\mathbf{a}), S(\mathbf{b}))$ with the original vectors, we use the normalized angle formed by $\mathbf{a}$ and $\mathbf{b}$, i.e.

$$\frac{\theta}{\pi} = \frac{1}{\pi} \arccos \langle \mathbf{a}, \mathbf{b} \rangle. \tag{3}$$

where $\theta = \arccos \langle \mathbf{a}, \mathbf{b} \rangle$.

Lemma 2.

$$\mathbb{P}[sign(\Phi)_i(\mathbf{a}) \neq sign(\Phi)_i(\mathbf{b})] = \frac{1}{\pi} \arccos \langle \mathbf{a}, \mathbf{b} \rangle \tag{4}$$

Proof. By symmetry,

$$\mathbb{P}[[\text{sign}(\Phi_i(\mathbf{a})) \neq \text{sign}(\Phi_i(\mathbf{b}))]] = 2 \cdot \mathbb{P}[\Phi_i(\mathbf{a}) \geq 0, \Phi_i(\mathbf{b}) \leq 0] = 2 \cdot \frac{\theta}{2\pi} = \frac{\theta}{\pi}.$$

□

Therefore, we can claim

Claim 3. *Let* $\theta = \arccos\langle \mathbf{a}, \mathbf{b}\rangle$*, then*

$$\mathbb{E}\left[\mathbb{1}[sign(\Phi_i(\mathbf{a})) \neq sign(\Phi_i(\mathbf{b}))]\right] = \frac{\theta}{\pi}, \tag{5}$$

and

$$\mathrm{Var}\left[\mathbb{1}[sign(\Phi_i(\mathbf{a})) \neq sign(\Phi_i(\mathbf{b}))]\right] = \frac{\theta}{\pi}\left(1 - \frac{\theta}{\pi}\right). \tag{6}$$

Proof. Equation (5) directly follows from Claim 6, then since indicator variable $\mathbb{1}[\mathrm{sign}(\Phi_i(\mathbf{a})) \neq \mathrm{sign}(\Phi_i(\mathbf{b}))]$ is either 0 or 1, we have

$$\begin{aligned}
&\mathrm{Var}\left[\mathbb{1}[\mathrm{sign}(\Phi_i(\mathbf{a})) \neq \mathrm{sign}(\Phi_i(\mathbf{b}))]\right] \\
&= \mathbb{E}\left[([\mathbb{1}[\mathrm{sign}(\Phi_i(\mathbf{a})) \neq \mathrm{sign}(\Phi_i(\mathbf{b}))])^2\right] - (\mathbb{E}\left[\mathbb{1}[\mathrm{sign}(\Phi_i(\mathbf{a})) \neq \mathrm{sign}(\Phi_i(\mathbf{b}))]\right])^2 \\
&= \frac{\theta}{\pi} - \left(\frac{\theta}{\pi}\right)^2 = \frac{\theta}{\pi}\left(1 - \frac{\theta}{\pi}\right),
\end{aligned}$$

which gives us Equation (6). □

Now since

$$d_{\mathcal{H}} = \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}[S(\mathbf{a})_i \neq S(\mathbf{b})_i] = \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}[\mathrm{sign}(\Phi_i(\mathbf{a})) \neq \mathrm{sign}(\Phi_i(\mathbf{b}))],$$

we can write

$$\mathbb{E}\left[d_{\mathcal{H}}\right] = \mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m} \mathbb{1}[\mathrm{sign}(\Phi_i(\mathbf{a})) \neq \mathrm{sign}(\Phi_i(\mathbf{b}))]\right] = \frac{\theta}{\pi}, \tag{7}$$

and

$$\mathrm{Var}\left[d_{\mathcal{H}}\right] = \mathrm{Var}\left[\frac{1}{m}\sum_{i=1}^{m} \mathbb{1}[\mathrm{sign}(\Phi_i(\mathbf{a})) \neq \mathrm{sign}(\Phi_i(\mathbf{b}))]\right] = \frac{1}{m}\cdot\frac{\theta}{\pi}\left(1 - \frac{\theta}{\pi}\right). \tag{8}$$

This variance implies that when $\theta = 0$, $\mathrm{Var}\left[d_{\mathcal{H}}\right] = 0$, which leads to zero error in estimation.

Then it follows that

$$\mathbb{E}\left[cos(\pi \cdot d_{\mathcal{H}}(\mathbf{a}, \mathbf{b})|) \cdot ||\mathbf{a}||_2 \cdot ||\mathbf{b}||_2\right] = \langle \mathbf{a}, \mathbf{b}\rangle,$$

where $||\mathbf{a}||_2 = \sqrt{\sum_{i=1}^{n} \mathbf{a}_i}$, we propose the following theorem [6, 7].

Theorem 4. *For vectors* $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$*, and target sketch size m, let* $\Phi \sim \mathcal{N}(0,1)^{m\times n}$ *be a Guassian random matrix, and let the mapping be defined as Equation (1), then*

$$\mathbb{P}[|\cos\left(\frac{\pi}{2m} \cdot ||S(\mathbf{a}) - S(\mathbf{b})||\right) \cdot ||\mathbf{a}||_2 \cdot ||\mathbf{b}||_2 - \langle \mathbf{a}, \mathbf{b}\rangle| \leq \epsilon \cdot \pi \cdot ||\mathbf{a}||_2 \cdot ||\mathbf{b}||_2] \geq 1 - \exp\left(-2\epsilon^2 m\right), \tag{9}$$

which follows the fact that

$$\mathbb{P}[|d_{\mathcal{H}}(S(\mathbf{a}), S(\mathbf{b})) - d_{\mathcal{S}}(\mathbf{a}, \mathbf{b})| \leq \epsilon] \geq 1 - \exp\left(-2\epsilon^2 m\right). \tag{10}$$

Proof. By Lemma 2, we have

$$\mathbb{E}[\mathbb{1}[S(\mathbf{a})_i \neq S(\mathbf{b})_i]] = d_{\mathcal{S}}(S(\mathbf{a}), S(\mathbf{b})), \tag{11}$$

and $0 \leq \mathbb{E}[\mathbb{1}[S(\mathbf{a})_i \neq S(\mathbf{b})_i] \leq 1, \ \forall i$. Hence by Hoeffding's inequality, we have

$$\mathbb{P}[|d_{\mathcal{H}}(S(\mathbf{a}), S(\mathbf{b})) - d_{\mathcal{S}}(\mathbf{a}, \mathbf{b})| \leq \epsilon] \geq 1 - \exp(-2\epsilon^2 m), \tag{12}$$

where

$$d_{\mathcal{H}}(S(\mathbf{a}), S(\mathbf{b})) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}[S(\mathbf{a})_i \neq S(\mathbf{b})_i].$$

Now since both $d_{\mathcal{H}}(S(\mathbf{a}), S(\mathbf{b}))$ and $d_{\mathcal{S}}(\mathbf{a}, \mathbf{b})$ are on $[0, 1]$, from

$$|d_{\mathcal{H}}(S(\mathbf{a}), S(\mathbf{b})) - d_{\mathcal{S}}(\mathbf{a}, \mathbf{b})| \leq \epsilon,$$

we can derive

$$\begin{aligned} |\cos\left(\pi \cdot d_{\mathcal{H}}(S(\mathbf{a}), S(\mathbf{b}))\right) - \cos\left(\pi \cdot d_{\mathcal{S}}(\mathbf{a}, \mathbf{b})\right)| &= |-\pi \sin(\pi z)| \cdot |d_{\mathcal{H}}(S(\mathbf{a}), S(\mathbf{b})) - d_{\mathcal{S}}(\mathbf{a}, \mathbf{b})| \\ &\leq \pi\epsilon \end{aligned}$$

by Mean Value Theorem (MVT), for some $z \in [d_{\mathcal{H}}(S(\mathbf{a}), S(\mathbf{b})), d_{\mathcal{S}}(\mathbf{a}, \mathbf{b})]$ (or vice versa). Thus

$$\mathbb{P}[|\cos\left(\pi \cdot d_{\mathcal{H}}(S(\mathbf{a}), S(\mathbf{b}))\right) - \cos\left(\pi \cdot d_{\mathcal{S}}(\mathbf{a}, \mathbf{b})\right)| \leq \pi\epsilon] \geq 1 - \exp(-2\epsilon^2 m), \tag{13}$$

and hence by multiplying $||\mathbf{a}||_2 \cdot ||\mathbf{b}||_2$, we get

$$\mathbb{P}[|\cos\left(\pi \cdot d_{\mathcal{H}}(S(\mathbf{a}), S(\mathbf{b}))\right) \cdot ||\mathbf{a}||_2 \cdot ||\mathbf{b}||_2 - \langle a, b\rangle| \leq \epsilon \cdot \pi \cdot ||\mathbf{a}||_2 \cdot ||\mathbf{b}||_2] \geq 1 - \exp(-2\epsilon^2 m), \tag{14}$$

which is equivalent to Equation (9). □

2.3 Uniform Priority Sampling

Priority Sampling is proposed by [3], but we are going to use the uniform version as we focus on binary data in this project. In brief, the algorithm as described in Algorithm 3 employs a random hash function to select a subset of vector elements based on their hashed values. By comparing the positions where both vectors have non-zero values, it constructs smaller representative sets of elements and corresponding values. The inner product of the vectors is then approximated by summing the product of these representative elements, appropriately scaled according to a threshold value determined by the hash function. In the weighted version, entries are sampled proportional to their corresponding absolute values, and the threshold helps to ensure that the sampling is prioritized, considering the most significant elements first. The outcome is an effective balance between computational efficiency and the accuracy of the inner product estimation, making the algorithm particularly useful for large-scale data applications where full vector computations are impractical.

Theorem 5. *For vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$ with entries bounded in $[-c, c]$, and sketch size m, let W be the inner product estimate returned by Algorithm 3, then we have*

$$\mathbb{E}[W] = \langle \mathbf{a}, \mathbf{b}\rangle, \tag{15}$$

and

$$\mathrm{Var}[W] \leq \frac{2n}{m-1}(||\mathbf{a}||_2^2 ||\mathbf{b}||_2^2) \leq \frac{2n}{m-1} \cdot c^4 \cdot |\mathcal{A} \cap \mathcal{B}|. \tag{16}$$

Algorithm 3 Unweigted Priority Sampling

Input: Length n vector a, Length n vector b, random seed s, target sketch size m.
Output: Estimate of $\langle \mathbf{a}, \mathbf{b} \rangle$

Use random seed s to select a uniformly random hash function $h : \{1, \ldots, n\} \to [0, 1]$. Initialize $K_\mathbf{a}$, $V_\mathbf{a}$ and $K_\mathbf{b}$, $V_\mathbf{b}$ to be empty lists.
Generate $h(i)$ for i from 1 to n.
Set $\tau_\mathbf{a}$ equal to the $(m+1)^{\text{st}}$ smallest value $h(i)$, or set $\tau_\mathbf{a} = \infty$ if $\mathbf{a}$ has less than $m+1$ non-zero values. Similar for $\tau_\mathbf{b}$.
for i such that $\mathbf{a}[i] \neq 0$ **do**
    if $h(i) < \tau_\mathbf{a}$ **then**
        Append i to $K_\mathbf{a}$, append $\mathbf{a}_i$ to $V_\mathbf{a}$.
    if $h(i) < \tau_b$ **then**
        Append i to $K_\mathbf{b}$, append $\mathbf{b}_i$ to $V_\mathbf{b}$.
Compute $\mathcal{T} = K_\mathbf{a} \cap K_\mathbf{b}$. Hence $\forall i \in \mathcal{T}, \mathbf{a}_i \in V_\mathbf{a}, \mathbf{b}_i \in V_\mathbf{b}$.
return

$$W = \sum_{i \in \mathcal{T}} \frac{\mathbf{a}_i \mathbf{b}_i}{\min(1, \tau_\mathbf{a}, \tau_\mathbf{b})}$$

Proof. The proof follows [3, 8]. Let $\mathcal{A} = \{i : \mathbf{a}_i \neq 0\}$ denote the set of indices where $\mathbf{a}$ is non-zero, and $\mathcal{B} = \{i : \mathbf{b}_i \neq 0\}$ denote the set of indices where $\mathbf{a}$ is non-zero. Let $\mathcal{K} = K_\mathbf{a} \cap K_\mathbf{b}$ be defined as in Algorithm 3. Let $\tau_\mathbf{a}^i$ denote the m^{th} smallest of $h(i)$ over all $\mathcal{A} \backslash \{i\}$. Hence it follows that for any $i \in \mathcal{T}$, $\tau_\mathbf{a}^i = \tau_\mathbf{a}$. Similarly, we define $\tau_\mathbf{b}^i$. Then for all i, we define w_i as follows:

$$w_i = \begin{cases} \frac{\mathbf{a}_i \mathbf{b}_i}{\min(1, \tau_\mathbf{a}, \tau_\mathbf{b})}, & \text{if } i \in T \\ 0, & \text{if } i \notin T \end{cases} \tag{17}$$

In this way, we can obtain that

$$W = \sum_{i=1}^{n} w_i.$$

Now we can write the probability of $i \in \mathcal{T}$ for any i as

$$\mathbb{P}[i \in \mathcal{T}] = \mathbb{P}[h(i) < \tau_\mathbf{a}^i \text{ and } \mathbb{P}[h(i) < \tau_\mathbf{b}^i] = \min(1, \tau_\mathbf{a}^i, \tau_\mathbf{b}^i).$$

Thus we have

$$\begin{aligned} \mathbb{E}[w_i] &= \frac{\mathbf{a}_i \mathbf{b}_i}{\min(1, \tau_\mathbf{a}, \tau_\mathbf{b})} \cdot \mathbb{P}[i \in \mathcal{T}] \\ &= \frac{\mathbf{a}_i \mathbf{b}_i}{\min(1, \tau_\mathbf{a}, \tau_\mathbf{b})} \cdot \min(1, \tau_\mathbf{a}^i, \tau_\mathbf{b}^i) \\ &= \mathbf{a}_i \mathbf{b}_i, \end{aligned}$$

which implies that

$$\mathbb{E}[W] = \sum_{i=1}^{n} \mathbb{E}[w_i] = \langle \mathbf{a}, \mathbf{b} \rangle.$$

Now to prove Equation (16), we first want to show the linearity of variance in the equation by showing that for any i, j, w_i and w_j are uncorrelated, i.e.

$$\mathbb{E}[w_i w_j] = \mathbb{E}[w_i]\, \mathbb{E}[w_j] \tag{18}$$

Let $\tau_a^{i,j}$ represent the $(m-1)^{\text{st}}$ smallest of $h(k)$ over $\mathcal{Z}\backslash\{i,j\}$. Then

$$\mathbb{P}[i,j \in \mathcal{T}] = \min\left(1, \tau_{\mathbf{a}}^{i,j}, \tau_{\mathbf{b}}^{i,j}\right)^2,$$

and thus

$$\begin{aligned}\mathbb{E}[w_i w_j] &= \frac{\mathbf{a}_i \mathbf{b}_i}{\min\left(1, \tau_{\mathbf{a}}^{i,j}, \tau_{\mathbf{b}}^{i,j}\right)} \cdot \frac{\mathbf{a}_j \mathbf{b}_j}{\min\left(1, \tau_{\mathbf{a}}^{i,j}, \tau_{\mathbf{b}}^{i,j}\right)} \cdot \mathbb{P}[i,j \in \mathcal{T}] \\ &= \mathbf{a}_i \mathbf{b}_i \cdot \mathbf{a}_j \mathbf{b}_j \\ &= \mathbb{E}[w_i]\,\mathbb{E}[w_j].\end{aligned}$$

Hence we can write

$$\mathrm{Var}[W] = \sum_{i=1}^{n} \mathrm{Var}[w_i].$$

Now we want to bound $\mathrm{Var}[w_i]$ for each $i \in \mathcal{Z}$. Note that for any realization of τ_a^i and τ_b^i,

$$\begin{aligned}\mathbb{E}[w_i^2 | \tau_a^i = t, \tau_b^i = t'] &= \left(\frac{\mathbf{a}_i \mathbf{b}_i}{\min(1,t,t')}\right)^2 \cdot \mathbb{P}[i \in \mathcal{T}] \\ &= \mathbf{a}_i^2 \mathbf{b}_i^2 \max\left(1, \frac{1}{t}, \frac{1}{t'}\right),\end{aligned}$$

then we have

$$\begin{aligned}\mathrm{Var}[w_i] &= \mathbb{E}[w_i^2] - \mathbb{E}[w_i]^2 \\ &= \mathbf{a}_i^2 \mathbf{b}_i^2 \int_0^\infty \int_0^\infty \max\left(1, \frac{1}{t}, \frac{1}{t'}\right) \mathbb{P}[\tau_{\mathbf{a}}^i = t, \tau_{\mathbf{b}}^i = t']\, dt\, dt' - \mathbf{a}_i^2 \mathbf{b}_i^2 \\ &\le \mathbf{a}_i^2 \mathbf{b}_i^2\, \mathbb{E}[\frac{1}{\tau_b^i}] + \mathbf{a}_i^2 \mathbf{b}_i^2\, \mathbb{E}[\frac{1}{\tau_a^i}] && \text{by Proof of Theorem 3 in [3]} \\ &\le 2\mathbf{a}_i^2 \mathbf{b}_i^2 \frac{n}{m-1} && \text{by Claim 5 in [8]} \\ &\le 2c^4 \frac{n}{m-1}\end{aligned}$$

given that the sum of weights in this case is $n \cdot 1 = n$ for both $\mathbf{a}$ and $\mathbf{b}$. Therefore,

$$\mathrm{Var}[W] = \sum_{i \in \mathcal{A} \cap \mathcal{B}}^{n} \mathrm{Var}[w_i] \le \frac{2n}{m-1} \cdot c^4 \cdot |\mathcal{A} \cap \mathcal{B}|$$

□

We can analyze weighted and uniform priority sampling by comparing n and $\sum_{i \in \mathcal{A} \cap \mathcal{B}} ||\mathbf{a}||_2^2 / \mathbf{a}_i^2$ (similar for $\mathbf{b}$). However, in this project, we are analyzing binary data, so $c = 1$ and therefore $\mathrm{Var}[W] \le \frac{2n}{m-1} \cdot |\mathcal{A} \cap \mathcal{B}|$. This implies that as the overlap ratio gets larger, the performance of uniform priority sampling gets worse.

3 1-bit MinHash

In this paper, we propose 1-bit MinHash [9] for inner product estimation. The basic 1-bit MinHash sketching method is shown in Algorithm 4, which returns $H_{\mathbf{a}}^{hash}$ and $H_{\mathbf{a}}^{val}$ as binary vectors of minimum hashes in one bit. Note that

Algorithm 4 1-bit MinHash Sketch

Input: Length n vector $\mathbf{a}$, sample number m, random seed s.
Output: Sketch $H_{\mathbf{a}} = \{H_{\mathbf{a}}^{hash}, \|\mathbf{a}\|\}$, where $H_{\mathbf{a}}^{hash}$ is a length m vector of values in $\{\pm 1\}$ and $\|\mathbf{a}\|$ is a scalar, the 1 norm of $\mathbf{a}$.

Initialize random number generator with seed s.
for i = 1, …, m **do**
    Select uniformly random hash func. $h^i : \{1, ..., n\} \to [0, 1]$.
    Compute $j^* = \arg\min_{j \in \{1,...,n\}, \mathrm{a}[j] \neq 0} h^i(j)$.
    Set $H_{\mathbf{a}}^{hash}[i] = 1$ if $h^i(j^*)$ is odd otherwise 0
return $\{H_{\mathbf{a}}^{hash}, \|\mathbf{a}\|\}$

Algorithm 5 1-bit MinHash Estimate

Input: Sketches $H_{\mathbf{a}} = \{H_{\mathbf{a}}^{hash}, \|\mathbf{a}\|\}, H_{\mathbf{b}} = \{H_{\mathbf{b}}^{hash}, \|\mathbf{b}\|\}$ constructed using Algorithm 4 with the same inputs m, s.
Output: Estimate of $\langle \mathbf{a}, \mathbf{b} \rangle$.

Set $\tilde{U} = \frac{m \cdot (\|\mathbf{a}\| + \|\mathbf{b}\|)}{2 \sum_{i=1}^{m} \mathbb{1}\left[H_{\mathbf{a}}^{hash}[i] = H_{\mathbf{b}}^{hash}[i]\right]}$
$I = \tilde{U} \cdot \left(\frac{2}{m} \cdot \sum_{i=1}^{m} \mathbb{1}\left[H_{\mathbf{a}}^{hash}[i] = H_{\mathbf{b}}^{hash}[i]\right] - 1 \right)$
return I

their corresponding vector values should all be 1. Specifically, consider a vector $\mathbf{a}$, to create an entry in the standard MinHash sketch (refer to [10]), we hash each index of non-zero elements in $\mathbf{a}$ into the interval $[0, 1]$. The smallest hash value associated with a non-zero element is then examined. If this hash value is odd, we record a 1; if it is even, we record a 0. This procedure is repeated m times using different random hash functions. For binary vectors $\mathbf{a}$ and $\mathbf{b}$, with non-zero index sets $\mathcal{A} = \{i : \mathbf{a}_i = 1\}$ and $\mathcal{B} = \{i : \mathbf{b}_i = 1\}$ respectively, the minimum hash value can estimate the Jaccard similarity $|\mathcal{A} \cap \mathcal{B}|/|\mathcal{A} \cup \mathcal{B}|$ or calculate the union size $|\mathcal{A} \cup \mathcal{B}|$. Note that in this way, $\langle \mathbf{a}, \mathbf{b} \rangle = |\mathcal{A} \cap \mathcal{B}|$

Claim 6. *Consider vectors* $\mathbf{a}$ *and* $\mathbf{b}$ *sketched using Algorithm 4 to produce sketches* $H_{\mathbf{a}}$ *and* $H_{\mathbf{b}}$. *Define the sets* $\mathcal{A} = \{i : \mathbf{a}_i \neq 0\}$ *and* $\mathcal{B} = \{i : \mathbf{b}_i \neq 0\}$. *Then for all* $i \in \{1, \ldots, m\}$ *we have:*

$$\mathbb{E}\left[\mathbb{1}\left[H_{\mathbf{a}}^{hash}[i] = H_{\mathbf{b}}^{hash}[i]\right]\right] = \frac{1}{2}\left(\frac{\|\mathbf{a}\| + \|\mathbf{b}\|}{\sum_{i=1}^{n} \max(\mathbf{a}_i, \mathbf{b}_i)}\right) \tag{19}$$

Proof. Consider the 1-bit MinHash sketches of vectors $\mathbf{a}$ and $\mathbf{b}$. The indicator variable $\mathbb{1}\left[H_{\mathbf{a}}^{\text{hash}}[i] = H_{\mathbf{b}}^{\text{hash}}[i]\right]$ is one with a probability that depends on the similarity between $\mathbf{a}$ and $\mathbf{b}$. Precisely, it equals one with a probability of $\frac{|\mathcal{A} \cap \mathcal{B}|}{|\mathcal{A} \cup \mathcal{B}|}$, indicating the occurrence where MinHash elements for both vectors coincide, resulting in identical hashes.

Alternatively, if the probability is $1 - \frac{|\mathcal{A} \cap \mathcal{B}|}{|\mathcal{A} \cup \mathcal{B}|}$, the indices differ. Nevertheless, there remains a 50% chance that the hashes will match, specifically in cases where both indices are either odd or even. This leads to the following

conclusion:

$$\begin{aligned}
\mathbb{E}\left[\mathbb{1}\left[H_{\mathbf{a}}^{hash}[i] = H_{\mathbf{b}}^{hash}[i]\right]\right] &= \mathbb{P}\left[H_{\mathbf{a}}^{hash}[i] = H_{\mathbf{b}}^{hash}[i]\right] \\
&= \frac{|\mathcal{A} \cap \mathcal{B}|}{|\mathcal{A} \cup \mathcal{B}|} + \frac{1}{2} \cdot (1 - |\mathcal{A} \cup \mathcal{B}|) \\
&= \frac{1}{2} \cdot (1 + |\mathcal{A} \cup \mathcal{B}|) \\
&= \frac{1}{2}\left(\frac{\sum_{i=0}^{n} \min(|\mathbf{a}_i|, |\mathbf{b}_i|) + \sum_{i=0}^{n} \max(|\mathbf{a}_i|, |\mathbf{b}_i|)}{\sum_{i=0}^{n} \max(|\mathbf{a}_i|, |\mathbf{b}_i|)}\right) \\
&= \frac{1}{2}\left(\frac{\sum_{i=0}^{n} \mathbf{a}_i + \mathbf{b}_i}{\sum_{i=0}^{n} \max(|\mathbf{a}_i|, |\mathbf{b}_i|)}\right) \\
&= \frac{1}{2}\left(\frac{\|\mathbf{a}\| + \|\mathbf{b}\|}{\sum_{i=0}^{n} \max(|\mathbf{a}_i|, |\mathbf{b}_i|)}\right)
\end{aligned}$$

□

Now let us consider $\tilde{U} = \frac{m \cdot (\|\mathbf{a}\| + \|\mathbf{b}\|)}{2 \sum_{i=1}^{m} \mathbb{1}\left[H_{\mathbf{a}}^{hash}[i] = H_{\mathbf{b}}^{hash}[i]\right]}$ from line 1 of Algorithm 5. In the following, we will prove that $\tilde{U}$ is a good estimator for $|\mathcal{A} \cup \mathcal{B}|$. We first note that since $\mathbf{a}$ and $\mathbf{b}$ are both binary vectors, we can write

$$\begin{aligned}
|\mathcal{A} \cup \mathcal{B}| &= \sum_{i=0}^{n} \begin{cases} 1 & \text{if } \mathbf{a}_i = 1 \text{ or } \mathbf{b}_i = 1 \\ 0 & \text{if } \mathbf{a}_i = 0 \text{ and } \mathbf{b}_i = 0 \end{cases} \\
&= \sum_{i=0}^{n} \max(|\mathbf{a}_i|, |\mathbf{b}_i|)
\end{aligned}$$

Claim 7. *Let $\tilde{U} = \frac{m \cdot (\|\mathbf{a}\| + \|\mathbf{b}\|)}{2 \sum_{i=1}^{m} \mathbb{1}\left[H_{\mathbf{a}}^{hash}[i] = H_{\mathbf{b}}^{hash}[i]\right]}$ as defined in line 1 of Algorithm 5, then the estimator $\tilde{U}$ satisfies:*

$$\mathbb{E}\left[\frac{1}{\tilde{U}}\right] = \frac{1}{\sum_{i=0}^{n} \max(|\mathbf{a}_i|, |\mathbf{b}_i|)} \tag{20}$$

Proof. Since $\sum_{i=1}^{m} \mathbb{1}\left[H_{\mathbf{a}}^{hash}[i] = H_{\mathbf{b}}^{hash}[i]\right] / m$ is an estimator for $\mathbb{1}\left[H_{\mathbf{a}}^{hash}[i] = H_{\mathbf{b}}^{hash}[i]\right]$, by Claim 6, we have

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{\tilde{U}}\right] &= \mathbb{E}\left[\frac{2 \sum_{i=1}^{m} \mathbb{1}\left[H_{\mathbf{a}}^{hash}[i] = H_{\mathbf{b}}^{hash}[i]\right]}{m \cdot (\|\mathbf{a}\| + \|\mathbf{b}\|)}\right] \\
&= \frac{2}{\|\mathbf{a}\| + \|\mathbf{b}\|} \cdot \mathbb{E}\left[\mathbb{1}\left[H_{\mathbf{a}}^{hash}[i] = H_{\mathbf{b}}^{hash}[i]\right]\right] \\
&= \frac{2}{\|\mathbf{a}\| + \|\mathbf{b}\|} \cdot \frac{1}{2}\left(\frac{\|\mathbf{a}\| + \|\mathbf{b}\|}{\sum_{i=0}^{n} \max(|\mathbf{a}_i|, |\mathbf{b}_i|)}\right) \\
&= \frac{1}{\sum_{i=0}^{n} \max(|\mathbf{a}_i|, |\mathbf{b}_i|)}
\end{aligned}$$

□

Claim 8. *The variance of estimator $\tilde{U}$ is bounded as follows:*

$$\mathrm{Var}\left[\frac{1}{\tilde{U}}\right] \leq \left(\frac{2}{m \cdot (\|\mathbf{a}\| + \|\mathbf{b}\|)}\right) \cdot \frac{1}{\sum_{i=0}^{n} \max(|\mathbf{a}_i|, |\mathbf{b}_i|)} \tag{21}$$

Proof.

$$\begin{aligned}\mathrm{Var}\left[\frac{1}{\tilde{U}}\right] &= \mathrm{Var}\left[\frac{2\sum_{i=1}^{m}\mathbb{1}\left[H_{\mathbf{a}}^{hash}[i]=H_{\mathbf{b}}^{hash}[i]\right]}{m\cdot(\|\mathbf{a}\|+\|\mathbf{b}\|)}\right]\\ &= \left(\frac{2}{m\cdot(\|\mathbf{a}\|+\|\mathbf{b}\|)}\right)^2\cdot\sum_{j=1}^{m}\mathrm{Var}\left[\mathbb{1}\left[H_{\mathbf{a}}^{hash}[i]=H_{\mathbf{b}}^{hash}[i]\right]\right]\end{aligned}$$

Because $\mathbb{1}\left[H_{\mathbf{a}}^{hash}[i]=H_{\mathbf{b}}^{hash}[i]\right]$ is a binary variable, the variance of that can be bounded by its probability of being 1. Therefore, with the same reasoning as presented in Claim 7, we have the following:

$$\begin{aligned}\mathrm{Var}\left[\frac{1}{\tilde{U}}\right] &\le \left(\frac{2}{m\cdot(\|\mathbf{a}\|+\|\mathbf{b}\|)}\right)^2\cdot\sum_{j=1}^{m}\frac{1}{2}\left(\frac{\|\mathbf{a}\|+\|\mathbf{b}\|}{\sum_{i=0}^{n}\max(|\mathbf{a}_i|,|\mathbf{b}_i|)}\right)\\ &= \left(\frac{2}{m\cdot(\|\mathbf{a}\|+\|\mathbf{b}\|)}\right)\cdot\frac{1}{\sum_{i=0}^{n}\max(|\mathbf{a}_i|,|\mathbf{b}_i|)}\end{aligned}$$

□

Then, using Claim 7, Claim 8, we can apply Chebyshev's inequality, which gives us, with high probability if m is large enough,

$$(1-\epsilon)\frac{1}{U}\le\frac{1}{\tilde{U}}\le(1+\epsilon)\frac{1}{U}$$

So, with the assumption of a small ϵ, with high probability, we have the following:

$$(1-2\epsilon)U\le\tilde{U}\le(1+2\epsilon)U$$

Therefore, we can assume $\tilde{U}$ is approximately equal to $U=\sum_{i=0}^{n}\max(|\mathbf{a}_i|,|\mathbf{b}_i|)=|\mathcal{A}\cup\mathcal{B}|$. Now we prove that I is a good estimator for the inner product of vector $\mathbf{a}$ and $\mathbf{b}$.

Claim 9. *Let $\tilde{I}=U\cdot\left(\frac{2}{m}\cdot\sum_{i=1}^{m}\mathbb{1}\left[H_{\mathbf{a}}^{hash}[i]=H_{\mathbf{b}}^{hash}[i]\right]-1\right)$ as defined in line 3 of Algorithm 5, then the estimator I satisfies: $\mathbb{E}[I]=\langle\mathbf{a},\mathbf{b}\rangle$*

Proof. By Claim 7, we have

$$\begin{aligned}\mathbb{E}\left[\tilde{I}\right] &= \mathbb{E}\left[U\cdot\left(\frac{2}{m}\cdot\sum_{i=1}^{m}\mathbb{1}\left[H_{\mathbf{a}}^{hash}[i]=H_{\mathbf{b}}^{hash}[i]\right]-1\right)\right]\\ &= \sum_{i=0}^{n}\max(|\mathbf{a}_i|,|\mathbf{b}_i|)\cdot\left(\frac{\|\mathbf{a}\|+\|\mathbf{b}\|}{\sum_{i=0}^{n}\max(|\mathbf{a}_i|,|\mathbf{b}_i|)}-1\right)\\ &= \|\mathbf{a}\|+\|\mathbf{b}\|-\sum_{i=0}^{n}\max(|\mathbf{a}_i|,|\mathbf{b}_i|)\\ &= \sum_{i=0}^{n}|\mathbf{a}_i|+|\mathbf{b}_i|+\max(|\mathbf{a}_i|,|\mathbf{b}_i|)\\ &= \sum_{i=0}^{n}\min(|\mathbf{a}_i|,|\mathbf{b}_i|)=|\mathcal{A}\cap\mathcal{B}|=\langle\mathbf{a},\mathbf{b}\rangle\end{aligned}$$

□

Now we want to bound the Var $\left[\tilde{I}\right]$.

Claim 10. *The variance of estimator $\tilde{I}$ is bounded as follows:*

$$\text{Var}\left[\tilde{I}\right] = \frac{2}{m}|\mathcal{A} \cup \mathcal{B}| \cdot (\|\mathbf{a}\| + \|\mathbf{b}\|) - \frac{1}{m}(\|\mathbf{a}\| + \|\mathbf{b}\|)^2 \tag{22}$$

Proof.

$$\begin{aligned}\text{Var}\left[\tilde{I}\right] &= \text{Var}\left[U \cdot \left(\frac{2}{m} \cdot \sum_{i=1}^{m} \mathbb{1}\left[H_{\mathbf{a}}^{hash}[i] = H_{\mathbf{b}}^{hash}[i]\right] - 1\right)\right] \\ &= \left(\frac{2U}{m}\right)^2 \cdot \sum_{j=1}^{m} \text{Var}\left[\mathbb{1}\left[H_{\mathbf{a}}^{hash}[i] = H_{\mathbf{b}}^{hash}[i]\right]\right]\end{aligned}$$

Then similar to the proof of Claim 8, we can write

$$\begin{aligned}\text{Var}\left[\tilde{I}\right] &= \left(\frac{2U}{m}\right)^2 \cdot \sum_{j=1}^{m} \left(\left(\frac{1}{2} \cdot \frac{\|\mathbf{a}\| + \|\mathbf{b}\|}{\sum_{i=0}^{n} \max(|\mathbf{a}_i|, |\mathbf{b}_i|)}\right) - \left(\frac{1}{2} \cdot \frac{\|\mathbf{a}\| + \|\mathbf{b}\|}{\sum_{i=0}^{n} \max(|\mathbf{a}_i|, |\mathbf{b}_i|)}\right)^2\right) \\ &= \frac{2}{m}|\mathcal{A} \cup \mathcal{B}| \cdot (\|\mathbf{a}\| + \|\mathbf{b}\|) - \frac{1}{m}(\|\mathbf{a}\| + \|\mathbf{b}\|)^2\end{aligned}$$

□

Now given this variance, suppose vector $\|\mathbf{a}\|$ and $\|\mathbf{b}\|$ are identical, then in this case, $2 \cdot |\mathcal{A} \cup \mathcal{B}| = \|\mathbf{a}\| + \|\mathbf{b}\|$, and hence

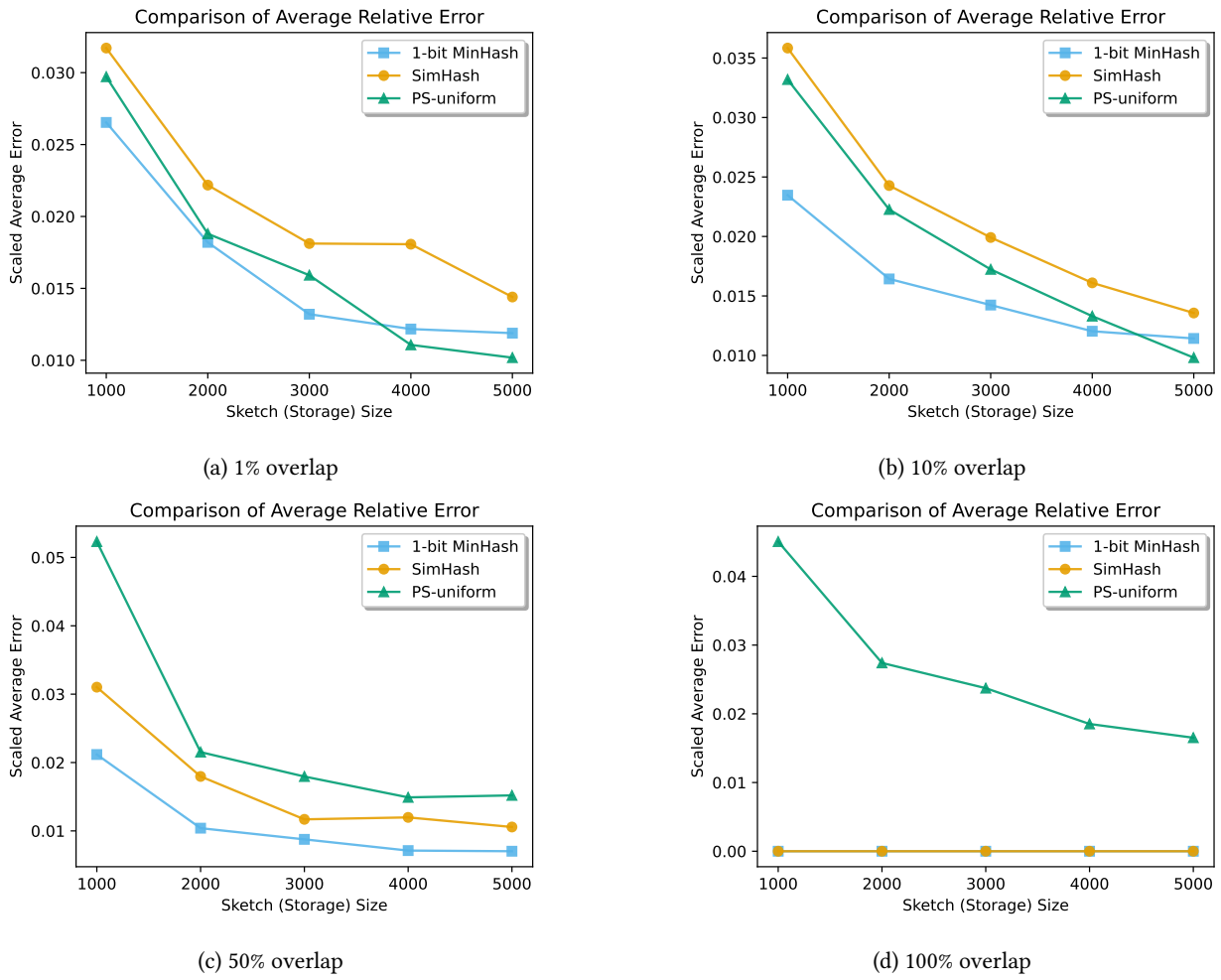
$$\text{Var}\left[\tilde{I}\right] = \frac{1}{m}(\|\mathbf{a}\| + \|\mathbf{b}\|)^2 - \frac{1}{m}(\|\mathbf{a}\| + \|\mathbf{b}\|)^2 = 0.$$

Therefore, we have guarantee that if $\|\mathbf{a}\|$ and $\|\mathbf{b}\|$ are identical, Var $\left[\tilde{I}\right] = 0$. This implies that the estimation error in such a case is zero, effectively achieving perfect estimation.

4 Experiments

Our methodology is initially assessed using synthetic datasets, which allows for precise control over various parameters. Binary vectors $\mathbf{a}$ and $\mathbf{b}$, each of length 10000, are generated for this purpose. To simulate diverse scenarios with varying degrees of overlap, the proportion of non-zero entries that are common to both $\mathbf{a}$ and $\mathbf{b}$—termed as *overlap*—is systematically varied. In the primary experimental series, the entries of $\mathbf{a}$ and $\mathbf{b}$ are randomly selected from the set $\{0, 1\}$. Subsequently, in a second series of experiments designed to emulate sparse vector conditions, 50% of the entries in each vector are set to zero, while the remaining entries are randomly assigned values from $\{0, 1\}$. This approach is employed to rigorously evaluate the efficacy of our proposed method under different vector characteristics.

To thoroughly assess our proposed approach under various vector characteristics, the next series of experiments focuses on the scaled average difference between the actual and estimated inner products. These estimations are obtained via SimHash, 1-bit MinHash, and Uniform Priority Sampling (PS-uniform). Given the findings in [3]—where PS-uniform is shown to outperform the Johnson-Lindenstrauss (JL) —our baseline only includes PS-uniform and SimHash, with the latter not being evaluated in the cited study. The x-axis in our plots represents the *Sketch size*



(a) 1% overlap

(b) 10% overlap

(c) 50% overlap

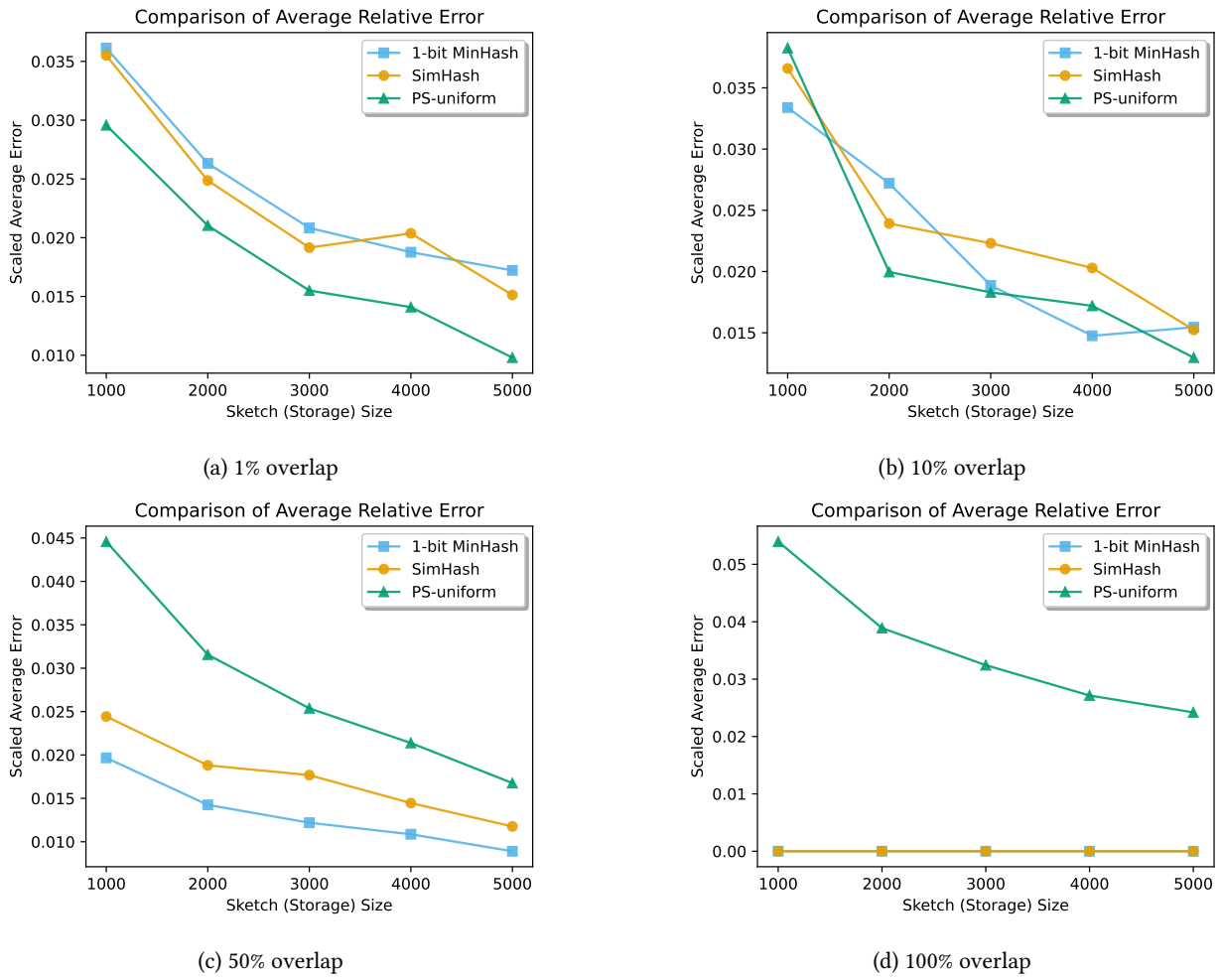
(d) 100% overlap

Figure 1: Inner product estimation for real-valued synthetic uniform binary vectors $\mathbf{a}$ and $\mathbf{b}$ of length 10000, where the entries are randomly selected from the set $\{0, 1\}$.

for SimHash and 1-bit MinHash, which utilizes a single bit for each entry in an m-sized sketch. For fairness in comparison, the sketch size for PS-uniform is set to $m/1.5$, approximating its storage size. We conduct experiments for various sketch sizes: $m = 1000, 2000, 3000, 4000, 5000$, over 100 iterations to calculate the average error, under overlap ratios of 1%, 10%, 50%, and 100%.

4.1 Uniform Binary Vectors

In this experimental series, the elements of vectors $\mathbf{a}$ and $\mathbf{b}$ are independently sampled from the binary set $\{0, 1\}$. The findings, depicted in Figure 1, reveal that for overlap ratios of 1%, 10%, and 50%, 1-bit MinHash consistently outperforms the baselines, particularly SimHash. This is notable as both SimHash and 1-bit MinHash employ quantization methods utilizing a single bit per entry, making them directly comparable. At an overlap ratio of 100%, both SimHash and 1-bit MinHash exhibit zero error. *This occurs because, for SimHash, a 100% overlap equates to* $\theta = 0$, *and it follows that* $\mathrm{Var}\left[d_{\mathcal{H}}\right] = 0$ *as explained after Claim 3. In this case, the inner product is simply the product of the*



(a) 1% overlap

(b) 10% overlap

(c) 50% overlap

(d) 100% overlap

Figure 2: Inner product estimation for real-valued synthetic sparse binary vectors $\mathbf{a}$ and $\mathbf{b}$ of length 10000, where 50% of the entries in each vector are set to zero, while the remaining entries are randomly assigned values from $\{0, 1\}$.

norms $\|\mathbf{a}\|\|\mathbf{b}\|$. *For 1-bit MinHash, this scenario leads to* $2 \cdot |\mathcal{A} \cup \mathcal{B}| = \|\mathbf{a}\| + \|\mathbf{b}\|$, *resulting in* $\mathrm{Var}\left[\tilde{I}\right] = 0$, *as discussed following Claim 10.*

4.2 Sparse Binary Vectors

This experimental set aims to replicate sparse binary vector conditions by setting 50% of the elements in each vector to zero and assigning the remaining elements randomly from $\{0, 1\}$. The results, presented in Figure 2, indicate that while PS-uniform does not perform as well as in the uniform case, SimHash and 1-bit MinHash yield comparable results. Notably, at an overlap ratio of 50%, 1-bit MinHash significantly outperforms its counterparts. For an overlap ratio of 100%, the observations align with those for the uniform binary vector case.

Future work will include evaluating quantization methods alongside weighted Priority Sampling, which has shown superior performance in non-uniform cases compared to uniform Priority Sampling [3].

5 Cross Correlation Analysis

5.1 Problem

The cross-correlation analysis enables the investigation of neuronal connectivity and interaction, especially for Neuralpixel data. To formalize the problem, Given a set of spike trains, the goal is to compute the cross-correlation between pairs of these spike trains. For each spike train $s_j(t)$, we define:

$$s_j(t) = \begin{cases} 1 & \text{if there is a spike in train } j \text{ at time } t, \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

for $j = 1, 2, ..., n$, and where t represents discrete time points with a total count of T. The output of our analysis is a set of $\frac{n(n-1)}{2}$ cross-correlation functions, y_{ij}, each with a length of $2T - 1$, measuring the correlation between spike train s_i and spike train s_j. The cross-correlation function between spike train s_i and s_j is defined as:

$$y_{ij}(\tau) = \frac{1}{T - |\tau|} \sum_{t=1}^{T-|\tau|} s_i(t) s_j(t + |\tau|) \tag{24}$$

where τ is the time lag, varying from $-(T-1)$ to $T-1$, and $y_{ij}(\tau)$ quantifies the degree to which the spikes in train s_i predict the spikes in train s_j at time lag τ. The naive solution is to iterate over each possible time lag and compute the sum of products for overlapping elements. The Fourier Transform method computes cross-correlation by taking the inverse Fourier Transform of the product of the Fourier Transforms of two signals, exploiting the convolution theorem for computational efficiency. The downstream task of the cross-correlation in neural data processing to find the peak time lag with some standard. Here we simplify the condition to find the time lag τ that maximizes cross-correlation value.

The computational complexity for cross-correlation using the naive approach is $O(n^2T^2)$, while employing the Fourier Transform reduces it to $O(n^2T \log T)$. In both cases, the key factor contributing to this complexity is the quadratic number of pairs (n^2) requiring computation. To address this, we propose the application of Locality Sensitivity Hashing (LSH) to selectively reduce the number of pairs evaluated. This method is particularly effective as it employs inner product sketching to expedite computation. Additionally, the integration of sketching with quantization techniques presents a promising avenue for adapting these methods to binary arrays.

5.2 Methods

Our approach leverages MinHash LSH and SimHash LSH to reduce the computational burden associated with extensive cross-correlation analyses. The underlying rationale for employing these methods is rooted in the hypothesis that spike trains exhibiting similar profiles are likely to demonstrate higher Jaccard similarity and Cosine similarity.

Minhash LSH

The MinHash LSH method is shown in Algorithm 6 and the following is the notation.

- $\mathcal{A}$: Set of binary arrays. Each array represents a set or item in binary form.
- k: Number of hash functions used in the MinHash algorithm.

Algorithm 6 MinHash LSH

```
procedure CreateMinHash(𝒜, k)
    n ← length of the binary arrays in 𝒜
    Initialize M[1...n] to ∞ for each array in 𝒜
    for i = 1 to k do
        h ← a random permutation of 1...n
        for each array A in 𝒜 do
            for each index j where A[j] = 1 do
                M[j] ← min(M[j], h(j))
    return M for each array in 𝒜
procedure FindNearestNeighbor(Q, ℳ, τ)
    BestMatch ← null
    for each signature S in ℳ do
        if S ≠ Q then
            similarity ← ComputeJaccardSimilarity(Q, S)
            if similarity ≥ τ then
                BestMatch ← S
    return BestMatch
```

- n: Length of each binary array in $\mathcal{A}$.
- M: Array of MinHash signatures. Each signature is a compressed representation of the corresponding binary array.
- h: A random permutation function used to generate hash values in MinHash.
- Q: The query signature for which the nearest neighbor is sought.
- $\mathcal{M}$: Set of MinHash signatures of all items in the dataset.
- τ: Similarity threshold for selecting the nearest neighbor. It determines how close two items must be to be considered similar.
- $BestMatch$: The signature of the item that is the nearest neighbor to the query signature Q.

Simhash LSH

The SimHash LSH method is shown in Algorithm 7 and the following is notation.

- $\mathcal{I}$: The set of items to be hashed. Each item could be a document, a set of features, etc.
- $f(\cdot)$: The feature extraction function that converts each item into a feature vector.
- k: The number of bits in the SimHash fingerprint.
- $SimHash(\cdot)$: The SimHash function that computes a k-bit fingerprint for each item.
- $T[1 \ldots n]$: An array of n hash tables used in the LSH process.
- B: The number of bits used from the SimHash value for each hash table.

Algorithm 7 SimHash LSH

```
procedure LSH(d, h)
    d ← dimension of the vectors
    h ← number of hash tables
    Initialize T[1 . . . h] as empty dictionaries
    Initialize R[1 . . . h] with random vectors of size d
function HASH(v, i)
    return [⟨v, R[i]⟩ > 0]
procedure ADD(v, ℓ, i)
    hv ← HASH(v, i)
    if hv not in T[i] then
        T[i][hv] ← empty list
    Append ℓ to T[i][hv]
procedure FIT(D, L)
    for i ← 1 to h do
        for all (ℓ, v) in zip(L, D) do
            ADD(v, ℓ, i)
procedure QUERY(q)
    C ← empty set
    for i ← 1 to h do
        hv ← HASH(q, i)
        if hv in T[i] then
            C ← C ∪ T[i][hv]
    return C
```

- $\mathcal{H}[1 \dots B]$: An array of hash functions, one for each bit range in the SimHash fingerprint.
- $\mathcal{Q}$: A query item for which similar items are sought.
- $\mathcal{S}$: The set of items deemed similar to the query item.

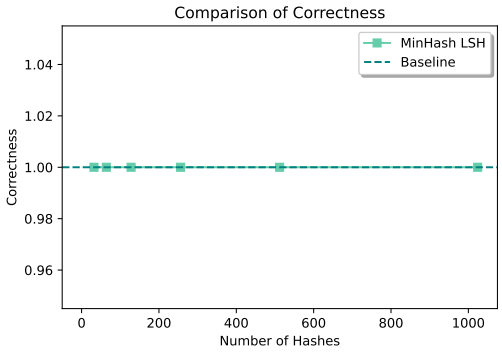
Dataset

In the Visual Coding – Neuropixels project, the Allen Institute employs high-density extracellular electrophysiology (Ecephys) probes for extensive spike recording across various regions of the mouse brain. Our study utilizes a dataset comprising 884 spike trains, each spanning 6000 timesteps, derived from 15 distinct brain regions in a cohort of 25 mice. This rich dataset provides a comprehensive basis for analyzing neural activity and connectivity patterns.

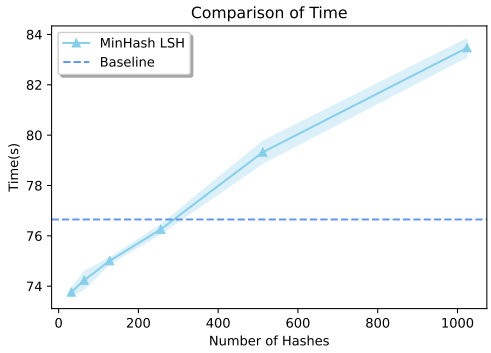
5.3 Metrics

To assess the accuracy of the implemented LSH method, we conduct an analysis of the peak time lag for each pair of spike trains. This involves computing the proportion of instances where the method correctly identifies the peak time lag, thereby providing a quantitative measure of its performance and reliability.

$$correctness = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} \mathbb{1}(argmax_{\tau} CCG_{raw}[ij] == argmax_{\tau} CCG_{LSH}[ij])}{n^2}$$

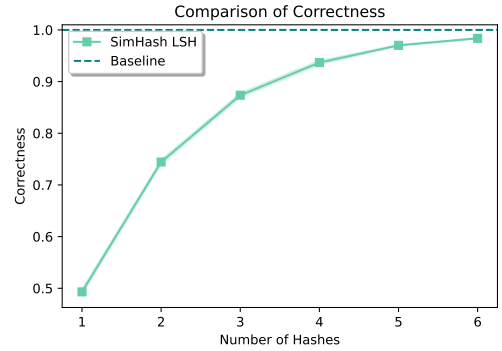


(a) Comparison of Correctness

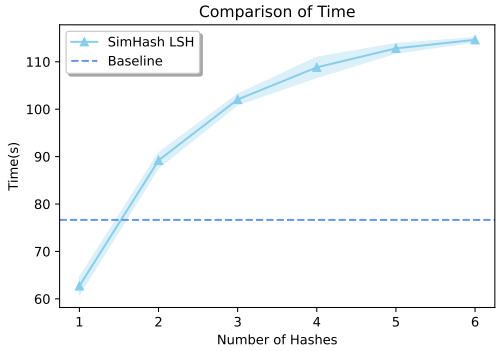


(b) Comparison of Time

Figure 3: MinHash LSH for cross-correlation computation



(a) Comparison of Correctness



(b) Comparison of Time

Figure 4: SimHash LSH for cross-correlation computation

5.4 Analysis

For our initial investigation, we simplify our analysis to a subset of the available data, focusing on spike train recordings from two brain regions during a single trial from one mouse. This preliminary approach allows for a manageable yet insightful exploration of the dataset. The baseline computation time for this dataset is recorded at 76.65 seconds.

In our experiments with MinHash LSH, we vary the number of hash tables across several values: 32, 64, 128, 256, 512, and 1024. As illustrated in Figure 3, the MinHash LSH methodology maintains a high level of accuracy with a relatively efficient runtime with the number of hash tables smaller than 256. However, it is noteworthy that the runtime exceeds the original computation time when the number of hash tables is larger than 256. Conversely, for SimHash LSH, our experiments involve adjusting the number of hash tables, h, testing values from 1 to 5. The results, depicted in Figure 4, reveal that as the number of hash tables increases, the SimHash LSH method consistently achieves an accuracy above 0.5, while the runtime remains below that of the original computation only with 1 hash table.

6 Future Directions

Inner Product Estimation

Firstly, an extension of the 1-bit MinHash technique to a k-bit MinHash variant can be proposed to investigate the correlation between the number of bits used and the resulting accuracy enhancement. Furthermore, we aim to introduce a k-bit weighted MinHash or Priority Sampling algorithm applicable to general vectors. Preliminary progress has been made in both areas, and further research and development are planned for the upcoming winter period.

Cross-Correlation

The primary challenge in this application lies in the inherent limitations of SimHash LSH and MinHash LSH, which are predicated on similarity measures that do not effectively incorporate cross-correlation. To enhance both the time efficiency and accuracy of LSH in computing cross-correlation, our research will explore LSH methods grounded in similarity metrics that account for time lag. In pursuit of this goal, we will investigate more rapid LSH techniques, as outlined by Lv et al. [11]. Additionally, we plan to integrate other pre-processing approaches, such as Dynamic Time Warping (DTW), to analyze the similarity between two spike trains, particularly when considering a time lag offset.

7 Acknowledgement

We extend our sincere thanks to Professor Christopher Musco, Majid Daliri, and Haoxiang Zhang for their invaluable advice and insights on our project. Their guidance has been instrumental in the development of this work.

References

[1] Rosa I Arriaga and Santosh Vempala. An algorithmic theory of learning: Robust concepts and random projection. *Machine learning*, 63:161–182, 2006.

[2] Aline Bessa, Majid Daliri, Juliana Freire, Cameron Musco, Christopher Musco, Aécio Santos, and Haoxiang Zhang. Weighted minwise hashing beats linear sketching for inner product estimation. In *Proceedings of the 42nd ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 169–181, 2023.

[3] Majid Daliri, Juliana Freire, Christopher Musco, Aécio Santos, and Haoxiang Zhang. Sampling methods for inner product sketching.

[4] 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.

[5] Laurent Jacques. A quantized johnson–lindenstrauss lemma: The finding of buffon's needle. *IEEE Transactions on Information Theory*, 61(9):5012–5027, 2015.

[6] Laurent Jacques, Jason N Laska, Petros T Boufounos, and Richard G Baraniuk. Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors. *IEEE transactions on information theory*, 59(4):2082–2102, 2013.

[7] Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995.

[8] Majid Daliri, Juliana Freire, Christopher Musco, Aécio Santos, and Haoxiang Zhang. Simple analysis of priority sampling. *arXiv preprint arXiv:2308.05907*, 2023.

[9] Anshumali Shrivastava and Ping Li. In defense of minhash over simhash. In *Artificial Intelligence and Statistics*, pages 886–894. PMLR, 2014.

[10] Andrei Z Broder. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*, pages 21–29. IEEE, 1997.

[11] Qin Lv, William Josephson, Zhe Wang, Moses Charikar, and Kai Li. Multi-probe lsh: efficient indexing for high-dimensional similarity search. In *Proceedings of the 33rd international conference on Very large data bases*, pages 950–961, 2007.