Machine Learning (2022) 111:2631–2662
https://doi.org/10.1007/s10994-022-06166-z

Variance reduction in feature hashing using MLE and control variate method

Bhisham Dev Verma[1] · **Rameshwar Pratap**[1] · **Manoj Thakur**[1]

Received: 24 January 2021 / Revised: 4 March 2022 / Accepted: 7 March 2022 /
Published online: 2 May 2022

Abstract
The feature hashing algorithm introduced by Weinberger et al. (2009) is a popular dimensionality reduction algorithm that compresses high dimensional data points into low dimensional data points that closely approximate the pairwise inner product. This algorithm has been used in many fundamental machine learning applications such as model compression (Chen et al. 2015), spam classification (Weinberger et al. 2009), compressing text classifiers (Joulin et al. 2016), large scale image classification (Mensink et al. 2012). However, a limitation of this approach is that the variance of its estimator for the inner product tends to be large for small values of the reduced dimensions, making the estimate less reliable. We address this challenge and suggest two simple and practical solutions in this work. Our approach relies on control variate (CV) and maximum likelihood estimator (MLE), which are popular variance reduction techniques used in statistics. We show that these methods lead to significant variance reduction in the inner product similarity estimation. We give theoretical bounds on the same and complement it via extensive experiments on synthetic and real-world datasets. Given the simplicity and effectiveness of our approach, we hope that it can be adapted in practice.

Keywords Dimensionality reduction · Variance reduction · Control variate · Maximum likelihood estimator · Sketching

Editor: Jean-Philippe Vert.

B.D. Verma and R. Pratap are equal contributors of the paper.

✉ Bhisham Dev Verma
bhishamdevverma@gmail.com

Rameshwar Pratap
rameshwar.pratap@gmail.com

Manoj Thakur
manoj@iitmandi.ac.in

[1] IIT Mandi, Kamand, H.P., India

1 Introduction

Due to recent technological advancements, the last decade has witnessed a dramatic increase in the ability to collect data from various sources like social media platforms, mobile applications, finance, WWW, IoT, biology, remote sensing, etc. In many of these applications, the datasets are of terascale order, with the dimension being in the order of trillions (Agarwal et al., 2014; Wu et al., 2014; Zhai et al., 2014). Further, to get useful insight, we need to perform analytics on such high-dimensional datasets. Many fundamental algorithms such as clustering, classification, regression, nearest neighbor search, and ranking are basic subroutines of these analytics algorithms. However, running these algorithms/analytics on such high dimensional datasets becomes computationally expensive due to phenomena called the *"curse of dimensionality"* (Bellman, 1966).

To tackle the high dimensionality of the datasets, several dimensionality reduction algorithms have been proposed that compresses the dimension of the data while closely approximating the pairwise distances between the data points (Pratap et al., 2019; Johnson & Lindenstrauss 1983; Dasgupta et al., 2010; Weinberger et al., 2009; Broder et al., 1998; Charikar, 2002; Pratap et al., 2019; Pratap et al., 2018). As the geometry of the data points is preserved in the low-dimension, the corresponding results of the algorithms such as clustering, classification, regression, etc on the low-dimension also closely approximate the corresponding result in the full-dimension. It essentially leads to several benefits—for *e.g.*: smaller memory requirement; faster running time of the algorithms; faster inference/prediction time; smaller model size, *etc.*

Dimensionality reduction algorithms for the real-valued datasets can be broadly classified into two categories—a) random projection and b) feature hashing. The random projection approach is based on projecting the data matrix on a random matrix whose entries are sampled from a Gaussian distribution (Johnson & Lindenstrauss, 1983; Dasgupta et al., 2010; Dasgupta & Gupta, 2003; Charikar, 2002; Yu et al., 2014, or Bernoulli/sparse Bernoulli distribution Achlioptas, 2003; Li et al., 2006). The projected matrix is in low dimension and simultaneously approximates the original pairwise similarity/distance. On the other hand, the feature hashing (Weinberger et al., 2009) approach is based on randomly assigning each feature (dimension) into several bins, and a sketch value for each bin is generated by aggregating all the feature values fallen into the particular bin. Aggregating all the sketch values into a vector generates a low-dimensional representation of the input data. A major advantage of feature hashing over random projection is that 1) the compressed data preserves the sparsity of the input, and 2) it does not require any additional space to store the projection matrix [We refer the readers to Weinberger et al. (2009) for details]. This work focuses on the feature hashing algorithm for dimensionality reduction. We recall it as follows:

Definition 1 (Feature hashing - Definition 1 of Weinberger et al. 2009) Let $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k, \ldots, \alpha_N)$, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_k, \ldots, \beta_N) \in \mathbb{R}^N$ be N-dimensional sketches of input vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^D$, respectively, obtained from feature hashing algorithm such that

$$\alpha_k = \sum_{i=1}^{D} a_i g(i) z_i^{(k)},$$

$$\beta_k = \sum_{i=1}^{D} b_i g(i) z_i^{(k)},$$

where $g : [D] \mapsto \{-1, +1\}$, and $h : [D] \mapsto [N]$ are hash functions from 2-universal hash families, and $z_i^{(k)}$ is indicator of the event $h(i) = k$.

For a pair of vectors $\mathbf{a} = [a_1, a_2, \ldots, a_D]$ and $\mathbf{b} = [b_1, b_2, \ldots, b_D]$, the inner product of their respective sketches $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_N]$ and $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_N]$ (obtained using Definition 1) is an unbiased estimate of the inner product between $\mathbf{a}$ and $\mathbf{b}$. That is

$$\mathbb{E}[\langle \boldsymbol{\alpha}, \boldsymbol{\beta} \rangle] = \langle \mathbf{a}, \mathbf{b} \rangle. \tag{1}$$

One of the major applications of sketching results such as random projection and feature hashing is to compute the `Gram Matrix` of the given data matrix. Let $\mathbf{P} \in \mathbb{R}^{M \times D}$ be our input matrix, then its Gram Matrix is defined as $\mathbf{P}\mathbf{P}^T$. The computational complexity of computing the Gram Matrix on the full-dimensional dataset is $O(M^2 D)$, which may be impractical for large values of M and D. The inner product of the sketch of data points obtained using feature hashing is an unbiased estimate of the inner product between the original full dimensional data points. The complexity of computing the Gram matrix using feature hashing is $O(M^2 N + \text{nnz}(P)N)$ [due to Weinberger et al. (2009)], where N is the dimension of the sketch, and $\text{nnz}(P)$ denote the number of non-zero entries of $\mathbf{P}$. The savings from $O(M^2 D)$ to $O(M^2 N + \text{nnz}(P)N)$ is quite significant, especially when $N \ll D$. However, the variance of the similarity estimate tends to be large for the small values of N, making the estimate less reliable. We state the expression for the variance of the estimate as follows [see Theorem 3 and Weinberger et al. (2009)]

$$\text{Var}[\langle \boldsymbol{\alpha}, \boldsymbol{\beta} \rangle] = \frac{1}{N} \sum_{i \neq j, i, j = 1}^{D} \left(a_i^2 b_j^2 + a_i b_i a_j b_j \right). \tag{2}$$

In this work, we address this challenge and suggest practical and straightforward solutions for variance reduction in the inner product estimate of the feature hashing method (Weinberger et al., 2009). Our technique relies on the classical variance reduction techniques—control variate method (CV) and maximum likelihood estimator (MLE). In the following two subsections, we briefly discuss these two techniques and state our results.

Variance reduction using control variate trick: The control-variate is one of the classical techniques used for variance reduction in Monte-Carlo simulation (Lavenberg & Welch, 1981). We illustrate this with an example as follows: consider a process that generates a random variable Y, and we are interested in computing the term $\mathbb{E}[Y]$. Let us have another process for generating another random variable Z such that we know the exact value of its true mean $\mathbb{E}[Z]$. Then for any constant c, the expression $Y + c(Z - \mathbb{E}[Z])$ is an unbiased estimator of Y:

$$\mathbb{E}[Y + c \cdot (Z - \mathbb{E}[Z])] = \mathbb{E}[Y] + c \cdot \mathbb{E}[Z - \mathbb{E}[Z]] \quad = \mathbb{E}[Y] + 0 = \mathbb{E}[Y]. \tag{3}$$

The variance of $Y + c \cdot (Z - \mathbb{E}[Z])$ is given by

$$\text{Var}[Y + c \cdot (Z - \mathbb{E}[Z])] = \text{Var}[Y] + c^2 \cdot \text{Var}[Z] + 2c \cdot \text{Cov}[Y, Z]. \tag{4}$$

By elementary calculus, we can find the appropriate value of c, which minimizes the above expression. Suppose we denote that value by $\hat{c}$, then

$$\hat{c} = -\frac{\text{Cov}[Y, Z]}{\text{Var}[Z]}. \tag{5}$$

Equations (4), (5) give us the following

$$\text{Var}[Y + c \cdot (Z - \mathbb{E}[Z])] = \text{Var}[Y] - \frac{\text{Cov}[Y, Z]^2}{\text{Var}[Z]}. \tag{6}$$

To summarize the above, for a random variable Y, we generate another random variable $Y + c \cdot (Z - \mathbb{E}[Z])$ that gives an unbiased estimator of Y. Further, the variance of $Y + c \cdot (Z - \mathbb{E}[Z])$ is smaller than or equal to that of Y because the $\text{Cov}[Y, Z]^2/\text{Var}[Z]$ is always non-negative— with the equality if there is no correlation between Y and Z. The random variable Z is called control variate, and the term $\hat{c}$ is called the control variate coefficient.

Variance reduction using the maximum likelihood estimator (MLE): Maximum likelihood estimation (MLE) (Murphy 2013) is a popular statistical estimation method used for estimating parameters in statistical modeling. We discuss it as follows:

Suppose $X_1, X_2, \ldots, X_n \overset{\text{i.i.d.}}{\sim} f(x|\theta)$, where $f(x|\theta)$ represents a probability density function (PDF) when X is a continuous random variable, and represents a probability mass function (PMF) when X is a discrete random variable, and θ is an unknown parameter. For every observed sample $x_1, x_2, \ldots, x_n$, we define

$$L(\theta) = f(x_1, x_2, \ldots, x_n|\theta) = f(x_1|\theta) \cdot f(x_2|\theta) \ldots f(x_n|\theta) = \prod_{i=1}^{n} f(x_i|\theta). \tag{7}$$

We call $f(x_1, x_2, \ldots, x_n|\theta)$ as the *likelihood function* and denote it by $L(\theta)$. The term $L(\theta)$ consists of a product of n terms. Further, maximizing $L(\theta)$ is equivalent to maximizing $\log L(\theta)$, as log is a monotonically increasing function. We therefore focus on maximizing the term $\log L(\theta)$, which is called as the *log likelihood function*, and we denote it as $l(\theta)$, that is,

$$l(\theta) = \log L(\theta) = \log \prod_{i=1}^{n} f(x_i|\theta) = \sum_{i=1}^{n} f(x_i|\theta). \tag{8}$$

Our aim is to find a value of θ that maximizes the likelihood function $L(\theta)$ (or $l(\theta)$). Thus, the maximum likelihood estimator (MLE) of θ is defined as

$$\hat{\theta} = \text{argmax}_\theta L(\theta) = \text{argmax}_\theta l(\theta),$$

Where "arg" returns the argument at which the maxima is attained.

1.1 Our contribution

This work proposes two simple, effective, and practical approaches for variance reduction in the inner product estimate obtained via feature hashing. We use the above mentioned variance reduction techniques—control variate (CV) and maximum likelihood estimate (MLE) for this purpose. To apply these methods, we need to know the marginal norms (ℓ_2 norm) of the data points. We note that the computing norm of M data points in D-dimension is of complexity $O(MD)$, which can be computed by taking just one pass over the dataset. As mentioned earlier, the variance of the feature hashing estimator tends to be large for small values of N. Our both variance reduction methods mitigate this problem and suggest significant variance reduction in such a scenario. We require a word of notation to state our technical contribution. We mention it as follows: Suppose we have two D-dimensional data points $\mathbf{a} = [a_1, a_2, \ldots, a_D]$ and $\mathbf{b} = [b_1, b_2, \ldots, b_D]$ such that their squared ℓ_2 norms

and inner product are m_1, m_2, and λ, respectively—that is $m_1 = \sum_{i=1}^{D} a_i^2$, $m_2 = \sum_{i=1}^{D} b_i^2$ and $\lambda = \sum_{i=1}^{D} a_i b_i$. The vectors $\mathbf{a}$ and $\mathbf{b}$ are compressed into N dimensional real-valued vectors $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_k, \ldots, \alpha_N]$, and $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_k, \ldots, \beta_N]$, respectively, using feature hashing algorithm (Weinberger et al., 2009), where $N \ll D$.

– Our first estimator is based on the control variate method, and we refer it as control variate feature hashing (CV-FH). Our proposed estimator remains unbiased and offers significant variance reduction. If we denote $Y := \langle \boldsymbol{\alpha}, \boldsymbol{\beta} \rangle$ as the estimator of feature hashing, and $\hat{Y}_{CV}$ as the estimate obtained via our CV method, then we have

$$\begin{aligned} \text{Var}[\hat{Y}_{CV}] &= \text{Var}(Y) - \frac{2\lambda^2(m_1 + m_2)^2}{N(m_1^2 + m_2^2 + 2\lambda^2)}, \quad \text{where,} \\ \text{Var}[Y] &= \frac{1}{N} \sum_{i \neq j, i,j=1}^{D} \left(a_i^2 b_j^2 + a_i b_i a_j b_j \right) \text{ (due to Weinberger et al. 2009).} \end{aligned} \tag{9}$$

$$= \frac{1}{N} \left(m_1 m_2 + \lambda^2 - 2 \sum_{i=1}^{D} a_i^2 b_i^2 \right). \tag{10}$$

We state further details in Theorem 6.

– Our other estimator is based on the maximum likelihood estimate (MLE), and we refer to it as MLE feature hashing (MLE-FH). If $\hat{Y}_{MLE}$ denote our estimator of the MLE method, then

$$\text{Var}[\hat{Y}_{MLE}] = \frac{(m_1 m_2 - \lambda^2)^2}{N(m_1 m_2 + \lambda^2)}. \tag{11}$$

Further as $\lambda = \sqrt{m_1}\sqrt{m_2}\cos\theta$, where θ is the angle between $\mathbf{a}$ and $\mathbf{b}$, then Equation (11) simplifies to the following:

$$\text{Var}[\hat{Y}_{MLE}] = \frac{(1 - \cos^2(\theta))^2}{N(1 + \cos^2(\theta))}. \tag{12}$$

We state further details in Theorem 8.

– Both of our estimators are simple, practical, and effective. We show their applicability by performing extensive experiments on several real-world datasets (see Sect. 4).

A major bottleneck in applying CV and MLE techniques is to show that the sketch vector $[\boldsymbol{\alpha}, \boldsymbol{\beta}] = [\alpha_1, \ldots, \alpha_N, \beta_1, \ldots, \beta_N]$ follows multivariate normal distribution, where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are N dimensional sketches of vectors $\mathbf{a}$ and $\mathbf{b}$ obtained via feature hashing. We prove this (in Theorem 4) under the asymptotic convergence in distribution, as $D \to \infty$, and we use multivariate Lyapunov's central limit Theorem 2 for this purpose. Theorem 4 along with CV and MLE method leads to variance reduction in the inner product estimation.

Theoretical plotting on variance reduction: We wish to understand the variance reduction obtained by our proposals CV-FH and MLE-FH. To do so, we generate several pairs of real-valued vectors in the 5000 dimension such that the angle between them is θ, and their squared ℓ_2 norms are m_1 and m_2, respectively. We generate different such pairs for various values of $\theta \in \{10^o, 30^o, 60^o, 90^o\}$, and the ratio $m_2/m_1 = \{0.1, 0.4, 0.7, 1\}$. Then we compute the variance of our estimators $\hat{Y}_{MLE}$ and $\hat{Y}_{CV}$ (using Equations (11), (9)), and

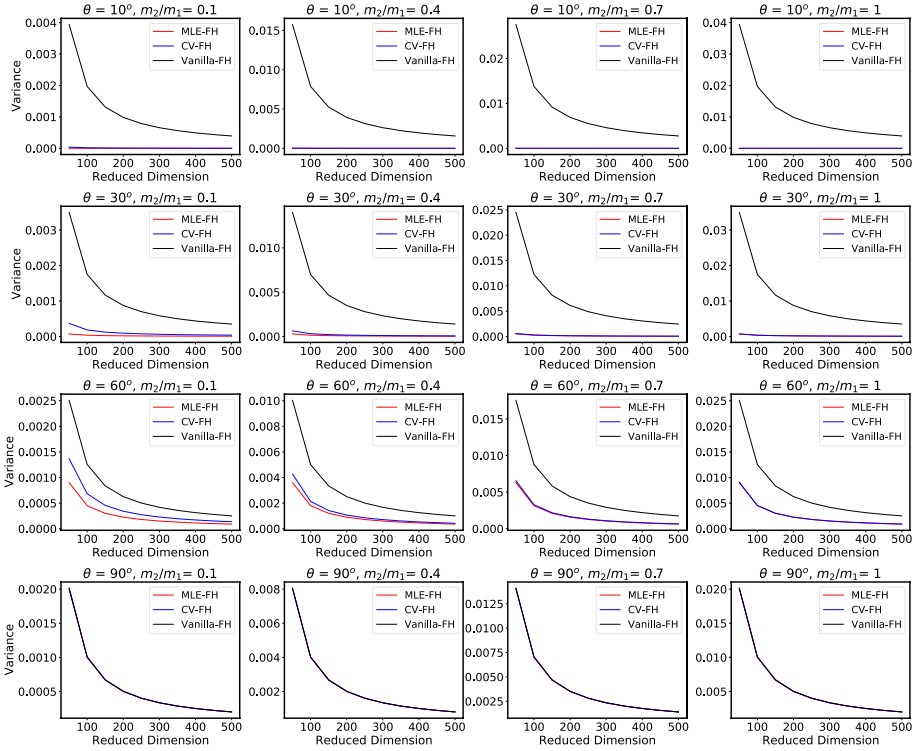

Fig. 1 Theoretical plots of variance for CV-FH, MLE-FH and Vanilla-FH feature hashing at different angles θ and different m_2/m_1 ratio

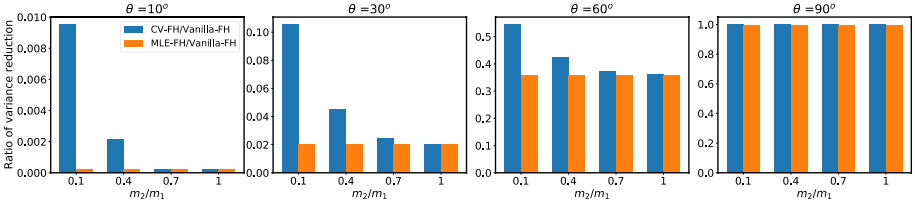

Fig. 2 Reduction ratio of variance for MLE-FH and CV-FH trick *w.r.t.* Vanilla-FH feature hashing at different m_2/m_1 ratio

the variance of vanilla feature hashing (Vanilla-FH) (using Eq. (10)), for several values of the reduced dimensions N. We summarize our observations in Fig. 1. We also note the corresponding ratio of variance reduction, that is, the ratio of the variance of CV-FH and MLE-FH with that of Vanilla-FH, and summarize it in Fig. 2. We note the following insights from the Figs.

– In Fig. 1, we observe that for smaller values of θ, the variances of our proposals CV-FH and MLE-FH are much smaller than that of Vanilla-FH and tend to increase with the increase of θ. Finally, all the three variances converge when $\theta = 90^o$. Furthermore, the

variance of MLE-FH is smaller than that of CV-FH, and both of them converge when $m_2/m_1 \to 1$.
– The variance of MLE-FH is independent of the ratio m_2/m_1 (Eq. (12)). Further, in Fig. 2, we observe that for smaller values of m_2/m_1, the ratio between the variances of MLE-FH and Vanilla-FH is much smaller than the corresponding ratio between the variances of CV-FH and Vanilla-FH. However, as $m_2/m_1 \to 1$, these ratios converge to each other. Therefore, when $m_2/m_1 \to 1$ is small, as the variance of MLE-FH is smaller, it gives a more accurate estimate of the inner product, whereas when $m_2/m_1 = 1$, both the estimators are almost equally accurate. We empirically observe that (see Sect. 4.3) the running time of CV-FH is much faster than that of MLE-FH. Therefore, CV-FH can potentially be used when $m_2/m_1 = 1$, as its running time is much faster than MLE-FH, and simultaneously it offers a similar variance.

Importance of variance reduction: The standard way to achieve the variance reduction in the pairwise similarity estimation is to generate several *i.i.d.* copies of the sketches (or hash values) of given input pairs. Needless to say, this is an expensive routine. A major advantage of our approach is that it significantly reduces the variance that occurred in the similarity estimation by proposing a new estimator that exploits the existing available sketches. A major advantage of our proposal is that it provides significant variance reductions by exploiting the existing available sketches and doesn't require generating their *i.i.d.* copies. As a consequence, we can achieve the same accuracy during the similarity estimation at a considerably lower reduced dimension.

1.2 Related work

Dimensionality reduction is a well-known algorithmic technique for compressing high-dimensional data to a low-dimensional format while approximating pairwise similarity/distance. Several dimensionality reduction techniques have been developed, considering the various data types and the underlying similarity measures. We mention a few notable results: Johnson Lindenstrauss (JL) lemma (or random projection) (Johnson & Lindenstrauss, 1983; Dasgupta & Gupta, 2003) and its improved variants (Dasgupta et al., 2010; Li & Li, 2019) suggest compressing real-valued vectors while preserving pairwise euclidean distance and inner product. Random projection using $\alpha-$stable distribution suggests compressing real-valued vectors while preserving the pairwise l_α distance $(0 < \alpha \leq 2)$ (Indyk, 2006; Li, 2007; Li, 2008; Li & Hastie, 2007; Li et al., 2006). Min-wise independent permutation (Broder et al. 1998) and its improved variants suggest compressing binary vectors (or sets) while preserving pairwise jaccard similarity (Li & König, 2011; Li et al., 2012; Shrivastava, 2017). Weighted minwise hashing algorithms (Ioffe, 2010; Shrivastava, 2016; Ertl, 2018; Wei et al., 2018) compress real vectors (or weighted sets) and preserve the weighted jaccard similarity (generalized jaccard similarity). Signed random projection (SimHash) (Charikar 2002) and its improved variants give compression of real-valued vectors while preserving pairwise cosine similarity (Yu et al., 2014; Ji et al., 2012; Shrivastava & Li, 2014; Li, 2019). BinSketch (Pratap et al. 2019) gives a compression of binary vectors while preserving multiple similarity measures such as hamming distance, inner product, cosine, and jaccard similarity in the same sketch. Feature hashing (Weinberger et al. 2009) suggests compression of real-valued vectors while preserving the pairwise inner product.

Most of these dimensionality reduction algorithms are randomized, and their similarity estimators incur high variance, especially at smaller reduced dimensions, making the estimate less reliable. To address this, several variance reduction techniques have been attempted. The control-variate (CV) and the maximum likelihood estimator (MLE) are two notable techniques in this regard. We discuss their known applications for variance reduction in dimensionality reduction algorithms as follows:

Both control variate and MLE methods have been used for variance reduction in random projection (Johnson & Lindenstrauss, 1983). The result of Li et al. (2006) suggests variance reduction for random projection using the MLE method, under the assumption that the marginal norms of the data points are known. Later Li et al. (2006) extends this and suggests variance reduction for very sparse random projection (Li et al., 2006). The result of Kang and Pin (2018) suggests variance reduction in the estimator of SimHash (Charikar 2002) using the MLE method. Typically in the MLE method, the technique includes formulating a cubic polynomial equation (using the marginal norms of the data points and their respective sketches) whose roots closely estimate the desired inner product between original data points. Kang and Hooker (2017), Kang (2017) exploit control variate method for improving the estimates of inner product and euclidean distance obtained from the random projection. They further extended their result by adding so-called dummy vectors in the dataset and obtained a better estimator for the inner product (Li et al., 2020). To the best of our knowledge, variance reduction techniques have not been attempted in the context of the feature hashing algorithm. In this work, we show that both MLE and CV offer a simple, effective and practical solution for variance reduction in the feature hashing estimator, which in turn offers a more accurate similarity estimation at the cost of little computational overhead.

Organization of the paper: The rest of the paper is organized as follows: in Sect. 2, we introduce the notations and state some preliminary results that are required to build the results stated in the paper. In Sect. 3 we give a theoretical analysis of the variance reduction obtained using CV and MLE methods. In Sect. 4, we complement our theoretical results and show their practical applicability by doing extensive experiments on several real-world datasets. Finally, in Sect. 5, we conclude our discussion and state some potential open questions of the work.

2 Background

The mathematical notations used in this paper are defined in Table 1.

We state the following lemma from Provost and Mathai (1992).

Lemma 1 (Provost and Mathai 1992) *Let* $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$, *and* $\mathbf{A}, \mathbf{B}$ *are symmetric matrices, then*

$$\begin{aligned} \mathrm{Var}[\mathbf{w^T A w}] &= 2\mathrm{Tr}[\mathbf{A\Sigma A\Sigma}], \\ \mathrm{Cov}[\mathbf{w^T A w}, \mathbf{w^T B w}] &= 2\mathrm{Tr}[\mathbf{A\Sigma B\Sigma}], \end{aligned}$$

where Tr *is the trace operator of the matrix.*

We use the following multivariate Lyapunov's central limit theorem.

Theorem 2 Multivariate Lyapunov CLT (Feller 1968) *Let* $\{\mathbf{X}_1, \dots \mathbf{X}_D\}$ *be a sequence of independent random vectors such that each entry of the both*

(i) *the expected value of the random vector* $\{\mathbf{X}_i\}_{i=1}^{D}$,
(ii) *and the corresponding covariance matrix* $\mathbf{\Sigma}_i$,

is finite. We define

$$\mathbf{V}_D = \sum_{i=1}^{D} \mathbf{\Sigma}_i.$$

If for some $\delta > 0$ *the following condition holds true*

$$\lim_{D\to\infty} ||\mathbf{V}_D^{-\frac{1}{2}}||^{2+\delta} \sum_{i=1}^{D} \mathbb{E}\left[||\mathbf{X}_i - \mathbb{E}[\mathbf{X}_i]||^{2+\delta}\right] = 0, \text{ then}$$

$$\mathbf{V}_D^{-\frac{1}{2}} \sum_{i=1}^{D} (\mathbf{X}_i - \mathbb{E}[\mathbf{X}_i]) \xrightarrow{d} \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

as D tends to infinity. Where $\xrightarrow{d}$ *denotes the convergence in distribution*; $\mathbf{0}$ *and* $\mathbf{I}$ *denote zero vector and the identity matrix, respectively.*

We state some results from Weinberger et al. (2009) which we require to prove our results.

Theorem 3 (Adapted from the results of Weinberger et al. 2009) *Given vectors* $\mathbf{a} = [a_1, \dots, a_D]$, $\mathbf{b} = [b_1, \dots, b_D]$ *get compressed into vectors* $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_k, \dots, \alpha_N]$ *and* $\boldsymbol{\beta} = [\beta_1, \dots, \beta_k, \dots, \beta_N]$, *respectively, using the feature hashing algorithm* (*stated in Definition* 1, *Weinberger et al.* 2009), *where* $1 \le k \le N$. *Then*

$$\mathbb{E}[\boldsymbol{\alpha}] = \mathbb{E}[\boldsymbol{\beta}] = \mathbf{0}. \tag{13}$$

$$\mathbb{E}[\langle \boldsymbol{\alpha}, \boldsymbol{\beta} \rangle] = \langle \mathbf{a}, \mathbf{b} \rangle. \tag{14}$$

$$\mathbb{E}[||\boldsymbol{\alpha}||^2 + ||\boldsymbol{\beta}||^2] = ||\mathbf{a}||^2 + ||\mathbf{b}||^2. \tag{15}$$

$$\text{Var}[\langle \boldsymbol{\alpha}, \boldsymbol{\beta} \rangle] = \frac{1}{N} \sum_{i\neq j, i,j=1}^{D} \left(a_i^2 b_j^2 + a_i b_i a_j b_j \right). \tag{16}$$

$$= \frac{1}{N} \left(||\mathbf{a}||^2 \cdot ||\mathbf{b}||^2 + \langle \mathbf{a}, \mathbf{b} \rangle^2 - 2 \sum_{i=1}^{D} a_i^2 b_i^2 \right). \tag{17}$$

3 Analysis

We first show that the vector $[\boldsymbol{\alpha}, \boldsymbol{\beta}]^T$ asymptotically converges to multivariate normal distribution. To prove the asymptotic results, we assume that the fourth moment of input features are bounded e.g. $\mathbb{E}[a_i^4] < \infty$, $\mathbb{E}[b_i^4] < \infty$ and $\mathbb{E}[a_i^2 b_i^2] < \infty$ (similar assumption to Li et al. 2006, Section 4). This assumption essentially states that all the input dimensions are almost equally important. Then using this result, we obtain our results on variance reduction.

Theorem 4 *If* $\forall i, 1 \leq i \leq D, \mathbb{E}\left[|a_i|^{2+\delta}\right], \mathbb{E}\left[|b_i|^{2+\delta}\right]$ *take finite, nonzero values, the pairwise angle between vectors* $\mathbf{a}$ *and* $\mathbf{b}$ *is non-zero, and* $N = o\left(D^{\frac{\delta}{2(\delta+2)}}\right)$ *for some* $\delta > 0$*, then as* $D \to \infty$*, we have*

$$\mathbf{V_D}^{-\frac{1}{2}} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} \xrightarrow{d} \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{18}$$

where $\xrightarrow{d}$ *denotes convergence in distribution,* $\mathbf{V_D} = \begin{bmatrix} \Sigma_{\boldsymbol{\alpha}} & \Sigma_{\boldsymbol{\alpha\beta}} \\ \Sigma_{\boldsymbol{\alpha\beta}} & \Sigma_{\boldsymbol{\beta}} \end{bmatrix}_{2N \times 2N}$, $\mathbf{0}$ *is a* $(2N \times 1)$ *dimensional vector with each entry as zero,* $\mathbf{I}$ *is an* $(2N \times 2N)$ *identity matrix, and*

$$\Sigma_{\boldsymbol{\alpha}} = \frac{1}{N} \begin{bmatrix} ||\mathbf{a}||^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & ||\mathbf{a}||^2 \end{bmatrix}_{N \times N},$$

$$\Sigma_{\boldsymbol{\alpha\beta}} = \frac{1}{N} \begin{bmatrix} \langle \mathbf{a}, \mathbf{b} \rangle & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \langle \mathbf{a}, \mathbf{b} \rangle \end{bmatrix}_{N \times N},$$

$$\Sigma_{\boldsymbol{\beta}} = \frac{1}{N} \begin{bmatrix} ||\mathbf{b}||^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & ||\mathbf{b}||^2 \end{bmatrix}_{N \times N}.$$

Proof Let us denote $\hat{\boldsymbol{\alpha}}_i = \left[a_i x_i z_i^{(1)}, \dots, a_i x_i z_i^{(N)}\right]^T$ and $\hat{\boldsymbol{\beta}}_i = \left[b_i x_i z_i^{(1)}, \dots, b_i x_i z_i^{(N)}\right]^T$. We define a sequence of $2N$ dimensional random vectors $\{\mathbf{X}_i\}_{i=1}^D$, which is obtained via concatenation of vectors $\hat{\boldsymbol{\alpha}}_i$ and $\hat{\boldsymbol{\beta}}_i$ as follows:

$$\begin{aligned} \mathbf{X}_i &= \begin{bmatrix} \hat{\boldsymbol{\alpha}}_i \\ \hat{\boldsymbol{\beta}}_i \end{bmatrix} \\ &= \left[a_i x_i z_i^{(1)}, \dots, a_i x_i z_i^{(N)}, b_i x_i z_i^{(1)}, \dots, b_i x_i z_i^{(N)}\right]^T. \end{aligned} \tag{19}$$

We compute the expected value and covariance matrix of the vector $\mathbf{X}_i$ as follows.

$$\mathbb{E}[\mathbf{X}_i] = \mathbb{E}\begin{bmatrix} a_i x_i z_i^{(1)} \\ \vdots \\ a_i x_i z_i^{(N)} \\ b_i x_i z_i^{(1)} \\ \vdots \\ b_i x_i z_i^{(N)} \end{bmatrix} = \begin{bmatrix} a_i \mathbb{E}[x_i] \cdot \mathbb{E}[z_i^{(1)}] \\ \vdots \\ a_i \mathbb{E}[x_i] \cdot \mathbb{E}[z_i^{(N)}] \\ b_i \mathbb{E}[x_i] \cdot \mathbb{E}[z_i^{(1)}] \\ \vdots \\ b_i \mathbb{E}[x_i] \cdot \mathbb{E}[z_i^{(N)}] \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ \cdot \\ \cdot \\ \vdots \\ 0 \end{bmatrix}. \tag{20}$$

$$\boldsymbol{\Sigma}_i = \text{Cov}[\mathbf{X}_i] = \begin{bmatrix} \text{Cov}[\hat{\boldsymbol{\alpha}}_i] & \text{Cov}[\hat{\boldsymbol{\alpha}}_i, \hat{\boldsymbol{\beta}}_i] \\ \text{Cov}[\hat{\boldsymbol{\alpha}}_i, \hat{\boldsymbol{\beta}}_i] & \text{Cov}[\hat{\boldsymbol{\beta}}_i] \end{bmatrix}_{2N \times 2N} \tag{21}$$

where,

$$\text{Cov}[\hat{\boldsymbol{\alpha}}_i] = \frac{1}{N} \begin{bmatrix} a_i^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & a_i^2 \end{bmatrix}_{N \times N}, \tag{22}$$

$$\text{Cov}[\hat{\boldsymbol{\alpha}}_i, \hat{\boldsymbol{\beta}}_i] = \frac{1}{N} \begin{bmatrix} a_i b_i & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & a_i b_i \end{bmatrix}_{N \times N}, \tag{23}$$

$$\text{Cov}[\hat{\boldsymbol{\beta}}_i] = \frac{1}{N} \begin{bmatrix} b_i^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & b_i^2 \end{bmatrix}_{N \times N}. \tag{24}$$

Equations (22), (23), and (24) holds due to the following, where $1 \leq k \leq N$.

$$\mathbb{E}\left[a_i x_i z_i^{(k)}\right] = \mathbb{E}\left[b_i x_i z_i^{(k)}\right] = 0, \qquad \left(\because \mathbb{E}[x_i] = 0\right). \tag{25}$$

$$\begin{aligned} \text{Cov}\left[a_i x_i z_i^{(k)}, a_i x_i z_i^{(k)}\right] &= \mathbb{E}\left[\left(a_i x_i z_i^{(k)}\right)^2\right] - \left(\mathbb{E}\left[a_i x_i z_i^{(k)}\right]\right)^2. \\ &= a_i^2 \mathbb{E}\left[x_i^2\right] \mathbb{E}\left[z_i^{(k)}\right], \qquad \left(\because \left(z_i^{(k)}\right)^2 = z_i^{(k)}\right). \\ &= \frac{a_i^2}{N}, \qquad \left(\because \mathbb{E}\left[x_i^2\right] = 1 \text{ and } \mathbb{E}\left[z_i^{(k)}\right] = \frac{1}{N}\right). \end{aligned} \tag{26}$$

$$\begin{aligned} \text{Cov}[a_i x_i z_i^{(k)}, b_i x_i z_i^{(k)}] &= \mathbb{E}\left[\left(a_i x_i z_i^{(k)}\right)\left(b_i x_i z_i^{(k)}\right)\right] - \mathbb{E}\left[a_i x_i z_i^{(k)}\right] \mathbb{E}\left[b_i x_i z_i^{(k)}\right]. \\ &= a_i b_i \mathbb{E}\left[x_i^2\right] \mathbb{E}\left[z_i^{(k)}\right]. \\ &= \frac{a_i b_i}{N}. \end{aligned} \tag{27}$$

Similarly,

$$\text{Cov}[a_i x_i z_i^{(k)}, a_j x_j z_j^{(k)}] = 0, \qquad \text{where } i \neq j.$$

$$\text{Cov}[a_i x_i z_i^{(k)}, b_j x_j z_j^{(k)}] = 0, \qquad \text{where } i \neq j.$$

We need to calculate

$$\mathbb{E}[||\mathbf{X}_i - \mathbb{E}[\mathbf{X}_i]||^{2+\delta}] = \mathbb{E}[||\mathbf{X}_i||^{2+\delta}]. \tag{28}$$

To do so, we first calculate the following using Eq. (19)

$$
\begin{aligned}
||\mathbf{X}_i||^2 &= a_i^2 x_i^2 (z_i^{(1)})^2 + \cdots + a_i^2 x_i^2 (z_i^{(N)})^2 \\
&\quad + b_i^2 x_i^2 (z_i^{(1)})^2 + \cdots + b_i^2 x_i^2 (z_i^{(N)})^2. \\
&= \sum_{k=1}^{N} (a_i^2 + b_i^2) x_i^2 (z_i^{(k)})^2. \\
&= (a_i^2 + b_i^2) \sum_{k=1}^{N} z_i^{(k)} = (a_i^2 + b_i^2). \\
\implies ||\mathbf{X}_i|| &= (a_i^2 + b_i^2)^{\frac{1}{2}}.
\end{aligned} \tag{29}
$$

Equations (28), (29), and (20) give us the following:

$$
\begin{aligned}
\mathbb{E}\left[||\mathbf{X}_i - \mathbb{E}[\mathbf{X}_i]||^{2+\delta}\right] &= \mathbb{E}\left[||\mathbf{X}_i||^{2+\delta}\right]. \\
&= \mathbb{E}\left[(a_i^2 + b_i^2)^{\frac{2+\delta}{2}}\right] = (a_i^2 + b_i^2)^{\frac{2+\delta}{2}}.
\end{aligned} \tag{30}
$$

We now compute $\mathbf{V}_D$ using Eq. (21)

$$
\begin{aligned}
\mathbf{V}_D &= \sum_{i=1}^{D} \mathbf{\Sigma}_i = \sum_{i=1}^{D} \mathrm{Cov}(X_i). \\
&= \begin{bmatrix} \sum_{i=1}^{D} \mathrm{Cov}[\hat{\boldsymbol{\alpha}}_i] & \sum_{i=1}^{D} \mathrm{Cov}[\hat{\boldsymbol{\alpha}}_i, \hat{\boldsymbol{\beta}}_i] \\ \sum_{i=1}^{D} \mathrm{Cov}[\hat{\boldsymbol{\alpha}}_i, \hat{\boldsymbol{\beta}}_i] & \sum_{i=1}^{D} \mathrm{Cov}[\hat{\boldsymbol{\beta}}_i] \end{bmatrix}_{2N \times 2N}. \\
&= \frac{1}{N} \begin{bmatrix} ||\mathbf{a}||^2 & \dots & 0 & \langle \mathbf{a}, \mathbf{b} \rangle & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & ||\mathbf{a}||^2 & 0 & \dots & \langle \mathbf{a}, \mathbf{b} \rangle \\ \langle \mathbf{a}, \mathbf{b} \rangle & \dots & 0 & ||\mathbf{b}||^2 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \langle \mathbf{a}, \mathbf{b} \rangle & 0 & \dots & ||\mathbf{b}||^2 \end{bmatrix}_{2N \times 2N}. \\
&= \begin{bmatrix} \Sigma_{\alpha} & \Sigma_{\alpha\beta} \\ \Sigma_{\alpha\beta} & \Sigma_{\beta} \end{bmatrix}_{2N \times 2N},
\end{aligned} \tag{31}
$$

where matrices $\Sigma_{\alpha}, \Sigma_{\alpha\beta}, \Sigma_{\beta}$ are defined in the theorem statement. Note that the matrix $\mathbf{V}_D$ is a symmetric positive definite matrix. We calculate the symmetric positive definite matrix $\mathbf{V}_D^{-1}$ and its value is

$$
\mathbf{V}_D^{-1} = \frac{N}{\left(||\mathbf{a}||^2 ||\mathbf{b}||^2 - \langle \mathbf{a}, \mathbf{b} \rangle^2\right)} \begin{bmatrix} ||\mathbf{b}||^2 & \dots & 0 & -\langle \mathbf{a}, \mathbf{b} \rangle & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & ||\mathbf{b}||^2 & 0 & \dots & -\langle \mathbf{a}, \mathbf{b} \rangle \\ -\langle \mathbf{a}, \mathbf{b} \rangle & \dots & 0 & ||\mathbf{a}||^2 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & -\langle \mathbf{a}, \mathbf{b} \rangle & 0 & \dots & ||\mathbf{a}||^2 \end{bmatrix}_{2N \times 2N}. \tag{32}
$$

Note that if the pairwise angle between $\mathbf{a}, \mathbf{b}$ is zero, then $\langle \mathbf{a}, \mathbf{b} \rangle = ||\mathbf{a}||||\mathbf{b}||$ which makes the above expression undefined.

We know the facts that for any matrix $\mathbf{M}$, $||\mathbf{M}||_F^2 = \text{Tr}(\mathbf{M}\mathbf{M}^T) = \text{Tr}(\mathbf{M}^T\mathbf{M})$ and for any positive definite matrix $\mathbf{P}$, there exists a unique symmetric matrix $\mathbf{Q}$ such that $\mathbf{P} = \mathbf{Q}\mathbf{Q}$. Matrix $\mathbf{Q}$ is called square root of matrix $\mathbf{P}$. Hence, from these facts, we have

$$\begin{aligned} ||\mathbf{V}_D^{-1/2}||_F^2 &= \text{Tr}\Big(\mathbf{V}_D^{-1/2}(\mathbf{V}_D^{-1/2})^T\Big). \\ &= \text{Tr}\Big(\mathbf{V}_D^{-1/2}\mathbf{V}_D^{-1/2}\Big) \qquad \Big[\because V_D^{-1/2} \text{ is symmetric}\Big]. \\ &= \text{Tr}(\mathbf{V}_D^{-1}). \\ &= \frac{N^2\big(||\mathbf{a}||^2 + ||\mathbf{b}||^2\big)}{\big(||\mathbf{a}||^2||\mathbf{b}||^2 - \langle \mathbf{a}, \mathbf{b}\rangle^2\big)}. \end{aligned} \tag{33}$$

We need to show

$$\begin{aligned} &\lim_{D\to\infty} ||\mathbf{V}_D{}^{-\frac{1}{2}}||_F^{2+\delta} \sum_{i=1}^{D} \mathbb{E}[||X_i||^{2+\delta}] = 0. \\ &||\mathbf{V}_D{}^{-\frac{1}{2}}||_F^{2+\delta} \sum_{i=1}^{D} \mathbb{E}[||X_i||^{2+\delta}] \\ &\quad = \left(\frac{(||\mathbf{a}||^2 + ||\mathbf{b}||^2)N^2}{||\mathbf{a}||^2||\mathbf{b}||^2 - \langle \mathbf{a}, \mathbf{b}\rangle^2}\right)^{\frac{2+\delta}{2}} \sum_{i=1}^{D} (a_i^2 + b_i^2)^{\frac{2+\delta}{2}}. \end{aligned} \tag{34}$$

$$\begin{aligned} &= N^{2+\delta}\left(\frac{||\mathbf{a}||^2 + ||\mathbf{b}||^2}{||\mathbf{a}||^2||\mathbf{b}||^2 - \langle \mathbf{a}, \mathbf{b}\rangle^2}\right)^{\frac{2+\delta}{2}} \sum_{i=1}^{D} (a_i^2 + b_i^2)^{\frac{2+\delta}{2}}. \\ &= \frac{N^{2+\delta}}{D^{\frac{\delta}{2}}}\left(\frac{\frac{||\mathbf{a}||^2}{D} + \frac{||\mathbf{b}||^2}{D}}{\frac{||\mathbf{a}||^2}{D}\frac{||\mathbf{b}||^2}{D} - \frac{\langle \mathbf{a}, \mathbf{b}\rangle^2}{D^2}}\right)^{\frac{2+\delta}{2}} \sum_{i=1}^{D}\left(\frac{(a_i^2 + b_i^2)^{\frac{2+\delta}{2}}}{D}\right). \\ &= \frac{N^{2+\delta}}{D^{\frac{\delta}{2}}}\left(\frac{\sum_{i=1}^{D}\frac{a_i^2}{D} + \sum_{i=1}^{D}\frac{b_i^2}{D}}{\sum_{i=1}^{D}\frac{a_i^2}{D}\sum_{i=1}^{D}\frac{b_i^2}{D} - \left(\sum_{i=1}^{D}\frac{a_i b_i}{D}\right)^2}\right)^{\frac{2+\delta}{2}} \sum_{i=1}^{D}\left(\frac{(a_i^2 + b_i^2)^{\frac{2+\delta}{2}}}{D}\right). \\ &= \frac{N^{2+\delta}}{D^{\frac{\delta}{2}}}\left(\frac{\mathbb{E}[a_i^2] + \mathbb{E}[b_i^2]}{\mathbb{E}[a_i^2]\mathbb{E}[b_i^2] - \big(\mathbb{E}[a_i b_i]\big)^2}\right)^{\frac{2+\delta}{2}} \mathbb{E}\Big[(a_i^2 + b_i^2)^{\frac{2+\delta}{2}}\Big]. \\ &= \left(\frac{N}{D^{\frac{\delta}{2(\delta+2)}}}\right)^{2+\delta}\left(\frac{\mathbb{E}[a_i^2] + \mathbb{E}[b_i^2]}{\mathbb{E}[a_i^2]\mathbb{E}[b_i^2] - \big(\mathbb{E}[a_i b_i]\big)^2}\right)^{\frac{2+\delta}{2}} \mathbb{E}\Big[(a_i^2 + b_i^2)^{\frac{2+\delta}{2}}\Big]. \\ &\le \left(\frac{N}{D^{\frac{\delta}{2(\delta+2)}}}\right)^{2+\delta}\left(\frac{\mathbb{E}[a_i^2] + \mathbb{E}[b_i^2]}{\mathbb{E}[a_i^2]\mathbb{E}[b_i^2] - \big(\mathbb{E}[a_i b_i]\big)^2}\right)^{\frac{2+\delta}{2}} \mathbb{E}\Big[\big(2\max\{a_i^2, b_i^2\}\big)^{\frac{2+\delta}{2}}\Big]. \\ &= \left(\frac{N}{D^{\frac{\delta}{2(\delta+2)}}}\right)^{2+\delta}\left(\frac{\mathbb{E}[a_i^2] + \mathbb{E}[b_i^2]}{\mathbb{E}[a_i^2]\mathbb{E}[b_i^2] - \big(\mathbb{E}[a_i b_i]\big)^2}\right)^{\frac{2+\delta}{2}} 2^{\frac{2+\delta}{2}} \mathbb{E}\Big[\max\{|a_i|^{(2+\delta)}, |b_i|^{(2+\delta)}\}\Big]. \\ &\le \left(\frac{N}{D^{\frac{\delta}{2(\delta+2)}}}\right)^{2+\delta}\left(\frac{\mathbb{E}[a_i^2] + \mathbb{E}[b_i^2]}{\mathbb{E}[a_i^2]\mathbb{E}[b_i^2] - \big(\mathbb{E}[a_i b_i]\big)^2}\right)^{\frac{2+\delta}{2}} 2^{\frac{2+\delta}{2}} \mathbb{E}\Big[|a_i|^{(2+\delta)} + |b_i|^{(2+\delta)}\Big]. \\ &\le \left(\frac{N}{D^{\frac{\delta}{2(\delta+2)}}}\right)^{2+\delta}\left(\frac{\mathbb{E}[a_i^2] + \mathbb{E}[b_i^2]}{\mathbb{E}[a_i^2]\mathbb{E}[b_i^2] - \big(\mathbb{E}[a_i b_i]\big)^2}\right)^{\frac{2+\delta}{2}} 2^{\frac{2+\delta}{2}} \Big(\mathbb{E}\Big[|a_i|^{(2+\delta)}\Big] + \mathbb{E}\Big[|b_i|^{(2+\delta)}\Big]\Big). \end{aligned} \tag{35}$$

$$\to 0 \quad as \quad D \to \infty. \tag{36}$$

Equation (34) holds due to Eq. (33) along with Eq. (30). Finally, Eq. (36) holds due to Theorem 2 and due to the fact that $N = o\left(D^{\frac{\delta}{2(\delta+2)}}\right)$, $\mathbb{E}\left[|a_i|^{(2+\delta)}\right]$, and $\mathbb{E}\left[|b_i|^{(2+\delta)}\right]$ have finite nonzero limit. Thus due to Theorem 2, we have

$$\begin{aligned} &\mathbf{V}_D^{-1/2} \sum_{i=1}^{D} \mathbf{X}_i \xrightarrow{d} \mathcal{N}(\mathbf{0}, \mathbf{I}). \\ \implies &\mathbf{V}_D^{-1/2} \sum_{i=1}^{D} \begin{bmatrix} \boldsymbol{\alpha}_i \\ \boldsymbol{\beta}_i \end{bmatrix} = \mathbf{V}_D^{-1/2} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} \xrightarrow{d} \mathcal{N}(\mathbf{0}, \mathbf{I}). \end{aligned} \tag{37}$$

□

Effect of Sparsity on Theorem 4: Let s denotes the sparsity of the input vectors $\mathbf{a}$ and $\mathbf{b}$ *i.e.* $s = \max\{\text{nnz}(\mathbf{a}), \text{nnz}(\mathbf{b})\}$, where $\text{nnz}(\cdot)$ returns the number of nonzero entries of the input vector. Recall that the Theorem 4 requires $\mathbb{E}\left[|a_i|^{2+\delta}\right], \mathbb{E}\left[|b_i|^{2+\delta}\right]$ to have finite non zero limits. We note that input sparsity s, crucially affects this distributional assumption. We require $s = O(D)$ to satisfy this assumption[1]. On the contrary, if $s = o(D)$, then $\mathbb{E}\left[|a_i|^{2+\delta}\right]$, $\mathbb{E}\left[|b_i|^{2+\delta}\right]$ tends to zero as $D \to \infty$, which do not satisfy the assumptions of the Theorem 4.

Corollary 5 *Following the assumption stated in Theorem 4,*

$$\mathbf{V}_D^{-1/2} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} \xrightarrow{d} \mathcal{N}(\mathbf{0}, \mathbf{I}) \implies \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} \xrightarrow{d} \mathcal{N}\left(\mathbf{0}, \mathbf{V}_D\right). \tag{38}$$

Proof We know that for any random vector $\mathbf{X}$ of dimension $N \times 1$ and any random matrix $\mathbf{A}$ of dimension $N \times N$

$$\begin{aligned} \mathbb{E}[\mathbf{AX}] &= \mathbf{A} \cdot \mathbb{E}[\mathbf{X}]. \\ \text{Cov}(\mathbf{AX}) &= \mathbf{A} \cdot \text{Cov}(\mathbf{X}) \cdot \mathbf{A}^T. \end{aligned}$$

Therefore,

$$\begin{aligned} \mathbb{E}\left[\mathbf{V}_D^{\frac{1}{2}} \mathbf{V}_D^{-1/2} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix}\right] &= \mathbf{V}_D^{\frac{1}{2}} \mathbb{E}\left[\mathbf{V}_D^{-1/2} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix}\right]. \\ &= \mathbf{V}_D^{\frac{1}{2}} \mathbf{0} = \mathbf{0}. \\ \implies \mathbb{E}\left[\mathbf{V}_D^{\frac{1}{2}} \mathbf{V}_D^{-1/2} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix}\right] &= \mathbb{E}\begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} = \mathbf{0}. \end{aligned} \tag{39}$$

[1] We note that

$$\mathbb{E}\left[|a_i|^{(2+\delta)}\right] = \frac{\sum_{i=1}^{D} |a_i|^{(2+\delta)}}{D} \leq \frac{s_a \cdot \left(\max\{|a_i|\}_{i=1}^{D}\right)^{(2+\delta)}}{D} = \frac{s_a \cdot \phi^{(2+\delta)}}{D},$$

where s_a denotes the number of nonzero entries in input vector $\mathbf{a}$ and $\phi = \max\{|a_i|\}_{i=1}^{D}$. For finite value of $\phi^{(2+\delta)}$, it is easy to see that

$$\frac{s_a \cdot \phi^{(2+\delta)}}{D} \to \begin{cases} 0 & \text{if } s_a = o(D), \\ \text{finite non zero} & \text{if } s_a = O(D), \end{cases} \quad \text{as} \quad D \to \infty.$$

$$\begin{aligned}
\text{Cov}\left(\mathbf{V}_D^{\frac{1}{2}}\mathbf{V}_D^{-1/2}\begin{bmatrix}\boldsymbol{\alpha}\\ \boldsymbol{\beta}\end{bmatrix}\right) &= \mathbf{V}_D^{\frac{1}{2}}\text{Cov}\left(\mathbf{V}_D^{-1/2}\begin{bmatrix}\boldsymbol{\alpha}\\ \boldsymbol{\beta}\end{bmatrix}\right)\left(\mathbf{V}_D^{\frac{1}{2}}\right)^T.\\
&= \mathbf{V}_D^{\frac{1}{2}}\mathbf{I}\left(\mathbf{V}_D^{\frac{1}{2}}\right)^T.\\
&= \mathbf{V}_D^{\frac{1}{2}}\left(\mathbf{V}_D^{\frac{1}{2}}\right)^T.\\
&= \mathbf{V}_D.\\
\implies \text{Cov}\left(\begin{bmatrix}\boldsymbol{\alpha}\\ \boldsymbol{\beta}\end{bmatrix}\right) &= \mathbf{V}_D.
\end{aligned} \tag{40}$$

Equation (39) and (40) implies

$$\begin{bmatrix}\boldsymbol{\alpha}\\ \boldsymbol{\beta}\end{bmatrix} \xrightarrow{d} \mathcal{N}(\mathbf{0}, \mathbf{V}_D).$$

□

Remark 1 In the proof of Theorem 4, the random variable x_i (Eq. (19)) takes value ± 1 with probability 1/2. In order to to prove Theorem 4, we only require the values of $\mathbb{E}[x_i]$, $\mathbb{E}[x_i^2]$, and $\text{Var}[x_i]$. We note that even if $x_i \sim \mathcal{N}(0,1)$ (instead of $\{-1,+1\}$ with probability 1/2) the corresponding values of these expressions are the same. Therefore, the proof of Theorem 4 holds for $x_i \sim \mathcal{N}(0,1)$ as well.

3.1 Variance reduction using control variate method:

In the following, we present our result of applying the control variate method to reduce variance in the inner product estimation.

Theorem 6 *Let $Y := \langle \boldsymbol{\alpha}, \boldsymbol{\beta} \rangle$ be the random variable denoting the estimate of $\langle \mathbf{a}, \mathbf{b} \rangle$ in the feature hashing (Weinberger et al., 2009) (see Algorithm 1, Theorem 3). Then there exists a control variate random variable Z, and the corresponding control variate coefficient $\hat{c}$ such that the variance of the control variate estimator, denoted by $\hat{Y}_{CV}$, is the following*:

$$\begin{aligned}
\text{Var}[\hat{Y}_{CV}] &= \text{Var}(Y + \hat{c}(Z - \mathbb{E}[Z])).\\
&= \text{Var}(Y) - \frac{2\lambda^2(m_1+m_2)^2}{N(m_1^2+m_2^2+2\lambda^2)}, \text{ where}\\
\text{Var}(Y) &= \text{Var}[\langle \boldsymbol{\alpha}, \boldsymbol{\beta} \rangle] = \frac{1}{N}\sum_{i\neq j, i,j=1}^{D}\left(a_i^2 b_j^2 + a_i b_i a_j b_j\right),
\end{aligned}$$

as noted in Weinberger et al. (2009) (Theorem 3, Equation (16)).

Proof We first state the estimator of the feature hashing (denoted as Y) and our control variate random variable (denoted as Z) as follows:

$$Y = \mathbf{w}^T\mathbf{A}\mathbf{w} = \langle \boldsymbol{\alpha}, \boldsymbol{\beta} \rangle. \tag{41}$$

$$
\begin{aligned}
Z &= \mathbf{w}^T\mathbf{B}\mathbf{w} = ||\boldsymbol{\alpha}||^2 + ||\boldsymbol{\beta}||^2, \qquad \text{where} \\
\mathbf{w} &= \left[\boldsymbol{\alpha}, \boldsymbol{\beta}\right]^T = \left[\alpha_1, \ldots, \alpha_N, \beta_1, \ldots, \beta_N\right]^T, \\
\mathbf{A} &= \begin{bmatrix} 0 & \ldots & 0 & 1/2 & \ldots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & 0 & 0 & \ldots & 1/2 \\ 1/2 & \ldots & 0 & 0 & \ldots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & 1/2 & 0 & \ldots & 0 \end{bmatrix}_{2N\times 2N}, \\
\mathbf{B} &= \begin{bmatrix} 1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & 1 \end{bmatrix}_{2N\times 2N}.
\end{aligned} \tag{42}
$$

Here $\mathbf{A}$ and $\mathbf{B}$ are symmetric matrices. We state the covariance of $\mathbf{w}$ as follows which can be easily computed using Theorem 3. For brevity, in the following Equation we denote $||\mathbf{a}||^2, ||\mathbf{b}||^2$ and $\langle \mathbf{a}, \mathbf{b}\rangle$ with m_1, m_2 and λ, respectively.

$$
\Sigma_{\mathbf{w}} = \frac{1}{N}\begin{bmatrix} m_1 & \ldots & 0 & \lambda & \ldots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & m_1 & 0 & \ldots & \lambda \\ \lambda & \ldots & 0 & m_2 & \ldots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & \lambda & 0 & \ldots & m_2 \end{bmatrix}_{2N\times 2N}. \tag{43}
$$

Due to Theorem 3, we have

$$
\mathbb{E}[Y] = \mathbb{E}[\langle \boldsymbol{\alpha}, \boldsymbol{\beta}\rangle] = \langle \mathbf{a}, \mathbf{b}\rangle. \tag{44}
$$

$$
\mathbb{E}[Z] = \mathbb{E}\left[||\boldsymbol{\alpha}||^2 + ||\boldsymbol{\beta}||^2\right] = ||\mathbf{a}||^2 + ||\mathbf{b}||^2. \tag{45}
$$

We need to estimate the term $\mathrm{Cov}[Y, Z]$ and $\mathrm{Var}[Z]$ to compute the control variate coefficient term $\hat{c}$, and the variance reduction term $\mathrm{Cov}[Y, Z]^2/\mathrm{Var}[Z]$ (see Eqs. (5),(6)). Note that in Theorem 4 we show that $\mathbf{w}$ is normal. Further matrices $\mathbf{A}$ and $\mathbf{B}$ are symmetric. Therefore we can use Lemma 1 to compute the terms $\mathrm{Var}[Z]$ and $\mathrm{Cov}[Y, Z]$.

$$
\begin{aligned}
\mathrm{Var}[Z] &= \mathrm{Var}\left[\mathbf{w}^T\mathbf{B}\mathbf{w}\right] = 2\mathrm{Tr}\left[\mathbf{B}\times\Sigma_{\mathbf{w}}\times\mathbf{B}\times\Sigma_{\mathbf{w}}\right]. \\
&= \frac{2}{N^2}\mathrm{Tr}\begin{bmatrix} m_1^2+\lambda^2 & \ldots & 0 & \lambda(m_1+m_2) & \ldots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & m_1^2+\lambda^2 & 0 & \ldots & \lambda(m_1+m_2) \\ \lambda(m_1+m_2) & \ldots & 0 & m_2^2+\lambda^2 & \ldots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & \lambda(m_1+m_2) & 0 & \ldots & m_2^2+\lambda^2 \end{bmatrix}_{2N\times 2N}.
\end{aligned} \tag{46}
$$

$$
= \frac{2N(m_1^2+m_2^2+2\lambda^2)}{N^2} = \frac{2(m_1^2+m_2^2+2\lambda^2)}{N}. \tag{47}
$$

Equation (46) obtained after computing the term $\mathbf{B}\times\Sigma_{\mathbf{w}}\times\mathbf{B}\times\Sigma_{\mathbf{w}}$. Similarly, we compute the term $\mathrm{Cov}[Y, Z]$ using the Lemma 1 as follows

$$
\begin{aligned}
\mathrm{Cov}[Y,Z] &= \mathrm{Cov}\left[\mathbf{w}^T\mathbf{A}\mathbf{w}, \mathbf{w}^T\mathbf{B}\mathbf{w}\right].\\
&= 2\mathrm{Tr}\left[\mathbf{A}\times\Sigma_{\mathbf{w}}\times\mathbf{B}\times\Sigma_{\mathbf{w}}\right].\\
&= \frac{1}{N^2}\mathrm{Tr}\begin{bmatrix}
\lambda(m_1+m_2) & \dots & 0 & m_2^2+\lambda^2 & \dots & 0\\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\
0 & \dots & \lambda(m_1+m_2) & 0 & \dots & m_2^2+\lambda^2\\
m_1^2+\lambda^2 & \dots & 0 & \lambda(m_1+m_2) & \dots & 0\\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\
0 & \dots & m_1^2+\lambda^2 & 0 & \dots & \lambda(m_1+m_2)
\end{bmatrix}_{2N\times 2N}.\\
&= \frac{2N\lambda(m_1+m_2)}{N^2} = \frac{2\lambda(m_1+m_2)}{N}.
\end{aligned}
\tag{48}
$$

Equation (48) along with Eq. (47) give the control variate coefficient and variance reduction as follows (see Equations (5), (6)):

$$
\begin{aligned}
\hat{c} &= -\frac{\mathrm{Cov}[Y,Z]}{\mathrm{Var}[Z]} = -\frac{\frac{2\lambda(m_1+m_2)}{N}}{\frac{2}{N}(m_1^2+m_2^2+2\lambda^2)}.\\
&= -\frac{\lambda(m_1+m_2)}{(m_1^2+m_2^2+2\lambda^2)}.
\end{aligned}
\tag{49}
$$

$$
\begin{aligned}
\text{Variance Reduction} &= \frac{\mathrm{Cov}[Y,Z]^2}{\mathrm{Var}[Z]} = \frac{\left(\frac{2\lambda(m_1+m_2)}{N}\right)^2}{\frac{2}{N}(m_1^2+m_2^2+2\lambda^2)}.\\
&= \frac{2\lambda^2(m_1+m_2)^2}{N(m_1^2+m_2^2+2\lambda^2)}.
\end{aligned}
\tag{50}
$$

Equation (41), (42) and (45) along with Eq. (49) gives the following control variate estimator

$$
\begin{aligned}
\hat{Y}_{CV} &= Y + \hat{c}\cdot(Z-\mathbb{E}[Z]).\\
&= Y - \frac{\lambda(m_1+m_2)}{(m_1^2+m_2^2+2\lambda^2)}[||\boldsymbol{\alpha}||^2+||\boldsymbol{\beta}||^2-m_1-m_2].
\end{aligned}
\tag{51}
$$

Equation (6) along with Eq. (50) gives the following

$$
\begin{aligned}
\mathrm{Var}[Y+c.(Z-\mathbb{E}(Z))] &= \mathrm{Var}[Y] - \frac{\mathrm{Cov}[Y,Z]^2}{\mathrm{Var}[Z]}.\\
&= \mathrm{Var}[Y] - \frac{2\lambda^2(m_1+m_2)^2}{N(m_1^2+m_2^2+2\lambda^2)}
\end{aligned}
\tag{52}
$$

Equation (52) completes a proof of the theorem. □

In the following, we give a concentration analysis on Theorem 6.

Corollary 7 *Let $\hat{Y}_{CV}$ be the control variate estimate stated in Theorem 6, then for any $\epsilon > 0$ and $\Delta > 0$ the following holds*

$$\Pr\left[|\hat{Y}_{CV} - \lambda| \geq \epsilon\right] \leq \Delta, \qquad \text{for } N > \frac{2\lambda^4 - \lambda^2(m_1+m_2)^2 + m_1m_2(m_1^2+m_2^2)}{\Delta(m_1^2+m_2^2+2\lambda^2)\epsilon^2}. \tag{53}$$

Proof We apply Chebyshev inquisitive on our control variate estimator $\hat{Y}_{CV}$ as follows:

$$\begin{aligned}
\Pr\left[|\hat{Y}_{CV} - \lambda| \geq \epsilon\right] &\leq \frac{\text{Var}[\hat{Y}_{CV}]}{\epsilon^2}. \\
&= \frac{\text{Var}[Y] - \frac{2\lambda^2(m_1+m_2)^2}{N(m_1^2+m_2^2+2\lambda^2)}}{\epsilon^2}. \\
&= \frac{\frac{1}{N}\left(m_1m_2 + \lambda^2 - 2\sum_{i=1}^{D} a_i^2b_i^2\right) - \frac{2\lambda^2(m_1+m_2)^2}{N(m_1^2+m_2^2+2\lambda^2)}}{\epsilon^2}. \\
&= \frac{m_1m_2 + \lambda^2 - 2\sum_{i=1}^{D} a_i^2b_i^2 - \frac{2\lambda^2(m_1+m_2)^2}{(m_1^2+m_2^2+2\lambda^2)}}{N\epsilon^2}.
\end{aligned} \tag{54}$$

$$\begin{aligned}
&\leq \frac{2\lambda^4 - \lambda^2(m_1+m_2)^2 + m_1m_2(m_1^2+m_2^2)}{N\epsilon^2(m_1^2+m_2^2+2\lambda^2)}. \qquad \text{(upon simplification)} \\
&\leq \Delta \quad \text{(if we choose } N > \frac{2\lambda^4 - \lambda^2(m_1+m_2)^2 + m_1m_2(m_1^2+m_2^2)}{\Delta(m_1^2+m_2^2+2\lambda^2)\epsilon^2} \\
&\quad \text{in above expression)} .
\end{aligned} \tag{55}$$

Equation (55) completes a proof. □

3.2 Variance reduction using MLE method

In the following, we present our result of applying the MLE method to reduce variance in the inner product estimation.

Theorem 8 *Let $Y := \langle \boldsymbol{\alpha}, \boldsymbol{\beta} \rangle$ be the random variable denoting the estimate of $\langle \mathbf{a}, \mathbf{b} \rangle$ in the feature hashing (Weinberger et al., 2009) (see Algorithm 1, Theorem 3). Then the maximum likelihood estimator (MLE) of the inner product is the solution of the following cubic equation*

$$\lambda^3 - \lambda^2\left(\boldsymbol{\alpha}\boldsymbol{\beta}^T\right) + \lambda\left(-m_1m_2 + m_1||\boldsymbol{\beta}||^2 + m_2||\boldsymbol{\alpha}||^2\right) - m_1m_2\left(\boldsymbol{\alpha}\boldsymbol{\beta}^T\right) = 0. \tag{56}$$

Denoted by $\hat{Y}_{MLE}$, the asymptotic variance of this estimator is

$$\text{Var}[\hat{Y}_{MLE}] = \frac{(m_1m_2 - \lambda^2)^2}{N(m_1m_2 + \lambda^2)}. \tag{57}$$

Proof Due to Theorem 2, we have

$$\mathbf{V_D}^{-\frac{1}{2}} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} \xrightarrow{D} \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

where $\mathbf{V_D} = \begin{bmatrix} \Sigma_\alpha & \Sigma_{\alpha\beta} \\ \Sigma_{\alpha\beta} & \Sigma_\beta \end{bmatrix}$; $\mathbf{0}$ is $(2N \times 1)$ zero vector and $\mathbf{I}$ is $(2N \times 2N)$ identity matrix. That is the joint distribution of $[\boldsymbol{\alpha}, \boldsymbol{\beta}]$ follows multivariate normal distribution under the convergence of distribution as $D \to \infty$.[2] The joint probability density function of multivariate normal distribution of $[\boldsymbol{\alpha}, \boldsymbol{\beta}]$ is given by

$$\text{lik}([\boldsymbol{\alpha}, \boldsymbol{\beta}]) \propto |\mathbf{V}_D|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} \begin{bmatrix} \boldsymbol{\alpha} & \boldsymbol{\beta} \end{bmatrix} \mathbf{V}_D^{-1} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix}\right). \tag{58}$$

$$\text{Where } |\mathbf{V}_D| = \frac{(m_1 m_2 - \lambda^2)^N}{N^{2N}}, \tag{59}$$

$$|\mathbf{V}_D|^{-\frac{1}{2}} = \frac{N^N}{(m_1 m_2 - \lambda^2)^{\frac{N}{2}}}, \qquad \text{and} \tag{60}$$

$$\mathbf{V}_D^{-1} = \frac{N}{(m_1 m_2 - \lambda^2)} \begin{bmatrix} m_2 & \dots & 0 & -\lambda & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & m_2 & 0 & \dots & -\lambda \\ -\lambda & \dots & 0 & m_1 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & -\lambda & 0 & \dots & m_1 \end{bmatrix}. \tag{61}$$

Equations (59) and (61) holds by computing the determinant and inverse of the matrix $\mathbf{V}_D$. We now calculate the term

$$\begin{bmatrix} \boldsymbol{\alpha} & \boldsymbol{\beta} \end{bmatrix} \mathbf{V}_D^{-1} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} = \frac{N}{(m_1 m_2 - \lambda^2)} \left(\sum_{i=1}^{N} \left(\alpha_i^2 m_2 + \beta_i^2 m_1 - 2\alpha_i \beta_i \lambda \right) \right). \tag{62}$$

The above equality holds by putting the value of $\mathbf{V}_D^{-1}$ from Eq. (61) followed by some algebraic calculation. Equations (60), (62) along with Eq. (58) give us the following:

$$\begin{aligned} \text{lik}([\boldsymbol{\alpha}, \boldsymbol{\beta}]) \propto & \frac{N^N}{(m_1 m_2 - \lambda^2)^{\frac{N}{2}}} \\ & \times \exp\left(-\frac{N}{2(m_1 m_2 - \lambda^2)} \sum_{i=1}^{N} \left(\alpha_i^2 m_2 + \beta_i^2 m_1 - 2\alpha_i \beta_i \lambda \right) \right). \end{aligned}$$

Taking logarithm of the above equation gives us

$$\begin{aligned} l(\lambda) = & N \log(N) - \frac{N}{2} \log(m_1 m_2 - \lambda^2) \\ & - \frac{N}{2(m_1 m2 - \lambda^2)} \sum_{i=1}^{N} \left(\alpha_i^2 m_2 + \beta_i^2 m_1 - 2\alpha_i \beta_i \lambda \right). \end{aligned}$$

[2] A proof of Theorem 8 follows using similar analysis as of the proof of Lemma 7 of Li et al. (2006) or Lemma 2 of Li et al. (2006).

By setting $l'(\lambda)$ equal to zero we get $\hat{Y}_{MLE}$, which is solution to equation

$$\lambda^3 - \lambda^2\left(\boldsymbol{\alpha}\boldsymbol{\beta}^T\right) + \lambda\left(-m_1m_2 + m_1||\boldsymbol{\beta}||^2 + m_2||\boldsymbol{\alpha}||^2\right) - m_1m_2\left(\boldsymbol{\alpha}\boldsymbol{\beta}^T\right) = 0. \tag{63}$$

Finally, $\text{Var}[\lambda]$ is given as follows (due to the well known results of large sample theory, see Li et al. 2006)

$$\text{Var}[\lambda] = -\frac{1}{\mathbb{E}[l''(\lambda)]} = \frac{(m_1m_2 - \lambda^2)^2}{N(m_1m_2 + \lambda^2)}. \tag{64}$$

Equations (64) and (63) completes a proof of the theorem. □

Corollary 9 *Let $\hat{Y}_{MLE}$ be the inner product estimator of the MLE method stated in Theorem 8. Then for any $\epsilon > 0$ and $\Delta > 0$, the following holds*

$$\Pr\left[|\hat{Y}_{MLE} - \lambda| \geq \epsilon\right] \leq \Delta, \quad \text{for } N > \frac{(m_1m_2 - \lambda^2)^2}{\Delta(m_1m_2 + \lambda^2)\epsilon^2}. \tag{65}$$

Proof The variance of $\hat{Y}_{MLE}$ (from Theorem 8) is

$$\text{Var}[\hat{Y}_{MLE}] = \frac{(m_1m_2 - \lambda^2)^2}{N(m_1m_2 + \lambda^2)}. \tag{66}$$

From Chevyshev's inequality, we have

$$\begin{aligned}\Pr\left[|\hat{Y}_{MLE} - \lambda| \geq \epsilon\right] &\leq \frac{\text{Var}[\hat{Y}_{MLE}]}{\epsilon^2}.\\ &= \frac{(m_1m_2 - \lambda^2)^2}{\epsilon^2 N(m_1m_2 + \lambda^2)}.\\ &\leq \Delta \qquad \left(\text{if we choose } N > \frac{(m_1m_2 - \lambda^2)^2}{\Delta(m_1m_2 + \lambda^2)\epsilon^2} \text{ in above expression}\right).\end{aligned} \tag{67}$$

Equation (67) completes a proof. □

The MLE method typically introduces some bias and requires a bias correction. However, our result shows that the bias introduced due to the MLE approach is negligible and of the order $O(1/N^2)$. Further, we show that the probability of attaining multiple real roots in cubic polynomials (stated in Eq. (56)) is very small. It ensures that the only real root corresponds to the pairwise inner product. We summarise it in the following Lemma, whose proof follows exactly from the Lemma 3 and 4 of Li et al. (2006).

Lemma 10 (Adapted from Lemma 3 and 4 of Li et al. 2006) *The bias correction for the maximum likelihood estimator, $\hat{Y}_{MLE}$, derived in Theorem 8 is given by*

$$\mathbb{E}[\hat{Y}_{MLE}] = \langle \mathbf{a}, \mathbf{b} \rangle + O\left(\frac{1}{N^2}\right).$$

Further, the cubic equation stated in Eq. (56) (Theorem 8) admits multiple real roots with a very small probability, given by

$$\Pr[\text{multiple real roots}] = \Pr\left(P^2(11 - Q^2/4 - 4Q + P^2) + (Q-1)^3 \leq 0\right),$$

where $P = \frac{\langle \boldsymbol{\alpha}, \boldsymbol{\beta} \rangle}{\sqrt{m_1 m_2}}$, $Q = \frac{||\boldsymbol{\alpha}||^2}{m_1} + \frac{||\boldsymbol{\beta}||^2}{m_2}$. *This probability is crudely bounded by*

$$\Pr[\text{multiple real roots}] \leq e^{-0.0085N} + e^{-0.0966N}.$$

Remark 2 Our both results stated in Theorems 6, 8 crucially build on the result of Theorem 4 which shows that the vector $\mathbf{w} = \left[\boldsymbol{\alpha}, \boldsymbol{\beta}\right]^T$ is multivariate normal. Note that Theorem 4 holds for small values of $N = o\left(D^{\frac{\delta}{2(\delta+2)}}\right)$ such that $\mathbb{E}\left[|a_i|^{2+\delta}\right]$ and $\mathbb{E}\left[|b_i|^{2+\delta}\right]$ are finite for any $\delta > 0$. Note that for smaller values of N the variance of the feature hashing estimator is large (see Theorem 3), and our results (Theorems 6, 8) mitigate this problem by showing significant variance reduction on such instance.

Assumptions and overhead of our estimators: It's worth mentioning that both these results require the norm of the data points to compute their respective similarity estimates. The overhead of our control variate estimator (CV-FH, Theorem 6) is that it requires computing the expected value of the control variate random variable Z (see Eq. (45)), and control variate coefficient $\hat{c}$ (see Eq. (49)). Further, computing $\hat{c}$ requires knowing the value of λ—the very quantity which we want to estimate. Empirically, we use the estimate obtained from Vanilla-FH as a proxy for this. The overhead of our MLE estimator (MLE-FH, Theorem 8) is that it requires computing roots of the cubic polynomial stated in Eq. (56) to get an estimate of the inner product.

4 Experiments

Hardware description: CPU: Intel(R) Core(TM) i7-8750H CPU @ 2.21GHz x 6; Memory: 16 GB; OS: Windows 10; Model: Lenovo Legion Y530.

Datasets: We use the following datasets for our experiments:

- **Synthetic dataset:** In synthetic dataset, we generate 2000 random data points in 20000 dimension such that each feature of the data point is randomly chosen from [1, 10].
- **PEMS-SF** (Lichman 2013): This dataset contains the daily occupancy rates of various car lanes on the San Francisco Bay Freeway from the California Department of Transportation PEMS website over a 15-month period. These data range from [0, 1], with 440 data samples of size 138672. The link of the dataset is available here[3].
- **UCI Bag-of-word datasets—NYTimes articles** (Lichman 2013): This dataset consists of a corpus of documents that include 300000 points in 102660 dimension. The raw documents were pre-processed by tokenization and removal of stopwords. A vocabulary of unique words was generated (by keeping only those words that occurred more than ten times). The size of the vocabulary determines the dimension of the dataset. A

[3] https://archive.ics.uci.edu/ml/datasets/PEMS-SF

document is represented by the frequency vector of the words in it. The link of the dataset is available here[4].

– **Gisette dataset** (Guyon et al. 2005; Lichman 2013): This dataset consists of handwritten digit images and is constructed from the MNIST dataset. This dataset consists of 13, 500 real-valued vectors in 5000 dimension. The digits have been size-normalized and centered in a fixed-size image of dimension 28×28. From the images, pixels were sampled at random so that it contains the necessary information to disambiguate between the digit 4 from 9. Further higher-order features were created to project the problem into a higher dimensional feature space. The link of the dataset is available here. [5]

4.1 Methodology

Let $Y = \langle \boldsymbol{\alpha}, \boldsymbol{\beta} \rangle$ be the random variable denoting the estimate of inner product obtained via the feature hashing. In these experiments, we aim to show that the variance of our control variate estimate (CV-FH) and MLE estimate (MLE-FH) is smaller than that of the vanilla feature hashing estimate. In what follows, we describe the procedure to empirically compute our estimates (CV-FH and MLE-FH). In Sect. 4.2, we discuss the evaluation metric, and finally in Sect. 4.3, we discuss our experimental insights. Our estimates require computing the ℓ_2 norm of the data points that can be easily computed by taking a pass over the dataset.

Computing control variate estimate (CV-FH) : Recall that our control variate estimate is given by $Y + c(Z - \mathbb{E}[Z])$, where c is the control variate coefficient, and Z is the control variate random variable. From Eq. (49) the optimum value of c, denoted by $\hat{c}$, can be described as follows:

$$\hat{c} = -\frac{\text{Cov}[Y, Z]}{\text{Var}[Z]} = -\frac{\lambda(m_1 + m_2)}{(m_1^2 + m_2^2 + 2\lambda^2)}.$$

Recall that our control variate random variable is $Z = ||\boldsymbol{\alpha}||^2 + ||\boldsymbol{\beta}||^2$ (see Eq. (42)), and $\mathbb{E}[Z] = m_1 + m_2$. To compute the value of $\hat{c}$, we need to know the value of λ—the very quantity we want to estimate. We use the estimate of the inner product obtained using the feature hashing as a proxy for λ. As the sketch and norm of data points are known to us, we can compute the terms $Y, Z, \mathbb{E}[Z]$ and $\hat{c}$, and as a consequence, we can compute the CV estimate $Y + \hat{c}(Z - \mathbb{E}[Z])$.

Computing MLE estimate (MLE-FH): Recall that from Eq. (56), our MLE estimator is the real root of the following cubic polynomial

$$\lambda^3 - \lambda^2\left(\boldsymbol{\alpha}\boldsymbol{\beta}^T\right) + \lambda\left(-m_1 m_2 + m_1||\boldsymbol{\beta}||^2 + m_2||\boldsymbol{\alpha}||^2\right) - m_1 m_2\left(\boldsymbol{\alpha}\boldsymbol{\beta}^T\right) = 0.$$

As we know the sketch $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and the norm of the data points, we can generate the above polynomial. We show that the polynomial has only one real root (see Lemma 10), and we can get the estimation of the inner product by computing the real root of this cubic polynomial. To compute the real root, we use the following expression deduced from Cardano's formula (Cardano 1993).

[4] https://archive.ics.uci.edu/ml/datasets/Bag+of+Words

[5] https://archive.ics.uci.edu/ml/datasets/Gisette

$$\lambda = -\frac{1}{3}\left(p + C + \frac{d_0}{C}\right), \text{ where}$$

$$C = \left(\frac{d_1 \pm \sqrt{d_1^2 - 4d_0^3}}{2}\right)^{\frac{1}{3}},$$

$$d_0 = p^2 - 3q,$$

$$d_1 = 2p^3 - 9pq + 27r,$$

$$p = -\boldsymbol{\alpha}\boldsymbol{\beta}^T, q = -m_1 m_2 + m_1||\boldsymbol{\beta}||^2 + m_2||\boldsymbol{\alpha}||^2, \text{ and } , r = -m_1 m_2\left(\boldsymbol{\alpha}\boldsymbol{\beta}^T\right).$$

4.2 Evaluation metric

We evaluate the performance of our CV estimate (CV-FH) and MLE estimate (MLE-FH) with the vanilla feature hashing (Vanilla-FH) on the following metrics (Table 1).

- Variance analysis *via* box-plot,
- Mean absolute error (MAE) for a pair of points,
- Mean absolute error (MAE) for a large number of points, and
- the running time.

We elaborate our experimental procedure as follows:

Variance analysis via box-plot: In this experiment, we aim to compare the variance of all the three estimates—CV feature hashing (CV-FH), MLE feature hashing (MLE-FH), and baseline feature hashing (Vanilla-FH). We perform this experiment on both synthetic pairs and pairs sampled from the real-world datasets. For synthetic pairs, we generate several pairs of real-valued vectors in 10000 dimension such that the angle between them is $\theta = \{10^o, 30^o, 60^o, 90^o\}$. For pairs from real-world datasets, we pick a random pair of points for each dataset mentioned in Table 2. We run CV-FH, MLE-FH, and feature hashing (Vanilla-FH) on these pairs, 100 times for different values of reduced dimension N. It gives us 100 different estimates for each of the baseline algorithms. We use these estimates to generate box plots for variance analysis. We summarize our observations in Fig. 3 for synthetic pairs, and in Fig. 4 for pairs sampled from the real-world datasets.

MAE for a pair of points: In this experiment, for a pair of data points, we aim to compare the mean absolute error of all three estimates with respect to the ground similarity. We perform this for synthetic pairs as well as the pairs sampled from the real-world datasets mentioned above. We run CV-FH, MLE-FH, and feature hashing 100 times on these input pairs for various values of reduced dimension N. It gives us 100 different estimates for each of the baseline algorithms on various values of N. We use these estimates to calculate the mean absolute error (MAE)—by computing the average of the absolute difference of estimates obtained from the baselines with ground truth similarity—for different values of reduced dimension. We summarize our observations in Fig. 5 for synthetic pairs, and in Fig. 6 for pairs sampled from the real-world datasets.

MAE for a large number of pairs: In this experiment, we aim to compare the MAE of all three estimates with respect to the ground similarity for all pairs of points. To do so, we take a random sample of 2000 data points (except the PEMS-SF dataset, which has a lesser

Table 1 Notations

Notations	
D	dimension of the input data.
N	dimension of the compressed data.
M	number of data points.
$\mathbf{a}$	$[a_1, a_2, \dots a_D]$ input vector.
$\mathbf{b}$	$[b_1, b_2, \dots b_D]$ input vector.
$\boldsymbol{\alpha}$	$[\alpha_1, ..\alpha_k, .. \dots \alpha_N]$ compressed vector of $\mathbf{a}$.
$\boldsymbol{\beta}$	$[\beta_1, .., \beta_k, .. \dots \beta_N]$ compressed vector of $\mathbf{b}$.
$[\boldsymbol{\alpha}, \boldsymbol{\beta}]$	$[\alpha_1, \dots, \alpha_N, \beta_1, \dots, \beta_N]$
$\begin{bmatrix}\boldsymbol{\alpha}\\ \boldsymbol{\beta}\end{bmatrix}$	$[\boldsymbol{\alpha}, \boldsymbol{\beta}]^T$
m_1	$\|\|\mathbf{a}\|\|_2^2 = \sum_{i=1}^{D} a_i^2$ squared norm of $\mathbf{a}$.
m_2	$\|\|\mathbf{b}\|\|_2^2 = \sum_{i=1}^{D} b_i^2$ squared norm of $\mathbf{b}$.
λ	$\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{i=1}^{D} a_i b_i$ dot product of $\mathbf{a}$ and $\mathbf{b}$.
θ	angle between $\mathbf{a}$ and $\mathbf{b}$.

Table 2 Dataset description. (Sparsity denotes the maximum number of non-zero entries in any data point)

Dataset	Attributes	Dimension	Sparsity
Synthetic Dataset	Real	20000	20000
PEMS-SF (Lichman 2013)	Real	138672	138655
NYTimes articles (Lichman 2013)	Integer	102660	871
Gisette (Lichman 2013)	Integer	5000	1409

number of points) from each of the datasets. We repeat the above experiment (MAE for a pair) 10 times for every pair of points (out of the possible $\binom{2000}{2} = 19,99,000$ pairs). We compute the MAE - by computing the mean (overall the iteration and all possible pairs) absolute difference of estimates obtained from the baselines with ground truth similarity. We summarise our results in Fig. 7. We note that if we include all points from the datasets, then we are getting *"out-of-memory"* error due to the very large number of pairs generated. Therefore, we decided to include a random sample of 2000 points in our experiments. The number of pairs generated by 2000 points is 19, 99, 000, which is large enough to cover a wide spectrum of pairwise similarity. We also note the average time (over 10 repetitions) taken by each method for this experiment to compare their respective time complexity. We summarise it in Fig. 8.

4.3 Insight

From Figs. 3, 4 it is evident that the interquartile range of our proposed techniques CV-FH and MLE-FH is smaller than that of the Vanilla-FH. In particular, in Fig. 3, we notice that our proposals have a smaller interquartile range when the pairwise angle is small. This indicates that the variances of the proposed techniques are much smaller than the variance of the Vanilla-FH, especially when pairwise angles are small. In Figs. 5, 6, we notice that the MAEs of the proposed techniques are always smaller than that of Vanilla-FH. Again, due

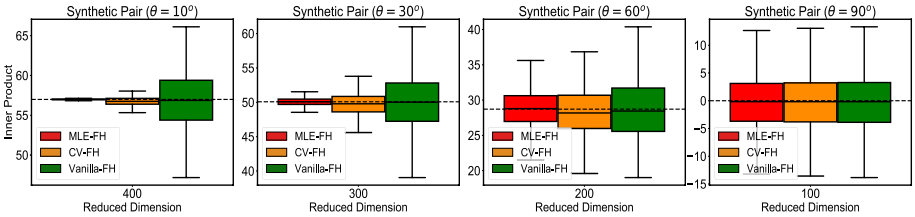

Fig. 3 Comparison among vanilla feature hashing (Vanilla-FH) and our proposed estimates based on CV (CV-FH), and MLE (MLE-FH) on the variance analysis *via* box-plot for synthetic pairs. Smaller interquartile range is an indication of better performance. The dotted line corresponds to the ground truth inner product

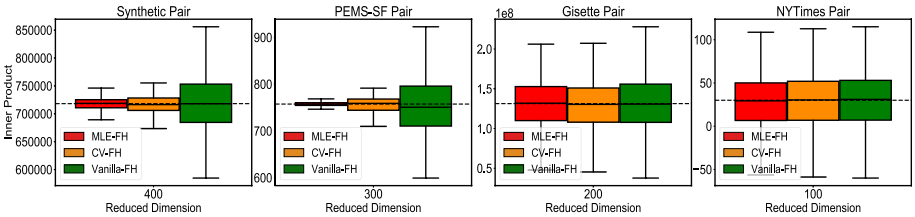

Fig. 4 Comparison among vanilla feature hashing (Vanilla-FH) and our proposed estimates based on CV (CV-FH), and MLE (MLE-FH) on the variance analysis *via* box-plot for pairs sampled from real world datasets. Smaller interquartile range is an indication of better performance. The dotted line corresponds to the ground truth inner product

to Fig. 5, we observe that our proposals have smaller MAE values when the pairwise angle between input pairs is small. This indicates that the errors in our estimates are smaller, and they closely approximate the ground truth inner product, especially when pairwise angles are small. Further, in Figs. 7, we again observe that MAEs of our proposals are smaller than that of Vanilla-FH. These observations indicate that, on average, our proposals correctly estimate the ground truth pairwise similarity. The MAE values of our estimators are comparable to that of Vanilla-FH for the NYTimes dataset. This is because a large number of input pairs are almost orthogonal to each other. We computed some statistics on the pairwise angle for the NYTimes dataset and observed that mean angle = 87.3^o, median angle = 88.2^o. This is also consistent with our observations on the theoretical variance plot (Fig. 1), where we notice that our proposal CV-FH and MLE-FH doesn't give any advantage over Vanilla-FH on orthogonal pairs. We note that the MLE estimates generally offer better performance (lower variance and MAE) as compared to the CV estimate. However, this comes at the cost of the slower computational time of the MLE estimate, possibly due to the computation involved in computing the roots of the cubic polynomials. We summarise the average speedup (mean of the speedup obtained at various reduced dimensions) of CV-FH *w.r.t.* MLE-FH in Table 3 and notice that the CV-FH is roughly 2.5× faster than the MLE-FH on our datasets.

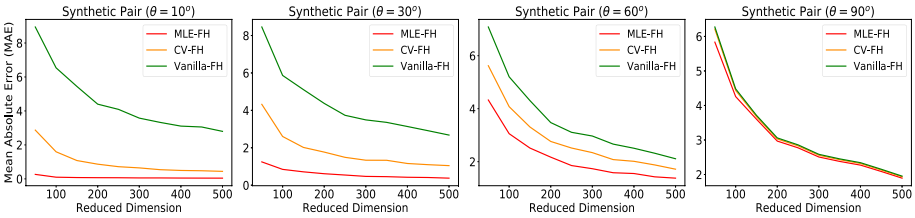

Fig. 5 Comparison among Vanilla-FH and our proposed estimates CV-FH and MLE-FH on the mean-absolute-error (MAE) metric for a synthetic data pair. A smaller value of MAE is an indication of better performance

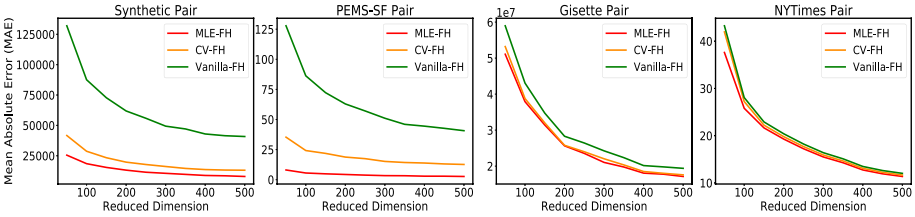

Fig. 6 Comparison among Vanilla-FH and our proposed estimates CV-FH and MLE-FH on the mean-absolute-error (MAE) metric for a pair of points sampled from the real world datasets. A smaller value of MAE is an indication of better performance

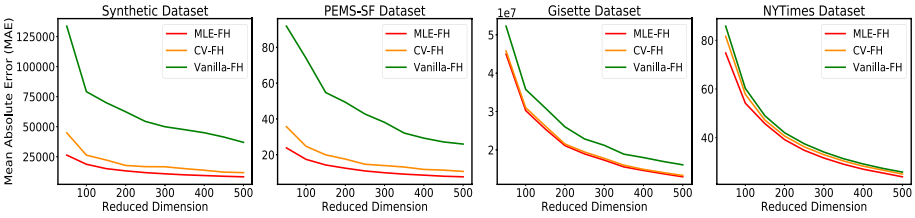

Fig. 7 Comparison among Vanilla-FH and our proposed estimates CV-FH and MLE-FH on the mean-absolute-error (MAE) metric for $\binom{2000}{2}$ pairs from real-world datasets. A smaller value of MAE is an indication of better performance

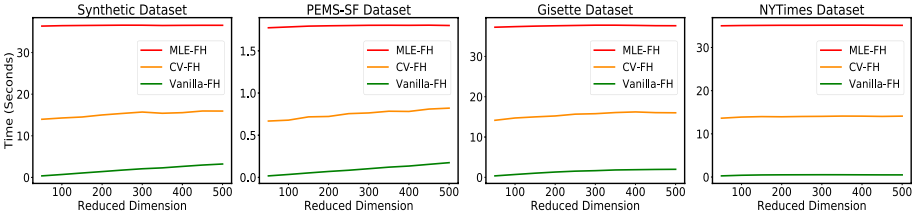

Fig. 8 Comparison on the average running time among Vanilla-FH, CV-FH, and MLE-FH

Table 3 Average speedup of CV-FH *w.r.t.* MLE-FH, which is the mean of speedup of CV-FH *w.r.t.* MLE-FH obtained at various reduced dimensions

Dataset	Synthetic Dataset	PEMS-SF	Gisette	NYTimes articles
Avg. Speedup	2.41×	2.40×	2.43×	2.51×

5 Conclusion

We provide a simple and effective improvement to the feature hashing algorithm, a popular dimensionality reduction technique. Our improvement is to achieve the variance reduction in the inner product estimate obtained from the feature hashing algorithm. We propose variance reduction via two approaches—control variate (CV) and maximum likelihood estimator (MLE) methods. We present a clean theoretical analysis of our approaches and complement it with rigorous experiments on synthetic as well as real-world datasets. We observed (theoretically as well as empirically) that both the methods offer significant variance reduction, especially when data points are highly similar. In comparison between these two methods, the MLE tends to offer a higher variance reduction but at the cost of higher running time (see Table 3). Our proposals (CV-FH and MLE-FH) are simple, effective, and can be easily adopted in practice.

Appendix—missing proofs

Proof of Theorem 3:

Proof We can write the k-th feature of the sketch vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ as follows:

$$\alpha_k = \sum_{i=1}^{D} a_i x_i z_i^{(k)}. \tag{68}$$

$$\beta_k = \sum_{i=1}^{D} b_i x_i z_i^{(k)}. \tag{69}$$

Where x_i takes value between $\{+1, -1\}$ each with probability 1/2 and $z_i^{(k)}$ is an indicator random variable defined as follows:

$$z_i^{(k)} = \begin{cases} 1, & \text{if } i\text{-th feature falls into the } k\text{-th bucket} \\ 0, & \text{otherwise.} \end{cases} \tag{70}$$

We now compute the expected value of $\boldsymbol{\alpha}$. In order to compute this, first we compute expected value of α_k

$$\begin{aligned}\mathbb{E}[\alpha_k] &= \mathbb{E}\left[\sum_{i=1}^{D} a_i x_i z_i^{(k)}\right].\\ &= \sum_{i=1}^{D} a_i \mathbb{E}\left[x_i z_i^{(k)}\right].\\ &= \sum_{i=1}^{D} a_i \mathbb{E}[x_i]\mathbb{E}[z_i^{(k)}].\\ &= \sum_{i=1}^{D} a_i \times 0 \times \mathbb{E}[z_i^{(k)}].\end{aligned} \tag{71}$$

$$\begin{aligned}&= 0. \implies \mathbb{E}[\boldsymbol{\alpha}] &&= \mathbb{E}[(\alpha_1, \alpha_2, ..., \alpha_N)].\\ &= (\mathbb{E}[\alpha_1], \mathbb{E}[\alpha_2], ..., \mathbb{E}[\alpha_N]).\\ &= (0, 0, ..., 0).\\ &= \mathbf{0}.\end{aligned} \tag{72}$$

Equation (71) holds because $\mathbb{E}[x_i] = 0$. This is true because the random variable x_i takes value between $\{-1, +1\}$ each with probability 1/2. Similarly, we can compute the value of $\mathbb{E}[\boldsymbol{\beta}]$. We now compute the inner product of the compressed vectors $\langle \boldsymbol{\alpha}, \boldsymbol{\beta} \rangle$ as follows:

$$\begin{aligned}\langle \boldsymbol{\alpha}, \boldsymbol{\beta} \rangle &= \sum_{k=1}^{N} \alpha_k \beta_k = \sum_{k=1}^{N} \left(\sum_{i=1}^{D} a_i x_i z_i^{(k)}\right)\left(\sum_{i=1}^{D} b_i x_i z_i^{(k)}\right).\\ &= \sum_{k=1}^{N} \left(\sum_{i=1}^{D} a_i b_i x_i^2 {z_i^{(k)}}^2 + \sum_{i \neq j} a_i b_j x_i x_j z_i^{(k)} z_j^{(k)}\right).\end{aligned} \tag{73}$$

$$\begin{aligned}&= \sum_{k=1}^{N} \left(\sum_{i=1}^{D} a_i b_i z_i^{(k)} + \sum_{i \neq j} a_i b_j x_i x_j z_i^{(k)} z_j^{(k)}\right).\\ &= \sum_{i=1}^{D} a_i b_i \sum_{k=1}^{N} z_i^{(k)} + \sum_{k=1}^{N} \left(\sum_{i \neq j} a_i b_j x_i x_j z_i^{(k)} z_j^{(k)}\right).\end{aligned} \tag{74}$$

$$\begin{aligned}&= \sum_{i=1}^{D} a_i b_i + \sum_{k=1}^{N} \left(\sum_{i \neq j} a_i b_j x_i x_j z_i^{(k)} z_j^{(k)}\right).\\ &= \langle \mathbf{a}, \mathbf{b} \rangle + \sum_{k=1}^{N} \left(\sum_{i \neq j} a_i b_j x_i x_j z_i^{(k)} z_j^{(k)}\right).\end{aligned} \tag{75}$$

Equation (74) follows from Eq. (73) because $x_i^2 = 1$ as $x_i = \pm 1$, and $z_i^2 = z_i$ as z_i takes value either 1 or 0. We continue from Eq. (75) and compute the expectation of the random variable $\langle \boldsymbol{\alpha}, \boldsymbol{\beta} \rangle$ as follows:

$$\begin{aligned}\mathbb{E}[\langle \boldsymbol{\alpha}, \boldsymbol{\beta}\rangle] &= \mathbb{E}\left[\langle \mathbf{a}, \mathbf{b}\rangle + \sum_{k=1}^{N}\left(\sum_{i\neq j} a_i b_j x_i x_j z_i^{(k)} z_j^{(k)}\right)\right].\\ &= \mathbb{E}[\langle \mathbf{a}, \mathbf{b}\rangle] + \mathbb{E}\left[\sum_{k=1}^{N}\left(\sum_{i\neq j} a_i b_j x_i x_j z_i^{(k)} z_j^{(k)}\right)\right]. \qquad (76)\\ &= \langle \mathbf{a}, \mathbf{b}\rangle + \sum_{k=1}^{N}\left(\sum_{i\neq j} \mathbb{E}\left[a_i b_j x_i x_j z_i^{(k)} z_j^{(k)}\right]\right).\end{aligned}$$

$$\begin{aligned}&= \langle \mathbf{a}, \mathbf{b}\rangle + \sum_{k=1}^{N}\left(\sum_{i\neq j} a_i b_j \mathbb{E}\left[x_i x_j z_i^{(k)} z_j^{(k)}\right]\right). \qquad (77)\\ &= \langle \mathbf{a}, \mathbf{b}\rangle.\end{aligned}$$

Equation (76) holds due to the linearity of expectation. Equation (77) holds because $\mathbb{E}[x_i x_j z_i^{(k)} z_j^{(k)}] = 0$ as both x_i and x_j take a value between $\{-1, +1\}$ each with probability 1/2 which leads to $\mathbb{E}[x_i x_j] = 0$. Now, we will compute the expected value of the norm square of compressed vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$

$$\begin{aligned}\mathbb{E}\left[||\boldsymbol{\alpha}||^2\right] &= \mathbb{E}\left[\sum_{i=1}^{N} \alpha_i^2\right].\\ &= \sum_{i=1}^{N} \mathbb{E}[\alpha_i^2]. \qquad (78)\\ &= \sum_{i=1}^{N} \frac{||\mathbf{a}||^2}{N}.\end{aligned}$$

$$= ||\mathbf{a}||^2. \qquad (79)$$

Equation (78) holds due to the following

$$\begin{aligned}\mathbb{E}\left[\alpha_k^2\right] &= \mathbb{E}\left[\left(\sum_{i=1}^{D} a_i x_i z_i^{(k)}\right)^2\right].\\ &= \mathbb{E}\left[\sum_{i=1}^{D} a_i^2 x_i^2 z_i^{(k)^2} + \sum_{i\neq j}\left(a_i a_j x_i x_j z_i^{(k)} z_j^{(k)}\right)\right].\\ &= \mathbb{E}\left[\sum_{i=1}^{D} a_i^2 x_i^2 z_i^{(k)^2}\right] + \mathbb{E}\left[\sum_{i\neq j}\left(a_i a_j x_i x_j z_i^{(k)} z_j^{(k)}\right)\right]. \qquad (80)\\ &= \sum_{i=1}^{D} a_i^2 \mathbb{E}\left[z_i^{(k)}\right] + \sum_{i\neq j}\left(a_i a_j \mathbb{E}\left[x_i x_j z_i^{(k)} z_j^{(k)}\right]\right).\\ &= \sum_{i}^{D} a_i^2 \times \frac{1}{N} + \sum_{i\neq j} a_i a_j \times 0.\\ &= \frac{||\mathbf{a}||^2}{N}.\end{aligned}$$

In similar fashion we can calculate

$$\mathbb{E}\left[||\boldsymbol{\beta}||^2\right] = ||\mathbf{b}||^2. \tag{81}$$

Now we will compute the expectation of $||\boldsymbol{\alpha}||^2 + ||\boldsymbol{\beta}||^2$

$$\begin{aligned}\mathbb{E}\left[||\boldsymbol{\alpha}||^2 + ||\boldsymbol{\beta}||^2\right] &= \mathbb{E}\left[||\boldsymbol{\alpha}||^2\right] + \mathbb{E}\left[||\boldsymbol{\beta}||^2\right]. \\ &= ||\mathbf{a}||^2 + ||\mathbf{b}||^2.\end{aligned} \tag{82}$$

Equation (82) holds because of Eq. (79) and (81). Using similar analysis techniques we compute the $\text{Var}[\langle \boldsymbol{\alpha}, \boldsymbol{\beta} \rangle]$ which turn out to be the following due to Weinberger et al. (2009).

$$\text{Var}[\langle \boldsymbol{\alpha}, \boldsymbol{\beta} \rangle] = \frac{1}{N} \sum_{i \neq j, i,j=1}^{D} \left(a_i^2 b_j^2 + a_i b_i a_j b_j \right).$$

□

Author Contributions BDV Methodology, Formal analysis, Experimentation, Validation, Writing—original draft, Writing— review & editing. RP Methodology, Formal analysis, Validation, Writing—original draft, Writing—review & editing. MT Methodology, Formal analysis, Writing– review & editing.

Funding Not applicable.

Data availibility statement The datasets used in this study are publicly available.

Declarations

Conflict of interest The authors declare that there is no conflict of interest.

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication All authors participated in this study give the publisher the permission to publish this work.

Code availability Please contact the authors for code.

References

Achlioptas, Dimitris. (2003). Database-friendly random projections: Johnson-lindenstrauss with binary coins. *The Journal of Computer and System Sciences, 66*(4), 671–687.

Agarwal, A, Chapelle, O., Dudík, M., & Langford, J. (2014). *A reliable effective terascale linear learning system., 15*, 1111–1133.

Bellman, Richard. (1966). Dynamic programming. *Science, 153*(3731), 34–37.

Broder, A.Z., Charikar, M., Frieze, A.M., & Mitzenmacher, M. (1998). Min-wise independent permutations (extended abstract). In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pp. 327–336.

Cardano, G. (1993). *Ars magna or the rules of algebra*. Dover Publications.

Charikar, M. (2002). Similarity estimation techniques from rounding algorithms. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pp. 380–388.

Chen, W., Wilson, J., Tyree, S., Weinberger, K., & Chen, Y. (2015). Compressing neural networks with the hashing trick. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2285–2294, Lille, France, 07–09 Jul 2015. PMLR.

Dasgupta, A., Kumar, R., & Sarlós, T., (2010). A sparse johnson: Lindenstrauss transform. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pp. 341–350.

Dasgupta, S., & Gupta, A. (2003). An elementary proof of a theorem of johnson and lindenstrauss. *Random Structures and Algorithms, 22*(1), 60–65.

Ertl, O. (2018). Bagminhash-minwise hashing algorithm for weighted sets. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1368–1377.

Feller, W. (1968). *An Introduction to Probability Theory and Its Applications* (Vol. 1). Wiley.

Guyon, I., Gunn, S., Ben-Hur, A., & Dror, G. (2005). Result analysis of the nips 2003 feature selection challenge. In L. K. Saul, Y. Weiss, & L. Bottou (Eds.), *Advances in Neural Information Processing Systems 17* (pp. 545–552). MIT Press.

Indyk, P. (2006). Stable distributions, pseudorandom generators, embeddings, and data stream computation. *Journal of the ACM (JACM), 53*(3), 307–323.

Ioffe, S. (2010). Improved consistent sampling, weighted minhash and l1 sketching. In *2010 IEEE International Conference on Data Mining*, pp. 246–255. IEEE.

Ji, J., Li, J., Yan, S., Zhang, B., & Tian, Q. (2012). Super-bit locality-sensitive hashing. In *Advances in neural information processing systems*, pp. 108–116.

Johnson, W.B., & Lindenstrauss, J. (1983). Extensions of lipschitz mappings into a hilbert space. *Conference in modern analysis and probability (New Haven, Conn., 1982), Amer. Math. Soc., Providence, R.I.*, pages 189–206.

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., & Mikolov, T. (2016). Fasttext.zip: Compressing text classification models. CoRR, arXiv:abs/1612.03651.

Kang, K. (2017). Using the multivariate normal to improve random projections. In *Intelligent Data Engineering and Automated Learning - IDEAL 2017 - 18th International Conference, Guilin, China, October 30 - November 1, 2017, Proceedings*, pp. 397–405.

Kang, K., & Hooker, G. (2017). Random projections with control variates. In *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods, ICPRAM 2017, Porto, Portugal, February 24-26, 2017*, pp. 138–147.

Kang, K., & Pin, W.W. (2018). Improving sign random projections with additional information. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pp. 2484–2492.

Lavenberg, S., & Welch, P. (1981). A perspective on the use of control variables to increase the efficiency of monte carlo simulations. *Management Science, 27,* 322–335.

Li, P. (2007). Very sparse stable random projections for dimension reduction in l α $(0 < \alpha \leq 2)$ norm. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 440–449.

Li, P. (2008). Estimators and tail bounds for dimension reduction in l_α $(0 < \alpha \leq 2)$ using stable random projections. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 10–19.

Li, P. (2019). Sign-full random projections. In *Proceedings of the AAAI Conference on Artificial Intelligence, 33*, 4205–4212.

Li, P., & Hastie, T. (2007). A unified near-optimal estimator for dimension reduction in l_α $(0 < \alpha \leq 2)$ using stable random projections. In *NIPS*, pp. 905–912. Citeseer.

Li, P., Hastie, T., & Church, K.W. (2006). *Practical Procedures for Dimension Reduction in l1*. Citeseer.

Li, P., Hastie, T., & Church, K.W. (2006). Improving random projections using marginal information. In *Learning Theory, 19th Annual Conference on Learning Theory, COLT 2006, Pittsburgh, PA, USA, June 22-25, 2006, Proceedings*, pp. 635–649.

Li, P., Hastie, T., & Church, K.W. (2006). Very sparse random projections. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006*, pp. 287–296.

Li, P., & König, A. C. (2011). Theory and applications of b-bit minwise hashing. *Commun. ACM, 54*(8), 101–109.

Li, P, Owen, A., & Zhang, C.H. (2012). One permutation hashing. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pp. 3122–3130.

Li, X, & Li, P. (2019). Random projections with asymmetric quantization. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pp. 10858–10867.

Li, Yulong, Kuang, Zhihao, Li, Jiang Yan, & Kang, Keegan. (2020). Improving random projections with extra vectors to approximate inner products. *IEEE Access, 8*, 78590–78607.

Lichman, M. (2013). UCI machine learning repository.

Mensink, T., Verbeek, J., Perronnin, F., & Csurka, G., (2012). Metric learning for large scale image classification: Generalizing to new classes at near-zero cost. In *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part II*, pp. 488–501.

Murphy, K.P. (2013). *Machine learning : a probabilistic perspective*. MIT Press, Cambridge, Mass. [u.a.].

Pratap, R., Bera, D., Revanuru, K., (2019). Efficient sketching algorithm for sparse binary data. In Jianyong Wang, Kyuseok Shim, and Xindong Wu, editors, *2019 IEEE International Conference on Data Mining, ICDM 2019, Beijing, China, November 8-11, 2019*, pp. 508–517. IEEE.

Pratap, R., Kulkarni, R., & Sohony, I. (2018). Efficient dimensionality reduction for sparse binary data. In Naoki Abe, Huan Liu, Calton Pu, Xiaohua Hu, Nesreen K. Ahmed, Mu Qiao, Yang Song, Donald Kossmann, Bing Liu, Kisung Lee, Jiliang Tang, Jingrui He, and Jeffrey S. Saltz, editors, *IEEE International Conference on Big Data, Big Data 2018, Seattle, WA, USA, December 10-13, 2018*, pages 152–157. IEEE.

Provost, S.B., & Mathai, A.M. (1992). *Quadratic Forms in Random Variables: Theory and Applications/ A.M. Mathai, Serge B. Provost*. Statistics : textbooks and monographs. Marcel Dek ker.

Shrivastava, A. (2016). Simple and efficient weighted minwise hashing. In *Advances in Neural Information Processing Systems*, pages 1498–1506.

Shrivastava, A. (2017). Optimal densification for fast and accurate minwise hashing. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 3154–3163.

Shrivastava, A., & Li, P. (2014). In defense of minhash over simhash. In *Artificial Intelligence and Statistics*, pages 886–894. PMLR.

Weinberger, K., Dasgupta, A., Langford, J., Smola, A., & Attenberg, J. (2009). Feature hashing for large scale multitask learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14–18, 2009*, pages 1113–1120.

Wei, Wu., Li, Bin, Chen, Ling, Zhang, Chengqi, & Philip, S Yu. (2018). Improved consistent weighted sampling revisited. *IEEE Transactions on Knowledge and Data Engineering, 31*(12), 2332–2345.

Wu, X., Zhu, X., Wu, G., & Ding, W. (2014). Data mining with big data. *IEEE Transactions on Knowledge and Data Engineering, 26*(1), 97–107.

Yu, F., Kumar, S., Gong, Y., & Chang, S.F., (2014). Circulant binary embedding. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–946–II–954. JMLR.org.

Zhai, Y., Ong, Y., & Tsang, I. W. (2014). The emerging "big dimensionality''. *IEEE Computational Intelligence Magazine, 9*(3), 14–26.