Binary embeddings with structured hashed projections

Anna Choromanska[1] ACHOROMA@CIMS.NYU.EDU
Courant Institute of Mathematical Sciences, NY

Krzysztof Choromanski[1] KCHORO@GOOGLE.COM
Google Research, NY

Mariusz Bojarski MBOJARSKI@NVIDIA.COM
NVIDIA, NY

Tony Jebara JEBARA@CS.COLUMBIA.EDU
Columbia University, NY

Sanjiv Kumar SANJIVK@GOOGLE.COM
Google Research, NY

Yann LeCun YANN@CS.NYU.EDU
Courant Institute of Mathematical Sciences, NY and Facebook AI Research

Abstract

We consider the hashing mechanism for constructing binary embeddings, that involves pseudo-random projections followed by nonlinear (sign function) mappings. The pseudo-random projection is described by a matrix, where not all entries are independent random variables but instead a fixed "budget of randomness" is distributed across the matrix. Such matrices can be efficiently stored in sub-quadratic or even linear space, provide reduction in randomness usage (i.e. number of required random values), and very often lead to computational speed ups. We prove several theoretical results showing that projections via various structured matrices followed by nonlinear mappings accurately preserve the angular distance between input high-dimensional vectors. To the best of our knowledge, these results are the first that give theoretical ground for the use of general structured matrices in the nonlinear setting. In particular, they generalize previous extensions of the Johnson-Lindenstrauss lemma and prove the plausibility of the approach that was so far only heuristically confirmed for some special structured matrices. Consequently, we show that many structured matrices can be used as an efficient information compression mechanism. Our findings build a better understanding of certain deep architectures, which contain randomly weighted and untrained layers, and yet achieve high performance on different learning tasks. We empirically verify our theoretical findings and show the dependence of learning via structured hashed projections on the performance of neural network as well as nearest neighbor classifier.

[1]Equal contribution.

1. Introduction

The paradigm of binary embedding for data compression is of the central focus of this paper. The paradigm has been studied in some earlier works (see: (Weiss et al., 2008), (Gong et al., 2013), (Plan & Vershynin, 2014), (Yi et al., 2015)), and in particular it was observed that by using linear projections and then applying sign function as a nonlinear map one does not loose completely the information about the angular distance between vectors, but instead the information might be approximately reconstructed from the Hamming distance between hashes. In this paper we are interested in using pseudo-random projections via structured matrices in the linear projection phase. The pseudo-random projection is described by a matrix, where not all the entries are independent random variables but instead a fixed "budget of randomness" is distributed across the matrix. Thus they can be efficiently stored in a sub-quadratic or even linear space and provide reduction in the randomness usage. Moreover, using them often leads to computational speed ups since they provide fast matrix-vector multiplications via Fast Fourier Transform. We prove an extension of the Johnson-Lindenstrauss lemma (Sivakumar, 2002) for general pseudo-random structured projections followed by nonlinear mappings. We show that the angular distance between high-dimensional data vectors is approximately preserved in the hashed space. This result is also new compared to previous extensions (Hinrichs & Vybral, 2011; Vybral, 2011) of the Johnson-Lindenstrauss lemma, that consider special cases of our structured projections (namely: circulant matrices) and do not consider at all the action of the non-linear mapping. We give theoretical explanation of the approach that was so far only heuristically confirmed for some special structured matrices.

Our theoretical findings imply that many types of matrices, such as circulant or Toeplitz Gaussian matrices, can be used as a preprocessing step in neural networks. Structured matrices were used before in different contexts also in deep learning, see for example (Saxe et al., 2011; Mathieu et al., 2014; Sindhwani et al., 2015)). Our theoretical results however extend to more general class of matrices.

Our work has primarily theoretical focus, but we also ask an empirical question: how the action of the random projection followed by non-linear transformation may influence learning? We focus on the deep learning setting, where the architecture contains completely random or pseudo-random structured layers that are not trained. Little is known from the theoretical point of view about these fast deep architectures, which achieve significant speed ups of computation and space usage reduction with simultaneous little or no loss in performance (Saxe et al., 2011; Jarrett et al., 2009; Pinto et al., 2009; Pinto & Cox, 2010; Huang et al., 2006). The high-level intuition justifying the success of these approaches is that not only does the performance of the deep learning system depend on learning, but also on the intrinsic properties of the architecture. These findings coincide with the notion of high redundancy in network parametrization (Denil et al., 2013; Denton et al., 2014; Choromanska et al., 2015). In this paper we consider a simple model of the fully-connected feed-forward neural network where the input layer is hashed by a structured pseudo-random projection followed by a point-wise nonlinear mapping. Thus the input is effectively hashed and learning is conducted in the fully connected subsequent layers that act in the hashed space. We empirically verify how the distortion introduced in the first layer by hashing (where we reduce the dimensionality of the data) affects the performance of the network (in the supervised learning setting). Finally, we show how our structured nonlinear embeddings can be used in the k-nn setting (Altman, 1992).

This article is organized as follows: Section 2 discusses related work, Section 3 explains the hashing mechanism, Section 4 provides theoretical results, Section 5 shows experimental results, and Section 6 concludes. Supplementary material contains additional proofs and experimental results.

2. Related work

The idea of using random projections to facilitate learning with high-dimensional data stems from the early work on random projections (Dasgupta, 1999) showing in particular that learning of high-dimensional mixtures of Gaussians can be simplified when first projecting the data into a randomly chosen subspace of low dimension (this is a consequence of the curse of dimensionality and the fact that high-dimensional data often has low intrinsic dimensionality). This idea was subsequently successfully applied to both synthetic and real datasets (Dasgupta, 2000; Bingham & Mannila, 2001), and then adopted to a number of learning approaches such as random projection trees (Dasgupta & Freund, 2008), kernel and feature-selection techniques (Blum, 2006), clustering (Fern & Brodley, 2003), privacy-preserving machine learning (Liu et al., 2006; Choromanska et al., 2013), learning with large databases (Achlioptas, 2003), and more recently - deep learning (see (Saxe et al., 2011) for convenient review of such approaches). Using linear projections with completely random Gaussian weights, instead of learned ones, was recently studied from both theoretical and practical point of view in (Giryes et al., 2015), but that work did not consider structured matrices which is a central point of our interest since structured matrices can be stored much more efficiently. Beyond applying methods that use random Gaussian matrix projections (Dasgupta, 1999; 2000; Giryes et al., 2015) and random binary matrix projections (Achlioptas, 2003), it is also possible to construct deterministic projections that preserve angles and distances (Jafarpour et al., 2009). In some sense these methods use structured matrices as well, yet they do not have the same projection efficiency of circulant matrices and projections explored in this article. Our hybrid approach, where a fixed "budget of randomness" is distributed across the entire matrix in the structured way enables us to take advantage of both: the ability of completely random projection to preserve information and the compactness that comes from the highly-organized internal structure of the linear mapping.

This work studies the paradigm of constructing a binary embedding for data compression, where hashes are obtained by applying linear projections to the data followed by the non-linear (sign function) mappings. The point-wise nonlinearity was not considered in many previous works on structured matrices (Haupt et al., 2010; Rauhut et al., 2010; Krahmer et al., 2014; Yap et al., 2011) (moreover note that these works also consider the set of structured matrices which is a strict subset of the class of matrices considered here). Designing binary embeddings for high dimensional data with low distortion is addressed in many recent works (Weiss et al., 2008; Gong et al., 2013; Yi et al., 2015; Raginsky & Lazebnik, 2009; Salakhutdinov & Hinton, 2009). In the context of our work, one of the recent articles (Yi et al., 2015) is especially important since the authors introduce the pipeline of constructing hashes with the use of structured matrices in the linear step, instead of completely random ones. They prove several theoretical results regarding the quality of the produced hash, and extend some previous theoretical results (Jacques et al., 2011; Plan & Vershynin, 2014). Their pipeline is more complicated than ours, i.e. they first apply Hadamard transformation and then a sequence of partial Toeplitz Gaussian matrices.

Some general results (unbiasedness of the angular distance estimator) were also known for short hashing pipelines involving circulant matrices ((Yu et al., 2014)). These works do not provide guarantees regarding concentration of the estimator around its mean, which is crucial for all practical applications. Our results for general structured matrices, which include circulant Gaussian matrices and a larger class of Toeplitz Gaussian matrices as special subclasses, provide such concentration guarantees, and thus establish a solid mathematical foundation for using various types of structured matrices in binary embeddings. In contrast to (Yi et al., 2015), we present our theoretical results for simpler hashing models (our hashing mechanism is explained in Section 3 and consists of two very simple steps that we call preprocessing step and the actual hashing step, where the latter consists of pseudo-random projection followed by the nonlinear mapping).

In the context of deep learning, using random network parametrization, where certain layers have random and untrained weights, often accelerates training. Introducing randomness to networks was explored for various architectures, in example feedforward networks (Huang et al., 2006), convolutional networks (Jarrett et al., 2009; Saxe et al., 2011), and recurrent networks (Jaeger & Haas, 2004; White et al., 2004; Boedecker et al., 2009). We also refer the reader to (Ganguli & Sompolinsky, 2012), where the authors describe how neural systems cope with the challenge of processing data in high dimensions and discuss random projections. Hashing in neural networks that we consider in this paper is a new direction of research. Very recently (see: (Chen et al., 2015)) it was empirically showed that hashing in neural nets may achieve drastic reduction in model sizes with no significant loss of the quality, by heavily exploiting the phenomenon of redundancies in neural nets. HashedNets introduced in (Chen et al., 2015) do not give any theoretical guarantees regarding the quality of the proposed hashing. Our work aims to touch both grounds. We experimentally show the plausibility of the approach, but also explain theoretically why the hashing we consider compresses important information about the data that suffices for accurate classification.

3. Hashing mechanism

In this section we explain our hashing mechanism for dimensionality reduction that we next analyze.

3.1. Structured matrices

We first introduce the aforementioned family of structured matrices, that we call: Ψ-regular matrices $\mathcal{P}$. This is the key ingredient of the method.

Definition 3.1. *$\mathcal{M}$ is a circulant Gaussian matrix if its first row is a sequence of independent Gaussian random variables taken from the distribution $\mathcal{N}(0,1)$ and next rows are obtained from the previous ones by left/right shifts.*

Definition 3.2. *$\mathcal{M}$ is a Toeplitz Gaussian matrix if each of its descending diagonals from left to right is of the form $(g,...,g)$ for some $g \sim \mathcal{N}(0,1)$ and different descending diagonals are independent.*

Remark 3.1. *The circulant Gaussian matrix with right shifts is a special type of the Toeplitz Gaussian matrix.*

Assume that k is the size of the hash and n is the dimensionality of the data.

Definition 3.3. *Let t be the size of the pool of independent random Gaussian variables $\{g_1,...,g_t\}$, where each $g_i \sim \mathcal{N}(0,1)$. Assume that $k \leq n \leq t \leq kn$. We take Ψ to be a natural number, i.e. $\Psi \in \mathbb{N}$. $\mathcal{P}$ is Ψ-regular random matrix if it has the following form*

$$\begin{pmatrix} \sum_{l\in\mathcal{S}_{1,1}} g_l & \dots & \sum_{l\in\mathcal{S}_{1,j}} g_l & \dots & \sum_{l\in\mathcal{S}_{1,n}} g_l \\ \dots & \dots & \dots & \dots & \dots \\ \sum_{l\in\mathcal{S}_{i,1}} g_l & \dots & \sum_{l\in\mathcal{S}_{i,j}} g_l & \dots & \sum_{l\in\mathcal{S}_{i,n}} g_l \\ \dots & \dots & \dots & \dots & \dots \\ \sum_{l\in\mathcal{S}_{k,1}} g_l & \dots & \sum_{l\in\mathcal{S}_{k,j}} g_l & \dots & \sum_{l\in\mathcal{S}_{k,n}} g_l \end{pmatrix} \tag{1}$$

where $S_{i,j} \subseteq \{1,...,t\}$ for $i \in \{1,...,k\}$, $j \in \{1,...,n\}$, $|S_{i,1}| = ... = |S_{i,n}|$ for $i = 1,...,k$, $S_{i,j_1} \cap S_{i,j_2} = \emptyset$ for $i \in \{1,...,k\}$, $j_1, j_2 \in \{1,...,n\}$, $j_1 \neq j_2$, and furthermore the following holds:

- *for a fixed column $\mathcal{C}$ of $\mathcal{P}$ and fixed $l \in \{1,...,t\}$ random variable g_l appears in at most $\Psi + 1$ entries from $\mathcal{C}$.*

Remark 3.2. *Circulant Gaussian matrices and Toeplitz Gaussian matrices are special cases of the 0-regular matrices. Toeplitz Gaussian matrix is 0-regular, where subsets $\mathcal{S}_{i,j}$ are singletons.*

In the experimental section of this paper we consider six different kinds of structured matrices, which are examples of general structured matrices covered by our theoretical analysis. Those are:

- *Circulant* (see: Definition 3.1),
- *BinCirc* - a matrix, where the first row is partitioned into consecutive equal-length blocks of elements and each row is obtained by the right shift of the blocks from the first row,
- *HalfShift* - a matrix, where next row is obtained from the previous one by swapping its halves and then performing right shift by one,
- *VerHorShift* - a matrix that is obtained by the following two phase-procedure: first each row is obtained from the previous one by the right shift of a fixed length and then in the obtained matrix each column is shifted up by a fixed number of elements,
- *BinPerm* - a matrix, where the first row is partitioned into consecutive equal-length blocks of elements and each row is obtained as a random permutation of the blocks from the first row,
- *Toeplitz* (see: Definition 3.2).

Remark 3.3. *Computing hashes for structured matrices: Toeplitz, BinCirc, HalfShift, and VerHorShift can be done faster than in time $\mathcal{O}(nk)$ (e.g. for Toeplitz one can use FFT to reduce computations to $\mathcal{O}(n \log k)$). Thus our structured approach leads to speed-ups, storage compression (since many structured matrices covered by our theoretical model can be stored in linear space) and reduction in randomness usage. The goal of this paper is not to analyze in details fast matrix-vector product algorithms since that requires a separate paper. We however point out that well-known fast matrix-vector product algorithms are some of the key advantages of our structured approach.*

3.2. Hashing methods

Let ϕ be a function satisfying $\lim_{x \to \infty} \phi(x) = 1$ and $\lim_{x \to -\infty} \phi(x) = -1$. We will consider two hashing methods, both of which consist of what we refer to as a *preprocessing* step followed by the actual *hashing* step, where the latter consists of pseudo-random projection followed by nonlinear (sign function) mapping. The first mechanism, that we call *extended Ψ-regular hashing*, applies first random diagonal matrix $\mathcal{R}$ to the data point x, then the L_2-normalized Hadamard matrix $\mathcal{H}$, next another random diagonal matrix $\mathcal{D}$, then the Ψ-regular projection matrix $\mathcal{P}_\Psi$ and finally function ϕ (the latter one applied point-wise). The overall scheme is presented below:

$$\underbrace{x \xrightarrow{\mathcal{R}} x_{\mathcal{R}} \xrightarrow{\mathcal{H}} x_{\mathcal{H}} \xrightarrow{\mathcal{D}} x_{\mathcal{D}}}_{\text{preprocessing}} \underbrace{\xrightarrow{\mathcal{P}_\Psi} x_{\mathcal{P}_\Psi} \xrightarrow{\phi} h(x)}_{\text{hashing}} \in \mathbb{R}^k. \quad (2)$$

The diagonal entries of matrices $\mathcal{R}$ and $\mathcal{D}$ are chosen independently from the binary set $\{-1, 1\}$, each value being chosen with probability $\frac{1}{2}$. We also propose a shorter pipeline, that we call *short Ψ-regular hashing*, where we avoid applying first random matrix $\mathcal{R}$ and the Hadamard matrix $\mathcal{H}$, i.e. the overall pipeline is of the form:

$$\underbrace{x \xrightarrow{\mathcal{D}} x_{\mathcal{D}}}_{\text{preprocessing}} \underbrace{\xrightarrow{\mathcal{P}_\Psi} x_{\mathcal{P}_\Psi} \xrightarrow{\phi} h(x)}_{\text{hashing}} \in \mathbb{R}^k. \quad (3)$$

The goal is to compute good approximation of the angular distance between given vectors p, r, given their compact hashed versions: $h(p), h(r)$. To achieve this goal we consider the L_1-distances in the k-dimensional space of hashes. Let $\theta_{p,r}$ denote the angle between vectors p and r. We define the *normalized approximate angle between p and r* as:

$$\tilde{\theta}^n_{p,r} = \frac{1}{2k} \|h(p) - h(r)\|_1 \quad (4)$$

In the next section we show that the normalized approximate angle between vectors p and r leads to a very precise estimation of the actual angle for $\phi(x) = sign(x)$ if the chosen parameter Ψ is not too large. Furthermore, we show an intriguing connection between theoretical guarantees regarding the quality of the produced hash and the chromatic number of some specific undirected graph encoding the structure of $\mathcal{P}$. For many of the structured matrices under consideration this graph is induced by an algebraic group operation defining the structure of $\mathcal{P}$ (for instance, for the circular matrix the group is a single shift and the underlying graph is a collection of pairwise disjoint cycles, thus its chromatic number is at most 3).

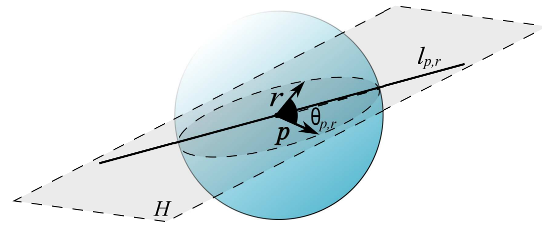


Figure 1: Two vectors: p, r spanning two-dimensional hyperplane H and with the angular distance $\theta_{p,r}$ between them. We have: $l_{p,r} = g^i_{\mathcal{D},H,\perp}$. Line $l_{p,r}$ is dividing $\theta_{p,r}$ and thus g^i contributes to $\|h(p) - h(r)\|_1$.

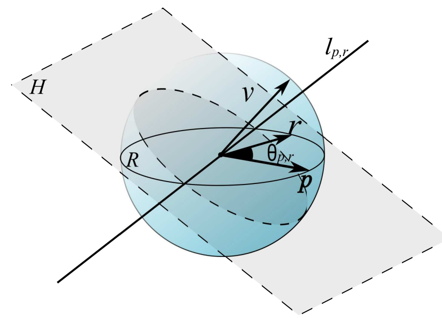


Figure 2: Similar setting to the one presented on Figure 1. Vector v represents L_2-normalized version of g^i and is perpendicular to the two-dimensional plane R. The intersection $R \cap H$ of that plane with the 2-dimensional plane H spanned by p, r is a line $l_{p,r}$ that this time is outside $\mathcal{U}_{p,r}$. Thus g^i does not contribute to $\|h(p) - h(r)\|_1$.

4. Theoretical results

4.1. Unbiasedness of the estimator

We are ready to provide theoretical guarantees regarding the quality of the produced hash. Our guarantees will be given for a *sign* function, i.e for ϕ defined as: $\phi(x) = 1$ for $x \geq 0$, $\phi(x) = -1$ for $x < 0$. We first show that $\tilde{\theta}^n_{p,r}$ is an unbiased estimator of $\frac{\theta_{p,r}}{\pi}$, i.e. $E(\tilde{\theta}^n_{p,r}) = \frac{\theta_{p,r}}{\pi}$.

Lemma 4.1. *Let $\mathcal{M}$ be a Ψ-regular hashing model (either extended or a short one) and $\|p\|_2 = \|r\|_2 = 1$. Then $\tilde{\theta}^n_{p,r}$ is an unbiased estimator of $\frac{\theta_{p,r}}{\pi}$, i.e. $E(\tilde{\theta}^n_{p,r}) = \frac{\theta_{p,r}}{\pi}$.*

Let us give a short sketch of the proof first. Note that the value of the particular entry in the constructed hash depends only on the sign of the dot product between the corresponding Gaussian vector representing the row of the Ψ-regular matrix and the given vector. Fix two vectors: p and r with angular distance θ. Note that considered dot products (and thus also their signs) are preserved when instead of taking the Gaussian vector representing the row one takes its projection onto a linear space spanned by p and r. The Hamming distance between hashes of p and r is built up by these entries for which one dot product is negative and the other one is positive. One can note that this happens if the projected vector is inside a 2-dimensional cone covering angle 2θ. The last observation that completes the proof is that the projection of the Gaussian vector is isotropic (since it is also Gaussian), thus the probability that the two dot products will have different signs is $\frac{\theta}{\pi}$.

Proof. Note first that the ith row, call it g^i, of the matrix $\mathcal{P}$ is a n-dimensional Gaussian vector with mean 0 and where each element has variance σ_i^2 for $\sigma_i^2 = |\mathcal{S}_{i,1}| = ... = |\mathcal{S}_{i,n}|$ ($i = 1, ..., k$). Thus, after applying matrix $\mathcal{D}$ the new vector $g^i_{\mathcal{D}}$ is still Gaussian and of the same distribution. Let us consider first the short Ψ-regular hashing model. Fix some vectors p, r (without loss of generality we may assume that they are not collinear). Let $H_{p,r}$, shortly called by us H, be the 2-dimensional hyperplane spanned by $\{p, r\}$. Denote by $g^i_{\mathcal{D},H}$ the projection of $g^i_{\mathcal{D}}$ into H and by $g^i_{\mathcal{D},H,\perp}$ the line in H perpendicular to $g^i_{\mathcal{D},H}$. Let ϕ be a *sign* function. Note that the contribution to the L_1-sum $\|h(p) - h(r)\|_1$ comes from those g^i for which $g^i_{\mathcal{D},H,\perp}$ divides an angle between p and r (see: Figure 1), i.e. from those g^i for which $g^i_{\mathcal{D},H}$ is inside the union $\mathcal{U}_{p,r}$ of two 2-dimensional cones bounded by two lines in H perpendicular to p and r respectively. If the angle is not divided (see: Figure 2) then the two corresponding entries in the hash have the same value and thus do not contribute to the overall distance between hashes.

Observe that, from what we have just said, we can conclude that $\tilde{\theta}^n_{p,r} = \frac{X_1 + ... + X_k}{k}$, where:

$$X_i = \begin{cases} 1 & \text{if } g^i_{\mathcal{D},H} \in \mathcal{U}_{p,r}, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Now it suffices to note that vector $g^i_{\mathcal{D},H}$ is a 2-dimensional Gaussian vector and thus its direction is uniformly distributed over all directions. Thus each X_i is nonzero with probability exactly $\frac{\theta_{p,r}}{\pi}$ and the theorem follows. For the extended Ψ-regular hashing model the analysis is very similar. The only difference is that data is preprocessed by applying $\mathcal{HR}$ linear mapping first. Both $\mathcal{H}$ and $\mathcal{R}$ are orthogonal matrices though, thus their product is also an orthogonal matrix. Since orthogonal matrices do not change angular distance, the former analysis can be applied again and yields the proof. □

We next focus on the concentration of the random variable $\tilde{\theta}^n_{p,r}$ around its mean $\frac{\theta_{p,r}}{\pi}$. We prove strong exponential concentration results for the extended Ψ-regular hashing method. Interestingly, the application of the Hadamard mechanism is not necessary and it is possible to get concentration results, yet weaker than in the former case, also for short Ψ-regular hashing.

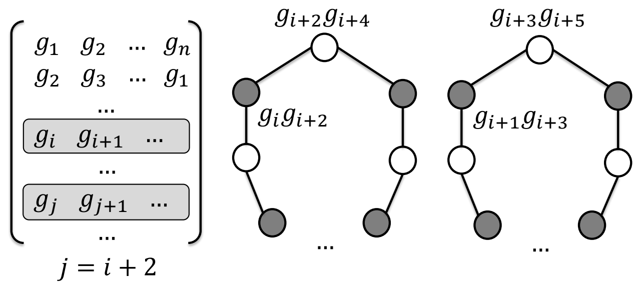


Figure 3: On the left: matrix $\mathcal{P}$ with two highlighted rows of indices: $k_1 = i$ and $k_2 = j$ respectively, where $j = i + 2$. On the right: corresponding graph that consists of two cycles. If each cycle is even then this graph is 2-colorable, as indicated on the picture. Thus we have: $\chi(\mathcal{G}_{\mathcal{P}}(k_1, k_2)) = 2$.

4.2. The $\mathcal{P}$-chromatic number

The highly well organized structure of the projection matrix $\mathcal{P}$ gives rise to the underlying undirected graph that encodes dependencies between different entries of $\mathcal{P}$. More formally, let us fix two rows of $\mathcal{P}$ of indices $1 \leq k_1 < k_2 \leq k$ respectively. We define a graph $\mathcal{G}_{\mathcal{P}}(k_1, k_2)$ as follows:

- $V(\mathcal{G}_{\mathcal{P}}(k_1, k_2)) = \{\{j_1, j_2\} : \exists l \in \{1, ..., t\} s.t. g_l \in \mathcal{S}_{k_1,j_1} \cap \mathcal{S}_{k_2,j_2}, j_1 \neq j_2\}$,
- there exists an edge between vertices $\{j_1, j_2\}$ and $\{j_3, j_4\}$ iff $\{j_1, j_2\} \cap \{j_3, j_4\} \neq \emptyset$.

The chromatic number $\chi(\mathcal{G})$ of the graph $\mathcal{G}$ is the minimal number of colors that can be used to color the vertices of the graph in such a way that no two adjacent vertices have the same color.

Definition 4.1. *Let $\mathcal{P}$ be a Ψ-regular matrix. We define the $\mathcal{P}$-chromatic number $\chi(\mathcal{P})$ as:*

$$\chi(\mathcal{P}) = \max_{1 \leq k_1 < k_2 \leq k} \chi(\mathcal{G}(k_1, k_2)).$$

The graph associated with each structured matrix that we have just described enables us to encode dependencies between entries of the structured matrix in the compact form and gives us quantitative ways to efficiently measure these dependencies by analyzing several core parameters of this graph such as its chromatic number. More dependencies that usually lead to more structured form mean more edges in the associated graph and often lead to higher chromatic number. On the other hand, fewer dependencies produce graphs with much lower chromatic number (see Figure 3, where the graph associated with the circulant matrix is a collection of vertex disjoint cycles and thus has chromatic number 3 if it contains an odd length cycle and 2 otherwise).

4.3. Concentration inequalities for structured hashing with *sign* function

We present now our main theoretical results. The proofs are deferred to the Supplementary material. We focus on the concentration results regarding produced hashes that are crucial for practical applications of the proposed scheme.

We first start with the short description of the methods used and then rigorously formulate all the results. If all the rows of the projection matrix are independent then standard concentration inequalities can be used. This is however not the case in our setting since the matrix is structured. We still want to say that any two rows are "close" to independent Gaussian vectors and that will give us bounds regarding the variance of the distance between the hashes (in general, we observe that any system of k rows is "close" to the system of k independent Gaussian vectors and get bounds involving kth moments). We proceed as follows:

- We take two rows and project then onto the linear space spanned by given vectors: p and r.
- We consider the four coordinates obtained in this way (two for each vector). These coordinates are obviously Gaussian, but what is crucial, they are "almost independent".
- The latter observation is implied by the fact that these are the coordinates of the projection of a fixed Gaussian vector onto directions that are "almost orthogonal".
- We use the property of the Gaussian vector, namely that its projections onto orthogonal directions are independent.
- To prove that directions considered in our setting are close to orthogonal with high probability, we compute their dot product. This is the place where the structure of the matrix, the chromatic number of the underlying graph and the fact that in our hashing scheme we use random diagonal matrices come into action. We decompose each dot product into roughly speaking χ components (where χ stands for the chromatic number), such that each component is a sum of independent random variables with mean 0. Now we can use standard concentration inequalities to get tight concentration results.
- The Hadamard matrix used in the extended model preprocesses input vectors to distribute their mass uniformly over all the coordinates, while not changing L_2 distances (note that it is a unitary matrix). It turns out that balanced vectors lead to much stronger concentration results.

Now we are ready to rigorously state our results. By $poly(x)$ we denote a function x^r for some $r > 0$. The following theorems guarantee strong concentration of $\tilde{\theta}^n_{p,r}$ around its mean and therefore justify theoretically the effectiveness of the structured hashing method.

Let us consider first the extended Ψ-regular hashing model. The following is our main theoretical result.

Theorem 4.1. *Consider extended Ψ-regular hashing model $\mathcal{M}$ with t independent Gaussian random variables: $g_1, ..., g_t$, each of distribution $\mathcal{N}(0,1)$. Let N be the size of the dataset $\mathcal{D}$. Denote by k the size of the hash and by n the dimensionality of the data. Let $f(n)$ be an arbitrary positive function. Let $\theta_{p,r}$ be the angular distance between vectors $p, r \in \mathcal{D}$. Then for every $a, \epsilon > 0$ the following is true:*

$$\mathbb{P}\left(\forall_{p,r\in\mathcal{D}} \left|\tilde{\theta}^n_{p,r} - \frac{\theta_{p,r}}{\pi}\right| \leq \epsilon\right) \geq \left[1 - 4\binom{N}{2}e^{-\frac{f^2(n)}{2}} - 4\chi(\mathcal{P})\binom{k}{2}e^{-\frac{2a^2t}{f^4(t)}}\right](1-\Lambda),$$

where $\Lambda = \frac{1}{\pi}\sum_{j=\lfloor\frac{\epsilon k}{2}\rfloor}^{k}\frac{1}{\sqrt{j}}(\frac{ke}{j})^j\mu^j(1-\mu)^{k-j} + 2e^{-\frac{\epsilon^2 k}{2}}$ and $\mu = \frac{8k(a\chi(\mathcal{P})+\Psi\frac{f^2(n)}{n})}{\theta_{p,r}}$.

Note how the upper bound on the probability of failure $\mathbb{P}_\epsilon$ depends on the $\mathcal{P}$-chromatic number. The theorem above guarantees strong concentration of $\tilde{\theta}^n_{p,r}$ around its mean and therefore justifies theoretically the effectiveness of the structured hashing method. It becomes more clear below.

As a corollary, we obtain the following result:

Corollary 4.1. *Consider extended Ψ-regular hashing model $\mathcal{M}$. Assume that the projection matrix $\mathcal{P}$ is Toeplitz Gaussian. Let N, n, k be as above and denote by $\theta_{p,r}$ be the angular distance between vectors $p, r \in \mathcal{D}$. Then the following is true for n large enough:*

$$\mathbb{P}\left(\forall_{p,r\in\mathcal{D}} \left|\tilde{\theta}^n_{p,r} - \frac{\theta_{p,r}}{\pi}\right| \leq k^{-\frac{1}{3}}\right) \geq \left[1 - O\left(\frac{N^2}{e^{poly(n)}} + k^2 e^{-n^{\frac{3}{10}}}\right)\right]\left(1 - 3e^{-\frac{k^{\frac{1}{3}}}{2}}\right).$$

Corollary 4.1 follows from Theorem 4.1 by taking: $a = n^{-\frac{1}{3}}$, $\epsilon = k^{-\frac{1}{3}}$, $f(n) = n^p$ for small enough constant $p > 0$, noticing that every Toeplitz Gaussian matrix is 0-regular and the corresponding $\mathcal{P}$-chromatic number $\chi(\mathcal{P})$ is at most 3.

Let us consider now the short Ψ-regular hashing model. The theorem presented below is an application of the Chebyshev's inequality preceded by the careful analysis of the variance of $\tilde{\theta}^n_{p,r}$.

Theorem 4.2. *Consider short Ψ-regular hashing model $\mathcal{M}$, where $\mathcal{P}$ is a Toeplitz Gaussian matrix. Denote by k the size of the hash. Let $\theta_{p,r}$ be the angular distance between vectors $p, r \in \mathcal{D}$, where $\mathcal{D}$ is the dataset. Then the following is true*

$$\forall_{p,r\in\mathcal{D}} Var(\tilde{\theta}^n_{p,r}) \leq \frac{1}{k}\frac{\theta_{p,r}(\pi - \theta_{p,r})}{\pi^2} + \left(\frac{\log(k)}{k^2}\right)^{\frac{1}{3}}, \quad (6)$$

and thus for any $c > 0$ and $p, r \in \mathcal{D}$:

$$\mathbb{P}\left(\left|\tilde{\theta}^n_{p,r} - \frac{\theta_{p,r}}{\pi}\right| \geq c\left(\frac{\sqrt{\log(k)}}{k}\right)^{\frac{1}{3}}\right) = O\left(\frac{1}{c^2}\right).$$

Figure 4 and 5 show the dependence of the upper bound on the variance of the normalized approximate angle $\tilde{\theta}^n_{p,r}$ on resp. the true angular distance $\theta_{p,r}$ and the size of the hash k when resp. k and $\theta_{p,r}$ are fixed.

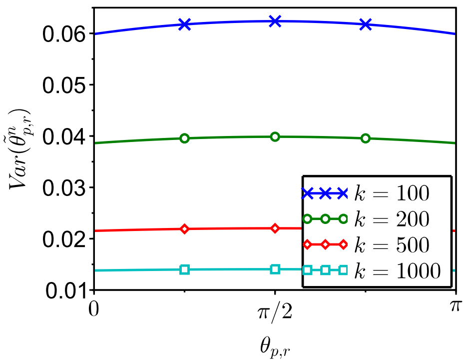


Figure 4: The dependence of the upper bound on the variance of the normalized approximate angle $\tilde{\theta}^n_{p,r}$ on an angle when the size of the hash k is fixed. The upper bound scales as $\frac{1}{k}$ and is almost independent of $\theta_{p,r}$.

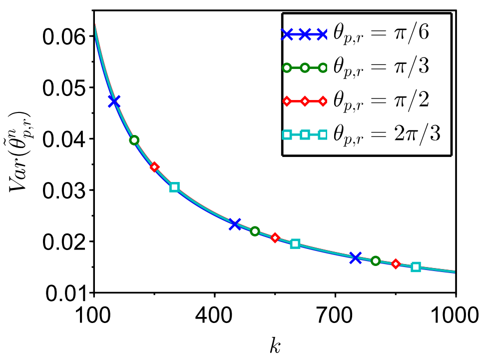


Figure 5: The dependence of the upper bound on the variance of the normalized approximate angle $\tilde{\theta}^n_{p,r}$ on the size of the hash k when the true angular distance $\theta_{p,r}$ is fixed. The upper bound converges to 0 as $k \to \infty$.

5. Numerical experiments

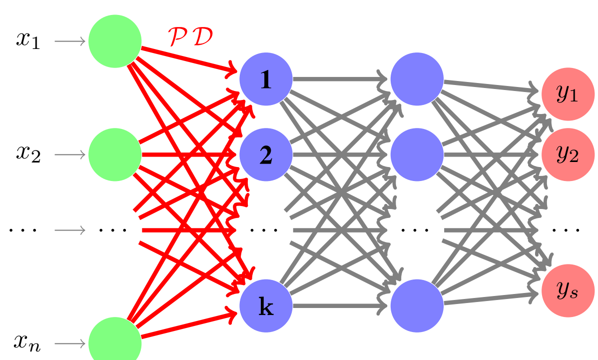
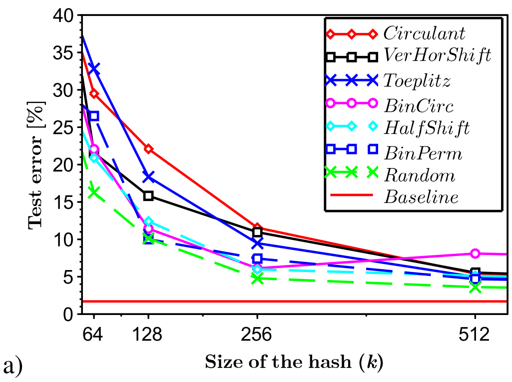
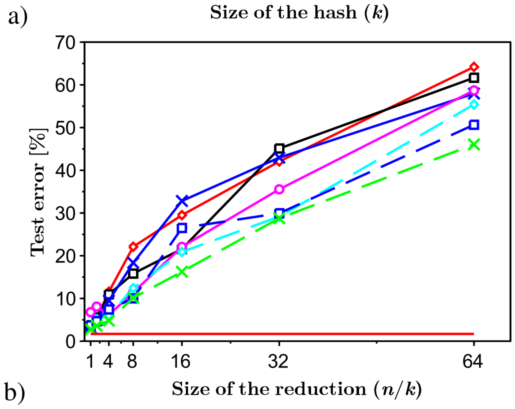


Figure 6: Fully-connected network with randomized input layer (red edges correspond to structured matrix). $k < n$. $\mathcal{D}$ is a random diagonal matrix with diagonal entries chosen independently from the binary set $\{-1, 1\}$, each value being chosen with probability $\frac{1}{2}$, and $\mathcal{P}$ is a structured matrix. *The figure should be viewed in color.*

We performed experiments on *MNIST* dataset downloaded from `http://yann.lecun.com/exdb/mnist/`. The data was preprocessed[2] according to the *short* hashing scheme (the *extended* hashing scheme gave results of no significant statistical difference) before being given to the input of the network. We considered a simple model of the fully-connected feed-forward neural network with two hidden layers, where the first hidden layer had k units that use sign non-linearity (we explored $k = \{16, 32, 64, 128, 256, 512, 1024\}$), and the second hidden layer had 100 units that use ReLU non-linearity. The size of the second hidden layer was chosen as follows. We first investigated the dependence of the test error on this size in case when $n = k$ and the inputs instead of being randomly projected are multiplied by identity (it is equivalent to eliminating first hidden layer). We then chose as a size the threshold below which test performance was rapidly deteriorating.



a)



b)

Figure 7: Mean test error versus a) the size of the hash (k) (zoomed plot[3]), b) the size of the reduction (n/k) for the network. Baseline corresponds to 1.7%.

The first hidden layer contains random untrained weights, and we only train the parameters of the second layer and the output layer. The network we consider is shown in Figure 6. Each experiment was initialized from a random set of parameters sampled uniformly within the unit cube, and was repeated 1000 times. All networks were trained for 30 epochs using SGD (Bottou, 1998). The experiments with constant learning rate are reported (we also explored learning rate decay, but obtained similar results), where the learning rate was chosen from the set

[2]Preprocessing is discussed in Section 3.

Table 1: Mean and std of the test error versus the size of the hash (k) / size of the reduction (n/k) for the network.

k / $\frac{n}{k}$	Circulant [%]	Random [%]	BinPerm [%]	BinCirc [%]	HalfShift [%]	Toeplitz [%]	VerHorShift [%]
1024 / 1	3.53 ± 0.16	2.78 ± 0.10	3.69 ± 0.21	6.79 ± 0.49	3.54 ± 0.16	3.16 ± 0.19	3.74 ± 0.16
512 / 2	5.42 ± 0.83	3.61 ± 0.19	4.68 ± 0.35	8.10 ± 1.85	5.13 ± 2.15	4.97 ± 0.53	5.55 ± 0.62
256 / 4	11.56 ± 1.42	4.79 ± 0.13	7.43 ± 1.31	6.13 ± 1.42	5.98 ± 1.05	9.48 ± 1.88	10.96 ± 2.78
128 / 8	22.10 ± 5.42	10.13 ± 0.24	10.02 ± 0.50	11.43 ± 0.92	12.42 ± 0.95	18.35 ± 2.36	15.82 ± 1.63
64 / 16	29.50 ± 1.13	16.26 ± 1.02	26.50 ± 10.55	22.07 ± 1.35	20.90 ± 2.25	32.82 ± 4.83	21.59 ± 3.05
32 / 32	42.07 ± 4.16	28.77 ± 2.28	29.94 ± 3.48	35.55 ± 3.12	29.15 ± 0.97	42.97 ± 2.08	45.10 ± 4.46
16 / 64	64.20 ± 6.76	46.06 ± 1.03	50.65 ± 5.66	58.70 ± 7.15	55.40 ± 6.90	57.96 ± 3.65	61.66 ± 4.08

Table 2: Memory complexity and number of required random values for structured matrices and *Random* matrix.

Matrix	Random	Circulant	BinPerm	HalfShift	VerHorShift	BinCirc	Toeplitz
# of random values	$\mathcal{O}(nk)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$
Memory complexity	$\mathcal{O}(nk)$	$\mathcal{O}(n)$	$\mathcal{O}(nk)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$

$\{0.0005, 0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1\}$ to minimize the test error. The weights of the first hidden layer correspond to the entries in the "preprocessed" structured matrix. We explored seven kinds of random matrices (first six are structured): *Circulant*, *Toeplitz*, *HalfShift*, *VerHorShift*, *BinPerm*, *BinCirc*, and *Random* (entries are independent and drawn from Gaussian distribution $\mathcal{N}(0, 1)$). All codes were implemented in Torch7.

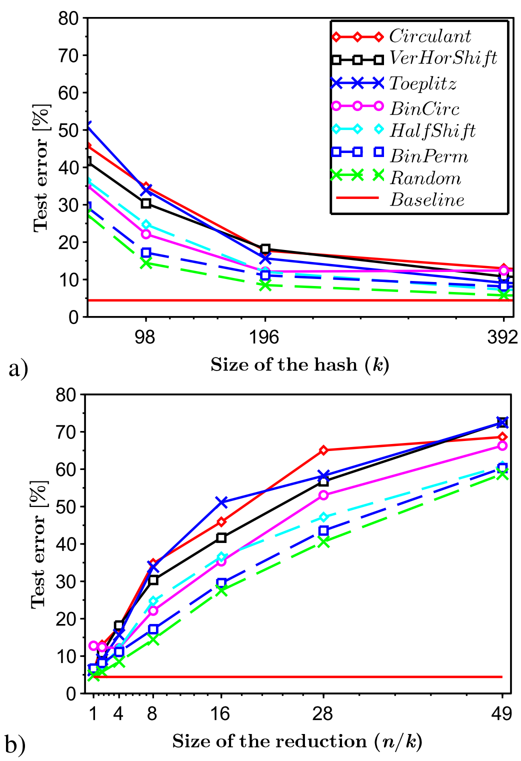


Figure 8: Mean test error versus a) the size of the hash (k) (zoomed plot[3]), b) the size of the reduction (n/k) for 1-NN. Baseline corresponds to 4.5%.

Figure 7a shows how the mean test error is affected by the size of the hash, and Figure 7b shows how the mean test error changes with the size of the reduction, where the size of the reduction is defined as the ratio n/k. In Table 1 we report both the mean and the standard deviation (std) of the test error across our experiments. Training results are reported in the Supplementary material. *Baseline* refers to the network with one hidden layer containing 100 hidden units, where all parameters are trained.

Experimental results show how the performance is affected by using structured hashed projections to reduce data dimensionality. Figure 7b and Table 1 show close to linear dependence between the error and the size of the reduction. Simultaneously, this approach leads to computational savings and the reduction of memory storage (in example for *Circulant* matrix the memory storage reduction[4] is of the order $\mathcal{O}(n/k)$). Memory complexity[5] and the number of required random values for different structured matrices and *Random* matrix are summarized in Table 2.

Experiments show that using fully random matrix gives the best performance as predicted in theory. *BinPerm* matrix exhibits comparable performance to the *Random* matrix, which might be explained by the fact that applying permutation itself adds an additional source of randomness. The next best performer is *HalfShift*, which generation is less random than in case of *BinPerm* or *Random*. Thus its performance, as expected, is worse than for these two other matrices. However, as opposed to *BinPerm* and *Random* matrices, *HalfShift* matrix can be stored in linear space. The results also show that in general all structured matrices perform relatively well for medium-size reductions. Finally, all structured matrices except for *BinPerm* lead to the biggest memory savings and require the smallest "budget of randomness". Moreover, they often lead to computational efficiency, e.g. *Toeplitz* matrix-vector multiplications can

[3]Original plot is in the Supplement.

[4]The reduction of memory storage is computed as the reduction of the number of input weights for the hidden layer (the memory required for storing *Circulant* matrix is negligible compared to the number of weights).

[5]Memory complexity is the required memory to store the matrix.

be efficiently implemented via Fast Fourier Transform (Yu et al., 2014). But, as mentioned before, faster than naive matrix-vector product computations can be performed also for *BinPerm*, *HalfShift*, and *VerHorShift*.

Finally, we also report how the performance of 1-NN algorithm is affected by using structured hashed projections for the dimensionality reduction. We obtained similar plots as for the case of neural networks. They are captured in Figure 8. The table showing the mean and the standard deviation of the test error for experiments with 1-NN is enclosed in the Supplementary material.

6. Conclusions

This paper shows that using structured hashed projections well-preserves the angular distance between input data instances. Our theoretical results consider mapping the data to lower-dimensional space using various structured matrices, where the structured linear projections are followed by the *sign* nonlinearity. This non-linear operation was not considered for such a wide range of structured matrices in previous related theoretical works. The theoretical setting naturally applies to the multilayer network framework, where the basic components of the architecture perform matrix-vector multiplication followed by the nonlinear mapping. We empirically verify our theoretical findings and show how using structured hashed projections for dimensionality reduction affects the performance of neural network and nearest neighbor classifier.

References

Achlioptas, D. Database-friendly random projections: Johnson-lindenstrauss with binary coins. *J. Comput. Syst. Sci.*, 66(4):671–687, 2003.

Altman, N. S. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *The American Statistician*, 46(3):175–185, 1992.

Bingham, E. and Mannila, H. Random projection in dimensionality reduction: Applications to image and text data. In *KDD*, 2001.

Blum, A. Random projection, margins, kernels, and feature-selection. In *SLSFS*, 2006.

Boedecker, J., Obst, O., Mayer, N. M., and Asada, M. Initialization and self-organized optimization of recurrent neural network connectivity. *HFSP journal*, 3(5):340–9, 2009.

Bottou, L. Online algorithms and stochastic approximations. In *Online Learning and Neural Networks*. Cambridge University Press, 1998.

Chen, W., Wilson, J. T., Tyree, S., Weinberger, K. Q., and Chen, Y. Compressing neural networks with the hashing trick. *CoRR*, abs/1504.04788, 2015.

Choromanska, A., Choromanski, K., Jagannathan, G., and Monteleoni, C. Differentially-private learning of low dimensional manifolds. In *ALT*, 2013.

Choromanska, A., Henaff, M., Mathieu, M., Arous, G. Ben, and LeCun, Y. The loss surfaces of multilayer networks. In *AISTATS*, 2015.

Dasgupta, S. Learning mixtures of gaussians. In *FOCS*, 1999.

Dasgupta, S. Experiments with random projection. In *UAI*, 2000.

Dasgupta, S. and Freund, Y. Random projection trees and low dimensional manifolds. In *STOC*, 2008.

Denil, M., Shakibi, B., Dinh, L., Ranzato, M., and Freitas, N. D. Predicting parameters in deep learning. In *NIPS*. 2013.

Denton, E., Zaremba, W., Bruna, J., LeCun, Y., and Fergus, R. Exploiting linear structure within convolutional networks for efficient evaluation. In *NIPS*. 2014.

Fern, X. Z. and Brodley, C. E. Random projection for high dimensional data clustering: A cluster ensemble approach. In *ICML*, 2003.

Ganguli, S. and Sompolinsky, H. Compressed sensing, sparsity, and dimensionality in neuronal information processing and data analysis. *Annual Review of Neuroscience*, 35:485–508, 2012.

Giryes, R., Sapiro, G., and Bronstein, A. M. Deep neural networks with random gaussian weights: A universal classification strategy? *CoRR*, abs/1504.08291, 2015.

Gong, Y., Lazebnik, S., Gordo, A., and Perronnin, F. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(12):2916–2929, 2013.

Haupt, J., Bajwa, W. U., Raz, G., and Nowak, R. Toeplitz compressed sensing matrices with applications to sparse channel estimation. *Information Theory, IEEE Transactions on*, 56(11):5862–5875, 2010.

Hinrichs, A. and Vybral, J. Johnson-lindenstrauss lemma for circulant matrices. *Random Struct. Algorithms*, 39 (3):391–398, 2011.

Huang, G.-B., Zhu, Q.-Y., and Siew, C.-K. Extreme learning machine: Theory and applications. *Neurocomputing*, 70(13):489 – 501, 2006.

Jacques, L., Laska, J. N., Boufounos, P., and Baraniuk, R. G. Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors. *CoRR*, abs/1104.3160, 2011.

Jaeger, H. and Haas, H. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, pp. 78–80, 2004.

Jafarpour, S., Xu, W., Hassibi, B., and Calderbank, R. Efficient and Robust Compressed Sensing Using Optimized Expander Graphs. *Information Theory, IEEE Transactions on*, 55(9):4299–4308, 2009.

Jarrett, K., Kavukcuoglu, K., Ranzato, M., and LeCun, Y. What is the best multi-stage architecture for object recognition? In *ICCV*, 2009.

Krahmer, F., Mendelson, S., and Rauhut, H. Suprema of chaos processes and the restricted isometry property. *Communications on Pure and Applied Mathematics*, 67 (11):1877–1904, 2014.

Liu, K., Kargupta, H., and Ryan, J. Random projection-based multiplicative data perturbation for privacy preserving distributed data mining. *IEEE Trans. on Knowl. and Data Eng.*, 18(1):92–106, 2006.

Mathieu, M., Henaff, M., and LeCun, Y. Fast training of convolutional networks through ffts. In *ICLR*, 2014.

Pinto, N. and Cox, D. D. An Evaluation of the Invariance Properties of a Biologically-Inspired System for Unconstrained Face Recognition. In *BIONETICS*, 2010.

Pinto, N., Doukhan, D., DiCarlo, J. J., and Cox, D. D. A high-throughput screening approach to discovering good forms of biologically inspired visual representation. *PLoS Computational Biology*, 5(11), 2009.

Plan, Y. and Vershynin, R. Dimension reduction by random hyperplane tessellations. *Discrete & Computational Geometry*, 51(2):438–461, 2014.

Raginsky, M. and Lazebnik, S. Locality-sensitive binary codes from shift-invariant kernels. In *NIPS*. 2009.

Rauhut, H., Romberg, J. K., and Tropp, J. A. Restricted isometries for partial random circulant matrices. *CoRR*, abs/1010.1847, 2010.

Salakhutdinov, R. and Hinton, G. Semantic hashing. *Int. J. Approx. Reasoning*, 50(7):969–978, 2009.

Saxe, A., Koh, P. W., Chen, Z., Bhand, M., Suresh, B., and Ng, A. On random weights and unsupervised feature learning. In *ICML*, 2011.

Sindhwani, V., Sainath, T., and Kumar, S. Structured transforms for small-footprint deep learning. In *NIPS*, 2015.

Sivakumar, D. Algorithmic derandomization via complexity theory. In *STOC*, 2002.

Vybral, J. A variant of the johnsonlindenstrauss lemma for circulant matrices. *Journal of Functional Analysis*, 260 (4):1096 – 1105, 2011.

Weiss, Y., Torralba, A., and Fergus, R. Spectral hashing. In *NIPS*, 2008.

White, O. L., Lee, D. D., and Sompolinsky, H. Short-term memory in orthogonal neural networks. *Physical review letters*, 92(14), 2004.

Yap, H.L., Eftekhari, A., Wakin, M.B., and Rozell, C.J. The restricted isometry property for block diagonal matrices. In *CISS*, 2011.

Yi, X., Caramanis, C., and Price, E. Binary embedding: Fundamental limits and fast algorithm. *CoRR*, abs/1502.05746, 2015.

Yu, F. X., Kumar, S., Gong, Y., and Chang, S.-F. Circulant binary embedding. In *ICML*, 2014.

Binary embeddings with structured hashed projections (Supplementary Material)

In this section we prove Theorem 4.1 and Theorem 4.2. We will use notation from Lemma 4.1.

7. Proof of Theorem 4.1

We start with the following technical lemma:

Lemma 7.1. *Let $\{Z_1, ..., Z_k\}$ be the set of k independent random variables defined on Ω such that each Z_i has the same distribution and $Z_i \in \{0, 1\}$. Let $\{\mathcal{F}_1, ..., \mathcal{F}_k\}$ be the set of events, where each $\mathcal{F}_i$ is in the σ-field defined by Z_i (in particular $\mathcal{F}_i$ does not depend on the σ-field $\sigma(Z_1, ..., Z_{i-1}, Z_{i+1}, ...Z_k)$). Assume that there exists $\mu < \frac{1}{2}$ such that: $\mathbb{P}(\mathcal{F}_i) \leq \mu$ for $i = 1, ..., k$. Let $\{U_1, ..., U_k\}$ be the set of k random variables such that $U_i \in \{0, 1\}$ and $U_i|\mathcal{F}_i = Z_i|\mathcal{F}_i$ for $i = 1, ..., k$, where $X|\mathcal{F}$ stands for the random variable X truncated to the event $\mathcal{F}$. Assume furthermore that $E(U_i) = E(Z_i)$ for $i = 1, ..., k$. Denote $U = \frac{U_1+...+U_k}{k}$. Then the following is true.*

$$\mathbb{P}(|U - EU| > \epsilon) \leq \frac{1}{\pi} \sum_{r=\frac{\epsilon k}{2}}^{k} \frac{1}{\sqrt{r}} (\frac{ke}{r})^r \mu^r (1-\mu)^{k-r} + 2e^{-\frac{\epsilon^2 k}{2}}. \tag{7}$$

Proof. Let us consider the event $\mathcal{F}_{bad} = \mathcal{F}_1 \cup ... \cup \mathcal{F}_k$. Note that $\mathcal{F}_{bad}$ may be represented by the union of the so-called r-blocks, i.e.

$$\mathcal{F}_{bad} = \bigcup_{Q \subseteq \{1,...,k\}} (\bigcap_{q \in Q} \mathcal{F}_q \bigcap_{q \in \{1,...,k\} \setminus Q} \mathcal{F}_q^c), \tag{8}$$

where $\mathcal{F}^c$ stands for the complement of event $\mathcal{F}$. Let us fix now some $Q \subseteq \{1, ..., k\}$. Denote

$$\mathcal{F}_Q = \bigcap_{q \in Q} \mathcal{F}_q \bigcap_{q \in \{1,...,k\} \setminus Q} \mathcal{F}_q^c. \tag{9}$$

note that $\mathbb{P}(\mathcal{F}_Q) \leq \mu^r (1-\mu)^{k-r}$. It follows directly from the Bernoulli scheme.

Denote $Z = \frac{Z_1+...+Z_k}{k}$. From what we have just said and from the definition of $\{\mathcal{F}_1, ..., \mathcal{F}_k\}$ we conclude that for any given c the following holds:

$$\mathbb{P}(|U - Z| > c) \leq \sum_{r=ck}^{k} \binom{k}{r} \mu^r (1-\mu)^{k-r}. \tag{10}$$

Note also that from the assumptions of the lemma we trivially get: $E(U) = E(Z)$.

Let us consider now the expression $\mathbb{P}(|U - E(U)|) > \epsilon$.

We get: $\mathbb{P}(|U - E(U)| > \epsilon) = \mathbb{P}(|U - E(Z)| > \epsilon) = \mathbb{P}(|U - Z + Z - E(Z)| > \epsilon) \leq \mathbb{P}(|U - Z| + |Z - E(Z)| > \epsilon) \leq \mathbb{P}(|U - Z| > \frac{\epsilon}{2}) + \mathbb{P}(|Z - E(Z)| > \frac{\epsilon}{2})$.

From 10 we get:

$$\mathbb{P}(|U - Z| > \frac{\epsilon}{2}) \leq \sum_{r=\frac{\epsilon}{2}}^{k} \binom{k}{r} \mu^r (1-\mu)^{k-r}. \tag{11}$$

Let us consider now the expression:

$$\xi = \sum_{r=\frac{\epsilon k}{2}}^{k} \binom{k}{r} \mu^r (1-\mu)^{k-r}. \tag{12}$$

We have:

$$\begin{aligned} \xi &\leq \sum_{r=\frac{\epsilon k}{2}}^{k} \frac{(k-r+1)...(k)}{r!} \mu^r (1-\mu)^{k-r} \\ &\leq \sum_{r=\frac{\epsilon k}{2}}^{k} \frac{k^r}{r!} \mu^r (1-\mu)^{k-r} \end{aligned} \tag{13}$$

From the Stirling's formula we get: $r! = \frac{2\pi r^{r+\frac{1}{2}}}{e^r}(1 + o_r(1))$. Thus we obtain:

$$\begin{aligned} \xi &\leq (1 + o_r(1)) \sum_{r=\frac{\epsilon k}{2}}^{k} \frac{k^r e^r}{2\pi r^{r+\frac{1}{2}}} \mu^r (1-\mu)^{k-r} \\ &\leq \frac{1}{\pi} \sum_{r=\frac{\epsilon k}{2}}^{k} \frac{1}{\sqrt{r}} (\frac{ke}{r})^r \mu^r (1-\mu)^{k-r} \end{aligned} \tag{14}$$

for r large enough.

Now we will use the following version of standard Azuma's inequality:

Lemma 7.2. *Let $W_1, ..., W_k$ be k independent random variables such that $E(W_1) = ...E(W_k) = 0$. Assume that $-\alpha_i \leq W_{i+1} - W_i \leq \beta_i$ for $i = 2, ..., k-1$. Then the following is true:*

$$\mathbb{P}(|\sum_{i=1}^{k} W_i| > a) \leq 2e^{-\frac{2a^2}{\sum_{i=1}^{k}(\alpha_i+\beta_i)^2}}$$

Now, using Lemma 7.2 for $W_i = X_i - E(X_i)$ and $\alpha_i = E(X_i), \beta_i = 1 - E(X_i)$ we obtain:

$$\mathbb{P}(|X - EX| > \frac{a}{2}) \leq 2e^{-\frac{a^2 k}{2}}. \tag{15}$$

Combining 14 and 15, we obtain the statement of the lemma.

□

Our next lemma explains the role the Hadamard matrix plays in the entire extended Ψ-regular hashing mechanism.

Lemma 7.3. *Let n denote data dimensionality and let $f(n)$ be an arbitrary positive function. Let D be the set of all L_2-normalized data points, where no two data points are identical. Assume that $|D| = N$. Consider the $\binom{N}{2}$ hyperplanes $H_{p,r}$ spanned by pairs of different vectors $\{p, r\}$ from D. Then after applying linear transformation $\mathcal{HR}$ each hyperplane $H_{p,r}$ is transformed into another hyperplane $H^{\mathcal{HR}}_{p,r}$. Furthermore, the probability $\mathcal{P}_{\mathcal{HR}}$that for every $H^{\mathcal{HR}}_{p,r}$ there exist two orthonormal vectors $x = (x_1, ..., x_n), y = (y_1, ..., y_n)$ in $H^{\mathcal{HR}}_{p,r}$ such that: $|x_i|, |y_i| \leq \frac{f(n)}{\sqrt{n}}$ satisfies:*

$$\mathcal{P}_{\mathcal{HR}} \geq 1 - 4\binom{N}{2} e^{-\frac{f^2(n)}{2}}.$$

Proof. We have already noted in the proof of Lemma 4.1 that $\mathcal{HR}$ is an orthogonal matrix. Thus, as an isometry, it clearly transforms each 2-dimensional hyperplane into another 2-dimensional hyperplane. For every pair $\{p, r\}$, let us consider an arbitrary fixed orthonormal pair $\{u, v\}$ spanning $H_{p,r}$. Denote $u = (u_1, ..., u_n)$. Let us denote by $u^{\mathcal{HR}}$ vector obtained from u after applying transformation $\mathcal{HR}$. Note that the j^{th} coordinate of $u^{\mathcal{HR}}$ is of the form:

$$u^{\mathcal{HR}}_j = u_1 T_1 + ... + u_n T_n, \tag{16}$$

where $T_1, ..., T_n$ are independent random variables satisfying:

$$T_i = \begin{cases} \frac{1}{\sqrt{n}} & \text{w.p } \frac{1}{2}, \\ -\frac{1}{\sqrt{n}} & \text{otherwise.} \end{cases} \tag{17}$$

The latter comes straightforwardly from the form of the L_2-normalized Hadamard matrix (i.e a Hadamard matrix, where each row and column is L_2-normalized).

But then, from Lemma 7.2, and the fact that $\|u\|_2 = 1$, we get for any $a > 0$:

$$\mathbb{P}(|u_1 T_1 + ... + u_n T_n| \geq a) \leq 2e^{-\frac{2a^2}{\sum_{i=1}^{n}(2u_i)^2}} \leq 2e^{-\frac{a^2}{2}}. \tag{18}$$

Similar analysis is correct for $v^{\mathcal{HR}}$. Note that $v^{\mathcal{HR}}$ is orthogonal to $u^{\mathcal{HR}}$ since v and u are orthogonal. Furthermore, both $v^{\mathcal{HR}}$ and $u^{\mathcal{HR}}$ are L_2-normalized. Thus $\{u^{\mathcal{HR}}, v^{\mathcal{HR}}\}$ is an orthonormal pair.

To complete the proof, it suffices to take $a = f(n)$ and apply the union bound over all vectors $u^{\mathcal{HR}}$, $v^{\mathcal{HR}}$ for all $\binom{N}{2}$ hyperplanes. □

From the lemma above we see that applying Hadamard matrix enables us to assume with high probability that for every hyperplane $H_{p,r}$ there exists an orthonormal basis consisting of vectors with elements of absolute values at most $\frac{f(n)}{\sqrt{n}}$. We call this event $\mathcal{E}_f$. Note that whether $\mathcal{E}_f$ holds or not is determined only by $\mathcal{H}$, $\mathcal{R}$ and the initial dataset D.

Let us proceed with the proof of Theorem 4.1. Let us assume that event $\mathcal{E}_f$ holds. Without loss of generality we may assume that we have the short Ψ-regular hashing mechanism with an extra property that every $H_{p,r}$ has an orthonormal basis consisting of vectors with elements of absolute value at most $\frac{f(n)}{\sqrt{n}}$. Fix two vectors p, r from the dataset D. Denote by $\{x, y\}$ the orthonormal basis of $H_{p,r}$ with the above property. Let us fix the ith row of $\mathcal{P}$ and denote it as $(p_{i,1}, ..., p_{i,n})$. After being multiplied by the diagonal matrix $\mathcal{D}$ we obtain another vector:

$$w = (\mathcal{P}_{i,1} d_1, ..., \mathcal{P}_{i,n} d_n), \tag{19}$$

where:

$$\mathcal{D}_{i,j} = \begin{pmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_n \end{pmatrix}. \tag{20}$$

We have already noted that in the proof of Lemma 4.1 that it is the projection of w into $H_{p,r}$ that determines whether the value of the associated random variable X_i is 0 or 1. To be more specific, we showed that $X_i = 1$ iff the projection is in the region $\mathcal{U}_{p,r}$. Let us write down the coordinates of the projection of w into $H_{p,r}$ in the $\{x, y\}$-coordinate system. The coordinates are the dot-products of w with x and y respectively thus in the $\{x, y\}$-coordinate system we can write w as:

$$w_{\{x,y\}} = (\mathcal{P}_{i,1} d_1 x_1, ..., \mathcal{P}_{i,n} d_n x_n, \mathcal{P}_{i,1} d_1 y_1, ..., \mathcal{P}_{i,n} d_n y_n). \tag{21}$$

Note that both coordinates are Gaussian random variables and they are independent since they were constructed by projecting a Gaussian vector into two orthogonal vectors. Now note that from our assumption about the structure of $\mathcal{P}$ we can conclude that both coordinates may be represented as sums of weighted Gaussian random variables g_i for $i = 1, ..., t$, i.e.:

$$w_{\{x,y\}} = (g_1 s_{i,1} + ... + g_t s_{i,t}, g_1 v_{i,1} + ... + g_t v_{i,t}), \tag{22}$$

where each $s_{i,j}, v_{i,j}$ is of the form $d_z x_z$ or $d_z y_z$ for some z that depends only on i, j. Note also that

$$s^2_{i,1} + ... + s^2_{i,t} = v^2_{i,1} + ... + v^2_{i,t}. \tag{23}$$

The latter inequality comes from the fact that, by 21, both coordinates of $w_{\{x,y\}}$ have the same distribution.

Let us denote $s_i = (s_{i,1}, ..., s_{i,t})$, $v_i = (v_{i,1}, ..., v_{i,t})$ for $i = 1, ..., k$. We need the following lemma stating that with high probability vectors $s_1, ..., s_k, v_1, ..., v_k$ are close to be pairwise orthogonal.

Lemma 7.4. *Let us assume that $\mathcal{E}_f$ holds. Let $f(n)$ be an arbitrary positive function. Then for every $a > 0$ with probability at least $\mathbb{P}_{succ} \geq 1 - 4\binom{k}{2}e^{-\frac{2a^2 n}{f^4(n)}}$, taken under coin tosses used to construct $\mathcal{D}$, the following is true for every $1 \leq i_1 \neq i_2 \leq k$:*

$$|\sum_{u=1}^{n} s_{i_1,u}v_{i_1,u}| \leq a\chi(\mathcal{P}) + \Psi\frac{f^2(n)}{n},$$

$$|\sum_{u=1}^{n} s_{i_1,u}s_{i_2,u}| \leq a\chi(\mathcal{P}) + \Psi\frac{f^2(n)}{n},$$

$$|\sum_{u=1}^{n} v_{i_1,u}v_{i_2,u}| \leq a\chi(\mathcal{P}) + \Psi\frac{f^2(n)}{n},$$

$$|\sum_{u=1}^{n} s_{i_1,u}v_{i_2,u}| \leq a\chi(\mathcal{P}) + \Psi\frac{f^2(n)}{n}.$$

Proof. Note that the we get the first inequality for free from the fact that x is orthogonal to y (in other words, $\sum_{u=1}^{n} s_{i_1,u}v_{i_1,u}$ can be represented as $C\sum_{u=1}^{n} x_i y_i$ and the latter expression is clearly 0). Let us consider now one of the three remaining expressions. Note that they can be rewritten as:

$$E = \sum_{i=1}^{n} d_{\rho(i)}d_{\lambda(i)}x_{\zeta(i)}x_{\gamma(i)} \tag{24}$$

or

$$E = \sum_{i=1}^{n} d_{\rho(i)}d_{\lambda(i)}y_{\zeta(i)}y_{\gamma(i)} \tag{25}$$

or

$$E = \sum_{i=1}^{n} d_{\rho(i)}d_{\lambda(i)}x_{\zeta(i)}y_{\gamma(i)} \tag{26}$$

for some $\rho, \lambda, \zeta, \gamma$. Note also that from the Ψ-regularity condition we immediately obtain that $\rho(i) = \lambda(i)$ for at most Ψ elements of each sum. Get rid of these elements from each sum and consider the remaining ones. From the definition of the $\mathcal{P}$-chromatic number, those remaining ones can be partitioned into at most $\chi(\mathcal{P})$ parts, each consisting of elements that are independent random variables (since in the corresponding graph there are no edges between them). Thus, for the sum corresponding to each part one can apply Lemma 7.2. Thus one can conclude that the sum differs from its expectation (which clearly is 0 since $E(d_i d_j) = 0$ for $i \neq j$) by a with probability at most:

$$\mathbb{P}_a \leq 2e^{-\frac{2a^2}{\sum_{i=1}^{n} x_{\zeta(i)}x_{\gamma(i)}}}, \tag{27}$$

or

$$\mathbb{P}_a \leq 2e^{-\frac{2a^2}{\sum_{i=1}^{n} y_{\zeta(i)}y_{\gamma(i)}}}, \tag{28}$$

or

$$\mathbb{P}_a \leq 2e^{-\frac{2a^2}{\sum_{i=1}^{n} x_{\zeta(i)}y_{\gamma(i)}}}. \tag{29}$$

Now it is time to use the fact that event $\mathcal{E}_f$ holds. Then we know that: $|x_i|, |y_i| \leq \frac{f(n)}{\sqrt{n}}$ for $i = 1, ..., n$. Substituting this upper bound for $|x_i|, |y_i|$ in the derived expressions on the probabilities coming from Lemma 7.2, and then taking the union bound, we complete the proof. □

We can finish the proof of Theorem 4.1. From Lemma 7.4 we see that $s_1, ..., s_k, v_1, ..., v_k$ are close to pairwise orthogonal with high probability. Let us fix some positive function $f(n) > 0$ and some $a > 0$. Denote

$$\Delta = a\chi(\mathcal{P}) + \Psi\frac{f^2(n)}{n}. \tag{30}$$

Note that , by Lemma 7.4 we see that applying Gram-Schmidt process we can obtain a system of pairwise orthogonal vectors $\tilde{s}_1, ..., \tilde{s}_k, \tilde{v}_1, ..., \tilde{v}_k$ such that

$$\|\tilde{v}_i - v_i\|_2 \leq k\Delta. \tag{31}$$

and

$$\|\tilde{s}_i - s_i\|_2 \leq k\Delta. \tag{32}$$

Let us consider again $w_{x,y}$. Replacing s_i by $\tilde{s}_i$ and v_i by $\tilde{v}_i$ in the formula on $w_{x,y}$, we obtain another Gaussian vector: $\tilde{w}_{x,y}$ for each row i of the matrix $\mathcal{P}$. Note however that vectors $\tilde{w}_{x,y}$ have one crucial advantage over vectors $w_{x,y}$, namely they are independent. That comes from the fact that $\tilde{s}_1, ..., \tilde{s}_k, \tilde{v}_1, ..., \tilde{v}_k$ are pairwise orthogonal. Note also that from 37 and 38 we obtain that the angular distance between $w_{x,y}$ and $\tilde{w}_{x,y}$ is at most $k\Delta$.

Let Z_i for $i = 1, ...k$ be an indicator random variable that is zero if $\tilde{w}_{x,y}$ is inside the region $\mathcal{U}_{p,r}$ and zero otherwise. Let U_i for $i = 1, ...k$ be an indicator random variable that is zero if $w_{x,y}$ is inside the region $\mathcal{U}_{p,r}$ and zero otherwise. Note that $\tilde{\theta}^n_{p,r} = \frac{U_1 + ... + U_k}{k}$. Furthermore, random variables $Z_1, ..., Z_k, U_1, ..., U_k$ satisfy the assumptions of Lemma 7.1 with $\mu \leq \frac{8\tau}{\theta_{p,r}}$, where $\tau = k\Delta$. Indeed, random variables Z_i are independent since vectors $\tilde{w}_{x,y}$ are independent. From what we have said so far we know that each of them takes value one with probability exactly $\frac{\theta_{p,r}}{\pi}$. Furthermore $Z_i \neq U_i$ only if $w_{x,y}$ is inside $\mathcal{U}_{p,r}$ and $\tilde{w}_{x,y}$ is outside $\mathcal{U}_{p,r}$ or vice versa. The latter event implies (thus it is included in the event) that $w_{x,y}$ is near the border of the region $\mathcal{U}_{p,r}$, namely within an angular distance $\frac{\epsilon}{\theta_{p,r}}$ from one of the four semi-lines defining $\mathcal{U}_{p,r}$. Thus in particular, an event $Z_i \neq U_i$ is contained in the event of probability at most $2 \cdot 4 \cdot \frac{\epsilon}{\theta_{p,r}}$ that depends only on one $w_{x,y}$.

But then we can apply Lemma 7.1. All we need is to assume that the premises of Lemma 7.4 are satisfied. But this is the case with probability specified in Lemma 7.3 and this probability is taken under random coin tosses used to product $\mathcal{H}$ and $\mathcal{R}$, thus independently from the random coin tosses used to produce $\mathcal{D}$. Putting it all together we obtain the statement of Theorem 4.1.

8. Proof of Theorem 4.2

We will borrow some notation from the proof of Theorem 4.1. Note however that in this setting no preprocessing with the use of matrices $\mathcal{H}$ and $\mathcal{R}$ is applied.

Lemma 8.1. *Define $U_1,...,U_k$ as in the proof of Theorem 4.1. Assume that the following is true:*

$$|\sum_{u=1}^{n} s_{i_1,u}v_{i_1,u}| \leq \Delta,$$

$$|\sum_{u=1}^{n} s_{i_1,u}s_{i_2,u}| \leq \Delta,$$

$$|\sum_{u=1}^{n} v_{i_1,u}v_{i_2,u}| \leq \Delta,$$

$$|\sum_{u=1}^{n} s_{i_1,u}v_{i_2,u}| \leq \Delta.$$

for some $0 < \Delta < 1$. The the following is true for every fixed $1 \leq i < j \leq k$:

$$|\mathbb{P}(U_iU_j = 1) - \mathbb{P}(U_i = 1)\mathbb{P}(U_j = 1)| = O(\Delta).$$

The lemma follows from the exactly the same analysis that was done in the last section of the proof of Theorem 4.1 thus we leave it to the reader as an exercise.

Note that we have:

$$\begin{aligned} Var(\tilde{\theta}^n_{p,r}) &= Var(\frac{U_1+...+U_k}{k}) \\ &= \frac{1}{k^2}(\sum_{i=1}^{k} Var(U_i) + \sum_{i \neq j} Cov(U_i,U_j)). \end{aligned} \quad (33)$$

Since U_i is an indicator random variable that takes value one with probability $\frac{\theta_{p,r}}{\pi}$, we get:

$$Var(U_i) = E(U_i^2) - E(U_i)^2 = \frac{\theta_{p,r}}{\pi}(1 - \frac{\theta_{p,r}}{\pi}). \quad (34)$$

Thus we have:

$$Var(\tilde{\theta}^n_{p,r}) = \frac{1}{k}\frac{\theta_{p,r}(\pi - \theta_{p,r})}{\pi^2} + \frac{1}{k^2}\sum_{i \neq j} Cov(U_i,U_j). \quad (35)$$

Note however that $Cov(U_i,U_j)$ is exactly: $\mathbb{P}(U_iU_j = 1) - \mathbb{P}(U_i = 1)\mathbb{P}(U_j = 1)$.

Therefore, using Lemma 8.1, we obtain:

$$Var(\tilde{\theta}^n_{p,r}) = \frac{1}{k}\frac{\theta_{p,r}(\pi - \theta_{p,r})}{\pi^2} + O(\Delta). \quad (36)$$

It suffices to estimate parameter Δ. We proceed as in the previous proof. We only need to be a little bit more cautious since the condition: $|x_i|,|y_i| \leq \frac{f(n)}{\sqrt{n}}$ cannot be assumed right now. We select two rows: i_1,i_2 of $\mathcal{P}$. Note that again we see that applying Gram-Schmidt process, we can obtain a system of pairwise orthogonal vectors $\tilde{s}_{i_1},\tilde{s}_{i_i},\tilde{v}_{i_i},\tilde{v}_{i_2}$ such that

$$\|\tilde{v}_{i_1} - v_{i_2}\|_2 \leq \Delta, \quad (37)$$

and

$$\|\tilde{s}_{i_1} - s_{i_2}\|_2 \leq \Delta. \quad (38)$$

The fact that right now the above upper bounds are not multiplied by k, as it was the case in the previous proof, plays key role in obtaining nontrivial concentration results even when no Hadamard mechanism is applied.

We consider the related sums: $E_1 = \sum_{i=1}^{n} d_{\rho(i)}d_{\lambda(i)}x_{\zeta(i)}x_{\gamma(i)}, E_2 = \sum_{i=1}^{n} d_{\rho(i)}d_{\lambda(i)}y_{\zeta(i)}y_{\gamma(i)}, E_3 = \sum_{i=1}^{n} d_{\rho(i)}d_{\lambda(i)}x_{\zeta(i)}y_{\gamma(i)}$ as before. We can again partition each sum into at most $\chi(\mathcal{P})$ sub-chunks, where this time $\chi(\mathcal{P}) \leq 3$ (since $\mathcal{P}$ is Toeplitz Gaussian). The problem is that applying Lemma 7.2, we get bounds that depend on the expressions of the form

$$\alpha_{x,i} = \sum_{j=1}^{n} x_j^2 x_{j+i}^2, \quad (39)$$

and

$$\alpha_{y,i} = \sum_{j=1}^{n} y_j^2 y_{j+i}^2, \quad (40)$$

where indices are added modulo n and this time we cannot assume that all $|x_i|,|y_i|$ are small. Fortunately we have:

$$\sum_{i=1}^{n} \alpha_{x,i} = 1, \quad (41)$$

and

$$\sum_{i=1}^{n} \alpha_{y,i} = 1 \quad (42)$$

Let us fix some positive function $f(k)$. We can conclude that the number of variables $\alpha_{x,i}$ such that $\alpha_{x,i} \geq \frac{f(k)}{\binom{k}{2}}$ is at most $\frac{\binom{k}{2}}{f(k)}$. Note that each such $\alpha_{x,i}$ and each such $\alpha_{y,i}$ corresponds to a pair $\{i_{1,2}\}$ of rows of the matrix $\mathcal{P}$ and consequently to the unique element $Cov(U_{i_1},U_{i_2})$ of the entire covariance sum (scaled by $\frac{1}{k^2}$). Since trivially we have $|Cov(U_{i_1},U_{i_2})| = O(1)$, we conclude that the contribution of these elements to the entire covariance sum is of order $\frac{1}{f(k)}$. Let us now consider these $\alpha_{x,i}$ and $\alpha_{y,i}$

Table 3: Mean and std of the train error versus the size of the hash (k) / size of the reduction (n/k) for the network.

k / $\frac{n}{k}$	Circulant [%]	Random [%]	BinPerm [%]	BinCirc [%]	HalfShift [%]	Toeplitz [%]	VerHorShift [%]
1024 / 1	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.30 ± 0.44	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
512 / 2	0.04 ± 0.06	0.00 ± 0.00	0.00 ± 0.00	2.66 ± 2.98	1.44 ± 2.89	0.00 ± 0.00	0.00 ± 0.01
256 / 4	6.46 ± 2.27	0.00 ± 0.00	0.79 ± 1.57	0.60 ± 1.19	0.49 ± 0.93	2.09 ± 1.69	3.98 ± 3.96
128 / 8	16.89 ± 6.57	4.69 ± 0.43	4.44 ± 0.50	5.62 ± 1.03	7.34 ± 1.27	11.82 ± 2.17	10.51 ± 1.27
64 / 16	26.47 ± 0.98	13.35 ± 0.61	23.98 ± 11.54	18.68 ± 0.78	17.64 ± 2.01	29.97 ± 5.29	18.68 ± 3.26
32 / 32	40.79 ± 3.82	27.51 ± 2.04	28.28 ± 3.23	33.91 ± 3.23	27.90 ± 1.05	41.49 ± 2.14	43.51 ± 3.78
16 / 64	63.96 ± 5.62	46.31 ± 0.73	50.03 ± 6.18	58.71 ± 6.96	54.88 ± 6.47	57.72 ± 3.42	60.91 ± 4.53

that are at most $\frac{f(k)}{\binom{k}{2}}$. These sums are small (if we take $f(k) = o(k^2)$) and thus it makes sense to apply Lemma 7.2 to them. That gives us upper bound $a = \Delta$ with probability:

$$\mathbb{P}^* \geq 1 - e^{-\Omega(a^2 \frac{k^2}{f(k)})}. \quad (43)$$

Taking $f(k) = (\frac{k^2}{\log(k)})^{\frac{1}{3}}$ and $a = \Delta = \frac{1}{f(k)}$, we conclude that:

$$Var(\tilde{\theta}^n_{p,r}) \leq \frac{1}{k}\frac{\theta_{p,r}(\pi - \theta_{p,r})}{\pi^2} + (\frac{\log(k)}{k^2})^{\frac{1}{3}} \quad (44)$$

Thus, from the Chebyshev's inequality, we get the following for every $c > 0$ and fixed points p, r:

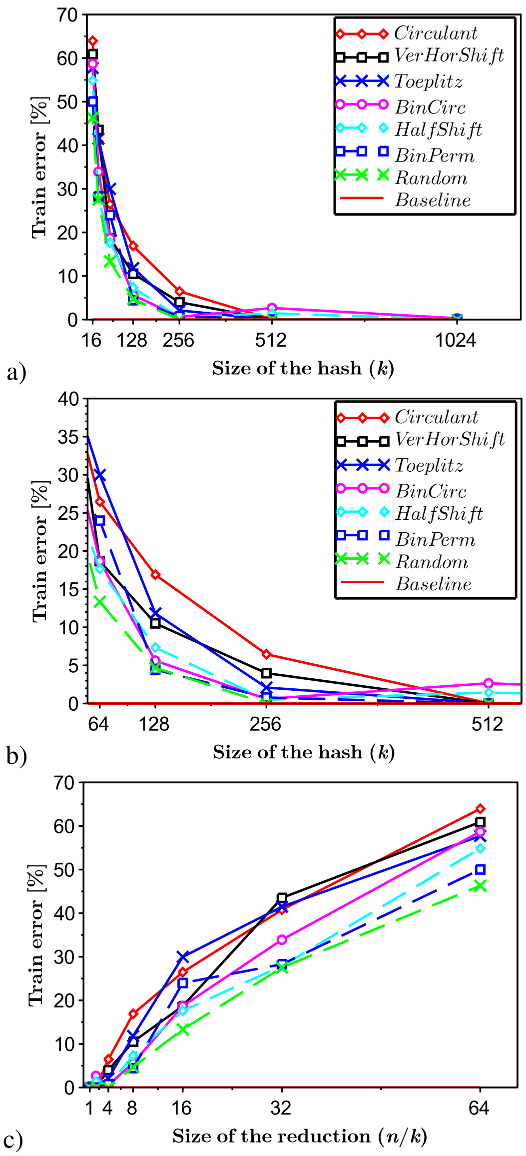
$$\mathbb{P}(|\tilde{\theta}^n_{p,r} - \frac{\theta_{p,r}}{\pi}| \geq c(\frac{\sqrt{\log(k)}}{k})^{\frac{1}{3}}) = O(\frac{1}{c^2}). \quad (45)$$

That completes the proof of Theorem 4.2.

9. Additional figures

Figure 9a and Figure 9b show how the mean train error is affected by the size of the hash, and Figure 9c shows how the mean train error changes with the size of the reduction for the neural network experiment. In Table 3 we report both the mean and the standard deviation of the train error across our neural network experiments. *Baseline* refers to the network with one hidden layer containing 100 hidden units, where all parameters are trained.

Figure 10a shows the original version of Figure 7a (before zoom). Figure 10b shows the original version of Figure 8a (before zoom). Finally, Table 4 shows the mean and the standard deviation of the test error versus the size of the hash (k)/size of the reduction (n/k) for 1-NN.



a) b) c)

Figure 9: Mean train error versus a), b) the size of the hash (k), c) the size of the reduction (n/k) for the network. b) is a zoomed a). Baseline corresponds to 0%.

Table 4: Mean and std of the test error versus the size of the hash (k) / size of the reduction (n/k) for 1-NN.

k / $\frac{n}{k}$	Circulant [%]	Random [%]	BinPerm [%]	BinCirc [%]	HalfShift [%]	Toeplitz [%]	VerHorShift [%]
1024 / 1	6.02 ± 0.64	4.83 ± 0.19	6.67 ± 0.65	12.77 ± 2.86	6.38 ± 0.44	6.22 ± 1.20	6.30 ± 0.76
512 / 2	12.98 ± 11.29	5.77 ± 0.11	8.15 ± 0.56	12.40 ± 2.32	7.25 ± 0.71	9.11 ± 2.28	10.81 ± 4.31
256 / 4	17.73 ± 6.66	8.51 ± 0.35	11.11 ± 1.15	12.13 ± 4.35	12.05 ± 2.94	15.66 ± 3.36	18.19 ± 5.46
128 / 8	34.80 ± 14.59	14.44 ± 0.89	17.20 ± 2.26	22.15 ± 6.45	24.74 ± 8.14	33.90 ± 13.90	30.37 ± 7.52
64 / 16	45.91 ± 5.50	27.57 ± 1.58	29.53 ± 3.40	35.33 ± 5.58	36.58 ± 10.71	51.10 ± 13.98	41.66 ± 8.08
32 / 32	65.06 ± 9.60	40.58 ± 2.49	43.58 ± 4.66	53.05 ± 5.39	47.18 ± 7.19	58.24 ± 8.87	56.73 ± 6.09
16 / 64	68.61 ± 5.72	58.72 ± 3.08	60.30 ± 6.11	66.29 ± 4.79	60.84 ± 5.31	72.50 ± 6.04	72.50 ± 5.91

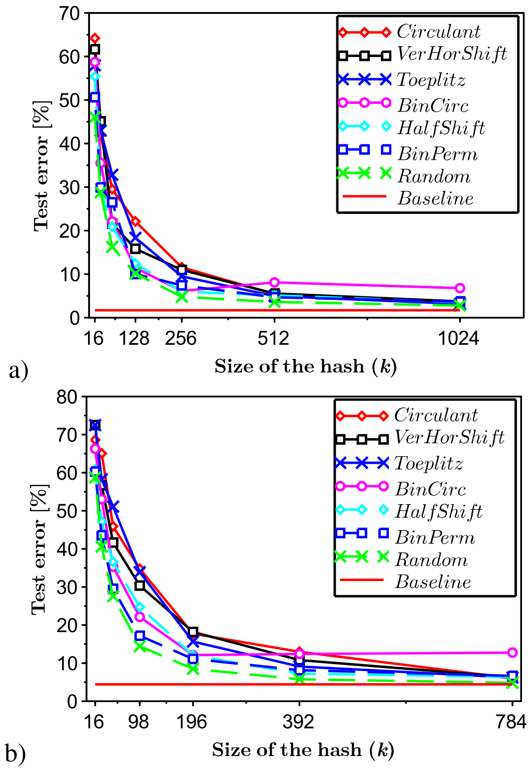


Figure 10: Mean test error versus the size of the hash (k) (original plot) for a) the network, b) 1-NN.