Edo Liberty

Accelerated Dense Random Projections

CONTENTS

1. LIST OF COMMON NOTATIONS

d	Incoming vectors' Original dimension
$\mathbb{R}^d$	Real d dimensional space
x or x_i	Incoming vector $\in \mathbb{R}^d$
n	Number of incoming vectors or a constant polynomial in that number
k	Target dimension
ε	Constant required precision, $0 < \varepsilon < 1/2$
δ	A constant larger then zero
$\mathbb{S}^{d-1}$	The set $\{x \in \mathbb{R}^d \mid \|x\|_2 = 1\}$
$\|\cdot\|$	The ℓ_2 norm for vectors and the Spectral norm for matrices
$\|\cdot\|_p$	ℓ_p norm (for vectors)
$\|\cdot\|_{p \to q}$	Operator norm from ℓ_p to ℓ_q (for matrices)
H, H_d	A $d \times d$ Walsh Hadamard transform
± 1	$\{+1, -1\}$
χ	A subset of $\mathbb{R}^d$
FJLT	The fast JL transform by Ailon Chazelle [1]
FJLTr	A revised version of the FJLT algorithm, Chapter 3 and [2]
FWI	A two stage projection process, chapter 5 and [2]

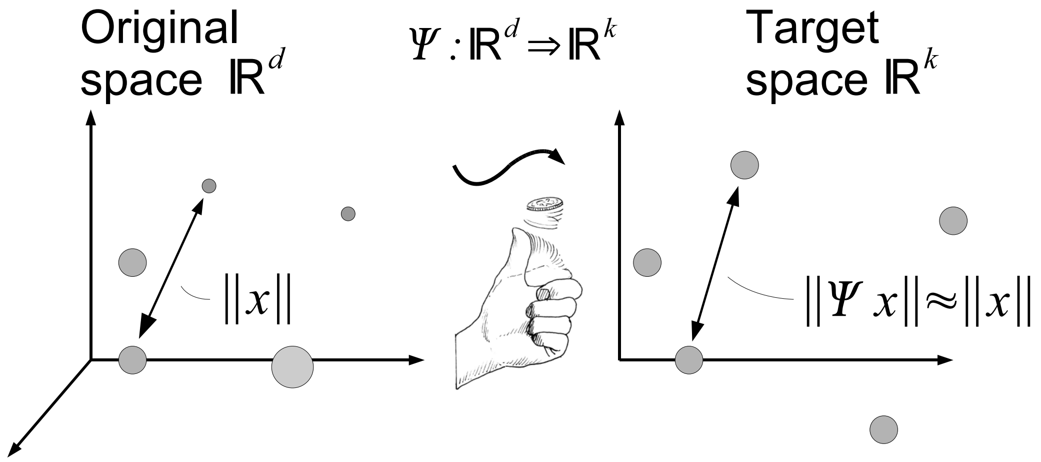
ABSTRACT

In dimensionality reduction, a set of points in $\mathbb{R}^d$ is mapped into $\mathbb{R}^k$ such that all distances between points are approximately preserved although the target dimension k is smaller then the original dimension d. One method for achieving this, which has lately received much attention, is called Random Projections. The purpose of this thesis is twofold. One, to drive at the theoretical roots of this phenomenon. Second, to purpose improved algorithmic constructions that achieve it. Our results divide into two main paradigms. For the case where k is significantly smaller then d ($k \in o(\text{poly}(d))$) we gain an improvement by designing more efficient algorithms for applying certain linear algebraic transforms. For the case where k is only moderately smaller then d ($k \in \omega(d^{1/3})$) we propose a new construction and prove its correctness using a new framework which is formally defined. Our results are shown to be impossible to achieve using existing approaches. We supplement our theoretical work with relevant experimental results which demonstrate its practicality.

2. INTRODUCTION TO RANDOM PROJECTIONS

2.1 Linear embedding and the JL property

In many applications one is given a set of n points in some high dimension, say d, and is interested in embedding these points into a space of lower dimension, k, such that all distances are preserved almost exactly.


Original space $\mathbb{R}^d$
$\Psi : \mathbb{R}^d \Rightarrow \mathbb{R}^k$
Target space $\mathbb{R}^k$
$\|x\|$
$\|\Psi x\| \approx \|x\|$

More precisely, given n vectors $\{x_1, \ldots, x_n\}$ in $\mathbb{R}^d$ we are interested in a mapping $\Psi : \mathbb{R}^d \to \mathbb{R}^k$ such that $k \ll d$ and

$$\forall i, j \;\; (1 - \varepsilon)\|\mathbf{x}_i - \mathbf{x}_j\| \le \|\Psi(\mathbf{x}_i) - \Psi(\mathbf{x}_j)\| \le (1 + \varepsilon)\|\mathbf{x}_i - \mathbf{x}_j\| \tag{2.1}$$

for a constant $0 \le \varepsilon \le 1/2$.

Naturally, one might approach this task in a deterministic fashion by evaluating the incoming vectors. However, it has been well known that a randomized approach to this problem is significantly easier. Johnson and Lindenstrauss in [3] were the first to give such a randomized construction and the final step in their proof is still common to all random projection schemes; Let Ψ be a linear mapping (a $k \times d$ matrix) chosen from a probability distribution $\mathbb{D}_{k,d}$ such that the length of any vector $x \in \mathbb{R}^d$ is ε preserved with very high probability. Let the vector x be the difference $x_i - x_j$. Since Ψ is a linear operator we have that $\Psi(\mathbf{x}_i) - \Psi(\mathbf{x}_j) = \Psi(\mathbf{x}_i - \mathbf{x}_j) = \Psi(\mathbf{x})$. Moreover, since there are $\binom{n}{2} < n^2$ pairwise distances, if we ε preserve the length of each vector x with probability larger then $1 - \frac{1}{2n^2}$ then, by the union bound, the entire metric is ε preserved with probability at least 1/2. Without loss of generality, we can consider only unit vectors, $\|x\|_2 = 1$. The notation $\mathbb{S}^{d-1}$ stands for the $d-1$ dimensional sphere, i.e the set $\{\mathbf{x} \in \mathbb{R}^d \mid \|x\|_2 = 1\}$.

We have that if

$$\forall \mathbf{x} \in \mathbb{S}^{d-1} \quad \Pr_{\Psi \sim \mathbb{D}_{k,d}} [||\|\Psi \mathbf{x}\| - 1| > \varepsilon] \leq \frac{1}{2n^2} \tag{2.2}$$

Then for every i and j simultaneously

$$\forall i, j \ \ (1-\varepsilon)\|\mathbf{x}_i - \mathbf{x}_j\| \leq \|\Psi(\mathbf{x}_i) - \Psi(\mathbf{x}_j)\| \leq (1+\varepsilon)\|\mathbf{x}_i - \mathbf{x}_j\| \tag{2.3}$$

with probability at least 1/2.

The surprising fact about the JL (Johnson Lindenstrauss) lemma is that this can be achieved by a matrix who's size is logarithmic in n. To see this we define the JL property.

Definition 2.1.1. *A distribution $\mathbb{D}_{k,d}$ over $k \times d$ matrices exhibits the JL property (JLP) if*

$$\forall \mathbf{x} \in \mathbb{S}^{d-1} \quad \Pr_{\Psi \sim \mathbb{D}_{k,d}} [||\|\Psi \mathbf{x}\| - 1| > \varepsilon] \leq c_1 e^{-c_2 k \varepsilon^2} \tag{2.4}$$

For some constants c_1 and c_2.

Given such a distribution, we satisfy equation 2.2 for $k = \Omega(\log(n)/\varepsilon^2)$.

$$\begin{aligned} c_1 e^{-c_2 k \varepsilon^2} &\leq \frac{1}{2n^2} \\ k &= \Omega(\log(n)/\varepsilon^2) \end{aligned}$$

Johnson and Lindenstrauss showed that the uniform distribution over $k \times d$ projection matrices exhibits the *JL* property. This means that *a*ny set of n points in $\mathbb{R}^d$ (equipped with the ℓ_2 metric) can be embedded into dimension $k = O(\log(n)/\varepsilon^2)$ with distortion ε using a randomly generated linear mapping Ψ. Moreover Noga Alon [4] showed that this result is essentially tight (in its dependence on n).

It is remarkable to notice that the target dimension k is independent of the original dimension d. Further, the randomized algorithm which achieves this is independent of the input vectors $\{x_1, \ldots, x_n\}$, (depends only on their number). These properties make random projections a critical ingredient in many algorithms. Examples for such application can by found in: approximate nearest neighbor searching [5, 6, 7, 8, 9], learning [10, 11, 12], matrix low rank approximation [13, 14, 15, 16, 17, 18, 19, 20], others linear algebraic operations [21, 22, 23, 24], and many other algorithms and applications, e.g, [25, 26, 27, 28, 29, 30].

2.2 Classic results, review of known JL distributions

The construction of Johnson and Lindenstrauss is simple. They proposed to choose Ψ uniformly at random from the space of projection matrices. In other words, Ψ contains a random k dimensional subspace of $\mathbb{R}^d$. One technique to sample from this distribution is to start with a $k \times d$ matrix whose entries are chosen i.i.d gaussian. Then, orthogonalize and normalize its rows (Using Grahm-Schmidt for example). The idea of the proof was as follows; since the distribution is rotational invariant, projecting any *fixed* vector x onto a random subspace is equivalent (statistically) to projecting a *random* vector in $\mathbb{R}^d$ onto the first k vectors of the canonical basis. Johnson and Lindenstrauss proceed to give the relevant concentration over $\mathbb{S}^{d-1}$ which proves the lemma.

Their proof, however, can be made significantly simpler by considering a slightly modified construction. Gupta and Dasgupta [31] as well as Frankl and Maehara [32] suggested that each entry in Ψ be chosen uniformly at random from a Gaussian distribution (without orthogonalization). These proofs still relay on the rotational invariance of the distribution but are significantly easier. A sketch of a possible proof is given below.[1]

Since the distribution of Ψ is the same as the that of $\Psi' = \Psi U$ for any complete orthogonal transformation U. We look at $\Psi U U^T x$ for U such that $U\mathbf{x} = e_1 = (1, 0, \ldots, 0)^T$ (w.l.o.g $\|\mathbf{x}\| = 1$). Since $\|\Psi' e_1\|$ is the norm of $\Psi'^{(1)}$, the first column of Ψ', and Ψ' distributes like Ψ, we need only show that $\Pr[\big|\|\Psi^{(1)}\| - 1\big| > \varepsilon] \leq$

[1] This proof is not the one supplied in [31] or in [32]. The authors encountered this idea in several informal discussions but are not aware of its specific origin.

$c_1 e^{-c_2 k \varepsilon^2}$. In other words, because the distribution is rotationally invariant the probability for failure (large distortion) is equal for all vectors in $\mathbb{R}^d$. We therefor choose to calculate it for the vector $(1, 0, \ldots, 0)^T$ which is trivial given the tail properties of the χ-square distribution.

More difficult to prove are cases where the distribution is not rotationally invariant. The first such construction was given by Dimitris Achlioptas [33] who proposed a matrix Ψ such that $\Psi(i, j) \in \{-1, 0, 1\}$ with constant probabilities. Matousek [34] extended this result to any i.i.d sub-gaussian variables. These proofs relay on slightly weaker condition which is the independence of the rows of Ψ.

Denote by $\Psi_{(i)}$ the i'th row of Ψ, $\|\Psi \mathbf{x}\|^2 = \sum_{i=1}^{k} \langle \Psi_{(i)}, \mathbf{x} \rangle^2$. We notice that if the rows of $\Psi_{(i)}$ are i.i.d then $\langle \Psi_{(i)}, \mathbf{x} \rangle^2$ are also i.i.d. By characterizing the distribution of the variables $\langle \Psi_{(i)}, \mathbf{x} \rangle^2$ one can derive a concentration result using a quantitative version of the central limit theorem. We review such a result by Matousek [34] further into the introduction.

2.3 Motivation for accelerating random projections

In many of the applications mentioned in section 2.1 the original dimension d is very large whereas the target dimension in manageable in size. An implementation of any of the constructions described in the last section requires $O(kd)$ bits to store and $O(kd)$ operations to apply to each input vector. This is, in many cases, impractical. For example, if the incoming vectors are the grey scale values of a $5MP$ images, and the target dimension is 1000, the matrix Ψ will occupy $20G$ of space/memory (in 4 byte float precision).[2] This amount of memory use is (as of today) extremely inconvenient. On most modern machines it will invoke intensive paging and thus perform very poorly.

In some situation, one might be able to avoid storing the matrix Ψ altogether. Mainly, when the task is to embed only a given fixed set of points. In this case, one can generate each row of Ψ, apply it to all the points, and discard it. This generate-and-forget method can not be used in the Nearest Neighbor setup for example because query points must be projected using the same matrix as the data points and are not

[2] It is worth mentioning that Achlioptas's matrix will only occupy 1/32 of this amount since each entry is representable by 1 bit instead of a 32 bit float.

known in advance. Nevertheless, this idea is used in practice quite intensively mainly due to it's simplicity.

Regardless of space usage and handling, applying these matrices is prohibitive from the time stand point. For the described modest application even when the matrix fits in memory the running time on a $3Ghz$ processor will be in the minutes, for *each* projected vector. This prevents this method from being used in any real time system or on any moderately large data sets. If random projection is to be used in practice one must come up with faster, more efficient, ways of accomplishing it.

2.4 Sparse Projective matrices

The first direction worth exploring in order to accelerate the projection process is to sparsify Ψ. Applying a matrix to any vector requires as many operations as the number of nonzeros in the matrix. If the i.i.d entries in Ψ are very likely to be zero then the number of non-zeros in Ψ should be much less then kd. However intuitive this idea is, it can be shown that unless the number of nonzeros in Ψ is $O(kd)$ the same success probability cannot be achieved. One way to see this is to consider projecting the vector $[1, 0, \ldots, 0]^T$. The resulting quantity $\|\Psi x\|_2$ is exactly the norm of the first column of Ψ. Hance, the norm of the first column of Ψ must statistically concentrate around 1, i.e, $|\sum_{i=1}^{k} \Psi(i, 1)^2 - 1| \leq \varepsilon$ with probability at least $1 - 1/n$. Since any event with probability $1 - 1/n \leq \Pr < 1$ must relay on at least $\log(n)$ random bits we get that any column of Ψ contains $O(\log(n))$ nonzeros[3]. Since we view ε as a constant we get that each column contains also $O(k)$ nonzeros. Therefore, the entire matrix contains $O(kd)$ nonzeros and our hope for sparse projection matrices is shattered.

As we saw, sparse matrices cannot exhibit the JL property. We showed this by considering vectors like $x = [1, 0, \ldots, 0]^T$. Notice however that these horribly sparse vectors are actually very few. By "few" we mean that a very small portion of $\mathbb{S}^{d-1}$ is occupied by sparse vectors. We can therefore ask: can sparse projective matrices preserve lengths of non-sparse vectors? This question was asked and answered by Ailon and Chazelle in [1] which proposed the FJLT algorithm. It is worth mentioning that this result was the first to give an asymptotical acceleration of a JL transform. Their work gave theoretical insights and motivations

[3] Under the assumption that producing each $\Psi(i, j)$ requires a constant number of random bits

the can be seen throughout this document. They showed that if the ℓ_∞ of the input vector x is bounded by $\sqrt{k/d}$ then Ψ can (in expectancy) contain only $O(k^3)$ nonzeros. This result was then generalized by Matousek [34] who showed that any vector x such that $\|x\|_\infty \leq \eta$ can be projected with ε distortion and high probability by a matrix containing only $O(k^2\eta^2 d)$ non-zeros in expectancy. Ailon and Chazelle further showed that any vector in $x \in \mathbb{R}^d$ can be isometrically, randomly, rotated such that their $\|\Phi x\|_\infty \leq \sqrt{k/d}$. Here Φ is a fast randomized isometry which takes only $O(d\log(d))$ operations to apply. For completeness we show a sketch of the proof by Matousek [34].

The idea is as follows, when we consider the term $\|\Psi x\|_2^2$ we can look at it as $\sum y^2(i)$ where $y_i = \langle \Psi_{(i)}, x\rangle$ and $\Psi_{(i)}$ denotes the i'th row of Ψ. Since the entries of Ψ are i.i.d, so are the random variables y_i. We need only show that the sum of k such variables concentrates in the right way. It turns out that it is sufficient for y_i to be distribute s.t $E[y_i] = 0$, $Var[y_i] = k^{-1/2}$ and y_i have a uniform sub-gaussian tail. the definitions are given below.

Definition 2.4.1 (Matousek [34])**.** *A real random variable Y is said to have a sub-gaussian upper tail if for some α*

$$\Pr(Y > t) \leq E[e^{-\alpha t^2}] \tag{2.5}$$

for all $t > 0$. If this condition holds only up to some $t \leq t_0$, the distribution of Y is said to have a sub-gaussian upper tail up to t_0. A collection of variables Y_i is said to have uniform *sub-gaussian upper tail if they are all sub-gaussian with the same constant α*

Lemma 2.4.1 (Matousek [34])**.** *Let Y_i be i.i.d random variables with $E[Y_i] = 0$, $Var[Y_i] = 1$ and Y_i have a uniform sub-gaussian tail up to at least $\sqrt{k}$. Define the random variable $Z = \frac{1}{\sqrt{k}}(\sum_{i=1}^{k} Y_i^2 - 1)$. The variable Z has a sub-gaussian tail up to at least $\sqrt{k}$.*

Proving lemma 2.4.1 turns out to be rather technical and it is given in full detail in [34]. However, given its correctness, the JL property follows almost directly. Let $Y_i = \sqrt{k} y_i$ where $y_i = \langle \Psi_i, x \rangle$. First notice that:

$$\begin{aligned} \|\Psi x\|_2^2 - 1 &= \sum_{i=1}^{k} y_i^2 - 1 && (2.6) \\ &= \frac{1}{k} \sum_{i=1}^{k} Y_i^2 - k && (2.7) \\ &= \frac{1}{\sqrt{k}} Z && (2.8) \end{aligned}$$

Where $Z = \frac{1}{\sqrt{k}}(\sum_{i=1}^{k} Y_i^2 - 1)$. Assume that $E[Y_i] = 0$, $Var[Y_i] = 1$ and Y_i exhibit a uniform sub-gaussian upper tail up to $\sqrt{k}$. According to lemma 2.4.1, Z is sub-gaussian up to $\sqrt{k}$. Which is used as follows:

$$\begin{aligned} \Pr[|\|\Psi x\|_2 - 1| > \epsilon] &\leq 2\Pr[\|\Psi x\|_2^2 - 1 > 2\epsilon] && (2.9) \\ &= 2\Pr[Z > 2\sqrt{k}\epsilon] && (2.10) \\ &\leq 2e^{-C(2\sqrt{k}\epsilon)^2} = c_1 e^{-c_2 k \epsilon^2} && (2.11) \end{aligned}$$

The last equation follows from the fact that Z is sub-gaussian up to $\sqrt{k}$ and $\varepsilon \leq 1/2$. This matches the success probability required by the JL property (definition 2.1.1).

A simple example can be random gaussian entries for $\Psi(i,j)$ (normalized be $1/\sqrt{k}$). Due to the rotational invariance of the gaussian distribution Y_i is itself distributed like a gaussian with mean zero and variance one. Clearly a gaussian has a sub-gaussian upper tail. This proves the JL lemma in yet another way.

Matousek further gives the connection between the ℓ_∞ of x and the required (expected) density of Ψ in order for Y_i to have the appropriate sub gaussian tails. We take $s(i)$ as independent copies of s such that

$$s = \begin{cases} +\frac{1}{\sqrt{q}} & \text{with probability } q/2 \\ -\frac{1}{\sqrt{q}} & \text{with probability } q/2 \\ 0 & \text{with probability} 1 - q \end{cases} \qquad (2.12)$$

Lemma 2.4.2 (Matousek [34]). *Let $\eta^2 \leq q$, let $x \in \mathbb{S}^{d-1}$ such that $\|x\|_\infty \leq \eta$, and let $Y = \sum_{i=1}^{d} s(i)x(i)$, where the $s(i)$ are as described above. Then Y has a sub-gaussian tail up to $\sqrt{2q}/\eta$.*

Combining lemmas 2.4.1 and 2.4.2 we obtain the minimal expected sparsity for Ψ required for vectors whose ℓ_∞ is bounded by η. Lemma 2.4.1 requires $\sqrt{2q}/\eta \geq \sqrt{k}$ and thus $q = \Theta(\eta^2 k)$. The expected number of non-zeros in Ψ is therefore $kdq = O(k^2\eta^2 d)$.

Putting these ideas together proves the following lemma

Lemma 2.4.3 (Matousek [34]). *Let $\eta \in [1/\sqrt{d}, 1]$ be a constant. Set q to be*

$$q = C_0 \eta^2 k \tag{2.13}$$

for some sufficiently large constant C_0. Let $\Psi(i,j)$ be i.i.d

$$\Psi(i,j) = \begin{cases} +\frac{1}{\sqrt{qk}} & \text{with probability } q/2 \\ -\frac{1}{\sqrt{qk}} & \text{with probability } q/2 \\ 0 & \text{with probability } 1-q \end{cases} \tag{2.14}$$

The lemma claims that:

$$\Pr\left[|\|\Psi x\| - 1|\right] \le c_1 e^{-c_2 k \varepsilon^2} \tag{2.15}$$

For all $x \in \mathbb{S}^{d-1}$ such that $\|x\|_\infty \le \eta$.

Furthermore, Matousek claims that the above bound for q is, up to a constant, tight.

We turn to discuss the FJLT algorithm. Notice that for n random vectors in $\mathbb{S}^{d-1}$, with constant probability, $\max_i \|x_i\|_\infty \le O(\sqrt{\log(n)/d})$ (assuming $n \ge d$). Also, the required expected number of non-zeros in Ψ is $O(kdq) = O(k^2\eta^2 d)$. n random vectors in $\mathbb{R}^d$ can, therefore, be projected using a matrix containing $O(k^3)$ entries (in expectancy). This is, of course, better than $O(kd)$ for any $k \in o(d^{1/2})$. The FJLT algorithm therefore begins with *isometrically* rotating all incoming vectors x_i such that the ℓ_∞ norms of the resulting (rotated) vectors are all bounded by $O(\sqrt{\log(n)/d})$ and then use a sparse projective matrix. The result is recaped below:

Lemma 2.4.4 (Ailon, Chazelle [1]). *Let $\Psi = PHD$ be chosen according to the following distribution:*

- *H: The Walsh Hadamard $d \times d$ Matrix (deterministic).*
- *D_s: A diagonal matrix whose diagonal is random ± 1 with probability $1/2$*
- *P: A k by d matrix whose i.i.d entries are either zero with probability $1-q$ or normally distributed according to $N(0, q^{-1})$ with probability q.*

where $q = \Theta(\varepsilon^3 \log^2(n) d^{-1})$.[4] *The distribution for Ψ exhibits the JL property. Moreover, applying Ψ to any*

[4] We assume that $q < 1$. if $q \ge 1$ we set $q = 1$ and the claim is trivial due to [32]

vector $x \in \mathbb{R}^d$ requires $O(d\log(d) + \min\{kd, \log^3(n)\varepsilon^{-2})\}$ operations in expectancy.

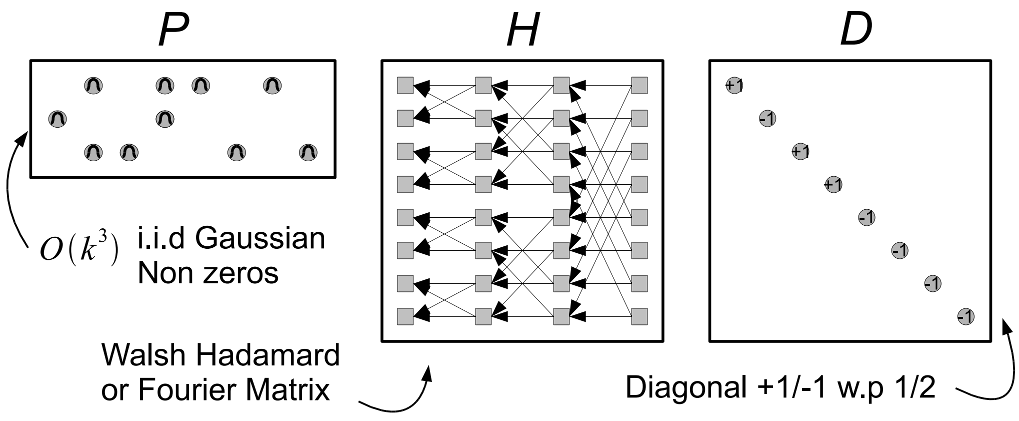


Fig. 2.1: A sketch of the FJLT construction. The FJLT algorithm was given in [1] and recaped in lemma 2.4.4.

Given the formalization above in order to prove this lemma we only need show that

$$\forall\, x \in \mathbb{S}^{d-1} \quad \Pr[\|HDx\|_\infty > \sqrt{log(n)/d}] \leq 1/n. \tag{2.16}$$

This can be easily seen by noticing that each $H(i,j) = \pm 1/\sqrt{d}$ and so $(HDx)(i) = \frac{1}{\sqrt{d}}\sum_{j=1}^{d} b(j)x(j)$ where $b(j)$ are ± 1 variables w.p 1/2 each. Using the Hoeffding bound and then a union bound for all coordinates and and all vectors yields the desired result.

Finally, the running time can be thought of as $O(d\log(d) + k^3)$. Where $d\log(d)$ operations are needed to perform the Walsh Hadamard transform and $O(k^3)$ to apply the sparse projective matrix P.[5]

2.5 Our contributions

2.5.1 Fast linear transforms

The first improvement we propose is a slight revision to the FJLT algorithm. We notice that when k is significantly smaller than d, the FJLT algorithm wastefully computes the entire Fourier of Hadamard transforms. This is inefficient since at most $O(k^3)$ coefficient are required which is potentially much less

[5] It is worth mentioning that this result holds also for mapping into $\mathbb{R}^k$ equipped with the ℓ_1 matric.

than d. We show simple algorithms which compute partial Walsh Hadamard and partial Fourier transforms in $O(d\log(k))$ operations instead of $O(d\log(d))$. This simple modification reduces the running of the $FJLT$ algorithm to $O(d\log(k)+k^3)$ which gives an improvement over the inefficient construction whenever $k \in o(\text{poly}(d))$. This will be discussed in detail in chapter 3.

Another interesting fact is the following. Given any $\log(d) \times d$ matrix over a finite alphabet A, one can apply A to any vector in $\mathbb{R}^d$ in $O(d)$ operations. Although similar results have been known for over fifty years our construction is new and slightly more general. We nickname our algorithm the mailman algorithm and we describe it in chapter **??**. Combining the mailman algorithm with Achlioptas's ±1 projection matrix gives an optimal $O(d)$ projection time for the case where $k \in O(\log(d))$.

2.5.2 A two stage framework

Inspired by the ideas in [1] and later [34] we consider linear dimensionality reduction as a two stage process. In the first stage, each vector $x \in \mathbb{S}^{d-1}$ is rotated isometrically in $\mathbb{R}^d$, using a random isometry Φ, such that with high probability Φx lays in some smaller subset of $\mathbb{S}^{d-1}$ which we denote by χ. A useful example discussed above for χ is the set of unit vectors such that $\|x\|_\infty \leq \eta$ for some constant η. In the second stage, the vectors Φx are projected into dimension k using a $k \times d$ matrix A. Our requirement from A is that

$$\forall x \in \chi \ \Pr\left[|\|AD_s x\|_2 - 1| \geq \varepsilon)\right] \leq 1/n \tag{2.17}$$

The matrix D_s is a diagonal random ±1 matrix. It is added so that A itself might be deterministic. In words, the requirement from A is that it (composed with D_s) projects vectors from χ into $\mathbb{R}^k$ with high probability and accuracy. There is no such guaranty for vectors not in χ. We say that χ is the probabilistic domain of A. Intuitively, this requirement is easier to fulfill for a smaller χ. However, mapping all vectors into a smaller χ (applying Φ) might require more randomness and computational effort. This intuitive notion seems to repeat itself.

Chapter 4 is dedicated to characterizing the relation between any matrix A and its corresponding probabilistic domain χ.

2.5.3 Dense fast projective matrices

The accelerations described thus far all relay on the spareness of the projection matrix A. Their proofs relied on the independence of A's entries. We propose to replace the choice of A from sparse and random i.i.d to *dense* and *deterministic*.

In chapter 5 we take A to be a ±1 four-wise independent matrix (definition 5.1.2). We show that one can apply both A and its appropriate Φ in time $O(d\log(k))$ for $k \in O(d^{1/2-\delta})$ for any positive constant δ. This matches the FJLTr algorithm when $k \in O((d\log(d))^{1/3})$ but outperforms it when $k \in O(d^{1/2-\delta})$ and $k \in \omega((d\log(d))^{1/3})$.

The second construction, described in chapter 6, is a dense orthogonal ±1 matrix which can be applied to any vector in $\mathbb{R}^d$ in linear time, i.e $O(d)$. We term these matrices *lean Walsh matrices*. Since the trivial lower bound on the running time of dimension reduction is $O(d)$ searching for projections which require this amount of time is worthwhile. We show that lean Walsh matrices are strictly better suited for this task then sparse projections. It is, however, not clear if there exists and appropriate random rotation which is also applicable in linear time.

2.5.4 Results summary

	Naïve or Slower	Faster then naïve	$O(d\log(k))$	Optimal, $O(d)$
k in $o(\log d)$	JL, FJLT		FJLTr, FWI	JL + Mailman
k in $\omega(\log d)$ and $o(\text{poly}(d))$	JL	FJLT	FJLTr, FWI	
k in $\Omega(\text{poly}(d))$ and $o((d\log(d)^{1/3})$	JL		FJLT, FJLTr, FWI	
k in $\omega((d\log d)^{1/3})$ and $O(d^{1/2-\delta})$	JL	FJLT, FJLTr	FWI	
k in $O(d^{1/2-\delta})$ and $k < d$	JL, FJLT, FJLTr	JL concatenation		

Tab. 2.1: Result summary. Schematic comparison of asymptotic running time of six projection algorithms. 1) JL: a naïve implementation of Johnson-Lindenstrauss. 2) FJLT: the fast JL transform by Ailon Chazelle [1]. 3) FJLTr: a revised version of the FJLT algorithm, Chapter 3 and [2]. 4) FWI: A new two stage projection process, chapter 5 and [2]. 5) JL concatenation: a concatenation of several independent projections. 6) JL + Mailman: implementation of the Mailman algorithm [35] to Achlioptas's result. The results termed FJLTr, FWI, JL + mailman, and JL concatenation are described in this thesis.

	The rectangular $k \times d$ matrix A	Application time	$x \in \chi$ if $\|x\|_2 = 1$ and:
Johnson, Lindenstrauss [3]	k rows of a random unitary matrix	$O(kd)$	
Various Authors [32, 31, 33]	i.i.d random entries	$O(kd)$	
Ailon, Chazelle [1]	Sparse Gaussian entrees	$O(k^3)$	$\|x\|_\infty = O((d/k)^{-1/2})$
Matousek [34]	Sparse ±1 entries	$O(k^2 d\eta^2)$	$\|x\|_\infty \leq \eta$
This thesis [2]	4-wise independent ±1 matrix (deterministic)	$O(d \log k)$	$\|x\|_4 = O(d^{-1/4})$
This thesis [36]	Any deterministic matrix (deterministic)	?	$\|x\|_A = O(k^{-1/2})$
This thesis [36]	Lean Walsh Transform (deterministic)	$O(d)$	$\|x\|_\infty = O(k^{-1/2} d^{-\delta})$

Tab. 2.2: Different distributions for $k \times d$ matrices and the subsets χ of $\mathbb{S}^{d-1}$ for which they constitute a random projection (composed with a random diagonal ±1 matrix). The meaning of $\| \cdot \|_A$ is given in Definition 4.2.1.

3. REVISED FAST JOHNSON LINDENSTRAUSS TRANSFORM

In this chapter we review the Fast Johnson Lindenstrauss transform result by Ailon and Chazelle [?] and improve its running time by revising it slightly. The FJLT algorithm (recaped in subsection 3.1) performers dimensionality reduction from dimension d to dimension k in $O(d\log(d) + k^3)$ operations (per vector). This running time cannot be optimal. One way to see this is the case where $k \in o(\log(d))$. In this case $dk \in o(d\log(d) + k^3)$ and the FJLT algorithm is slower than a naïve implementation of the JL lemma. This chapter is dedicated to revising the FJLT to use efficient versions of the Walsh Hadamard and Discrete Fourier transforms. Other than using more efficient transforms the revised FJLT algorithm (FJLTr) is identical to the FJLT algorithm and does not need to be reproved. This chapter's results appeared in [2] and in [37].

3.1 Review of the FJLT algorithm

The FJLT result claims that a composition of three matrices PHD exhibits the JLP for P, H, and D being:

- D: A diagonal matrix who's diagonal is random ±1 with probability 1/2
- H: Either a Walsh Hadamard $d \times d$ Matrix or a $d \times d$ Discrete Fourier matrix.
- P: A k by d matrix who's i.i.d entries are either zeros with probability $1 - q$ or normally distributed according to $N(0, q^{-1})$

where $q = \Theta(\varepsilon^3 \log^2(n) d^{-1})$ and we assume that $q < 1$. (if $q \geq 1$ we set $q = 1$ and the claim is trivial due to prior constructions).

Lemma 3.1.1 (Ailon, Chazelle [1]). *Let $\Psi = PHD$ be chosen according to the distribution described above, then Ψ exhibits the JL property. Moreover, applying Ψ to any vector in $x \in \mathbb{R}^d$ requires* $O(d\log(d) +$

$\min\{kd, \log^3(n)\varepsilon^{-2})\}$ *operations in expectancy.*

The running time required for applying Ψ depends on the number of nonzeroes, n_{nnz}, in P, $E(n_{nnz}) = kdq = k\varepsilon^3\log^2(n)$. By recalling that $k = \Theta(\log(n)/\varepsilon^2)$ and viewing ε as a constant we get that $E(n_{nnz}) = O(k^3)$. Using Markov's inequality for the random variable n_{nnz} gives $\Pr(n_{nnz} > \frac{1}{\delta}E(n_{nnz})) \leq \delta$. Thus, the case where $n_{nnz} > \frac{1}{\delta}E(n_{nnz})$ can be added to the overall failure probability and the number of non-zeros in P can be thought of as $O(k^3)$.

Observe that computing $\Psi x = PHDx$ as proposed by the FJLT algorithm, i.e, $P(H(Dx))$, is wasteful. The number of coefficients needed is at most the number of non-zeros in P which is $O(k^3)$. The number of coefficients computed by $H(Dx))$ is d which is potentially mush larger then k^3. Clearly, computing only the relevant $O(k^3)$ coefficients is beneficial. The next two sections describe how to efficiently achieve this for both the Walsh Hadamard and the Distraite Fourier transforms.

3.2 Trimmed Walsh Hadamard transform

The Walsh Hadamard transform of a vector $x \in \mathbb{R}^d$ is the result of the matrix-vector multiplication Hx where H is a $d \times d$ matrix whose entries are $H(i,j) = (-1)^{\langle i,j \rangle}$. Here $\langle i,j \rangle$ means the dot product over $\mathbb{F}_2$ of the bit representation of i and j as binary vectors of length $(\log d)$.

The number of operation to compute a single coefficient $(Hx)(i) = \sum_{i=0}^{d-1}(-1)^{\langle i,j \rangle}x(j)$, denoted by $T(d,1)$, is $O(d)$. Moreover, all d coefficients can be computed in $O(d\log(d))$. We remind the reader that the Walsh-Hadamard matrix (up to normalization) can be recursively described as

$$H_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad H_q = \begin{pmatrix} H_{q/2} & H_{q/2} \\ H_{q/2} & -H_{q/2} \end{pmatrix}$$

Claim 3.2.1. *Any k' coefficients of a $d \times d$ Hadamard transform can be computed in $T(d,k') = O(d\log(k'))$ operations.*

Proof. Computing k' coefficients out of a $d \times d$ Hadamard transform H can be viewed as computing the outcome of PHx where P is $k' \times d$ matrix containing k' nonzeros, one per row. Define x_1 and x_2 to be the first and second halves of x. Similarly, we define P_1 and P_2 as the left and right halves of P respectively.

$$\begin{aligned} PH_q x = & \begin{pmatrix} P_1 & P_2 \end{pmatrix} \begin{pmatrix} H_{q/2} & H_{q/2} \\ H_{q/2} & -H_{q/2} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ = & P_1 H_{q/2}(x_1 + x_2) + P_2 H_{q/2}(x_1 - x_2) \end{aligned} \tag{3.1}$$

Let P_1 and P_2 contain k'_1 and k'_2 nonzeros respectively, $k_1 + k_2 = k'$. Equation 3.1 yields the following recurrence relation $T(d, k') = T(d/2, k_1) + T(d/2, k_2) + d$. The base cases are $T(d, 0) = 0$ and $T(d, 1) = d$. We use induction to show that $T(d, k') \leq 2d \log_2(k' + 1)$.

$$\begin{aligned} T(d,k) &= T(d/2,k'_1) + T(d/2,k'_2) + d \text{ By the recusance relation} \\ &\leq 2\frac{d}{2}\log_2(k'_1+1) + 2\frac{d}{2}\log_2(k'_2+1) + d \text{ By the induction assumption} \\ &\leq 2d\log_2(\sqrt{2(k'_1+1)(k'_2+1)}) \\ &\leq 2d\log_2(k'_1+k'_2+1) \text{ for all } k'_1+k'_2 = k' \geq 1 \\ &\leq 2d\log_2(k'+1) \end{aligned}$$

The last sequence of inequalities together with the base cases also give a simple and efficient Divide and Conquer algorithm. □

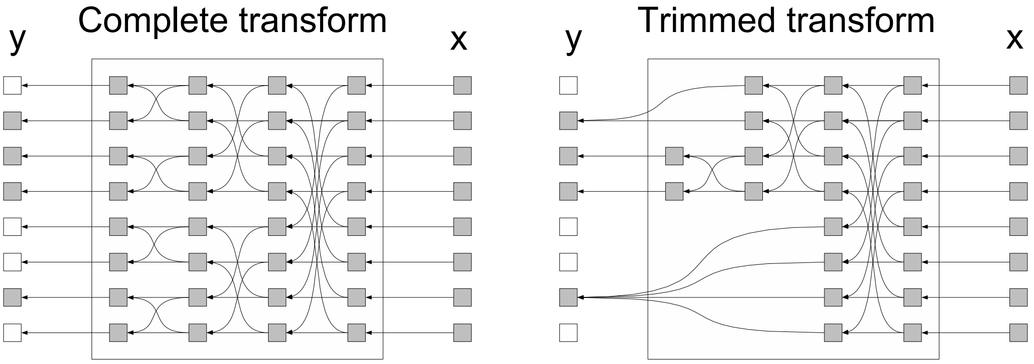


Fig. 3.1: A diagram describing the trimmed Walsh Hadamard transform.

3.3 Trimmed Discrete Fourier Transform

Since in [1] both Hadamard and Fourier transforms were considered, for completeness, we also remind the reader the Cooley Tukey DFT (Discrete Fourier Transform) algorithm and describe a simple trimmed Version of it.

The discrete Fourier Transform $f_x = DFT(x)$ for a vector $x \in \mathbb{R}^d$ is given by $f_x(i) = \sum_{j=0}^{d-1} e^{2\pi\sqrt{-1}ij/d}$. Thus, any single coefficient $f_x(i)$ is easily computed in $O(d)$ operations. Fast DFT algorithms compute all d coefficients in $O(d\log(d))$ operations instead of $O(d^2)$. The Coley Tukey algorithm is a generalization of the more well known Radix-2 algorithm. A sketch of the algorithm is shown in figure 3.3.

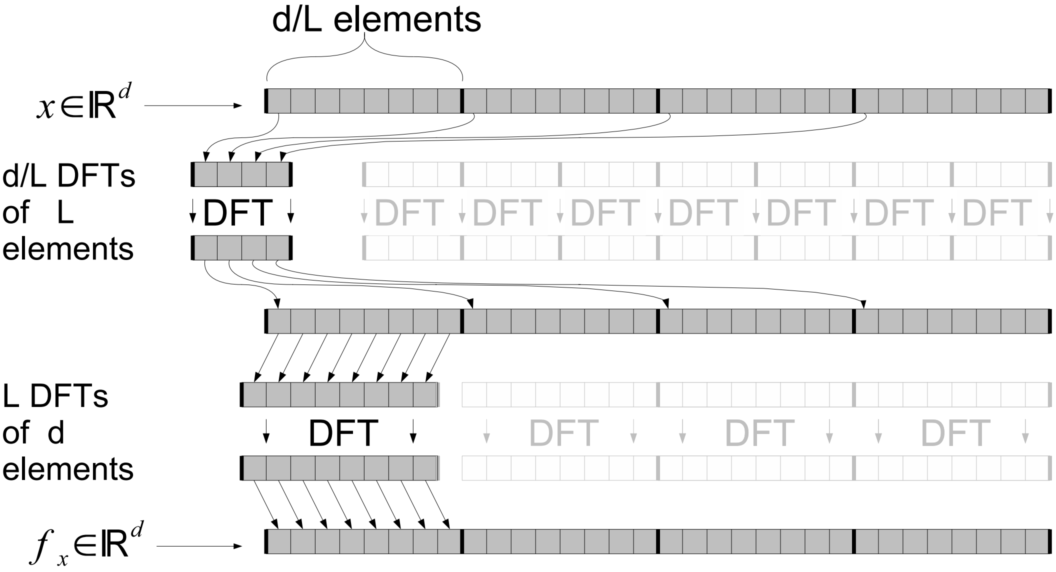


Fig. 3.2: A sketch describing the famous Fast DFT Coley Tukey Algorithm. The sketch does not show the multiplication by twiddle factors of the temporary array (in the middle).

In order to compute k' coefficients from $DFT(x)$ we divide x into L blocks of size d/L and begin with the first step of the Cooley Tukey algorithm which performs d/L DFT's of size L between the blocks (and multiplies them by twiddle factors). In the second step, instead of computing DFT's inside each block, each coefficient is computed directly, by summation, inside its block. These two steps require $(d/L) \cdot L\log(L)$ and $k'd/L$ operations respectively. By choosing $k'/\log(k') \leq L \leq k'$ we achieve a running time of $O(d\log(k'))$.

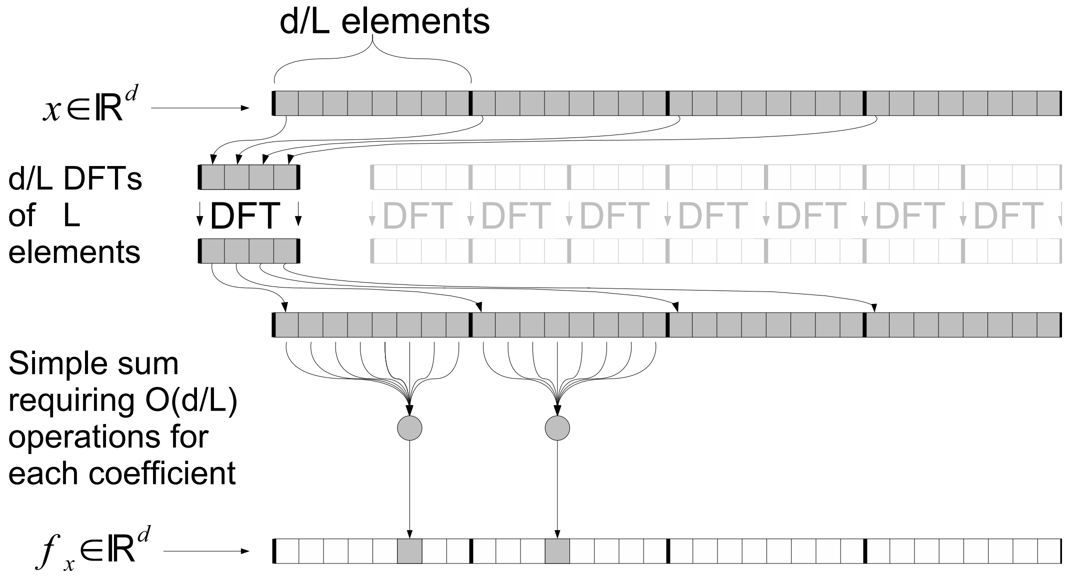


Fig. 3.3: A sketch describing the trimmed DFT Coley Tukey Algorithm. If $k'/\log(k') \leq L \leq k'^{O(1)}$ computing k' coefficient requires only $O(d\log(k'))$ operations.

3.4 FJLTr conclusion

The proposed revised FJLT algorithm (FJLTr) is thus identical to the original FJLT algorithm except for replacing the application of Ψ as $P(H(Dx))$ by $(PH)(Dx)$ using the fast trimmed transforms above. Taking $k' = k^3$ gives the running time of the FJLTr algorithm to be $O(d\log(k) + k^3)$. This is asymptotically faster then the naïve $O(dk)$ for arbitrary small values of k. The FJLTr and FJLT algorithms perform asymptotically identical when $k = O(\text{poly}(d))$. However, for any $k \in o(\text{poly}(d))$ the FJLTr algorithm outperforms the FJLT algorithm. Table 3.4 summarizes this chapter's result.

	Naïve or Slower	Faster then naïve	$O(d\log(k))$
k in $o(\log d)$	JL,FJLT		FJLTr
k in $\omega(\log d)$ and $o(\text{poly}(d))$	JL	FJLT	FJLTr
k in $\Omega(\text{poly}(d))$ and $o(d\log(d)^{1/3})$	JL		FJLT, FJLTr
k in $\omega((d\log d)^{1/3})$ and $o(d^{1/2})$	JL	FJLT, FJLTr	
k in $\Omega(d^{1/2})$ and $O(d)$	JL, FJLT, FJLTr		

Tab. 3.1: Result summary. Schematic comparison of asymptotic running time of three projection algorithms. a naïve implementation of Johnson-Lindenstrauss (JL), the fast Johnson-Lindenstrauss transform by Ailon Chazelle (FJLT), and the revised Fast Johnson-Lindenstrauss transform (FJLTr)

.

4. TWO STAGE PROJECTION PROCESS

Given a fixed matrix A we consider the concentration behavior of a the random variable $Y = \|AD_s x\|_2$ where D_s is a random ± 1 diagonal matrix. If Y concentrates sufficiently well around the value $\|x\|_2$ we get that AD_s behaves like a good random projection for x. The concentration of Y very much depends on both A and x. This chapter is dedicated to exploring the relation between A, x, and the concentration of Y. Namely, we seek a set $\chi(A) \subset \mathbb{R}^d$ such that $x \in \chi(A)$ guaranties that AD_s is a good random projection for x.

4.1 Concentration result

Since our mapping is linear we can assume without loss of generality that all the vectors in χ have norm 1. We say that a $k \times d$ matrix, A, is a good projection for χ if:

$$\forall x \in \chi \;\; \Pr[|\|AD_s x\|^2 - 1| \geq \varepsilon] \leq 1/n \tag{4.1}$$

for some constants n and $0 < \varepsilon < 1/2$. We also denote by D_s a diagonal matrix who's entrees are ± 1 with probability 1/2 each.[1]

Consider the term $AD_s x$, it can be expanded into the sum $AD_s x = \sum_{i=1}^{d} A^{(i)} x(i) s(i)$, where $s(i)$ are random i.i.d ± 1 variables. One can consider this to be a random walk in dimension k where the i'th step is the vector $A^{(i)} x(i) \in \mathbb{R}^k$. The variable Y is thus the distance from the origin that such a random walk yields. We measure the concentration of Y using a result by Talagrand [38]. His result actually holds in a much more general setting of convex Lipschitz bounded functions over Banach spaces. In our case, the finite

[1] Notice that if A is taken to be any of the known constructions (denoted by Ψ in the introduction) adding D_s does not change their distributions.

k dimensional vector space suffices. Notice that we can replace the term $AD_s x$ with $AD_x s$ where D_x is a diagonal matrix holding on its diagonal the values of x, i.e $D_x(i,i) = x(i)$ and s is a vector of random ± 1. The convex function in mind, is a function $f(s) = \|Ms\|$ on the random sign vector s, where $M = AD_x$.

Lemma 4.1.1 (Variation on Talagrand [38])**.** *For a matrix M and a random ± 1 vector s. Define the random variable $Y = \|Ms\|_2$. Denote by μ the median of Y and $\sigma = \|M\|_{2\to 2}$ the spectral norm of M, then:*

$$\Pr[|Y - \mu| > \varepsilon] < 4e^{-\varepsilon^2/8\sigma^2} \tag{4.2}$$

The lemma asserts that $\|AD_s x\|$ distributes like a (sub) Gaussian around it's median, with standard deviation 2σ. Let us first compute the expected value of $Y^2 = \|AD_s x\|^2$ by expanding the square term.

$$\begin{aligned} E(Y^2) = E(\|AD_s x\|_2^2) &= E\left(\sum_{i,j=1}^{d} \langle A^{(i)}, A^{(j)}\rangle x(i)x(j)s(i)s(j)\right) && (4.3)\\ &= \sum_{i=1}^{d} \|A^{(i)}\|^2 x^2(i) = \|x\|^2 = 1 && (4.4)\end{aligned}$$

The last equation holds if the columns of A are normalized. From this point on we shell assume that this is the case. To estimate the median, μ, we substitute $t^2 \to t'$ and compute:

$$\begin{aligned} E[(Y-\mu)^2] &= \int_0^\infty \Pr[(Y-\mu)^2] > t']dt' \\ &\le \int_0^\infty 4e^{-t'/(8\sigma^2)}dt' = 32\sigma^2 \end{aligned}$$

Furthermore, $(E[Y])^2 \le E[Y^2] = 1$, and so $E[(Y-\mu)^2] = E[Y^2] - 2\mu E[Y] + \mu^2 \ge 1 - 2\mu + \mu^2 = (1-\mu)^2$. Combining, $|1-\mu| \le \sqrt{32}\sigma$. We set $\varepsilon = t + |1-\mu|$:

$$\Pr[|Y - 1| > \varepsilon] \le 4e^{-\varepsilon^2/32\sigma^2} \quad \text{,for} \ \ \varepsilon > 2|1-\mu| \tag{4.5}$$

If we set $k = 33\log(n)/\varepsilon^2$ (for $\log(n)$ larger than a sufficient constant) and set $\sigma \le k^{-1/2}$ equation 4.5 meets the requirements of equation 4.1. Moreover, since $|1-\mu| \le \sqrt{32}\sigma$ the condition $\varepsilon > 2|1-\mu|$ is met for any constant ε. We see that $\sigma = \|AD_x\|_{2\to 2} \le k^{-1/2}$ is a sufficient condition for the projection to succeed w.h.p.

4.2 $\chi(A)$ the probabilistic Image

As discussed above the value of the term $\sigma = \|AD_x\|$ plays a crucial role in the random projection concentration phenomenon. We formally define $\|x\|_A$ as $\|AD_x\|_{2\to2}$ below.

Definition 4.2.1. *For a given matrix $A \in \mathbb{R}^{k\times d}$ we define the vector seminorm of $x \in \mathbb{R}^d$ with respect to A as $\|x\|_A \equiv \|AD_x\|_{2\to2}$ where D_x is a diagonal matrix such that $D_x(i,i) = x(i)$.*

Claim 4.2.1. *Let A be a constant matrix such that no column of A has zero norm. $\|x\|_A$ induces a proper norm on $\mathbb{R}^d$.*

Proof. Scalability) For any x and a constant c, $\|cx\|_A = |c| \cdot \|x\|_A$.

positive definiteness) For any $x > 0$ and a matrix A with no zero columns $\|x\|_A > 0$. To see this we multiply AD_x from the left by a test vector y^T.

$$
\begin{aligned}
(y^T AD_x)(i) &= \langle y^T, A^{(i)}\rangle x(i) && (4.6)\\
\|AD_x\|_{2\to2}^2 &\le \|y^T AD_x\|_2^2 && (4.7)\\
&= \sum_{i=1}^{d} \langle y^T, A^{(i)}\rangle^2 x^2(i) && (4.8)
\end{aligned}
$$

The last sum is larger the zero if y is set to $A^{(i)}$ for i such that $|x(i)| > 0$.

The triangle inequality) For any x_1 and x_2 we have:

$$
\begin{aligned}
\|x_1 + x_2\|_A &= \|AD_{x_1+x_2}\|_{2\to2} && (4.9)\\
&= \|AD_{x_1} + AD_{x_2}\|_{2\to2} && (4.10)\\
&\ge \|AD_{x_1}\|_{2\to2} + \|AD_{x_2}\|_{2\to2} && (4.11)\\
&= \|x_1\|_A + \|x_2\|_A && (4.12)
\end{aligned}
$$

□

In the case of dense random projections all columns of A have norm 1 and thus $\|x\|_A$ induces a proper norm. Also, recall from the previous section that a sufficient condition on x is that $\|AD_x\| = \|x\|_A \le$

$O(k^{-1/2})$. This gives us a concise geometric description of χ as the intersection of the Euclidian unit sphere and a ball of radius $k^{-1/2}$ in A-norm.

Definition 4.2.2. *Let A be a column normalized matrix. Let n and $0 < \varepsilon < 1/2$ be constants and let $k = 33\log(n)/\varepsilon^2$.*

$$\chi(A, \varepsilon, n) \equiv \mathbb{S}^{d-1} \bigcap \left\{ x \mid \|x\|_A \le k^{-1/2} \right\} \tag{4.13}$$

Lemma 4.2.1. *For any column normalized matrix A and χ as in definition 4.2.2 the following holds:*

$$\forall x \in \chi(A, \varepsilon, n) \;\; \Pr\left[\left| \|AD_s x\|^2 - 1 \right| \ge \varepsilon) \right] \le 1/n \tag{4.14}$$

Proof. To see this we substitute $\|x\|_A^2 = \sigma^2 \le 1/k$ into equation 4.5. □

4.3 ℓ_p bounds on A-norms

We turn to bound $\|x\|_A$ for a given A and x using more manageable terms. We use $\|x\|_p$ to denote the p-norm of x, $\|x\|_p = \left(\sum_{i=1}^d |x_i|^p\right)^{1/p}$ where $1 \le p < \infty$ and $\|x\|_\infty = \max_{i=1}^d |x_i|$. The dual norm index q is defined by the solution to $1/q + 1/p = 1$. We remind the reader that $\|x\|_p = \sup_{y, \|y\|_q = 1} x^T y$. For a real $k \times d$ matrix A, the matrix norm $\|A\|_{p_1 \to p_2}$ is defined as the operator norm of $A : (\mathbb{R}^d, \ell_{p_1}) \to (\mathbb{R}^k, \ell_{p_2})$

$$\|A\|_{p_1 \to p_2} = \sup_{x \in \mathbb{R}^d, \|x\|_{p_1} = 1} \|Ax\|_{p_2} = \sup_{x \in \mathbb{R}^d, \|x\|_{p_1} = 1} \; \sup_{y \in \mathbb{R}^k, \|x\|_{q_2} = 1} y^T Ax \tag{4.15}$$

Lemma 4.3.1. *For any dual norm indices p and q*

$$\|x\|_A \le \|x\|_{2p} \|A^T\|_{2 \to 2q} \tag{4.16}$$

Proof. We multiply the matrix AD_x from the left by a test vector $y \in \mathbb{R}^k$.

$$\begin{aligned}
\|x\|_A^2 &= \|AD_x\|_{2\to 2}^2 = \max_{y, \|y\|_2 = 1} \|y^T AD_x\|_2^2 && (4.17)\\
&= \max_{y, \|y\|_2 = 1} \sum_{i=1}^d x^2(i) (y^T A^{(i)})^2 && (4.18)\\
&\le \left(\sum_{i=1}^d x^{2p}(i) \right)^{1/p} \left(\max_{y, \|y\|_2 = 1} \sum_{i=1}^d (y^T A^{(i)})^{2q} \right)^{1/q} && (4.19)\\
&= \|x\|_{2p}^2 \|A^T\|_{2\to 2q}^2 && (4.20)
\end{aligned}$$

The transition from the second to the third line follows from Hölder's inequality which states that for vectors $z_1, z_2 \in R^d$, $\sum_{i=1}^{d} z_1(i)z_2(i) \leq \|z_1\|_p \|z_2\|_q$ for dual norms p and q. □

4.4 Conclusion

This chapter gave two results which together provide the skeleton on which we build from this point on. First, a matrix A can be used to project any vector x such $\|x\|_A \leq k^{-1/2}$. Second, $\|x\|_A \leq \|x\|_{2p} \|A^T\|_{2\to 2q}$ for any dual norms p and q. This gives a convenient relation between A and $\chi(A)$, namely

$$\{x \in \mathbb{S}^{d-1} | \|x\|_{2p} \|A^T\|_{2\to 2q} \leq k^{-1/2}\} \subset \chi(A) \tag{4.21}$$

Our framework is thus as follows: Choose a column normalized matrix $A \in \mathbb{R}^{k\times d}$ and a norm index q. Compute $\|A^T\|_{2\to 2q}$ and set $\eta = k^{-1/2}/\|A^T\|_{2\to 2q}$. The randomized isometry Φ is then required to achieve w.h.p $\|\Phi x\|_{2p} \leq \eta$. Given the results of this chapter, such a construction guaranties that the combination $AD_s\Phi$ exhibits the JL property.

Note that A might consist of more the k rows which might help in reducing its operator norm. In this case $AD_s\Phi$ is still a good random projection matrix but it does not necessarily exhibit the JL property. This is the case, for example, in chapter 6.

5. FOUR-WISE INDEPENDENCE AND RANDOM PROJECTIONS

In previous sections we showed that the running time of the FJLT algorithm can be reduced to $O(d\log(k)+k^3)$ by using efficient partial fast transforms. This, however, reduces to $O(k^3)$ when $d\log(k) = o(k^3)$. As claimed in the introductory chapter of this thesis and in [34] the k^3 term is unavoidable if one uses sparse random i.i.d projection matrices. The goal of this chapter is to show that fixed dense projections can be used to improve the running time to $O(d\log(k))$ for larger values of k, namely, for $k \in d^{1/2-\delta}$ for any positive constant δ.

This would require the use of a two stage projection (chapter 4). We will use a $k \times d$ four-wise independent matrix (definition **??**) B and claim that $\chi(B) = \{x \in \mathbb{S}^{d-1} \mid \|x\|_4 = O(d^{-1/4})\}$. We further claim that there exists a mapping Φ such that for any $x \in \mathbb{S}^{d-1}$ with high probability $\Phi(x) \in \chi(B)$. Finally, we claim that both Φ and B can be applied in $O(d\log(d))$ operations to any vector. Since we are currently interested in the case where $k \in O(\text{poly}(d))$ this also serves as a solution in $O(d\log(k))$ running time.

Unfortunately, $k \times d$ four-wise independent matrices only exist when $k \in O(d^{1/2})$. Moreover, the mapping Φ only succeeds with high enough probability when $k \in O(d^{1/2-\delta})$ for a constant positive δ. We thus improve on the FJLT algorithm for values of k such that $k \in O(d^{1/2-\delta})$ and $k \in \theta((d\log(d))^{1/3})$. This chapters main contribution is given formally in theorem 5.0.1.

Theorem 5.0.1. *Let $\delta > 0$ be some arbitrarily small constant. For any d, k satisfying $k \le d^{1/2-\delta}$ there exists an algorithm constructing a random matrix A of size $k \times d$ satisfying JLP, such that the time to compute $x \mapsto Ax$ for any $x \in \mathbb{R}^d$ is $O(d\log k)$. The construction uses $O(d)$ random bits and applies to both the Euclidean and the Manhattan cases.*

We will prove a slightly weaker running time of $O(d\log(d))$ since, as explained above, we are interested in the case where $k \in O(\text{poly}(d))$ and so $O(d\log(d)) = O(d\log(k))$. We will however provide in section 5.5

a sketch for reducing our constructions running time to $O(d \log k)$ for smaller values of k.

5.1 Tools from Error Correcting Codes

Definition 5.1.1. *A matrix $A \in \mathbb{R}^{k \times d}$ is a* code *matrix if every row of A is equal to some row of H_d multiplied by $\sqrt{d/k}$. The normalization is chosen so that columns have Euclidean norm* 1.

Let A be a code matrix, as defined above. The columns of A can be viewed as vectors over $\mathbb{F}_2$ under the usual transformation $((+) \rightarrow 0, (-) \rightarrow 1)$. Clearly, the set of vectors thus obtained are closed under addition, and hence constitute a linear subspace of $\mathbb{F}_2^m$. Conversely, any linear subspace V of $\mathbb{F}_2^m$ of dimension ν can be encoded as an $m \times 2^\nu$ code matrix (by choosing some ordered basis of V). We will borrow well known constructions of subspaces from coding theory, hence the terminology. Incidentally, note that H_d encodes the Hadamard code, equivalent to a dual BCH code of designed distance 3.

Definition 5.1.2. *A code matrix A of size $k \times d$ is a-*wise independent *if for each $1 \le i_1 < i_2 < \ldots < i_a \le k$ and $(b_1, b_2, \ldots, b_a) \in \{+1, -1\}^a$, the number of columns $A^{(j)}$ for which $(A^{(j)}_{i_1}, A^{(j)}_{i_2}, \ldots, A^{(j)}_{i_a}) = k^{-1/2}(b_1, b_2, \ldots, b_a)$ is exactly $d/2^a$.*

Lemma 5.1.1. *There exists a* 4*-wise independent code matrix of size $k \times f_{BCH}(k)$, where $f_{BCH}(k) = \Theta(k^2)$.*

The family of matrices is known as binary dual BCH codes of designed distance 5. Details of the construction can be found in [**?**].

Finally, we remind the reader the results of lemmas 4.2.1 and 4.3.1. Let D_x denote the diagonal matrix such $D(i,i) = x(i)$, and $\|x\|_B \equiv \|BD_x\|_{2 \rightarrow 2}$. First, we have that if $\|x\|_B = O(k^{-1/2})$ then the matrix BD_s exhibits the JL property with respect to x, where D_s is a diagonal random i.i.d ± 1 matrix. Second $\|x\|_B \le \|B^T\|_{2 \rightarrow 4} \|x\|_4$. We thus turn to compute the two different factors $\|B^T\|_{2 \rightarrow 4}$ and $\|x\|_4$.

5.2 Bounding $\|B^T\|_{2 \rightarrow 4}$ Using BCH Codes

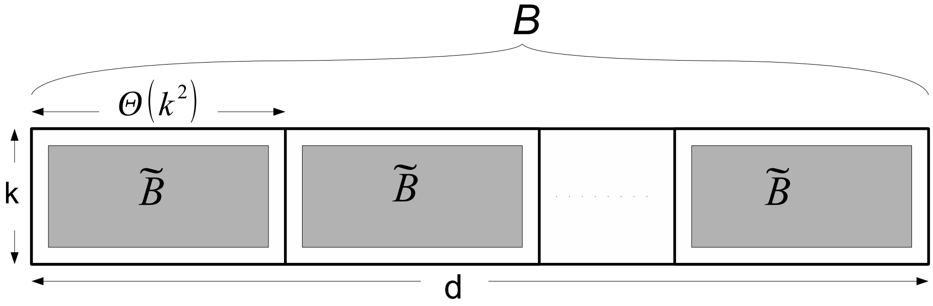
Lemma 5.2.1. *Assume B is a $k \times d$* 4*-wise independent code matrix. Then $\|B^T\|_{2 \rightarrow 4} \le (3d)^{1/4} k^{-1/2}$.*

Proof. For $y \in \ell_2^k, \|y\| = 1$,

$$\begin{aligned}
\|y^T B\|_4^4 &= dE_{j\in[d]}[(y^T B^{(j)})^4] \\
&= dk^{-2} \sum_{i_1,i_2,i_3,i_4=1}^{k} E_{b_{i_1},b_{i_2},b_{i_3},b_{i_4}}[y_{i_1}y_{i_2}y_{i_3}y_{i_4}b_{i_1}b_{i_2}b_{i_3}b_{i_4}] \qquad (5.1)\\
&= dk^{-2}(3\|y\|_2^4 - 2\|y\|_4^4) \le 3dk^{-2} \ ,
\end{aligned}$$

□

where b_{i_1} through b_{i_k} are independent random $\{+1, -1\}$ variables. We now use the BCH codes. Let $\widetilde{B}$ denote the $k \times f_{\text{BCH}}(k)$ matrix from the Lemma 5.1.1 (we assume here that $k = 2^a - 1$ for some integer a; This is harmless because otherwise we can reduce onto some $k' = 2^a - 1$ such that $k/2 \le k' \le k$ and pad the output with $k - k'$ zeros). In order to construct a matrix B of size $k \times d$ for $k \le d^{1/2-\delta}$, we first make sure that d is divisible by $f_{\text{BCH}}(k)$ (by at most multiplying d by a constant factor and padding with zeros), and then define B to be $d/f_{\text{BCH}}(k)$ copies of $\widetilde{B}$ side by side. Clearly B remains 4-wise independent. Note that B may no longer be a code matrix, but $x \mapsto Bx$ is still computable in time $O(d \log k)$ by performing $d/f_{\text{BCH}}(k)$ Walsh transforms on blocks of size $f_{\text{BCH}}(k)$.



5.3 Controlling $\|x\|_4$ for $k < d^{1/2-\delta}$

We define a randomized orthogonal transformation Φ that is computable in $O(d \log d)$ time and succeeds with probability $1 - O(e^{-k})$ for all $k < d^{1/2-\delta}$. Success means that $\|\Phi x\|_4 = O(d^{-1/4})$. (Note: Both big-$O$'s hide factors depending on δ). Note that this construction gives a running time of $O(d \log d)$. We discuss later how to do this for arbitrarily small k with running time $O(d \log k)$.

The basic building block is the product HD', where $H = H_d$ is the Walsh-Hadamard matrix and D' is a diagonal matrix with random i.i.d. uniform $\{\pm 1\}$ on the diagonal. Note that this random transformation was the main ingredient in [**?**]. Let $H^{(i)}$ denote the i'th column of H.

We are interested in the random variable $X = \|HD'x\|_4$. We define M as the $d \times d$ matrix with the i'th column $M^{(i)}$ being $x_i H^{(i)}$, we let $p = 4$ ($q = 4/3$), and notice that X is the norm of the Rademacher random variable in ℓ_4^d corresponding to M (using the notation of Section **??**). We compute the deviation σ,

$$\begin{aligned}
\sigma &= \|M\|_{2\to 4} = \|M^T\|_{4/3\to 2} \\
&= \sup_{\substack{y\in\ell_{4/3}^k \\ \|y\|_{4/3}=1}} \left(\sum_i x_i^2 (y^T H^{(i)})^2\right)^{1/2} \\
&\le \left(\sum x_i^4\right)^{1/4} \sup \left(\sum_i (y^T H^{(i)})^4\right)^{1/4} \\
&= \|x\|_4 \|H^T\|_{\frac{4}{3}\to 4} \ .
\end{aligned} \tag{5.2}$$

(Note that $H^T = H$.) By the Hausdorff-Young theorem, $\|H\|_{\frac{4}{3}\to 4} \le d^{-1/4}$. Hence, $\sigma \le \|x\|_4 d^{-1/4}$. We now get by Theorem **??** that for all $t \ge 0$,

$$\Pr[|\|HD'x\|_4 - \mu| > t] \le 4e^{-t^2/(8\|x\|_4^2 d^{-1/2})} \ , \tag{5.3}$$

where μ is a median of X.

Claim 5.3.1. $\mu = O(d^{-1/4})$.

Proof. To see the claim, notice that for each separate coordinate, $E[(HD'x)_i^4] = O(d^{-2})$ and then use linearity of expectation to get $E[\|HD'x\|_4^4] = O(d^{-1})$. By Jensen's inequality, $E[\|HD'x\|_4^b] \le E[\|HD'x\|_4^4]^{b/4} = O(d^{-b/4})$ for $b = 1, 3$. Now

$$\begin{aligned}
E[(\|HD'x\|_4 - \mu)^4] &= \int_0^\infty \Pr[(\|HD'x\|_4 - \mu)^4 > s] ds \\
&\le \int_0^\infty 4e^{-s^{1/2}/(8\|x\|_4^2 d^{-1/2})} ds \\
&= O(d^{-1}) \ .
\end{aligned}$$

This implies by multiplying the LHS out that $-\gamma_1 d^{-3/4}\mu - \gamma_2 d^{-1/4}\mu^3 + \mu^4 \le \gamma_3 d^{-1}$, where $\gamma_i > 0$ are global constants for $i = 1, 2, 3$. The statement of the claim immediately follows. □

Let c_9 be such that $\mu_4 \le c_9 d^{-1/4}$. We weaken inequality (5.3) using the last claim to obtain the following convenient form:

$$\Pr[\|HD'x\|_4 > c_9 d^{-1/4} + t] \le 4e^{-t^2/(8\|x\|_4^2 d^{-1/2})} \ . \tag{5.4}$$

In order to get a desired failure probability of $O(e^{-k})$ set $t = c_8 k^{1/2}\|x\|_4 d^{-1/4}$. For $k < d^{1/2-\delta}$ this gives $t < c_8 d^{-\delta/2}\|x\|_4$. In other words, with probability $1 - O(e^{-k})$ we get

$$\|HD'x\|_4 \le c_9 d^{-1/4} + c_8 d^{-\delta/2}\|x\|_4 \ .$$

Now compose this r times: Take independent random diagonal $\{\pm 1\}$ matrices $D' = D^{(1)}, D^{(2)}, \ldots, D^{(r)}$ and define $\Phi_d^{(r)} = HD^{(r)}HD^{(r-1)}\cdots HD^{(1)}$. Using a union bound on the conditional failure probabilities, we easily get:

Lemma 5.3.1. [ℓ_4 **reduction for** $k < d^{1/2-\delta}$] *With probability* $1 - O(e^{-k})$

$$\|\Phi^{(r)}x\|_4 = O(d^{-1/4}) \tag{5.5}$$

for $r = \lceil 1/2\delta \rceil$.

Note that the constant hiding in the bound (5.5) is exponential in $1/\delta$.

Combining the above, the random transformation $A = BD\Phi^{(r)}$ has Euclidean JLP for $k < d^{1/2-\delta}$, and can be applied to a vector in time $O(d \log d)$. This proves the Euclidean case of Theorem 5.0.1.

5.4 *Reducing to Manhattan Space for* $k < d^{1/2-\delta}$

We sketch this simpler case. As we did for the Euclidean case, we start by studying the random variable $W \in \ell_1^k$ defined as $W = \|k^{1/2}BDx\|_1$ for B as described in Section **??** and D a random ±1-diagonal matrix. In order to characterize the concentration of W (the norm of a Rademacher r.v. in ℓ_1^k) we compute the deviation σ, and estimate a median μ. As before, we set M to be the $k \times d$ matrix with the i'th column being $k^{1/2}B^{(i)}x_i$.

$$\begin{aligned}\sigma = \sup_{\substack{y\in\ell_\infty^k\\ \|y\|=1}} \|y^T M\|_2 &= \sup\left(k\sum_{i=1}^d x_i^2 (y^T B^{(i)})^2\right)^{1/2} \\ &\leq \sup k^{1/2}\|x\|_4 \|y^T B^{(i)}\|_4 = k^{1/2}\|x\|_4\|B^T\|_{\infty\to 4}\end{aligned} \tag{5.6}$$

Using the tools developed in the Euclidean case, we can reduce $\|x\|_4$ to $O(d^{-1/4})$ with probability $1 - O(e^{-k})$ using $\Phi_r(d)$, in time $O(d\log d)$ (in fact, $O(d\log k)$ using the improvement from Section 5.5). Also we already know from Section 5.2 that $\|B^T\|_{2\to 4} = O(d^{1/4}k^{-1/2})$ if B is comprised of $k \times f_{\text{BCH}}(k)$ dual BCH codes (of designed distance 5) matrices side by side (assume $f_{\text{BCH}}(k)$ divides d). Since $\|y\|_\infty \geq k^{-1/2}\|y\|_2$ for any $y \in \ell_k$, we conclude that $\|B^T\|_{\infty\to 4} = O(d^{1/4})$. Combining, we get $\sigma = O(k^{1/2})$. We now estimate the median μ of W.

In order to calculate μ we first calculate $E(W) = kE[|P|]$ where P is any single coordinate of $k^{1/2}BDx$. We follow (almost exactly) a proof by Matousek in [34] where he uses a quantitative version of the Central Limit Theorem by König, Schütt, and Tomczak [39].

Lemma 5.4.1. [König-Schütt-Tomczak] *Let $z_1 \dots z_d$ be independent symmetric random variables with $\sum_{i=1}^d E[z_i^2] = 1$, let $F(t) = \Pr[\sum_{i=1}^d z_i < t]$, and let $\overline{\varphi}(t) = \frac{1}{2\pi}\int_{-\infty}^t e^{-x^2/2}dx$. Then*

$$|F(t) - \overline{\varphi}(t)| \leq \frac{C}{1+|t|^3}\sum_{i=1}^d E[|z_i|^3]$$

for all $t \in \mathbb{R}$ and some constant C.

Clearly we can write $P = \sum_{i=1}^d z_i$ where $z_i = D'_i x_i$ and each D'_i is a random ±1. Note that $\sum_{i=1}^d E[|z_i|^3] = \|x\|_3^3$. Let β be the constant $\int_{-\infty}^\infty |t| d\overline{\varphi}(t)$ (the expectation of the absolute value of a Gaussian).

$$\begin{aligned}|E[|P|] - \beta| &= \left|\int_{-\infty}^\infty |t|dF(t) - \int_{-\infty}^\infty |t|d\overline{\varphi}(t)\right| \\ &\leq \int_{-\infty}^\infty |F(t) - \overline{\varphi}(t)|\, dt \\ &\leq \|x\|_3^3 \int_{-\infty}^\infty \frac{C}{1+|t|^3}dt\ .\end{aligned}$$

We claim that $\|x\|_3^3 = O(k^{-1})$. To see this, recall that $\|x\|_2 = 1$, $\|x\|_4 = O(d^{-1/4})$. Equivalently, $\|x^T\|_{2\to 2} = 1$ and $\|x^T\|_{4/3\to 2} = O(d^{-1/4})$. By applying Riesz-Thorin, we get that $\|x\|_3 = \|x^T\|_{3/2\to 2} = O(d^{-1/6})$, hence $\|x\|_3^3 = O(d^{-1/2})$. Since $k = O(d^{1/2})$ the claim is proved.

By linearity of expectation we get $E(W) = k\beta(1 \pm O(k^{-1}))$. We now bound the distance of the median from the expected value.

$$\begin{aligned}|E(W) - \mu| &\leq E[|W - \mu|] \\ &= \int_0^\infty \Pr[|W - \mu| > t]dt \\ &\leq \int_0^\infty 4e^{-t^2/(8\sigma^2)}dt = O(k^{1/2})\end{aligned}$$

(we used our estimate $\sigma = O(k^{1/2})$ above.) We conclude that $\mu = k\beta(1 + O(k^{-1/2}))$. This clearly shows that (up to normalization) the random transformation $BD\Phi^{(r)}$ (where $r = \lceil 1/\delta \rceil$) has the JL property with respect to embedding into Manhattan space. The running time is $O(d \log d)$.

5.5 Reducing the running time to $O(d \log k)$ for small k

Recall the construction in Section **??**: $\delta > 0$ is an arbitrarily small constant, we assume that $k \leq d^{1/2-\delta}$, that k^δ is an integer and that $\beta = f_{\mathrm{BCH}}(k)k^\delta$ divides d (all these requirements can be easily satisfied by slightly reducing δ and at most doubling d). The matrix B is of size $k \times d$, and was defined as follows:

$$B = (B_k \quad B_k \cdots B_k) \ ,$$

where B_k is the $k \times f_{\mathrm{BCH}}(k)$ code matrix from Lemma 5.1.1. Let $\hat{B}$ denote k^δ copies of B_k, side by side. So $\hat{B}$ is of size $k \times \beta$ and B consists of d/β copies of $\hat{B}$. As in Section **??** we start our construction by studying the distribution of the ℓ_2 estimator $Y = \|BDx\|_2$, where D is our usual random ± 1 diagonal matrix. Going back to (**??**) (recall that M is the matrix whose i'th column $M^{(i)}$ is $x_i B^{(i)}$), we recompute the deviation σ:

$$\begin{aligned}\sigma = \|M\|_{2\to 2} &= \sup_{\substack{y \in \ell_2^k \\ \|y\|=1}} \|y^T M\|_2 \\ &= \sup \left(\sum_{i=1}^d x_i^2 (y^T B^{(i)})^2 \right)^{1/2} \\ &= \sup \left(\sum_{j=1}^{d/\beta} \sum_{i \in I_j} x_i^2 (y^T B^{(i)})^2 \right)^{1/2} ,\end{aligned}$$

where I_j is the j'th block of β consecutive integers between 1 and d. Applying Cauchy-Schwartz, we get

$$\begin{aligned} \sigma &\leq \sup_{\substack{y \in \ell_2^k \\ \|y\|=1}} \left(\sum_{j=1}^{d/\beta} \|x_{I_j}\|_4^2 \|y^T \hat{B}\|_4^2 \right)^{1/2} \\ &= \left(\sup \|y^T \hat{B}\|_4 \right) \|x\|_{(4,2)} = \|\hat{B}^T\|_{2\to4} \|x\|_{(4,2)} \ , \end{aligned} \tag{5.7}$$

where $\|\cdot\|_{(p_1,p_2)}$ is defined by

$$\|x\|_{(p_1,p_2)} = \left(\sum_{j=1}^{d/\beta} \|x_{I_j}\|_{p_1}^{p_2} \right)^{1/p_2}$$

and $x_{I_j} \in \ell_{p_1}^{\beta}$ is the projection of x onto the set of coordinates I_j. Our goal, as in Section **??**, is to get $\sigma = O(k^{-1/2})$. By the properties of dual BCH code matrices (Lemma 5.2.1), we readily have that $\|\hat{B}^T\|_{2\to4} = O((f_{\mathrm{BCH}}(k)k^{\delta})^{1/4}k^{-1/2})$ which is $O(k^{\delta/4})$ by our construction. We now need to somehow "ensure" that $\|x\|_{(4,2)} = O(k^{-1/2-\delta/4})$ in order to complete the construction.

As before, we cannot directly control x (and its norms), but we can multiply it by random orthogonal matrices without losing ℓ_2 information. Let H' be a block diagonal $d \times d$ matrix with d/β blocks of the Walsh-Hadamard matrix H_β:

$$H' = \begin{pmatrix} H_\beta & & & \\ & H_\beta & & \\ & & \ddots & \\ & & & H_\beta \end{pmatrix} .$$

Let D' be a random diagonal $d \times d$ matrix over ±1. The random matrix $H'D'$ is orthogonal. We study the random variable $X' = \|H'D'x\|_{(4,2)}$. Let M' be the matrix with the i'th column $M'^{(i)}$ defined as $x_i H'^{(i)}$. We notice that X' is the norm of the Rademacher random variable in $\ell_{(4,2)}^d$ corresponding to M.

Remark: The results on Rademacher random variables, presented in Section **??**, apply also to "nonstandard" norms such as $\|\cdot\|_{(p_1,p_2)}$. The dual of $\|\cdot\|_{(p_1,p_2)}$ is $\|\cdot\|_{(q_1,q_2)}$, where q_1, q_2 are the usual dual norm indices of p_1, p_2, respectively. It is an exercise to check that $\|x\|_{(p_1,p_2)} = \sup_{\|y\|_{(q_1,q_2)}=1} x^T y$. We compute the deviation σ' and a median μ' of X' (as we did in (5.2)):

$$\begin{aligned}
\sigma' &= \|M\|_{2\to(4,2)} = \|M^T\|_{(4/3,2)\to 2} \\
&= \sup_{\substack{y\in\ell^k_{(4/3,2)}\\ \|y\|=1}} \left(\sum_i x_i^2 (y^T H^{(i)})^2\right)^{1/2} \\
&= \sup \left(\sum_{j=1}^{d/\beta}\sum_{i\in I_j} x_i^2 (y^T H'^{(i)})^2\right)^{1/2} \\
&\le \sup \left(\sum_{j=1}^{d/\beta} \|x_{I_j}\|_4^2 \|y_{I_j}^T H_\beta\|_4^2\right)^{1/2} \\
&\le \sup \left(\sum_{j=1}^{d/\beta} \|x_{I_j}\|_4^2 \|y_{I_j}\|_{4/3}^2 \|H_\beta^T\|_{4/3\to 4}^2\right)^{1/2} \\
&= \|H_\beta\|_{4/3\to 4} \sup \left(\sum_{j=1}^{d/\beta} \|x_{I_j}\|_4^2 \|y_{I_j}\|_{4/3}^2\right)^{1/2} ,
\end{aligned}$$

where the first inequality is Cauchy-Schwartz. By the inequality $(\sum_j A_j)^{1/2} \le \sum_j A_j^{1/2}$ holding for all nonnegative $A_1, A_2, \dots$, we get

$$\begin{aligned}
\sigma' &\le \|H_\beta\|_{4/3\to 4} \sup_{\substack{y\in\ell^k_{(4/3,2)}\\ \|y\|=1}} \sum_{j=1}^{d/\beta} \|x_{I_j}\|_4 \|y_{I_j}\|_{4/3} \\
&\le \|H_\beta\|_{4/3\to 4} \|x\|_{(4,2)} \ .
\end{aligned}$$

(The rightmost inequality is from the fact that $\sum_{j=1}^{d/\beta} \|y_{I_j}\|_{4/3}^2 = 1$ and the definition of $\|x\|_{(4,2)}$.) By Hausdorff-Young, $\|H_\beta\|_{4/3\to 4} \le \beta^{-1/4} = O(k^{-1/2-\delta/4})$, hence $\sigma' = O(k^{-1/2-\delta/4}\|x\|_{(4,2)})$. Any median μ' of X' is $O(k^{-1/2-\delta/4})$ (details omitted). Applying Theorem **??**, we get that for all $t \ge 0$,

$$\begin{aligned}
\Pr[X' > \mu' + t] &\le 4e^{-t^2/(8\sigma'^2)} \\
&\le \hat{c}_1 \exp\{-\hat{c}_2 t^2 k^{1+\delta/2}/\|x\|_{(4,2)}^2\} \ ,
\end{aligned}$$

for some global $\hat{c}_1, \hat{c}_2 > 0$. Setting $t = \Theta(\|x\|_{(4,2)} k^{-\delta/4})$, we get that

$$\Pr[\|H'D'x\|_{(4,2)} > \mu' + t] = O(e^{-k}) \ .$$

Similarly to the arguments leading to Lemma 5.3.1, and with possible readjustment of the parameter δ, we get using a union bound

Lemma 5.5.1. [$\ell_{(4,2)}$ **reduction for** $k < d^{1/2-\delta}$] *Let H', D' be as above, and let $\Phi' = H'D'$. Define $\Phi'^{(r)}$ to be a composition of r i.i.d. matrices, each drawn from the same distribution as Φ'. Then With probability $1 - O(e^{-k})$*

$$\|\Phi'^{(r)}x\|_{(4,2)} = O(k^{-1/2-\delta/4})$$

for $r = \lceil 1/2\delta \rceil$.

Combining the above, the random transformation $A = BD\Phi'^{(r)}$ has the JL Euclidean property for $k < d^{1/2-\delta}$, and can be applied to a vector in time $O(d \log k)$, as required. Indeed, multiplying by Φ' is done by doing a Walsh transform on d/β blocks of size β each, resulting in time $O(d \log k)$. Clearly the number of random bits used in choosing A is $O(d)$.

5.6 JL concatenation

The previous section produced a fast random projection from dimension d to k in $O(d\log(k))$ as long as $k \leq d^{1/2-\delta}$ for any constant positive δ. It, however, fails to exhibit the JL property for larger values of k. The only construction described thus far that exhibits the JL property for these values of k is the trivial one which requires $O(kd)$ operations to apply. This sudden increase in running time for a small increase in the target dimension k is unnatural. This section resolves this problem and allows to smoothly increase the running time from $O(d\log(k))$ to $O(dk)$ as k grows. The idea described is a natural one. We claim that a concatenation of projections which independently exhibit the JL property also exhibits the JLP. This improves our performance by allowing us to break the target dimension k into m sections of length k' each and apply m fast transforms from dimension d to dimension k'.

Lemma 5.6.1. *Let $\mathbb{D}_{k',d}$ be a distribution over $k' \times d$ matrices which exhibits the JL property. Define the distribution $\mathbb{D}^m_{k',d}$ to be a vertical concatenation of m matrices chosen independently from $\mathbb{D}_{k',d}$ normalized by $\frac{1}{\sqrt{m}}$. The distribution $\mathbb{D}^m_{k',d}$ over $k'm \times d$ matrices exhibits the JL property as well.*

Proof. Let A be a vertical concatenation of m, $k' \times d$, matrices $A_1, A_2, \ldots, A_m$ such that $A_1, A_2, \ldots, A_m$ are chosen i.i.d from a distribution $\mathbb{D}_{k',d}$ which exhibits the JL property. Let $y_i = A_i x$, let $Y_i = \|y_i\|_2^2$ and let

$Z_i = \sqrt{k'}(Y_i - 1)$. Since $\mathbb{D}_{k',d}$ is JL we have that

$$\begin{aligned}
\Pr(Y_i > 1 + 3\varepsilon) &\leq \Pr(\|y_i\| > 1 + \varepsilon) && (5.8)\\
&< c_1 e^{c_2 k' \varepsilon^2} && (5.9)\\
\Pr(Z_i > u) &\leq c_1 e^{-c_2 u^2} \text{ for } u \leq \varepsilon\sqrt{k'}. && (5.10)
\end{aligned}$$

Moreover we have that $E(Z_i) = 0$. Let us define the random variable $Y = \|Ax\|_2^2$. From the concatenation structure we have that $Y = \sum_{i=1}^{m} \frac{1}{m} Y_i$.

$$\begin{aligned}
Y - 1 &= \sum_{i=1}^{m} \frac{1}{m}(Y_i - 1) && (5.11)\\
&= \frac{1}{\sqrt{k'm}} \sum_{i=1}^{m} \frac{1}{\sqrt{m}} \sqrt{k'}(Y_i - 1) && (5.12)\\
&= \frac{1}{\sqrt{k}} \sum_{i=1}^{m} \frac{1}{\sqrt{m}} Z_i && (5.13)\\
& && (5.14)
\end{aligned}$$

Lemma 5.6.2. *(Matousek [34] (Lemma 2.2)) Let $Z_1, \ldots, Z_m$ be independent random variables, satisfying $E[Z_i] = 0$, $Var[Z_1] = 1$ and all having a uniform sub-gaussian tail. Let $\alpha_1, \ldots, \alpha_m$ be such that $\sum_{i=1}^{m} \alpha_i^2 = 1$, the variable $Z = \sum_{i=1}^{m} \alpha_i Z_i$ has $E[Z] = 0$, $Var[Z] = 1$ and a sub-gaussian tail.*

The following lemma holds also if $Var[Z_1] = Const$ for a constant other then 1. From equation 5.8 we have that Z_i are sub-gaussian, mean zero and constant variance. Therefore, the variable $Z = \sum_{i=1}^{m} \frac{1}{\sqrt{m}} Z_i$ also distributed like a sub-gaussian. Finally we have that $\sqrt{k}(Y - 1) = Z$ and thus $\Pr(\sqrt{k}(Y - 1) > u) \leq c_1 e^{-c_2 u^2}$. Replacing $u = \sqrt{k}\varepsilon$ we get $\Pr(Y > 1 + \varepsilon) \leq c_1 e^{-c_2 k \varepsilon^2}$ which is the JL property. □

Composing the result of lemma 5.6.1 and the efficient FJLT construction gives us the best result possible for the cases where $k \geq d^{1/2-\delta}$. We take k' to be $d^{1/2-\delta}$ and perform $m = k/k' = kd^{-1/2+\delta}$ independent transforms. Since each transform can be computed in $O(d\log(d))$ operations, the entire transform is computable in $O(kd^{1/2+\delta}\log(d))$. This outperforms the naive algorithm up to $k = \Omega(d)$ achieving a running time of $O(d^{3/2+\delta}\log(d))$.

5.7 Results summary

The current chapter described another approach in which a random projection can be obtained faster then the FJLT algorithm. Using dense orthogonal fast transforms was suggested. This permitted the acceleration to be useful for larger values of k than the FJLT algorithm permitted. The asymptotic running times of the improvements we achieved thus far are give in table 5.7.

	Naïve or Slower	Faster then naïve	$O(d\log(k))$
k in $o(\log d)$	JL,FJLT		FJLTr, FWI
k in $\omega(\log d)$ and $o(\text{poly}(d))$	JL	FJLT	FJLTr, FWI
k in $\Omega(\text{poly}(d))$ and $o((d\log(d)^{1/3})$	JL		FJLT, FJLTr, FWI
k in $\omega((d\log d)^{1/3})$ and $O(d^{1/2-\delta})$	JL	FJLT, FJLTr	FWI
k in $O(d^{1/2-\delta})$ and $k < d$	JL,FJLT,FJLTr	JL concatenation	

Tab. 5.1: Result summary. Schematic comparison of asymptotic running time of five projection algorithms. A naïve implementation of Johnson-Lindenstrauss (JL), the fast JL transform by Ailon Chazelle (FJLT), denoted by FJLTr the revised FJLT algorithm (chapter 3 and [2]), the result of this chapter denoted by FWI, and the projection composition method described above (JL concatenation)

6. TOWARDS LINEAR TIME DIMENSIONALITY REDUCTION

In this chapter we present a $k_{JL} \times d$ random projection matrix that is applicable to vectors $x \in \mathbb{R}^d$ in $O(d)$ operations if $d \geq k_{JL}^{2+\delta'}$. Here, k_{JL} is the minimal Johnson Lindenstrauss dimension and δ' is arbitrarily small. The projection succeeds, with probability $1 - 1/n$, in preserving vector lengths, up to distortion ε, for all vectors such that $\|x\|_\infty \leq \|x\|_2 k_{JL}^{-1/2} d^{-\delta}$ (for arbitrary small δ). Sampling based approaches are either not applicable in linear time or require a bound on $\|x\|_\infty$ that is strongly dependant on d. Our method overcomes these shortcomings by rapidly applying dense tensor power matrices to incoming vectors.

In the present work we examine the connection between A and χ for any matrix A (Section **??**). We propose in Section 6.1 a new type of fast applicable matrices and in Section 6.2 explore their χ. These matrices are constructed using tensor products and can be applied to any vector in $\mathbb{R}^d$ in linear time, i.e, in $O(d)$. Due to the similarity in their construction to Walsh-Hadamard matrices and their rectangular shape we term them *Lean Walsh Matrices*[1].

Due to their construction the Lean Walsh matrices are of size $\widetilde{d} \times d$ where $\widetilde{d} = d^\alpha$ for some $0 < \alpha < 1$. In order to reduce the dimension to $k_{JL} \leq \widetilde{d}$, $k_{JL} = O(\log(n)/\varepsilon^2))$, we compose the lean Walsh matrix, A, with a known Johnson Lindenstrauss matrix construction R. Applying R in $O(d)$ requires some relation between d, k_{JL} and α as explained in subsection 6.2.1.

[1] The terms *Lean Walsh Transform* or simply *Lean Walsh* are also used interchangeably.

	The rectangular $k_{JL} \times d$ matrix A	Application time	$x \in \chi$ if $\|x\|_2 = 1$ and:
Johnson, Lindenstrauss [?]	k_{JL} rows of a random unitary matrix	$O(kd)$	
Various Authors [?, ?, 33, 34]	i.i.d random entries	$O(kd)$	
Ailon, Chazelle [?]	Sparse Gaussian entrees	$O(k^3)$	$\|x\|_\infty = O((d/k)^{-1/2})$
Matousek [34]	Sparse ±1 entrees	$O(k^2 d\eta^2)$	$\|x\|_\infty \leq \eta$
this work [2]	4-wise independent Code matrix	$O(d \log k)$	$\|x\|_4 = O(d^{-1/4})$
This work	Any deterministic matrix	?	$\|x\|_A = O(k^{-1/2})$
This work	Lean Walsh Transform	$O(d)$	$\|x\|_\infty = O(k^{-1/2} d^{-\delta})$

Tab. 6.1: Types of $k \times d$ matrices and the subsets χ of $\mathbb{R}^d$ for which they constitute a random projection. The meaning of the norm $\|\cdot\|_A$ is given in Definition 4.2.1.

6.1 Lean Walsh transforms

The *Lean* Walsh Transform, similar to the Walsh Transform, is a recursive tensor product matrix. It is initialized by a constant seed matrix, A_1, and constructed recursively by using Kronecker products $A_{\ell'} = A_1 \otimes A_{\ell'-1}$. The main difference is that the Lean Walsh seeds have fewer rows than columns. We formally define them as follows:

Definition 6.1.1. *A_1 is a Lean Walsh seed (or simply 'seed') if i) A_1 is a rectangular matrix $A_1 \in \mathbb{C}^{r \times c}$, such that $r < c$; ii) A_1 is absolute valued $1/\sqrt{r}$ entree-wise, i.e, $|A_1(i,j)| = r^{-1/2}$; iii) the rows of A_1 are orthogonal; and iv) all inner products between its different columns are equal in absolute value to a constant $\rho \leq 1/\sqrt{(c-1)}$. ρ is called the Coherence of A_1.*

Definition 6.1.2. *A_ℓ is a Lean Walsh transform, of order ℓ, if for all $\ell' \leq \ell$ we have $A'_\ell = A_1 \otimes A_{\ell'-1}$, where $\otimes$ stands for the Kronecker product and A_1 is a seed according to definition 6.1.1.*

The following are examples of seed matrices:

$$A_1' = \tfrac{1}{\sqrt{3}} \begin{pmatrix} 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \qquad A_1'' = \tfrac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 1 \\ 1 & e^{2\pi i/3} & e^{4\pi i/3} \end{pmatrix} \tag{6.1}$$

$$r' = 3,\ c' = 4,\ \rho' = 1/3 \qquad\qquad r'' = 2,\ c'' = 3,\ \rho'' = 1/2$$

These examples are a part of a large family of possible seeds. This family includes, amongst other constructions, sub-Hadamard matrices (like A_1') or sub-Fourier matrices (like A_1''). A simple construction is given for possible larger seeds.

Fact 6.1.1. *Let F be the $c \times c$ Discrete Fourier matrix such that $F(i,j) = e^{2\pi\sqrt{-1}ij/c}$. Define A_1 to be the matrix consisting of the first $r = c-1$ rows of F normalized by $1/\sqrt{r}$. A_1 is a lean Walsh seed with coherence $1/r$.*

Proof. The facts that $|A_1(i,j)| = 1/\sqrt{r}$ and that the rows of A_1 are orthogonal are trivial. Moreover, due to the orthogonality of the columns of F, the inner product of two different columns of A_1 must equal $\rho = 1/r$ in absolute value.

$$\left|\langle A_1^{(j_1)}, A_1^{(j_2)}\rangle\right| = \frac{1}{r}\left|\sum_i^r \bar{F}(i,j_1)F(i,j_2)\right| = \frac{1}{r}\left|-\bar{F}(c,j_1)F(c,j_2)\right| = \frac{1}{r} \tag{6.2}$$

here $\bar{F}(\cdot,\cdot)$ stands for the complex conjugate of $F(\cdot,\cdot)$. □

We use elementary properties of Kronecker products to characterize A_ℓ in terms of the number of rows, r, the number of columns, c, and the coherence, ρ, of A_1. The following facts hold true for A_ℓ:

Fact 6.1.2. *i) A_ℓ is of size[2] $d^\alpha \times d$, where $\alpha = \log(r)/\log(c) < 1$ is the skewness of A_1 ii) for all i and j, $A_\ell(i,j) \in \pm\widetilde{d}^{-1/2}$ which means that A_ℓ is column normalized; and iii) the rows of A_ℓ are orthogonal.*

Fact 6.1.3. *The time complexity of applying A_ℓ to any vector $z \in \mathbb{R}^d$ is $O(d)$.*

[2] The size of A_ℓ is $r^\ell \times c^\ell$. Since the running time is linear, we can always pad vectors to be of length c^ℓ without effecting the asymptotic running time. From this point on we assume w.l.o.g $d = c^\ell$ for some integer ℓ

Proof. Let $z = [z_1; \dots; z_c]$ where z_i are sections of length d/c of the vector z. Using the recursive decomposition for A_ℓ we compute $A_\ell z$ by first summing over the different z_i according to the values of A_1 and applying to each sum the matrix $A_{\ell-1}$. Denoting by $T(d)$ the time to apply A_ℓ to $z \in \mathbb{R}^d$ we get that $T(d) = rT(d/c) + rd$. Due to the Master Theorem, and the fact that $r < c$ we have that $T(d) = O(d)$. More precisely, $T(d) \leq dcr/(c-r)$. □

For clarity, we demonstrate Fact 6.1.3 for A'_1 (equation 6.1):

$$A'_\ell z = A'_\ell \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{pmatrix} = \frac{1}{\sqrt{3}} \begin{pmatrix} A'_{\ell-1}(z_1 + z_2 - z_3 - z_4) \\ A'_{\ell-1}(z_1 - z_2 + z_3 - z_4) \\ A'_{\ell-1}(z_1 - z_2 - z_3 + z_4) \end{pmatrix} \tag{6.3}$$

In what follows we characterize $\chi(A, \varepsilon, n)$ for a general Lean Walsh transform by the parameters of its seed, r, c and ρ. The omitted notation, A, stands for A_ℓ of the right size to be applied to x, i.e, $\ell = \log(d)/\log(c)$. Moreover, we freely use α to denote the skewness $\log(r)/\log(c)$ of the seed at hand.

6.2 An ℓ_p bound on $\|\cdot\|_A$

After describing the lean Walsh transforms we turn our attention to exploring their "good" sets χ .We remind the reader that $\|x\|_A \leq k^{-1/2}$ entails $x \in \chi$:

$$\begin{aligned} \|x\|_A^2 &= \|AD_x\|_{2\to 2}^2 = \max_{y, \|y\|_2 = 1} \|y^T A D_x\|_2^2 && (6.4) \\ &= \max_{y, \|y\|_2 = 1} \sum_{i=1}^{d} x^2(i) (y^T A^{(i)})^2 && (6.5) \\ &\leq \left(\sum_{i=1}^{d} x^{2p}(i) \right)^{1/p} \left(\max_{y, \|y\|_2 = 1} \sum_{i=1}^{d} (y^T A^{(i)})^{2q} \right)^{1/q} && (6.6) \\ &= \|x\|_{2p}^2 \|A^T\|_{2\to 2q}^2 && (6.7) \end{aligned}$$

The transition from the second to the third line follows from Hölder's inequality for dual norms p and q, satisfying $1/p + 1/q = 1$. We are now faced with the computing $\|A^T\|_{2\to 2q}$ in order to obtain the constraint on $\|x\|_{2p}$.

Theorem 6.2.1. **[Riesz-Thorin]** *For an arbitrary matrix B, assume $\|B\|_{p_1\to r_1} \le C_1$ and $\|B\|_{p_2\to r_2} \le C_2$ for some norm indices p_1, r_1, p_2, r_2 such that $p_1 \le r_1$ and $p_2 \le r_2$. Let λ be a real number in the interval $[0, 1]$, and let p, r be such that $1/p = \lambda(1/p_1) + (1 - \lambda)(1/p_2)$ and $1/r = \lambda(1/r_1) + (1 - \lambda)(1/r_2)$. Then $\|B\|_{p\to r} \le C_1^{\lambda} C_2^{1-\lambda}$.*

In order to use the theorem, let us compute $\|A^T\|_{2\to 2}$ and $\|A^T\|_{2\to\infty}$. From $\|A^T\|_{2\to 2} = \|A\|_{2\to 2}$ and the orthogonality of the rows of A we get that $\|A^T\|_{2\to 2} = \sqrt{d/\widetilde{d}} = d^{(1-\alpha)/2}$. From the normalization of the columns of A we get that $\|A^T\|_{2\to\infty} = 1$. Using the theorem for $\lambda = 1/q$, for any $q \ge 1$, we obtain $\|A^T\|_{2\to 2q} \le d^{(1-\alpha)/2q}$. It is worth noting that $\|A^T\|_{2\to 2q}$ might actually be significantly lower then the given bound. For a specific seed, A_1, one should calculate $\|A_1^T\|_{2\to 2q}$ and use $\|A_\ell^T\|_{2\to 2q} = \|A_1^T\|_{2\to 2q}^{\ell}$ to achieve a possibly lower value for $\|A^T\|_{2\to 2q}$.

Lemma 6.2.1. *For a lean Walsh transform, A, we have that for any $p > 1$ the following holds:*

$$\{x \in \mathbb{R}^d \mid \|x\|_2 = 1, \|x\|_{2p} \le k_{JL}^{-1/2} d^{-\frac{1-\alpha}{2}(1-\frac{1}{p})}\} \subset \chi(A, \varepsilon, n) \tag{6.8}$$

where $k_{JL} = O(\log(n)/\varepsilon^2)$, $\alpha = \log(r)/\log(c)$, r is the number of rows, and c is the number of columns in the seed of A.

Proof. We combine the above and use the duality of p and q:

$$\begin{aligned} \|x\|_A &\le \|x\|_{2p}\|A^T\|_{2\to 2q} && (6.9)\\ &\le \|x\|_{2p} d^{\frac{1-\alpha}{2q}} && (6.10)\\ &\le \|x\|_{2p} d^{\frac{1-\alpha}{2}(1-\frac{1}{p})} && (6.11) \end{aligned}$$

The desired property, $\|x\|_A \le k_{JL}^{-1/2}$, is achieved if $\|x\|_{2p} \le k_{JL}^{-1/2} d^{-\frac{1-\alpha}{2}(1-\frac{1}{p})}$ for any $p > 1$. □

6.2.1 Controlling α and choosing R

We see that increasing α is beneficial from the theoretical stand point since it weakens the constraint on $\|x\|_p$. However, the application oriented reader should keep in mind that this requires the use of a larger seed, which subsequently increases the constant hiding in the big O notation of the running time.

Consider the seed constructions described in Fact 6.1.1 for which $r = c - 1$. Their skewness $\alpha = \log(r)/\log(c)$ approaches 1 as their size increases. Namely, for any positive constant δ there exists a constant size seed such that $1 - 2\delta \leq \alpha \leq 1$.

Lemma 6.2.2. *For any positive constant $\delta > 0$ there exists a Lean Walsh matrix, A, such that:*

$$\{x \in \mathbb{R}^d \mid \|x\|_2 = 1\,,\ \|x\|_\infty \leq k^{-1/2}d^{-\delta}\} \subset \chi(A, \varepsilon, n) \tag{6.12}$$

Proof. Generate A from a seed such that its skewness $\alpha = \log(r)/\log(c) \geq 1 - 2\delta$ and substitute $p = \infty$ into the statement of Lemma 6.2.1. □

The constant α also determines the minimal dimension d (relative to k_{JL}) for which the projection can be completed in $O(d)$ operations, the reason being that the vectors $z = AD_s x$ must be mapped from dimension $\widetilde{d}$ ($\widetilde{d} = d^\alpha$) to dimension k_{JL} in $O(d)$ operations. This is done using the Ailon and Liberty [2] construction serving as the random projection matrix R. R is a $k_{JL} \times \widetilde{d}$ Johnson Lindenstrauss projection matrix which can be applied in $\widetilde{d}\log(k_{JL})$ operations if $\widetilde{d} = d^\alpha \geq k_{JL}^{2+\delta''}$ for arbitrary small δ''. For the same choice of a seed as in lemma 6.2.2, the condition becomes $d \geq k_{JL}^{2+\delta''+2\delta}$ which can be achieved by $d \geq k_{JL}^{2+\delta'}$ for arbitrary small δ' depending on δ and δ''. Therefore for such values of d the matrix R exists and requires $O(d^\alpha \log(k_{JL})) = O(d)$ operations to apply.

6.3 Comparison to sparse projections

Sparse random ±1 projection matrices where analyzed by Matousek in [34]. For completeness we restate his result. Theorem 4.1 in [34] (slightly rephrased) claims the following:

Theorem 6.3.1 (Matousek 2006 [34])**.** *let $\varepsilon \in (0, 1/2)$ and $\alpha \in [1/\sqrt{d}, 1]$ be constant parameters. Set $q = C_0\alpha^2\log(n)$ for a sufficiently large constant C_0. Let S be a random variable such that*

$$S = \begin{cases} +\frac{1}{\sqrt{q}} & \textit{with probability } q/2 \\ -\frac{1}{\sqrt{q}} & \textit{with probability } q/2 \\ 0 & \textit{with probability } 1 - q \end{cases} \tag{6.13}$$

Let k be $C_1 \log(n)/\varepsilon^2$ for a sufficiently large C_1. Let the matrix $A \in \{-\frac{1}{\sqrt{q}}, 0, +\frac{1}{\sqrt{q}}\}^{k\times d}$ contain i.i.d copies of S then

$$\Pr[|\|Ax\|_2^2 - 1| > \varepsilon] \leq 1/n \tag{6.14}$$

For any $x \in \mathbb{S}^{d-1}$ such that $\|x\|_\infty \leq \alpha$.

With constant probability the number of nonzeros in A is $O(kdq) = O(k^2 d\alpha^2)$ (since ε is a constant $\log(n) = O(k)$). In the terminology of this paper we say that for A containing $O(k^2 d\alpha^2)$ nonzeros (as above) $\chi(A, \varepsilon, n) = \{x \in \mathbb{S}^{d-1} | \|x\|_\infty \leq \alpha\}$.

Notice that for a linear application time, $O(d)$, lean Walsh matrices require a weaker lower bound on the ℓ_∞ norm of x. By setting the number of nonzeros in the sparse A to $O(d)$ we get $\|x\|_\infty \leq \alpha \leq k^{-1}$. Whereas lean Walsh matrices require $\|x\|_\infty \leq k^{-1/2} d^{-\delta}$ which is larger as long as d is polynomial in k.

6.4 Conclusions

We have shown that any $k \times d$ (column normalized) matrix, A, can be composed with a random diagonal matrix to constitute a random projection matrix for some part of the Euclidian space, χ. Moreover, we have given sufficient conditions, on $x \in \mathbb{R}^d$, for belonging to χ depending on different $\ell_2 \to \ell_p$ operator norms of A^T and l_p norms of x. We have also seen that lean Walsh matrices exhibit both a "large" χ and a linear time computation scheme. These properties make them good building blocks for the purpose of random projections.

However, as explained in the introduction, in order for the projection to be complete, one must design a linear time preprocessing matrix Ψ which maps all vectors in $\mathbb{R}^d$ into χ (w.h.p). Achieving such Ψ would be extremely interesting from both the theoretical and practical stand point. Possible choices for Ψ may include random permutations, various wavelet/wavelet-like transforms, or any other sparse orthogonal transformation.

BIBLIOGRAPHY

[1] Nir Ailon and Bernard Chazelle. Approximate nearest neighbors and the fast johnson-lindenstrauss transform. In *STOC '06: Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 557–563, New York, NY, USA, 2006. ACM Press.

[2] Nir Ailon and Edo Liberty. Fast dimension reduction using rademacher series on dual bch codes. In *SODA*, pages 1–9, 2008.

[3] W.B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemp. Math.*, 26:189–206, 1984.

[4] Noga Alon. Problems and results in extremal combinatorics–I. *Discrete Mathematics*, 273(1-3):31–53, 2003.

[5] E. Kushilevitz, R. Ostrovsky, and Y. Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. *SIAM Journal on Computing*, 30(2):457–474, 2000.

[6] Sunil Arya and David M. Mount. Approximate nearest neighbor queries in fixed dimensions. In *Proceedings of the 4th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 271–280, Austin, Texas, United States, 1993.

[7] Piotr Indyk. On approximate nearest neighbors in non-Euclidean spaces. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 148–155, 1998.

[8] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC)*, pages 604–613, 1998.

[9] Piotr Indyk. Nearest neighbors in high-dimensional spaces. In *Handbook of Discrete and Computational Geometry*. CRC Press, 2004.

[10] Rosa I. Arriaga and Santosh Vempala. An algorithmic theory of learning: Robust concepts and random projection. *Machine Learning*, 63(2):161–182, 2006.

[11] Leslie G. Valiant. A neuroidal architecture for cognitive computation. *Lecture Notes in Computer Science*, 1443:642, 1998.

[12] N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, 1988.

[13] Alan M. Frieze, Ravi Kannan, and Santosh Vempala. Fast monte-carlo algorithms for finding low-rank approximations. In *IEEE Symposium on Foundations of Computer Science*, pages 370–378, 1998.

[14] Petros Drineas and Ravi Kannan. Fast monte-carlo algorithms for approximate matrix multiplication. In *IEEE Symposium on Foundations of Computer Science*, pages 452–459, 2001.

[15] P. Drineas, R. Kannan, and M. Mahoney. Fast monte carlo algorithms for matrices ii: Computing a low-rank approximation to a matrix, 2004.

[16] P. Drineas, R. Kannan, and M. Mahoney. Fast monte carlo algorithms for matrices iii: Computing a compressed approximate matrix decomposition, 2004.

[17] Tamás Sarlós. Improved approximation algorithms for large matrices via random projections. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, Berkeley, CA, 2006.

[18] Achlioptas and McSherry. Fast computation of low rank matrix approximations. In *STOC: ACM Symposium on Theory of Computing (STOC)*, 2001.

[19] P.G. Martinsson, V. Rokhlin, and M. Tygert. A randomized algorithm for the approximation of matrices.

[20] Liberty Edo, Woolfe Franco, Martinsson Per-Gunnar, Rokhlin Vladimir, and Tygert Mark. Randomized algorithms for the low-rank approximation of matrices. *Proceedings of the National Academy of Sciences*, December 2007.

[21] P. Drineas, M.W. Mahoney, and S. Muthukrishnan. Relative-error cur matrix decompositions. *TR arXiv:0708.3696*, submitted for publication, 2007.

[22] P. Drineas, M. W. Mahoney, S. Muthukrishnan, and T. Sarlos. Faster least squares approximation. *TR arXiv:0710.1435*, submitted for publication, 2007.

[23] P. Drineas, M. W. Mahoney, and S.M. Muthukrishnan. Sampling algorithms for ℓ_2 regression and applications. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Miami, Florida, United States, 2006.

[24] A. Dasgupta, P. Drineas, B. Harb, R. Kumar, and M. W. Mahoney. Sampling algorithms and coresets for ℓ_p regression. *Proc. of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2008.

[25] Santosh Vempala. *The Random Projection Method.* DIMACS Series in Discrete Mathematics and Theoretical Computer Science. 2004.

[26] Sariel Har-Peled. A replacement for Voronoi diagrams of near linear size. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 94–103, Las Vegas, Nevada, USA, 2001.

[27] P. Paschou, E. Ziv, E. Burchard, S. Choudhry, W. Rodriguez-Cintron, M. W. Mahoney, and P. Drineas. Pca-correlated snps for structure identification in worldwide human populations. *PLOS Genetics, 3, pp. 1672-1686*, 2007.

[28] P. Paschou, M. W. Mahoney, J. Kidd, A. Pakstis, K. Kidd S. Gu, and P. Drineas. Intra- and inter-population genotype reconstruction from tagging snps. *Genome Research, 17(1), pp. 96-107*, 2007.

[29] Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Knowledge Discovery and Data Mining*, pages 245–250, 2001.

[30] C.H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala. Latent semantic indexing: A probabilistic analysis. In *Proceedings of the 17th Annual Symposium of Database Systems*, pages 159–168, 1998.

[31] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of the johnson-lindenstrauss lemma. Technical Report TR-99-006, Berkeley, CA, 1999.

[32] P. Frankl and H. Maehara. The johnson-lindenstrauss lemma and the sphericity of some graphs. *J. Comb. Theory Ser. A*, 44(3):355–362, 1987.

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

[34] J. Matousek. On variants of the Johnson-Lindenstrauss lemma. *Private communication*, 2006.

[35] Edo Liberty and Steven Zucker. The mailman algorithm: a note on matrix vector multiplication. In *Yale university technical report #1402*, 2008.

[36] Edo Liberty, Nir Ailon, and Amit Singer. Dense fast random projections and lean walsh transforms. In *submitted*, 2008.

[37] Franco Woolfe, Edo Liberty, Vladimir Rokhlin, and Tygert Mark. A preliminary report on a fast randomized algorithm for the approximation of matrices. *Yale university computer science technical report YALE/DCS/TR1380*, April 2007.

[38] M. Ledoux and M. Talagrand. *Probability in Banach Spaces: Isoperimetry and Processes.* Springer-Verlag, 1991.

[39] Carsten Schütt Hermann König and Nicole Tomczak Jaegermann. Projection constants of symmetric spaces and variants of khintchine's inequality. *J. Reine Angew. Math*, 511:1–42, 1999.