A Sparse Johnson–Lindenstrauss Transform

Anirban Dasgupta  Ravi Kumar  Tamás Sarlós
Yahoo! Research
701 First Avenue
Sunnyvale, CA 94089,
{anirban, ravikumar, stamas}@yahoo-inc.com

ABSTRACT

Dimension reduction is a key algorithmic tool with many applications including nearest-neighbor search, compressed sensing and linear algebra in the streaming model. In this work we obtain a sparse version of the fundamental tool in dimension reduction — the Johnson–Lindenstrauss transform. Using hashing and local densification, we construct a sparse projection matrix with just \(O(\frac{1}{\epsilon^2})\) non-zero entries per column. We also show a matching lower bound on the sparsity for a large class of projection matrices. Our bounds are somewhat surprising, given the known lower bounds of \(\Omega(\frac{1}{\epsilon^2})\) both on the number of rows of any projection matrix and on the sparsity of projection matrices generated by natural constructions.

Using this, we achieve an \(O(\frac{1}{\epsilon^2})\) update time per non-zero element for a \((1 \pm \epsilon)\)-approximate projection, thereby substantially outperforming the \(O(\frac{1}{\epsilon^2})\) update time required by prior approaches. A variant of our method offers the same guarantees for sparse vectors, yet its \(O(d)\) worst case running time matches the best approach of Ailon and Liberty.

1. INTRODUCTION

Dimension reduction is a fundamental primitive with many algorithmic applications including nearest-neighbor search [2, 19], compressed sensing [11], data stream computations [5], computational geometry [13], numerical linear algebra [14, 17, 26, 28], machine learning [8, 33], graph sparsification [30], and more; see the monograph [32] for further applications. The seminal random projection method of Johnson and Lindenstrauss [20] consists of just multiplying the input vector by a suitably sampled random projection matrix — \(n\) vectors in \(d\)-dimensional space can be mapped into an \(O(\frac{1}{\epsilon^2} \log n)\)-dimensional subspace such that the length of each vector is distorted by at most \((1 \pm \epsilon)\). This simple and elegant method has the following desirable properties: (i) it is linear, (ii) it is oblivious to the input, (iii) it works with high probability for a given set of input points, and (iv) the target dimension is independent of \(d\).

Given its algorithmic importance, much effort has been devoted to speeding up the mapping. One line of work achieves this goal by making the projection matrix sparse, and hence its multiplication with the input vectors faster. Sparsity is typically achieved by independently setting each matrix entry to zero with a certain probability \([11, 2]\). There is however a limit on the extent of sparsity achievable by this approach: a result of Matousek [23, Theorem 4.1] states that such matrices need to contain \(\Omega(\frac{1}{\epsilon^2})\) non-zeros in expectation per column, if they were to preserve the length of a unit vector with infinity norm at most \(\alpha\).

Our results. We obtain a sparse random projection matrix of size \(k \times d\) that has \(O(\frac{1}{\epsilon^2} \log^2(\frac{1}{\epsilon}))\) non-zero entries per column, where \(k = O(\frac{1}{\epsilon^2} \log(\frac{1}{\epsilon}))\). This is the first construction with \(o(\frac{1}{\epsilon^2})\) non-zero entries in the projection matrix. (For our results to be improvements, we need to assume that \(\log^2(\frac{1}{\epsilon}) \approx o(\frac{1}{\epsilon^2})\). Our analysis, however, does not need this assumption.)

A highlight of our approach is to construct the projection matrix itself with care. Instead of using independent random variables, as is typically done, we construct it out of a hash function that entails some dependency among the entries. This construction is implicit in the work of Langford et al. [21] and Weinberger et al. [33], where it played a role mostly as a practical heuristic. The hash-based construction introduces new technical difficulties, but ensures that we have exactly a fixed number of non-zero entries in each column, thereby relaxing the requirements on the density of input vectors.

Specifically, whereas prior work requires that for a unit vector \(x\), \(\|x\|_{\infty} = O(\epsilon)\), for a constant number of expected entries per column of the projection matrix, we only need \(\|x\|_{\infty} = O(\sqrt{\epsilon})\). In order to achieve this level of densification, we can use a simple replication technique on \(x\) [33].

To manage the technical difficulties that arise from the dependencies, we show that the contribution from each hash bucket is bounded, and that the total amount of noise arising from the collisions in each hash bucket is small. The reduction in overall variance comes from the fact that each dimension is mapped to exactly one hash bucket, and the lack of self-collisions (which would be present if the entries in the matrix were i.i.d.) leads to a reduction in the variance of the cross-product error. There are several subtleties in analyzing this, in particular, the errors from different hash buckets being correlated. We handle this by an application of the FKG inequality on the product of the moment generating function of the random variables capturing the errors. This helps us in obtaining a...
concentration on the sum of the errors. Our choice of \( \pm 1 \) random variables (instead of Gaussian random variables\(^1\)) plays a critical role in making our proofs work.

**Implications for sparse vectors.** The resulting running time for an input vector \( x \) having \( n_{\text{na}}(x) \) non-zeros is \( \mathcal{O}(2^{n_{\text{na}}(x)} / \epsilon) \) — better than the running time obtained by \(^{22,23}\) for sparse vectors in terms of the sparsity ratio \( n_{\text{na}}(x) \) as well as by the factor \( \frac{1}{\epsilon} \). Furthermore, using a block-Hadamard based preconditioner, instead of a global Hadamard transform, we can actually ensure that our running time for all vectors is \( \mathcal{O}(\min(n_{\text{na}}(x), d)) \), which is once again an improvement over existing results. The qualitative difference in the running times is starker in the turnstile model of streaming. Since the updates in the stream come as \((i, v_i)\), updating any sketch that requires computing a global Hadamard transform is very expensive, taking \( \mathcal{O}(d) \) time per update. Our update time, on the other hand, is only \( \mathcal{O}(1) \) per entry.

Our technique speeds up nearest-neighbor computation for sparse vectors as well. We can use our construction to preprocess the input vectors and further improve the running time over even the most basic Johnson–Lindenstrauss transform. Unfortunately, their claim is false due to densification by replication was suggested by Weinberger et al.\(^{33}\). Although they claim a concentration bound for hashing-based dimensionality reduction, unfortunately, their claim is false due to an error in the application of Talagrand’s inequality.

\[^{22}\] Moreover, using a block-Hadamard based preconditioner, instead of a global Hadamard transform, we can actually ensure that our running time for all vectors is \( \mathcal{O}(\min(n_{\text{na}}(x), d)) \), which is once again an improvement over existing results. The qualitative difference in the running times is starker in the turnstile model of streaming. Since the updates in the stream come as \((i, v_i)\), updating any sketch that requires computing a global Hadamard transform is very expensive, taking \( \mathcal{O}(d) \) time per update. Our update time, on the other hand, is only \( \mathcal{O}(1) \) per entry.

Our technique speeds up nearest-neighbor computation for sparse vectors as well. We can use our construction to preprocess the input vectors and further improve the running time over even the most basic Johnson–Lindenstrauss transform. Unfortunately, their claim is false due to densification by replication was suggested by Weinberger et al.\(^{33}\). Although they claim a concentration bound for hashing-based dimensionality reduction, unfortunately, their claim is false due to an error in the application of Talagrand’s inequality.

\[^{23}\] Moreover, using a block-Hadamard based preconditioner, instead of a global Hadamard transform, we can actually ensure that our running time for all vectors is \( \mathcal{O}(\min(n_{\text{na}}(x), d)) \), which is once again an improvement over existing results. The qualitative difference in the running times is starker in the turnstile model of streaming. Since the updates in the stream come as \((i, v_i)\), updating any sketch that requires computing a global Hadamard transform is very expensive, taking \( \mathcal{O}(d) \) time per update. Our update time, on the other hand, is only \( \mathcal{O}(1) \) per entry.

**Implications for sparse vectors.** The resulting running time for an input vector \( x \) having \( n_{\text{na}}(x) \) non-zeros is \( \mathcal{O}(2^{n_{\text{na}}(x)} / \epsilon) \) — better than the running time obtained by \(^{22,23}\) for sparse vectors in terms of the sparsity ratio \( n_{\text{na}}(x) \) as well as by the factor \( \frac{1}{\epsilon} \). Furthermore, using a block-Hadamard based preconditioner, instead of a global Hadamard transform, we can actually ensure that our running time for all vectors is \( \mathcal{O}(\min(n_{\text{na}}(x), d)) \), which is once again an improvement over existing results. The qualitative difference in the running times is starker in the turnstile model of streaming. Since the updates in the stream come as \((i, v_i)\), updating any sketch that requires computing a global Hadamard transform is very expensive, taking \( \mathcal{O}(d) \) time per update. Our update time, on the other hand, is only \( \mathcal{O}(1) \) per entry.

Our technique speeds up nearest-neighbor computation for sparse vectors as well. We can use our construction to preprocess the input vectors and further improve the running time over even the most basic Johnson–Lindenstrauss transform. Unfortunately, their claim is false due to densification by replication was suggested by Weinberger et al.\(^{33}\). Although they claim a concentration bound for hashing-based dimensionality reduction, unfortunately, their claim is false due to an error in the application of Talagrand’s inequality.

\[^{22}\] Moreover, using a block-Hadamard based preconditioner, instead of a global Hadamard transform, we can actually ensure that our running time for all vectors is \( \mathcal{O}(\min(n_{\text{na}}(x), d)) \), which is once again an improvement over existing results. The qualitative difference in the running times is starker in the turnstile model of streaming. Since the updates in the stream come as \((i, v_i)\), updating any sketch that requires computing a global Hadamard transform is very expensive, taking \( \mathcal{O}(d) \) time per update. Our update time, on the other hand, is only \( \mathcal{O}(1) \) per entry.

Our technique speeds up nearest-neighbor computation for sparse vectors as well. We can use our construction to preprocess the input vectors and further improve the running time over even the most basic Johnson–Lindenstrauss transform. Unfortunately, their claim is false due to densification by replication was suggested by Weinberger et al.\(^{33}\). Although they claim a concentration bound for hashing-based dimensionality reduction, unfortunately, their claim is false due to an error in the application of Talagrand’s inequality.

\[^{23}\] Moreover, using a block-Hadamard based preconditioner, instead of a global Hadamard transform, we can actually ensure that our running time for all vectors is \( \mathcal{O}(\min(n_{\text{na}}(x), d)) \), which is once again an improvement over existing results. The qualitative difference in the running times is starker in the turnstile model of streaming. Since the updates in the stream come as \((i, v_i)\), updating any sketch that requires computing a global Hadamard transform is very expensive, taking \( \mathcal{O}(d) \) time per update. Our update time, on the other hand, is only \( \mathcal{O}(1) \) per entry.

Our technique speeds up nearest-neighbor computation for sparse vectors as well. We can use our construction to preprocess the input vectors and further improve the running time over even the most basic Johnson–Lindenstrauss transform. Unfortunately, their claim is false due to densification by replication was suggested by Weinberger et al.\(^{33}\). Although they claim a concentration bound for hashing-based dimensionality reduction, unfortunately, their claim is false due to an error in the application of Talagrand’s inequality.

\[^{1}\] In fact, we need an average of \( \frac{1}{\epsilon} \) Gaussians to get a \((1 \pm \epsilon)\)-approximation.

\[^{2}\] It is not hard to see that \( \sigma \) of \(^{22}\) equals to \( \max\{\sigma_i\} \) studied in Lemma\(^{6}\).

\[^{3}\] Let \( d > 6c \log(1/\epsilon) \). There exists a preconditioner \( G \in \mathbb{R}^{d \times d} \) such that for any input vector \( x \in \mathbb{R}^d \), with probability \( 1 - 4\delta \),

\[
(1 - \epsilon)\|x\|_2 \leq \|(HG)x\|_2 \leq (1 + \epsilon)\|x\|_2^2.
\]

The time required to compute \((HG)x\) is given by

\[
O\left(\min\left(\frac{n_{\text{na}}(x)}{\epsilon}, \log^4\left(\frac{1}{\epsilon\delta}\right), d\right) \log\left(\frac{1}{\epsilon\delta}\right)\right).
\]
3. PROOF OF THEOREM 2

3.1 Preliminaries

Without loss of generality, we can assume $||x||_2 = 1$. Let $Y_i = \sum_j H_{ij}x_j = \sum_j \delta_{h(j)} r_j x_j$, and let $\sigma^2 = E_r[Y_i^2]$, where $E_r$ is the expectation taken with respect to the random variables $r = \{r_j\}$. Thus,

$$\sigma^2 = E_r[Y_i^2] = E_r \left[ \left( \sum_{j \in [d]} \delta_{h(j)} r_j x_j \right)^2 \right] = \sum_{j \in [d]} \delta_{h(j)} x_j^2,$$

since the cross-product terms cancel out by the independence i.e., $E_r[r_j r_{j'}] = 0$ for $j \neq j'$. Let $Z_i = Y^2 - E_r[Y_i^2] = Y_i^2 - \sigma^2$.

The outline of the proof is as follows. We need to prove that $\sum_i Y_i^2$ is concentrated around zero. In fact, we will show that $\sum_i Z_i$ is concentrated around zero. Indeed, since our hash function guarantees that each coordinate $j \in [d]$ is mapped to one and exactly one hash bucket, we have that $\sum_{i=1}^n \sigma_i^2 = ||x||_2^2 = 1$. Therefore, $\sum_{i=1}^n Y_i^2 = \sum_i \sigma_i^2 + \sum_i Z_i = 1 + \sum_{i=1}^n Z_i$. Showing that $\sum_i Z_i$ is concentrated around zero is thus enough.

We will utilize the following form of the FKG inequality [6, Theorem 6.2.1].

**Theorem 4 (FKG Inequality)** Let $L$ be a finite distributive lattice and let $\mu : L \rightarrow \mathbb{R}^+$ be a log-supermodular function. Then, for an increasing function $f$ and a decreasing function $g$, we have that

$$\sum_{x \in L} \mu(x) f(x) g(x) \sum_{x \in L} \mu(x) \leq \sum_{x \in L} \mu(x) f(x) \sum_{x \in L} \mu(x) g(x).$$

3.2 Notation

Recall that $k = \frac{\alpha}{4\log(\frac{1}{\delta})}$. Define

$$\alpha = \alpha(k) = \frac{1}{\epsilon \ln(\frac{1}{\delta})}, \quad \sigma^2 = \frac{1 + \alpha}{k}, \quad \text{and} \quad \theta = \frac{4\sigma^2 k}{\delta};$$

we will assume $\alpha \geq 3$. We define the following function as a shorthand to denote the upper bound on conditional expectation of the MGF with respect to the $\{r_j\}$ variables.

$$G(u, t) = 1 + \frac{1}{\theta^2}(\exp(u\theta) - 1 - u\theta)t + \frac{4\delta}{k}.$$

**Definition 5 (Goodness)** A set $A \subseteq [d]$ is good if

$$\sum_{j \in A} x_j^2 \leq \sigma^2.$$

The $i$th hash bucket is good if $h^{-1}(i)$ is good, i.e., if $\sigma_i^2 = \sum_{j, h(j) = i} x_j^2 \leq \sigma^2$ and the hash function $h$ is good if $h^{-1}(i)$ is good for all $i$.

For a given $h$, let $\mathcal{G}_i$ denote the event that the $i$th hash bucket is good. Let $\mathcal{G}$ be the event that the hash function $h$ is good. By abusing notation we use $\mathcal{G}$ and $\mathcal{G}_i$ to represent the indicator variables of the corresponding events.

3.3 Proof details

Recall that $Z_i = Y_i^2 - E_r[Y_i^2]$, where $Y_i = \sum_j \delta_{h(j)} r_j x_j$, i.e.,

$$Z_i = \sum_{j \neq j', j', j \in [d]} \delta_{h(j)} \delta_{h(j')} r_j r_{j'} x_j x_{j'}.$$

Observe that $E[Z_i] = 0$ and our goal is to show that $\sum_i Z_i$ is concentrated around 0.

Here is an overview of the proof. We first show in Lemma 5 that most $h$ are good. In Lemma 6 we bound the moment generating function (MGF) of the random variable $Z_i$, for a fixed $h$. A usual step at this point would be to remove the effect of the bad choice of the random variables from the MGF by perhaps considering a truncated random variable $Z_i = \min(Z_i, M)$. In our case, however, such a construction would introduce a dependence among the $\{r_j\}$ and $h$ variables, which appears to be insurmountable when trying to apply the FKG inequality. We have to instead utilize the notion of goodness of $h$ only in defining the truncated random variable $Z_i$. Using the result of Lemma 2, we then get Corollary 3 that gives the expected and the worst-case bounds on the MGF for a good hash function $h$. We utilize these bounds to define $\tilde{Z}_i$ in $6$. Next, in Lemma 9 we define two set functions $f_s$ and $g_s$ and show that they are monotone, in accordance with the requirements of the FKG inequality (Theorem 4). These functions are then used in Lemma 10 to show that the MGF of $\sum_i Z_i$ can be bound by the product of the individual MGF’s $Z_i$. We then bound the probability of an $\epsilon$-deviation for $\sum_i Z_i$ in Theorem 11. Subsequently, we use Theorem 11 to prove Theorems 1 and 2. Section 4 gives the proof of Theorem 5.

**Lemma 6** If $c = \frac{4\epsilon}{\log(\frac{1}{\delta})}$, then $Pr[\mathcal{G}] \geq 1 - \delta$.

The proof (Appendix A.1) is an application of the Bernstein’s inequality [24, Theorem 2.7] and utilizes the fact that since $||x||_\infty \leq 1$, the hash function is random, with high probability, no $\sigma_i$ can be too large. In essence, this generalizes well-known facts about the maximum load in the balls into bins problem for the weighted case.

The following lemma gives a bound on the MGF of the variable $Z_i$ for a fixed $h$. The proof can be found in Appendix A.2.

**Lemma 7** If $u < \frac{1}{4k^2}$, then for a fixed $h$,

$$E_h[\exp(uZ_i)] \leq G(u, E_r[Z^2_i]).$$

**Lemma 7** leads to the following.

**Corollary 8** If $0 < u < \frac{1}{4k^2}$, then the expectation of the MGF can be bounded as

$$E_{h,r}[\exp(uZ_i) | \mathcal{G}] \leq G(u, \frac{1}{k^2}).$$

Similarly,

$$\max_{h \in \mathcal{G}} E_r[\exp(uZ_i)] \leq G(u, \sigma^2).$$

**Proof.** By taking expectation over $h$ and using

$$E_h[\exp(uZ_i) | \mathcal{G}] \leq 2E[Z^2_i] \leq \frac{2}{k^2},$$

we have that

$$E_{h,r}[\exp(uZ_i) | \mathcal{G}] \leq 1 + \frac{2}{k^2} \exp(u\theta) - 1 - u\theta + \frac{4\delta}{k} \leq \exp\left(\frac{2}{k^2} \exp(u\theta) - 1 - u\theta + \frac{4\delta}{k}\right).$$

3 Sanders [27] contains a proof of the expected load for the weighted ball-and-bins problem, but does not contain a proof of the high probability statement.
The upper bound on $E_r[\exp(uZ_i) \mid \Sigma]$ is given by

$$
\max_{h \in \Sigma} E_r[\exp(uZ_i)] \\
\leq 1 + \frac{1}{\theta^2}(\exp(u\theta) - 1 - u\theta) \max_h E_r[Z_i^2 \mid \Sigma] + \frac{4\delta}{k},
$$

where we use $E_r[Z_i^2] < \sigma_i^2$. □

Next, we have to handle the fact that the $Z_i$ variables are not independent. Yet, intuitively, since $Z_i$ is roughly related to the cross-product of the set of entries $x_j$ that map into the $i$th hash bucket, conditioned on the fact that one of the $Z_i$ variables has achieved a large value, the probability that another $Z_i'$ is large also decreases. In fact, we show that we can apply the FKG inequality (Theorem 4) on the MGF of the $Z_i$ random variables. Note that this situation is more involved that the simple negative dependence obtained on a set of random variables by conditioning their sum to be a constant — we cannot make such claims on $\sum_i Z_i$. For all $i = 1, \ldots, k$ let us define

$$
\hat{Z}_i = \begin{cases} 
Z_i & \text{if } \Sigma_i, \\
\frac{1}{u} \log G(u, \sigma_i^2) & \text{else.}
\end{cases}
$$

We first need the following lemma in preparation for the application of the FKG inequality (Theorem 4).

**Lemma 9**  For $1 \leq s \leq k$, $u < \frac{1}{4\sigma_i^2}$ and $A \subseteq [d]$, let us define

$$
\begin{align*}
\mu_s(A) &= E_r \left[ \exp \left( u\hat{Z}_s \right) \mid h^{-1}(s) = A \right] \\
g_s(A) &= E_{h,r} \left[ \exp \left( u\sum_{i=1}^{s-1} \hat{Z}_i \right) \mid h^{-1}(s) = A \right].
\end{align*}
$$

Then $\mu_s$ is an increasing and $g_s$ is a decreasing set function.

**Proof.** First we prove that $f_s$ is increasing by showing that for all $A \subseteq [d]$ and for all $a \in [d] \setminus A$, it holds that $f_s(A \cup \{a\}) \geq f_s(A)$.

Observe that if $h^{-1}(s)$ is good (i.e., if $\Sigma_s$ holds), then by Corollary 8 we have $E_r[\exp(uZ_i)] \leq G(u, \sigma_i^2)$. Thus for all $h$ and $s$, it holds from (5) that

$$
E_r[\exp(u\hat{Z}_s)] \leq G(u, \sigma_i^2).
$$

(6)

There are two cases to consider. Suppose $A \cup \{a\}$ is bad. Then, $\hat{Z}_s = \frac{1}{u} \log G(u, \sigma_i^2)$ and hence $f_s(A \cup \{a\}) = G(u, \sigma_i^2) \geq f_s(A)$ from (6). Suppose $A \cup \{a\}$ is good. Now, let us define

$$
V_A = \sum_{j: g \in A, j \neq g} r_j r_g x_j x_g \\
W_A = x_a \sum_{j \in A} x_j r_j.
$$

Also note that if $h^{-1}(s) = A \cup \{a\}$ and the $s$th bucket is good, then $\hat{Z}_s = Z_s = V_A + r_a W_A$. Therefore we have that

$$
\begin{align*}
f_s(A \cup \{a\}) &= E_r \left[ \exp \left( u\hat{Z}_s \right) \mid h^{-1}(s) = A \cup \{a\} \right] \\
&= E_r \left[ \exp \left( uV_A + u \cdot r_a W_A \right) \mid h^{-1}(s) = A \cup \{a\} \right] \\
&\geq E_r \left[ \exp \left( E_r \left[ \exp \left( uV_A + u \cdot r_a W_A \mid h^{-1}(s) = A \cup \{a\} \right) \right] \mid h^{-1}(s) = A \cup \{a\} \right] \\
&\geq E_r \left[ \exp \left( E_r \left[ \exp \left( uV_A + u \cdot r_a W_A \mid h^{-1}(s) = A \cup \{a\}, \{r_j\}_{j \in A} \right) \mid h^{-1}(s) = A \cup \{a\} \right] \right) \\
&(By Jensen's inequality, $E_r[\exp(x)] \geq \exp(E_r[x]))
\end{align*}
$$

Here, (a) follows since only $r_a$ is random in the inner expectation and $E_r[uV_A + u \cdot r_a W_A \mid h^{-1}(s) = A \cup \{a\}, \{r_j\}_{j \in A}] = uV_A$.

And, (b) follows since $a \notin A$ and $V_A$ does not depend on $h(a)$ by the independence of the values of $r$ and $h$. Finally, (c) follows since if $A \cup \{a\}$ is then so is $A$; therefore if $h^{-1}(s) = A$, then we have that $\hat{Z}_s = Z_s = V_A$. The proof that $f_s$ is increasing is complete.

The proof of $g_s$ being a decreasing function is similar, and can be found in Appendix 9.3 □

Given our construction of the two functions, $f_s$ and $g_s$, we can now proceed to apply the FKG inequality (Theorem 4) to show that the MGF of the random variable $\sum_i Z_i$ is bounded by the product of the MGF’s of each $\hat{Z}_i$ variable.

**Lemma 10** It holds that

$$
E \left[ \exp \left( u \sum_{i=1}^{k} \hat{Z}_i \right) \right] \leq \prod_{i=1}^{k} E \left[ \exp \left( u\hat{Z}_i \right) \right],
$$

where the expectation is taken over both $h$ and $r = \{r_j\}$.

**Proof.** For all $1 \leq s \leq k$, we prove

$$
E \left[ \exp \left( u \sum_{i=1}^{s-1} \hat{Z}_i \right) \right] \leq \prod_{i=1}^{s} E \left[ \exp \left( u\hat{Z}_i \right) \right],
$$

(7)

by induction on $s$. The base case of $s = 1$ is obvious.

Now assume that the inductive hypothesis (7) holds for $s - 1$. For all $A \subseteq [d]$ let us define

$$
\mu_s(A) = \Pr[h^{-1}(s) = A] = \prod_{j \in A} \Pr[h(j) = s] \prod_{j \notin A} \Pr[h(j) \neq s].
$$

It is easy to check that $\mu_s$ is a log-supermodular measure over the subsets of $[d]$. Recalling the definition of the increasing function $f_s$ and the decreasing function $g_s$ from Lemma 9 it follows from the FKG inequality (Theorem 5) that

$$
E_{\mu_s}[f_s g_s] \leq E_{\mu_s}[f_s] E_{\mu_s}[g_s].
$$

See [6] Section 6.2, page 87 for a precise definition and proof of this fact.
Furthermore, observe that for any random variable $X$ we have

$$E_{\mu_x}[E[X | h^{-1}(s) = A]] = \sum_{A \subseteq [d]} \Pr[h^{-1}(s) = A] E[X | h^{-1}(s) = A] = E[X],$$

and consequently,

$$E[\exp(u \sum_{i=1}^{s-1} Z_i) \exp(u \tilde{Z}_s)] \leq E[\exp(u \sum_{i=1}^{s-1} Z_i)] E[\exp(u \tilde{Z}_s)].$$

Combining the latter with the induction hypothesis for $s-1$ concludes the proof.

**Theorem 11** For the variables $Z_i$, we have

(i) \[ \Pr \left[ \sum Z_i \geq \epsilon \right] \leq \exp \left( \frac{-3k \epsilon^2}{4(3 + (1 + \alpha) \epsilon \ln(\frac{1}{\delta}))} + 4\delta \right), \]

(ii) \[ \Pr \left[ \sum Z_i < -\epsilon \right] < \exp \left( \frac{-\epsilon^2 k}{12} + \delta \right). \]

The proof of Theorem 11 involves a standard but tedious calculation that is similar to one done by Matousek [22]. The proof can be found in Appendix 9.4. Finally, we are ready to prove the main result.

**Proof.** (of Theorem 2) Recall that $Y_i = \sum_j H_{ij} x_j$, thus $||Hx||_2^2 = \sum_i Y_i^2$. Also recall that $\sigma_i^2 = E[|Y_i|^2]$. Thus, $\sum_{i=1}^{k} \sigma_i^2 = \|x\|_2^2$. Therefore, $\sum_{i=1}^{k} Y_i^2 = \sum_{i=1}^{k} \sigma_i^2 + \sum_{i=1}^{k} Z_i = 1 + \sum_{i=1}^{k} Z_i$. Plugging these values in Theorem 11(i), we have $\sum_{i=1}^{k} Z_i > \epsilon$, with probability at most $\exp(-\frac{\epsilon^2 k}{12} + 4\delta) + \delta < 2\delta$, for $\delta = \frac{\epsilon}{10}$. Similarly, from Theorem 11(ii), we have $\sum_{i=1}^{k} Z_i < -\epsilon$ with probability at most $2\delta$. Putting them together, with probability at least $1 - 4\delta$, $|\sum Y_i^2 - 1| = |\sum Z_i| < \epsilon$, and hence $||Hx||_2^2 - \|x\|_2^2 < \epsilon ||x||_2^2$.

**Proof.** (of Theorem 1) Theorem 1 easily follows from Theorem 2 by noting that if $y = P x$, then $||y||_2 = 1$ and $||y||_\infty \leq \frac{1}{\epsilon}$. The running time is obtained as computing both $y = P x$ and $H y$ requires $O(\epsilon \cdot n_{n_{ax}}(x))$ time.

4. **PROOF OF THEOREM 3**

**Definition 12** (Randomized Hadamard matrix [21]) Construct the $m \times m$ Hadamard matrix $F$ as $F_{ij} = m^{-1/2}(-1)^{(i-1)(j-1)}$ and the diagonal matrix $D$ by choosing each $D_{ii}$ independently from $\{-1, 1\}$ with probability $1/2$ for each value. The matrix $A = FD$ is defined to be an $m \times m$ randomized Hadamard matrix.

Using multiple small copies the randomized Hadamard matrix, we create the following preconditioner. Without loss of generality, we assume that $\frac{1}{\epsilon}$ is an integer, for the given value of $b$. We note that $\frac{1}{\epsilon}$ also contains a similar construct; here we present a more straightforward analysis using a different vector norm.

**Lemma 13** Let $x \in \mathbb{R}^d$, $||x|| = 1$, and $1 > \delta > 0$, and $c \geq 1$. Define $b = 6c \log(\frac{1}{\epsilon \delta})$ and assume $b \leq d$. Construct $G \in \mathbb{R}^{2d \times d}$ to be a random block-diagonal matrix, where each of the $d/b$ diagonal blocks of $G$ consist of an independent copy of a $b \times b$ randomized Hadamard matrix. Then we have that

$$\Pr \left[ ||Gx||_\infty \geq \frac{1}{\sqrt{c}} \right] \leq \delta.$$
Using the fourth moment method \cite{9}, we show that

$$E[Y_1^2] = E[U_1^2]E[U_1^2] = O(\frac{c}{d^2})$$.

By our assumptions it holds that $E[Y_1^2] = E[U_1^2]E[U_1^2] = O(\frac{c}{d^2})$.

If $s \neq t$ then $E[Y_s^2 Y_t^2] = E[U_s^2 U_t^2]E[U_s^2 U_t^2] \leq 0$ holds by independence and hence from Hölder’s inequality we have that $E[U_s^2 U_t^2] \leq \sqrt{E[U_s^2]E[U_t^2]}$. Thus it holds that $E[Y_s^2 Y_t^2] \leq O(\frac{c}{d^2})$.

Lastly, recall that for all $s, t_1, t_2, t_3$ where $s \notin \{t_1, t_2, t_3\}$ we have that $E[Y_s^2 Y_{t_1} Y_{t_2} Y_{t_3}] = 0$.

Now \cite[Theorem 3.5]{1} states that

$$\Pr \left[ |S| \geq \frac{1}{2} \sqrt{E[S^2]} \right] \geq \frac{(3/4)^2}{E[S^2]} - \frac{2}{16}$$.

Therefore we have that

$$\Pr \left[ |S| \geq \frac{\sqrt{x_{ij}}}{2c} \right] \geq \frac{(3/4)^2}{o(1)} = \Omega(1).$$ (10)

On the other hand, it follows from the assumed JL property of $P$ that with probability $1 - o(1)$, for all $1 \leq i < j \leq d$, we have that

$$\|P(e_i + e_j)\|_2^2 - 2\leq 2c,$$

Therefore from combining equality \textbf{(9)} with inequality \textbf{(10)} it follows that $\sqrt{x_{ij}} \leq 4c$ must hold for all $i \neq j$, or equivalently $|C_i \cap C_j| \leq z$ with $z = 16c^2$ for all $i \neq j$.

If $z < 1$, then the $C_i$ are pairwise disjoint and therefore $k \geq dc \geq d$, a contradiction. Thus $z \geq 1$, and hence $c \geq \frac{1}{16c^2}$ immediately. In what follows we strengthen the latter lower bound for a large range of $d$ and $k$ as claimed.

If $c \geq \frac{1}{16c^2}$ then the lemma clearly holds as $\Omega(\frac{1}{c^2})$ is the largest of the lower bounds claimed.

Now note that $c \geq 2$. Since if $c = 1$ were to hold, then from $c < 1/4$ it follows that $z < 1$, which is a contradiction as before. Therefore if $c \leq \frac{1}{8c^2}$ then all $C_i$’s are distinct as $z = 1 = (c/2)(32c^2 + 1) \leq c/2 + c/2 = c$ holds.

Observe that any $z + 1$ element set is contained in at most one $C_i$, Therefore the number of distinct $C_i$ is at most

$$f(k, c, z + 1) = \left( \frac{k}{z + 1} \right)^z \left( \frac{c}{z + 1} \right)$$,

a well known fact from block designs and set packing \cite{13}. From the Stirling formula, for all $n \geq 1$, $\sqrt{2\pi n} \left( \frac{n}{e} \right)^n \leq n! \leq 1.1 \sqrt{2\pi n} \left( \frac{n}{e} \right)^n$, and it follows that for all $1 \leq y < x$ it holds that

$$\left( \frac{y}{y} \right)^y \leq \frac{1}{\sqrt{2\pi y}} \sqrt{\frac{e}{y-x}}$$.

Therefore we have that

$$f(k, c, z + 1) \leq \left( \frac{k}{c} \right)^{z+1} 2k \leq k^{z+3}. \quad (11)$$

Now observe that $d \leq f(k, c, z + 1)$ as all $C_i$ are distinct. Combining the latter with inequality \textbf{(11)}, we have that $\log_2(d) \leq 3z$. Recalling that $1 \leq z = 16c^2$ concludes the proof. \hfill \blacksquare

Using a replication argument it is easy to see that if a matrix $P$ only has the JL property for vectors $x$ with $\|x\|_2^2 \leq \alpha$ for some $\alpha$, then under the conditions of Theorem \cite{14} we have that at least one column of $P$ contains $\Omega(\alpha^2 \min \left\{ \frac{1}{4}, \frac{1}{\log_2(d)} \right\})$ non-zeroes.

If the fourth moment of the random entries per column scales with the number of non-zeroes per column, the next theorem strengthens the previous claim by bounding the average number of non-zeroes per column. This condition is satisfied, say, if the non-zero entries are independent scaled $\pm 1$ random variables.
Theorem 15 Let \( 0 < \epsilon \leq 1/4 \) and \( M \) be an arbitrary \( k \times d \) 0-1 matrix with \( 2k^2 < d \). Let \( c_i \) denote the number of non-zeros in the \( i \)th column of \( M \). Let \( P \) be a \( k \times d \) random matrix of the following form \( P_{ij} = \begin{cases} 0 & \text{if } M_{ij} = 0 \\ U_{ij} & \text{if } M_{ij} = 1. \end{cases} \) Here the vector valued \( U \) is a vector of random variables independent and for each \( j \) it holds that \( E[\sum_i P_{ij}^2] = 1 \) and \( E[\sum_i P_{ij}^4] = O(1/c_i) \).

If \( P \) has JL property with probability at least \( 1 - \frac{e(1)}{2} \), then
\[
\sum_{i=1}^{d} \frac{c_i}{d} \geq \Omega \left( \frac{1}{c^2} \sqrt{\log_d(d)} \right). 
\]

**Proof.** Set
\[
s = \Omega \left( \frac{1}{c^2} \sqrt{\log_d(d)} \right). 
\]

For all \( j = 1, \ldots, k \), assemble the columns of \( P \) with \( c_i \) into the \( k \times n_j \) matrix \( P_j \). For all \( j \), if \( n_j > k \) then from assumed JL property of \( P \) it follows that \( P_j \) satisfies the conditions of Theorem 14 with \( c^2 = j \) and thus \( j > s \).

Therefore for all \( j < s \) we have that \( n_j \leq k \). The number of non-zeros in \( P \) is
\[
\sum_{i=1}^{d} c_i = \sum_{j=1}^{k} n_j, 
\]
which we lower bound as follows
\[
\sum_{j=1}^{k} n_j \geq \sum_{j=1}^{s-1} n_j s 
\]
= \( (d - sk) s \) \( \geq (d - k^2) s \) \( \geq \frac{d}{2} s \). \( \Box \)

6. EMBEDDING INTO \( \ell_1 \)

We can show the following result for the case that the target metric is \( \ell_1 \). The result and the corresponding proof is similar to that of Ailon and Chazelle [2]. We construct the matrix \( H \) as follows: \( H_{ij} = \delta_{a(j),j} r_j \), where \( r_j \) are now drawn i.i.d. random variables \( N(0, 1) \) instead of being \( \pm 1 \). We then have the following. Let \( \beta_0 = E[||z||] \) where \( z \sim N(0, 1) \). By the 2-stability of the normal distribution, \( Y_1 = \sum_j x_j r_j \delta_{a(j)} \sim N(0, \sigma^2) \) where
\[
\sigma^2 = \sum_{j=1}^{k} x_j^2. 
\]
Thus, \( E_r [||Y_1||] = \sigma \beta_0 \).

**Theorem 16** There exists a constant \( c_0 \) such that for all \( \epsilon < c_0 \), if \( c = k/\epsilon \), and \( k = O \left( \frac{1}{\epsilon} \log (\frac{1}{\epsilon}) \right) \), \( Y = \sum_{j=1}^{k} x_j \), we have that \( \Pr[|Y| - 1| > \epsilon] < \delta \).

The proof is omitted in this version.

7. DISCUSSIONS

The most important open question is resolving the gap between the upper and lower bounds with respect to the error probability. It would be interesting to see whether our claims could be proven more directly using stronger concentration inequalities.

Application of the current result to streaming settings would also require proving the claims for a \( k \)-wise independent hash-function and \( \pm 1 \) variables. The chief hurdle in applying the techniques of Clarkson and Woodruff [14] seems to be proving the FKG inequality for the limited independence case. Note that Nisan’s pseudo-random number generator construction [23] can be used to derandomize the hash function, but the naive way of doing this increases the update time to \( k \). We leave efficient derandomization as an open question.

It is worthwhile to note that the hash-function represents a bipartite expander. In a similar vein, Berinde et al. [10] use an unbalanced expander graph based construction to create matrices with restricted isometry property for sparse signal recovery. Their argument crucially uses two facts — that the error-norm is \( \ell_1 \), and that the input vector is sparse. It would interesting to investigate possible connections between these results.

Acknowledgments. The authors would like to thank Flavio Chierichetti, Edo Liberty, and Alex Smola for helpful discussions. We also thank the anonymous reviewers for feedback and for suggesting future directions.

8. REFERENCES

[1] D. Achlioptas. Database-friendly random projections: Johnson–Lindenstrauss with with binary coins. Journal of Computer and System Sciences, 66(4):671–687, 2003.

[2] N. Ailon and B. Chazelle. The fast Johnson–Lindenstrauss transform and approximate nearest neighbors. SIAM Journal on Computing, 39(1):302–322, 2009.

[3] N. Ailon and E. Liberty. Fast dimension reduction using Rademacher series on dual BCH codes. Discrete and Computational Geometry, 42(4):615–630, 2009.

[4] N. Ailon. Problems and results in extremal combinatorics, Part I. Discrete Math, 273:31–53, 2003.

[5] N. Ailon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. Journal of Computer and System Sciences, 58(1):137–147, 1999.

[6] N. Ailon and J. H. Spencer. The Probabilistic Method. Wiley-Interscience, 2nd edition, 2000.

[7] R. I. Arriaga and S. Vempala. An algorithmic theory of learning: Robust concepts and random and random projection. In Proc. 40th IEEE Symposium on Foundations of Computer Science, pages 616–623, 1999.

[8] M. F. Balcán, A. Blum, and S. Vempala. Kernels as features: On kernels, marginals, and low-dimensional mappings. Machine Learning, 65(1):79–94, 2006.

[9] B. Berger. The fourth moment method. SIAM Journal on Computing, 26(4):1188–1207, 1997.

[10] B. Berinde, A. Gilbert, P. Indyk, H. Karloff, and M. Strauss. Combining geometry and combinatorics: A unified approach to sparse signal recovery. In 46th Annual Allerton Conference, pages 798–805, 2008.

[11] E. J. Candès and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? IEEE Transactions on Information Theory, 52(12):4845–4852, 2006.

[12] M. Charikar, K. Chen, and M. Farach-Colton. Finding frequent items in data streams. Theoretical Computer Science, 312(1):3–15, 2004.

[13] K. L. Clarkson. Tighter bounds for random projections of manifolds. In Proc. 24th Annual Symposium on Computational geometry, pages 39–48, 2008.

[14] K. L. Clarkson and D. P. Woodruff. Numerical linear algebra in the streaming model. In Proc. 41st Annual ACM Symposium on Theory of Computing, pages 205–214, 2009.

[15] G. Cormode and M. Garofalakis. Sketching streams through the net: Distributed approximate query tracking. In Proc. 31st International Conference on Very Large Data Bases, pages 13–24, 2005.

[16] S. Dasgupta and A. Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. Random Structures and Algorithms, 22(1):60–66, 2003.

[17] P. Drineas, M. W. Mahoney, S. Muthukrishnan, and T. Sarlós. Faster least squares approximation. Arxiv preprint arXiv:0710.1435, 2007.

[18] Z. Füredi and Z. Katona. Multiply intersecting families of sets. Journal of Combinatorial Theory, Series A, 106(2):315–326, 2004.

[19] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In Proc. 20th Annual ACM Symposium on Theory of Computing, pages 604–613, 1998.

[20] W. B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. Contemporary Mathematics, 26:189–206, 1984.
9. APPENDIX

9.1 Proof of Lemma 6

PROOF. We show that $\sigma_1^2 - \frac{1}{k} \leq \sigma_2^2$, with probability $1 - \delta/k$; the proof will then follow from the union bound.

Define the random variable $X_j = \delta_{1h(j)} x_j^2 - \frac{2}{k}$. Then, $E_h[X_j] = 0$ and since $||x||_\infty < \frac{1}{\sqrt{k}}$, we have $X_j < \frac{1}{k}$. We also have $E_h[X_j^2] = \mathbb{E}_h \left[ \left( \delta_{1h(j)} x_j^2 - \frac{2}{k} \right)^2 \right] = \frac{x_j^2}{k} \left( \frac{1}{k} - \frac{1}{k^2} \right)$.

$$\sum_j \mathbb{E}_h[X_j^2] \leq \sum_j \frac{x_j^2}{c} \left( \frac{1}{k} - \frac{1}{k^2} \right) \leq \frac{1}{c} \left( \frac{1}{k} - \frac{1}{k^2} \right).$$

Also, $\sum_j X_j = \sigma_1^2 - \frac{1}{k}$. Plugging this into the Bernstein’s inequality [24], Theorem 2.7),

$$\Pr \left[ \sigma_1^2 - \frac{1}{k} > \frac{\alpha}{k} \right] = \Pr \left[ \sum_j X_j > \frac{\alpha}{k} \right] \leq \exp \left( -\frac{\alpha^2/2}{(k-1) + \alpha/3} \right) \leq \exp \left( -\frac{\alpha^2/2}{k + \alpha/3} \right).$$

Since $\alpha \geq 3$,

$$\Pr \left[ \sigma_1^2 - \frac{1}{k} > \frac{\alpha}{k} \right] \leq \exp \left( -\frac{\alpha^2/2}{2\alpha k/3} \right) \leq \exp \left( -\frac{3\alpha}{4k} \right).$$

Choosing $c = \frac{4k}{\alpha} \log\left( \frac{1}{\delta} \right)$, we get the above probability to be smaller than $\delta/k$. Since $\alpha = \epsilon \log(k/\delta)$, and $k = \frac{12}{c^2} \log(1/\delta)$, we have that choosing $c = \frac{4k}{\alpha} \log\left( \frac{1}{\delta} \right)$ is sufficient. 

9.2 Bounding the MGF’s

We first compute the expectation of the MGF for different conditions on the hashing function. We begin by proving Lemma 7.

9.2.1 Proof of Lemma 7

PROOF. We have that $Z_i = \sum_{j \neq g \in h \backslash \{i\}} r_{jg} x_j x_g$. Hence

$$Z_i = \left( \sum_{j \in h \backslash \{i\}} r_{j} x_j \right)^2 - \sum_{j \in h \backslash \{i\}} x_j^2 = Y_i^2 - \sigma_i^2,$$

where $Y_i = \sum_{j \in h \backslash \{i\}} x_j r_j$. Then,

$$E_r[\exp(u Y_i)] = \prod_{j \in h \backslash \{i\}} E_r[\exp(u r_j x_j)] = \prod_{j \in h \backslash \{i\}} \left( \frac{1}{2} \exp(u x_j) + \frac{1}{2} \exp(-u x_j) \right) \leq \prod_{j \in h \backslash \{i\}} \exp \left( \frac{u^2 x_j^2}{2} \right) \leq \exp \left( \frac{u^2 \sigma_i^2}{2} \right).$$

By the Markov inequality, we get the probability of $Y_i$ being larger than $t$ as

$$\Pr [Y_i > t] \leq \frac{E_r[\exp(u Y_i)]}{\exp(ut)} \leq \exp \left( \frac{u^2 \sigma_i^2}{2 ut} \right) \leq \exp \left( -\frac{t^2}{2 \sigma_i^2} \right),$$

by choosing $u = \frac{t}{\sigma_i}$. Note that we do not need to worry about $\sigma_i$ being zero, as then $Y_i = 0$. Then, we bound $E_r[\exp(u Z_i)]$ as follows. Denote $p(t) = \Pr[Z_i = t]$. We first compute the expectation with respect to $r$. For any value of $\theta > 0$, we have

$$E_r[\exp(u Z_i)] = \sum_{t \in (-\infty, \infty)} \exp(ut)p(t) \leq \sum_{t \in (-\infty, \theta]} \exp(ut)p(t) + \sum_{t > \theta} \exp(ut)p(t).$$
The first term can be bounded as follows:

\[
\sum_{t \in (-\infty, \theta)} \exp(ut)p(t) \leq \sum_{t \in [0, \theta]} \left(1 + ut + \sum_{j=2}^{\infty} \frac{u^j t^j}{j!} \right) p(t)
\]

\[
\leq \sum_{t \in (-\infty, \theta]} p(t) + \sum_{t \in (-\infty, \theta]} ut p(t) + \sum_{t \in (-\infty, \theta]} \sum_{j=2}^{\infty} \frac{u^j t^j}{j!} dF(t)
\]

\[
\leq \sum_{t \in (-\infty, \infty)} p(t) + \sum_{t \in (-\infty, \infty)} ut p(t) + \sum_{t \in (-\infty, \infty)} \sum_{j=2}^{\infty} \frac{u^j t^j}{j!} p(t),
\]

where the last inequality follows since in the range \([\theta, \infty]\), the integral is positive. Then, the calculation can be simplified as follows:

\[
\sum_{t \in (-\infty, \theta)} \exp(ut)p(t) \leq \sum_{t \in (-\infty, \infty)} (p(t) + ut p(t))
\]

\[
+ \sum_{t \in (-\infty, \theta)] \sum_{j=2}^{\infty} \frac{u^j t^j}{j!} p(t)
\]

\[
\leq 1 + 0 + \sum_{t \in (-\infty, \theta]} \sum_{j=2}^{\infty} \frac{u^j t^j}{j!} p(t)
\]

\[
\leq 1 + \frac{1}{\theta^2} (\exp(u\theta) - 1 - u\theta) E_r [Z_t^2].
\]

For the second term, we have

\[
\sum_{t > \theta} \exp(ut)p(t)
\]

\[
\leq \sum_{t=u}^{\infty} \exp(\ell + 1) \left( \Pr \left[ Z_i > \frac{\ell}{u} + 1 \right] \right)
\]

\[
\leq \sum_{t=u}^{\infty} \exp(\ell + 1) \Pr \left[ Z_i > \frac{\ell}{u} + 1 \right]
\]

\[
\leq \sum_{t=u}^{\infty} \exp(\ell + 1) \Pr \left[ Y_i^2 > \frac{\sigma^2}{u} + \frac{\ell}{u} + 1 \right]
\]

\[
\leq \sum_{t=u}^{\infty} \exp(\ell + 1) \exp \left( -\frac{\ell}{2u \sigma_i^2} \right)
\]

\[
\leq \sum_{t=u}^{\infty} \exp \left( \ell + 1 - \frac{\ell}{2 \sigma_i^2} \right)
\]

\[
\leq \sqrt{\theta} \sum_{t=u}^{\infty} \exp \left( \ell - \frac{\ell}{2 \sigma_i^2} \right).
\]

By assumption of the lemma, since \(u < \frac{\ell}{4 \sigma_i^2}\), we have that \(\ell - \frac{\ell}{2 \sigma_i^2} < -\frac{\ell}{4 \sigma_i^2}\). With this restriction,

\[
\sum_{t > \theta} \exp(ut)p(t) \leq \sqrt{e} \sum_{t=u}^{\infty} \exp \left( -\frac{u \theta}{4 \sigma_i^2} \right)
\]

\[
\leq 2 \sqrt{e} \exp \left( -\frac{u \theta}{4 \sigma_i^2} \right) = 2 \sqrt{e} \exp \left( -\frac{\theta}{4 \sigma_i^2} \right) \leq 4 \exp \left( -\frac{\theta}{4 \sigma_i^2} \right).
\]

By putting together the two parts, we have that

\[
E_r [\exp(uZ_t)] \leq \left( 1 + \frac{1}{\theta^2} (\exp(u\theta) - 1 - u\theta) E_r [Z_t^2] \right)
\]

\[
+ 4 \exp \left( -\frac{\theta}{4 \sigma_i^2} \right).
\]

Choosing \(\theta = 4\sigma_i^2 \ln(k/\delta)\), the proof is complete. \(\square\)

9.3 Continued proof of Lemma 9

We finish the proof of Lemma 9 by showing that \(g_s\) is decreasing. To this end, we prove that for all \(A \subseteq [d]\) and for all \(a \in [d] \setminus A\), \(g_s(A \cup \{a\}) \leq g_s(A)\). Recalling the definition of \(g_s(A)\), we have

\[
g_s(A) = E \left[ \exp \left( u \sum_{i=1}^{s} Z_i \right) \mid h^{-1}(s) = A \right]
\]

\[
= E \left[ \exp \left( u \sum_{i=1}^{s} Z_i \right) \mid \forall j : h(j) \right] \mid h^{-1}(s) = A \right],
\]

(12)

where the inner expectation is over the random variables \(\{r_j\}\) only. Since \(h\) is completely independent we have that

\[
g_s(A) = \sum_{(h_1, \ldots, h_d) \in [k]^d \setminus A} E \left[ \exp \left( u \sum_{i=1}^{s} Z_i \right) \mid \forall j : h(j) = h_j \right]
\]

\[
= \sum_{(h_1, \ldots, h_d) \in [k]^d \setminus A} E \left[ \exp \left( u \sum_{i=1}^{s} Z_i \right) \mid \forall j : h(j) = h_j \right]
\]

\[
\leq \sum_{(h_1, \ldots, h_d) \in [k]^d \setminus A} E \left[ \exp \left( u \sum_{i=1}^{s} Z_i \right) \mid \forall j : h(j) = h_j, h(a) = s \right].
\]

Therefore it is sufficient to show that for all

\[
(h_1, \ldots, h_{a-1}, h_{a+1}, \ldots, h_d) \in [k]^{d-1}
\]

with \(\forall j \neq a : h_j = s \Leftrightarrow j \in A\) it holds that

\[
\sum_{h_a \in [k] \setminus \{s\}} E \left[ \exp \left( u \sum_{i=1}^{s-1} Z_i \right) \mid \forall j : h(j) = h_j \right]
\]

\[
\geq E \left[ \exp \left( u \sum_{i=1}^{s-1} Z_i \right) \mid \forall j : h(j) \neq a : h(j) = h_j, h(a) = s \right].
\]

We shall prove the following stronger inequality: for all \((h_1, \ldots, h_{a-1}, h_{a+1}, \ldots, h_d) \in [k]^{d-1}\) with \(\forall j \neq a : h(j) = s \Leftrightarrow j \in A\) and for all \(h_a \in [k] \setminus \{s\}\) it holds that

\[
E \left[ \exp \left( u \sum_{i=1}^{s-1} Z_i \right) \mid \forall j : h(j) = h_j \right]
\]

\[
\geq E \left[ \exp \left( u \sum_{i=1}^{s-1} Z_i \right) \mid \forall j : h(j) \neq a : h(j) = h_j, h(a) = s \right].
\]

Now observe that only \(r\) are random in the above expectations and that \(Z_i\) are conditionally independent given \(h\). Therefore,

\[
E \left[ \exp \left( u \sum_{i=1}^{s-1} Z_i \right) \mid h \right] = \prod_{i=1}^{s-1} E \left[ \exp \left( u Z_i \right) \mid h \right].
\]
From the non-negativity of the exponential function, it follows that it is sufficient to show that for all \( i = 1, \ldots, s - 1 \) and for all \( (h_1, \ldots, h_{a-1}, h_{a+1}, \ldots, h_d) \in [k]^{d-1} \) with \( \forall j \neq a : h_j = s \Leftrightarrow j \in A \) and for all \( h_a \in [k] \setminus \{a\} \) it holds that

\[
E_L \geq E_R, \text{ where } \quad E_L = E \left[ \exp \left( u \hat{Z}_i \right) \mid \forall j : h(j) = h_i \right], \\
E_R = E \left[ \exp \left( u \hat{Z}_i \right) \mid \forall j \neq a : h(j) = h_j, h(a) = s \right].
\]

We prove inequality \( \text{(13)} \) by a case analysis. If \( h_a \neq i \), then \( E_L = E_R \) by definition. If \( h_a = i \) and the \( i \)th bucket of \( E_L \)'s hash function is bad, then \( E_L = G(u, \sigma^*_a^2) \geq E_R \), as shown earlier in Corollary 8. If \( h_a = i \) and the \( i \)th bucket of \( E_L \)'s hash function is good then, the \( i \)th bucket of \( E_R \)'s hash function is also good. As before, define

\[
V_a = \sum_{j \neq a} \sum_{u \neq h_a, j \neq a} r_j g_j x_j g_h(j) \delta_h(j),
\]

and

\[
W_a = \sum_{j \neq a} r_j x_j g_h(j) \delta_h(j).
\]

Again, note that if the \( i \)th bucket is good as assumed then \( \hat{Z}_i = Z_i = V_a + r_a W_a \). Therefore we have that

\[
E_L = E \left[ \exp \left( u V_a + u \cdot r_a W_a \right) \mid \forall j : h(j) = h_i, \forall j \neq a : r_j \right] \left[ \forall j : h(j) = h_i \right].
\]

Now observe that only \( r_a \) is random in the inner expectation and

\[
E \left[ u V_a + u \cdot r_a W_a \mid \forall j : h(j) = h_i, \forall j \neq a : r_j \right] = u V_a.
\]

Thus from \( E[\exp(x)] \geq \exp(E[x]) \), it follows that

\[
E \left[ \exp \left( u V_a + u \cdot r_a W_a \right) \mid \forall j : h(j) = h_i, \forall j \neq a : r_j \right] \geq \exp \left( u V_a \right)
\]
as before. Plugging the latter into \( \text{(13)} \) we arrive at

\[
E_L \geq E \left[ \exp \left( u V_a \right) \mid \forall j : h(j) = h_i \right] = E \left[ \exp \left( u V_a \right) \mid \forall j \neq a : h(j) = h_j, h(a) = s \right].
\]

Here the last equality follows from the fact that all \( r \) and \( h \) values are independent and since \( V_a \) does not depend on \( a \). If \( h^{-1}(s) = A \cup \{a\} \) and the \( i \)th bucket is good as assumed then \( \hat{Z}_i = Z_i = V_a \) and we observe that

\[
E \left[ \exp \left( u V_a \right) \mid \forall j \neq a : h(j) = h_j, h(a) = s \right] = E_R
\]

Combining \( \text{(15)} \) and \( \text{(16)} \), we conclude that \( E_L \geq E_R \) for all cases and hence \( g_i \) is decreasing as claimed.

9.4 Proof of Theorem 11(ii)

**Proof.** Recall that the random variable \( Z_i \) is defined as

\[
\hat{Z}_i = \begin{cases} Z_i & \text{if } g_i, \\ \frac{1}{u} \log G(u, \sigma^*_a^2) & \text{else.} \end{cases}
\]

Note that \( G(u, \sigma^*_a^2) > 1 \), and hence for \( u > 0 \), \( \frac{1}{u} \log G(u, \sigma^*_a^2) > 0 \). Also recall that \( g_i \) is the indicator vector for bucket \( i \) being good, and \( \hat{S} \) is the indicator for the hash function being good. By definition of \( Z_i \), since \( \frac{1}{u} \log G(u, \sigma^*_a^2) > 0 \), \( \sum_i Z_i \wedge \hat{S} \leq \sum_i \hat{Z}_i \) and hence we have that

\[
\Pr \left[ \sum_i Z_i > \epsilon \right] \leq \Pr \left[ \sum_i \hat{Z}_i > \epsilon \wedge \hat{S} \right] + \Pr[\hat{S}]
\]

\[
\leq \Pr \left[ \sum_i \hat{Z}_i > \epsilon \right] + \delta.
\]

Thus, we prove a bound on \( E[\exp(u \sum_i \hat{Z}_i)] \) and thus bound on \( \Pr[\sum_i \hat{Z}_i > \epsilon] \). Taking expectations over both \( r \) and \( h \), and using \( \Pr[\hat{S}] < \delta \),

\[
E[\exp(u \hat{Z}_i)] \leq E[\exp(u \hat{Z}_i) \mid \hat{S}] \Pr[\hat{S}] + G(u, \sigma^*_a^2) \Pr[\hat{S}]
\]

\[
\leq 1 + \frac{1}{\theta^2} (\exp(u\theta) - 1 - u\theta) \left( \frac{1}{k^2(1 - \delta)} + \sigma^*_a^2 \delta \right) + \frac{4\delta}{k}
\]

where we combined the appropriate terms from the two parts of the sum. Recall that \( \sigma^*_a^4 < 1 \). By choosing \( \delta < \frac{1}{k^2} \), we have that

\[
E[\exp(u \hat{Z}_i)] \leq 1 + \frac{1}{\theta^2} (\exp(u\theta) - 1 - u\theta) \frac{2}{k^2} + \frac{4\delta}{k}
\]

\[
\leq \exp \left( \frac{2}{k^2\theta^2} (\exp(u\theta) - 1 - u\theta) + \frac{4\delta}{k} \right).
\]

Taking the product over the \( k \) terms, by using Lemma 10

\[
E \left[ \prod_i \exp(u \hat{Z}_i) \right] \leq \prod_i E[\exp(u \hat{Z}_i)]
\]

\[
\leq \exp \left( \frac{2}{k^2\theta^2} (\exp(u\theta) - 1 - u\theta) + \frac{4\delta}{k} \right).
\]

For completeness, we show how to determine the optimal \( \theta \).

\[
\Pr \left[ \sum_i \hat{Z}_i > \epsilon \right] \leq \frac{E[\exp(u \sum_i \hat{Z}_i)]}{\exp(u\epsilon)} \leq \prod_i E[\exp(u \hat{Z}_i)] \exp(-u\epsilon)
\]

\[
\leq \exp \left( \frac{2}{k^2\theta^2} (\exp(u\theta) - 1 - u\theta) + \frac{4\delta}{k} \right) \leq \exp \left( H(u) + \delta \right),
\]

where we define \( H(u) = \frac{1}{4\sigma^*_a^2}(\exp(u\theta) - 1 - u\theta) - \frac{u}{2} \). Setting the derivative \( H'(u) = 0 \), we get

\[
\frac{1}{4\sigma^*_a^2}(\exp(u\theta) - 1 - u\theta) = \frac{u}{2},
\]

and hence

\[
u = \frac{1}{\theta} \ln \left( 1 + \frac{k \theta}{4} \right) = \frac{1}{4 \sigma^*_a^2} \ln(k/\delta).
\]

Note that we need to restrict \( u < \frac{1}{4 \sigma^*_a^2} \). We need

\[
\frac{\ln(1 + \frac{k \theta}{4} \ln(k/\delta))}{\ln(k/\delta)} < 1,
\]

which is true if setting

\[
\frac{\kappa x}{2} \cdot \frac{\ln(1 + \frac{k \theta}{4} \ln(k/\delta))}{\ln(k/\delta)} < \frac{k \theta}{2} \quad \text{or} \quad \epsilon < \frac{\ln(k/\delta)}{2(1 + \frac{k \theta}{4})}.
\]

which is permissive. Using this value of \( u \), we have that (skipping the simplifications)

\[
H = \frac{2}{k^2\theta^2} \left( \frac{k \theta}{2} - \frac{1 + \frac{k \theta}{4} \theta}{2} \right) \ln \left( 1 + \frac{k \theta}{2} \right) \ln \left( 1 + \frac{k \theta}{2} \right)
\]

\[
\leq \frac{2}{k^2\theta^2} \frac{k^2 \theta^2}{2 + \frac{k \theta}{4}} \leq \frac{-k \theta^2}{2(2 + \frac{k \theta}{4})}
\]

which is the trick that Bernstein uses: \((1 + x) \ln(1 + x) - x \leq \frac{x^2}{2 + 2x/3} \) Plugging in this value of \( H \), we get that

\[
\Pr \left[ \sum_i \hat{Z}_i > \epsilon \right] \leq \exp \left( \frac{-3k \theta^2}{4(1 + \alpha) \ln(k/\delta)} + \frac{4\delta}{k} \right).
\]

The proof of Theorem 11(ii) is similar to the above and is omitted in this version.