These lecture notes are based on a manuscript of a book on matrix rigidity by Chi-Ning Chou and Sasha Golovnev.

Chapter 1

Introduction

1.1 Definitions and examples

Lecture 1

One of the main questions in computational complexity is that of proving lower bounds on the size of Boolean circuits computing explicitly given functions. While most Boolean functions of n inputs require circuits of size $2^n/n$ [Sha49a, Lup59a], we can only prove small linear lower bounds for explicitly defined functions [LR01, IM02, Blu83, DK11, FGHK16]. [1]

The same question remains open for *linear* circuits computing *linear* Boolean functions. Since any linear function with one output can be computed by a circuit of size at most n, we study linear functions with n inputs and n outputs. A random linear map with n outputs requires circuits of size $n^2/\log n$ [Lup56], but the best known lower bound for an explicit linear map is only $3n - o(n)$ [Cha94a].

The notion of matrix rigidity was introduced by Valiant [Val77] as a tool for proving lower bounds against linear circuits. (A related notion of separability was introduced by Grigoriev [Gri76].)

We will use the following notation. A matrix A is called s-sparse, if the number of non-zero entries in A is at most s. We will use $I_n, 0_n$ and J_n to denote the identity matrix, zero matrix, and all-ones matrix of size $n \times n$. For a matrix $A \in \mathbb{F}^{n\times n}$, by $\|A\|_0$ we denote the number of non-zero entries in A.

Definition 1.1 (Rigidity)**.** *Let $\mathbb{F}$ be a field, $A \in \mathbb{F}^{n\times n}$ be a matrix, and $0 \le r \le n$. The rigidity of A over $\mathbb{F}$, denoted by $\mathcal{R}_A^{\mathbb{F}}(r)$, is the Hamming distance between A and the set of matrices of rank at most r. Formally,*

$$\mathcal{R}_A^{\mathbb{F}}(r) := \min_{\mathrm{rank}(A+C)\le r} \|C\|_0 \ .$$

In other words, a matrix A has rigidity $\mathcal{R}_A^{\mathbb{F}}(r) \ge s$ if and only if $A \in \mathbb{F}^{n\times n}$ *cannot* be written as a sum

$$A = S + L \ ,$$

where $S \in \mathbb{F}^{n\times n}$ is $(s-1)$-sparse matrix, and $L \in \mathbb{F}^{n\times n}$ is low-rank: $\mathrm{rank}(L) \le r$.

Valiant [Val77] proved that any linear map $A \in \mathbb{F}^{n\times n}$ computed by a linear circuit (over a field $\mathbb{F}$) of depth $O(\log n)$ and size $o(n \log\log n)$ has rigidity at most $\mathcal{R}_A^{\mathbb{F}}(\varepsilon n) \le n^{1+\delta}$ for every constant $\varepsilon, \delta > 0$. Therefore, an explicit matrix of higher rigidity would give us a super-linear lower bound against linear circuits of logarithmic depth. Despite more than 40 years of research, the problem of proving super-linear lower bounds for such circuits remains open.

Let us now see the rigidity of a few specific matrices.

- If $A \in \mathbb{F}^{n\times n}$ has rank $\mathrm{rank}(A) = k$ over the field $\mathbb{F}$, then $\mathcal{R}_A^{\mathbb{F}}(r) = 0$ for every $r \ge k$. Indeed, A can be written as a sum of A and 0_n, where $\mathrm{rank}(A) \le r$ and 0_n is 0-sparse. Similarly, an s-sparse matrix $A \in \mathbb{F}^{n\times n}$ has rigidity $\mathcal{R}_A^{\mathbb{F}}(r) \le s$ for any value of r.

[1] Here by explicit functions we mean functions computable in time polynomial in n. We will later discuss the notion of explicitness in greater detail.

- For any $0 \le r \le n$, $\mathcal{R}^{\mathbb{F}}_{I_n}(r) = n - r$. Indeed, if we change $n - r$ ones of I_n to zeros, then the resulting matrix has rank r, which implies that $\mathcal{R}^{\mathbb{F}}_{I_n}(r) \le n-r$. On the other hand, for any $(n-r)$-sparse matrix B, from subadditivity of rank,

$$\mathrm{rank}(I_n + B) \ge \mathrm{rank}(I_n) - \mathrm{rank}(B) \ge n - (n - r) = r \ ,$$

which gives us that $\mathcal{R}^{\mathbb{F}}_{I_n}(r) \ge n - r$.

- Let n be a multiple of $2r$, and let $M_n \in \mathbb{F}^{n\times n}$ be a matrix consisting of matrices I_{2r} stacked together side by side:

$$M_n = \begin{pmatrix} I_{2r} & \cdots & I_{2r} \\ \vdots & \ddots & \vdots \\ I_{2r} & \cdots & I_{2r} \end{pmatrix} .$$

We will show that this matrix has rigidity $\mathcal{R}^{\mathbb{F}}_{A}(r) = \frac{n^2}{4r}$.

Theorem 1.2 ([Mid05])**.** *For any field $\mathbb{F}$, and any n divisible by $1 \le 2r \le n$,*

$$\mathcal{R}^{\mathbb{F}}_{M_n}(r) = \frac{n^2}{4r} .$$

Proof of Theorem 1.2. M_n consists of $\frac{n^2}{4r^2}$ copies of the identity matrix I_{2r}. In order to drop the rank of A to r, the rank of each copy of I_{2r} must be dropped to r. From the previous example we know that in order to decrease the rank of I_{2r} to r, one needs to change at least r elements. Thus, $\frac{n^2}{4r^2} \cdot r = \frac{n^2}{4r}$ entries of M_n must be changed. Note that this bounds is tight, *i.e.*, $\mathcal{R}^{\mathbb{F}}_{M_n}(r) = \frac{n^2}{4r}$. □

The bound of Theorem 1.2 easily generalizes to all values $r \le n/2$ with a loss of a multiplicative factor of 2. This theorem was proven by Midrijānis [Mid05], and it gives a simple matrix with rigidity $\mathcal{R}^{\mathbb{F}}_{M_n}(r) \ge \frac{n^2}{8r}$.

We will see later that there exist matrices with much higher rigidity $\tilde{\Omega}\left((n-r)^2\right)$. Embarrassingly, the best known lower bound for an *explicit* matrix improves on the $\frac{n^2}{8r}$ bound only by a logarithmic factor.

1.2 Circuit Complexity

A circuit corresponds to a simple straight line program where every instruction performs a binary operation on two operands, each of which is either an input or the result of a previous instruction. The structure of this program is extremely simple: no loops, no conditional statements. Still, we know no functions in P (or even NP, or even E^{NP}) that requires even $3.1n$ binary instructions ("size") to compute on inputs of length n. This is in sharp contrast with the fact that it is easy to *non-constructively* find such functions: simple counting arguments show a random function on n variables has circuit size $\Omega(2^n/n)$ with probability $1 - o(1)$ [Sha49b, Lup59b].

For small-depth circuits we know several strong lower bounds. (Note that when working with circuits of constant depth, we do not pose bounds on the fan-ins of the gates.) Depth-2 circuits (after a simple normalization) are just CNFs or DNFs. It is easy to see that the parity function $\oplus_n$ of n inputs requires CNFs and DNFs of size $\Omega(2^n)$. For depth-d circuits, we know a lower bound of $2^{\Omega(n^{(1/(d-1))})}$ [Hås86, HJP93, PPZ97, Bop97, PPSZ05, MW17]. Thus, for depth $d = o(\log n/\log\log n)$ we have non-trivial lower bounds even if the fan-ins of the gates are unbounded. For circuits with fan-in 2, we known functions which cannot be computed by circuits of depth $1.99 \log n$ [Nec66]. Thus, a problem on the frontier is

Problem 1.3. *Prove a lower bound of $10n$ against circuits of depth $10 \log n$.*
More generally, a lower bound of $\omega(n)$ against circuits of depth $O(\log n)$.

Super-linear lower bounds are not known even for linear circuits, i.e., circuits consisting of only gates computing linear combinations of their two inputs. Note that every linear function with one output has a circuit of size $n - 1$ (and depth $\log n$). For linear circuits, we consider *linear transformations*, multi-output functions of the form $f(x) = Ax$ where $A \in \mathbb{F}^{n\times n}$. For a random matrix $A \in \{0,1\}^{n\times n}$, the size of the smallest linear circuit computing Ax is $\Theta(n^2/\log n)$ [Lup56] with probability $1 - o(1)$, but for explicitly-constructed matrices the strongest known lower bound is $3n - o(n)$ [Cha94b]. This leads us to another problem on the frontier:

Problem 1.4. *Prove a lower bound of $\omega(n)$ against* linear *circuits of depth $O(\log n)$.*

Formally, Problem 1.3 and Problem 1.4 are incomparable, as in the linear case we study a weaker computational model (which makes it easier to prove lower bounds), but are limited to proving lower bounds for a smaller class of problem (which makes it harder to prove lower bounds).

1.3 Circuits and Rigidity

In this section, we will present a seminal result of Valiant [Val77] showing that rigid matrices require log-depth circuits of super-linear size. We start with the definition of linear circuits.

Definition 1.5 (Linear circuits)**.** *Let $\mathbb{F}$ be a field and $n \in \mathbb{N}$. A circuit C with n inputs and n outputs is a directed acyclic graph where n vertices have fan-in zero and are labeled by the inputs, all other vertices have fan-in two and are labeled with* affine functions *(over $\mathbb{F}$) of their two inputs, n of these vertices are labeled as outputs. For every fixed input, the value at each node is computed by applying the corresponding functions. Such a circuit C naturally defines a linear map $f : \mathbb{F}^n \to \mathbb{F}^n$, and the corresponding matrix $A \in \mathbb{F}^{n\times n}$ such that $f(x) = Ax$.*

The depth $d(C)$ of a circuit C is the length of the longest path in the circuit. The size $s(C)$ of C is defined as the number of vertices in C.

The following theorem shows a connection between lower bounds for linear circuits and matrix rigidity.

Theorem 1.6. *Let $\mathbb{F}$ be a field, and $A \in \mathbb{F}^{n\times n}$ be a family of matrices for $n \in \mathbb{N}$. If $\mathcal{R}^{\mathbb{F}}_A(\varepsilon n) > n^{1+\delta}$ for constant $\varepsilon, \delta > 0$, then any $O(\log n)$-depth linear circuit computing $x \to Ax$ must be of size $\Omega(n \cdot \log\log n)$.*

The proof of Theorem 1.6 repeatedly uses the following beautiful graph theoretic lemma due to Erdös, Graham, and Szemerédi [EGS76]: If G is a directed acyclic graph with s edges and of depth d, then there is a set of $s/\log d$ edges whose removal decreases the depth of G by a factor of two. We will follow the proof of this lemma from [Vio09].

Lemma 1.7 ([EGS76])**.** *Let G be an acyclic digraph with s edges and of depth $d = 2^k$. There exists a set of $s/\log d$ edges in G such that after their removal, the longest path in G has length at most $d/2$.*

Proof of Lemma 1.7. For ease of exposition, we follow [Vio09] and define a *depth function*. Let $G = (V,E)$ be an acyclic digraph. We say that $D : V \to \{0, 1, \ldots, d\}$ is a depth function for G if for any $(a, b) \in E$, $D(a) < D(b)$. It is not difficult to see that G has depth at most d if and only if there exists a depth function $D : V \to \{0, 1, \ldots, d-1\}$ for G.

We start with G of depth at most $d = 2^k$, and its depth function $D : V \to \{0, 1, \ldots, 2^k\}$. Now, consider the following partition of E using the depth function D. For each $i \in [k]$, define

$$E_i = \{(a, b) \in E : \text{ the most significant bit where } D(a), D(b) \text{ differ is the } i^{\text{th}} \text{ bit}\}.$$

As $\{E_i\}_{i\in[k]}$ is a partition of E, by the averaging argument, there exists $i^* \in [k]$ such that

$$|E_{i^*}| \le \frac{|E|}{k} \le \frac{|E|}{\log d}\,.$$

Now, it suffices to show that the depth of $G' = (V,E')$, where $E' = E \backslash E_{i^*}$, is at most 2^{k-1}. This can be shown by exhibiting a depth function $D' : V \to \{0, 1, \ldots, 2^{k-1} - 1\}$ for G'. The following shows that we can take $D'(v)$ to be $D(v)$ without the $i^{*\text{th}}$ bit.

Consider an edge $(a, b) \in E'$. Since $(a, b) \in E$, $D(a) < D(b)$. In particular, there exists $i \in [k]$ such that the most significant bit where $D(a)$ and $D(b)$ differ is i. Since $(a, b) \in E'$, the edge (a, b) was not removed, so $i \neq i^*$. Therefore, after removing the bit i^*, this bit i is still the most significant bit where $D'(a)$ and $D'(b)$ differ. This implies that $D'(a) < D'(b)$, and that $D' : V \to \{0, 1, \ldots, 2^{k-1} - 1\}$ is a depth function for G'. ∎

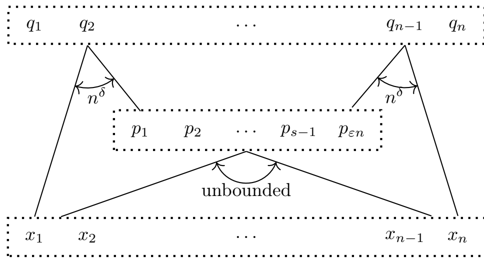

Figure 1.1: In order to compute the values of the outputs of the circuit C, first we precompute the values of εn removed edges (or vertices V'), and store them in variables p_i. Now each output q_j of the circuit C can be computed from n^δ inputs and precomputed bits. In particular, $A = BM + C$, where C encodes the dependence of the outputs q_i on the inputs x_j; B encodes the dependence of q_i on p_j; M encodes the dependence of p_i on x_j. Since C is sparse, and BM is low rank, the matrix A is not rigid.

Now we finish the proof of Theorem 1.6. **Lecture 2**

Proof of Theorem 1.6. We will show that for every constant $c_d \geq 2$, every circuit of depth at most $c_d \log n$ computing $x \to Ax$ must be of size at least $c_s n \log\log n$ for a constant $c_s = \frac{\varepsilon}{\log c_d + \log 1/\delta}$. Suppose, to the contrary, that there is a linear circuit C of size $s = c_s n \log\log n$ and depth $d = c_d \log n = 2^k$ that computes $x \to Ax$. Let G be the underlying acyclic digraph of C.

First, we apply Lemma 1.7 to G t times, and get a graph G' such that (i) only

$$s \cdot \left(\frac{1}{\log d} + \frac{1}{\log d - 1} + \cdots \frac{1}{\log d - (t-1)} \right) \leq \frac{st}{\log d - (t-1)}$$

edges are removed from G and (ii) the longest path in G' is of length at most $d' \leq d/2^t$.

By setting $t = \log c_d + \log 1/\delta$, the longest path in G' has length $\leq d/2^t = \delta \log n$, and the number of removed edges is at most

$$\frac{st}{\log d - (t-1)} = \frac{st}{\log d/2} \leq \frac{tc_s n \log\log n}{\log\log n} = \varepsilon n\,.$$

Now, let E be the set of removed edges and V' be the set of *tail* vertices of the edges from E. Since all paths in G' are no longer than d' and all in-degrees are at most 2, every output vertex in G' is now connected to at most $2^{d'}$ input variables. Therefore, every output is a (linear) function of at most $2^{d'}$ inputs and the functions computed at the removed edges (or the vertices V').

More specifically, let A_i be the i^{th} row of A, *i.e.,* the linear form computed by the i^{th} output vertex of G. Then A_i can be written as the following sum

$$A_i = \sum_{j \in [|V'|]} b_{ij} v_j + c_i$$

where v_j is the linear form computed by the j^{th} element in V' and c_i is the linear form computed by the i^{th} output vertex in G'. Note that since c_i only depends on at most $2^{d'}$ input variables.

Therefore, the matrix A can be written as follows.

$$A = BM + C$$

where $B \in \mathbb{F}^{n \times |V'|}$ consists of the coefficients b_{ij}, rows of $M \in \mathbb{F}^{|V'| \times n}$ compute linear forms of vertices from V', and $C \in \mathbb{F}^{n \times n}$ is a row sparse matrix where the number of non-zero entries in each row is at most $2^{d'} = n^\delta$.

The above argument gives us that $\tilde{\mathcal{R}}^{\mathbb{F}}_A(|V'|) = \tilde{\mathcal{R}}^{\mathbb{F}}_A(\varepsilon n) \leq n^\delta$, which contradicts the assumption on the rigidity of A.

□

1.4 Existence of Rigid Matrices

In this section, we will show that for any field $\mathbb{F}$, most of the $n \times n$ matrices have the highest possible rigidity for any rank parameter r.

It turns out that for every matrix A and field $\mathbb{F}$, there is a simple upper bound $\mathcal{R}_A^{\mathbb{F}}(r) \le (n-r)^2$. Valiant [Val77] showed that this upper bound is essentially tight for a random matrix. First, we give a proof of the upper bound.

Theorem 1.8 (Simple upper bound)**.** *For any field $\mathbb{F}$, matrix $A \in \mathbb{F}^{n\times n}$, and integer $0 \le r \le n$, we have that*

$$\mathcal{R}_A^{\mathbb{F}}(r) \le (n-r)^2 \ .$$

Proof of Theorem 1.8. If $\operatorname{rank}(A) \le r$, then $\mathcal{R}_A^{\mathbb{F}}(r) = 0 \le (n-r)^2$. Thus, it suffices to focus on the case where there is an $r \times r$ *full-rank* submatrix $B \in \mathbb{F}^{r\times r}$ of A. Without loss of generality, assume that B is located in the top left corner of A:

$$A = \begin{pmatrix} B & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \ , \tag{1.9}$$

where $A_{12} \in \mathbb{F}^{r\times(n-r)}, A_{21} \in \mathbb{F}^{(n-r)\times r}, A_{22} \in \mathbb{F}^{(n-r)\times(n-r)}$. In order to prove that $\mathcal{R}_A^{\mathbb{F}}(r) \le (n-r)^2$, we will show that it is possible to change the entries in $A_{22} \in \mathbb{F}^{(n-r)\times(n-r)}$ and reduce the rank of A to r. Since B has full rank, each row in A_{21} is a *unique linear combination* of the rows in B. Thus, we can change the entries in A_{22} according to these linear combinations so that each row in A is now a linear combination of the first r rows, *i.e.,* the rank of the modified matrix is at most r. [2]

Note that the above algorithm only modifies the entries of $A_{22} \in \mathbb{F}^{(n-r)\times(n-r)}$. Thus, at most $(n-r)^2$ many entries in A are changed, and $\mathcal{R}_A^{\mathbb{F}}(r) \le (n-r)^2$. □

We will now prove that almost all matrices have rigidity $(n-r)^2$.

Theorem 1.10 (Valiant's lower bounds [Val77])**.** *For any field $\mathbb{F}$,*

- *if $\mathbb{F}$ is infinite, then for all $0 \le r \le n$ there exists a matrix $M \in \mathbb{F}^{n\times n}$ of rigidity*

$$\mathcal{R}_M^{\mathbb{F}}(r) = (n-r)^2 \ ;$$

- *if $\mathbb{F}$ is finite, then for all $0 \le r \le n - \Omega(\sqrt{n})$ there exists a matrix $M \in \mathbb{F}^{n\times n}$ of rigidity*

$$\mathcal{R}_M^{\mathbb{F}}(r) = \Omega\left((n-r)^2/\log n\right) \ .$$

Proof of Theorem 1.10. Let $M_{r,s} = \{A \in \mathbb{F}^{n\times n} : \mathcal{R}_A^{\mathbb{F}}(r) \le s\}$ be the set of all matrices of r-rigidity at most s. We will show that the n^2 elements of matrices from $M_{r,s}$ lie in the union of images of a few rational maps from $\mathbb{F}^{n^2+s-(n-r)^2}$ to $\mathbb{F}^{n^2}$. Intuitively, since for $s \ll (n-r)^2$ these images cover only a negligible fraction of all matrices in $\mathbb{F}^{n\times n}$, we will have that "most" of the matrices are rigid.

For every matrix $M \in M_{r,s}$, there exists an s-sparse matrix $S \in \mathbb{F}^{n\times n}$ and a low-rank matrix $L \in \mathbb{F}^{n\times n}, \operatorname{rank}(L) = k \le r$ such that $M = S + L$. After one of at most $\binom{n}{k}^2$ permutations of rows and columns, we have the first k rows and columns of L linearly independent. The same permutations of rows and columns applied to M, give us a matrix of the form

$$\begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \ , \tag{1.11}$$

where $M_{11} \in \mathbb{F}^{k\times k}, M_{12} \in \mathbb{F}^{k\times(n-k)}, M_{21} \in \mathbb{F}^{(n-k)\times k}, M_{22} \in \mathbb{F}^{(n-k)\times(n-k)}$. Moreover, for at least one out of $\binom{n^2}{s}$ choices of s entries of the matrix, we have that a change in those entries makes $\operatorname{rank}(M_{11}) = \operatorname{rank}(M)$. Similarly to Theorem 1.8, this implies that all entries of M_{22} are then rational maps of the entries in

[2] Formally, we set $A_{22} = A_{21}B^{-1}A_{12}$.

M_{11}, M_{12}, M_{21}. That is, the n^2 entries of any matrix $M \in M_{r,s}$ lie in the union of at most $\binom{n}{r}^2 \cdot \binom{n^2}{s}$ rational maps from $\mathbb{F}^{s+n^2-(n-r)^2}$ to $\mathbb{F}^{n^2}$.

When $\mathbb{F}$ is infinite and $s < (n-r)^2$, every matrix in $M_{r,s}$ is in the union of finitely many images of rational functions from $\mathbb{F}^{n^2-1}$ to $\mathbb{F}^{n^2}$. Since n^2 rational functions of n^2-1 variables are algebraically dependent (see, e.g., [For92]), a finite union of such images is the set of roots of a non-zero polynomial. This implies that some matrices from $\mathbb{F}^{n\times n}$ do not belong to $M_{r,s}$.

When $|\mathbb{F}| = q < \infty$ is finite, each $M \in M_{r,s}$ is uniquely specified by one out of $\binom{n}{r}^2$ permutations, one of $\binom{n^2}{s}$ choices of s elements, values of those s elements, and values of the entries in M_{11}, M_{12}, M_{21}. Thus, the size of $M_{r,s}$ is bounded from above by

$$\binom{n}{r}^2 \cdot \binom{n^2}{s} \cdot q^s \cdot q^{n^2-(n-r)^2} \leq 2^{2n+2s\log n} \cdot q^{n^2+s-(n-r)^2},$$

which is at most $o(q^{n^2})$ for every $s < (n-r)^2/\Omega(\log_q n)$ and $r = n - \Omega(\sqrt{n})$. □

Note that the proof of the $\tilde{\Omega}\left((n-r)^2\right)$ lower bound in Theorem 1.10 does not provide a description of a rigid matrix, it merely proves its existence. This brings us to a discussion on explicitness of matrix constructions.

Lecture 3

1.5 On explicitness

In this section we will see two constructions of very rigid matrices. The main drawback of these constructions is that we do not know a polynomial time algorithm outputting the entries of these matrices.

First we show that a matrix consisting of algebraically independent elements has maximal rigidity. (One simple way to construct n^2 algebraically independent elements is given by Lindemann-Weierstrass Theorem.)

Lemma 1.12. *Let $M \in \mathbb{R}^{n\times n}$ be a matrix where all n^2 elements are algebraically independent over $\mathbb{Q}$. Then for every $0 \leq r \leq n$,*

$$\mathcal{R}^{\mathbb{R}}_M(r) = (n-r)^2 \, .$$

Proof. Let $s = (n-r)^2 - 1$. Assume, for the sake of contradiction, that $\mathcal{R}^{\mathbb{R}}_M(r) \leq s$. Then there exists an s-sparse matrix S, such that $\operatorname{rank}(M+S) \leq r$. Similarly to Theorem 1.10, the n^2 entries of M are rational functions of the s non-zero entries of S and at most $n^2-(n-r)^2$ entries of M. Therefore, polynomials of at most $s+n^2-(n-r)^2 < n^2$ elements generate all n^2 entries of M, which contradicts the assumption on algebraic independence of the elements of M. This implies that $\mathcal{R}^{\mathbb{R}}_M(r) \geq (n-r)^2$. On the other hand, Theorem 1.8 gives us that $\mathcal{R}^{\mathbb{R}}_M(r) \leq (n-r)^2$. □

While the construction of Lemma 1.12 has optimal rigidity, and each entry of such a matrix may have a very succinct mathematical description, there is no efficient algorithm outputting all digits of these entries. Thus, we will require that *explicit* constructions of matrices have polynomial-time algorithms outputting their entries.

Another non-explicit construction of rigid matrices is via an exponential-time algorithm. Suppose that we have a fixed finite field $\mathbb{F}$ of size $|\mathbb{F}| = q$. Then there is a trivial algorithm which runs in time $q^{O(n^2)}$ and outputs a rigid matrix. Let us fix an $0 \leq r \leq n-\Omega(\sqrt{n})$, and $s = \Omega\left((n-r)^2/\log n\right)$. In time $q^{O(n^2)}$, one can go over all pairs of matrices $M, S \in \mathbb{F}^{n\times n}$. For every such pair, the algorithm checks whether S is s-sparse and $\operatorname{rank}(M+S) \leq r$. When the algorithm finds an M for which there is no S with the above conditions, it outputs M as a rigid matrix and halts. (Theorem 1.10 guarantees existence of such a rigid matrix.)

When the field $\mathbb{F}$ is infinite, an algorithm cannot enumerate all matrices. But even in this case it is possible to construct a rigid matrix in time $2^{O(n^2)}$. In order to prove this, we will first show that there exists a rigid matrix with all entries from $\{0,1\}$. This will give us a way to enumerate all such matrices in time $2^{O(n^2)}$. Next, we will show that given such a matrix, one can check its rigidity in time $2^{O(n^2)}$, which will finish the proof.

Theorem 1.13 ([PR94]). *For all large enough n, there exists a matrix $M \in \{0,1\}^{n\times n}$ such that*

$$\mathcal{R}^{\mathbb{R}}_M\left(\frac{n}{200}\right) \geq \frac{n^2}{100}\,.$$

Proof. Let $r = \frac{n}{200}$, and assume that all matrices $M \in \{0,1\}^{n\times n}$ have rigidity $\mathcal{R}^{\mathbb{R}}_M(r) \leq s$. We will show the lower bound of $s \geq \frac{n^2}{100}$. Each such matrix M can be written as

$$M = S + L_1L_2\,, \tag{1.14}$$

where $S \in \mathbb{R}^{n\times n}$ is s-sparse, $L_1 \in \mathbb{R}^{n\times r}$ and $L_2 \in \mathbb{R}^{r\times n}$. There are $\binom{n^2}{\leq s}$ ways to choose the set of non-zero entries in S, let us fix one such set Γ. From Equation 1.14, each entry of M is a degree-2 polynomial of the entries of L_1, L_2 and Γ. In particular, there exist a set of n^2 degree-2 polynomials $\{f^{\Gamma}_{ij}\}_{i,j\in[n]}$ with variables being the entries of L_1, L_2, Γ such that $M_{ij} = f^{\Gamma}_{ij}(L_1, L_2, S)$.

For a set of t-variate polynomials $F = \{f_{ij}\}_{i,j\in[n]}$, we define its set of *zero-patterns* as the set of all sequences of zero-non-zero outputs of functions from F:

$$Z(F) = \{M \in \{0,1\}^{n\times n} :\ \exists x \in \mathbb{R}^t\ \forall i,j \in [n],\ M_{i,j} = \mathbf{1}_{f_{ij}(x)\neq 0}\}.$$

We will use the following lemma which asserts that for a set F of low-degree polynomials, $Z(F)$ is small.

Lemma 1.15 ([RBG01]). *If $F = \{f_{ij}\}_{i,j\in[n]}$ is a collection of t-variate polynomials of degree at most d, then*

$$|Z(F)| \leq \binom{t+dn^2}{t}\,.$$

Proof of Lemma 1.15. Let $m = n^2$ be the number of polynomials, and let $N = |Z(F)|$. Let $x_1, x_2, \ldots, x_N \in \mathbb{R}^t$ be a set of points witnessing the N distinct zero-patterns of F. For an $i \in [N]$, let $S_i \subseteq [m]$ be the set of (indices of) polynomials from F which *are not* zeros at the point x_i. For every $i \in [N]$, define the following polynomial

$$g_i = \prod_{k\in S_i} f_k\,.$$

Note that $g_i(x_j) = 0$ if and only if there exists $f_k \in S_i \setminus S_j$. Therefore, we have $g_i(x_j) = 0$ if and only if $S_i \not\subseteq S_j$.

Now we prove that all $\{g_i\}_{i\in[N]}$ are linearly independent. Suppose, to the contrary, that there exist $a_1, a_2, \ldots, a_N \in \mathbb{R}$ such that $\sum_{i\in[N]} a_ig_i = 0$, and at least one $a_i \neq 0$. Let

$$i^* = \operatorname*{arg\,min}_{i\in[N],\ a_i\neq 0} |S_i|\,.$$

We have that $a_{i^*}g_{i^*}(x_{i^*}) \neq 0$ and $\sum_{i\in[N]} a_ig_i(x_{i^*}) = 0$. Due to the minimality of S_{i^*}, for every $a_i \neq 0$, $S_i \not\subseteq S_{i^*}$. This implies that $a_ig_i(x_{i^*}) = 0$ for all $i \neq i^*$, and, thus, $\sum_{i\in[N]} a_ig_i(x_{i^*}) \neq 0$, which leads to a contradiction.

Finally, since the degree of each g_i is at most dm, and all g_i are linearly independent, N is bounded from above by the dimension of the space spanned by t-variate polynomials of degree at most dm. Thus, $N \leq \binom{t+dm}{t}$. □

Recall that from Equation 1.14, all matrices $M \in \{0,1\}^{n\times n}$ can be described by $\{f^{\Gamma}_{ij}\}_{i,j\in[n]}$ for some $\Gamma \in \binom{[n]\times[n]}{s}$, where each polynomial has degree at most 2 and depends on $2rn + s$ variables. Now, from Lemma 1.15 with $t = 2rn + s$ and $d = 2$, we have that

$$\binom{2rn+s+2n^2}{2rn+s} \cdot \binom{n^2}{\leq s} \geq 2^{n^2}\,. \tag{1.16}$$

Assume, for the sake of contradiction, that for some $r \leq \frac{n}{200}$ we have $s < \frac{n^2}{100}$. We have that $2rn+s \leq \frac{n^2}{50}$. Now, the left-hand side of Equation 1.16 can be bounded from above as follows:

$$\binom{2rn+s+2n^2}{2rn+s} \cdot \binom{n^2}{\leq s} \leq \binom{\frac{101n^2}{50}}{\frac{n^2}{50}} \cdot \binom{n^2}{\leq \frac{n^2}{100}} \leq (101e)^{\frac{n^2}{50}} \cdot (100e)^{\frac{n^2}{100}} \leq 2^{\frac{n^2}{2}} ,$$

which contradicts Equation 1.16. Thus, we conclude that for any $r \leq \frac{n}{200}$, there exists a matrix $M \in \{0,1\}^{n\times n}$ such that $\mathcal{R}^{\mathbb{R}}_M(r) \geq \frac{n^2}{100}$. □

Now we show that one can check whether a given matrix $M \in \{0,1\}^{n\times n}$ is rigid in time $2^{O(n^2)}$. [3]

Theorem 1.17. *Let $M \in \{0,1\}^{n\times n}$, and r and s be non-negative integers. Then one can decide whether $\mathcal{R}^{\mathbb{R}}_M(r) > s$ in time $2^{O(n^2)}$.*

Proof. Note that $\mathcal{R}^{\mathbb{R}}_M(r) \leq s$ if and only if $M = S + L_1L_2$ for s-sparse S and $L_1 \in \mathbb{R}^{n\times r}, L_2 \in \mathbb{R}^{r\times n}$. For any choice of non-zero entries of S, we have that the entries of M are degree-2 polynomials of $t = 2nr + s$ variables with $\{0,1\}$-coefficients. It is known that deciding whether such a system of polynomial equations has a real solution can be solved in time $2^{O(n^2)}$ (see, e.g., Proposition 13.19 in [BPR07]). Since there are $\binom{n^2}{s} \leq 2^{n^2}$ choices of s non-zero entries, we have that the total running time of the algorithm is $2^{O(n^2)}$. □

This way we have a set of 2^{n^2} matrices such that at least one of them is rigid, and rigidity of each matrix can be checked in time $2^{O(n^2)}$. This gives us a $2^{O(n^2)}$-time algorithm for constructing a rigid matrix over the reals.

Although the above algorithms construct matrices of high rigidity, their running time is $2^{\Omega(n^2)}$. We define *explicit* constructions of matrices as matrices that have algorithms outputting all their entries in *polynomial time*.

1.6 Summary

In Theorem 1.8 we showed that for every field $\mathbb{F}$, matrix $M \in \mathbb{F}^{n\times n}$, and integer $0 \leq r \leq n$, $\mathcal{R}^{\mathbb{F}}_M(r) \leq (n-r)^2$. Below we summarize the non-explicit *lower bounds on rigidity* presented in this chapter.

rigidity	field	running time	reference
$\frac{(n-r)^2}{\log n}$	any finite field	existence	Theorem 1.10
$(n-r)^2$	any infinite field	existence	Theorem 1.10
$(n-r)^2$	$\mathbb{R}$	algebraically independent entries	Lemma 1.12
$\frac{(n-r)^2}{\log n}$	any fixed finite field	$2^{O(n^2)}$	section 1.5
$(n-r)^2$	$\mathbb{R}$	$2^{O(n^2)}$	Theorem 1.13, Theorem 1.17

Table 1.1: Summary of non-explicit lower bounds.

[3] For more efficient algorithms for the case of low rigidity parameters see [FLM+18]. A **PSPACE**-algorithm for this problem follows immediately from the fact that existential theory of the reals lies in **PSPACE** [Can88].

1.7 Notes

Rigidity was introduced as a means to study circuit complexity by Valiant [Val77] and Grigoriev [Gri76]. An excellent presentation of the known lower bounds on rigidity over large fields can be found in the book of Lokam [Lok09]. The books of Jukna [Juk12] and Jukna and Sergeev [JS13] include many applications of matrix rigidity to circuit complexity. Earlier surveys on rigidity are due to Codenotti [Cod00] and Cheraghchi [Che05]. The tight upper bound of Theorem 1.8, and the non-constructive lower bounds of Theorem 1.10 and Theorem 1.13 were proven by Valiant [Val77], and Pudlák and Rödl [PR94]. The proof of Theorem 1.2 was first given by Midrijānis [Mid05].

Chapter 2

Explicit Constructions

In this chapter we give three proofs [Fri93, PR94, SSS97] of the best known explicit lower bound of $\mathcal{R}(r) \geq \Omega\left(\frac{n^2}{r} \cdot \log \frac{n}{r}\right)$ on matrix rigidity. All the three proofs work for (almost) any generator matrix of a good linear code. Since there are explicit linear codes over all finite fields, the presented proofs work for all finite fields. We will see that the last construction (due to Shokrollahi, Spielman and Stemann) easily generalizes to infinite fields. **Lecture 4**

A *linear* code over a field $\mathbb{F}$ is a linear subspace $C \subseteq \mathbb{F}^n$ of dimension k. The *distance* of the code is the minimum Hamming distance between two vectors in C or, equivalently, the minimum Hamming weight of a non-zero vector in C.

Definition 2.1 (Linear code). *Let $\mathbb{F}$ be a field, and n, k, d be positive integers such that $d, k < n$. A subspace $C \subseteq \mathbb{F}^n$ is a linear code of dimension k with minimum distance d if*

1. $\dim(C) = k$;
2. *for every $x \in C \setminus \{0\}$, $\|x\|_0 \geq d$.*

A linear code $C \subseteq \mathbb{F}^n$ of dimension k can be specified by a *generator matrix* $G \in \mathbb{F}^{n \times k}$ such that C is the column space of G. Since we focus on asymptotic behavior of codes, by a code we will mean an infinite sequence $C = \{C_i : i \in \mathbb{N}\}$ where $C_i \subseteq \mathbb{F}^n$. For our purposes, it will suffice to say that a code is *explicit* if there is a polynomial time algorithm that for every n, outputs a generator matrix of C_n in time $\mathsf{poly}(n)$. We will say that a code C is *good* if for the codes of this sequence we have that $k = \Theta(n)$ and $d = \Theta(n)$.

There are exist explicit good linear error correcting codes over all finite fields (see, e.g., Justesen and Goppa codes in [MS77, LG88, vL12]).

Proposition 2.2. *For any finite field $\mathbb{F}$, there exists an explicit family of linear error correcting codes over $\mathbb{F}$ of dimension $k = n/4$ and minimum distance $d = \delta n$ for a constant $\delta > 0$.*

The main result of this chapter is the following.

Theorem 2.3. *Let F be a fixed finite field, and $G \in \mathbb{F}^{n \times k}$ be a generator matrix of a linear code of dimension $k = \Theta(n)$ and distance $d = \Theta(n)$, then for every $\Omega(\log n) < r < O(n)$,*

$$\mathcal{R}_G^{\mathbb{F}}(r) \geq \Omega\left(\frac{n^2}{r} \cdot \log \frac{n}{r}\right) .$$

2.1 The lower bound of Friedman

Let us fix a generator matrix $G \in \mathbb{F}^{n \times k}$ of a good linear code with distance and dimension $d, k = \Theta(n)$. We will prove the lower bound of Friedman in two steps. First, in Theorem 2.4 we will show that G has high "column rigidity". That is, in order to drop the rank of G to r, one has to modify at least $\Omega(\frac{n}{r} \log_q \frac{k}{r})$ entries in *some* column of G. Second, in Theorem 2.5 we will use a simple averaging argument to reduce column rigidity to rigidity.

Theorem 2.4 ([Fri93]). *Let $\mathbb{F} = \mathbb{F}_q$ be a finite field of size q. Let $G \in \mathbb{F}^{n\times k}$ be a generator matrix of a code of dimension k and distance δn for a constant $0 < \delta < 1$. For any $\log_q k \le r \le \frac{k}{4}$, if every column of $B \in \mathbb{F}^{n\times k}$ contains at most $\frac{\delta n}{4r}\log_q \frac{k}{r}$ non-zero entries, then*

$$\operatorname{rank}(G+B) > r\ .$$

Proof of Theorem 2.4. Assume, for the sake of contradiction, that there exists $B \in \mathbb{F}_q^{n\times k}$ such that $\operatorname{rank}(G+B) \le r$ and each column of B has at most $\frac{\delta n}{4r}\log_q \frac{k}{r}$ nonzero entries.

The proof employs two ideas. First, using a packing argument, we will show that the *kernel* of $G+B$ must contain a sparse vector $x \in \mathbb{F}^k$. Second, as Gx is a codeword of C and $Gx + Bx = \mathbf{0}$, Bx is also a codeword of C and, thus, $\|Bx\|_0$ must be large due to the minimum distance property of C. This leads to a contradiction as x and the columns of B are sparse.

Let us draw a Hamming ball of radius $d/2$ around each point in the kernel of $G+B$. Since we assume that $\operatorname{rank}(\ker(G+B)) = k - \operatorname{rank}(G+B) \ge k-r$, if

$$q^{k-r}\cdot\left|\text{Hamming ball of radius } d/2 \text{ in } \mathbb{F}_q^k\right| > q^k,$$

then there must be two distinct points in the null space of $G+B$ such that their Hamming balls intersect. This gives us a non-zero vector x in the kernel of $G+B$ of sparsity at most d. The following shows that it suffice to pick d as an even number between $\frac{2r}{\log_q \frac{k}{r}} \le d \le \frac{2r}{\log_q \frac{k}{r}} + 2$.

$$\begin{aligned}\left|\text{Hamming ball of radius } d/2 \text{ in } \mathbb{F}_q^k\right| &\ge \binom{k}{d/2}\cdot(q-1)^{d/2}\\ &\ge q^{\frac{d}{2}\cdot\log_q\left[\frac{2k}{d}(q-1)\right]} > q^r\ .\end{aligned}$$

Next, since Gx is a *non-zero* codeword of C and $Gx + Bx = \mathbf{0}$, we know that Bx is a non-zero codeword of C and, thus, $\|Bx\|_0 \ge \delta n$. On the other hand, x has only d non-zero coordinates, and each column of B has at most $\frac{\delta n}{4r}\log_q\frac{k}{r}$ non-zero entries. We have that

$$\|Bx\|_0 \le d\cdot\frac{\delta n}{4r}\log_q\frac{k}{r} \le \left(\frac{2r}{\log_q\frac{k}{r}}+2\right)\frac{\delta n}{4r}\log_q\frac{k}{r} < \delta n\ ,$$

which contradicts the distance property of C. □

Theorem 2.5. *Let $\mathbb{F} = \mathbb{F}_q$ be a finite field of size q. Let $G \in \mathbb{F}^{n\times k}$ be a generator matrix of a code of dimension k and distance δn for a constant $0 < \delta < 1$. Then for any $\frac{\log_q k}{2} \le r \le \frac{k}{8}$,*

$$\mathcal{R}_G^{\mathbb{F}}(r) \ge \frac{\delta k n \log_q \frac{k}{2r}}{8r}\ .$$

Proof. Assume, for the sake of contradiction, that there exists $S \in \mathbb{F}_q^{n\times k}$ such that $\operatorname{rank}(G+S) \le r$ and $\|S\|_0 \le \frac{\delta k n\log_q\frac{k}{2r}}{8r}$. Let $J \subset [k]$ be the indices of the $\frac{k}{2}$ sparsest columns of S, and let S_J be the sub-matrix of S restricted to the columns in J. By Markov's inequality, each column of S_J has at most

$$\left(\frac{\delta k n\log_q\frac{k}{2r}}{8r}\right)\Big/\left(\frac{k}{2}\right) = \frac{\delta n}{4r}\log_q\frac{k}{2r}$$

many non-zero entries. Now Theorem 2.4 applied to G_J and S_J implies that a column in S_J must contain more than $\frac{\delta n}{4r}\log_q\frac{k}{2r}$ non-zero entries, which leads to a contradiction. □

Now, using good codes from Proposition 2.2, we get a lower bound of $\mathcal{R}_G^{\mathbb{F}}(r) \ge \Omega\left(\frac{n^2}{r}\cdot\log\frac{n}{r}\right)$ for any $\Omega(\log_q n) \le r \le \frac{n}{32}$.

2.2 The lower bound of Pudlák and Rödl

Lecture 5

Theorem 2.6 ([PR94]). *Let $\mathbb{F} = \mathbb{F}_q$ be a finite field of size q. Let $G \in \mathbb{F}^{n\times k}$ be a generator matrix of a code of dimension k and distance δn for a constant $0 < \delta < 1$. Then for any $1 \le r \le \frac{k}{q^2}$,*

$$\mathcal{R}_G^{\mathbb{F}}(r) \ge \frac{\delta kn \log_q \frac{k}{r}}{8r} .$$

Proof of Theorem 2.6. Assume that $G = L + S$, where $\operatorname{rank}(L) \le r$ and $\|S\|_0 \le \frac{\delta nk}{4\ell}$ for an even integer ℓ to be chosen later. Then, by Markov's inequality, there exist $\frac{k}{2}$ columns of S each having at most $\frac{\delta n}{2\ell}$ non-zero entries. Let $G', L', S' \in \mathbb{F}^{n\times k/2}$ be the matrices G, L and S restricted to these columns. In particular, we have that $\operatorname{rank}(L') \le r$.

The proof is based on the following beautiful idea. By the distance property of the code, each non-zero linear combination of the columns of G' has weight at least δn. Now, since the columns of $L' = G' - S'$ differ from the columns of G' only in a few positions, any *short* linear combination of the columns of L' is still non-zero. This guarantees that L' must generate a quite large space, which contradicts the assumption about the low rank of L'.

Observe that for any $x \in \mathbb{F}_q^{k/2} \setminus \{0^{k/2}\}$ such that $\|x\|_0 \le \ell$, we have

$$\|L'x\|_0 = \|(G' - S')x\|_0 \ge \|G'x\|_0 - \|S'x\|_0 \ge \delta n - \frac{\delta n}{2\ell}\cdot \ell = \frac{\delta n}{2} > 0 ,$$

where $\|G'x\|_0 \ge \delta n$ follows from the fact that $G'x$ is a non-zero codeword, and the other term follows from the column-sparsity of S'. This observation implies that for any distinct $y_1, y_2 \in \mathbb{F}_q^{k/2}$ with $\|y_1\|_0, \|y_2\|_0 \le \frac{\ell}{2}$, $L'y_1 \ne L'y_2$.

Note that the above gives us a lower bound on the number of vectors in the column span of L'. Thus,

$$\operatorname{rank}(L') \ge \log_q \binom{k/2}{\ell/2}.$$

Picking $\frac{2r}{\log_q \frac{k}{r}} \le \ell \le \frac{2r}{\log_q \frac{k}{r}} + 2$ as an even integer, we have

$$\operatorname{rank}(L') \ge \log_q \binom{k/2}{\ell/2} > \frac{\ell}{2}\log_q \frac{k}{\ell} \ge r,$$

which leads to a contradiction. Thus, we conclude that $\mathcal{R}_G^{\mathbb{F}_q}(r) \ge \frac{\delta nk}{4\ell} \ge \frac{\delta nk}{8r}\log\frac{k}{r}$. □

2.3 The lower bound of Shokrollahi, Spielman and Stemann

In this section we show that a few changes in a matrix always leave some large submatrix unchanged. Namely, if one makes only $O(\frac{n^2}{r}\cdot\log\frac{n}{r})$ changes in an $n\times n$ matrix, then there must be an untouched $r\times r$ submatrix in it. In particular, if we start with a matrix whose every $r\times r$ submatrix has high (or even full) rank, then the matrix remains high-rank even after $O(\frac{n^2}{r}\cdot\log\frac{n}{r})$ changes, which implies rigidity.

First, in Lemma 2.8 we will prove that after $O(\frac{n^2}{r}\cdot\log\frac{n}{r})$ changes in $n\times n$ matrix there always remains an untouched submatrix. Then, in Theorem 2.9, we will apply this lemma to prove simple lower bounds on rigidity of explicit matrices, and in Theorem 2.11 to prove lower bounds on the rigidity of (normalized) generator matrices of error-correcting codes.

We will need the classical Kővári-Sós-Turán theorem from extremal graph theory.

Theorem 2.7 (Zarankiewicz problem [KST54, Bol04]). *Let $n, s \in \mathbb{N}$ such that $s \le n$ and G be an $n\times n$ bipartite graph. If G has no $s\times s$ bi-clique, then the number of edges in G is at most*

$$(s-1)^{1/s}(n-s+1)n^{1-1/s} + (s-1)n.$$

Proof of Theorem 2.7. Let G be an $n\times n$ $K_{s,s}$-free bipartite graph. Let $\mathrm{Star}_s=\{(u,T)\subseteq[n]\times\binom{[n]}{s}:(u,v)\in E(G),\ \forall v\in T\}$ be the set of left s-stars in G. There are two ways to count Star_s: (i) let $d_1,d_2,\ldots,d_n$ be the degrees of left vertices in G, then $|\mathrm{Star}_s|=\sum_{i\in[n]}\binom{d_i}{s}$ and (ii) for each $T\in\binom{[n]}{s}$, T forms at most $s-1$ many left s-star due to the $K_{s,s}$-free property. Namely,

$$\sum_{i\in[n]}\binom{d_i}{s}\le|\mathrm{Star}_s|\le(s-1)\binom{n}{s}.$$

By convexity, this implies

$$\sum_{i\in[n]}(d_i-s+1)^s\le(s-1)(n-s+1)^s.$$

By Hölder's inequality,

$$\sum_{i\in[n]}(d_i-s+1)\le\left(\sum_{i\in[n]}(d_i-s+1)^s\right)^{1/s}\cdot n^{1-1/s}\le(s-1)^{1/s}(n-s+1)n^{1-1/s}.$$

Finally, as $|E(G)|=\sum_{i\in[n]}d_i$, we have that

$$|E(G)|=\sum_{i\in[n]}d_i=\sum_{i\in[n]}(d_i-s+1)+(s-1)n\le(s-1)^{1/s}(n-s+1)n^{1-1/s}+(s-1)n.$$

□

Lemma 2.8. *Let $n,r\in\mathbb{N}$ such that $\log n\le r\le n$, and A be an $n\times n$ matrix. If fewer than $\frac{n(n-r)}{2(r+1)}\log\frac{n}{r}$ entries of A are changed, then some $(r+1)\times(r+1)$ submatrix of A remains untouched.*

Proof of Lemma 2.8. Consider an $n\times n$ bipartite graph G, whose vertices in one part correspond to the rows of A, and vertices in the other part correspond to the columns of A. We connect the vertex i from the first part to the vertex j from the second part if and only if the entry A_{ij} remains unchanged. Thus, an $(r+1)\times(r+1)$ unchanged submatrix corresponds to an $(r+1)\times(r+1)$ bi-clique in the bipartite graph G.

We assume, towards a contradiction, that there is no $(r+1)\times(r+1)$ untouched submatrix in A, *i.e.,* no $(r+1)\times(r+1)$ bi-clique in G. We apply Theorem 2.7 with $s=r+1$, and conclude that a graph without an $(r+1)\times(r+1)$ bi-clique has at most

$$\begin{aligned}r^{1/(r+1)}(n-r)n^{1-1/(r+1)}+rn&=n^2-n(n-r)\cdot\left[1-\left(\frac{r}{n}\right)^{1/(r+1)}\right]\\&<n^2-n(n-r)\frac{\log\frac{n}{r}}{2(r+1)}\end{aligned}$$

edges, where the last inequality uses the approximation $e^{-x}<1-\frac{x}{2}$ for $x\in[0,1]$ and holds whenever $r\ge\log n$.

Finally, we conclude that if fewer than $\frac{n(n-r)}{2(r+1)}\log\frac{n}{r}$ entries in A are changed, *i.e.,* G has more than $n^2-\frac{n(n-r)}{2(r+1)}\log\frac{n}{r}$ edges, then some $(r+1)\times(r+1)$ submatrix of A remains untouched. □

An immediate corollary of Lemma 2.8 is that if every $(r+1)\times(r+1)$ submatrix of A is full-rank, then $\mathcal{R}_A(r)\ge\frac{n^2}{4(r+1)}\log\frac{n}{r}$ for $\log n\le r\le\frac{n}{2}$. The following theorem applies this idea to *Cauchy matrices* over small (but non-constant size) fields.

Theorem 2.9 ([SSS97], non-fixed field)**.** *Let $\mathbb{F}$ be a field containing at least $2n$ distinct elements denoted by $x_1,x_2,\ldots,x_n$ and $y_1,y_2,\ldots,y_n$. Let $A\in\mathbb{F}^{n\times n}$ be a Cauchy matrix: $A_{ij}=\frac{1}{(x_i-y_j)}$. Then*

$$\mathcal{R}^{\mathbb{F}}_A(r)\ge\frac{n^2}{4(r+1)}\log\frac{n}{r}$$

for $\log n\le r\le\frac{n}{2}$.

Proof of Theorem 2.9. By the above discussion, it suffices to show that every $(r+1)\times(r+1)$ submatrix of the Cauchy matrix A has full rank for $\log n \le r \le \frac{n}{2}$. As every such submatrix is also a Cauchy matrix, it suffices to show that every Cauchy matrix has full rank.

In Problem 3 of Homework 1, we will show that the determinant of the Cauchy matrix A is

$$\det(A) = \frac{\prod_{1\le i<j\le n}(x_j - x_i)(y_i - y_j)}{\prod_{1\le i,j\le n}(x_i - y_j)}\,. \tag{2.10}$$

Since all x_i and y_j are distinct, $\det(A) \ne 0$, which finishes the proof. □

Finally, we use the above idea to construct (moderately) rigid matrices over constant-size fields.

Theorem 2.11 ([SSS97], fixed field)**.** *Let $\mathbb{F}$ be a field, $n \in \mathbb{N}$, $\varepsilon \in (0,1)$, and $C \subseteq \mathbb{F}^{2n}$ be an explicit linear code of dimension n with minimum distance $(1-\varepsilon)n$. Then, there exists a matrix $A \in \mathbb{F}^{n\times n}$ that can be efficiently constructed from any generator matrix of C such that*

$$\mathcal{R}^{\mathbb{F}}_A(r) \ge \frac{n^2}{8(r+1)} \log \frac{n}{(2r+1)}$$

for any $\varepsilon n \le r \le \frac{n-2}{2}$.

Proof of Theorem 2.11. Let $G \in \mathbb{F}^{2n\times n}$ be a generator matrix of C. We run Gaussian elimination in polynomial time to write G in the *standard form*:

$$G' = \begin{pmatrix} I_n \\ A \end{pmatrix}\,,$$

where I_n is the $n\times n$ identity matrix and $A \in \mathbb{F}^{n\times n}$.

Claim 2.12. *Let $s = r+1$. Then, every $2s \times 2s$ submatrix of A has rank at least s.*

Proof of Claim 2.12. Assume that there is a $2s\times 2s$ submatrix of A of rank less than s. Without loss of generality, let it be the submatrix $A' \in \mathbb{F}^{2s\times 2s}$ in the top left corner of A. As $\mathrm{rank}(A') < s$, there exists a linear combination of s columns of A' which equals 0. Since any linear combination of the columns of G' is a codeword of C, we have a codeword $x \in \mathbb{F}^{2n}$ whose first n coordinates have s non-zeros, and the last n coordinates have at least $2s$ zeros. This gives us a codeword $x \in C$ of sparsity

$$\|x\|_0 \le s + (n-2s) = n - s = n - r - 1 < (1-\varepsilon)n\,,$$

which contradicts the assumption on the distance of the code C. □

Finally, Lemma 2.8 implies that in order to drop the rank of A to r, one needs to make at least

$$\frac{n(n-2r-2)}{4(r+1)} \log \frac{n}{2r+1} \ge \frac{n^2}{8(r+1)} \log \frac{n}{2r+1}$$

changes for any $\varepsilon n \le r \le \frac{n-2}{4}$. □

There are explicit constructions of algebraic-geometric codes [VT91, MS77] of dimension n in $\mathbb{F}_q^{2n}$ with minimum distance $(1-\varepsilon)n$ for $\varepsilon = \frac{2}{\sqrt{q}-1}$ for every prime square q. In particular, for every prime square $q > 25$, Theorem 2.11 applied to the algebraic-geometric codes gives rigidity lower bounds over $\mathbb{F}_q$ for some range of $r = O(n)$.

2.4 Rigidity of the Walsh-Hadamard Matrix

Lecture 6

Definition 2.13. *For any $N = 2^n$, the Walsh-Hadamard matrix $H_N \in \mathbb{C}^{N\times N}$ is defined as*

$$H_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},$$
$$H_N = H_2^{\otimes n},$$

where $\otimes$ denotes the Kronecker product.

First, in Problem 4 of Homework 1 we prove that H_N is not rigid for large rank $r \geq N/2$. Then, in Theorem 2.18 we will prove that for every rank $r \leq N/2$, the rigidity of H_N is at least $\mathcal{R}_H^{\mathbb{C}}(r) \geq \frac{N^2}{4r}$.

Lemma 2.14 (Homework 1, Problem 4)**.** *Let $H \in \mathbb{C}^{N\times N}$ be the Walsh-Hadamard matrix. Then*

$$\mathcal{R}_H^{\mathbb{C}}(N/2) \leq N .$$

In Problem 5 of Homework 1 we will prove the following facts about matrix norms.

Proposition 2.15. *Let $M \in \mathbb{C}^{m\times n}$ be a matrix, $k = \min(m, n)$, and $r = \operatorname{rank}(M)$. Let*

$$\sigma_1(M) \geq \ldots \geq \sigma_r(M) > \sigma_{r+1}(M) = \ldots = \sigma_k(M) = 0$$

be the singular values of M. Then

- *The Frobenius norm $\|M\|_F = \left(\sum_{i=1}^m \sum_{j=1}^n |M_{i,j}|^2\right)^{1/2} = \left(\sum_{i=1}^k \sigma_i^2(M)\right)^{1/2}$.*
- *The spectral norm $\|M\|_2 = \sigma_1(M)$.*
- *If M' is a submatrix of M, then $\sigma_i(M') \leq \sigma_i(M)$. In particular, $\|M'\|_2 \leq \|M\|_2$.*

We will also need the following lower bound on the rank of submatrices of the Walsh-Hadamard matrix due to Lokam [Lok95].

Lemma 2.16 ([Lok95])**.** *For any submatrix $H' \in \mathbb{C}^{a\times b}$ of the Walsh-Hadamard matrix $H \in \mathbb{C}^{N\times N}$,*

$$\operatorname{rank}(H') \geq ab/N .$$

Proof. By Proposition 2.15,

$$\|H'\|_F^2 = \sum_{i=1}^k \sigma_i^2(H') \leq \operatorname{rank}(H') \cdot \sigma_1^2(H') = \operatorname{rank}(H') \cdot \|H'\|_2^2 \leq \operatorname{rank}(H') \cdot \|H\|_2^2 . \tag{2.17}$$

Since the absolute values of all entries in H' are 1, $\|H'\|_F^2 = ab$. From $H \cdot H^T = N \cdot I_N$, we have that $\sigma_i(H) = \sqrt{N}$ for $1 \leq i \leq N$, and, thus, $\|H\|_2 = \sqrt{N}$.

Now, from Equation 2.17,

$$\operatorname{rank}(H') \geq \|H'\|_F^2 / \|H\|_2^2 = ab/N .$$

□

Now we are ready to present the best known lower bound on rigidity of the Walsh-Hadamard matrix due to de Wolf [De 06]. Later in the course, we will also prove an upper bound on rigidity of the Walsh-Hadamard matrix.

Theorem 2.18 ([De 06])**.** *Let $H \in \mathbb{C}^{N\times N}$ be the Walsh-Hadamard matrix. For every $r \leq N/2$,*

$$\mathcal{R}_H^{\mathbb{C}}(r) \geq \frac{N^2}{4r} .$$

Proof. Let $s = \mathcal{R}^{\mathbb{C}}_H(r)$, and let $S \in \mathbb{C}^{N\times N}$ be such that

$$\mathrm{rank}(H+S) \le r \text{ and } \|S\|_0 \le s\,.$$

Then by an averaging argument, there exists a set of $2r$ rows of S with at most $2rs/n$ non-zero entries. If $N \le 2rs/N$, then $s \ge \frac{N^2}{2r}$ concludes the proof. If $N > 2rs/N$, then we consider a submatrix $H' \in \mathbb{C}^{(2r)\times(N-2rs/N)}$ of H where all entries of S are zeros. By Lemma 2.16,

$$r \ge \mathrm{rank}(H') \ge 2r(N-2rs/N)/N\,,$$

which implies $s \ge \frac{N^2}{4r}$. □

In the table below we summarize the known rigidity lower bounds for explicit matrices.

rigidity	reference
$\frac{n^2}{r^4\log^2 r}$	[PS88]
$\frac{n^2}{r^3\log r}$	[Raz88]
$\frac{n^2}{r^2}$	[Alo94]
$\frac{n^2}{r^2}$	[Lok95]
$\frac{n^2}{256r}$	[KR98]
$\frac{n^2}{4r}$	[De 06]

Table 2.1: Lower bounds on the rigidity of the Walsh-Hadamard matrix.

2.5 Overview of Chapters 1 and 2

See the slides for Lectures 6 and 7 for an overview of the following tools (and their applications).

- Probabilistic Method
- Algebraic Independence **Lecture 7**
- Polynomial Method
- Zarankiewicz Problem
- Hölder's inequality
- Zero-patterns **Lecture 8**
- Hadamard Matrix
- Spectral Methods
- Error-Correcting Codes

Chapter 3

Semi-explicit constructions

Lecture 9

3.1 Constructions with independent entries

Shoup and Smolensky [SS91] used an algebraic dimension argument for proving super-linear lower bounds against linear circuits of sub-polynomial depth. Lokam [Lok00] extended their argument, and proved a lower bound on rigidity of Vandermonde matrices containing the powers of n algebraically independent entries. Later Lokam [Lok06] generalized these ideas for proving quadratic lower bounds on rigidity for linear rank of matrices with infinite precision in the entries. Kumar, Lokam, Patankar and Sarma [KLPS14] used algebraic geometry arguments to construct a matrix with optimal rigidity parameters for every r (at the expense of having infinite precision in the entries, too).

This week we will see some of these constructions. Namely, we will prove rigidity lower bounds for a Vandermonde matrix with algebraically independent entries [Lok00], and for a matrix consisting of square roots of distinct primes [Lok06]. See [LTV03, Lok09, KLPS14, GHIL16] for excellent overviews of the algebraic geometry approach to rigidity.

Recall from Lemma 1.12 that any matrix with n^2 algebraically independent entries is rigid. We would like to reduce the number of algebraically independent entries needed for constructions of rigid matrices, as well as to weaken the assumption on their independence. In the next class we will show that n^2 linearly independent (but not algebraically independent) numbers are also sufficient for high rigidity. Today we will show that even n algebraically independent entries are sufficient for (moderate) rigidity.

3.1.1 Vandermonde matrix with algebraically independence entries

Today we will show that Vandermonde matrices V with powers of algebraically independent entries have rigidity $\mathcal{R}_V^{\mathbb{F}}(r) \geq \Omega(n^2)$ for $r \leq O(\sqrt{n})$.

Definition 3.1 (Vandermonde matrix). *A Vandermonde matrix $V \in \mathbb{F}^{n\times n}$ is a matrix of the form $V_{ij} = x_i^j$ for some $x_1, x_2, \ldots, x_n \in \mathbb{F}$.*

We will prove that Vandermonde matrices V are rigid as follows. First, in Definition 3.3 we will introduce a measure of algebraic independence (over $\mathbb{Q}$) of a set of numbers, called the Shoup-Smolensky dimension. Second, in Lemma 3.4 we will prove that the Shoup-Smolensky dimension of low-rank matrices is low. Finally, in Lemma 3.5 we will show that the Shoup-Smolensky dimension of $V - S$ for any sparse matrix S is large. From this, in Theorem 3.2, we will conclude that V is rigid.

Theorem 3.2 (Vandermonde matrix with algebraically independent entries is rigid). *Let $x_1, \ldots, x_n \in \mathbb{C}$ be algebraically independent over $\mathbb{Q}$, and let $V \in \mathbb{C}^{n\times n}$ be the Vandermonde matrix $V_{i,j} = x_i^j$. For any $1 \leq r \leq \frac{\sqrt{n}}{10}$,*

$$\mathcal{R}_V^{\mathbb{C}}(r) \geq n(n - 100 \cdot r^2)/2\,.$$

For example, when $r \le \varepsilon\sqrt{n}$ for a small enough $\varepsilon > 0$, we have $\mathcal{R}^{\mathbb{C}}_V(r) \ge \Omega(n^2)$. However, it is an interesting open problem to prove non-trivial rigidity lower bound for such Vandermonde matrices in the regime of $r = \omega(\sqrt{n})$.

Definition 3.3 (Shoup-Smolensky dimension [SS91])**.** *For any $t, n \in \mathbb{N}$ and $A \in \mathbb{C}^{n\times n}$, the t-Shoup-Smolensky dimension of A, denoted $\dim^{SS}_t(A)$, is the dimension of the vector space over $\mathbb{Q}$ spanned by products of t distinct elements of A.*

To prove Theorem 3.2, we will to show that rank-r matrices have Shoup-Smolensky dimension $\le \binom{nr+t}{t}^2$, while V even after s changes has Shoup-Smolensky dimension $\ge (n - \frac{s}{t})^t$.

Lemma 3.4. *For any $t, n \in \mathbb{N}$, and $A \in \mathbb{C}^{n\times n}$ of rank $r = \operatorname{rank}(A)$,*

$$\dim^{SS}_t(A) \le \binom{nr+t}{t}^2 .$$

Proof of Lemma 3.4. Since A has rank r, there exists $B, C^\top \in \mathbb{C}^{n\times r}$ such that $A = BC$. Then each entry of A can be written as a degree-2 polynomial where each monomial is of the form b_1c_1 for some b_1 from B and c_1 from C.

Thus, any t-wise product of entries from A is a degree-$2t$ polynomial where each monomial is of the form $\prod_{i\in[t]} b_ic_i$ for some b_i from B and c_i from C for all $i \in [t]$. This implies that the vector space spanned by the products of t distinct elements of A is also spanned by these monomials. As the number of n-variate monomials of degree $\le d$ is $\binom{n+d}{d}$, we conclude that $\dim^{SS}_t(A) \le \binom{nr+t}{t}^2$. $\square$

Lemma 3.5. *Let $x_1, \dots, x_n \in \mathbb{C}$ be algebraically independent over $\mathbb{Q}$, and let $V \in \mathbb{C}^{n\times n}$ be the Vandermonde matrix $V_{i,j} = x_i^j$. For any $1 \le t \le \frac{n}{2}$, $1 \le s < tn$, and $S \in \mathbb{C}^{n\times n}$ such that $\|S\|_0 \le s$,*

$$\dim^{SS}_t(V - S) \ge \left(n - \frac{s}{t}\right)^t .$$

Proof of Lemma 3.5. Let $J \subset [n]$ be the indices of the t sparsest rows of S, and S_J, V_J to be the restrictions of S and V to the rows in J. By Markov's inequality, each row in S_J has at most $\frac{s}{t}$ non-zero entries.

Now, for each row in $V_J - S_J$, there are at least $(n - \frac{s}{t})$ unchanged entries of the form x_i^j. Note that if $x_1, x_2, \dots, x_n$ are algebraically independent, then any collection of distinct monomials of these variables is linearly independent. As there are at least $(n - \frac{s}{t})^t$ distinct monomials in the set of t-wise products of the entries from $V_J - S_J$, we conclude that $\dim^{SS}_t(V) \ge (n - \frac{s}{t})^t$. $\square$

Finally, Theorem 3.2 follows from Lemma 3.4 and Lemma 3.5.

Proof of Theorem 3.2. For the sake of contradiction, assume that $V = L + S$ where $\operatorname{rank}(L) \le r$ and $\|S\|_0 \le s$. Let us set $t = \frac{n}{2}$ and $s = n\left(n - 100r^2\right)/2$. By Lemma 3.4, we know that

$$\dim^{SS}_t(L) \le \binom{nr+t}{t}^2 \le \left(\frac{e\left(nr + \frac{n}{2}\right)}{\frac{n}{2}}\right)^{\frac{n}{2}} \le (81r^2)^{\frac{n}{2}} .$$

By Lemma 3.5,

$$\dim^{SS}_t(V - S) \ge \left(n - \frac{s}{t}\right)^t = \left(100r^2\right)^{\frac{n}{2}} .$$

Therefore, $\dim^{SS}_t(V - S) > \dim^{SS}_t(L)$, which contradicts the assumption that $V = L + S$. $\square$

3.1.2 Matrix with square roots of distinct primes

Today we will show that a matrix A with square roots of n^2 distinct primes has rigidity $\mathcal{R}^{\mathbb{C}}_A(r) \ge \Omega(n^2)$ for every $1 \le r \le \frac{n}{32}$. **Lecture 10**

Theorem 3.6. *Let $A \in \mathbb{C}^{n\times n}$ be a matrix with square roots of n^2 distinct primes as its entries. For any $1 \le r \le \frac{n}{32}$,*

$$\mathcal{R}^{\mathbb{C}}_A(r) \ge n(n - 16r) .$$

Similarly to the proof of Theorem 3.2, we will show that any matrix with square roots of distinct primes has large Shoup-Smolensky dimension even when some entries of the matrix are changed.

We will use the Besicovitch theorem [Bes40] about linear independence over $\mathbb{Q}$.

Theorem 3.7 (Besicovitch [Bes40])**.** *Let $a_1, a_2, \ldots, a_m$ be m distinct square roots of square-free integers, then they are all linearly independent over $\mathbb{Q}$.*

Lemma 3.8. *Let A be an $n \times n$ matrix with square roots of n^2 distinct primes as its entries, and $S \in \mathbb{C}^{n\times n}$ such that $\|S\|_0 \le s$. For any $1 \le s, t \le n^2$,*

$$\dim_t^{SS}(A - S) \ge \binom{n^2 - s}{t} .$$

Proof of Lemma 3.8. There are at least $n^2 - s$ square roots of distinct primes in the matrix $A - S$. Therefore, there are at least $\binom{n^2-s}{t}$ t-wise products of $A - S$ resulting in distinct square roots of square-free integers. Then, by Theorem 3.7, the t-Shoup-Smolensky dimension of $A - S$ is at least $\dim_t^{SS}(A - S) \ge \binom{n^2-s}{t}$. □

Now we finish the proof of Theorem 3.6.

Proof of Theorem 3.6. Let us set $t = nr$ and $s \le n(n - 16r)$. Assume, for the sake of contradiction, that $A = L + S$ where $\operatorname{rank}(L) \le r$ and $\|S\|_0 \le s$. From Lemma 3.4,

$$\dim_t^{SS}(L) \le \binom{nr + t}{t}^2 \le \binom{2nr}{nr}^2 < 2^{2nr\cdot 2} = 16^{nr} .$$

On the other hand, from Lemma 3.8,

$$\dim_t^{SS}(A - S) \ge \binom{n^2 - s}{t} \ge \left(\frac{16nr}{nr}\right)^{nr} = 16^{nr} .$$

Thus, $\dim_t^{SS}(A - S) < \dim_t^{SS}(L)$, which leads to a contradiction. □

3.1.3 Lower Bounds against Linear Circuits

While the rigidity lower bound from Section 3.1.2 implies a super-linear circuit lower bound against linear circuits for a (not fully explicit) matrix A with square roots of distinct primes, the same technique can be used to directly prove stronger circuit lower bounds for the same matrix A. Shoup and Smolensky [SS91] proved that any linear circuit (of any depth) computing the linear transformation given by such a matrix A must have size at least $\Omega(n^2/\log n)$ (which is optimal as every linear function can me computed by a circuit of size $O(n^2/\log n)$ [Lup56]).

For simplicity, in this section for an $n \times n$ matrix A, by its n^2-Shoup-Smolensky dimension $\dim_{n^2}^{SS}(A)$ we will mean the dimension of the vector space over $\mathbb{Q}$ spanned by products of *at most* n^2 elements of A (rather than exactly n^2 elements of A). We will use the following technical lemma.

Lemma 3.9. *Let C be a linear circuit of size s computing $x \to Bx$ for $B \in \mathbb{C}^{n\times n}$. Then*

$$\dim_{n^2}^{SS}(A) \le (n^2 + 2s)^{2s} .$$

Proof. See the class video and slides for a full proof. □

Now we can prove an optimal circuit lower bound for matrices consisting of square roots of distinct primes.

Theorem 3.10. *Let $A \in \mathbb{C}^{n\times n}$ be a matrix with square roots of n^2 distinct primes as its entries. For any $1 \le r \le \frac{n}{32}$,*

$$\mathcal{R}_A^{\mathbb{C}}(r) \ge n(n - 16r) .$$

Proof. By Besicovitch Theorem (Theorem 3.7), all 2^{n^2} products of subsets of elements of A are linearly independent, therefore

$$\dim^{SS}_{n^2}(A) \geq 2^{n^2}\,.$$

This bounds, together with the bound of Lemma 3.9, implies

$$s \geq \Omega(n^2/\log n)\,.$$

□

3.2 Project Topics

We discussed the following project ideas (see the slides and video for more details): **Lecture 11**

- Provable Cryptography
- Static Data Structures
- Random Algebraic Method

3.3 Rigidity of Hankel and Toeplitz matrices

Lecture 12

Definition 3.11 (Hankel/Toeplitz matrix)**.** *$A \in \mathbb{F}^{n\times n}$ is a Hankel matrix if $A_{i,j} = a_{i+j-1}$ for some $a_1, a_3, \ldots, a_{2n-1} \in \mathbb{F}$. $T \in \mathbb{F}^{n\times n}$ is a Toeplitz matrix if $T_{i,j} = t_{i-j}$ for some $t_{-(n-1)}, t_{-(n-2)}, \ldots, t_{n-1} \in \mathbb{F}$.*

$$A = \begin{pmatrix} a_1 & a_2 & \ldots & a_n \\ a_2 & a_3 & \ldots & a_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ a_n & a_{n+1} & \ldots & a_{2n-1} \end{pmatrix}, \qquad T = \begin{pmatrix} t_0 & t_{-1} & \ldots & t_{-(n-1)} \\ t_1 & t_0 & \ldots & t_{-(n-2)} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n-1} & t_{n-2} & \ldots & t_0 \end{pmatrix}.$$

A random Hankel matrix $A \in \mathbb{F}^{n\times n}$ is a Hankel matrix where $a_1, a_3, \ldots, a_{2n-1} \in \mathbb{F}$ are independent uniformly random elements of $\mathbb{F}$. In Theorem 3.13 we will prove a lower bound on the rigidity of a *random* Hankel matrix, and later in the course we will prove an upper bound on the rigidity of *all* Hankel matrices. These bounds naturally extend to the case of Toeplitz matrices. Note that only $O(n)$ random bits are needed to sample a random Hankel/Toeplitz matrix over $\mathbb{F}_2$. We will need the following generalization of Toeplitz matrices.

Definition 3.12. *$B \in \mathbb{F}^{n\times n}$ is a k-Hankel matrix for $k \in [n]$ if $B_{i,j} = b_{k(i-1)+j}$ for some $b_1, b_2, \ldots, b_{(n-1)k+n} \in \mathbb{F}$.*

$$B = \begin{pmatrix} b_1 & b_2 & \ldots & b_n \\ b_{k+1} & b_{k+2} & \ldots & b_{k+n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{k(n-1)+1} & b_{k(n-1)+2} & \ldots & b_{k(n-1)+n} \end{pmatrix}.$$

For example, 1-Hankel matrix is a usual Hankel matrix from Definition 3.11. In particular, every row of a random 1-Hankel matrix has one new random element. A random n-Hankel matrix is a uniform random matrix with n^2 random entries. In general, every row of a random k-Hankel matrix has k new random entries.

Theorem 3.13 (Rigidity of a random Hankel matrix [GT16])**.** *For any $\sqrt{n} \leq r \leq \frac{n}{32}$, a random Hankel matrix $A \in \mathbb{F}^{n\times n}$ has*

$$\mathcal{R}^{\mathbb{F}_2}_A(r) \geq \Omega\left(\frac{n^3}{r^2\log n}\right)$$

with probability $1 - o(1)$.

The bound of Theorem 3.13 improves on the known *explicit* rigidity bounds (presented in chapter 2) for $r = o\left(\frac{n}{\log n \log\log n}\right)$. For example, for $r = n^{1-\varepsilon}$ for a constant $\varepsilon \in (0, \frac{1}{2})$, Theorem 3.13 gives a lower bound of

$$\mathcal{R}_A^{\mathbb{F}_2}\left(n^{1-\varepsilon}\right) \geq \Omega\left(\frac{n^{1+2\varepsilon}}{\log n}\right),$$

while the bounds of chapter 2 give only $\mathcal{R}_A^{\mathbb{F}_2}\left(n^{1-\varepsilon}\right) \geq \Omega(\frac{n^2}{r} \cdot \log \frac{n}{r}) \geq \Omega(n^{1+\varepsilon} \log n)$.

We will prove Theorem 3.13 in two steps. Let n and k be integers such that k divides n. In the first step, we will show that any $n \times n$ Hankel matrix can be *partitioned* into k^2 matrices of size $\frac{n}{k} \times \frac{n}{k}$, each of which is k-Hankel. In the second step, we will prove that a random k-Hankel matrix is quite rigid with high probability. From these two facts, we will conclude that a random Hankel matrix has high rigidity (with high probability).

Lemma 3.14 (Partitioning of a Hankel matrix)**.** *Let $n, k \in \mathbb{N}$ such that k divides n. Any Hankel matrix $A \in \mathbb{F}^{n\times n}$ can be partitioned into k^2 matrices $A^{i,j} \in \mathbb{F}^{\frac{n}{k}\times\frac{n}{k}}$ for $i, j \in [k]$, s.t. each $A^{i,j}$ is k-Hankel.*

Proof of Lemma 3.14. Let $m = \frac{n}{k}$. For $i, j \in [k]$, let us define the following sets of indices of rows and columns

$$\begin{aligned} I_i &= \{i, i+k, \ldots, i+(m-1)k\}\,; \\ J_j &= \{(j-1)m+1, (j-1)m+2. \ldots, jm\}\,. \end{aligned}$$

We partition the Hankel matrix A into k^2 submatrices $A^{i,j}, i, j \in [k]$ as follows: the matrix $A^{i,j}$ contains all entries of A at the intersections of the rows I_i and the columns J_j. It is easy to see that each $A^{i,j} \in \mathbb{F}^{m\times m}$, and that $\{(I_i, J_j)\}_{i,j\in[m]}$ forms a partition of $[n] \times [n]$.

It remains to show that each $A^{i,j}$ is k-Hankel. Consider the element of the matrix $A^{i,j}$ located at the position (s, t):

$$A^{i,j}_{s,t} = A_{i+(s-1)k,(j-1)m+t}\,.$$

Since A is a Hankel matrix, by Definition 3.11, $A_{i+(s-1)k,(j-1)m+t} = a_{i+(j-1)m-1+(s-1)k+t}$. Now, let $b_\ell = a_{i+(j-1)m-1+\ell}$ for $\ell \in [(m-1)k + m]$. Then

$$A^{i,j}_{s,t} = A_{i+(s-1)k,(j-1)m+t} = a_{i+(j-1)m-1+(s-1)k+t} = b_{(s-1)k+t}\,,$$

which satisfies Definition 3.12. □

Now we prove Theorem 3.13 using Lemma 3.14 and Lemma 3.15.

Proof of Theorem 3.13. We will prove this theorem for every $\sqrt{n} \leq r \leq \frac{n}{32}$ such that $2r$ divides n, the same result (with a larger constant factor hidden in the Ω-notation) for other values of r will follow immediately. Let us set $m = 2r$ and $k = \frac{n}{m}$.

From Lemma 3.14, A can be partitioned into $m \times m$ k-Hankel matrices $\{A^{i,j}\}_{i,j\in[k]}$. For the sake of contradiction, assume that $A = S + L$ where $\|S\|_0 \leq \frac{n^3}{1600r^2 \log n}$ and $\operatorname{rank}(L) \leq r$. By averaging, there exist $i, j \in [k]$ such that $A^{i,j} = S^{i,j} + L^{i,j}$ where

$$\|S^{i,j}\|_0 \leq \frac{\|S\|_0}{k^2} \leq \frac{n}{400 \log n}\,,$$

and $\operatorname{rank}(L^{i,j}) \leq \operatorname{rank}(L) \leq r$. In particular, $A^{i,j}$ is not rigid: $\mathcal{R}^{\mathbb{F}_2}_{A^{i,j}}(r) \leq \frac{n}{400 \log n}$.

However, Lemma 3.15 and union bound imply that with probability $1 - m^2 \cdot 2^{-km/20} = 1 - o(1)$, for every $i, j \in [k]$, $\mathcal{R}^{\mathbb{F}_2}_{A^{i,j}}(m/2) \geq \frac{km}{400 \log m} = \frac{n}{400 \log m} > \frac{n}{400 \log n}$. This contradicts the assumption that $A = S + L$. We conclude that A has rigidity $\mathcal{R}_A^{\mathbb{F}_2}(r) \geq \frac{n^3}{1600r^2 \log n}$ with probability $1 - o(1)$. □

Lecture 13

Lemma 3.15 (Rigidity of k-Hankel matrices)**.** *For any $16 \le k \le m$, a random $m \times m$ k-Hankel matrix B has rigidity*

$$\mathcal{R}_B^{\mathbb{F}_2}(m/2) \ge \frac{km}{400 \log m}$$

with probability at least $1 - 2^{-km/20}$.

Proof of Lemma 3.15. Let S be a fixed $m \times m$ matrix, and $C = B + S$. In the following we will show that $\mathrm{rank}(C) \le m/2$ with probability at most $2^{-km/10}$, let us finish the proof of Lemma assuming this. Since for $s = \frac{km}{400 \log m}$, the number of s-sparse matrices S is bounded from above by $\binom{m^2}{\le s} \le 2^{4s \log m} \le 2^{km/100}$, we conclude that B has rigidity $\mathcal{R}_B^{\mathbb{F}_2}(m/2) \ge \frac{km}{400 \log m}$ with probability at least $1 - 2^{km/100} \cdot 2^{-km/10} \ge 1 - 2^{-km/20}$.

Let C_i denote the i^{th} row of C for $i \in [m]$. Assuming that $\mathrm{rank}(C) \le m/2$, let us greedily pick a row basis of C. Namely, we go from the first to the last rows of C, and include the index of the current row in the set I if and only if the current row is *not* spanned by the previous rows. We have $|I| \le m/2$, and for every $i \in [m] \setminus I$, the row C_i is spanned by the rows $(\{C_{i'}\}_{i' \in I \cap [i-1]})$. In particular,

$$\Pr_B[\mathrm{rank}(C) \le m/2] = \Pr_B\left[\exists I \subseteq [m],\ |I| \le m/2\colon\ \forall i \in [m] \setminus I,\ C_i \in \mathrm{span}(\{C_{i'}\}_{i' \in I \cap [i-1]})\right] \tag{3.16}$$

Now we fix a set $I \subseteq [m]$ of size $|I| \le m/2$, and we will show that

$$\Pr_B\left[\forall i \in [m] \setminus I,\ C_i \in \mathrm{span}(\{C_{i'}\}_{i' \in I \cap [i-1]})\right] \le 2^{-km/8}\,. \tag{3.17}$$

Then a union bound over fewer than 2^m choices of $I \subseteq [m]$ will imply that the expression in Equation 3.16 is upper bounded by $2^{-km/10}$, which will finish the proof.

It remains to show that for every fixed set $I \subseteq [m]$ of size $|I| \le m/2$, Equation 3.17 holds. First, let us greedily choose row indices $1 \le i_1 < i_2 < \cdots < i_\ell \le m$ such that (i) $i_t \notin I$ for each $t \in [\ell]$, and (ii) $i_{t+1} - i_t \ge \frac{m}{k}$ for each $t \in [\ell - 1]$. Namely, $i_1, i_2, \ldots, i_\ell$ are row indices of C that do not belong to the chosen basis I, and the distance between any two rows is at least $\frac{m}{k}$. If we greedily pick indices $i_1, i_2, \ldots, i_\ell$, then each time when we pick one row we remove at most $\frac{m}{k} - 1$ rows, and we have $\ell \ge \frac{m-|I|}{\lceil m/k \rceil} \ge \frac{m/2}{\lceil m/k \rceil} \ge k/4$. For every $t \in [\ell]$, let E_t be the event that $C_{i_t} \in \mathrm{span}(\{C_{i'}\}_{i' \in I \cap [i_t - 1]})$. Then the expression from Equation 3.17 is bounded by

$$\begin{aligned}
&\Pr_B\left[\forall i \in [m] \setminus I,\ C_i \in \mathrm{span}(\{C_{i'}\}_{i' \in I \cap [i-1]})\right] \\
\le &\Pr_B[E_t,\ \forall t \in [\ell]] \\
= &\prod_{t=1}^{\ell} \Pr_B[E_t \mid E_{t'},\ \forall t' < t]\,.
\end{aligned}$$

In what follows, we will prove that $\Pr_B[E_t \mid E_{t'},\ \forall t' < t] \le 2^{-m/2}$, and, since $\ell \ge k/4$, this will finish the proof of Equation 3.17 and the proof of Lemma.

Note that the events $E_{t'},\ \forall t' < t$ depend only on the row $C_{i_{t-1}}$ and above. Therefore, the values of the first i_{t-1} rows of B completely determine the events $E_{t'},\ \forall t' < t$. Since $i_{t-1} \le i_t - \frac{m}{k}$, the values of the first $i_t - \frac{m}{k}$ rows of B also completely determine those events. Since B is k-Hankel, the first $i_t - \frac{m}{k}$ rows of B are specified by elements $b_1, \ldots, b_{k(i_t - 1)}$ (see the picture below). Instead of conditioning on $E_{t'},\ \forall t' < t$, we will prove a stronger statement: for any values $b_1, \ldots, b_{k(i_t-1)}$ of the elements in the first $i_t - \frac{m}{k}$ rows of B, $\Pr_B\left[E_t \mid b_1, \ldots, b_{k(i_t-1)}\right] \le 2^{-m/2}$. (In particular, this holds for all values of $b_1, \ldots, b_{k(i_t-1)}$ that satisfy the events $E_{t'},\ \forall t' < t$.)

$$\begin{array}{cc}
\begin{pmatrix}
b_1 & b_2 & \cdots & b_m \\
b_{k+1} & b_{k+2} & \cdots & b_{k+m} \\
\vdots & \vdots & \ddots & \vdots \\
b_{k(i_t-1)-m+1} & b_{k(i_t-1)-m+2} & \cdots & b_{k(i_t-1)} \\
\vdots & \vdots & \ddots & \vdots \\
b_{k(i_t-1)+1} & b_{k(i_t-1)+2} & \cdots & b_{k(i_t-1)+m} \\
\vdots & \vdots & \ddots & \vdots \\
b_{k(m-1)+1} & b_{k(m-1)+2} & \cdots & b_{k(m-1)+m}
\end{pmatrix}
&
\begin{array}{c}
\\ \\ \\ \text{row } i_t - \frac{m}{k} \\ \\ \text{row } i_t \\ \\ \\
\end{array}
\end{array}$$

Now let the row C_{i_t} be spanned by the basis rows from above it:

$$C_{i_t} = \sum_{j \in I \cap [i_t - 1]} d_j C_j \tag{3.18}$$

for some constants $d_j \in \{0,1\}$. Let us fix the coefficients of this linear combination: fix $d_j \in \{0,1\}$ for each $j \in I \cap [i_t - 1]$. The final observation is that for fixed $b_1, \ldots, b_{k(i_t-1)}$ and fixed d_j, there is a unique assignment to the row B_{i_t} that satisfies Equation 3.18. Indeed, the values of C in the first k columns and the first $i_t - 1$ rows are fixed (as $b_1, \ldots, b_{k(i_t-1)}$ are fixed). Therefore, this linear combination gives a unique assignment to the first k elements of B_{i_t}: to the elements $b_{k(i_t-1)+1}, \ldots, b_{ki_t}$. After this, we have that the first $2k$ elements in the first $i_t - 1$ rows are fixed (as now $b_1, \ldots, b_{ki_t}$ are fixed). Now the fixed linear combination gives a unique assignment to the next k elements of B_{i_t}. We repeat this until we find a unique assignment to B_{i_t} that satisfies Equation 3.18. Thus, for each fixed set of values $d_j \in \{0,1\}$ for $j \in I \cap [i_t - 1]$, the probability of this event is exactly 2^{-m}. The number of such sets $d_j \in \{0,1\}$ is $2^{|I \cap [i_t - 1]|} \le 2^{|I|} \le 2^{m/2}$. By a union bound, this gives us that

$$\Pr_B [E_t \mid E_{t'},\ \forall t' < t] \le \Pr_B \left[E_t \mid b_1, \ldots, b_{k(i_t-1)}\right] \le 2^{m/2} \cdot 2^{-m} = 2^{-m/2}.$$

□

Note that the problem of checking whether a given matrix A has rigidity at least $\mathcal{R}^{\mathbb{F}}_A(r) \ge s$ is in **coNP** for all values of r and s. We can use Theorem 3.13 to construct a matrix M with rigidity $\mathcal{R}^{\mathbb{F}_2}_M(r) \ge \Omega\left(\frac{n^3}{r^2 \log n}\right)$ in $\mathbf{E}^{\mathbf{coNP}} = \mathbf{E}^{\mathbf{NP}}$.[1] Indeed, in time $2^{O(n)}$ we brute force all Hankel matrices, we use **coNP** oracle (or **NP** oracle) to check whether a matrix is rigid. Theorem 3.13 guarantees that this algorithm will find a matrix with the desired rigidity parameters.

In fact, such a matrix can be constructed in time **E** *without* an **NP** oracle. We will prove this in Homework 2.

3.4 Rigidity in sub-exponential time

Lecture 14

Theorem 3.19 ([AKTV18])**.** *Let n be a multiple of $2k$, and let $\mathbb{F}_q$ be the field of size q. In time $q^{O(k^2)}$, one can construct a matrix $A \in \mathbb{F}_q^{n \times n}$ such that*

$$\mathcal{R}^{\mathbb{F}_q}_A(k) \ge \Omega(n^2/\log k).$$

Proof of Theorem 3.19. By Theorem 1.10, there exists a matrix $M \in \mathbb{F}_q^{2k \times 2k}$ of rigidity $\mathcal{R}^{\mathbb{F}_q}_M(k) \ge \Omega(k^2/\log k)$. One can find such a matrix in time $q^{O(k^2)}$ as follows. Iterate over all pairs of matrices $M, S \in \mathbb{F}_q^{2k \times 2k}$. For

[1] When we say that a matrix M belongs to $\mathbf{E}^{\mathbf{NP}}$, we mean that there exists a family of matrices $M_n \in \mathbb{F}^{n \times n}$ for infinitely many values of n such that there is a $2^{O(n)}$-time algorithm with an **NP** oracle that on input 1^n outputs M_n.

every such pair, check whether S is $\Omega(k^2/\log k)$-sparse and $\mathrm{rank}(M+S) \le k$. Find a matrix M for which there is no S with the above conditions, this matrix M has the desired rigidity parameters.

Next we define the matrix $A \in \mathbb{F}_q^{n\times n}$ as $(n/2k)^2$ copies of the matrix $M \in \mathbb{F}_q^{2k\times 2k}$ stacked together. In order to drop the rank of A below k, one has to decrease the rank of each copy of M below k. Therefore, this requires at least

$$\left(\frac{n}{2k}\right)^2 \cdot \Omega(k^2/\log k) = \Omega(n^2/\log k)$$

changes in the entries of A. From this we conclude that $\mathcal{R}_A^{\mathbb{F}_q}\left(\frac{k}{2}\right) \ge \Omega(n^2/\log k)$. □

Corollary 3.20. *For any $\varepsilon > 0$ and large enough n, one can construct in sub-exponential time $2^{O(n^{1-2\varepsilon})}$ a matrix $A \in \mathbb{F}_2^{n\times n}$ such that*

$$\mathcal{R}_A^{\mathbb{F}_2}(n^{\frac{1}{2}-\varepsilon}) \ge \Omega(n^2/\log n)\,.$$

Kumar and Volk [KV20] generalize this result, and show that in time $2^{O(n-\Omega(1/d))}$ one can construct a matrix $A \in \mathbb{E}^{n\times n}$, where $\mathbb{E}$ is the extension of F_q of degree $2^{O(n-\Omega(1/d))}$, such that every depth-d circuit computing $x \to Ax$ must have size at least $n^{1+\Omega(1/d)}$.

3.5 Rigidity of sparse matrices

Recall that for new circuit lower bounds, we would like to construct a matrix M in the complexity class E^{NP} such that $\mathcal{R}_M^{\mathbb{F}}(\varepsilon n) \ge \Omega(n^{1+\delta})$ (for some constants $\varepsilon, \delta > 0$). Although this is still widely open, in this section, we will build such a matrix M in deterministic time $2^{O(n^{1+\delta})}$. For this, we will prove that there exists an $n^{1+\varepsilon}$-sparse matrix with these rigidity parameters, and then we will show how to find it in time $2^{O(n^{1+\delta})}$.

Theorem 3.21. *For any $\varepsilon > 0$, in time $2^{n^{1+\varepsilon}\log n}$, one can construct a matrix $A \in \mathbb{F}_2^{n\times n}$ such that*

$$\mathcal{R}_A^{\mathbb{F}_2}(n/1000) \ge \Omega(n^{1+\varepsilon})\,.$$

In fact, this matrix A will be an $n^{1+\varepsilon}$-sparse matrix.

Theorem 3.22. *For any $\varepsilon > 0$, there exists an $n^{1+\varepsilon}$-sparse matrix $A \in \mathbb{F}_2^{n\times n}$ such that*

$$\mathcal{R}_A^{\mathbb{F}_2}(n/1000) \ge \Omega(n^{1+\varepsilon})\,.$$

First, let us see how Theorem 3.21 is an immediate corollary if Theorem 3.22.

Proof of Theorem 3.21. In time $2^{O(n^{1+\varepsilon}\log n)}$ we enumerate all $n^{1+\varepsilon}$-sparse matrices A and all s-sparse matrices S for some $s = \Omega(n^{1+\varepsilon})$. Now, in time $\mathsf{poly}(n)$ we compute $\mathrm{rank}(A+S)$, and check whether A has rigidity $\mathcal{R}_A^{\mathbb{F}_2}(n/1000) \ge \Omega(n^{1+\varepsilon})$. Theorem 3.22 implies that this algorithm returns a rigid matrix for every large enough n. □

It will prove convenient to work with the following special cases of sparse and rigid matrices.

Definition 3.23. *A matrix $S \in \mathbb{F}^{n\times n}$ is t-regularly sparse if each row and each column of S has at most t non-zero entries. A matrix $M \in \mathbb{F}^{n\times n}$ is (r, s)-regularly rigid, if for every s-regularly sparse matrix S, $\mathrm{rank}(M+S) \ge r$.*

While every (r, s)-rigid matrix is also $(r, s/n)$-regularly rigid, we will show that for linear rank $r = \Omega(n)$ the converse holds too (with a constant loss in the parameters).

Claim 3.24. *If A is $(\varepsilon n, s/n)$-regularly rigid, then A has rigidity $\mathcal{R}_A^{\mathbb{F}}(\varepsilon n/2, \varepsilon s/4)$.*

Proof of Claim 3.24. Assume towards a contradiction that $A = L + S$, where $\mathrm{rank}(L) \le \varepsilon n/2$ and $\|S\|_0 \le \varepsilon s/4$.

Let $I, J \subseteq [n]$ be the sets of indices of the $\varepsilon n/4$ densest rows and $\varepsilon n/4$ densest columns of S, respectively. Let $S_{I,J}$ be an $n \times n$ matrix such that

$$S_{I,J})_{i,j} = \begin{cases} S_{i,j} & \text{if } i \in I \text{ or } j \in J \\ 0 & \text{otherwise} \end{cases}$$

and $S' = S - S_{I,J}$.

First, let us define $L' = L + S_{I,J}$. By sub-additivity of rank, we have $\text{rank}(L') \le \text{rank}(L) + \text{rank}(S_{I,J}) \le \varepsilon n$. Second, by Markov's inequality, the number of non-zero entries in each row and column of S' is at most s/n.

Thus, $A = L + S = (L + S_{I,J}) + (S - S_{I,J}) = L' + S'$, where $\text{rank}(L') \le \varepsilon n$ and S' is s/n-regularly rigid, which contradicts the assumption on regular rigidity of A. □

In order to prove Theorem 3.22, we need to upper bound the number of low-rank regularly-sparse matrices. We will do this via the following encoding argument.

Lemma 3.25 (Row and column space bases encode the entire matrix)**.** *Let $\mathcal{M}_n^r$ denote the set of matrices $M \in \mathbb{F}_2^{n\times n}$ of rank $\text{rank}(M) = r$. The mapping*

$$\phi\colon \mathcal{M}_n^r \to (\mathbb{F}^{1\times n})^r \times (\mathbb{F}^{n\times 1})^r \times [n]^{2r}$$

defined as

$$\phi(M) = (R, C, i_1, \ldots, i_r, j_1, \ldots, j_r)\ ,$$

is a one-to-one mapping, where $R = (\text{Row}_{i_1}(M), \ldots, \text{Row}_{i_r}(M))$ and $C = (\text{Col}_{j_1}(M), \ldots, \text{Col}_{j_r}(M))$ are, respectively, a row space basis and a column space basis of $M \in \mathcal{M}_n^r$ (taking, say, the lexicographically first if multiple bases exist).

Proof of Lemma 3.25. We first claim that the intersection of R and C has full rank, i.e., that the submatrix $M' \in \mathbb{F}^{r\times r}$ obtained by taking rows $i_1, \ldots, i_r$ and columns $j_1, \ldots, j_r$ has rank r. This is a standard fact, we include a proof for completeness. Assume for convenience that $(i_1, \ldots, i_r) = (1, \ldots, r)$ and $(j_1, \ldots, j_r) = (1, \ldots, r)$. Next, assume towards contradiction that $\text{rank}(M') = \text{rank}(\{\text{Col}_1(M'), \ldots, \text{Col}_k(M')\}) = r' < r$. Since C is a column space basis of M, every column $\text{Col}_i(M)$ is a linear combination of vectors from C, and in particular, every $\text{Col}_i(M)^{\le r}$ is a linear combination of $\{\text{Col}_1(M)^{\le r}, \ldots, \text{Col}_r(M)^{\le r}\}$. Therefore, the $r \times n$ submatrix $M'' \coloneqq (\text{Col}_1^{\le r}(M), \ldots, \text{Col}_n^{\le r}(M))$ has rank r'. On the other hand, the r rows of M'': $\text{Row}_1(M), \ldots, \text{Row}_r(M)$ were chosen to be linearly independent by construction. Thus, $\text{rank}(M'') = r > r'$, which leads to a contradiction.

In order to show that ϕ is one-to-one, we show that R and C (together with their indices) uniquely determine the remaining entries of M. We again assume for convenience that $(i_1, \ldots, i_r) = (1, \ldots, r)$ and $(j_1, \ldots, j_r) = (1, \ldots, r)$. Consider any column vector $\text{Col}_i(M)$, $i \in [n] \setminus [r]$. By definition, $\text{Col}_i(M) = \sum_{t=1}^r \alpha_{i,t} \cdot \text{Col}_t(M)$ for some coefficient vector $\alpha_i \coloneqq (\alpha_{i,1}, \ldots, \alpha_{i,r}) \in \mathbb{F}^{r\times 1}$. Thus, in order to completely specify all the entries of $\text{Col}_i(M)$, it suffices to determine the coefficient vector α_i. But M' has full rank, hence the equation

$$M'\alpha_i^T = \text{Col}_i^{\le r}(M)$$

has a *unique* solution. Therefore, the coefficient vector α_i is fully determined by M' and $\text{Col}_i^{\le r}(M)$. Thus, the matrix M can be uniquely recovered from R, C and the indices $\{i_1, \ldots, i_r\}, \{j_1, \ldots, j_r\}$. □

Lemma 3.25 gives us an asymptotically tight bound on the number of regularly-sparse matrices of low rank.

Corollary 3.26 (E.g., [GRW18])**.** *The number of s-regularly sparse matrices of rank r is bounded from above by n^{6rs}.*

Proof of Corollary 3.26. The function ϕ from Lemma 3.25 maps such matrices to $(R, C, i_1, \dots, i_r, j_1, \dots, j_r)$, where R and C are s-sparse bases. Therefore, the total number of such matrices is bounded from above by

$$\left(\binom{n}{s}^r \cdot 2^{rs}\right)^2 \cdot n^{2r} \le (n^{rs} \cdot 2^{rs})^2 \cdot n^{2r} \le n^{6rs}\,.$$

□

Now we are ready to finish the proof of Theorem 3.22.

Proof of Theorem 3.22. By Claim 3.24, it is sufficient to show that there exists an $n^{1+\varepsilon}$-sparse matrix M which is $(n/500, s/n)$-regularly rigid for some $s = \Omega(n^{1+\varepsilon})$. In particular, it suffices to show that there exists such an n^{ε}-regularly sparse matrix M.

We will prove this by the following counting argument. Assume that every n^{ε}-regularly sparse matrix $M = L + S$, where $\operatorname{rank}(L) \le n/500$, and S is s/n-regularly sparse. Then L is $(s/n + n^{\varepsilon})$-regularly sparse. This implies that

$$\begin{aligned}\Big|\{S:\ S \text{ is } s/n\text{-regularly sparse}\}\Big| \times \Big|\{L:\ \operatorname{rank}(L) \le n/500,\ L \text{ is } (s/n + n^{\varepsilon})\text{-regularly sparse}\}\Big| \\ \ge \Big|\{M:\ M \text{ is } n^{\varepsilon}\text{-regularly sparse}\}\Big|\,.\end{aligned}$$

The right-hand side

$$\Big|\{M:\ M \text{ is } n^{\varepsilon}\text{-regularly sparse}\}\Big| \ge \binom{n}{n^{\varepsilon}}^n \ge n^{n^{1+\varepsilon}/10}\,.$$

By Corollary 3.26, for $s = n^{1+\varepsilon}/40$, the left-hand side is upper bounded by

$$\left(\binom{n}{\le s/n}\right)^n \cdot n^{6\cdot(n/500)\cdot(s/n+n^{\varepsilon})} \le n^{2s + 6s/500 + 6n^{1+\varepsilon}/500} \ll n^{n^{1+\varepsilon}/10}\,.$$

This contradiction implies that there exists an $n^{1+\varepsilon}$-sparse matrix A with rigidity

$$\mathcal{R}_A^{\mathbb{F}_2}(n/1000) \ge n^{1+\varepsilon}/40\,.$$

□

3.6 Rigidity from PCPs

In a recent breakthrough line of work, Alman and Chen [AC19], and Bhangale, Harsha, Paradise, and Tal [BHPT20] gave a polynomial-time algorithm that uses an **NP** oracle, and outputs a matrix A of rigidity $\mathcal{R}_A^{\mathbb{F}}(2^{\Theta(2^{\log n/\log\log n})}) \ge \Omega(n^2)$. **Lecture 15**

Theorem 3.27 ([AC19], [BHPT20])**.** *There is a* $\mathbf{P}^{\mathbf{NP}}$ *machine that for infinitely many* n*, on input* 1^n*, outputs a matrix* $M_n \in \mathbb{F}_2^{n\times n}$ *that has rigidity*

$$\mathcal{R}_{M_n}^{\mathbb{F}_2}(r) \ge \Omega(n^2)$$

for $r = 2^{\Omega(\log n/\log\log n)}$.

In fact, this theorem holds over any fixed finite field $\mathbb{F}_q$, but for simplicity we will discuss only the case of $\mathbb{F}_2$. Also, one can notice that the constructed matrices will actually belong to the complexity class **FNP**.[2]

In this section we will sketch the proof of Theorem 3.27, but first we will review some tools used in the proof.

[2]The relation $R(x, y) \in$ **FNP** if there exists a polynomial-time non-deterministic algorithm that for every x outputs y such that $R(x, y) = 1$.

3.6.1 Ingredients

Orthogonal Vectors In the Orthogonal Vectors (OV) problem, one is given two sets S, T of n vectors from $\{0,1\}^d$, and the goal is to check whether there exist $s \in S$ and $t \in T$ such that $\langle s, t\rangle = \sum_{i=0}^{d} s_i t_i = 0$. The trivial brute force algorithm solves the OV problem in time $O(n^2 d)$, and it can also be solved in time $O(n2^d)$. Informally, the Orthogonal Vectors Conjecture states that these two algorithms are essentially optimal (the former is better for large d, while the latter is better for small d). Formally, the conjecture claims that no (randomized) algorithm can solve the OV problem in time $n^{2-\varepsilon}\mathsf{poly}(d)$ for a constant $\varepsilon > 0$. In fact, a sub-quadratic algorithm for OV would imply a better-than-2^n algorithm for the Satisfiability problem [Wil05], and, thus, would refute the celebrated Strong Exponential Time Hypothesis [IP01]. See [Vas18] for more curious connections between the Orthogonal Vectors problem and other algorithmic problems.

For our application, we will be interested in the version of OV over $\mathbb{F}_2$: here two vectors $s \in S$ and $t \in T$ are called orthogonal if $\langle s, t\rangle_2 = \sum_{i=0}^{d} s_i t_i = 0 \mod 2$. While OV over $\mathbb{F}_2$ can be solved in time $O(nd)$ [WY14], we will study a harder version of the problem over $\mathbb{F}_2$. In the #OV problem over $\mathbb{F}_2$ one needs to count the *number* of orthogonal pairs $(s,t) \in S \times T$. We will need a *deterministic* algorithm for solving this problem in time $n^{2-1/O(\log(c))}$, where $c = d/\log n$. This algorithm for #OV over the integers is due to Chan and Williams [CW16], and its simplified version over $\mathbb{F}_2$ is due to Alman and Chen [AC19]. In the proof, we will use the two following results.

Theorem 3.28 (Rectangular matrix multiplication [Cop82, Wil14])**.** *There is a deterministic algorithm that multiplies two matrices $A \in \mathbb{F}^{n\times n^{0.172}}$ and $B \in \mathbb{F}^{n^{0.172}\times n}$ using $n^2\mathsf{poly}\log n$ field operations.*

Theorem 3.29 (Modulus-amplifying polynomials [BT94])**.** *For every ℓ, the following univariate polynomial over $\mathbb{Z}$ of degree $2\ell - 1$*

$$F_\ell(x) = 1 - (1-x)^\ell \sum_{i=0}^{\ell-1} \binom{\ell+i-1}{i} x^i$$

has the property that for every $x \in \mathbb{Z}$,

$$\begin{aligned} x = 0 \mod 2 &\implies F_\ell(x) = 0 \mod 2^\ell \\ x = 1 \mod 2 &\implies F_\ell(x) = 1 \mod 2^\ell \end{aligned}$$

We are ready to present a faster algorithm for the #OV problem over $\mathbb{F}_2$.

Theorem 3.30 ([CW16, AC19])**.** *Let $S, T \subseteq \mathbb{F}_2^d, |S| = |T| = n$ for some $d = 2^{o(\log n/\log\log n)}$. There is a deterministic algorithm that solves $\#OV(S,T)$ over $\mathbb{F}_2$ in time $n^{2-1/O(\log(c))}$ where $c = d/\log n$.*

Proof of Theorem 3.30. Let b be a bucket size parameter to be chosen later. Let us arbitrarily partition the sets S and T into n/b sets of size b:

$$S = S_1 \sqcup S_2 \sqcup \ldots \sqcup S_{n/b} \quad T = T_1 \sqcup T_2 \sqcup \ldots \sqcup T_{n/b}\,,$$

where $|S_i| = |T_i| = n/b$ for every $i \in [n/b]$, and $\sqcup$ denotes disjoint union. Note that

$$\#\mathrm{OV}(S,T) = \sum_{i,j\in[n/b]} \#\mathrm{OV}(S_i, T_j)\,.$$

Thus, it will suffice to solve $(n/b)^2$ #OV problems on sets of size b. We will set $b = n^{1/O(\log(c))}$, and solve each of these smaller problems in amortized time $\mathsf{poly}(\log n)$. This will lead to the total running time $(n/b)^2\mathsf{poly}(\log n) = n^{2-1/O(\log(c))}$.

Let $X, Y \subseteq \mathbb{F}_2^d$ be two sets of size $|X| = |Y| = b$, and let $\ell = 3\log_2 b$. We define the polynomial

$$P(X,Y) = b^2 - \sum_{\substack{x\in X\\ y\in Y}} F_\ell(\langle x, y\rangle)\,,$$

where $\langle x, y\rangle$ is taken over $\mathbb{Z}$, and F_ℓ is the modulus-amplifying polynomial from Theorem 3.29. Note that by the definition of F_ℓ, $F_\ell(\langle x,y\rangle) = \langle x,y\rangle_2 \mod 2^\ell$. Specifically, $F_\ell(\langle x,y\rangle) = 0 \mod 2^\ell$ if x and y are orthogonal (over $\mathbb{F}_2$), and equals $1 \mod 2^\ell$ otherwise. Therefore, $P(X,Y) \mod 2^\ell$ exactly computes the number of orthogonal pairs $(x,y) \in X \times Y$. (Here we implicitly use that $b^2 \ll 2^\ell$ due to our choice of ℓ.)

Now it remains to efficiently compute the values of $P(S_i,T_j)$ for all $i,j \in [n/b]$. Observe that since F_ℓ is a polynomial of degree less than 2ℓ, $F_\ell(\langle x,y\rangle) = \sum_{i=1}^m c_i \cdot \prod_{j\in S_i} x_j y_j$ for some integer m and some sets $S_1,\ldots,S_m \subseteq [d]$ of size $|S_i| < 2\ell$. In particular, we have that $m \le \binom{d}{\le \ell}$.

For any $X \subseteq \mathbb{F}_2^d$ of size $|X| = b$, let $\Phi(X) \in \mathbb{Z}^M$ be a vector defined as

$$\Phi(X) = \left(\sum_{x\in X} c_1 \cdot \prod_{j\in S_1} x_j, \sum_{x\in X} c_2 \cdot \prod_{j\in S_2} x_j, \ldots \sum_{x\in X} c_m \cdot \prod_{j\in S_m} x_j\right).$$

Similarly, for any $Y \subseteq \mathbb{F}_2^d$ of size $|Y| = b$, we define

$$\Psi(Y) = \left(\sum_{y\in Y} \prod_{j\in S_1} y_j, \sum_{y\in Y} \prod_{j\in S_2} y_j, \ldots \sum_{y\in Y} \prod_{j\in S_m} y_j\right).$$

Finally, note that

$$\langle \Phi(X), \Psi(Y)\rangle = \sum_{i=1}^m \sum_{\substack{x\in X\\ y\in Y}} c_i \cdot \prod_{j\in S_i} x_j y_j = \sum_{\substack{x\in X\\ y\in Y}} F_\ell(x,y) = b^2 - P(X,Y).$$

Thus, given $\langle\Phi(X),\Psi(Y)\rangle$, one can efficiently compute $P(X,Y)$. Now let us define two matrices $A \in \mathbb{Z}^{n/b\times m}$ and $B \in \mathbb{Z}^{m\times n/b}$ as follows. The ith row of A is $\Phi(S_i)$, and the ith column of B is $\Psi(T_j)$ for $i \in [n/b]$. The product of these matrices $C = AB \in \mathbb{Z}^{n/b\times n/b}$ has $\langle\Phi(S_i),\Psi(T_j)\rangle$ as its (i,j)th entry. Therefore, given C, one immediately computes $P(S_i,T_j)$ for all $i,j \in [n/b]$, then computes $\#\mathrm{OV}(S_i,T_j)$, and then returns $\#\mathrm{OV}(S,T)$.

It remains to show how to pick the parameter b so that the matrix C could be computed efficiently. Recall that $d = c\log n$. By setting $b = n^{\varepsilon/\log c}$ for a small enough constant $\varepsilon > 0$, we have that $\ell = 3\log b = 3\varepsilon\log n/\log c$, and

$$m \le \binom{d}{\le \ell} \le \left(\frac{de}{\ell}\right)^{2\ell} \le \left(\frac{ce\log n\log c}{3\varepsilon\log n}\right)^{6\varepsilon\log n/\log c} \le (c\log c/\varepsilon)^{6\varepsilon\log n/\log c} < n^{0.172}.$$

Since all the numbers in the matrix C are upper bounded by $m2^{2\ell}$, one can multiply the matrices A and B over a field F_p for a prime $p > m2^{2\ell}$. In particular, all field operations in this field can be performed in time $\mathsf{poly}(\log p) = \mathsf{poly}(\ell,\log m) = \mathsf{poly}(\log n)$. Finally, by Theorem 3.28, A and B of size $n/b \times m$ and $m \times n/b$ can be multiplied over the field F_p in time $(n/b)^2\mathsf{poly}(\log n) = n^{2-2\varepsilon/\log c}\mathsf{poly}\log n = n^{2-2\varepsilon'/\log c}$ for every $d < 2^{o(\log n/\log\log n)}$. □

Lecture 16

Time Hierarchy Theorems

Theorem 3.31 ([HS65]). *For any time-constructible functions t, T satisfying $t(n)\cdot\log t(n) = o(T(n))$*

$$\mathbf{DTIME}[t(n)] \subsetneq \mathbf{DTIME}[T(n)].$$

Proof. For a proof see the slides or in [AB09a, Section 3.1]. □

We will use the classical Time Hierarchy Theorem for non-deterministic machines. To avoid ambiguity, let us denote by $\mathbf{NTime}[f(n)]$ the class of languages L that have non-deterministic witnesses of length $O(f(n))$ and deterministic verifiers running in time $O(f(n))$. Formally, the language $L \in \mathbf{NTime}[f(n)]$ if there exists a constant $c > 0$, a deterministic algorithm A such that $x \in \{0,1\}^n$ is in L if and only if $\exists y, |y| \le c\cdot f(n)$, and $A(x,y)$ accepts in time $c\cdot f(n)$. We will say that a language $L \in \{0,1\}^*$ is *unary* if L only contains strings of the form 1^n.

Theorem 3.32 ([Coo72, Žák83])**.** *For any time-constructible functions t, T satisfying $t(n+1) = o(T(n))$*

$$\mathbf{NTime}[t(n)] \subsetneq \mathbf{NTime}[T(n)]\,.$$

Moreover, there exists a unary language $L \subseteq \{1\}^*, L \in \mathbf{NTime}[T(n)] \setminus \mathbf{NTime}[t(n)]$.

Proof. For a proof see the slides or in [AB09a, Section 3.2]. □

Corollary 3.33. *There exists a unary language $L \subseteq \{1\}^*, L \in \mathbf{NTime}[2^n] \setminus \mathbf{NTime}[2^n/n]$.*

Lecture 17

Rectangular PCPs We refer the reader to [AB09b, Section 11] for a gentle introduction to Probabilistically Checkable Proofs (PCP). Below we define all relevant parameters of PCPs.

Definition 3.34 (PCP Verifier)**.** *A PCP verifier for a language $L \in \{0,1\}^*$ is a probabilistic algorithm V that on input $x \in \{0,1\}^n$ uses r bits of randomness and generates q queries $I = (i_1, \ldots, i_q)$, where each $i_j \in [m]$, and a predicate $D\colon \{0,1\}^q \to \{0,1\}$. The verifier V for each input x will query the proof $\pi \in \{0,1\}^m$ at positions from I, and V will accept this proof if $D(\pi_I) = 1$, and reject otherwise.*

- **Completeness:**

$$\text{If } x \in L \qquad \Longrightarrow \qquad \exists \pi \Pr[V \text{ accepts}] = 1\,.$$

- **Soundness:** *For a parameter $s \in (0,1]$,*

$$\text{If } x \notin L \qquad \Longrightarrow \qquad \forall \pi \Pr[V \text{ accepts}] < s\,.$$

- **Randomness complexity:** *For every input $x \in \{0,1\}^n$, V uses at most $r(n)$ bits of randomness.*
- **Query complexity:** *For every input $x \in \{0,1\}^n$, V makes at most $q(n)$ queries to π.*
- **Verifier runtime:** *For every input $x \in \{0,1\}^n$, V runs in time $t(n)$.*
- **Proof length:** *For every input $x \in \{0,1\}^n, x \in L$, there exists a correct proof π of length at most $m(n)$.*
- **Decision complexity:** *For every input $x \in \{0,1\}^n$, the predicate D can be computed in time $d(n)$.*
- **Smoothness:** *For every input $x \in \{0,1\}^n$, for an index $i \in [m]$, let $Q_x(i)$ be the probability that V queries the bit i of π on a random query number $k \in [q]$:*

$$Q_x(i) = \Pr_{R,k}[i_k = i]\,.$$

  V is smooth if for every $x \in \{0,1\}^n, i \in [m], j \in [m]$, $Q_x(i) = Q_x(j)$.

- **τ-rectangular:**[3] *Let the length of the proof π be a perfect square: $m = \ell^2$. We will think of π as a matrix $\pi \in \{0,1\}^{\ell \times \ell}$. The r random bits $R \in \{0,1\}^r$ of V are partitioned into three groups: $R_{shared} \in \{0,1\}^{\tau r}, R_{row}, R_{col} \in \{0,1\}^{(1-\tau)r/2}$. Given an input $x \in \{0,1\}^n$ and a proof $\pi \in \{0,1\}^{\ell \times \ell}$, the verifier V works as follows.*

  1. *Samples $R_{shared \in \{0,1\}^{\tau r}}$. Uses x and R_{shared} to generate a predicate $D\colon \{0,1\}^{q+p} \to \{0,1\}$, and a constant number p of linear functions $C_1, \ldots, C_p$ of* all *randomness $R \in \{0,1\}^r$.*
  2. *Samples $R_{row} \in \{0,1\}^{(1-\tau)r/2}$. Uses x, R_{shared}, and R_{row} to generate the first indices of the q queries to π: $i_1^{row}, \ldots, i_q^{row}$.*
  3. *Samples $R_{col} \in \{0,1\}^{(1-\tau)r/2}$. Uses x, R_{shared}, and R_{col} to generate the second indices of the q queries to π: $i_1^{col}, \ldots, i_q^{col}$.*

[3] For ease of exposition, we merge two different definitions of τ-rectangular and τ-randomness-oblivious predicates from [BHPT20] into the definition of τ-rectangular verifiers. Moreover, we fix some parameters from those definitions. These simplifications are sufficient for our application, but we refer the reader to [BHPT20] for constructions of more general PCPs.

4. *Queries $\pi \in \{0,1\}^{\ell\times\ell}$ at positions $(i_1^{row}, i_1^{col}), \dots, (i_q^{row}, i_q^{col})$, receives bits $b_1, \dots, b_q \in \{0,1\}^q$.*
5. *Evaluates the functions $C_1(R), \dots, C_p(R)$.*
6. *Outputs the result of the decision predicate $D(b_1, \dots, b_q, C_1(R), \dots, C_p(R))$.*

Now we present without a proof a construction of a PCP from [BHPT20] which simultaneously achieves essentially optimal values of all of these parameters.

Theorem 3.35 ([BHPT20])**.** *For any $L \in \mathbf{NTime}[2^n]$ and constants $s \in (0, 1/2)$ and $\tau \in (0,1)$, L has a PCP verifier with*

- *Soundness s;*
- *Randomness complexity $r(n) = n + O(\log n)$;*
- *Constant query complexity and decision complexity;*
- *Verifier runtime $t(n) = 2^{\tau n}$;*
- *Proof length $m(n) = 2^n \cdot \mathsf{poly}(n)$;*
- *V is smooth;*
- *τ-rectangular.*

Lecture 18

Corollary 3.36 (of Theorem 3.30)**.** *Let $A \in \mathbb{F}_2 n \times d$ and $B \in \mathbb{F}_2^{d_n}$ be two matrices. There is a deterministic algorithm that computes the number of zeros (and the number of ones) in AB in time $n^{2-1/O(\log(c))}$ where $c = d/\log n$.*

3.6.2 Proof of Theorem 3.27

Proof of Theorem 3.27. By Corollary 3.33, there exists a unary language $L \subseteq \{1\}^*, L \in \mathbf{NTime}[2^n] \setminus \mathbf{NTime}[2^n/n]$. By Theorem 3.35, L has a rectangular PCP verifier V with a proof of length $m(n) = \ell(n)^2$. We now describe a $\mathbf{P}^{\mathbf{NP}}$ algorithm $\mathcal{A}$ that for infinitely many inputs 1^N outputs rigid matrices $M \in \mathbb{F}_2^{N\times N}$. We will only consider (infinitely many) values of N such that (i) $N = \ell(n)$ for some n, and (ii) the string $1^n \in L$. For each such N, $\mathcal{A}$ outputs the proof $\pi \in \{0,1\}^{\ell(n)\times\ell(n)}$ for $1^n \in L$.[4] It remains to show that (i) $\mathcal{A}$ can be implemented in $\mathbf{P}^{\mathbf{NP}}$, and (ii) $\mathcal{A}$ outputs a rigid matrix infinitely often.

On input 1^N, the algorithm $\mathcal{A}$ guesses $\pi \in \{0,1\}^{\ell\times\ell}$ one bit at a time and uses the **NP** oracle to verify the guesses. Thus, $\mathcal{A}(1^N)$ rune in time $\mathsf{poly}(\ell^2) = \mathsf{poly}(N^2) = \mathsf{poly}(N)$.

Let q and s be the query complexity and soundness constants of the PCP verifier V, and let $\varepsilon = (1-s)/q$. Let us now show that the output of $\mathcal{A}$ is rigid infinitely often. Assume, towards a contradiction, that there exists n_0 such that for all $n > n_0$, if $1^n \in L$, then some proof $\pi \in \{0,1\}^{N\times N}$ is not rigid: $\mathcal{R}_\pi^{\mathbb{F}_2}(\Gamma) < \varepsilon N^2$ for $\Gamma = 2^{O(\log N/\log\log N)}$. Then there exist matrices $A \in \mathbb{F}_2^{N\times\Gamma}, B \in \mathbb{F}_2^{\Gamma\times N}$ such that $\|\pi - AB\|_0 \le \varepsilon N^2$. Since V is a smooth verifier, it queries each of the positions of π with equal probability. By union bound, it queries one of the εN^2 positions where π and AB differ only with probability $\le q\varepsilon = (1-s)$. Therefore, even given AB as a proof (instead of π), the probability of acceptance of this proof by V is $\ge s$, while for every wrong proof π', the probability of acceptance is $< s$.

It remains to show that one can compute the probability of acceptance of AB by V in non-deterministic time $O(2^n/n)$ (this will contradict the initial assumption that $L \notin \mathbf{NTime}[2^n/n]$). First, we can guess the matrices $A \in \mathbb{F}_2^{N\times\Gamma}, B \in \mathbb{F}_2^{\Gamma\times N}$ in time $O(N\Gamma)$. Then we brute force the shared randomness $R_{\text{shared}} \in \{0,1\}^{\tau r}$ of V, and for each R_{shared} do the following.

1. Compute the predicate $D\colon \{0,1\}^{q+p} \to \{0,1\}$, and the linear functions $C_1, \dots, C_p$ for a constant p.
2. For each query index $k \in [q]$, we define a matrix $A^{(k)} \in \mathbb{F}_2^{2^{(1-\tau)r/2}\times\Gamma}$ as follows. For each row randomness $R_{\text{row}} \in \{0,1\}^{(1-\tau)r/2}$, we set the corresponding row of $A^{(k)}$ to be the i_k^{row} row of A.

[4] If there are several proofs $\pi \in \{0,1\}^{\ell(n)\times\ell(n)}$, then we take the lexicographically first one.

3. For each query index $k \in [q]$, we define a matrix $B^{(k)} \in \mathbb{F}_2^{\Gamma \times 2^{(1-\tau)r/2}}$ as follows. For each column randomness $R_{\text{col}} \in \{0,1\}^{(1-\tau)r/2}$, we set the corresponding column of $B^{(k)}$ to be the i_k^{col} column of B.

4. Finally, for $j \in [p]$, we define vectors $A_{q+j} \in \mathbb{F}_2^{2^{(1-\tau)r/2} \times 1}$ and $B \in \mathbb{F}_2^{1 \times 2^{(1-\tau)r/2}}$ such that the linear function $C_j(R_{\text{row}}, R_{\text{col}}) = (A_{q+j}B_{q+j})_{R_{\text{row}}, R_{\text{column}}}$.

5. It is now easy to verify that for every randomness $R = (R_{\text{shared}}, R_{\text{row}}, R_{\text{col}})$, V accepts if and only if

$$D\left(\left(A^{(1)}B^{(1)}\right)_{R_{\text{row}},R_{\text{col}}}, \left(A^{(2)}B^{(2)}\right)_{R_{\text{row}},R_{\text{col}}}, \ldots, \left(A^{(q+p)}B^{(q+p)}\right)_{R_{\text{row}},R_{\text{col}}}\right) = 1\,.$$

6. Recall that $D\colon \{0,1\}^{q+p} \to \{0,1\}$ is a function of a constant number of inputs computable in constant time. In particular, we can compute the Fourier coefficients of D in constant time. From the Fourier representation of

$$D(y_1, \ldots, y_{q+p}) = \sum_{K \subseteq [q+p]} \hat{D}(K)(-1)^{\oplus_{i \in K} y_i}\,,$$

in order to compute the expected value of

$$D\left(\left(A^{(1)}B^{(1)}\right)_{R_{\text{row}},R_{\text{col}}}, \left(A^{(2)}B^{(2)}\right)_{R_{\text{row}},R_{\text{col}}}, \ldots, \left(A^{(q+p)}B^{(q+p)}\right)_{R_{\text{row}},R_{\text{col}}}\right)$$

for random $R_{\text{row}}, R_{\text{col}}$, we can just compute the expected value of all $\oplus_{i \in K}\left(A^{(i)}B^{(i)}\right)_{R_{\text{row}},R_{\text{col}}}$. This value is exactly the fraction of ones in the product $A'B'$ where $A' \in \mathbb{F}_2^{2^{(1-\tau)r/2} \times |K|\Gamma}$ and $B' \in \mathbb{F}_2^{|K|\Gamma \times 2^{(1-\tau)r/2}}$ are concatenations of $A^{(i)}$ and $B^{(i)}$ for $i \in K$.

Recall that t denotes the running time of V. The running time of guessing A and B is $O(N\Gamma)$, the running time of step 1 is $O(t)$, the running time of steps 2–4 is $O(2^{(1-\tau)r/2}(t+\Gamma))$. By Corollary 3.36, the running time of step 6 is $\left(2^{(1-\tau)r/2}\right)^{2-1/\log(\Gamma(q+p))}$. Since these steps of the algorithm are repeated $2^{\tau r}$ times, the total running time is

$$O\left(N\Gamma + 2^{\tau r}\left(2^{(1-\tau)r/2}(t+\Gamma) + \left(2^{(1-\tau)r/2}\right)^{2-1/\log(\Gamma(q+p))}\right)\right)\,.$$

Recall that the rigidity rank parameter $\Gamma = 2^{\Theta(\log N/\log\log N)}$, and from the PCP parameters from Theorem 3.35, $N = \ell = \sqrt{m} = 2^{n/2}\mathsf{poly}(n)$, $r = n + O(\log n)$, $q, p, \tau = \Theta(1)$. Then the running time is bounded from above by the term

$$O\left(2^{\tau r}\left(2^{(1-\tau)r/2}\right)^{2-1/\log(\Gamma(q+p))}\right) = O\left(2^{r-(1-\tau)r/(2\log\Gamma)}\right) = O(2^n/n)$$

for some $\Gamma = 2^{\Theta(\log N/\log\log N)}$. Thus, the language L can be decided in non-deterministic time $O(2^n/n)$ which contradicts the assumption that $L \notin \mathbf{NTime}[2^n/n]$. □

3.7 Summary

Recall that for circuit lower bounds one needs to find a matrix $A \in \mathbb{F}^{n \times n}$ with rigidity $\mathcal{R}_A^{\mathbb{F}}(\varepsilon n) \geq n^{1+\delta}$. In the table below we summarize the semi-explicit constructions of rigid matrices.

construction	rigidity	running time	reference
Vandermonde with algebraically independent entries	$\mathcal{R}(\sqrt{n}) \geq \delta n^2$	NA	Theorem 3.2
square roots of primes	$\mathcal{R}(\varepsilon n) \geq \delta n^2$	NA	Theorem 3.6
brute force	$\mathcal{R}(\varepsilon n) \geq n^2/\log n$	2^{n^2}	section 1.5
explicit	$\mathcal{R}(r) \geq \frac{n^2}{r} \cdot \log \frac{n}{r}$	$\mathsf{poly}(n)$	chapter 2
Hankel	$\mathcal{R}(r) \geq \frac{n^3}{r^2 \log n}$	2^n	Theorem 3.13
sub-exponential	$\mathcal{R}(n^{0.5-\varepsilon}) \geq n^2/\log n$	$2^{n^{1-\varepsilon}}$	Corollary 3.20
sparse	$\mathcal{R}(\varepsilon n) \geq n^{1+\delta}$	$2^{n^{1+\delta}\log n}$	Theorem 3.21
PCP	$\mathcal{R}(2^{\log n/\log\log n}) \geq \delta n^2$	$\mathbf{P}^{\mathbf{NP}}$	Theorem 3.27

Table 3.1: Summary of semi-explicit rigidity lower bounds.

Chapter 4

Limitations

Matrix rigidity lower bounds are a tool for proving hardness of computation, but in this chapter we focus on hardness of proving rigidity lower bounds. Despite more than forty years of research, the best known explicit lower bounds on rigidity for the interesting case of $r = \Omega(n)$ are still only linear in n. The results of the last few years partially explain this barrier: Apparently, many of the known approaches are not capable of proving stronger bounds, and some matrices previously conjectured to be rigid are not sufficiently rigid for circuit lower bounds. **Lecture 19**

4.1 Limitation of untouched minor method

Recall that the explicit $\Omega\left(\frac{n^2}{r}\log\frac{n}{r}\right)$ lower bound of [SSS97] was obtained by arguing that there exists a constant $c > 0$ such that for any $\log n \le r \le n$ and any $c \cdot \frac{n^2}{r}\log\frac{n}{r}$ entries in an $n \times n$ matrix, there must be an $r \times r$ submatrix untouched by these entries. Thus, the rigidity lower bound immediately follows from explicit constructions of matrices whose $r \times r$ submatrices are all full rank. This method of constructing rigid matrices is called the *untouched minor method*.

In this subsection, we are going to give a simple proof showing that this $\Omega\left(\frac{n^2}{r}\log\frac{n}{r}\right)$ lower bound is the best lower bounds that the untouched minor method can prove. Therefore, for better lower bounds we have to use different methods.

Theorem 4.1 ([Lok00, Lok09])**.** *There exists a constant $c > 0$ such that for any large enough n and $\log n \le r \le n$, there exists a set S of $c \cdot \frac{n^2}{r}\log\frac{n}{r}$ entries of an $n \times n$ matrix such that its every $r \times r$ submatrix intersects S.*

Proof of Theorem 4.1. For any n, r, let $U = [n] \times [n]$ be the set of entries in an $n \times n$ matrix and $V = \{X \times Y : \ X, Y \subseteq [n], \ |X| = |Y| = r\}$ be the set of $r \times r$ submatrices. Consider a bipartite graph G with vertex sets U and V. For any entry $(i, j) \in U$ and submatrix $X \times Y \in V$, there is an edge in G between them if $(i, j) \in X \times Y$. In order to prove this theorem, it suffices to show that there exists a constant $c > 0$ such that for any large enough n and $\log n \le r \le n$, there exists $S \subseteq U$ of size at most $c \cdot \frac{n^2}{r}\log\frac{n}{r}$ such that the set S touches all vertices in V.

There are two simple ways of proving the existence of such covering set S: a non-explicit construction via the probabilistic method, and an explicit construction via a greedy algorithm. Here, we give a short proof using the probabilistic method.

We pick a random set of s vertices in U for a parameter s to be chosen later, and we will show that the probability of having an uncovered vertex in V is strictly less than 1. This will imply that there exist a choice of s vertices in U that cover all vertices of V.

For a vertex v of a graph $G = (V, E)$, by $N(v)$ we denote the neighborhood of v, i.e., $N(v) = \{u : (v, u) \in E\}$. In the bipartite graph G, every vertex in V has degree $d = r^2$, and, thus, for a fixed $v \in V$, the probability that v is not covered by a random vertex in U is

$$\Pr_{u \in U}[v \notin N(u)] = 1 - \frac{d}{|U|} .$$

By union bound, we have

$$\Pr_{u_1,u_2,\ldots,u_s\in U}\left[\exists v\in V,\ v\notin N(u_i)\ \forall i\in[s]\right]\le |V|\cdot\left(1-\frac{d}{|U|}\right)^s .$$

The right hand side of the above inequality is less than 0.1 if we pick

$$s=O\left(\frac{|U|}{d}\log|V|\right)=O\left(\frac{n^2}{r^2}\log\left(\binom{n}{r}^2\right)\right)=O\left(\frac{n^2}{r}\log\frac{n}{r}\right) .$$

We conclude that there exists a set $S\subseteq U$ of size at most $s=c\cdot\frac{n^2}{r}\log\frac{n}{r}$ such that every vertex in V is covered by S. ☐

4.2 Non-rigid super regular matrices

In Section 4.1, we showed that one cannot prove strong rigidity lower bounds by arguing that a sparse matrix does not touch some $r\times r$ submatrix. Apparently, an even stronger limitation holds for the untouched minor method: there exists a matrix $A\in\mathbb{F}^{n\times n}$ such that *all* square submatrices of all sizes of A have full rank, yet $\mathcal{R}^{\mathbb{F}}_A(\varepsilon n)<n^{1+o(1)}$ for every constant $\varepsilon>0$.

Definition 4.2. *A matrix $A\in\mathbb{F}^{n\times n}$ is* super regular *if all of its square submatrices have full rank.*

Super regular matrices were used for super-linear lower bounds against circuits of bounded depth [Pip77, DDPW83, AP94, Pud94, Lok95, RTS00, RS03, Che08] (and they were also used in section 2.3 for proving the best know rigidity lower bounds). In this section we show a result of Valiant [Val75] showing that some super regular matrices have low rigidity (and, in fact, can be computed by circuits of linear size).

The study of super regular matrices is closely related to the study of superconcentrator graphs.

Definition 4.3 (Superconcentrator)**.** *Let G be a graph, and let I and O be two disjoint subsets of vertices of G called the inputs and outputs, respectively. G is a* superconcentrator *if for any $1\le k\le\min\{|I|,|O|\}$, $I'\subset I$ and $O'\subseteq O$ of size $|I'|=|O'|=k$, there exist k vertex-disjoint paths from I' to O'. The size of a superconcentrator G is the number of edges in G.*

The linear map given by a super regular matrix M can be only computed by a linear circuit whose graph G is a superconcentrator. Indeed, assume that for some set of inputs $I'\subset I$ and some set of outputs $O'\subseteq O$ of G, where $|I'|=|O'|=k$, there are only $k-1$ vertex disjoint paths from I' to O'. By Menger's theorem (Theorem 3.3.1 in [Die05]), there exists a vertex cut of size $k-1$: nodes $C=\{c_1,\ldots,c_{k-1}\}$ of G such that the outputs O' depend on the inputs I' only through the nodes C. In particular, the $k\times k$ submatrix of M corresponding to $I'\times O'$ is generated by the $k-1$ columns describing the linear combinations computed in the nodes C. Therefore, this submatrix has rank at most $k-1$ which contradicts super regularity of M. This implies that super regular matrices require superconcentrator circuits. Later in this section we will show that some form of the opposite statement holds as well: given a superconcentrator one can construct a super regular matrix with related rigidity parameters. Valiant conjectured that superconcentrators must have super-linear size, and, thus, any super regular matrix would require circuits of super-linear size. (Note that we know explicit super regular matrices, see, e.g., section 2.3.) But then Valiant himself found a counterexample [Val75] to his conjecture: a superconcentrator of size $O(n)$.

Next, we will show that there exist superconcentrators of linear size, and then we will see that there exist super regular matrices with low rigidity. As a building block we will use another pseudorandom object—bipartite expanders.

Definition 4.4 ((n,m)-bipartite expander)**.** *For any $n,m\in\mathbb{N}$ a (n,m)-bipartite expander is a bipartite graph $E_{n,m}$ with vertex sets U and V, where $|U|=n$ and $|V|=m$, such that for any $S\subseteq U$, $|S|\le\frac{n}{2}$, $|N(S)|\ge|S|$.*

A simple probabilistic arguments shows that bipartite expanders exist, we defer the proof of this lemma until the end of this section.

Lemma 4.5. *For any large enough n and $m=\lceil\frac{3n}{4}\rceil$, there exists an (n,m)-bipartite expander $E_{n,m}$ with at most $10n$ edges.*

Theorem 4.6 (Superconcentrators of linear size [Val75])**.** *For any $n \in \mathbb{N}$ large enough, there exists a superconcentrator G_n of size $O(n)$.*

Proof of Theorem 4.6. We will construct a superconcentrator by induction on n. For $n \le C$ for a large constant C, we observe that a complete bipartite graph with n inputs and n outputs is a superconcentrator. For $n \ge C$, we assume that there exists a superconcentrator G_m with m inputs, m outputs, and of size $C'm$, where $m = \lceil \frac{3n}{4} \rceil$ and $C' = \max\{C, 85\}$.

Now, we define G_n, the superconcentrator with input and output vertex sets $I_n = [n]$ and $O_n = [n]$, from G_m as follows.

1. Connect $u \in I_n$ and $v \in O_n$ if $u = v$.
2. Add a copy of the graph G_m with the input and output vertex sets I_m and O_m.
3. Connect I_n and I_m by an (n, m)-bipartite expander $E_{n,m}$. Similarly, connect O_n and O_m by an (n, m)-bipartite expander.

G_n has edges of three types, and the size of G_n (the number of edges in it) can be upper bounded as follows.

$$\text{size}(G_n) \le n + \text{size}(G_m) + 2 \cdot |E_{n,m}| \le 21n + \frac{3C'n}{4} \le C'n\,,$$

where we used Lemma 4.5 for a bound on the size of (n, m)-bipartite expanders, and the bound $C' \ge 85$. Therefore, G_n indeed has linear size. It remains to show that G' is a superconcentrator.

We will show that for any $k \in [n]$ and $S \subseteq I_n$, $T \subseteq O_n$ where $|S| = |T| = k$, there are k vertex disjoint paths from S to T. Let us consider two cases: (a) $k \le \frac{n}{2}$ and (b) $k > \frac{n}{2}$.

(a) When $k \le \frac{n}{2}$, by the expander property, we know that S has at least k neighbors S' in I_m, and T has at least k neighbors T' in O_m. Next, since G_m is a superconcentrator, we know that there are k vertex disjoint path from S' to T'. This gives us k vertex-disjoint paths from S to T.

(b) When $k > \frac{n}{2}$, let $S_\Delta = S \backslash (S \cap T)$ and $T_\Delta = T \backslash (S \cap T)$. We have that $|S_\Delta| = |T_\Delta| \le \frac{n}{2}$ and from the case (a), there are $|S_\Delta|$ vertex disjoint paths from S_Δ and T_Δ. Also, by the construction of G_n, there are edges connecting both sides of $S \cap T$. Thus, there are $|S_\Delta| + |S \cap T| = k$ vertex-disjoint paths from S to T.

This finishes our construction of superconcentrators of linear size. We note that the only non-explicit part of the presented construction is the bipartite expander graphs, and that this part can actually be made explicit, too. □

For completeness, we conclude this section with a proof of Lemma 4.5.

Proof of Lemma 4.5. We will prove this lemma using the probabilistic method. Let us sample a random bipartite graph with n left vertices and $m = \lceil \frac{3n}{4} \rceil$ right vertices, where each left vertex has 10 random neighbors on the right. It suffices to show that with non-zero probability, for all $2 \le k \le \frac{n}{2}$, any $S \subset [n]$ of size $|S| = k$ has $|N(S)| \ge k$.

For a fixed $S \subseteq [n]$ of size $|S| = k$, the probability of $|N(S)| < |S|$ is

$$\begin{aligned} \Pr\left[\exists T \subseteq [m],\ |T| < k,\ N(S) \subseteq T\right] &\le \Pr\left[\exists T \subseteq [m],\ |T| = k,\ N(S) \subseteq T\right] \\ &\le \binom{m}{k} \cdot \left(\frac{|T|}{m}\right)^{10|S|} \le \binom{m}{k} \cdot \left(\frac{k}{m}\right)^{10k} . \end{aligned}$$

Thus, the probability that there exists an $S \subseteq [n]$, of size $|S| \le \frac{n}{2}$ that has $|N(S)| < |S|$, is at most

$$\sum_{k=2}^{n/2} \binom{n}{k} \cdot \binom{m}{k} \cdot \left(\frac{k}{m}\right)^{10k} \le \sum_{k=2}^{n/2} \left(\frac{e^2 n m k^{10}}{k^2 m^{10}}\right)^k \le \sum_{k=2}^{n/2} \left(\frac{e^2 \left(\frac{1}{2}\right)^8}{\left(\frac{3}{4}\right)^9}\right) \le \sum_{k=2}^{n/2} \left(\frac{1}{2}\right)^k < 1\,.$$

From this, we conclude that there exist (n, m)-bipartite expanders. □

Bibliography

[AB09a] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[AB09b] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[AC19] Josh Alman and Lijie Chen. Efficient construction of rigid matrices using an NP oracle. In *Proceedings of the 60th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2019)*, pages 1034–1055. IEEE, 2019.

[AKTV18] Josh Alman, Mrinal Kumar, Avishay Tal, and Ben Lee Volk. Personal communication, 2018.

[Alo94] Noga Alon. Manuscript., 1994. See [Juk11, Lemma 14.6].

[AP94] Noga Alon and Pavel Pudlák. Superconcentrators of depths 2 and 3; odd levels help (rarely). *Journal of Computer and System Sciences*, 48(1):194–202, 1994.

[Bes40] Abram S. Besicovitch. On the linear independence of fractional powers of integers. *Journal of the London Mathematical Society*, 1(1):3–6, 1940.

[BHPT20] Amey Bhangale, Prahladh Harsha, Orr Paradise, and Avishay Tal. Rigid matrices from rectangular pcps. In *Proceedings of the 61st Annual IEEE Symposium on Foundations of Computer Science (FOCS 2020)*. IEEE, 2020.

[Blu83] Norbert Blum. A Boolean function requiring $3n$ network size. *Theoretical Computer Science*, 28(3):337–345, 1983.

[Bol04] Béla Bollobás. *Extremal graph theory*. Courier Corporation, 2004.

[Bop97] Ravi B. Boppana. The average sensitivity of bounded-depth circuits. *Inf. Process. Lett.*, 63(5):257–261, 1997.

[BPR07] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. *Algorithms in real algebraic geometry*, volume 10. Springer Science & Business Media, 2007.

[BT94] Richard Beigel and Jun Tarui. On ACC. *Computational Complexity*, 4(4):350–366, 1994.

[Can88] John Canny. Some algebraic and geometric computations in pspace. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC 1988)*, pages 460–467. ACM, 1988.

[Cha94a] Aleksandr V. Chashkin. On the complexity of Boolean matrices, graphs and their corresponding Boolean functions. *Diskretnaya matematika*, 4(3):229–257, 1994.

[Cha94b] Aleksandr V. Chashkin. On the complexity of Boolean matrices, graphs and their corresponding Boolean functions. *Discrete Math. and Appl.*, 4(3):229–257, 1994.

[Che05] Mahdi Cheraghchi. On matrix rigidity and the complexity of linear forms. *Electronic Colloquium on Computational Complexity (ECCC)*, 2005.

[Che08] Dmitriy Yu. Cherukhin. Lower bounds for Boolean circuits with finite depth and arbitrary gates. *"Electronic Colloquium on Computational Complexity (ECCC)"*, 2008.

[Cod00] Bruno Codenotti. Matrix rigidity. *Linear Algebra and its Applications*, 304(1-3):181–192, 2000.

[Coo72] Stephen A. Cook. A hierarchy for nondeterministic time complexity. In *Proceedings of the 4th Annual ACM Symposium on Theory of Computing (STOC 1972)*, pages 187–192. ACM, 1972.

[Cop82] Don Coppersmith. Rapid multiplication of rectangular matrices. *SIAM Journal of Computing*, 11(3):467–471, 1982.

[CW16] Timothy M Chan and Ryan Williams. Deterministic APSP, orthogonal vectors, and more: Quickly derandomizing Razborov-Smolensky. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2016)*, pages 1246–1255. SIAM, 2016.

[DDPW83] Danny Dolev, Cynthia Dwork, Nicholas Pippenger, and Avi Wigderson. Superconcentrators, generalizers and generalized connectors with limited depth. In *Proceedings of the fifteenth annual ACM symposium on Theory of computing*, pages 42–51, 1983.

[De 06] Ronald De Wolf. Lower bounds on matrix rigidity via a quantum argument. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP 2006)*, pages 62–71. Springer, 2006.

[Die05] Reinhard Diestel. Graph theory. 2005. *Graduate Texts in Math*, 101, 2005.

[DK11] Evgeny Demenkov and Alexander S Kulikov. An elementary proof of a $3n - o(n)$ lower bound on the circuit complexity of affine dispersers. In *Proceedings of the 36th Internationl Symposium on the Mathematical Foundations of Computer Science (MFCS 2011)*, pages 256–265. Springer, 2011.

[EGS76] Paul Erdös, Ronald L. Graham, and Endre Szemerédi. On sparse graphs with dense long paths. *Computers and Mathematics with Applications*, pages 365–369, 1976.

[FGHK16] Magnus Find, Alexander Golovnev, Edward A. Hirsch, and Alexander S. Kulikov. A better-than-$3n$ lower bound for the circuit complexity of an explicit function. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2016)*, pages 89–98. IEEE, 2016.

[FLM+18] Fedor V. Fomin, Daniel Lokshtanov, Syed Mohammad Meesum, Saket Saurabh, and Meirav Zehavi. Matrix rigidity from the viewpoint of parameterized complexity. *SIAM Journal on Discrete Mathematics*, 32(2):966–985, 2018.

[For92] Krister Forsman. Two themes in commutative algebra: Algebraic dependence and kähler differentials, 1992.

[Fri93] Joel Friedman. A note on matrix rigidity. *Combinatorica*, 13(2):235–239, 1993.

[GHIL16] Fulvio Gesmundo, Jonathan D Hauenstein, Christian Ikenmeyer, and Joseph M Landsberg. Complexity of linear circuits and geometry. *Foundations of Computational Mathematics*, 16(3):599–635, 2016.

[Gri76] Dmitrii Yu. Grigoriev. Application of separability and independence notions for proving lower bounds of circuit complexity. *Zapiski Nauchnykh Seminarov POMI*, 60:38–48, 1976.

[GRW18] Alexander Golovnev, Oded Regev, and Omri Weinstein. The minrank of random graphs. *IEEE Transactions on Information Theory*, 64(11):6990–6995, 2018.

[GT16] Oded Goldreich and Avishay Tal. Matrix rigidity of random toeplitz matrices. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC 2016)*, pages 91–104. ACM, 2016.

[Hås86] Johan Håstad. Almost optimal lower bounds for small depth circuits. In *STOC 1986*, pages 6–20, 1986.

[HJP93] Johan Håstad, Stasys Jukna, and Pavel Pudlák. Top-down lower bounds for depth 3 circuits. In *FOCS 1993*, pages 124–129, 1993.

[HS65] Juris Hartmanis and Richard E. Stearns. On the computational complexity of algorithms. *Transactions of the American Mathematical Society*, 117:285–306, 1965.

[IM02] Kazuo Iwama and Hiroki Morizumi. An explicit lower bound of $5n-o(n)$ for Boolean circuits. In *Proceedings of the 27th Internationl Symposium on the Mathematical Foundations of Computer Science (MFCS 2002)*, pages 353–364. Springer, 2002.

[IP01] Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001.

[JS13] Stasys Jukna and Igor Sergeev. Complexity of linear boolean operators. *Foundations and Trends in Theoretical Computer Science*, 9(1):1–123, 2013.

[Juk11] Stasys Jukna. *Extremal combinatorics: with applications in computer science*. Springer Science & Business Media, 2011.

[Juk12] Stasys Jukna. *Boolean function complexity: advances and frontiers*, volume 27. Springer Science & Business Media, 2012.

[KLPS14] Abhinav Kumar, Satyanarayana V. Lokam, Vijay M. Patankar, and Jayalal Sarma. Using elimination theory to construct rigid matrices. *Computational Complexity*, 23(4):531–563, 2014.

[KR98] Boris S. Kashin and Alexander A. Razborov. Improved lower bounds on the rigidity of hadamard matrices. *Mathematical Notes*, 63(4):471–475, 1998.

[KST54] Tamás Kővári, Vera T. Sós, and Pál Turán. On a problem of zarankiewicz. In *Colloquium Mathematicum*, volume 3, pages 50–57. Polska Akademia Nauk, 1954.

[KV20] Mrinal Kumar and Ben Lee Volk. Lower bounds for matrix factorization. In *Proceedings of the 35th Annual Computational Complexity Conference (CCC 2020)*, 2020.

[LG88] Jacobus Hendricus van Lint and Gerard van der Geer. *Introduction to coding theory and algebraic geometry*. Birkhäuser Basel, 1988.

[Lok95] Satyanarayana V. Lokam. Spectral methods for matrix rigidity with applications to size-depth tradeoffs and communication complexity. In *Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1995)*, pages 6–15. IEEE, 1995.

[Lok00] Satyanarayana V. Lokam. On the rigidity of vandermonde matrices. *Theoretical Computer Science*, 237(1-2):477–483, 2000.

[Lok06] Satyanarayana V. Lokam. Quadratic lower bounds on matrix rigidity. In *Proceedings of the 3rd International Conference on Theory and Applications of Models of Computation (TAMC)*, pages 295–307. Springer, 2006.

[Lok09] Satyanarayana V. Lokam. Complexity lower bounds using linear algebra. *Foundations and Trends in Theoretical Computer Science*, 4(1–2):1–155, 2009.

[LR01] Oded Lachish and Ran Raz. Explicit lower bound of $4.5n - o(n)$ for Boolean circuits. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC 2001)*, pages 399–408. ACM, 2001.

[LTV03] Joseph M. Landsberg, Jacob Taylor, and Nisheeth Kumar Vishnoi. The geometry of matrix rigidity. Technical report, Georgia Institute of Technology, 2003.

[Lup56] Oleg B. Lupanov. On rectifier and switching-and-rectifier schemes. In *Dokl. Akad. Nauk SSSR*, volume 111, pages 1171–1174, 1956. In Russian.

[Lup59a] Oleg B. Lupanov. A method of circuit synthesis. *Izv. VUZov, Radiofizika*, 1:120–140, 1959. In Russian.

[Lup59b] Oleg B. Lupanov. A method of circuit synthesis. *Izv. VUZov, Radiofizika*, 1:120–140, 1959. In Russian.

[Mid05] Gatis Midrijānis. Three lines proof of the lower bound for the matrix rigidity. *arXiv preprint cs/0506081*, 2005.

[MS77] Florence Jessie MacWilliams and Neil James Alexander Sloane. *The theory of error-correcting codes*. Elsevier, 1977.

[MW17] Or Meir and Avi Wigderson. Prediction from partial information and hindsight, with application to circuit lower bounds. In *ECCC*, volume 24, 2017.

[Nec66] Edward I. Nechiporuk. On a Boolean function. *Dokl. Akad. Nauk SSSR*, 169(4):765–766, 1966.

[Pip77] Nicholas Pippenger. Superconcentrators. *SIAM Journal on Computing*, 6(2):298–304, 1977.

[PPSZ05] Ramamohan Paturi, Pavel Pudlák, Michael E Saks, and Francis Zane. An improved exponential-time algorithm for k-SAT. *J. ACM*, 52(3):337–364, 2005.

[PPZ97] Ramamohan Paturi, Pavel Pudlák, and Francis Zane. Satisfiability coding lemma. In *FOCS 1997*, pages 566–574, 1997.

[PR94] Pavel Pudlák and Vojtech Rödl. Some combinatorial-algebraic problems from complexity theory. *Discrete Mathematics*, 1(136):253–279, 1994.

[PS88] Pavel Pudlák and Petr Savický. Unpublished, 1988. Cited in [Raz89].

[Pud94] Pavel Pudlák. Communication in bounded depth circuits. *Combinatorica*, 14(2):203–216, 1994.

[Raz88] Alexander A. Razborov. Unpublished, 1988. Cited in [Raz89].

[Raz89] Alexander A. Razborov. On rigid matrices. *Manuscript*, 1989. In Russian.

[RBG01] Lajos Rónyai, László Babai, and Murali Ganapathy. On the number of zero-patterns of a sequence of polynomials. *Journal of the American Mathematical Society*, 14(3):717–735, 2001.

[RS03] Ran Raz and Amir Shpilka. Lower bounds for matrix product in bounded depth circuits with arbitrary gates. *SIAM Journal on Computing*, 32(2):488–513, 2003.

[RTS00] Jaikumar Radhakrishnan and Amnon Ta-Shma. Bounds for dispersers, extractors, and depth-two superconcentrators. *SIAM Journal on Discrete Mathematics*, 13(1):2–24, 2000.

[Sha49a] Claude E. Shannon. The synthesis of two-terminal switching circuits. *Bell Syst. Tech. J.*, 28:59–98, 1949.

[Sha49b] Claude E. Shannon. The synthesis of two-terminal switching circuits. *Bell Syst. Tech. J.*, 28:59–98, 1949.

[SS91] Victor Shoup and Roman Smolensky. Lower bounds for polynomial evaluation and interpolation problems. In *Proceedings of the 32nd Annual IEEE Symposium on Foundations of Computer Science (FOCS 1991)*, pages 378–383. IEEE, 1991.

[SSS97] Mohammad A. Shokrollahi, Daniel A. Spielman, and Volker Stemann. A remark on matrix rigidity. *Information Processing Letters*, 64(6):283–285, 1997.

[Val75] Leslie G. Valiant. On non-linear lower bounds in computational complexity. In *Proceedings of the 7th Annual ACM Symposium on Theory of Computing (STOC 1975)*, pages 45–53. ACM, 1975.

[Val77] Leslie G. Valiant. Graph-theoretic arguments in low-level complexity. In *Proceedings of the 2nd Internationl Symposium on the Mathematical Foundations of Computer Science (MFCS 1977)*, pages 121–127. Springer, 1977.

[Vas18] Virginia Vassilevska Williams. On some fine-grained questions in algorithms and complexity. In *Proceedings of the ICM*, pages 3431–3472. World Scientific, 2018.

[Vio09] Emanuele Viola. On the power of small-depth computation. *Foundations and Trends in Theoretical Computer Science*, 5(1):1–72, 2009.

[vL12] Jacobus Hendricus van Lint. *Introduction to coding theory*, volume 86. Springer Science & Business Media, 2012.

[VT91] Serge G. Vlăduţ and Michael A. Tsfasman. Algebraic-Geometric codes. *Mathematics and Applications, Dordrecht et al*, 1991.

[Wil05] Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theoretical Computer Science*, 348(2-3):357–365, 2005.

[Wil14] Ryan Williams. New algorithms and lower bounds for circuits with linear threshold gates. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC 2014)*, pages 194–202, 2014.

[WY14] Ryan Williams and Huacheng Yu. Finding orthogonal vectors in discrete structures. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2014)*, pages 1867–1877. SIAM, 2014.

[Žák83] Stanislav Žák. A turing machine time hierarchy. *Theoretical Computer Science*, 26(3):327–333, 1983.