Algebraic and Analytical Methods in Coding Theory

Probabilistic and polynomial methods: hashing, trifference, and capacities

Simone Costa Stefano Della Fiore

16-hour course

First three lectures (8 hours): Simone Costa's module

The path through the first module

1. **Trifference and the probabilistic method.** An elementary upper bound, first moment, expurgation, and the tetracode.

The path through the first module

1. **Trifference and the probabilistic method.** An elementary upper bound, first moment, expurgation, and the tetracode.
2. **Polynomial method.** The capacity of the cyclic triangle and an attempt on trifference; **BUFFER**: cap sets and slice rank.

The path through the first module

1. **Trifference and the probabilistic method.** An elementary upper bound, first moment, expurgation, and the tetracode.
2. **Polynomial method.** The capacity of the cyclic triangle and an attempt on trifference; **BUFFER**: cap sets and slice rank.
3. **Fredman–Komlós.** Hansel's lemma, coordinate frequencies, and symmetrization.

The path through the first module

1. **Trifference and the probabilistic method.** An elementary upper bound, first moment, expurgation, and the tetracode.
2. **Polynomial method.** The capacity of the cyclic triangle and an attempt on trifference; **BUFFER**: cap sets and slice rank.
3. **Fredman–Komlós.** Hansel's lemma, coordinate frequencies, and symmetrization.

Guiding choice

We will not make a catalogue of techniques. We will follow a few closely related problems and see how the method changes when the question changes.

Codes as matrices: comparing columns

A code $C \subseteq \Sigma^n$ can be displayed with codewords as columns:

$$\begin{pmatrix} | & | & & | \\ x^{(1)} & x^{(2)} & \cdots & x^{(M)} \\ | & | & & | \end{pmatrix}.$$

Codes as matrices: comparing columns

A code $C \subseteq \Sigma^n$ can be displayed with codewords as columns:

$$\begin{pmatrix} | & | & & | \\ x^{(1)} & x^{(2)} & \cdots & x^{(M)} \\ | & | & & | \end{pmatrix}.$$

Each row is a coordinate.

Codes as matrices: comparing columns

A code $C \subseteq \Sigma^n$ can be displayed with codewords as columns:

$$\begin{pmatrix} | & | & & | \\ x^{(1)} & x^{(2)} & \cdots & x^{(M)} \\ | & | & & | \end{pmatrix}.$$

Each row is a coordinate. In the problems we study, we select a few columns and ask that **at least one row** display a pattern that separates them.

Codes as matrices: comparing columns

A code $C \subseteq \Sigma^n$ can be displayed with codewords as columns:

$$\begin{pmatrix} | & | & & | \\ x^{(1)} & x^{(2)} & \cdots & x^{(M)} \\ | & | & & | \end{pmatrix}.$$

Each row is a coordinate. In the problems we study, we select a few columns and ask that **at least one row** display a pattern that separates them.

Example

For trifference we select three columns and want to see the three symbols $0, 1, 2$ in some row.

Codes, size, and rate

A code of length n over an alphabet Σ is simply a subset

$$C \subseteq \Sigma^n.$$

Codes, size, and rate

A code of length n over an alphabet Σ is simply a subset

$$C \subseteq \Sigma^n.$$

If $A(n)$ denotes the largest size of a code with a prescribed property, we are mainly interested in the exponential growth of $A(n)$.

Codes, size, and rate

A code of length n over an alphabet Σ is simply a subset

$$C \subseteq \Sigma^n.$$

If $A(n)$ denotes the largest size of a code with a prescribed property, we are mainly interested in the exponential growth of $A(n)$. We use the **rate in bits**

$$R = \limsup_{n \to \infty} \frac{1}{n} \log_2 A(n).$$

If, at the exponential scale, $A(n) \simeq \alpha^n$, then $R = \log_2 \alpha$.

Trifference

Definition

A code $C \subseteq \{0,1,2\}^n$ is **trifferent** if, for every three distinct codewords $x, y, z \in C$, there is a coordinate i such that

$$\{x_i, y_i, z_i\} = \{0,1,2\}.$$

Trifference

Definition

A code $C \subseteq \{0,1,2\}^n$ is **trifferent** if, for every three distinct codewords $x, y, z \in C$, there is a coordinate i such that

$$\{x_i, y_i, z_i\} = \{0,1,2\}.$$

Let

$$T(n) = \max\{|C| : C \subseteq \{0,1,2\}^n \text{ is trifferent}\}.$$

Trifference

Definition

A code $C \subseteq \{0,1,2\}^n$ is **trifferent** if, for every three distinct codewords $x, y, z \in C$, there is a coordinate i such that

$$\{x_i, y_i, z_i\} = \{0,1,2\}.$$

Let

$$T(n) = \max\{|C| : C \subseteq \{0,1,2\}^n \text{ is trifferent}\}.$$

Our two basic questions are:

how large can $T(n)$ be? how can we construct large codes?

An elementary upper bound

We prove

$$\boxed{T(n) \leq 2\left(\frac{3}{2}\right)^n.}$$

An elementary upper bound

We prove

$$\boxed{T(n) \leq 2\left(\frac{3}{2}\right)^n.}$$

Let $C \subseteq \{0,1,2\}^n$ be trifferent and partition it according to the first coordinate:

$$C = C_0 \sqcup C_1 \sqcup C_2.$$

An elementary upper bound

We prove

$$\boxed{T(n) \leq 2\left(\frac{3}{2}\right)^n.}$$

Let $C \subseteq \{0,1,2\}^n$ be trifferent and partition it according to the first coordinate:

$$C = C_0 \sqcup C_1 \sqcup C_2.$$

Keep the two largest classes. Their union C' satisfies

$$|C'| \geq \frac{2}{3}|C|.$$

The first coordinate now uses at most two symbols, so it can no longer witness trifference.

Pruning and puncturing

Delete the first coordinate from the codewords in C'. The projection is still trifferent: every triple in C' already had to find its witness among the last $n-1$ coordinates. It is also injective when $|C'| \geq 3$: if two codewords became equal, together with a third one they could not be trifferent in any coordinate. The cases with fewer than three codewords are trivial.

Pruning and puncturing

Delete the first coordinate from the codewords in C'. The projection is still trifferent: every triple in C' already had to find its witness among the last $n-1$ coordinates. It is also injective when $|C'| \geq 3$: if two codewords became equal, together with a third one they could not be trifferent in any coordinate. The cases with fewer than three codewords are trivial. Hence

$$T(n-1) \geq |C'| \geq \frac{2}{3}|C|,$$

so

$$T(n) \leq \frac{3}{2}T(n-1).$$

Pruning and puncturing

Delete the first coordinate from the codewords in C'. The projection is still trifferent: every triple in C' already had to find its witness among the last $n-1$ coordinates. It is also injective when $|C'| \geq 3$: if two codewords became equal, together with a third one they could not be trifferent in any coordinate. The cases with fewer than three codewords are trivial. Hence

$$T(n-1) \geq |C'| \geq \frac{2}{3}|C|,$$

so

$$T(n) \leq \frac{3}{2} T(n-1).$$

Since $T(1) = 3$,

$$\boxed{T(n) \leq 2\left(\frac{3}{2}\right)^n.}$$

k-hashing and (b, k)-hashing

Trifference is the first case of a larger family. A code $C \subseteq [b]^n$ is a (b, k)-**hash code** if, for every k distinct codewords, there is a coordinate in which their k symbols are all distinct.

k-hashing and (b, k)-hashing

Trifference is the first case of a larger family. A code $C \subseteq [b]^n$ is a (b, k)-**hash code** if, for every k distinct codewords, there is a coordinate in which their k symbols are all distinct. When $b = k$ this is called k-**hashing**; for $b = k = 3$ we recover trifference.

k-hashing and (b, k)-hashing

Trifference is the first case of a larger family. A code $C \subseteq [b]^n$ is a (b, k)-**hash code** if, for every k distinct codewords, there is a coordinate in which their k symbols are all distinct. When $b = k$ this is called k-**hashing**; for $b = k = 3$ we recover trifference. We write

$$R(b, k) = \limsup_{n \to \infty} \frac{1}{n} \log_2 A(b, k, n)$$

for the optimal asymptotic rate.

Fredman–Komlós (1984).

Other problems in the same spirit

We will not develop them now, but the same viewpoint appears in several nearby problems.

- **Shannon and Sperner capacity**: only *two* codewords are compared at a time; the alphabet is the vertex set of a graph or digraph, and some coordinate must distinguish them.

Other problems in the same spirit

We will not develop them now, but the same viewpoint appears in several nearby problems.

- **Shannon and Sperner capacity**: only *two* codewords are compared at a time; the alphabet is the vertex set of a graph or digraph, and some coordinate must distinguish them.
- **Cap sets in $\mathbb{F}_3^n$**: one forbids distinct triples satisfying $x + y + z = 0$.

Other problems in the same spirit

We will not develop them now, but the same viewpoint appears in several nearby problems.

- **Shannon and Sperner capacity**: only *two* codewords are compared at a time; the alphabet is the vertex set of a graph or digraph, and some coordinate must distinguish them.
- **Cap sets in $\mathbb{F}_3^n$**: one forbids distinct triples satisfying $x + y + z = 0$.
- **Higher-degree EGZ**: the zero-sum condition is replaced by higher-degree polynomial conditions, coordinate by coordinate.

Other problems in the same spirit

We will not develop them now, but the same viewpoint appears in several nearby problems.

- **Shannon and Sperner capacity**: only *two* codewords are compared at a time; the alphabet is the vertex set of a graph or digraph, and some coordinate must distinguish them.
- **Cap sets in $\mathbb{F}_3^n$**: one forbids distinct triples satisfying $x + y + z = 0$.
- **Higher-degree EGZ**: the zero-sum condition is replaced by higher-degree polynomial conditions, coordinate by coordinate.

The cyclic-triangle capacity will return in Lecture 2 because it has an exceptionally short polynomial proof.

Blokhuis (1993); Ellenberg–Gijswijt (2017); Costa–Della Fiore (2024).

A probabilistic lower bound: one fixed triple

Choose M codewords independently and uniformly in $\{0,1,2\}^n$. Fix three distinct indices. In one coordinate the three symbols are all distinct with probability

$$\frac{3!}{3^3} = \frac{2}{9}.$$

A probabilistic lower bound: one fixed triple

Choose M codewords independently and uniformly in $\{0,1,2\}^n$. Fix three distinct indices. In one coordinate the three symbols are all distinct with probability

$$\frac{3!}{3^3} = \frac{2}{9}.$$

Thus that coordinate fails to separate the triple with probability

$$1 - \frac{2}{9} = \frac{7}{9}.$$

A probabilistic lower bound: one fixed triple

Choose M codewords independently and uniformly in $\{0,1,2\}^n$. Fix three distinct indices. In one coordinate the three symbols are all distinct with probability

$$\frac{3!}{3^3} = \frac{2}{9}.$$

Thus that coordinate fails to separate the triple with probability

$$1 - \frac{2}{9} = \frac{7}{9}.$$

The coordinates are independent, hence

$$\boxed{\Pr(\text{the triple is never separated}) = \left(\frac{7}{9}\right)^n.}$$

First moment

Let Z be the number of triples of indices that are not separated. Then

$$\mathbb{E}Z = \binom{M}{3}\left(\frac{7}{9}\right)^n.$$

If $\mathbb{E}Z < 1$, there is a choice with $Z = 0$, hence a trifferent code.

First moment

Let Z be the number of triples of indices that are not separated. Then

$$\mathbb{E}Z = \binom{M}{3}\left(\frac{7}{9}\right)^n.$$

If $\mathbb{E}Z < 1$, there is a choice with $Z = 0$, hence a trifferent code. It is enough to take

$$M = \left(\frac{9}{7}\right)^{n/3-o(n)}.$$

First moment

Let Z be the number of triples of indices that are not separated. Then

$$\mathbb{E}Z = \binom{M}{3}\left(\frac{7}{9}\right)^n.$$

If $\mathbb{E}Z < 1$, there is a choice with $Z = 0$, hence a trifferent code. It is enough to take

$$M = \left(\frac{9}{7}\right)^{n/3-o(n)}.$$

Therefore

$$\boxed{R_3 \geq \frac{1}{3}\log_2\frac{9}{7}.}$$

First moment

Let Z be the number of triples of indices that are not separated. Then

$$\mathbb{E}Z = \binom{M}{3}\left(\frac{7}{9}\right)^n.$$

If $\mathbb{E}Z < 1$, there is a choice with $Z = 0$, hence a trifferent code. It is enough to take

$$M = \left(\frac{9}{7}\right)^{n/3-o(n)}.$$

Therefore

$$\boxed{R_3 \geq \frac{1}{3}\log_2\frac{9}{7}.}$$

Message

The first-moment argument asks the random choice to contain *no* bad configuration at all.

Expurgation: delete afterwards

We can ask for less. Start with L random codewords and, after sampling them, delete one codeword from every bad triple that remains. If Z is the initial number of bad triples, at least

$$S \geq L - Z$$

codewords survive, and no bad triple remains.

Expurgation: delete afterwards

We can ask for less. Start with L random codewords and, after sampling them, delete one codeword from every bad triple that remains. If Z is the initial number of bad triples, at least

$$S \geq L - Z$$

codewords survive, and no bad triple remains. Hence

$$\mathbb{E}S \geq L - \binom{L}{3}\left(\frac{7}{9}\right)^n \geq L - \frac{L^3}{6}\left(\frac{7}{9}\right)^n.$$

Expurgation: choose the right scale

Set

$$L = c\left(\frac{9}{7}\right)^{n/2}.$$

Expurgation: choose the right scale

Set

$$L = c\left(\frac{9}{7}\right)^{n/2}.$$

Then

$$\begin{aligned}\mathbb{E}S &\geq L - \frac{L^3}{6}\left(\frac{7}{9}\right)^n \\ &= \left(c - \frac{c^3}{6}\right)\left(\frac{9}{7}\right)^{n/2} - o\left(\left(\frac{9}{7}\right)^{n/2}\right).\end{aligned}$$

Expurgation: choose the right scale

Set

$$L = c\left(\frac{9}{7}\right)^{n/2}.$$

Then

$$\begin{aligned}\mathbb{E}S &\geq L - \frac{L^3}{6}\left(\frac{7}{9}\right)^n \\ &= \left(c - \frac{c^3}{6}\right)\left(\frac{9}{7}\right)^{n/2} - o\left(\left(\frac{9}{7}\right)^{n/2}\right).\end{aligned}$$

Thus the correct exponential scale is already visible:

$$\left(\frac{9}{7}\right)^{n/2}.$$

Expurgation: choose the right scale

Set

$$L = c\left(\frac{9}{7}\right)^{n/2}.$$

Then

$$\begin{aligned}\mathbb{E}S &\geq L - \frac{L^3}{6}\left(\frac{7}{9}\right)^n \\ &= \left(c - \frac{c^3}{6}\right)\left(\frac{9}{7}\right)^{n/2} - o\left(\left(\frac{9}{7}\right)^{n/2}\right).\end{aligned}$$

Thus the correct exponential scale is already visible:

$$\left(\frac{9}{7}\right)^{n/2}.$$

It remains only to choose c to optimize the constant factor.

Expurgation: optimize the constant

We must maximize

$$f(c) = c - \frac{c^3}{6}.$$

Since

$$f'(c) = 1 - \frac{c^2}{2},$$

the positive maximum occurs at $c = \sqrt{2}$. Moreover,

$$f(\sqrt{2}) = \sqrt{2} - \frac{(\sqrt{2})^3}{6} = \frac{2\sqrt{2}}{3}.$$

Expurgation: optimize the constant

We must maximize

$$f(c) = c - \frac{c^3}{6}.$$

Since

$$f'(c) = 1 - \frac{c^2}{2},$$

the positive maximum occurs at $c = \sqrt{2}$. Moreover,

$$f(\sqrt{2}) = \sqrt{2} - \frac{(\sqrt{2})^3}{6} = \frac{2\sqrt{2}}{3}.$$

Hence

$$\boxed{T(n) \geq \left(\frac{2\sqrt{2}}{3} - o(1)\right)\left(\frac{9}{7}\right)^{n/2}.}$$

In particular,

$$\boxed{R_3 \geq \frac{1}{2}\log_2 \frac{9}{7}.}$$

The same calculation for (b, k)

The preceding proof generalizes without any new idea. For k random codewords over an alphabet of size b, the probability of failing to separate them in one coordinate is

$$q_{b,k} = 1 - \frac{(b)_k}{b^k}, \qquad (b)_k = b(b-1)\cdots(b-k+1).$$

The same calculation for (b, k)

The preceding proof generalizes without any new idea. For k random codewords over an alphabet of size b, the probability of failing to separate them in one coordinate is

$$q_{b,k} = 1 - \frac{(b)_k}{b^k}, \qquad (b)_k = b(b-1)\cdots(b-k+1).$$

At the level of rates one gets

$$\boxed{R(b,k) \geq \frac{1}{k} \log_2 \frac{1}{q_{b,k}} \qquad \text{(first moment)},}$$

$$\boxed{R(b,k) \geq \frac{1}{k-1} \log_2 \frac{1}{q_{b,k}} \qquad \text{(expurgation)}.}$$

We will use only the second bound, for $(b, k) = (9, 3)$.

The tetracode: nine words of length 4

Consider the following ternary code:

0000	0121	0212
1022	1110	1201
2011	2102	2220

The tetracode: nine words of length 4

Consider the following ternary code:

0000	0121	0212
1022	1110	1201
2011	2102	2220

It has 9 codewords of length 4.

The tetracode: nine words of length 4

Consider the following ternary code:

0000	0121	0212
1022	1110	1201
2011	2102	2220

It has 9 codewords of length 4.

Property

It is trifferent: every three distinct codewords have a coordinate containing $0, 1, 2$.

We will not use any algebraic description of the code; one elementary property of the table is enough.

The elementary property of the tetracode

From the table one checks directly that **every pair of codewords agrees in exactly one of the four coordinates**. Equivalently, every pair has Hamming distance 3.

The elementary property of the tetracode

From the table one checks directly that **every pair of codewords agrees in exactly one of the four coordinates**. Equivalently, every pair has Hamming distance 3. Take three distinct codewords x, y, z and, for each coordinate i, let

$$a_i = \#\{\text{pairs among } (x,y),(x,z),(y,z) \text{ that agree at } i\}.$$

Each of the three pairs agrees exactly once, hence

$$\boxed{a_1 + a_2 + a_3 + a_4 = 3.}$$

Why the tetracode is trifferent

If in some coordinate the three symbols are all distinct, we have found the desired witness. Suppose, for a contradiction, that this never happens.

Why the tetracode is trifferent

If in some coordinate the three symbols are all distinct, we have found the desired witness. Suppose, for a contradiction, that this never happens. Then in every coordinate at least two of the three symbols agree, so

$$a_i \geq 1 \qquad (i = 1, 2, 3, 4).$$

This would imply

$$a_1 + a_2 + a_3 + a_4 \geq 4,$$

contradicting $a_1 + \cdots + a_4 = 3$.

Why the tetracode is trifferent

If in some coordinate the three symbols are all distinct, we have found the desired witness. Suppose, for a contradiction, that this never happens. Then in every coordinate at least two of the three symbols agree, so

$$a_i \geq 1 \qquad (i = 1, 2, 3, 4).$$

This would imply

$$a_1 + a_2 + a_3 + a_4 \geq 4,$$

contradicting $a_1 + \cdots + a_4 = 3$. Therefore

the tetracode is trifferent.

The tetracode is optimal at length 4

From the pruning bound and integrality we successively get

$$T(1) = 3, \qquad T(2) \leq 4, \qquad T(3) \leq 6, \qquad T(4) \leq 9.$$

The tetracode is optimal at length 4

From the pruning bound and integrality we successively get

$$T(1) = 3, \qquad T(2) \leq 4, \qquad T(3) \leq 6, \qquad T(4) \leq 9.$$

The tetracode has exactly 9 codewords, hence

$$\boxed{T(4) = 9.}$$

The tetracode is optimal at length 4

From the pruning bound and integrality we successively get

$$T(1) = 3, \qquad T(2) \leq 4, \qquad T(3) \leq 6, \qquad T(4) \leq 9.$$

The tetracode has exactly 9 codewords, hence

$$\boxed{T(4) = 9.}$$

This small optimal construction will be our inner code.

Concatenation: use the tetracode as a new alphabet

Take a code

$$C_{\text{out}} \subseteq [9]^N$$

with the property that every three distinct codewords have a coordinate with three distinct symbols: this is a $(9, 3)$-hash code.

Concatenation: use the tetracode as a new alphabet

Take a code

$$C_{\text{out}} \subseteq [9]^N$$

with the property that every three distinct codewords have a coordinate with three distinct symbols: this is a $(9, 3)$-hash code. Identify the 9 alphabet symbols with the 9 tetracode words and replace every symbol with the corresponding ternary block of length 4:

$$[9]^N \longrightarrow [3]^{4N}.$$

The cardinality is unchanged; the length is multiplied by 4.

Why concatenation preserves trifference

Take three distinct outer codewords. Since C_{out} is $(9, 3)$-hash, there is an outer coordinate j containing three distinct symbols. Those symbols are replaced by three *distinct* tetracode words.

Why concatenation preserves trifference

Take three distinct outer codewords. Since C_{out} is $(9, 3)$-hash, there is an outer coordinate j containing three distinct symbols. Those symbols are replaced by three *distinct* tetracode words. Since the tetracode is trifferent, inside block j there is a coordinate containing $0, 1, 2$.

Why concatenation preserves trifference

Take three distinct outer codewords. Since C_{out} is $(9,3)$-hash, there is an outer coordinate j containing three distinct symbols. Those symbols are replaced by three *distinct* tetracode words. Since the tetracode is trifferent, inside block j there is a coordinate containing $0,1,2$. Thus the concatenated ternary code is trifferent.

$$\boxed{(9,3)\text{-hash on } N \text{ coordinates} \Longrightarrow \text{trifference on } 4N \text{ coordinates.}}$$

Use the probabilistic lower bound once more

For $(b, k) = (9, 3)$,

$$q_{9,3} = 1 - \frac{9 \cdot 8 \cdot 7}{9^3} = \frac{25}{81}.$$

Use the probabilistic lower bound once more

For $(b, k) = (9, 3)$,

$$q_{9,3} = 1 - \frac{9 \cdot 8 \cdot 7}{9^3} = \frac{25}{81}.$$

The general expurgation bound gives

$$R(9, 3) \geq \frac{1}{2} \log_2 \frac{81}{25} = \log_2 \frac{9}{5}.$$

Use the probabilistic lower bound once more

For $(b, k) = (9, 3)$,

$$q_{9,3} = 1 - \frac{9 \cdot 8 \cdot 7}{9^3} = \frac{25}{81}.$$

The general expurgation bound gives

$$R(9, 3) \geq \frac{1}{2} \log_2 \frac{81}{25} = \log_2 \frac{9}{5}.$$

Concatenation divides the rate by 4, because the length is multiplied by 4. Therefore

$$\boxed{R_3 \geq \frac{1}{4} \log_2 \frac{9}{5}.}$$

Use the probabilistic lower bound once more

For $(b, k) = (9, 3)$,

$$q_{9,3} = 1 - \frac{9 \cdot 8 \cdot 7}{9^3} = \frac{25}{81}.$$

The general expurgation bound gives

$$R(9,3) \geq \frac{1}{2} \log_2 \frac{81}{25} = \log_2 \frac{9}{5}.$$

Concatenation divides the rate by 4, because the length is multiplied by 4. Therefore

$$\boxed{R_3 \geq \frac{1}{4} \log_2 \frac{9}{5}.}$$

Equivalently, at the exponential scale,

$$\boxed{T(n) \geq \left(\frac{9}{5}\right)^{n/4 - o(n)}.}$$

Körner–Marton (1988).

Where we stand

For trifference we have seen

method	exponential base
first moment	$(9/7)^{1/3} = 1.0873\ldots$
expurgation	$(9/7)^{1/2} = 1.1338\ldots$
tetracode + concatenation	$(9/5)^{1/4} = 1.1582\ldots$
pruning upper bound	$3/2$

There is still a large gap between constructions and upper bounds.

Where we stand

For trifference we have seen

method	exponential base
first moment	$(9/7)^{1/3} = 1.0873\ldots$
expurgation	$(9/7)^{1/2} = 1.1338\ldots$
tetracode + concatenation	$(9/5)^{1/4} = 1.1582\ldots$
pruning upper bound	$3/2$

There is still a large gap between constructions and upper bounds.

Lecture 2

Can algebra lower the base $3/2$?

Possible further topics

- **Lovász Local Lemma**: separating hash families, covering arrays, and other probabilistic lower bounds with local dependencies.

Possible further topics

- **Lovász Local Lemma**: separating hash families, covering arrays, and other probabilistic lower bounds with local dependencies.
- **Moser–Tardos**: the algorithmic Local Lemma.

Possible further topics

- **Lovász Local Lemma**: separating hash families, covering arrays, and other probabilistic lower bounds with local dependencies.
- **Moser–Tardos**: the algorithmic Local Lemma.
- **Cover-free and superimposed codes**: group-testing problems still described by coordinate patterns.

Possible further topics

- **Lovász Local Lemma**: separating hash families, covering arrays, and other probabilistic lower bounds with local dependencies.
- **Moser–Tardos**: the algorithmic Local Lemma.
- **Cover-free and superimposed codes**: group-testing problems still described by coordinate patterns.
- **Cap sets and higher-degree EGZ**: two natural developments of the polynomial part of the next lecture.

Possible further topics

- **Lovász Local Lemma**: separating hash families, covering arrays, and other probabilistic lower bounds with local dependencies.
- **Moser–Tardos**: the algorithmic Local Lemma.
- **Cover-free and superimposed codes**: group-testing problems still described by coordinate patterns.
- **Cap sets and higher-degree EGZ**: two natural developments of the polynomial part of the next lecture.
- **Combinatorial Nullstellensatz**: another major form of the polynomial method, not needed in our main path.

Essential references — Lecture 1

- M. L. Fredman, J. Komlós, *On the Size of Separating Systems and Families of Perfect Hash Functions*, SIAM J. Algebraic Discrete Methods 5(1) (1984), 61–68.
- J. Körner, K. Marton, *New Bounds for Perfect Hashing via Information Theory*, European J. Combin. 9(6) (1988), 523–530.
- A. Blokhuis, *On the Sperner Capacity of the Cyclic Triangle*, J. Algebraic Combin. 2(2) (1993), 123–124.
- J. Ellenberg, D. Gijswijt, *On Large Subsets of $\mathbb{F}_q^n$ with No Three-Term Arithmetic Progression*, Ann. of Math. 185(1) (2017), 339–343.
- S. Costa, S. Della Fiore, *Bounds on the Higher Degree Erdős–Ginzburg–Ziv Constants over $\mathbb{F}_q^n$*, Arch. Math. 122 (2024), 17–29.

Cyclic-triangle capacity $\longrightarrow$ trifference

Lecture 2 — The polynomial method

$$\text{Cyclic-triangle capacity} \longrightarrow \text{trifference}$$

Main goal (2 hours)

See an *exact* polynomial proof for a capacity problem, and then apply the same philosophy to trifference, where the result is worse than the pruning bound.

Lecture 2 — The polynomial method

$$\text{Cyclic-triangle capacity} \longrightarrow \text{trifference}$$

Main goal (2 hours)

See an *exact* polynomial proof for a capacity problem, and then apply the same philosophy to trifference, where the result is worse than the pruning bound.

Final buffer

Lecture 2 — The polynomial method

$$\text{Cyclic-triangle capacity} \longrightarrow \text{trifference}$$

Main goal (2 hours)

See an *exact* polynomial proof for a capacity problem, and then apply the same philosophy to trifference, where the result is worse than the pruning bound.

Final buffer

Slice rank and cap sets are included only if time remains. The lecture is complete even if we stop before them.

The most elementary form of the polynomial method

The principle we use is purely linear-algebraic. For each object y in a family S, construct a polynomial or function F_y such that

$$F_y(x) = 0 \quad \text{for } x \neq y, \qquad F_y(y) \neq 0.$$

The most elementary form of the polynomial method

The principle we use is purely linear-algebraic. For each object y in a family S, construct a polynomial or function F_y such that

$$F_y(x) = 0 \quad \text{for } x \neq y, \qquad F_y(y) \neq 0.$$

Then the evaluation matrix

$$\big(F_y(x)\big)_{x,y \in S}$$

is diagonal with nonzero diagonal.

The most elementary form of the polynomial method

The principle we use is purely linear-algebraic. For each object y in a family S, construct a polynomial or function F_y such that

$$F_y(x) = 0 \quad \text{for } x \neq y, \qquad F_y(y) \neq 0.$$

Then the evaluation matrix

$$\big(F_y(x)\big)_{x,y \in S}$$

is diagonal with nonzero diagonal. Hence the functions $\{F_y : y \in S\}$ are linearly independent. If they all lie in a vector space V of dimension D, then

$$\boxed{|S| \leq D.}$$

Evaluation lemma: complete proof

Lemma

Let S be finite and let V be a vector space of functions $S \to \mathbb{F}$. If for every $y \in S$ there exists $F_y \in V$ such that

$$F_y(y) \neq 0, \qquad F_y(x) = 0 \quad (x \neq y),$$

then $|S| \leq \dim V$.

Proof.

Evaluation lemma: complete proof

Lemma

Let S be finite and let V be a vector space of functions $S \to \mathbb{F}$. If for every $y \in S$ there exists $F_y \in V$ such that

$$F_y(y) \neq 0, \qquad F_y(x) = 0 \quad (x \neq y),$$

then $|S| \leq \dim V$.

Proof. Suppose

$$\sum_{y \in S} a_y F_y = 0.$$

Evaluation lemma: complete proof

Lemma

Let S be finite and let V be a vector space of functions $S \to \mathbb{F}$. If for every $y \in S$ there exists $F_y \in V$ such that

$$F_y(y) \neq 0, \qquad F_y(x) = 0 \quad (x \neq y),$$

then $|S| \leq \dim V$.

Proof. Suppose

$$\sum_{y \in S} a_y F_y = 0.$$

Evaluate at a fixed $x \in S$:

$$0 = \sum_{y \in S} a_y F_y(x) = a_x F_x(x).$$

Evaluation lemma: complete proof

Lemma

Let S be finite and let V be a vector space of functions $S \to \mathbb{F}$. If for every $y \in S$ there exists $F_y \in V$ such that

$$F_y(y) \neq 0, \qquad F_y(x) = 0 \quad (x \neq y),$$

then $|S| \leq \dim V$.

Proof. Suppose

$$\sum_{y \in S} a_y F_y = 0.$$

Evaluate at a fixed $x \in S$:

$$0 = \sum_{y \in S} a_y F_y(x) = a_x F_x(x).$$

Since $F_x(x) \neq 0$, we get $a_x = 0$. This holds for every x, so the F_y are linearly independent and their number is at most $\dim V$. □

From Shannon to the cyclic-triangle capacity

Shannon capacity studies codes whose alphabet is the vertex set of a graph: we compare **two codewords at a time**, coordinate by coordinate, through a symmetric local relation. **Sperner capacity** replaces the graph by a digraph, so the local relation is directed and ordered pairs matter. For the directed cycle $0 \to 1 \to 2 \to 0$ the condition becomes

$$\forall x \neq y \in S \subseteq \mathbb{F}_3^n \quad \exists j : \quad x_j - y_j = 1 \pmod 3. \tag{$\triangle$}$$

From Shannon to the cyclic-triangle capacity

Shannon capacity studies codes whose alphabet is the vertex set of a graph: we compare **two codewords at a time**, coordinate by coordinate, through a symmetric local relation. **Sperner capacity** replaces the graph by a digraph, so the local relation is directed and ordered pairs matter. For the directed cycle $0 \to 1 \to 2 \to 0$ the condition becomes

$$\forall x \neq y \in S \subseteq \mathbb{F}_3^n \quad \exists j : \quad x_j - y_j = 1 \pmod 3. \tag{$\triangle$}$$

Since the quantifier ranges over every ordered pair, the same property also applies to (y, x).

From Shannon to the cyclic-triangle capacity

Shannon capacity studies codes whose alphabet is the vertex set of a graph: we compare **two codewords at a time**, coordinate by coordinate, through a symmetric local relation. **Sperner capacity** replaces the graph by a digraph, so the local relation is directed and ordered pairs matter. For the directed cycle $0 \to 1 \to 2 \to 0$ the condition becomes

$$\forall x \neq y \in S \subseteq \mathbb{F}_3^n \quad \exists j : \quad x_j - y_j = 1 \pmod 3. \tag{$\triangle$}$$

Since the quantifier ranges over every ordered pair, the same property also applies to (y, x). Blokhuis proves by the polynomial method that

$$\boxed{|S| \leq 2^n.}$$

Gargano–Körner–Vaccaro (1992); Blokhuis (1993).

First: why 2^n is the correct exponential scale

Consider

$$S_r = \{x \in \{0,1\}^n : \mathrm{wt}(x) = r\}, \qquad r = \lfloor n/2 \rfloor.$$

First: why 2^n is the correct exponential scale

Consider

$$S_r = \{x \in \{0,1\}^n : \mathrm{wt}(x) = r\}, \qquad r = \lfloor n/2 \rfloor.$$

Take $x \neq y$ in S_r. Because they have equal Hamming weight, it cannot happen that every 1 of x is also a 1 of y; otherwise $x = y$.

First: why 2^n is the correct exponential scale

Consider

$$S_r = \{x \in \{0,1\}^n : \mathrm{wt}(x) = r\}, \qquad r = \lfloor n/2 \rfloor.$$

Take $x \neq y$ in S_r. Because they have equal Hamming weight, it cannot happen that every 1 of x is also a 1 of y; otherwise $x = y$. Thus some coordinate j satisfies

$$x_j = 1, \qquad y_j = 0,$$

and therefore $x_j - y_j = 1 \pmod 3$.

First: why 2^n is the correct exponential scale

Consider

$$S_r = \{x \in \{0,1\}^n : \mathrm{wt}(x) = r\}, \qquad r = \lfloor n/2 \rfloor.$$

Take $x \neq y$ in S_r. Because they have equal Hamming weight, it cannot happen that every 1 of x is also a 1 of y; otherwise $x = y$. Thus some coordinate j satisfies

$$x_j = 1, \qquad y_j = 0,$$

and therefore $x_j - y_j = 1 \pmod 3$. Hence S_r satisfies ($\triangle$) and

$$|S_r| = \binom{n}{\lfloor n/2 \rfloor} = 2^{n - o(n)}.$$

So if we prove $|S| \leq 2^n$, the capacity is exactly 1 bit per coordinate.

Blokhuis's polynomial

Identify $\{0, 1, 2\}$ with $\mathbb{F}_3$.

Blokhuis's polynomial

Identify $\{0, 1, 2\}$ with $\mathbb{F}_3$. For each $y = (y_1, \ldots, y_n) \in S$, define

$$F_y(X_1, \ldots, X_n) = \prod_{i=1}^{n} (X_i - y_i - 1) \in \mathbb{F}_3[X_1, \ldots, X_n].$$

Each variable has degree at most 1.

Blokhuis's polynomial

Identify $\{0, 1, 2\}$ with $\mathbb{F}_3$. For each $y = (y_1, \ldots, y_n) \in S$, define

$$F_y(X_1, \ldots, X_n) = \prod_{i=1}^{n} (X_i - y_i - 1) \in \mathbb{F}_3[X_1, \ldots, X_n].$$

Each variable has degree at most 1. The factor $X_i - y_i - 1$ is chosen to vanish exactly when

$$X_i - y_i = 1.$$

A. Blokhuis, J. Algebraic Combin. 2(2) (1993), 123–124.

Evaluation on the diagonal

Take $x = y$. Then, in $\mathbb{F}_3$,

$$F_y(y) = \prod_{i=1}^{n} (y_i - y_i - 1) = (-1)^n.$$

Evaluation on the diagonal

Take $x = y$. Then, in $\mathbb{F}_3$,

$$F_y(y) = \prod_{i=1}^{n} (y_i - y_i - 1) = (-1)^n.$$

Therefore

$$\boxed{F_y(y) \neq 0.}$$

No special property of S was used on the diagonal.

Evaluation off the diagonal

Take distinct $x, y \in S$. By property $(\triangle)$ there is a coordinate j such that

$$x_j - y_j = 1 \pmod 3.$$

The product defining $F_y(x)$ therefore contains the zero factor

$$x_j - y_j - 1 = 0.$$

Evaluation off the diagonal

Take distinct $x, y \in S$. By property $(\triangle)$ there is a coordinate j such that

$$x_j - y_j = 1 \pmod 3.$$

The product defining $F_y(x)$ therefore contains the zero factor

$$x_j - y_j - 1 = 0.$$

Hence

$$\boxed{F_y(x) = 0 \qquad (x \neq y).}$$

The matrix $\big(F_y(x)\big)_{x,y \in S}$ is diagonal and invertible.

Where do the polynomials F_y live?

Each F_y has degree at most 1 in each variable. Thus it belongs to

$$V = \operatorname{span}\left\{\prod_{i \in I} X_i : I \subseteq [n]\right\}.$$

The multilinear monomials are indexed by subsets $I \subseteq [n]$, so

$$\dim V = 2^n.$$

Where do the polynomials F_y live?

Each F_y has degree at most 1 in each variable. Thus it belongs to

$$V = \operatorname{span}\left\{\prod_{i \in I} X_i : I \subseteq [n]\right\}.$$

The multilinear monomials are indexed by subsets $I \subseteq [n]$, so

$$\dim V = 2^n.$$

Since the F_y, $y \in S$, are linearly independent,

$$\boxed{|S| \leq 2^n.}$$

Together with the constant-weight construction,

$$\boxed{\Sigma_\triangle = 1.}$$

What made Blokhuis's proof work?

Three transparent ingredients:

1. the combinatorial condition creates a **zero** in at least one factor;

What made Blokhuis's proof work?

Three transparent ingredients:

1. the combinatorial condition creates a **zero** in at least one factor;
2. taking the product converts "there exists a coordinate" into "the product vanishes";

What made Blokhuis's proof work?

Three transparent ingredients:

1. the combinatorial condition creates a **zero** in at least one factor;
2. taking the product converts "there exists a coordinate" into "the product vanishes";
3. separate degree at most 1 gives an ambient space of dimension exactly 2^n.

What made Blokhuis's proof work?

Three transparent ingredients:

1. the combinatorial condition creates a **zero** in at least one factor;
2. taking the product converts "there exists a coordinate" into "the product vanishes";
3. separate degree at most 1 gives an ambient space of dimension exactly 2^n.

Question

Can we do something analogous for a ternary condition on *triples*, namely trifference?

Trifference: move to the cube roots of unity

Let

$$\omega = e^{2\pi i/3}, \qquad 1 + \omega + \omega^2 = 0.$$

Trifference: move to the cube roots of unity

Let

$$\omega = e^{2\pi i/3}, \qquad 1 + \omega + \omega^2 = 0.$$

Identify the three symbols with

$$\{1, \omega, \omega^2\}.$$

For $a, b, c \in \{1, \omega, \omega^2\}$,

$$a, b, c \text{ are pairwise distinct} \quad \Longleftrightarrow \quad a + b + c = 0.$$

Why? If they are distinct, they are exactly $1, \omega, \omega^2$. Conversely, if two were equal, say $a = b$, then $c = -2a$, which has modulus 2 and therefore cannot be a cube root of unity.

Index by pairs rather than by codewords

Let $A \subseteq \{1, \omega, \omega^2\}^n$ be trifferent and set $M = |A|$. Define

$$A^{(2)} = \big\{\{x, y\} : x, y \in A,\ x \neq y\big\},$$

so that

$$|A^{(2)}| = \binom{M}{2}.$$

Index by pairs rather than by codewords

Let $A \subseteq \{1, \omega, \omega^2\}^n$ be trifferent and set $M = |A|$. Define

$$A^{(2)} = \big\{\{x, y\} : x, y \in A,\ x \neq y\big\},$$

so that

$$|A^{(2)}| = \binom{M}{2}.$$

The idea is to construct, for every pair $\{x, y\}$, a function evaluated on *other pairs* $\{z, t\}$. This turns a condition on triples into a linear-independence problem for a matrix indexed by pairs.

Costa–Dalai, J. Combin. Theory Ser. A 177 (2021), 105335, §3.

The polynomial function associated with a pair

For $\{x, y\} \in A^{(2)}$, define

$$f_{x,y}(z,t) = \prod_{i=1}^{n}(x_i + y_i + z_i)(x_i + y_i + t_i), \qquad \{z,t\} \in A^{(2)}.$$

The polynomial function associated with a pair

For $\{x, y\} \in A^{(2)}$, define

$$f_{x,y}(z,t) = \prod_{i=1}^{n} (x_i + y_i + z_i)(x_i + y_i + t_i), \qquad \{z,t\} \in A^{(2)}.$$

The function is symmetric in x, y and in z, t, so it is well defined on unordered pairs.

The polynomial function associated with a pair

For $\{x, y\} \in A^{(2)}$, define

$$f_{x,y}(z,t) = \prod_{i=1}^{n} (x_i + y_i + z_i)(x_i + y_i + t_i), \qquad \{z,t\} \in A^{(2)}.$$

The function is symmetric in x, y and in z, t, so it is well defined on unordered pairs. We shall prove

$$f_{x,y}(z,t) = 0 \quad \text{if } \{x,y\} \neq \{z,t\},$$

whereas

$$f_{x,y}(x,y) \neq 0.$$

The polynomial function associated with a pair

For $\{x, y\} \in A^{(2)}$, define

$$f_{x,y}(z,t) = \prod_{i=1}^{n} (x_i + y_i + z_i)(x_i + y_i + t_i), \qquad \{z,t\} \in A^{(2)}.$$

The function is symmetric in x, y and in z, t, so it is well defined on unordered pairs. We shall prove

$$f_{x,y}(z,t) = 0 \quad \text{if } \{x,y\} \neq \{z,t\},$$

whereas

$$f_{x,y}(x,y) \neq 0.$$

Once again the evaluation matrix will be diagonal.

Polynomial trifference argument: the diagonal

On the diagonal $\{z, t\} = \{x, y\}$, order the pair as $z = x$, $t = y$. Then

$$f_{x,y}(x, y) = \prod_{i=1}^{n} (2x_i + y_i)(x_i + 2y_i).$$

Every factor is nonzero. For instance,

$$2x_i + y_i = 0 \quad \Longrightarrow \quad y_i = -2x_i,$$

but

$$|y_i| = 1 \qquad \text{and} \qquad |-2x_i| = 2,$$

a contradiction. The same argument applies to $x_i + 2y_i$.

Polynomial trifference argument: the diagonal

On the diagonal $\{z, t\} = \{x, y\}$, order the pair as $z = x$, $t = y$. Then

$$f_{x,y}(x, y) = \prod_{i=1}^{n} (2x_i + y_i)(x_i + 2y_i).$$

Every factor is nonzero. For instance,

$$2x_i + y_i = 0 \quad \Longrightarrow \quad y_i = -2x_i,$$

but

$$|y_i| = 1 \qquad \text{and} \qquad |-2x_i| = 2,$$

a contradiction. The same argument applies to $x_i + 2y_i$. Hence

$$\boxed{f_{x,y}(x, y) \neq 0.}$$

Polynomial trifference argument: off the diagonal

Suppose

$$\{x, y\} \neq \{z, t\}.$$

Polynomial trifference argument: off the diagonal

Suppose

$$\{x, y\} \neq \{z, t\}.$$

At least one of z, t does not belong to $\{x, y\}$. Without loss of generality, assume $z \notin \{x, y\}$.

Polynomial trifference argument: off the diagonal

Suppose

$$\{x, y\} \neq \{z, t\}.$$

At least one of z, t does not belong to $\{x, y\}$. Without loss of generality, assume $z \notin \{x, y\}$. Then x, y, z are three distinct codewords of A. By trifference, there exists a coordinate i in which x_i, y_i, z_i are the three distinct symbols, so

$$x_i + y_i + z_i = 1 + \omega + \omega^2 = 0.$$

The product defining $f_{x,y}(z, t)$ contains a zero factor, and therefore

$$\boxed{f_{x,y}(z, t) = 0.}$$

Consequence: $\binom{M}{2}$ independent functions

Suppose

$$\sum_{\{x,y\}\in A^{(2)}} a_{x,y} f_{x,y} = 0.$$

Consequence: $\binom{M}{2}$ independent functions

Suppose

$$\sum_{\{x,y\}\in A^{(2)}} a_{x,y} f_{x,y} = 0.$$

Evaluate at a fixed pair $\{z, t\}$. All off-diagonal terms vanish, leaving

$$a_{z,t} f_{z,t}(z, t) = 0.$$

Consequence: $\binom{M}{2}$ independent functions

Suppose

$$\sum_{\{x,y\}\in A^{(2)}} a_{x,y} f_{x,y} = 0.$$

Evaluate at a fixed pair $\{z, t\}$. All off-diagonal terms vanish, leaving

$$a_{z,t} f_{z,t}(z, t) = 0.$$

Since $f_{z,t}(z, t) \neq 0$,

$$a_{z,t} = 0.$$

This holds for every pair. Thus

$$\{f_{x,y} : \{x, y\} \in A^{(2)}\}$$

is a linearly independent family of cardinality

$$\binom{M}{2}.$$

Consequence: $\binom{M}{2}$ independent functions

Suppose

$$\sum_{\{x,y\}\in A^{(2)}} a_{x,y} f_{x,y} = 0.$$

Evaluate at a fixed pair $\{z, t\}$. All off-diagonal terms vanish, leaving

$$a_{z,t} f_{z,t}(z, t) = 0.$$

Since $f_{z,t}(z, t) \neq 0$,

$$a_{z,t} = 0.$$

This holds for every pair. Thus

$$\{f_{x,y} : \{x, y\} \in A^{(2)}\}$$

is a linearly independent family of cardinality

$$\binom{M}{2}.$$

It remains to place all these functions in a small ambient space.

The structure in one coordinate

For a fixed coordinate,

$$\begin{aligned}(x_i + y_i + z_i)(x_i + y_i + t_i) \\ = (x_i + y_i)^2 + (x_i + y_i)(z_i + t_i) + z_i t_i.\end{aligned}$$

The structure in one coordinate

For a fixed coordinate,

$$\begin{aligned}&(x_i + y_i + z_i)(x_i + y_i + t_i)\\ &\quad = (x_i + y_i)^2 + (x_i + y_i)(z_i + t_i) + z_i t_i.\end{aligned}$$

For fixed x, y, as a function of (z, t) this coordinate belongs to the span of three functions:

$$1, \qquad z_i + t_i, \qquad z_i t_i.$$

The structure in one coordinate

For a fixed coordinate,

$$\begin{aligned}&(x_i + y_i + z_i)(x_i + y_i + t_i)\\ &\quad = (x_i + y_i)^2 + (x_i + y_i)(z_i + t_i) + z_i t_i.\end{aligned}$$

For fixed x, y, as a function of (z, t) this coordinate belongs to the span of three functions:

$$1, \qquad z_i + t_i, \qquad z_i t_i.$$

The coefficients may depend on x_i, y_i; for dimension counting only the space generated by the functions of z, t matters.

Global dimension: three choices per coordinate

Multiplying over the n coordinates, every $f_{x,y}$ belongs to the span of the functions

$$\prod_{i \in I}(z_i + t_i) \prod_{j \in J} z_j t_j, \qquad I, J \subseteq [n], \quad I \cap J = \varnothing.$$

Global dimension: three choices per coordinate

Multiplying over the n coordinates, every $f_{x,y}$ belongs to the span of the functions

$$\prod_{i \in I}(z_i + t_i) \prod_{j \in J} z_j t_j, \qquad I, J \subseteq [n], \quad I \cap J = \varnothing.$$

For each coordinate there are exactly three choices:

$$1, \qquad z_i + t_i, \qquad z_i t_i.$$

Global dimension: three choices per coordinate

Multiplying over the n coordinates, every $f_{x,y}$ belongs to the span of the functions

$$\prod_{i \in I} (z_i + t_i) \prod_{j \in J} z_j t_j, \qquad I, J \subseteq [n], \quad I \cap J = \varnothing.$$

For each coordinate there are exactly three choices:

$$1, \qquad z_i + t_i, \qquad z_i t_i.$$

Hence the number of possible products is

$$3^n.$$

The generated space has dimension *at most* 3^n; there may be linear dependencies, but we do not need to identify them.

The polynomial upper bound for trifference

We found $\binom{M}{2}$ linearly independent functions in a space of dimension at most 3^n. Thus

$$\binom{M}{2} \leq 3^n.$$

Equivalently,

$$M(M-1) \leq 2 \cdot 3^n,$$

so

$$M \leq \frac{1+\sqrt{1+8\cdot 3^n}}{2}.$$

At exponential scale,

$$\boxed{T(n) \leq (\sqrt{3}+o(1))^n \approx (1.7321)^n.}$$

The polynomial upper bound for trifference

We found $\binom{M}{2}$ linearly independent functions in a space of dimension at most 3^n. Thus

$$\binom{M}{2} \leq 3^n.$$

Equivalently,

$$M(M-1) \leq 2 \cdot 3^n,$$

so

$$M \leq \frac{1+\sqrt{1+8\cdot 3^n}}{2}.$$

At exponential scale,

$$\boxed{T(n) \leq (\sqrt{3}+o(1))^n \approx (1.7321)^n.}$$

Outcome

This is a genuine polynomial-method upper bound, but it is *worse* than the pruning bound $2(3/2)^n$.

The pedagogical point of the failure

Blokhuis:

$$\text{polynomial} \longrightarrow \text{dimension } 2^n \longrightarrow \text{sharp bound.}$$

The pedagogical point of the failure

Blokhuis:

$$\text{polynomial} \longrightarrow \text{dimension } 2^n \longrightarrow \text{sharp bound.}$$

Trifference:

$$\text{polynomial on pairs} \longrightarrow \text{dimension } 3^n \longrightarrow M \lesssim 3^{n/2}.$$

The same philosophy produces a diagonal matrix, but the ambient space is too large.

The pedagogical point of the failure

Blokhuis:

$$\text{polynomial} \longrightarrow \text{dimension } 2^n \longrightarrow \text{sharp bound.}$$

Trifference:

$$\text{polynomial on pairs} \longrightarrow \text{dimension } 3^n \longrightarrow M \lesssim 3^{n/2}.$$

The same philosophy produces a diagonal matrix, but the ambient space is too large.

Message

The polynomial method is not an automatic machine for improving exponential bounds: its strength depends on the encoding and on the dimension or rank of the resulting algebraic object.

End of the required part of Lecture 2

At this point we already have a complete narrative:

cyclic triangle: exact success $\longrightarrow$ trifference: $\sqrt{3}^n$ bound, but insufficient.

If time is short, stopping here is completely natural.

End of the required part of Lecture 2

At this point we already have a complete narrative:

cyclic triangle: exact success $\longrightarrow$ trifference: $\sqrt{3}^n$ bound, but insufficient.

If time is short, stopping here is completely natural.

BUFFER

The following slides introduce slice rank and cap sets, and explain the obstruction to a direct "cap-set style" attack on trifference.

BUFFER: from matrix rank to slice rank

For a function $T : X \times Y \times Z \to \mathbb{F}$, a **slice** is a term that separates one variable from the other two, for example

$$f(x)g(y,z),$$

or $f(y)g(x,z)$ or $f(z)g(x,y)$.

BUFFER: from matrix rank to slice rank

For a function $T : X \times Y \times Z \to \mathbb{F}$, a **slice** is a term that separates one variable from the other two, for example

$$f(x)g(y,z),$$

or $f(y)g(x,z)$ or $f(z)g(x,y)$.

Definition

The **slice rank** $\operatorname{srank}(T)$ is the minimum number of slices whose sum is T.

For two variables this is ordinary matrix rank.

BUFFER: from matrix rank to slice rank

For a function $T : X \times Y \times Z \to \mathbb{F}$, a **slice** is a term that separates one variable from the other two, for example

$$f(x)g(y,z),$$

or $f(y)g(x,z)$ or $f(z)g(x,y)$.

Definition

The **slice rank** $\operatorname{srank}(T)$ is the minimum number of slices whose sum is T.

For two variables this is ordinary matrix rank. For three variables it treats tensors in a symmetric way.

Tao (2016); Costa–Dalai (2021).

BUFFER: Tao's diagonal lemma

Lemma

Let A be finite and let $T : A^3 \to \mathbb{F}$ satisfy

$$T(x, y, z) = 0 \quad \text{unless } x = y = z,$$

and

$$T(a, a, a) \neq 0 \quad \forall a \in A.$$

BUFFER: Tao's diagonal lemma

Lemma

Let A be finite and let $T : A^3 \to \mathbb{F}$ satisfy

$$T(x, y, z) = 0 \quad \text{unless } x = y = z,$$

and

$$T(a, a, a) \neq 0 \quad \forall a \in A.$$

Then

$$\boxed{\operatorname{srank}(T) = |A|.}$$

The upper bound $\operatorname{srank}(T) \leq |A|$ is immediate:

$$T(x, y, z) = \sum_{a \in A} T(a, a, a)\, \delta_a(x)\delta_a(y)\delta_a(z).$$

For the lower bound we use two short linear-algebra steps.

BUFFER: diagonal lemma — first step of the lower bound

Suppose we have a decomposition into $r + s + t$ slices:

$$T = \sum_{i=1}^{r} a_i(x) b_i(y, z) + \sum_{j=1}^{s} c_j(y) d_j(x, z) + \sum_{\ell=1}^{t} e_\ell(z) f_\ell(x, y).$$

BUFFER: diagonal lemma — first step of the lower bound

Suppose we have a decomposition into $r + s + t$ slices:

$$T = \sum_{i=1}^{r} a_i(x) b_i(y, z) + \sum_{j=1}^{s} c_j(y) d_j(x, z) + \sum_{\ell=1}^{t} e_\ell(z) f_\ell(x, y).$$

Let $W \subseteq \mathbb{F}^A$ be the orthogonal complement of the span of $a_1, \ldots, a_r$.

BUFFER: diagonal lemma — first step of the lower bound

Suppose we have a decomposition into $r + s + t$ slices:

$$T = \sum_{i=1}^{r} a_i(x) b_i(y, z) + \sum_{j=1}^{s} c_j(y) d_j(x, z) + \sum_{\ell=1}^{t} e_\ell(z) f_\ell(x, y).$$

Let $W \subseteq \mathbb{F}^A$ be the orthogonal complement of the span of $a_1, \ldots, a_r$. Then

$$\dim W \geq |A| - r.$$

Linear fact: a d-dimensional subspace of $\mathbb{F}^A$ contains a vector with at least d nonzero coordinates. Indeed, choose d coordinates on which the projection of W has rank d; after row reduction, choose a linear combination whose pivot coordinates are all nonzero.

BUFFER: diagonal lemma — first step of the lower bound

Suppose we have a decomposition into $r + s + t$ slices:

$$T = \sum_{i=1}^{r} a_i(x) b_i(y, z) + \sum_{j=1}^{s} c_j(y) d_j(x, z) + \sum_{\ell=1}^{t} e_\ell(z) f_\ell(x, y).$$

Let $W \subseteq \mathbb{F}^A$ be the orthogonal complement of the span of $a_1, \ldots, a_r$. Then

$$\dim W \geq |A| - r.$$

Linear fact: a d-dimensional subspace of $\mathbb{F}^A$ contains a vector with at least d nonzero coordinates. Indeed, choose d coordinates on which the projection of W has rank d; after row reduction, choose a linear combination whose pivot coordinates are all nonzero. Thus choose $h \in W$ that is nonzero at at least $|A| - r$ points.

BUFFER: diagonal lemma — contract to a matrix

Contract the x variable against h:

$$M(y,z) = \sum_{x \in A} h(x)\, T(x,y,z).$$

BUFFER: diagonal lemma — contract to a matrix

Contract the x variable against h:

$$M(y,z) = \sum_{x \in A} h(x)\, T(x,y,z).$$

Since T is diagonal,

$$M(y,z) = 0 \ (y \neq z), \qquad M(y,y) = h(y)\, T(y,y,y).$$

BUFFER: diagonal lemma — contract to a matrix

Contract the x variable against h:

$$M(y,z) = \sum_{x \in A} h(x)\, T(x,y,z).$$

Since T is diagonal,

$$M(y,z) = 0 \ (y \neq z), \qquad M(y,y) = h(y)\, T(y,y,y).$$

Thus M is diagonal with at least $|A| - r$ nonzero diagonal entries:

$$\operatorname{rank} M \geq |A| - r.$$

BUFFER: diagonal lemma — contract to a matrix

Contract the x variable against h:

$$M(y,z) = \sum_{x \in A} h(x)\, T(x,y,z).$$

Since T is diagonal,

$$M(y,z) = 0 \ (y \neq z), \qquad M(y,y) = h(y)\, T(y,y,y).$$

Thus M is diagonal with at least $|A| - r$ nonzero diagonal entries:

$$\operatorname{rank} M \geq |A| - r.$$

On the other hand, the first r groups vanish because $h \perp a_i$; the remaining $s + t$ terms become rank-one matrices. Therefore

$$\operatorname{rank} M \leq s + t.$$

BUFFER: diagonal lemma — contract to a matrix

Contract the x variable against h:

$$M(y,z) = \sum_{x \in A} h(x)\, T(x,y,z).$$

Since T is diagonal,

$$M(y,z) = 0 \ (y \neq z), \qquad M(y,y) = h(y)\, T(y,y,y).$$

Thus M is diagonal with at least $|A| - r$ nonzero diagonal entries:

$$\operatorname{rank} M \geq |A| - r.$$

On the other hand, the first r groups vanish because $h \perp a_i$; the remaining $s + t$ terms become rank-one matrices. Therefore

$$\operatorname{rank} M \leq s + t.$$

Hence

$$|A| - r \leq s + t,$$

or

$$|A| \leq r + s + t.$$

Minimizing over all decompositions gives $\operatorname{srank}(T) \geq |A|$. □

BUFFER: cap sets and an indicator tensor

Let $A \subseteq \mathbb{F}_3^n$ be a cap set. Define

$$T(x,y,z) = \prod_{i=1}^{n} \left(1 - (x_i + y_i + z_i)^2\right), \qquad x,y,z \in A.$$

In $\mathbb{F}_3$,

$$1 - u^2 = \begin{cases} 1, & u = 0, \\ 0, & u \neq 0. \end{cases}$$

BUFFER: cap sets and an indicator tensor

Let $A \subseteq \mathbb{F}_3^n$ be a cap set. Define

$$T(x, y, z) = \prod_{i=1}^{n} \left(1 - (x_i + y_i + z_i)^2\right), \qquad x, y, z \in A.$$

In $\mathbb{F}_3$,

$$1 - u^2 = \begin{cases} 1, & u = 0, \\ 0, & u \neq 0. \end{cases}$$

Hence

$$T(x, y, z) = \mathbf{1}[x + y + z = 0].$$

BUFFER: cap sets and an indicator tensor

Let $A \subseteq \mathbb{F}_3^n$ be a cap set. Define

$$T(x,y,z) = \prod_{i=1}^{n} \left(1 - (x_i + y_i + z_i)^2\right), \qquad x,y,z \in A.$$

In $\mathbb{F}_3$,

$$1 - u^2 = \begin{cases} 1, & u = 0, \\ 0, & u \neq 0. \end{cases}$$

Hence

$$T(x,y,z) = \mathbf{1}[x + y + z = 0].$$

Since A is a cap set, on A^3 the tensor is diagonal with diagonal equal to 1. By Tao's lemma,

$$\boxed{|A| = \mathrm{srank}(T).}$$

BUFFER: cap sets and an indicator tensor

Let $A \subseteq \mathbb{F}_3^n$ be a cap set. Define

$$T(x,y,z) = \prod_{i=1}^{n} \left(1 - (x_i + y_i + z_i)^2\right), \qquad x,y,z \in A.$$

In $\mathbb{F}_3$,

$$1 - u^2 = \begin{cases} 1, & u = 0, \\ 0, & u \neq 0. \end{cases}$$

Hence

$$T(x,y,z) = \mathbf{1}[x + y + z = 0].$$

Since A is a cap set, on A^3 the tensor is diagonal with diagonal equal to 1. By Tao's lemma,

$$\boxed{|A| = \mathrm{srank}(T).}$$

It remains to upper-bound the slice rank through the polynomial expansion.

BUFFER: why low-degree monomials appear

The polynomial

$$\prod_{i=1}^{n} \left(1 - (x_i + y_i + z_i)^2\right)$$

has total degree at most $2n$, and after reduction over $\mathbb{F}_3$ each individual exponent is at most 2. In every monomial, let d_x, d_y, d_z be the total degrees in the three variable blocks. Then

$$d_x + d_y + d_z \leq 2n.$$

By the pigeonhole principle, at least one satisfies

$$d_x \leq \frac{2n}{3}, \quad \text{or } d_y \leq \frac{2n}{3}, \quad \text{or } d_z \leq \frac{2n}{3}.$$

BUFFER: why low-degree monomials appear

The polynomial

$$\prod_{i=1}^{n} \left(1 - (x_i + y_i + z_i)^2\right)$$

has total degree at most $2n$, and after reduction over $\mathbb{F}_3$ each individual exponent is at most 2. In every monomial, let d_x, d_y, d_z be the total degrees in the three variable blocks. Then

$$d_x + d_y + d_z \leq 2n.$$

By the pigeonhole principle, at least one satisfies

$$d_x \leq \frac{2n}{3}, \quad \text{or } d_y \leq \frac{2n}{3}, \quad \text{or } d_z \leq \frac{2n}{3}.$$

Grouping monomials according to a low-degree block yields a decomposition into slices.

BUFFER: counting the slices

Set

$$m(n) = \# \left\{ \alpha \in \{0,1,2\}^n : \sum_{i=1}^{n} \alpha_i \leq \frac{2n}{3} \right\}.$$

Monomials for which the x block has degree at most $2n/3$ can be grouped into at most $m(n)$ slices of the form

$$x^\alpha \, G_\alpha(y,z).$$

The same applies to y and z.

BUFFER: counting the slices

Set

$$m(n) = \#\left\{\alpha \in \{0,1,2\}^n : \sum_{i=1}^{n} \alpha_i \leq \frac{2n}{3}\right\}.$$

Monomials for which the x block has degree at most $2n/3$ can be grouped into at most $m(n)$ slices of the form

$$x^{\alpha}\, G_{\alpha}(y,z).$$

The same applies to y and z. Thus

$$\operatorname{srank}(T) \leq 3m(n)$$

and therefore

$$|A| \leq 3m(n).$$

What remains is a generating-function estimate.

BUFFER: generating function and the constant 2.7551 . . .

For every $0 < x < 1$,

$$m(n)x^{2n/3} \leq \sum_{\alpha \in \{0,1,2\}^n} x^{\sum_i \alpha_i}$$
$$= (1 + x + x^2)^n.$$

BUFFER: generating function and the constant 2.7551...

For every $0 < x < 1$,

$$m(n)x^{2n/3} \le \sum_{\alpha \in \{0,1,2\}^n} x^{\sum_i \alpha_i} = (1 + x + x^2)^n.$$

Hence

$$m(n) \le \left(\frac{1 + x + x^2}{x^{2/3}} \right)^n.$$

Minimizing over $0 < x < 1$ gives

$$\theta_3 := \min_{0<x<1} \frac{1 + x + x^2}{x^{2/3}} = 2.7551046\ldots$$

with the minimum at $x \approx 0.59307$.

BUFFER: generating function and the constant 2.7551...

For every $0 < x < 1$,

$$m(n)x^{2n/3} \le \sum_{\alpha \in \{0,1,2\}^n} x^{\sum_i \alpha_i} = (1 + x + x^2)^n.$$

Hence

$$m(n) \le \left(\frac{1 + x + x^2}{x^{2/3}} \right)^n.$$

Minimizing over $0 < x < 1$ gives

$$\theta_3 := \min_{0<x<1} \frac{1 + x + x^2}{x^{2/3}} = 2.7551046\ldots$$

with the minimum at $x \approx 0.59307$. Therefore

$$\boxed{|A| \le 3\,\theta_3^n.}$$

The factor 3 is subexponential; the exponential base is $2.7551\ldots < 3$.

BUFFER: why cap-set style cannot directly break $3/2$ for trifference

Costa–Dalai prove a universal gap for asymptotic slice rank:

$$\widetilde{\mathrm{srank}}(T) = 1 \quad \text{or} \quad \widetilde{\mathrm{srank}}(T) \geq \frac{k}{(k-1)^{(k-1)/k}}$$

for every nontrivial k-tensor. For $k = 3$,

$$\frac{3}{2^{2/3}} = 1.88988\ldots > 1.5.$$

BUFFER: why cap-set style cannot directly break 3/2 for trifference

Costa–Dalai prove a universal gap for asymptotic slice rank:

$$\widetilde{\operatorname{srank}}(T) = 1 \quad \text{or} \quad \widetilde{\operatorname{srank}}(T) \geq \frac{k}{(k-1)^{(k-1)/k}}$$

for every nontrivial k-tensor. For $k = 3$,

$$\frac{3}{2^{2/3}} = 1.88988\ldots > 1.5.$$

Therefore a *straightforward* cap-set-style application based on tensor powers of one local 3-tensor cannot yield an exponential base below 1.5 for trifference. This does not rule out more elaborate algebraic arguments: the $\sqrt{3}^n$ bound above can itself be viewed through matrices, i.e. 2-tensors.

Costa–Dalai, J. Combin. Theory Ser. A 177 (2021), 105335.

Lecture 2: take-home messages

1. The polynomial method can be viewed as **diagonalization + dimension counting**.

Lecture 2: take-home messages

1. The polynomial method can be viewed as **diagonalization + dimension counting**.
2. For the cyclic triangle, this gives the exact 2^n bound using an extremely simple multilinear polynomial.

Lecture 2: take-home messages

1. The polynomial method can be viewed as **diagonalization + dimension counting**.
2. For the cyclic triangle, this gives the exact 2^n bound using an extremely simple multilinear polynomial.
3. For trifference, a natural pair-based encoding gives

$$T(n) \leq (\sqrt{3} + o(1))^n,$$

but loses to pruning.

Lecture 2: take-home messages

1. The polynomial method can be viewed as **diagonalization + dimension counting**.
2. For the cyclic triangle, this gives the exact 2^n bound using an extremely simple multilinear polynomial.
3. For trifference, a natural pair-based encoding gives

$$T(n) \leq (\sqrt{3} + o(1))^n,$$

   but loses to pruning.
4. **BUFFER**: cap sets illustrate the modern slice-rank version, while the gap theorem gives a structural obstruction to the direct slice-rank approach for trifference.

Possible further topics — polynomial method

- **Combinatorial Nullstellensatz**: for example Cauchy–Davenport and Erdős–Heilbronn.

Possible further topics — polynomial method

- **Combinatorial Nullstellensatz**: for example Cauchy–Davenport and Erdős–Heilbronn.
- **The full cap-set theorem**: Ellenberg–Gijswijt and Tao's slice-rank formulation.

Possible further topics — polynomial method

- **Combinatorial Nullstellensatz**: for example Cauchy–Davenport and Erdős–Heilbronn.
- **The full cap-set theorem**: Ellenberg–Gijswijt and Tao's slice-rank formulation.
- **Higher-degree EGZ**: polynomial/slice-rank methods for higher-degree polynomial constraints.

Possible further topics — polynomial method

- **Combinatorial Nullstellensatz**: for example Cauchy–Davenport and Erdős–Heilbronn.
- **The full cap-set theorem**: Ellenberg–Gijswijt and Tao's slice-rank formulation.
- **Higher-degree EGZ**: polynomial/slice-rank methods for higher-degree polynomial constraints.
- **Limitations of the method**: when the dimension or slice rank of the local encoding prevents the desired exponential bound.

Essential references — Lecture 2

- L. Gargano, J. Körner, U. Vaccaro, *Qualitative Independence and Sperner Problems for Directed Graphs*, J. Combin. Theory Ser. A 61 (1992), 173–192.
- A. Blokhuis, *On the Sperner Capacity of the Cyclic Triangle*, J. Algebraic Combin. 2(2) (1993), 123–124.
- J. Ellenberg, D. Gijswijt, *On Large Subsets of $\mathbb{F}_q^n$ with No Three-Term Arithmetic Progression*, Ann. of Math. 185(1) (2017), 339–343.
- T. Tao, *A symmetric formulation of the Croot–Lev–Pach–Ellenberg–Gijswijt capset bound*, blog post, 18 May 2016.
- S. Costa, M. Dalai, *A Gap in the Slice Rank of k-Tensors*, J. Combin. Theory Ser. A 177 (2021), 105335.
- S. Costa, S. Della Fiore, *Bounds on the Higher Degree Erdős–Ginzburg–Ziv Constants over $\mathbb{F}_q^n$*, Arch. Math. 122 (2024), 17–29.

Lecture 3 — The Fredman–Komlós bound

For $k \geq 4$ we want to prove

$$\boxed{R_k \leq \frac{k!}{k^{k-1}}.}$$

Lecture 3 — The Fredman–Komlós bound

For $k \geq 4$ we want to prove

$$\boxed{R_k \leq \frac{k!}{k^{k-1}}.}$$

The proof splits into three transparent steps:

1. fix $k-2$ codewords and reduce the problem to a **cover of a complete graph by bipartite graphs**;

Lecture 3 — The Fredman–Komlós bound

For $k \geq 4$ we want to prove

$$\boxed{R_k \leq \frac{k!}{k^{k-1}}.}$$

The proof splits into three transparent steps:

1. fix $k-2$ codewords and reduce the problem to a **cover of a complete graph by bipartite graphs**;
2. apply **Hansel's lemma**;

Lecture 3 — The Fredman–Komlós bound

For $k \geq 4$ we want to prove

$$\boxed{R_k \leq \frac{k!}{k^{k-1}}.}$$

The proof splits into three transparent steps:

1. fix $k-2$ codewords and reduce the problem to a **cover of a complete graph by bipartite graphs**;
2. apply **Hansel's lemma**;
3. average over the $k-2$ chosen words and maximize a **symmetric function of the column frequencies**.

Fredman–Komlós (1984); Körner (1986).

Comparison with the bounds already available

For a k-hash code:

$$\text{probabilistic lower bound:} \qquad R_k \geq \frac{1}{k-1} \log_2 \frac{1}{1 - k!/k^k};$$

$$\text{pruning upper bound:} \qquad R_k \leq \log_2 \frac{k}{k-1}.$$

Comparison with the bounds already available

For a k-hash code:

$$\text{probabilistic lower bound:} \qquad R_k \geq \frac{1}{k-1} \log_2 \frac{1}{1 - k!/k^k};$$

$$\text{pruning upper bound:} \qquad R_k \leq \log_2 \frac{k}{k-1}.$$

Fredman–Komlós gives, for $k \geq 4$,

$$R_k \leq \frac{k!}{k^{k-1}}.$$

Comparison with the bounds already available

For a k-hash code:

$$\text{probabilistic lower bound:} \qquad R_k \geq \frac{1}{k-1} \log_2 \frac{1}{1 - k!/k^k};$$

$$\text{pruning upper bound:} \qquad R_k \leq \log_2 \frac{k}{k-1}.$$

Fredman–Komlós gives, for $k \geq 4$,

$$R_k \leq \frac{k!}{k^{k-1}}.$$

For example:

k	$\log_2(k/(k-1))$	$k!/k^{k-1}$
3	0.58496	0.66667
4	0.41504	0.37500
5	0.32193	0.19200
6	0.26303	0.09259

Comparison with the bounds already available

For a k-hash code:

$$\text{probabilistic lower bound:} \qquad R_k \geq \frac{1}{k-1} \log_2 \frac{1}{1 - k!/k^k};$$

$$\text{pruning upper bound:} \qquad R_k \leq \log_2 \frac{k}{k-1}.$$

Fredman–Komlós gives, for $k \geq 4$,

$$R_k \leq \frac{k!}{k^{k-1}}.$$

For example:

k	$\log_2(k/(k-1))$	$k!/k^{k-1}$
3	0.58496	0.66667
4	0.41504	0.37500
5	0.32193	0.19200
6	0.26303	0.09259

For $k = 3$ pruning is better; from $k = 4$ onward the Fredman–Komlós bound becomes substantial.

Hansel's lemma: statement

Let K_m be the complete graph on m vertices. Let

$$G_1, \ldots, G_t$$

be bipartite graphs on the same vertex set (isolated vertices are allowed) such that

$$E(K_m) \subseteq \bigcup_{i=1}^{t} E(G_i).$$

Hansel's lemma: statement

Let K_m be the complete graph on m vertices. Let

$$G_1, \ldots, G_t$$

be bipartite graphs on the same vertex set (isolated vertices are allowed) such that

$$E(K_m) \subseteq \bigcup_{i=1}^{t} E(G_i).$$

For each G_i, define

$$\tau(G_i) = \frac{\#\{\text{non-isolated vertices of } G_i\}}{m}.$$

Hansel's lemma: statement

Let K_m be the complete graph on m vertices. Let

$$G_1, \ldots, G_t$$

be bipartite graphs on the same vertex set (isolated vertices are allowed) such that

$$E(K_m) \subseteq \bigcup_{i=1}^{t} E(G_i).$$

For each G_i, define

$$\tau(G_i) = \frac{\#\{\text{non-isolated vertices of } G_i\}}{m}.$$

Hansel's lemma

$$\boxed{\log_2 m \leq \sum_{i=1}^{t} \tau(G_i).}$$

The probabilistic proof is short enough to give in full.

Hansel: a random experiment on the bipartitions

For each i, fix a decomposition

$$V(G_i) = A_i \sqcup B_i \sqcup I_i,$$

where I_i is the set of isolated vertices and every edge of G_i joins A_i to B_i.

Hansel: a random experiment on the bipartitions

For each i, fix a decomposition

$$V(G_i) = A_i \sqcup B_i \sqcup I_i,$$

where I_i is the set of isolated vertices and every edge of G_i joins A_i to B_i. Independently for every i, choose with probability $1/2$ one of the two sides A_i or B_i.

Hansel: a random experiment on the bipartitions

For each i, fix a decomposition

$$V(G_i) = A_i \sqcup B_i \sqcup I_i,$$

where I_i is the set of isolated vertices and every edge of G_i joins A_i to B_i. Independently for every i, choose with probability $1/2$ one of the two sides A_i or B_i. We say that a vertex v **survives** if, for every i in which v is non-isolated, the chosen side of G_i contains v.

Hansel: a random experiment on the bipartitions

For each i, fix a decomposition

$$V(G_i) = A_i \sqcup B_i \sqcup I_i,$$

where I_i is the set of isolated vertices and every edge of G_i joins A_i to B_i. Independently for every i, choose with probability $1/2$ one of the two sides A_i or B_i. We say that a vertex v **survives** if, for every i in which v is non-isolated, the chosen side of G_i contains v. If

$$d(v) = \#\{i : v \text{ is non-isolated in } G_i\},$$

then

$$\mathbb{P}(v \text{ survives}) = 2^{-d(v)}.$$

Hansel: at most one vertex survives

Suppose that two distinct vertices u, v both survive. Because the graphs G_i cover all edges of K_m, there exists an i such that

$$\{u, v\} \in E(G_i).$$

Hansel: at most one vertex survives

Suppose that two distinct vertices u, v both survive. Because the graphs G_i cover all edges of K_m, there exists an i such that

$$\{u, v\} \in E(G_i).$$

Since G_i is bipartite, u and v lie on opposite sides of its bipartition. But the random experiment selected only *one* side of G_i, so they cannot both survive.

Hansel: at most one vertex survives

Suppose that two distinct vertices u, v both survive. Because the graphs G_i cover all edges of K_m, there exists an i such that

$$\{u, v\} \in E(G_i).$$

Since G_i is bipartite, u and v lie on opposite sides of its bipartition. But the random experiment selected only *one* side of G_i, so they cannot both survive. Hence the random number Z of surviving vertices satisfies

$$Z \leq 1.$$

Taking expectations gives

$$\sum_v 2^{-d(v)} = \mathbb{E}[Z] \leq 1. \tag{H1}$$

Hansel: Jensen completes the proof

The function $x \mapsto 2^{-x}$ is convex. Jensen's inequality gives

$$\frac{1}{m} \sum_{v} 2^{-d(v)} \geq 2^{-\frac{1}{m} \sum_{v} d(v)}.$$

Hansel: Jensen completes the proof

The function $x \mapsto 2^{-x}$ is convex. Jensen's inequality gives

$$\frac{1}{m} \sum_{v} 2^{-d(v)} \geq 2^{-\frac{1}{m} \sum_{v} d(v)}.$$

Using (H1),

$$\frac{1}{m} \geq 2^{-\frac{1}{m} \sum_{v} d(v)}.$$

Taking $\log_2$,

$$\log_2 m \leq \frac{1}{m} \sum_{v} d(v).$$

Finally, count incidences (v, i) for which v is non-isolated in G_i:

$$\sum_{v} d(v) = \sum_{i} m\tau(G_i).$$

Hansel: Jensen completes the proof

The function $x \mapsto 2^{-x}$ is convex. Jensen's inequality gives

$$\frac{1}{m}\sum_{v} 2^{-d(v)} \geq 2^{-\frac{1}{m}\sum_{v} d(v)}.$$

Using (H1),

$$\frac{1}{m} \geq 2^{-\frac{1}{m}\sum_{v} d(v)}.$$

Taking $\log_2$,

$$\log_2 m \leq \frac{1}{m}\sum_{v} d(v).$$

Finally, count incidences (v, i) for which v is non-isolated in G_i:

$$\sum_{v} d(v) = \sum_{i} m\tau(G_i).$$

Therefore

$$\boxed{\log_2 m \leq \sum_{i} \tau(G_i).}$$

From k-hashing to Hansel: fix $k-2$ words

Let

$$C \subseteq [k]^n, \qquad |C| = M,$$

be a k-hash code.

From k-hashing to Hansel: fix $k-2$ words

Let

$$C \subseteq [k]^n, \qquad |C| = M,$$

be a k-hash code. Choose $k-2$ distinct codewords

$$x_1, \ldots, x_{k-2} \in C.$$

From k-hashing to Hansel: fix $k-2$ words

Let

$$C \subseteq [k]^n, \qquad |C| = M,$$

be a k-hash code. Choose $k-2$ distinct codewords

$$x_1, \ldots, x_{k-2} \in C.$$

The vertices of our graphs will be the remaining codewords:

$$V = C \setminus \{x_1, \ldots, x_{k-2}\}, \qquad |V| = M - k + 2.$$

From k-hashing to Hansel: fix $k-2$ words

Let

$$C \subseteq [k]^n, \qquad |C| = M,$$

be a k-hash code. Choose $k-2$ distinct codewords

$$x_1, \ldots, x_{k-2} \in C.$$

The vertices of our graphs will be the remaining codewords:

$$V = C \setminus \{x_1, \ldots, x_{k-2}\}, \qquad |V| = M - k + 2.$$

For every coordinate i we now define a graph G_i on V.

The graph associated with coordinate i

Join two vertices $y_1, y_2 \in V$ when

$$(x_1)_i, \ldots, (x_{k-2})_i, (y_1)_i, (y_2)_i$$

are k pairwise distinct symbols. Equivalently,

$$\{y_1, y_2\} \in E(G_i) \iff |\{(x_1)_i, \ldots, (x_{k-2})_i, (y_1)_i, (y_2)_i\}| = k.$$

The graph associated with coordinate i

Join two vertices $y_1, y_2 \in V$ when

$$(x_1)_i, \ldots, (x_{k-2})_i, (y_1)_i, (y_2)_i$$

are k pairwise distinct symbols. Equivalently,

$$\{y_1, y_2\} \in E(G_i) \iff |\{(x_1)_i, \ldots, (x_{k-2})_i, (y_1)_i, (y_2)_i\}| = k.$$

If the symbols $(x_1)_i, \ldots, (x_{k-2})_i$ are not pairwise distinct, then G_i has no edges.

Why G_i is bipartite

Assume that

$$(x_1)_i, \ldots, (x_{k-2})_i$$

are pairwise distinct.

Why G_i is bipartite

Assume that

$$(x_1)_i, \ldots, (x_{k-2})_i$$

are pairwise distinct. Since the alphabet is $[k]$, exactly two symbols remain unused; call them a and b. The edges are exactly the pairs with one endpoint in

$$V_a = \{y : y_i = a\}$$

and one endpoint in

$$V_b = \{y : y_i = b\}.$$

Why G_i is bipartite

Assume that

$$(x_1)_i, \ldots, (x_{k-2})_i$$

are pairwise distinct. Since the alphabet is $[k]$, exactly two symbols remain unused; call them a and b. The edges are exactly the pairs with one endpoint in

$$V_a = \{y : y_i = a\}$$

and one endpoint in

$$V_b = \{y : y_i = b\}.$$

Thus G_i is the complete bipartite graph between V_a and V_b, together with possible isolated vertices. In particular, every non-isolated vertex satisfies

$$y_i \in \{a, b\}.$$

If one of V_a, V_b is empty, then vertices in the other class are isolated as well; later we shall therefore use an *upper bound* for the number of non-isolated vertices.

Why the graphs G_i cover the complete graph

Take two distinct vertices $y_1, y_2 \in V$. The k codewords

$$x_1, \ldots, x_{k-2}, y_1, y_2$$

are distinct.

Why the graphs G_i cover the complete graph

Take two distinct vertices $y_1, y_2 \in V$. The k codewords

$$x_1, \ldots, x_{k-2}, y_1, y_2$$

are distinct. Since C is a k-hash code, there is a coordinate i in which their k symbols are pairwise distinct. By definition,

$$\{y_1, y_2\} \in E(G_i).$$

Why the graphs G_i cover the complete graph

Take two distinct vertices $y_1, y_2 \in V$. The k codewords

$$x_1, \ldots, x_{k-2}, y_1, y_2$$

are distinct. Since C is a k-hash code, there is a coordinate i in which their k symbols are pairwise distinct. By definition,

$$\{y_1, y_2\} \in E(G_i).$$

Hence

$$E(K_{M-k+2}) \subseteq \bigcup_{i=1}^{n} E(G_i).$$

Hansel's lemma yields

$$\boxed{\log_2(M-k+2) \leq \sum_{i=1}^{n} \tau_i(x_1, \ldots, x_{k-2}).} \tag{1}$$

Symbol frequencies in the columns

For each coordinate i, define its frequency vector

$$f_i = (f_i[1], \ldots, f_i[k]) \in \Delta_k,$$

where

$$f_i[a] = \frac{1}{M} \#\{x \in C : x_i = a\}.$$

Symbol frequencies in the columns

For each coordinate i, define its frequency vector

$$f_i = (f_i[1], \ldots, f_i[k]) \in \Delta_k,$$

where

$$f_i[a] = \frac{1}{M} \#\{x \in C : x_i = a\}.$$

Of course,

$$f_i[a] \geq 0, \qquad \sum_{a=1}^{k} f_i[a] = 1.$$

Symbol frequencies in the columns

For each coordinate i, define its frequency vector

$$f_i = (f_i[1], \ldots, f_i[k]) \in \Delta_k,$$

where

$$f_i[a] = \frac{1}{M} \#\{x \in C : x_i = a\}.$$

Of course,

$$f_i[a] \geq 0, \qquad \sum_{a=1}^{k} f_i[a] = 1.$$

The quantity τ_i depends on the symbols taken by the selected codewords $x_1, \ldots, x_{k-2}$ in column i and on the frequency vector f_i.

An explicit upper bound for τ_i

Put

$$a_s = (x_s)_i, \qquad s = 1, \ldots, k-2.$$

If the a_s are not pairwise distinct, then G_i is empty and $\tau_i = 0$. Assume they are distinct. Every non-isolated vertex must use a symbol outside $\{a_1, \ldots, a_{k-2}\}$. In the whole code, the number of codewords using one of the two residual symbols is

$$M\left(1 - \sum_{s=1}^{k-2} f_i[a_s]\right).$$

None of the x_s is counted, since x_s uses the symbol a_s.

An explicit upper bound for τ_i

Put

$$a_s = (x_s)_i, \qquad s = 1, \ldots, k-2.$$

If the a_s are not pairwise distinct, then G_i is empty and $\tau_i = 0$. Assume they are distinct. Every non-isolated vertex must use a symbol outside $\{a_1, \ldots, a_{k-2}\}$. In the whole code, the number of codewords using one of the two residual symbols is

$$M\left(1 - \sum_{s=1}^{k-2} f_i[a_s]\right).$$

None of the x_s is counted, since x_s uses the symbol a_s. Dividing by the number $M - k + 2$ of vertices gives

$$\boxed{\tau_i \le \frac{M}{M-k+2}\left(1 - \sum_{s=1}^{k-2} f_i[a_s]\right)\mathbf{1}[a_1, \ldots, a_{k-2} \text{ all distinct}].} \tag{2}$$

The inequality can be strict if one of the two residual symbols does not occur among the vertices.

Now choose the $k-2$ words at random

Choose

$$x_1, \ldots, x_{k-2}$$

uniformly *without replacement* from C, keeping the order.

Now choose the $k-2$ words at random

Choose

$$x_1, \ldots, x_{k-2}$$

uniformly *without replacement* from C, keeping the order. Taking expectations in (1),

$$\log_2(M-k+2) \leq \sum_{i=1}^{n} \mathbb{E}[\tau_i]. \tag{3}$$

Now choose the $k-2$ words at random

Choose

$$x_1, \ldots, x_{k-2}$$

uniformly *without replacement* from C, keeping the order. Taking expectations in (1),

$$\log_2(M-k+2) \leq \sum_{i=1}^{n} \mathbb{E}[\tau_i]. \tag{3}$$

The advantage is that every $\mathbb{E}[\tau_i]$ can now be written solely in terms of the frequency vector f_i.

Probability of prescribed distinct symbols

Fix pairwise distinct symbols

$$a_1, \ldots, a_{k-2} \in [k].$$

The words are sampled without replacement. Because the prescribed symbols a_s are distinct, at step s none of the previously selected words belongs to the class with symbol a_s in coordinate i.

Probability of prescribed distinct symbols

Fix pairwise distinct symbols

$$a_1, \ldots, a_{k-2} \in [k].$$

The words are sampled without replacement. Because the prescribed symbols a_s are distinct, at step s none of the previously selected words belongs to the class with symbol a_s in coordinate i. Therefore

$$\begin{aligned}
&\mathbb{P}\big((x_1)_i = a_1, \ldots, (x_{k-2})_i = a_{k-2}\big) \\
&\quad = \frac{Mf_i[a_1]}{M} \frac{Mf_i[a_2]}{M-1} \cdots \frac{Mf_i[a_{k-2}]}{M-k+3} \\
&\quad = \left(\prod_{j=1}^{k-3} \frac{M}{M-j} \right) \prod_{s=1}^{k-2} f_i[a_s].
\end{aligned}$$

The expectation of τ_i

Substituting the previous probability into the upper bound (2) gives

$$\mathbb{E}[\tau_i] \leq Q_M \, \phi_k(f_i), \tag{4}$$

The expectation of τ_i

Substituting the previous probability into the upper bound (2) gives

$$\mathbb{E}[\tau_i] \leq Q_M \, \phi_k(f_i), \tag{4}$$

where

$$Q_M := \prod_{j=1}^{k-2} \frac{M}{M-j} = 1 + O_k(M^{-1})$$

The expectation of τ_i

Substituting the previous probability into the upper bound (2) gives

$$\mathbb{E}[\tau_i] \leq Q_M \, \phi_k(f_i), \tag{4}$$

where

$$Q_M := \prod_{j=1}^{k-2} \frac{M}{M-j} = 1 + O_k(M^{-1})$$

and

$$\phi_k(f) := \sum_{\substack{a_1,\ldots,a_{k-2} \in [k] \\ \text{all distinct}}} \left(\prod_{s=1}^{k-2} f[a_s] \right) \left(1 - \sum_{s=1}^{k-2} f[a_s] \right). \tag{5}$$

The expectation of τ_i

Substituting the previous probability into the upper bound (2) gives

$$\mathbb{E}[\tau_i] \leq Q_M \, \phi_k(f_i), \tag{4}$$

where

$$Q_M := \prod_{j=1}^{k-2} \frac{M}{M-j} = 1 + O_k(M^{-1})$$

and

$$\phi_k(f) := \sum_{\substack{a_1,\ldots,a_{k-2} \in [k] \\ \text{all distinct}}} \left(\prod_{s=1}^{k-2} f[a_s] \right) \left(1 - \sum_{s=1}^{k-2} f[a_s] \right). \tag{5}$$

The coding problem has become the maximization of a symmetric function on the simplex.

ϕ_k is a multiple of an elementary symmetric polynomial

Define

$$e_{k-1}(f) = \sum_{\substack{S \subseteq [k] \\ |S| = k-1}} \prod_{a \in S} f[a].$$

ϕ_k is a multiple of an elementary symmetric polynomial

Define

$$e_{k-1}(f) = \sum_{\substack{S \subseteq [k] \\ |S| = k-1}} \prod_{a \in S} f[a].$$

We claim that

$$\boxed{\phi_k(f) = (k-1)!\, e_{k-1}(f).} \tag{6}$$

ϕ_k is a multiple of an elementary symmetric polynomial

Define

$$e_{k-1}(f) = \sum_{\substack{S \subseteq [k] \\ |S| = k-1}} \prod_{a \in S} f[a].$$

We claim that

$$\boxed{\phi_k(f) = (k-1)!\, e_{k-1}(f).} \tag{6}$$

Indeed, for an ordered $(k-2)$-tuple of distinct symbols $(a_1, \ldots, a_{k-2})$,

$$1 - \sum_{s=1}^{k-2} f[a_s]$$

is the sum of the frequencies of the two missing symbols.

ϕ_k is a multiple of an elementary symmetric polynomial

Define

$$e_{k-1}(f) = \sum_{\substack{S \subseteq [k] \\ |S| = k-1}} \prod_{a \in S} f[a].$$

We claim that

$$\boxed{\phi_k(f) = (k-1)!\, e_{k-1}(f).} \tag{6}$$

Indeed, for an ordered $(k-2)$-tuple of distinct symbols $(a_1, \ldots, a_{k-2})$,

$$1 - \sum_{s=1}^{k-2} f[a_s]$$

is the sum of the frequencies of the two missing symbols. After expansion, every term is a product of $k-1$ distinct frequencies.

Counting the coefficient $(k-1)!$

Fix a $(k-1)$-element set

$$S \subseteq [k], \qquad |S| = k-1,$$

and the monomial

$$\prod_{a \in S} f[a].$$

To obtain it in the expansion of ϕ_k:

1. choose the element $d \in S$ coming from the factor

$$1 - \sum_{s=1}^{k-2} f[a_s];$$

there are $k-1$ choices;

Counting the coefficient $(k-1)!$

Fix a $(k-1)$-element set

$$S \subseteq [k], \qquad |S| = k-1,$$

and the monomial

$$\prod_{a \in S} f[a].$$

To obtain it in the expansion of ϕ_k:

1. choose the element $d \in S$ coming from the factor

$$1 - \sum_{s=1}^{k-2} f[a_s];$$

   there are $k-1$ choices;
2. the remaining $k-2$ elements of $S \setminus \{d\}$ can occur as $(a_1, \ldots, a_{k-2})$ in $(k-2)!$ orders.

Counting the coefficient $(k-1)!$

Fix a $(k-1)$-element set

$$S \subseteq [k], \qquad |S| = k-1,$$

and the monomial

$$\prod_{a \in S} f[a].$$

To obtain it in the expansion of ϕ_k:

1. choose the element $d \in S$ coming from the factor

$$1 - \sum_{s=1}^{k-2} f[a_s];$$

   there are $k-1$ choices;

2. the remaining $k-2$ elements of $S \setminus \{d\}$ can occur as $(a_1, \ldots, a_{k-2})$ in $(k-2)!$ orders.

Hence the total coefficient is

$$(k-1)(k-2)! = (k-1)!,$$

which proves (6).

Muirhead: the form we need

For an exponent vector $\alpha = (\alpha_1, \ldots, \alpha_k)$, let $[\alpha]$ denote the average of the monomials

$$f_1^{\beta_1} \cdots f_k^{\beta_k}$$

as β runs over the distinct permutations of α. We say that α **majorizes** β, and write $\alpha \succeq \beta$, if they have the same sum and, after arranging coordinates in decreasing order,

$$\sum_{j=1}^{r} \alpha_j \geq \sum_{j=1}^{r} \beta_j \qquad (1 \leq r < k).$$

Muirhead: the form we need

For an exponent vector $\alpha = (\alpha_1, \ldots, \alpha_k)$, let $[\alpha]$ denote the average of the monomials

$$f_1^{\beta_1} \cdots f_k^{\beta_k}$$

as β runs over the distinct permutations of α. We say that α **majorizes** β, and write $\alpha \succeq \beta$, if they have the same sum and, after arranging coordinates in decreasing order,

$$\sum_{j=1}^{r} \alpha_j \geq \sum_{j=1}^{r} \beta_j \qquad (1 \leq r < k).$$

Muirhead's inequality

For $f_1, \ldots, f_k \geq 0$,

$$\alpha \succeq \beta \quad \Longrightarrow \quad [\alpha] \geq [\beta].$$

Muirhead: the form we need

For an exponent vector $\alpha = (\alpha_1, \ldots, \alpha_k)$, let $[\alpha]$ denote the average of the monomials

$$f_1^{\beta_1} \cdots f_k^{\beta_k}$$

as β runs over the distinct permutations of α. We say that α **majorizes** β, and write $\alpha \succeq \beta$, if they have the same sum and, after arranging coordinates in decreasing order,

$$\sum_{j=1}^{r} \alpha_j \geq \sum_{j=1}^{r} \beta_j \qquad (1 \leq r < k).$$

Muirhead's inequality

For $f_1, \ldots, f_k \geq 0$,

$$\alpha \succeq \beta \quad \Longrightarrow \quad [\alpha] \geq [\beta].$$

We use Muirhead only to motivate the symmetric bound; immediately afterward we prove the special case we need directly.

Muirhead applied to the elementary symmetric polynomial

Set $h = k - 1$ and

$$\alpha = (\underbrace{1, \ldots, 1}_{h}, 0).$$

Every nonnegative integer exponent vector of total degree h majorizes α: placing two units on the same variable only makes the vector more concentrated.

Muirhead applied to the elementary symmetric polynomial

Set $h = k - 1$ and

$$\alpha = (\underbrace{1, \ldots, 1}_{h}, 0).$$

Every nonnegative integer exponent vector of total degree h majorizes α: placing two units on the same variable only makes the vector more concentrated. By Muirhead, the average of every monomial type of degree h is therefore at least

$$[\alpha] = \frac{e_h(f)}{\binom{k}{h}}.$$

Muirhead applied to the elementary symmetric polynomial

Set $h = k - 1$ and

$$\alpha = (\underbrace{1, \dots, 1}_{h}, 0).$$

Every nonnegative integer exponent vector of total degree h majorizes α: placing two units on the same variable only makes the vector more concentrated. By Muirhead, the average of every monomial type of degree h is therefore at least

$$[\alpha] = \frac{e_h(f)}{\binom{k}{h}}.$$

On the other hand,

$$\frac{1}{k^h} \left(\sum_{a=1}^{k} f[a] \right)^h = \frac{1}{k^h}$$

is the average of the k^h ordered products $f[i_1] \cdots f[i_h]$, hence a convex combination of such symmetric type-averages. Therefore

$$\frac{1}{k^h} \geq \frac{e_h(f)}{\binom{k}{h}},$$

so

$$\boxed{e_h(f) \leq \binom{k}{h} k^{-h}.} \tag{M}$$

For $h = k - 1$ this is exactly the bound we need.

The same inequality without a black box

To make the Fredman–Komlós proof self-contained, we now prove the case $h = k - 1$ of (M) directly. We keep all coordinates except two fixed and replace those two by their average, preserving their sum. We shall show that this **pairwise averaging** does not decrease e_{k-1}.

The same inequality without a black box

To make the Fredman–Komlós proof self-contained, we now prove the case $h = k - 1$ of (M) directly. We keep all coordinates except two fixed and replace those two by their average, preserving their sum. We shall show that this **pairwise averaging** does not decrease e_{k-1}. Repeating the operation drives the frequency vector toward the uniform distribution; by continuity, the uniform vector maximizes e_{k-1}.

Pairwise averaging: the exact formula

Fix the coordinates

$$r = (r_1, \ldots, r_{k-2})$$

and vary only two coordinates x, y while keeping

$$s = x + y$$

fixed.

Pairwise averaging: the exact formula

Fix the coordinates

$$r = (r_1, \ldots, r_{k-2})$$

and vary only two coordinates x, y while keeping

$$s = x + y$$

fixed. For the elementary symmetric polynomial of degree $k-1$,

$$e_{k-1}(x, y, r) = xy\, e_{k-3}(r) + (x + y) e_{k-2}(r). \tag{7}$$

Why these are the only two terms:

- if both x and y are selected, we must select $k-3$ additional variables from r;

Pairwise averaging: the exact formula

Fix the coordinates

$$r = (r_1, \ldots, r_{k-2})$$

and vary only two coordinates x, y while keeping

$$s = x + y$$

fixed. For the elementary symmetric polynomial of degree $k-1$,

$$e_{k-1}(x, y, r) = xy\, e_{k-3}(r) + (x + y) e_{k-2}(r). \tag{7}$$

Why these are the only two terms:

- if both x and y are selected, we must select $k-3$ additional variables from r;
- if exactly one of x, y is selected, we must select all $k-2$ variables in r.

Pairwise averaging: the exact formula

Fix the coordinates

$$r = (r_1, \ldots, r_{k-2})$$

and vary only two coordinates x, y while keeping

$$s = x + y$$

fixed. For the elementary symmetric polynomial of degree $k-1$,

$$e_{k-1}(x, y, r) = xy\, e_{k-3}(r) + (x + y) e_{k-2}(r). \tag{7}$$

Why these are the only two terms:

- if both x and y are selected, we must select $k-3$ additional variables from r;
- if exactly one of x, y is selected, we must select all $k-2$ variables in r.

For fixed s, the second term is constant.

Pairwise averaging increases e_{k-1}

By (7), when $x + y = s$ is fixed, the only variable term is

$$xy\, e_{k-3}(r).$$

Pairwise averaging increases e_{k-1}

By (7), when $x + y = s$ is fixed, the only variable term is

$$xy\, e_{k-3}(r).$$

Since $e_{k-3}(r) \geq 0$ and

$$xy \leq \left(\frac{x+y}{2}\right)^2 = \frac{s^2}{4},$$

replacing

$$(x, y) \longmapsto \left(\frac{s}{2}, \frac{s}{2}\right)$$

does not decrease e_{k-1}. Repeated averaging of unequal pairs approaches the uniform distribution without decreasing e_{k-1}.

Pairwise averaging increases e_{k-1}

By (7), when $x + y = s$ is fixed, the only variable term is

$$xy\, e_{k-3}(r).$$

Since $e_{k-3}(r) \geq 0$ and

$$xy \leq \left(\frac{x+y}{2}\right)^2 = \frac{s^2}{4},$$

replacing

$$(x, y) \longmapsto \left(\frac{s}{2}, \frac{s}{2}\right)$$

does not decrease e_{k-1}. Repeated averaging of unequal pairs approaches the uniform distribution without decreasing e_{k-1}. By continuity and compactness of the simplex, the maximum is attained at

$$f[1] = \cdots = f[k] = \frac{1}{k}.$$

Value of the maximum

At the uniform distribution

$$u = \left(\frac{1}{k}, \ldots, \frac{1}{k}\right),$$

we have

$$e_{k-1}(u) = \binom{k}{k-1}\left(\frac{1}{k}\right)^{k-1} = \frac{1}{k^{k-2}}.$$

Value of the maximum

At the uniform distribution

$$u = \left(\frac{1}{k}, \ldots, \frac{1}{k}\right),$$

we have

$$e_{k-1}(u) = \binom{k}{k-1}\left(\frac{1}{k}\right)^{k-1} = \frac{1}{k^{k-2}}.$$

Therefore, for every frequency vector f,

$$\phi_k(f) = (k-1)!e_{k-1}(f) \leq \frac{(k-1)!}{k^{k-2}}.$$

Value of the maximum

At the uniform distribution

$$u = \left(\frac{1}{k}, \ldots, \frac{1}{k}\right),$$

we have

$$e_{k-1}(u) = \binom{k}{k-1}\left(\frac{1}{k}\right)^{k-1} = \frac{1}{k^{k-2}}.$$

Therefore, for every frequency vector f,

$$\phi_k(f) = (k-1)!\, e_{k-1}(f) \leq \frac{(k-1)!}{k^{k-2}}.$$

Rewrite the right-hand side as

$$\frac{(k-1)!}{k^{k-2}} = \frac{k!}{k^{k-1}}. \tag{8}$$

This is precisely the Fredman–Komlós constant.

A uniform bound for every column

From (4) and (8), for every coordinate i,

$$\mathbb{E}[\tau_i] \leq Q_M \frac{k!}{k^{k-1}}, \qquad Q_M = \prod_{j=1}^{k-2} \frac{M}{M-j}.$$

A uniform bound for every column

From (4) and (8), for every coordinate i,

$$\mathbb{E}[\tau_i] \leq Q_M \frac{k!}{k^{k-1}}, \qquad Q_M = \prod_{j=1}^{k-2} \frac{M}{M-j}.$$

Substituting into (3),

$$\log_2(M-k+2) \leq nQ_M \frac{k!}{k^{k-1}}. \tag{9}$$

A uniform bound for every column

From (4) and (8), for every coordinate i,

$$\mathbb{E}[\tau_i] \leq Q_M \frac{k!}{k^{k-1}}, \qquad Q_M = \prod_{j=1}^{k-2} \frac{M}{M-j}.$$

Substituting into (3),

$$\log_2(M-k+2) \leq nQ_M \frac{k!}{k^{k-1}}. \tag{9}$$

This is already a finite inequality, valid for every k-hash code with $M \geq k$.

Asymptotics: Fredman–Komlós

Consider a sequence of k-hash codes C_n with $M_n = |C_n|$ realizing the limsup in the definition of the rate. If M_n does not tend to infinity along a relevant subsequence, the rate is 0 and there is nothing to prove.

Asymptotics: Fredman–Komlós

Consider a sequence of k-hash codes C_n with $M_n = |C_n|$ realizing the limsup in the definition of the rate. If M_n does not tend to infinity along a relevant subsequence, the rate is 0 and there is nothing to prove.

Otherwise,

$$Q_{M_n} = 1 + o(1)$$

and

$$\frac{1}{n}\log_2(M_n - k + 2) = \frac{1}{n}\log_2 M_n + o(1).$$

Divide (9) by n and take the limsup:

$$\boxed{R_k \leq \frac{k!}{k^{k-1}}.}$$

For $k \geq 4$ this improves the pruning bound.

Why the bound is not useful for $k = 3$

The formal Fredman–Komlós expression would give

$$\frac{3!}{3^2} = \frac{2}{3},$$

whereas pruning gives

$$\log_2 \frac{3}{2} = 0.58496\ldots < \frac{2}{3}.$$

Why the bound is not useful for $k = 3$

The formal Fredman–Komlós expression would give

$$\frac{3!}{3^2} = \frac{2}{3},$$

whereas pruning gives

$$\log_2 \frac{3}{2} = 0.58496\ldots < \frac{2}{3}.$$

Thus trifference is exceptional from the Fredman–Komlós point of view: the elementary one-symbol pruning argument is stronger. This is another reason why trifference is a particularly stubborn testing ground for general methods.

Where does the Fredman–Komlós proof lose information?

The crucial step is

$$\phi_k(f_i) \leq \max_{f \in \Delta_k} \phi_k(f) = \phi_k(u) = \frac{k!}{k^{k-1}}$$

separately for every column. Equality requires uniform frequencies.

Where does the Fredman–Komlós proof lose information?

The crucial step is

$$\phi_k(f_i) \le \max_{f \in \Delta_k} \phi_k(f) = \phi_k(u) = \frac{k!}{k^{k-1}}$$

separately for every column. Equality requires uniform frequencies. If a column is non-uniform, the balancing argument gives

$$\phi_k(f_i) < \phi_k(u)$$

apart from irrelevant boundary degeneracies.

Where does the Fredman–Komlós proof lose information?

The crucial step is

$$\phi_k(f_i) \le \max_{f \in \Delta_k} \phi_k(f) = \phi_k(u) = \frac{k!}{k^{k-1}}$$

separately for every column. Equality requires uniform frequencies. If a column is non-uniform, the balancing argument gives

$$\phi_k(f_i) < \phi_k(u)$$

apart from irrelevant boundary degeneracies.

The entry point for improvements

A stronger proof must exploit the fact that all columns cannot simultaneously behave, after the relevant conditioning, as independent worst-case uniform columns.

This is the natural starting point for modern refinements of the method.

Natural bridge to the second half of the course

The Fredman–Komlós mechanism can be summarized as

$$\boxed{k-2 \text{ fixed words} \to \text{bipartite cover} \to \text{Hansel} \to \text{averaging} \to \text{symmetrization}.}$$

Natural bridge to the second half of the course

The Fredman–Komlós mechanism can be summarized as

$$\boxed{k-2 \text{ fixed words} \to \text{bipartite cover} \to \text{Hansel} \to \text{averaging} \to \text{symmetrization.}}$$

The second half of the course can start exactly from the last inequality and ask:

- can the words $x_1, \dots, x_{k-2}$ be selected less blindly?

Natural bridge to the second half of the course

The Fredman–Komlós mechanism can be summarized as

$$\boxed{k-2 \text{ fixed words} \to \text{bipartite cover} \to \text{Hansel} \to \text{averaging} \to \text{symmetrization.}}$$

The second half of the course can start exactly from the last inequality and ask:

- can the words $x_1, \dots, x_{k-2}$ be selected less blindly?
- can non-uniform columns be exploited instead of immediately replacing them by the uniform worst case?

Natural bridge to the second half of the course

The Fredman–Komlós mechanism can be summarized as

$$\boxed{k-2 \text{ fixed words} \to \text{bipartite cover} \to \text{Hansel} \to \text{averaging} \to \text{symmetrization.}}$$

The second half of the course can start exactly from the last inequality and ask:

- can the words $x_1, \dots, x_{k-2}$ be selected less blindly?
- can non-uniform columns be exploited instead of immediately replacing them by the uniform worst case?
- how does the mechanism extend from $b = k$ to general (b, k)?

Natural bridge to the second half of the course

The Fredman–Komlós mechanism can be summarized as

$$\boxed{k-2 \text{ fixed words} \to \text{bipartite cover} \to \text{Hansel} \to \text{averaging} \to \text{symmetrization.}}$$

The second half of the course can start exactly from the last inequality and ask:

- can the words $x_1, \dots, x_{k-2}$ be selected less blindly?
- can non-uniform columns be exploited instead of immediately replacing them by the uniform worst case?
- how does the mechanism extend from $b = k$ to general (b, k)?

This is where modern improvements and the results on (b, k)-hashing enter.

The general (b, k) case

The same Fredman–Komlós strategy, fixing $k-2$ codewords at a time, also gives the general bound

$$\boxed{R(b,k) \le \frac{b^{\underline{k-1}}}{b^{k-1}} \log_2(b-k+2),}$$

where

$$b^{\underline{k-1}} = b(b-1)\cdots(b-k+2).$$

The general (b, k) case

The same Fredman–Komlós strategy, fixing $k-2$ codewords at a time, also gives the general bound

$$\boxed{R(b,k) \le \frac{b^{\underline{k-1}}}{b^{k-1}} \log_2(b-k+2),}$$

where

$$b^{\underline{k-1}} = b(b-1)\cdots(b-k+2).$$

For $b = k$ this immediately becomes

$$\frac{k!}{k^{k-1}},$$

since $\log_2 2 = 1$. If fewer than $k-2$ codewords are fixed, more than two codewords remain to be compared at once; this is where the **hypergraph** version of Hansel's lemma enters. We will not need it in the first module.

Fredman–Komlós (1984); Körner–Marton (1988).

Possible further topics after Lecture 3

- **Graph entropy**: an information-theoretic reformulation of the Fredman–Komlós mechanism.

Possible further topics after Lecture 3

- **Graph entropy**: an information-theoretic reformulation of the Fredman–Komlós mechanism.
- **Separating hash families**: variants in which the patterns to be separated are changed.

Possible further topics after Lecture 3

- **Graph entropy**: an information-theoretic reformulation of the Fredman–Komlós mechanism.
- **Separating hash families**: variants in which the patterns to be separated are changed.
- **Cover-free and superimposed codes**: connections with group testing.

Possible further topics after Lecture 3

- **Graph entropy**: an information-theoretic reformulation of the Fredman–Komlós mechanism.
- **Separating hash families**: variants in which the patterns to be separated are changed.
- **Cover-free and superimposed codes**: connections with group testing.
- **Zero-error list decoding**: hashing viewed as a channel-capacity problem.

Possible further topics after Lecture 3

- **Graph entropy**: an information-theoretic reformulation of the Fredman–Komlós mechanism.
- **Separating hash families**: variants in which the patterns to be separated are changed.
- **Cover-free and superimposed codes**: connections with group testing.
- **Zero-error list decoding**: hashing viewed as a channel-capacity problem.
- **Lovász Local Lemma and Moser–Tardos**: probabilistic constructions with local dependencies.

Possible further topics after Lecture 3

- **Graph entropy**: an information-theoretic reformulation of the Fredman–Komlós mechanism.
- **Separating hash families**: variants in which the patterns to be separated are changed.
- **Cover-free and superimposed codes**: connections with group testing.
- **Zero-error list decoding**: hashing viewed as a channel-capacity problem.
- **Lovász Local Lemma and Moser–Tardos**: probabilistic constructions with local dependencies.

Modern refinements of the Fredman–Komlós bound and results on (b, k)-hashing will instead be the subject of the second half of the course.

Essential references — Lecture 3


G. Hansel, *Nombre minimal de contacts de fermeture nécessaires pour réaliser une fonction booléenne symétrique de n variables*, C. R. Acad. Sci. Paris 258 (1964), 6037–6040.


M. L. Fredman, J. Komlós, *On the Size of Separating Systems and Families of Perfect Hash Functions*, SIAM J. Algebraic Discrete Methods 5(1) (1984), 61–68.


J. Körner, *Fredman–Komlós Bounds and Information Theory*, SIAM J. Algebraic Discrete Methods 7(4) (1986), 560–570.


J. Körner, K. Marton, *New Bounds for Perfect Hashing via Information Theory*, European J. Combin. 9(6) (1988), 523–530.

Closing the module

One family of problems, three viewpoints

Probability	:	how unlikely is a bad configuration?
Polynomials / rank	:	can local witnesses be turned into zeros and dimension bounds?
Hansel / frequencies	:	what does it cost to cover all residual pairs, coordinate by coordinate?

Closing the module

One family of problems, three viewpoints

Probability	:	how unlikely is a bad configuration?
Polynomials / rank	:	can local witnesses be turned into zeros and dimension bounds?
Hansel / frequencies	:	what does it cost to cover all residual pairs, coordinate by coordinate?

The second half of the course starts from the last viewpoint and shows how one actually improves classical bounds for perfect and (b, k)-hashing.