Randomized Algorithms

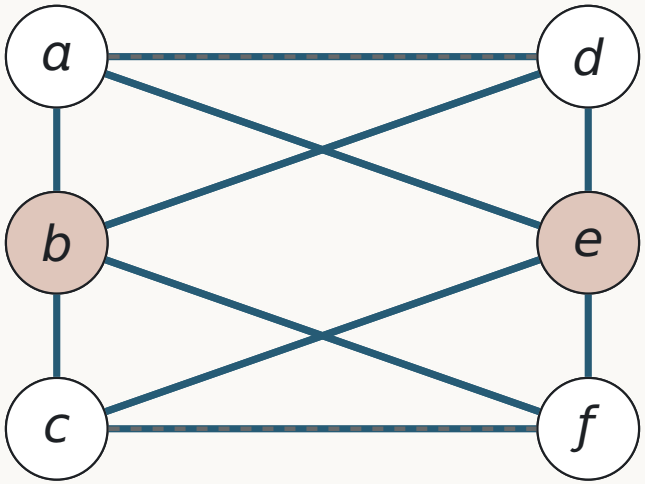
Probabilistic method I: existence via randomness

Alex Gittens
Department of Computer Science, Rensselaer Polytechnic Institute

Fall 2026

How many edges can we cut?

Split the vertices into two sides (shaded and unshaded). An edge is **cut** if its endpoints are on different sides.



This partition cuts 6 of the 10 edges (solid). Can we find a partition that cuts more? Putting b and e on one side cuts 8 of the 10 edges. Putting b and e on one side cuts 8 of the 10 edges.

Does every graph have a cut containing at least half its edges?

Existence and construction

The graph-cut problem raises two questions:

1. **Does** an object with the property we want exist?
2. Can we **find** one efficiently?

So far in this course, we have used randomness to design algorithms.

We will now use randomness to prove that an object exists, even if the proof does not tell us how to find it efficiently.

PART 1

MAX-CUT

1

MAX-CUT

Given $G = (V, E)$ with m edges, **MAX-CUT** asks for $S \subseteq V$ maximizing the number of edges between S and $V \setminus S$.

- Finding a maximum cut is NP-hard.
- A weaker question: does every graph have a cut containing *at least half* its edges?

Assign each vertex to a side by an independent fair coin flip.

- What fraction of the edges do you expect to be cut?
- Does the answer depend on the structure of G?

A random cut

Flip an independent fair coin for each vertex: heads goes in S. Fix an edge $\{u, v\}$. Of the four equally likely outcomes for its endpoints,

$$\underbrace{(S,S),\ (\bar{S},\bar{S})}_{\text{not cut}} \qquad \underbrace{(S,\bar{S}),\ (\bar{S},S)}_{\text{cut}} \qquad \Longrightarrow \qquad \mathbb{P}(uv \text{ is cut}) = \tfrac{1}{2}.$$

Write the cut size as a sum of indicators, $X = \sum_{uv \in E} I_{uv}$ with $I_{uv} = \mathbf{1}\{uv \text{ is cut}\}$.

By linearity of expectation,

$$\mathbb{E}X = \sum_{uv \in E} \mathbb{E}I_{uv} = \frac{m}{2}.$$

So some cut is at least that large. A random variable cannot be below its mean on every outcome, so some outcome of the coin flips has $X \geq m/2$.

Every graph has a large cut

Every graph with m edges has a cut containing at least $m/2$ edges.

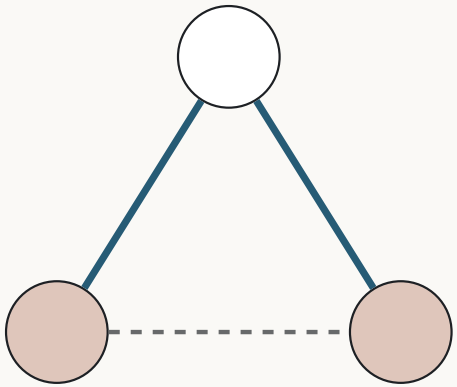
The cut-edge indicators are not independent

The indicators I_{uv} are *not* independent.

In a triangle, two of the three vertices always land on the same side, so at most two edges are cut:

$$\mathbb{P}(\text{all three cut}) = 0 \neq \tfrac{1}{8}.$$

The proof only used that each single edge is cut with probability $1/2$. Linearity of expectation holds for any random variables, dependent or not.



Can we guarantee more than half?

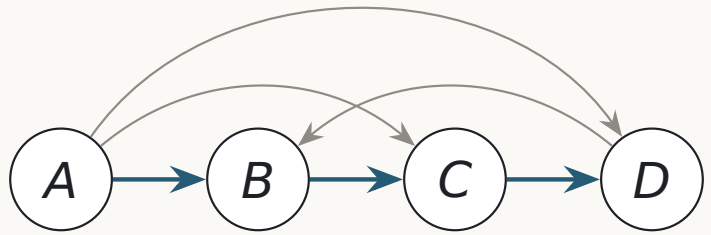
In K_n with n even, $m = n(n-1)/2$, and the best cut puts $n/2$ vertices on each side:

$$\frac{\mathrm{OPT}_{\mathrm{MaxCut}}(K_n)}{m} = \frac{(n/2)^2}{n(n-1)/2} = \frac{n}{2(n-1)} \longrightarrow \frac{1}{2}.$$

So no bound of the form "every graph has a cut of size $c\,m$" can have $c > 1/2$.

Exercise 1: Hamiltonian paths in tournaments

A **tournament** directs every edge of a complete graph. In a directed **Hamiltonian path**, consecutive vertices are joined by forward-directed edges.



one Hamiltonian path is highlighted

Show that some tournament on n vertices has at least

$$\frac{n!}{2^{n-1}}$$

directed Hamiltonian paths.
Hint. Orient the edges independently at random. For a fixed ordering $v_1, \ldots, v_n$, what is the probability that all $n-1$ consecutive edges point forward?

PART 2

Avoiding monochromatic cliques

2

Can we avoid a monochromatic triangle?

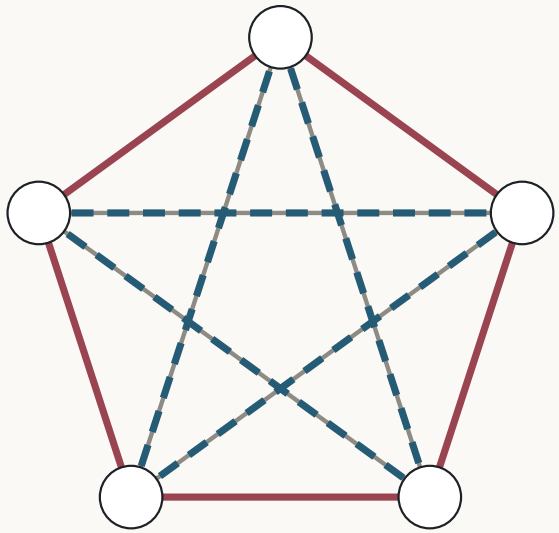
Can we color all the edges of a complete graph without creating a triangle whose edges have the same color?

Color each edge of K_5 red or blue.

Can you avoid every single-color triangle?

Yes: color the pentagon red and the pentagram blue (dashed).

Any triangle uses at least one pentagon edge and at least one pentagram edge, so it has both colors.



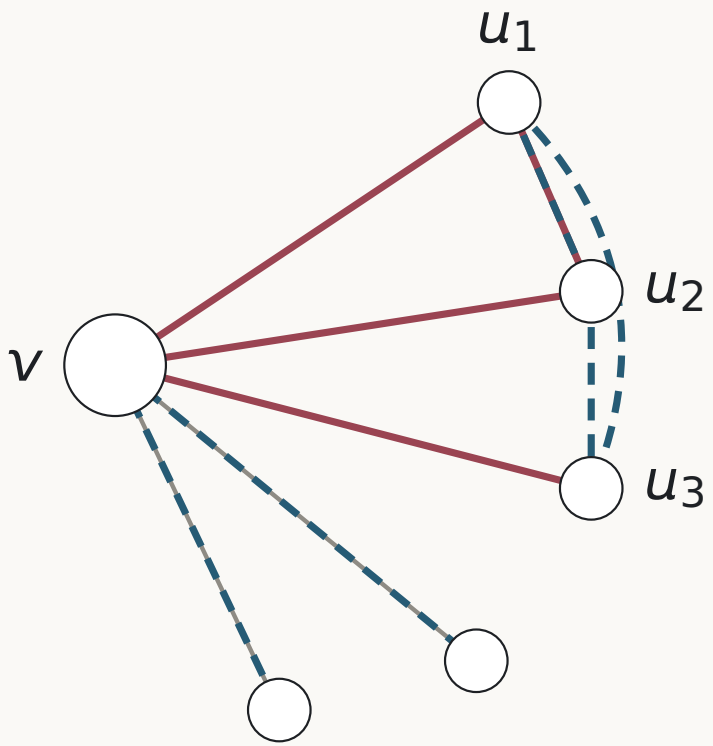
Why K_6 forces a monochromatic triangle

Claim. Every red-blue coloring of K_6 has a single-color triangle.

Fix a vertex v. It has 5 edges, so at least 3 share a color. Say vu_1, vu_2, vu_3 are red.

If any edge among u_1, u_2, u_3 is red, it forms a red triangle with v.

Otherwise all three are blue, and $u_1u_2u_3$ is a blue triangle.



So 5 vertices can avoid single-color triangles and 6 cannot. What about larger cliques?

Ramsey numbers

Diagonal Ramsey number

$R(k, k)$ is the smallest n such that every red-blue coloring of the edges of K_n contains k vertices whose edges all have the same color.

$$R(k, k) > n \quad \Longleftrightarrow \quad \text{some coloring of } K_n \text{ has no single-color } K_k.$$

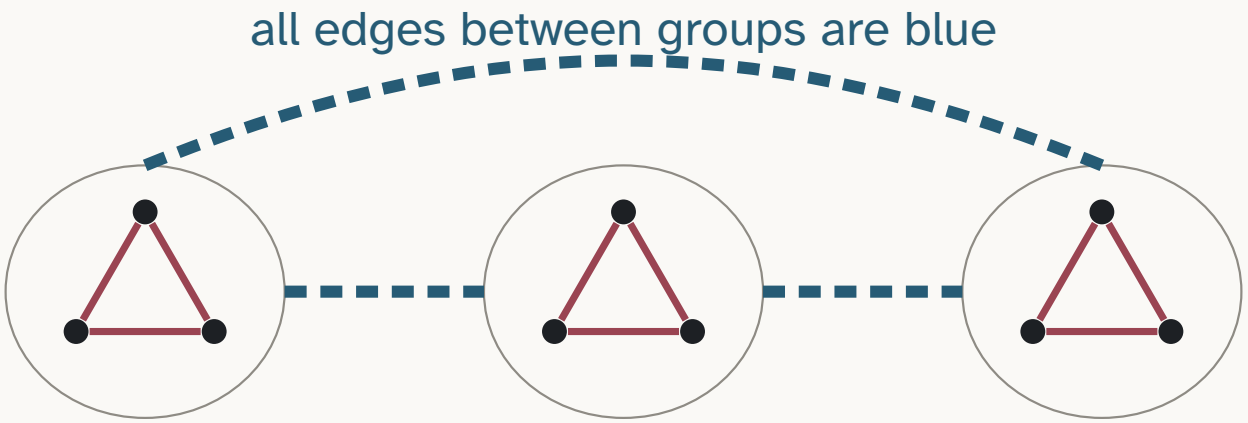
So a lower bound on $R(k, k)$ is proved by exhibiting, or proving the existence of, one good coloring.

We just showed $R(3, 3) = 6$. **How fast does $R(k, k)$ grow with k?**

Exact values are difficult to determine: $R(4, 4) = 18$, and $43 \leq R(5, 5) \leq 46$.

A simple construction gives $R(k, k) > (k-1)^2$

Split $(k-1)^2$ vertices into $k-1$ groups of size $k-1$. Color edges inside a group red and *every* edge between groups blue.



$k = 4$: three groups of three

- A red clique lies inside one group: at most $k-1$ vertices.
- A blue clique uses at most one vertex per group: at most $k-1$ vertices.

Is the truth polynomial in k, or exponential?

Random edge colorings

Our construction gives $R(k,k) > (k-1)^2$. Can we prove an exponential lower bound? Color every edge of K_n independently red or blue with probability $1/2$.

Fix a set S of k vertices. It spans $\binom{k}{2}$ edges, so

$$\mathbb{P}(S \text{ is single-color}) = 2 \cdot 2^{-\binom{k}{2}} = 2^{1-\binom{k}{2}}.$$

Let Y be the number of single-color k-sets. Summing over all $\binom{n}{k}$ sets,

$$\mathbb{E}Y = \binom{n}{k} 2^{1-\binom{k}{2}}.$$

What condition on $\mathbb{E}Y$ would guarantee a coloring with $Y = 0$?

Y is a nonnegative integer. If $Y \geq 1$ for every coloring, then $\mathbb{E}Y \geq 1$. So

$$\mathbb{E}Y < 1 \quad \Longrightarrow \quad \text{some coloring has } Y = 0.$$

How large a Ramsey coloring can we hope for?

For a random two-coloring of K_n, our first-moment argument works if

$$\binom{n}{k} 2^{1-\binom{k}{2}} < 1.$$

Using $\binom{n}{k} \leq n^k/k!$ gives

$$\mathbb{E}Y \leq \frac{n^k}{k!} 2^{1-k(k-1)/2}.$$

The principal competition is between n^k and $2^{k^2/2}$. This suggests trying n on the order of $2^{k/2}$. We still have to verify that the expected number is below one.

An exponential lower bound

Take $n = \lfloor 2^{k/2} \rfloor$. Substituting into the preceding estimate,

$$\mathbb{E}Y \;\le\; \frac{2^{k^2/2}}{k!} \cdot 2^{1-\frac{k^2}{2}+\frac{k}{2}} \;=\; \frac{2^{1+k/2}}{k!} \;<\; 1 \qquad (k \ge 3).$$

Erdős, 1947

For every $k \ge 3$, $R(k,k) > 2^{k/2}$.

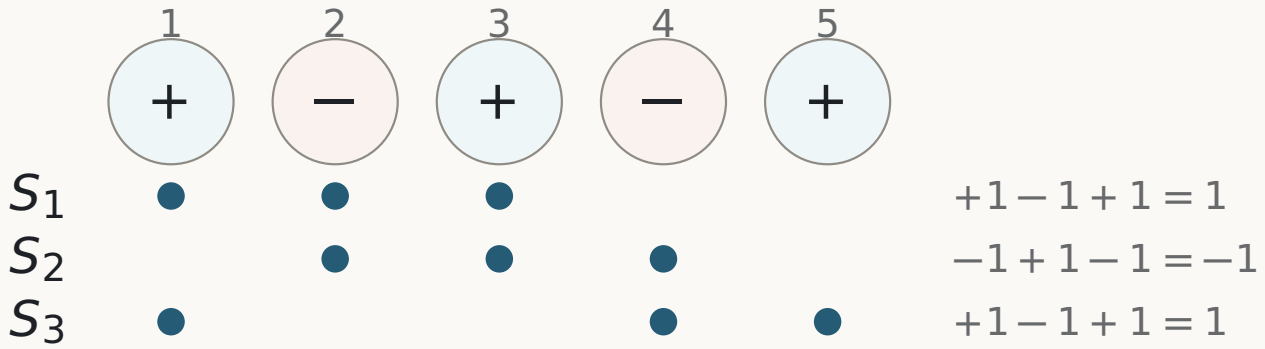
There is a coloring of $K_{\lfloor 2^{k/2} \rfloor}$ without a monochromatic K_k. Since Ramsey numbers are integers, the displayed bound follows.

	construction	random coloring
lower bound on $R(k,k)$	$(k-1)^2$	$2^{k/2}$
$k = 20$	361	1024
$k = 40$	1521	1048576

The random coloring improves the quadratic lower bound to an exponential one.

Exercise 2: balancing overlapping sets

Color each element of $[n]$ with a sign $+1$ or -1. We want the positive and negative signs to nearly cancel in every set.



Let $S_1, \ldots, S_m \subseteq [n]$, each with $|S_j| \le k$. Show that some $\chi : [n] \to \{-1, +1\}$ satisfies, for every j,

$$\left| \sum_{i \in S_j} \chi(i) \right| \le \sqrt{2k \ln(4m)}.$$

PART 3

Error-correcting codes

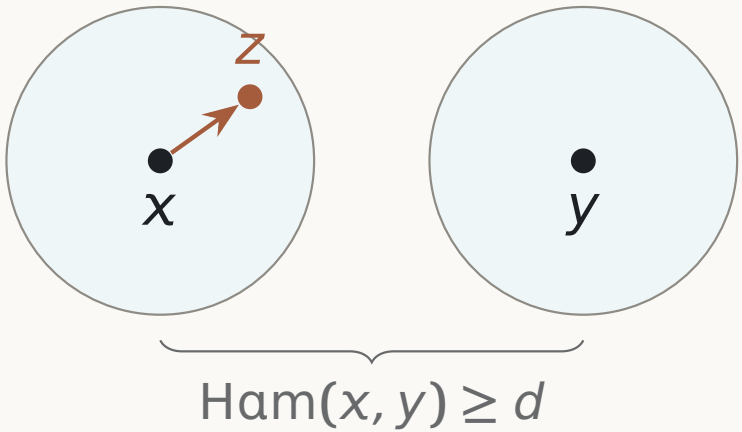
3

Correcting errors with distance

Send `00000` for 0 and `11111` for 1. If at most 2 bits flip, majority vote recovers the message. But we sent 5 bits to carry 1.

More generally, use a collection of binary strings as **codewords**. The farther apart these strings are, the more errors we can correct. A **code** is $C \subseteq \{0,1\}^n$; the **Hamming distance** is $\mathrm{Ham}(x,y) = \#\{i : x_i \neq y_i\}$.

schematic: Hamming balls drawn as discs



If all codewords are at distance $\geq d$, the balls of radius $\lfloor (d-1)/2 \rfloor$ around them are disjoint.

So if at most $\lfloor (d-1)/2 \rfloor$ bits flip, the received word z is still closest to the sent codeword x.

The repetition code above has $d = 5$, so it corrects 2 errors.

The tradeoff between rate and distance

A code with M codewords carries $\log_2 M$ message bits per n bits sent. Its **rate** is $R = \frac{\log_2 M}{n}$.

Code	M	distance	rate
All of $\{0,1\}^n$	2^n	1	1
Repetition: $\{0^n, 1^n\}$	2	n	$1/n \to 0$

Adding codewords lets us represent more messages, but makes it harder to keep every pair far apart.

Question. Fix $0 < \delta < 1/2$. Are there codes, for every large n, with

$$|C| = 2^{\Omega(n)} \qquad \text{and} \qquad \text{minimum distance } \geq \delta n \text{ ?}$$

That is, constant rate while correcting a constant fraction of errors (approximately half the relative minimum distance).

Counting bad pairs in a random code

Choose N binary strings independently and uniformly at random. How many pairs should we expect to violate the minimum-distance requirement?

Write the sampled strings as $X_1, \dots, X_N \in \{0,1\}^n$. Call a pair **bad** if $\mathrm{Ham}(X_i, X_j) < \delta n$, and let Y be the number of bad pairs.

For one pair, each bit disagrees independently with probability $1/2$, so

$$q := \mathbb{P}(\text{a given pair is bad}) = \mathbb{P}\big(\mathrm{Bin}(n, \tfrac{1}{2}) < \delta n\big).$$

For $\delta < 1/2$ this is a lower-tail event, so q is exponentially small in n. We will estimate it later.

There are $\binom{N}{2}$ pairs, so

$$\mathbb{E}Y = \binom{N}{2} q.$$

Repeated strings have distance 0, so they count as bad pairs.

First attempt: avoid all bad pairs

Apply the first-moment method. If $\mathbb{E}Y = \binom{N}{2}q < 1$, some sample has no bad pairs and is itself a code.

To satisfy the condition, we need

$$\frac{N(N-1)}{2}q < 1.$$

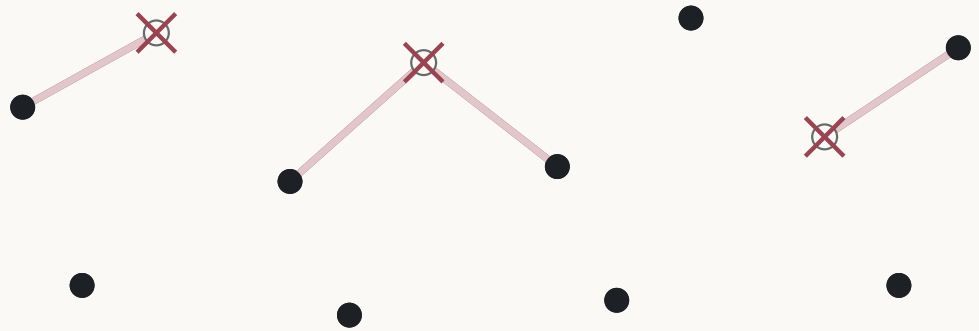
For large N, the left side is approximately $N^2 q/2$. Thus this argument allows N on the order of $q^{-1/2}$.

For larger N, the expected number of bad pairs increases. But a sample containing a few bad pairs need not be useless.

Can we remove the bad pairs without discarding the whole sample?

Remove one endpoint of each bad pair

Represent the sample by a graph: each vertex is a sampled string, and an edge joins two strings that are too close.



$N = 12$ sampled strings, $Y = 4$ bad pairs. Deleting one endpoint of each bad pair leaves no bad pair. Here 3 deletions suffice for 4 bad pairs.

Each bad pair costs at most one deletion, so at least $N - Y$ strings survive, and they form a code with distance $\geq \delta n$. Some sample does at least as well as average:

$$\text{some code has } \geq \mathbb{E}[N - Y] = N - \binom{N}{2} q \geq N - \frac{N^2 q}{2} \text{ codewords.}$$

The distance between two random strings

For two independent uniform binary strings X_i, X_j, define

$$Z_\ell = \mathbf{1}\{X_i(\ell) \neq X_j(\ell)\}, \qquad \ell = 1, \ldots, n.$$

X_i	0	1	0	1	1	0	0	1
X_j	0	0	0	0	1	1	0	1

three differing coordinates in this example

Each Z_ℓ is an independent Bernoulli$(1/2)$ variable. Therefore

$$D := \text{Ham}(X_i, X_j) = \sum_{\ell=1}^{n} Z_\ell, \qquad \mathbb{E}D = \frac{n}{2}.$$

A pair is bad if $D < \delta n$, where $\delta < 1/2$.

Chernoff bound for a bad pair

Recall the lower-tail bound for a sum of independent Bernoulli variables:

$$\mathbb{P}(Z \le (1-\eta)\mu) \le e^{-\eta^2\mu/2}, \qquad \mu = \mathbb{E}Z, \quad 0 \le \eta \le 1.$$

For $D = \mathrm{Ham}(X_i, X_j)$, the mean is $\mu = n/2$. To obtain the threshold δn, set

$$(1-\eta)\frac{n}{2} = \delta n \qquad \Longrightarrow \qquad \eta = 1 - 2\delta.$$

Thus the probability of a bad pair satisfies

$$\boxed{q := \mathbb{P}(D < \delta n) \;\le\; Q := \exp\left(-\frac{n(1-2\delta)^2}{4}\right).}$$

For fixed $\delta < 1/2$, the upper bound Q decreases exponentially with n.

How many codewords survive deletion?

We do not choose the probability q: it is determined by the random experiment. Chernoff supplies a value Q that we know satisfies $q \leq Q$. Choose

$$N = \left\lfloor \frac{1}{2Q} \right\rfloor.$$

For sufficiently large n, $N \geq 1$, and $NQ \leq 1/2$. Hence

$$\mathbb{E}[N - Y] \geq N - \frac{N^2 q}{2} \geq N - \frac{N^2 Q}{2} \geq \frac{3}{4} N.$$

Since $Q = e^{-cn}$ for $c = (1 - 2\delta)^2/4 > 0$, we have N on the order of e^{cn}. Some repaired sample therefore contains exponentially many well-separated strings.

Codes with constant rate and distance exist

Fix $0 < \delta < 1/2$. We proved that for all sufficiently large n, some $C \subseteq \{0,1\}^n$ has

$$\mathrm{Ham}(x,y) \geq \delta n \quad (x \neq y), \qquad |C| \geq \frac{3}{4}\left\lfloor \frac{e^{cn}}{2} \right\rfloor, \quad c = \frac{(1-2\delta)^2}{4}.$$

The constant factors and the floor are negligible compared with the exponential growth. In particular, for all sufficiently large n, $|C| \geq e^{cn/2}$, so

$$\frac{\log_2 |C|}{n} \geq \frac{c}{2 \ln 2} > 0.$$

Constant rate and relative distance

For every fixed $\delta < 1/2$, there are binary codes of minimum distance at least δn and rate bounded away from zero as n grows.

What have we proved?

Problem	Random experiment	Why a good object exists	Name
MAX-CUT	each vertex on a random side	$\mathbb{E}X = m/2$, so some cut has $X \geq m/2$	averaging
Ramsey	each edge a random color	$\mathbb{E}Y < 1$, so some coloring has $Y = 0$	first-moment method
Codes	random strings, then delete endpoints of bad pairs	$\mathbb{E}[N - Y]$ is large, so some repaired code is large	alteration

These proofs show that good objects exist. When can we efficiently find one?

Existence versus efficient construction

- **MAX-CUT.** Random sampling gives a cut with at least $m/2$ edges. We will later find such a cut deterministically.
- **Ramsey.** Random coloring proves that an exponentially large coloring without a monochromatic K_k exists. Finding comparably good explicit colorings is difficult.
- **Codes.** Alteration proves the existence of exponentially many well-separated strings, but directly constructing the codebook takes exponential time.

When can we turn an existence argument into an efficient algorithm, or remove the randomness entirely?

Questions?

Appendix: balanced cuts

The bound $m/2$ is asymptotically tight, but for an even number n of vertices we can guarantee slightly more. Choose a uniformly random subset S of exactly $n/2$ vertices.

For a fixed edge uv, condition on which side contains u:

$$\mathbb{P}(uv \text{ is cut}) = 2 \cdot \frac{1}{2} \cdot \frac{n/2}{n-1} = \frac{n}{2(n-1)}.$$

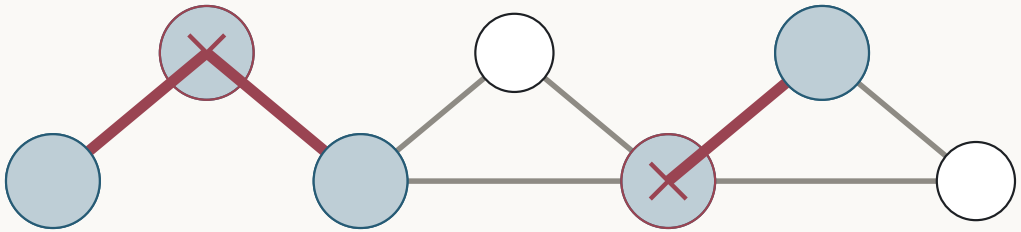
The endpoint assignments are dependent, but linearity of expectation still gives

$$\mathbb{E}|\partial S| = \frac{n}{2(n-1)} m.$$

Hence some cut has at least this many edges. Equality holds for K_n.

Appendix: independent sets by alteration

An **independent set** contains no two adjacent vertices. Select each vertex independently with probability p. In the selected vertices, some edges may remain.



Selected vertices are shaded. Thick red edges are conflicts within the selection. Delete an endpoint of each conflict. The remaining selected vertices form an independent set.

Let X count selected vertices and Y count edges with both endpoints selected. At least $X - Y$ vertices remain after deletion.

Appendix: independent-set calculation

Suppose G has n vertices and $m \geq n/2$ edges. Select vertices independently with probability p.

$$\mathbb{E}X = pn, \qquad \mathbb{E}Y = p^2 m, \qquad \mathbb{E}[X - Y] = pn - p^2 m.$$

The quadratic is largest at

$$p = \frac{n}{2m} \leq 1.$$

Therefore some selection leaves an independent set of size at least

$$pn - p^2 m = \frac{n^2}{2m} - \frac{n^2}{4m} = \boxed{\frac{n^2}{4m}}.$$

We did not require the random selection to be independent. We removed the internal edges afterward.

Appendix: improving the coding bound

The main lecture established that exponentially many binary codewords can have minimum distance proportional to n.

Can we guarantee a *larger* codebook? Our deletion argument retains a number of strings proportional to $1/q$, where q is the probability that two random strings are too close.

Chernoff gave an upper bound on q. We can do better by counting the strings that lie too close to a fixed string.

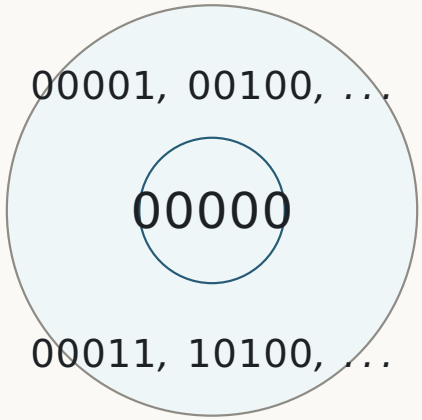
Gilbert–Varshamov bound: target

For $0 < \delta < 1/2$ and every $\varepsilon > 0$, sufficiently long binary codes can have relative minimum distance δ and rate at least $1 - H_2(\delta) - \varepsilon$.

We will explain the function H_2 and derive the bound.

Appendix: counting nearby binary strings

Fix $x = 00000$. To obtain a string at Hamming distance r, choose exactly r coordinates to change.



distance 0: $\binom{n}{0}$ strings

distance 1: $\binom{n}{1}$ strings

distance 2: $\binom{n}{2}$ strings

schematic Hamming ball

The number of strings within distance t of x is

$$B(n,t) = \sum_{r=0}^{t} \binom{n}{r}.$$

All 2^n strings are equally likely, so

$$\mathbb{P}(\mathrm{Ham}(x,X) \le t) = \frac{B(n,t)}{2^n}.$$

Appendix: the exact probability of a bad pair

Our distance condition is strict: a pair is bad when $\mathrm{Ham}(X_i, X_j) < \delta n$.

Hamming distance is an integer. The largest forbidden distance is

$$t = \lceil \delta n \rceil - 1.$$

For example, if $\delta n = 3$, then distances $0, 1, 2$ are bad; if $\delta n = 3.4$, distances $0, 1, 2, 3$ are bad.

For fixed X_i, the other string is uniform over all 2^n possibilities:

$$\boxed{q = \frac{1}{2^n} \sum_{r=0}^{t} \binom{n}{r}.}$$

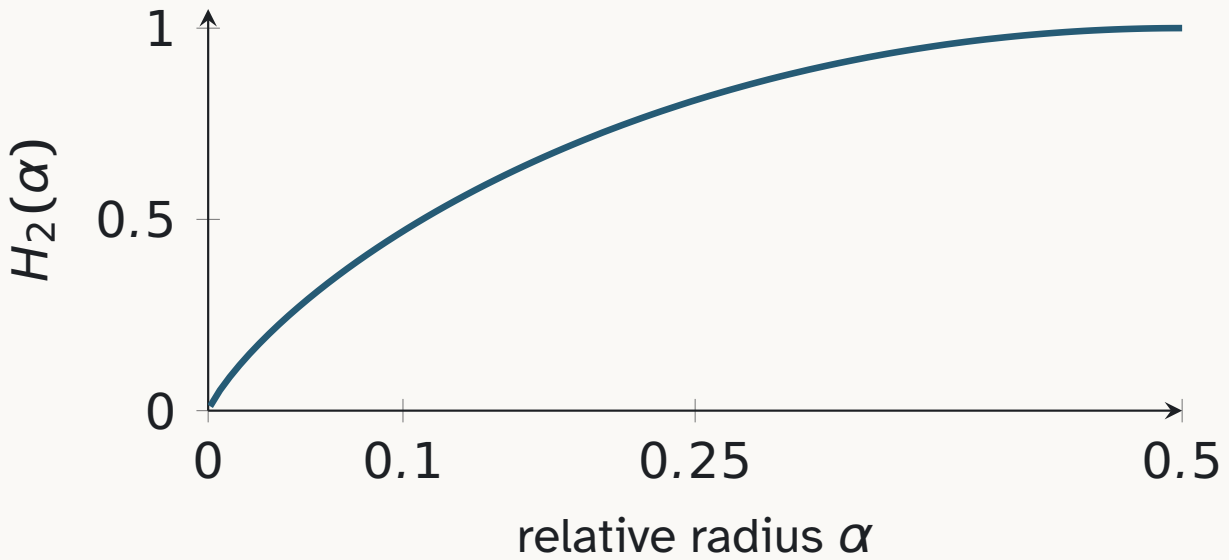
This is an *exact* probability. We now want a simple upper bound on the binomial sum, sharper than the Chernoff estimate.

Appendix: the binary entropy function

For $0 < \alpha < 1$, define

$$H_2(\alpha) = -\alpha \log_2 \alpha - (1-\alpha)\log_2(1-\alpha).$$

The quantity $2^{nH_2(\alpha)}$ describes the exponential scale of the number of strings within distance about αn of a fixed string.



For $\alpha < 1/2$, the Hamming ball occupies an exponentially small fraction of the full cube of 2^n strings.

Appendix: the Hamming-ball bound

Volume of a Hamming ball

For $0 < \alpha \leq 1/2$ with αn an integer,

$$\sum_{r=0}^{\alpha n} \binom{n}{r} \leq 2^{nH_2(\alpha)}.$$

The left-hand side counts the strings in the ball. The right-hand side gives its exponential growth rate.

The proof weights the strings by the probabilities $\alpha^r(1-\alpha)^{n-r}$. Inside the ball, each such weight is at least the weight at the boundary $r = \alpha n$. Their total weight cannot exceed one.

Appendix: proof of the Hamming-ball bound

For $0 < \alpha \leq 1/2$ and $r \leq \alpha n$,

$$\alpha^r (1-\alpha)^{n-r} \geq \alpha^{\alpha n} (1-\alpha)^{(1-\alpha)n}.$$

Using the binomial theorem,

$$\begin{aligned}
1 &= \sum_{r=0}^{n} \binom{n}{r} \alpha^r (1-\alpha)^{n-r} \\
&\geq \sum_{r=0}^{\alpha n} \binom{n}{r} \alpha^r (1-\alpha)^{n-r} \\
&\geq \alpha^{\alpha n} (1-\alpha)^{(1-\alpha)n} \sum_{r=0}^{\alpha n} \binom{n}{r} \\
&= 2^{-nH_2(\alpha)} \sum_{r=0}^{\alpha n} \binom{n}{r}.
\end{aligned}$$

Rearrange to obtain the claimed upper bound.

Appendix: the Gilbert–Varshamov bound

Set $t = \lceil \delta n \rceil - 1$ and $\alpha = t/n < \delta$. Because H_2 increases on $[0, 1/2]$, the volume bound gives

$$q = \frac{B(n,t)}{2^n} \le 2^{-n(1-H_2(\alpha))} \le 2^{-n(1-H_2(\delta))}.$$

Choose the sample size using this upper bound as we did with Chernoff. After deleting bad pairs, some code has size

$$|C| \ge \frac{3}{8} 2^{n(1-H_2(\delta))} - 1.$$

Thus for any $\varepsilon > 0$ and sufficiently large n,

$$\boxed{\frac{\log_2 |C|}{n} \ge 1 - H_2(\delta) - \varepsilon.}$$

The alteration method has not changed. Counting Hamming balls gave a stronger probability estimate and therefore a larger code.

Appendix: a greedy proof of the same bound

Start with an empty code. Repeatedly add any string at distance at least δn from every chosen codeword. Stop when none remains.

At termination, every binary string lies within distance $t = \lceil \delta n \rceil - 1$ of some codeword. Otherwise we could add it. The balls of radius t therefore cover the entire cube:

$$|C|\, B(n, t) \geq 2^n.$$

Using $B(n, t) \leq 2^{nH_2(\delta)}$,

$$|C| \geq 2^{n(1-H_2(\delta))}.$$

This is a deterministic existence argument, but a direct implementation still examines exponentially many strings.

Backup: a sharper Ramsey constant

Keep the factor $k!$ and use Stirling, $k! \geq \sqrt{2\pi k}\,(k/e)^k$. With

$$n = \left\lfloor \frac{k\,2^{k/2}}{e\sqrt{2}} \right\rfloor,$$

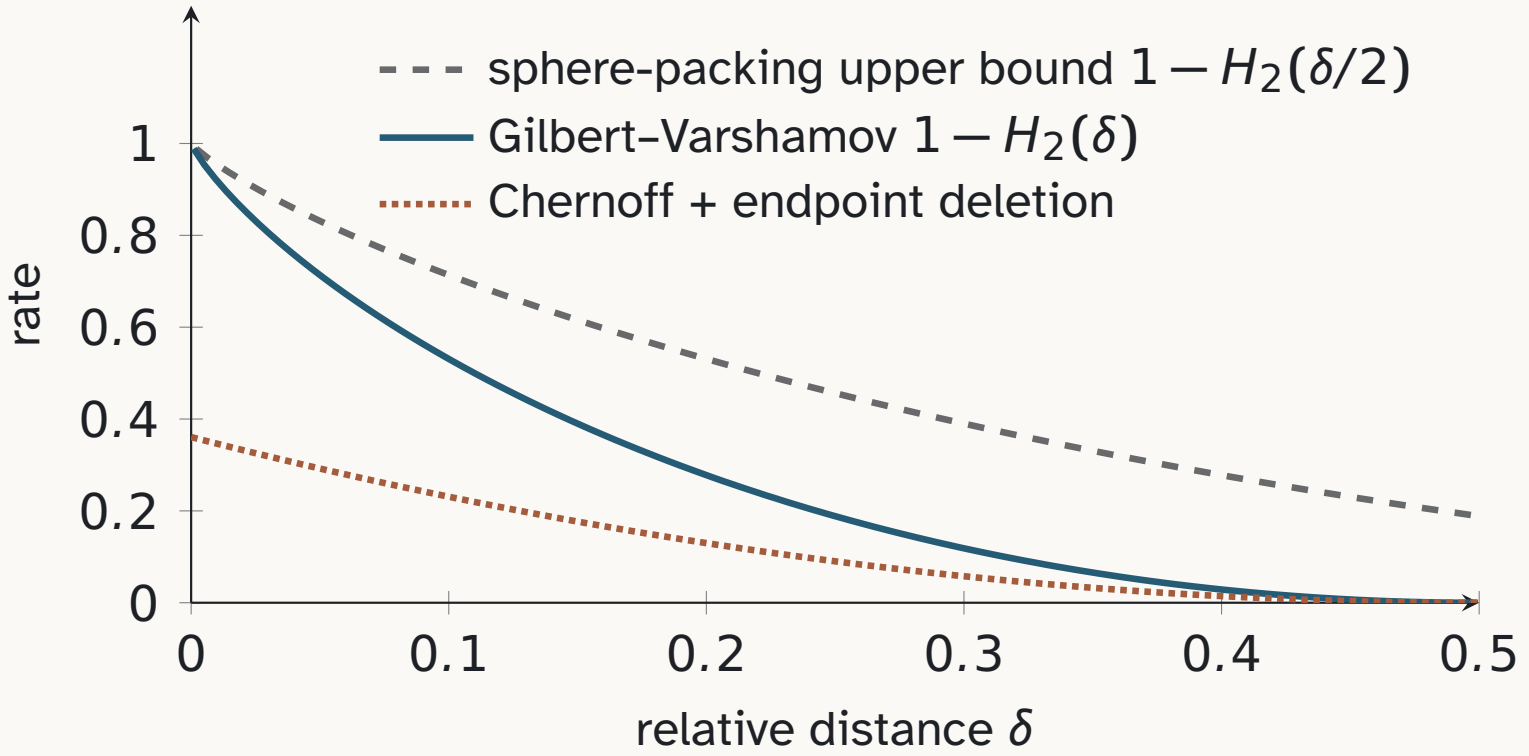
$$\mathbb{E}Y \leq \frac{n^k}{k!}\,2^{1-\frac{k(k-1)}{2}} \leq \frac{k^k\,2^{\frac{k^2}{2}-\frac{k}{2}}e^{-k}}{\sqrt{2\pi k}\,k^k e^{-k}} \cdot 2^{1-\frac{k^2}{2}+\frac{k}{2}} = \frac{2}{\sqrt{2\pi k}} < 1.$$

So $R(k,k) > \dfrac{k\,2^{k/2}}{e\sqrt{2}}$. This improves the bound by a factor proportional to k, using a sharper count of the k-sets rather than any new probabilistic idea.

Backup: what is known about $R(k, k)$

- Upper bound $R(k, k) \leq 4^k$ (Erdős–Szekeres, 1935), improved to $(4 - \varepsilon)^k$ for a fixed $\varepsilon > 0$ only in 2023 (Campos, Griffiths, Morris, Sahasrabudhe).
- Lower bound $2^{k/2}$ (Erdős, 1947). The best improvement since is a factor of 2 in the constant, by Spencer using the Lovász local lemma.
- Explicit constructions remain far below $2^{k/2}$.

Backup: bounds on the rate–distance tradeoff



The best possible rate lies between the solid and dashed curves. Whether Gilbert–Varshamov is tight for binary codes is a long-standing open problem.