

Problem Setting

Consider the exam cheating protocol where the true exam solution is a binary vector

$$B \in \{0,1\}^{20}.$$

The sender is allowed to transmit a message

$$M \in \{0,1\}^{10},$$

and the receiver must output a decoded answer vector

$$A \in \{0,1\}^{20}.$$

The score is the number of positions where A matches B . The goal of any encoding–decoding scheme is to maximise the guaranteed score, meaning the minimum score over all possible 2^{20} exams.

Equivalently, this problem can be stated as minimising the worst-case Hamming distance

$$d(B, A),$$

between the true solution and the decoded answer.

Reformulation as a Covering Problem

Since the message length is 10 bits, there are at most

$$2^{10} = 1024$$

distinct messages. For any deterministic decoding scheme, each message corresponds to exactly one decoded answer vector $A \in \{0,1\}^{20}$.

Hence, any encoding scheme implicitly defines a set

$$\mathcal{C} \subseteq \{0,1\}^{20}$$

of at most 1024 decoded answer vectors. For any true exam solution B , the encoding must select a message whose decoded vector $A \in \mathcal{C}$ is as close as possible to B .

The guaranteed performance of the scheme is therefore determined by the covering radius

$$R = \max_{B \in \{0,1\}^{20}} \min_{A \in \mathcal{C}} d(B, A).$$

The guaranteed exam score is then

$$20 - R.$$

Counting Argument (Sphere-Covering Bound)

For any vector $A \in \{0,1\}^{20}$, the number of vectors within Hamming distance at most R of A is

$$V(20, R) = \sum_{k=0}^R \binom{20}{k}.$$

For the set \mathcal{C} to cover the entire space $\{0,1\}^{20}$, the total number of vectors covered by all Hamming balls of radius R must satisfy:

$$|\mathcal{C}| \cdot V(20, R) \geq 2^{20}.$$

Since $|\mathcal{C}| \leq 1024$, this gives the necessary condition:

$$V(20, R) \geq \frac{2^{20}}{1024} = 1024.$$

Evaluating the Bound

We now evaluate $V(20, R)$ for small values of R :

- $R = 2$:

$$V(20, 2) = \binom{20}{0} + \binom{20}{1} + \binom{20}{2} = 1 + 20 + 190 = 211$$

- $R = 3$:

$$V(20, 3) = 211 + \binom{20}{3} = 211 + 1140 = 1351$$

Hence:

$$V(20, 2) < 1024 \text{ but } V(20, 3) > 1024.$$

This implies that no scheme using only 1024 decoded answer vectors can cover all possible exams with Hamming radius $R \leq 2$.

Therefore, for any encoding scheme:

$$R \geq 3.$$

Consequence for the Guaranteed Score

Since the worst-case number of incorrect answers is at least 3, the guaranteed score satisfies:

$$\text{Guaranteed score} \leq 20 - 3 = 17.$$

This establishes a fundamental upper bound:

No encoding scheme using only 10 bits can guarantee more than 17 correct answers in the worst case.

Discussion and Tightness of the Bound.

The above derived bound is information theoretic and is applicable to any conceivable encoding and decoding structure, independent of computational capability and algorithmic complexity.

The bound is also almost strict: to get a guaranteed score of 17 would need to build a set of 1024 decoded answer vectors with radius of coverage 3 in 20-dimensional Hamming space, a highly nontrivial coding theoretic problem. Simple, practical constructions usually cover radii of 4 or 5, the guaranteed scores of 16 or 15 respectively.

Therefore, a guaranteed score of 17 is theoretically, possible in principle, but, as the given constraints dictate, it is the true upper limit possible.

Conclusion

Through a counting argument, any 10-bit message encoding scheme can produce at most 1024 vectors of decoded answers. These vectors are not sufficient to span the space of all exams of 20 bits with Hamming radius less than 3. Therefore, there is no plan that can be sure of over 17 correct answers in the worst scenario.

This defines a strict and strict upper limit on the maximum guaranteed score that can be attained on the problem.

References

1. F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, 1977.
2. Wikipedia, *Covering Code*. https://en.wikipedia.org/wiki/Covering_code
3. Wikipedia, *Hamming Distance*. https://en.wikipedia.org/wiki/Hamming_distance