Combinatorial Version of the Slepian-Wolf Coding Theorem for Binary Strings

D.A. Chumbalov

Moscow Institute of Physics and Technology

CCR 2013
Moscow, September 25

Participants

Participants

Alice

Participants

Alice

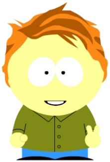
Bob

Problem statement.

Participants

Alice

Bob



Charlie

Problem statement.

- Alice is given a binary string X, $|X| = n$
- Bob is given a binary string Y, $|Y| = n$
- X and Y differ from each other in $c = c(n)$ positions

Problem statement.

- Alice is given a binary string X, $|X| = n$
- Bob is given a binary string Y, $|Y| = n$
- X and Y differ from each other in $c = c(n)$ positions

- Charlie wants to know both the strings X and Y
- Alice and Bob can send bits to Charlie

Problem statement.

- Alice is given a binary string X, $|X| = n$
- Bob is given a binary string Y, $|Y| = n$
- X and Y differ from each other in $c = c(n)$ positions

- Charlie wants to know both the strings X and Y
- Alice and Bob can send bits to Charlie

- All three participants know the parameters n and c.

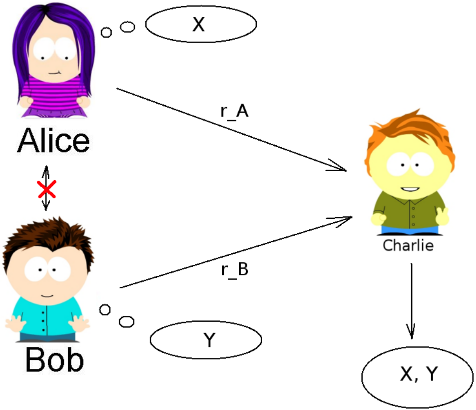
Problem statement.

- Alice is given a binary string X, $|X| = n$
- Bob is given a binary string Y, $|Y| = n$
- X and Y differ from each other in $c = c(n)$ positions

- Charlie wants to know both the strings X and Y
- Alice and Bob can send bits to Charlie

- All three participants know the parameters n and c.
- No communication is possible between Alice and Bob
- No feedback can be sent from Charlie to Alice and Bob.

Problem statement.

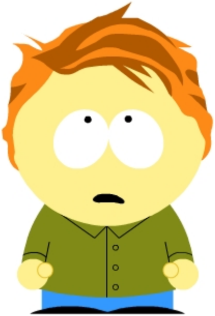


- Alice sends Charlie some message of lengh r_A
- Bob sends Charlie some message of lengh r_B
- Charlie reconstructs X and Y from the pair (r_A, r_B)

Question.

Question:

For which pairs (r_A, r_B) such a protocol exists?



Slepian-Wolf theorem.

Preparation

Let $(x_1, y_1), ..., (x_n, y_n)$ be i.i.d. pairs of random variables (each x_i and y_i range over some finite sets A and B resp.).
Alice holds $X = (x_1, ...x_n)$ and Bob holds $Y = (y_1, ..., y_n)$.
A pair of reals (R_X, R_Y) is called *achievable* if there exist functions

$$code_{A,n} : A^n \to \{0,1\}^{nR_X}$$

$$code_{B,n} : B^n \to \{0,1\}^{nR_Y}$$

$$decode_n : \{0,1\}^{nR_X} \times \{0,1\}^{nR_Y} \to A^n \times B^n$$

such that $Prob[decode_n(code_{A,n}(X), code_{B,n}(Y)) \neq (X,Y))] \to 0$ as $n \to \infty$.

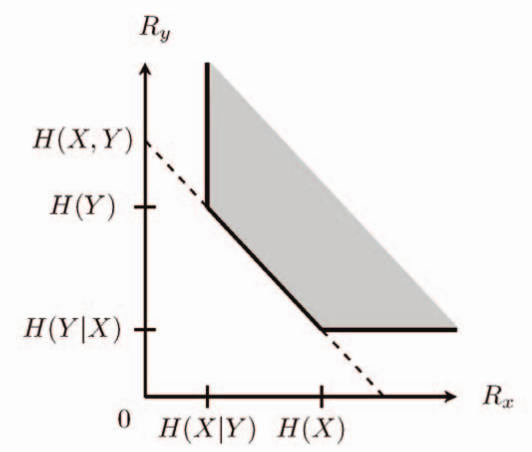
Slepian-Wolf theorem.

[D. Slepian, J. K. Wolf, 1973]

A pair of reals (R_X, R_Y) is achievable if

$$\begin{cases} R_X + R_Y \geq H(X,Y), \\ R_X \geq H(X|Y), \\ R_Y \geq H(Y|X). \end{cases}$$

Slepian-Wolf theorem.



Two cases.

Two main cases of the problem

- The distance between two strings c is an absolute constant
- The distance between two strings $c = \alpha n$, a constant fraction of n

Case 1: c is an absolute constant

Case 1: c is an absolute constant

Lower bound:

For every communication protocol for our problem

$$\begin{cases} r_A + r_B \geq n + \log_2(\sum_{k=0}^{c} C_n^k), \\ r_A \geq \log_2(\sum_{k=0}^{c} C_n^k), \\ r_B \geq \log_2(\sum_{k=0}^{c} C_n^k). \end{cases}$$

Case 1: c is an absolute constant

Lower bound:

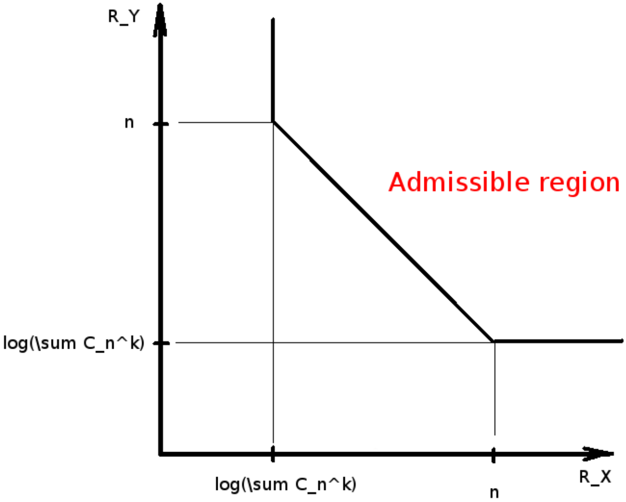
For every communication protocol for our problem

$$\begin{cases} r_A + r_B \geq n + \log_2(\sum_{k=0}^{c} C_n^k), \\ r_A \geq \log_2(\sum_{k=0}^{c} C_n^k), \\ r_B \geq \log_2(\sum_{k=0}^{c} C_n^k). \end{cases}$$

Idea of the proof

Standard counting argument.

Case 1: c is an absolute constant



Case 1: c is an absolute constant

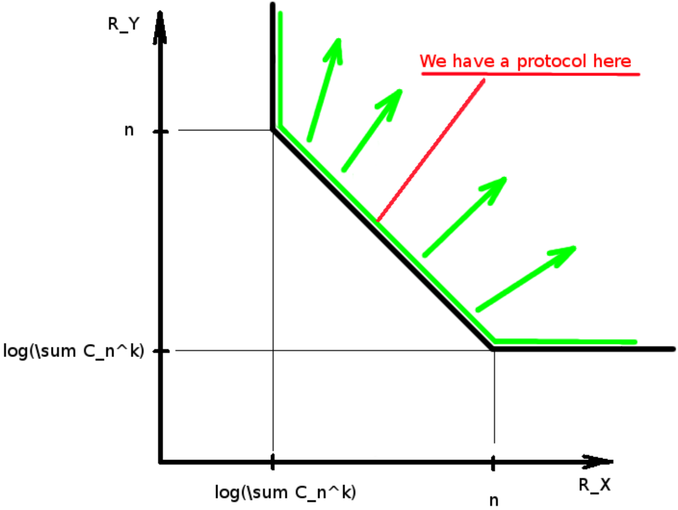
Upper bound:

For every c there exists a constant $d = d(c)$ such that our communication problem can be solved for all pairs (r_A, r_B) satisfying inequalities

$$\begin{cases} r_A + r_B = n + \log_2(\sum_{k=0}^{c} C_n^k) + d, \\ r_A \geq \log_2(\sum_{k=0}^{c} C_n^k), \\ r_B \geq \log_2(\sum_{k=0}^{c} C_n^k). \end{cases}$$

Moreover, there exists a communication protocol with effective (deterministic, polynomial in time) algorithms for all three participants.

Case 1: c is an absolute constant

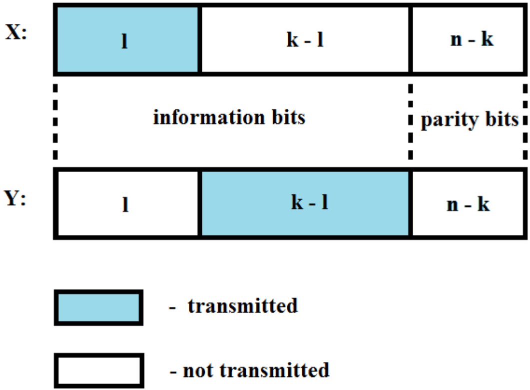


Case 1: c is an absolute constant

Protocol: (for Alice and Bob)

(let H be the parity check matrix of the BCH $(n, k, 2c+1)$-code.)

1. Send some parts of X and Y respectively
2. Send syndromes $H \cdot X^T$ and $H \cdot Y^T$ respectively



Case 1: c is an absolute constant

The total number of transmitted bits:

$n + c\log_2 n + 1 = n + \log_2(\sum_{k=0}^{c} C_n^k) + const(c)$.

Computations: (for Alice and Bob)

All in polynomial time.

Case 1: c is an absolute constant

Protocol: (for Charlie)

1. Recieves all the transmitted information
2. Restores the *error pattern* $E = X \oplus Y$ from the syndromes
3. Reconstructs strings X and Y

Case 1: c is an absolute constant

Problem:

How to restore E?

Case 1: c is an absolute constant

Problem:

How to restore E?

Idea:

1. $HX \oplus HY = H(X \oplus Y) = HE$
2. Find *some* word Z such that $HZ = HE$
3. Find the nearest codeword to $\hat{Z}$ to Z

Case 1: c is an absolute constant

Problem:

How to restore E?

Idea:

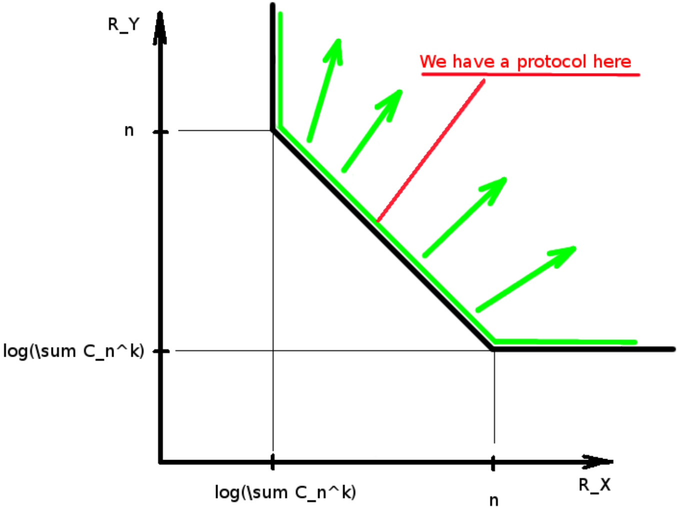
1. $HX \oplus HY = H(X \oplus Y) = HE$
2. Find *some* word Z such that $HZ = HE$
3. Find the nearest codeword to $\hat{Z}$ to Z

Statement:

$\hat{Z} \oplus Z = E$

Case 1: c is an absolute constant

We have an effective protocol that almost matches lower bounds.



Case 2: c is a fraction α of n

Case 2: $c = \alpha n$

Lower bound:

For every communication protocol for our problem

$$\begin{cases} r_A + r_B \geq (1 + h(\alpha))n - o(n), \\ r_A \geq h(\alpha)n - o(n), \\ r_B \geq h(\alpha)n - o(n). \end{cases}$$

where $h(\alpha) = \alpha \log_2 \frac{1}{\alpha} + (1 - \alpha) \log_2 \frac{1}{1-\alpha}$.

Case 2: $c = \alpha n$

Lower bound:

For every communication protocol for our problem

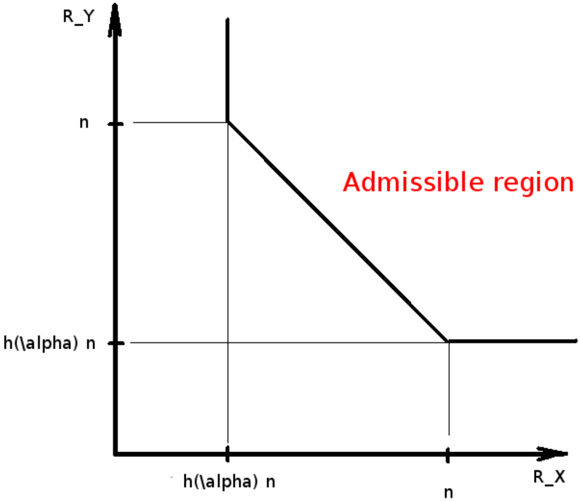
$$\begin{cases} r_A + r_B \geq (1 + h(\alpha))n - o(n), \\ r_A \geq h(\alpha)n - o(n), \\ r_B \geq h(\alpha)n - o(n). \end{cases}$$

where $h(\alpha) = \alpha \log_2 \frac{1}{\alpha} + (1 - \alpha) \log_2 \frac{1}{1-\alpha}$.

Idea of the proof

The same as in the first case, plus $\sum_{k=0}^{\alpha n} C_n^k = 2^{h(\alpha)n + o(n)}$.

Case 2: $c = \alpha n$



In the contrast to the **first case** not all the points satisfying these bounds can be achieved by some (deterministic) protocol.

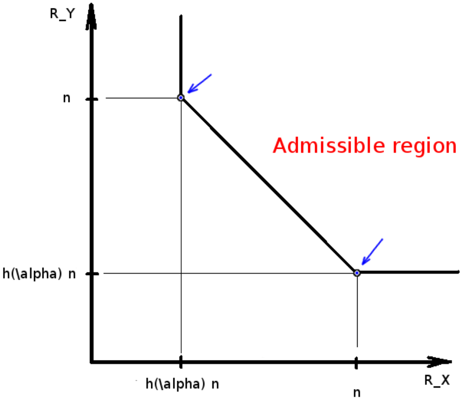
Case 2: $c = \alpha n$

Case 2: $c = \alpha n$

A. Orlitsky, 2003

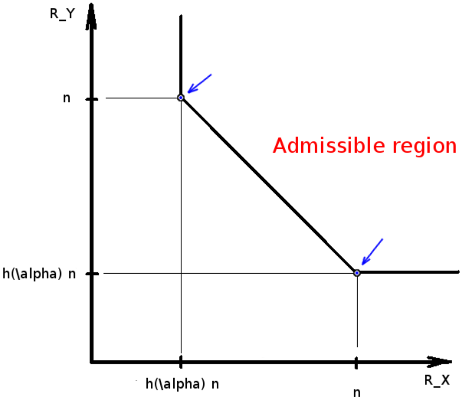
Case 2: $c = \alpha n$

A. Orlitsky, 2003



The pair $(r_A, r_B) = (n, h(\alpha)n)$ is not achievable.

A. Orlitsky, 2003



The pair $(r_A, r_B) = (n, h(\alpha)n)$ is not achievable.
Let's try *probabilistic* protocols!

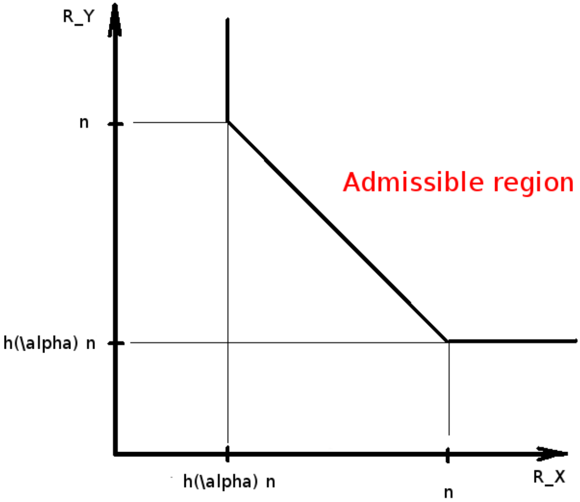
Case 2: $c = \alpha n$, *probabilistic protocols*

Lower bound:

For every $\epsilon < 1/2$ the communication complexity of a probabilistic protocol satisfies

$$\begin{cases} r_A + r_B \geq (1 + h(\alpha))n - o(n), \\ r_A \geq h(\alpha)n - o(n), \\ r_B \geq h(\alpha)n - o(n). \end{cases}$$

Case 2: $c = \alpha n$, *probabilistic protocols*



Case 2: $c = \alpha n$, *probabilistic protocols*

Upper bound:

For every $\epsilon > 0$, for $c = \alpha n$, and all pairs (r_A, r_B) satisfying inequalities

$$\begin{cases} r_A + r_B \geq (1 + h(\alpha))n + o(n), \\ r_A \geq h(\alpha)n + o(n), \\ r_B \geq h(\alpha)n + o(n). \end{cases}$$

there exists a probabilistic protocol for our problem (with an error less than ϵ).

Case 2: $c = \alpha n$, *probabilistic protocols*

Idea:

Case 2: $c = \alpha n$, *probabilistic protocols*

Idea:

1. "Old" protocol

Case 2: $c = \alpha n$, *probabilistic protocols*

Idea:

1. "Old" protocol
2. List-decodable codes (and list-decoding capacity theorem):

Case 2: $c = \alpha n$, *probabilistic protocols*

Idea:

1. "Old" protocol
2. List-decodable codes (and list-decoding capacity theorem):
3. Random hashing

Case 2: $c = \alpha n$, *probabilistic protocols*

Sending part modernization:

1. According to list-decoding capacity theorem participants chose a list-decodable code that corrects up to αn errors with the list's length L.
2. Alice and Bob chose in advance an integer $D = \frac{nL^2}{\epsilon} \log_2 \frac{nL^2}{\epsilon} + o(\frac{nL^2}{\epsilon})$
3. Sending to Charlie the results of applying hash-functions based on random primary numbers from $[1, D]$ and the prime numbers themselves

Case 2: $c = \alpha n$, *probabilistic protocols*

Recieving part modernization:

1. Charlie restores a $constant(L)$-length list of possible pairs $(\hat{X}, \hat{Y})$ and apply hash-functions to each pair
2. He mathes the results with hash-results he recieved from Alice and Bob
3. Take the first matched pair $(\bar{X}, \bar{Y})$
4. $Prob[(\bar{X}, \bar{Y}) \neq (X, Y)] < \epsilon$

Case 2: $c = \alpha n$, *probabilistic protocols*

The disadvantage of this protocol is that the participants need to perform *exponentially* long computations.

History of the problem:

1. *Probabilistic formulation:* S-W theorem, 1973
2. *Probabilistic formulation:* A protocol for any arbitrary constant c, 2005
3. *Combinatorial formulation:* Constant fraction α: lower bounds and the solving probabilistic protocol, 2013

Open questions:

Open questions:

1. Can we generalize Orlitsky's result?

Open questions:

1. Can we generalize Orlitsky's result?
2. An *effective* (randomized) polynomial protocol for $c = \alpha n$?

Thank you for your attention!

Questions?