Some Applications of Coding Theory in Computational Complexity

Luca Trevisan

U.C. Berkeley

Three Themes

- Error-correcting codes
- Average-case complexity and cryptography
- Probabilistically Checkable Proofs and approximation

Error-Correcting Codes

Motivations

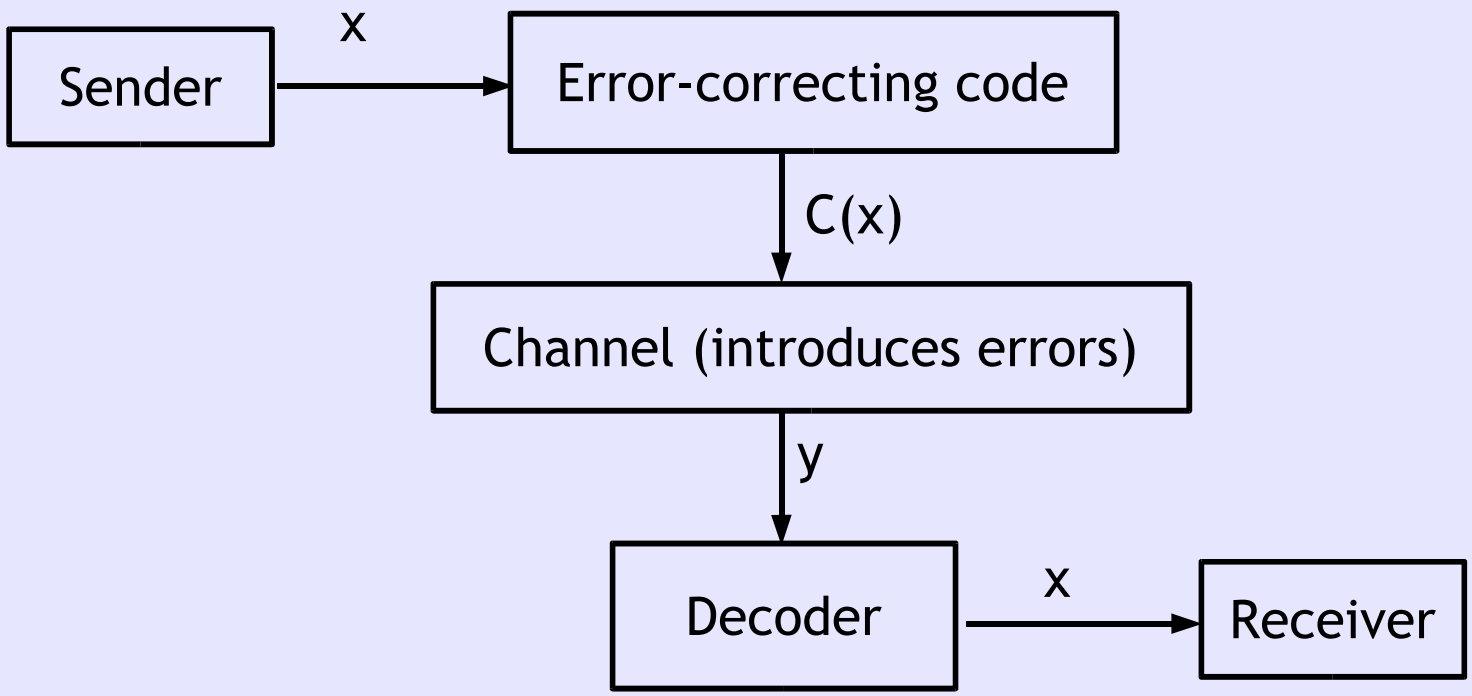
- Transmission of data over a noisy channel
- Storage of data on faulty device

Error correcting code:

map C: $\{0,1\}^k$ -> $\{0,1\}^n$ $n>k$

s.t., if $x \neq z$, then C(x) and C(z) differ a lot

Error-Correcting Codes



- Decoder: on input y, find x s.t. C(x) and y are closest

Average-Case Complexity

Motivation:

- Relation between worst-case complexity and average-case complexity

Application

- Pseudorandom generators, derandomization

Example of Desired Result:

- If EXP not solvable by sub-exponential size circuits
- Then EXP contains a problem such that all sub-exponential size circuits make mistakes on ~ ½ fraction of inputs

EXP:= decision problems solvable in $2^{\text{poly } n}$ time

Average-Case Complexity

Example of Desired Result:

- If EXP not solvable by sub-exponential size circuits
- Then EXP contains a problem such that all sub-exponential size circuits make mistakes on ~ ½ fraction of inputs

Proof Sketch

- L: problem in EXP not solvable by sub-exponential circuits
- Define new problem L', "encoding" of L
- Suppose C' solves L' on >> ½ fraction of inputs
- Devise a reduction transforming C' into circuit C that solves L on all inputs

Average-Case Complexity

Proof Sketch

- L: problem in EXP not solvable by sub-exponential circuits
- Define new problem L', "encoding" of L
- Suppose C' solves L' on >> ½ fraction of inputs
- Devise a reduction transforming C' into circuit C that solves L on all inputs

Relation with codes

- L ~ message
- L' ~ encoding
- C' ~ corrupted codeword
- C ~ decoding
- Reduction ~ decoder

Probabilistically Checkable Proofs

Def: a language L is in NP if there are polynomial-length polynomial-time checkable certificates of statements "x is in L"

PCP Theorem: for every NP problem (e.g. SAT), a certificate w of "x is in L" can be checked by reading O(1) bits of w and using O(log n) random bits

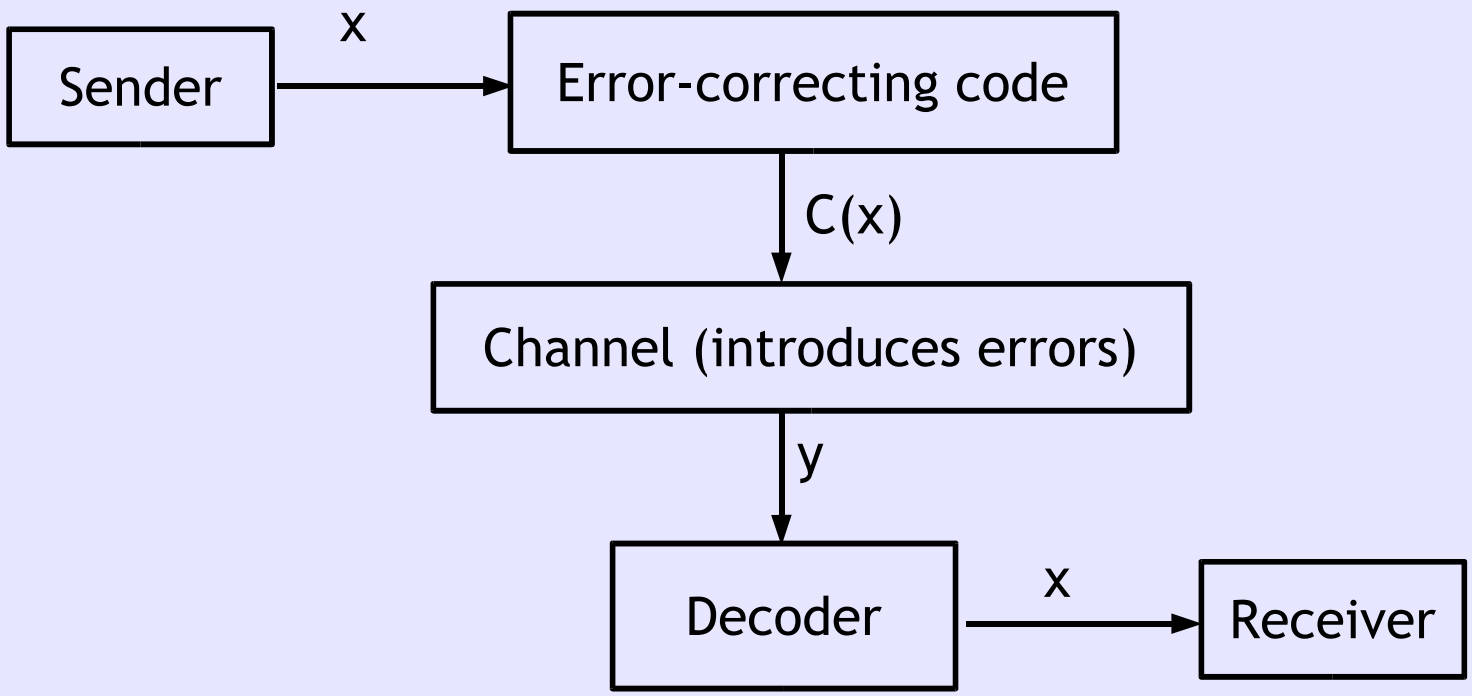
Relation with codes:

PCP certificate ~ encoding of standard certificate

Verification ~ distinguish valid codewords from strings far from code

Error-correcting Codes

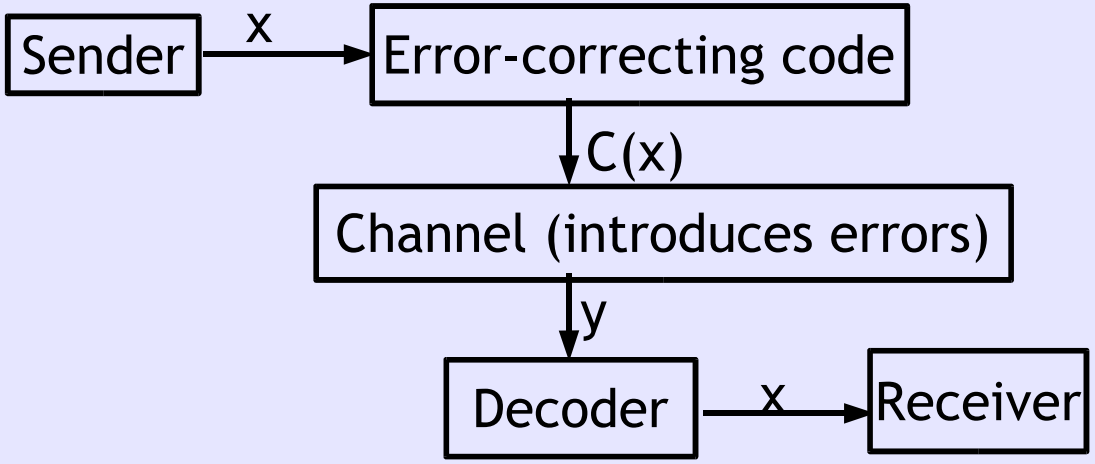
Error-Correcting Codes



- Decoder: on input y, find x s.t. C(x) and y are closest

Error-correcting Codes

- Suppose channel introduces $\leq e$ errors
- Suppose for every x,z, $C(x)$ and $C(z)$ differ in at least $2e+1$ entries
- Then decoder can correctly find x given the corrupted string y



Parameters of Interest

$C: \{0,1\}^k \rightarrow \{0,1\}^n$ (or $C: \Sigma^k \rightarrow \Sigma^n$)

- k: dimension (message length)
- n: block length (encoding length)
- d:= Hamming distance between closest pair of codewords
- Want to maximize:
  - k/n: rate
  - d/n: relative minimum distance

Some Limitations

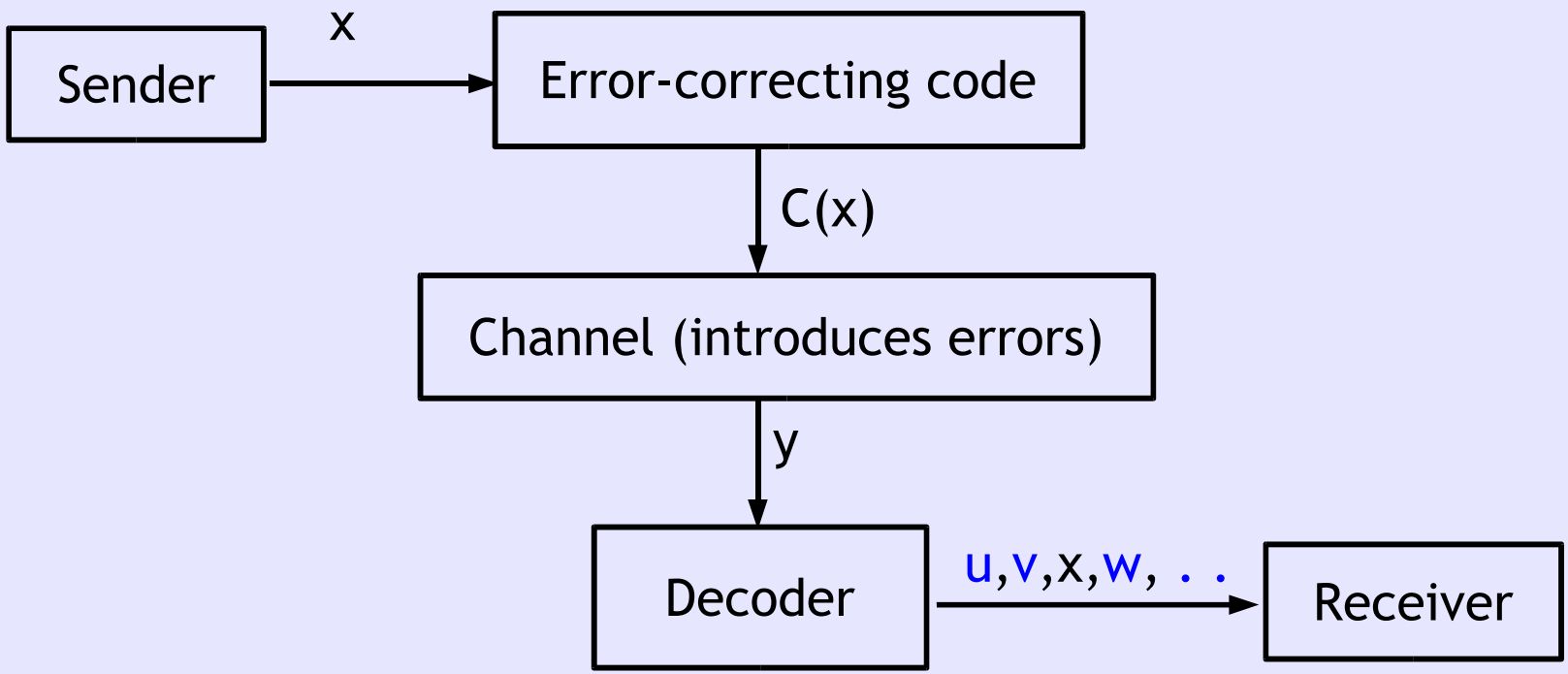
C: $\{0,1\}^k \rightarrow \{0,1\}^n$

- If $k/n \rightarrow 1$ then $d/n \rightarrow 0$
- If $d/n \rightarrow ½$ then $k/n \rightarrow 0$
- $d < n/2$
- Number of errors that can be corrected $< n/4$

C: $\Sigma^k \rightarrow \Sigma^n$

- If $k/n \rightarrow 1$ then $d/n \rightarrow 0$
- If $d/n \rightarrow 1-1/|\Sigma|$ then $k/n \rightarrow 0$
- $d < (1-1/|\Sigma|) * n$
- Number of errors $< (½ - 1/2|\Sigma|) * n$

List Decoding



List Decoding

One extra parameter: size of list

$C: \{0,1\}^k \rightarrow \{0,1\}^n$

- Possible to tolerate $(\frac{1}{2} - \varepsilon)n$ errors with $k/n \geq \text{poly}(\varepsilon)$, list size $\leq \text{poly}(1/\varepsilon)$ (versus $\leq \frac{1}{4}$ n errors in unique decoding)

$C: \Sigma^k \rightarrow \Sigma^n$

- Possible to tolerate $(1-1/|\Sigma| - \varepsilon)n$ errors (versus $< \frac{1}{2}$ n errors in unique decoding)

"Hadamard" Code

H: $\{0,1\}^k \to \{0,1\}^n$ with $n=2^k$

- Let $a_1,\ldots,a_n$ be an enumeration of $\{0,1\}^k$
- Then x is encoded as
  $H(x) := (a_1*x , a_2*x , \ldots , a_n*x)$

$a*b := a[1]b[1] + a[2]b[2] + \ldots + a[k]b[k] \bmod 2$

- Minimum distance: n/2

- Message ~ linear function $\{0,1\}^k \to \{0,1\}$
- Encoding ~ evaluation of function over $\{0,1\}^k$

Example k=3

Encoding of…				is…							
0	0	0		0	0	0	0	0	0	0	0
0	0	1		0	1	0	1	0	1	0	1
0	1	0		0	0	1	1	0	0	1	1
0	1	1		0	1	1	0	0	1	1	0
1	0	0		0	0	0	0	1	1	1	1
1	0	1		0	1	0	1	1	0	1	0
1	1	0		0	0	1	1	1	1	0	0
1	1	1		0	1	1	0	1	0	0	1

Reed-Muller Codes

RM: $\mathbf{F}^k \to \mathbf{F}^n$ $|\mathbf{F}| := q$, $\mathbf{F}$ finite field, $n = \mathbf{F}^m$

- Message ~ m-variate polynomial of degree t
  (choose t so there are $|\mathbf{F}^k|$ such polynomials)
- Encoding ~ evaluation of polynomial over $\mathbf{F}^m$
- Minimum distance ~ $1 - t/|\mathbf{F}|$

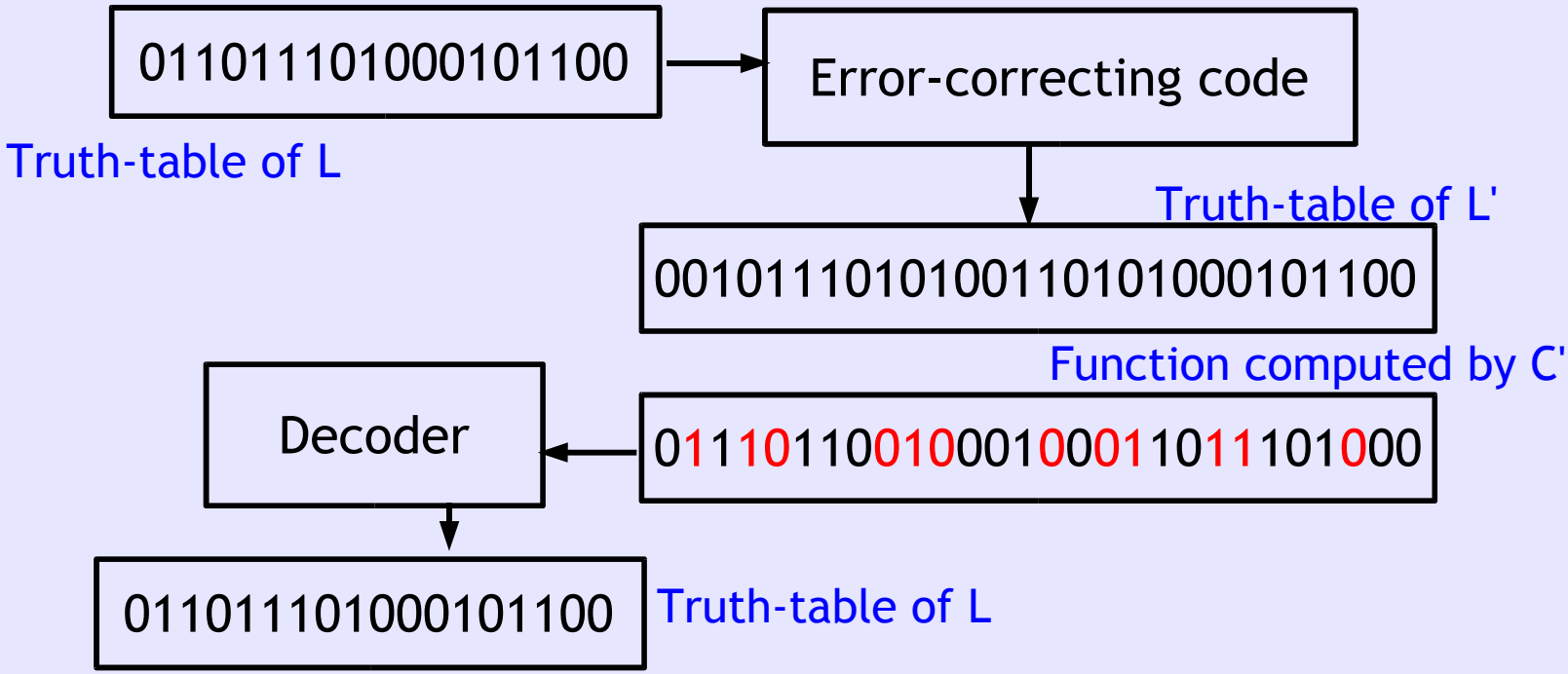
- m=k, t=1: Hadamard code
  Binary alphabet, bad rate
- m=1, t<q, n<q: Reed-Solomon code
  very good rate, large alphabet

Average-Case Complexity and Locally Decodable Codes

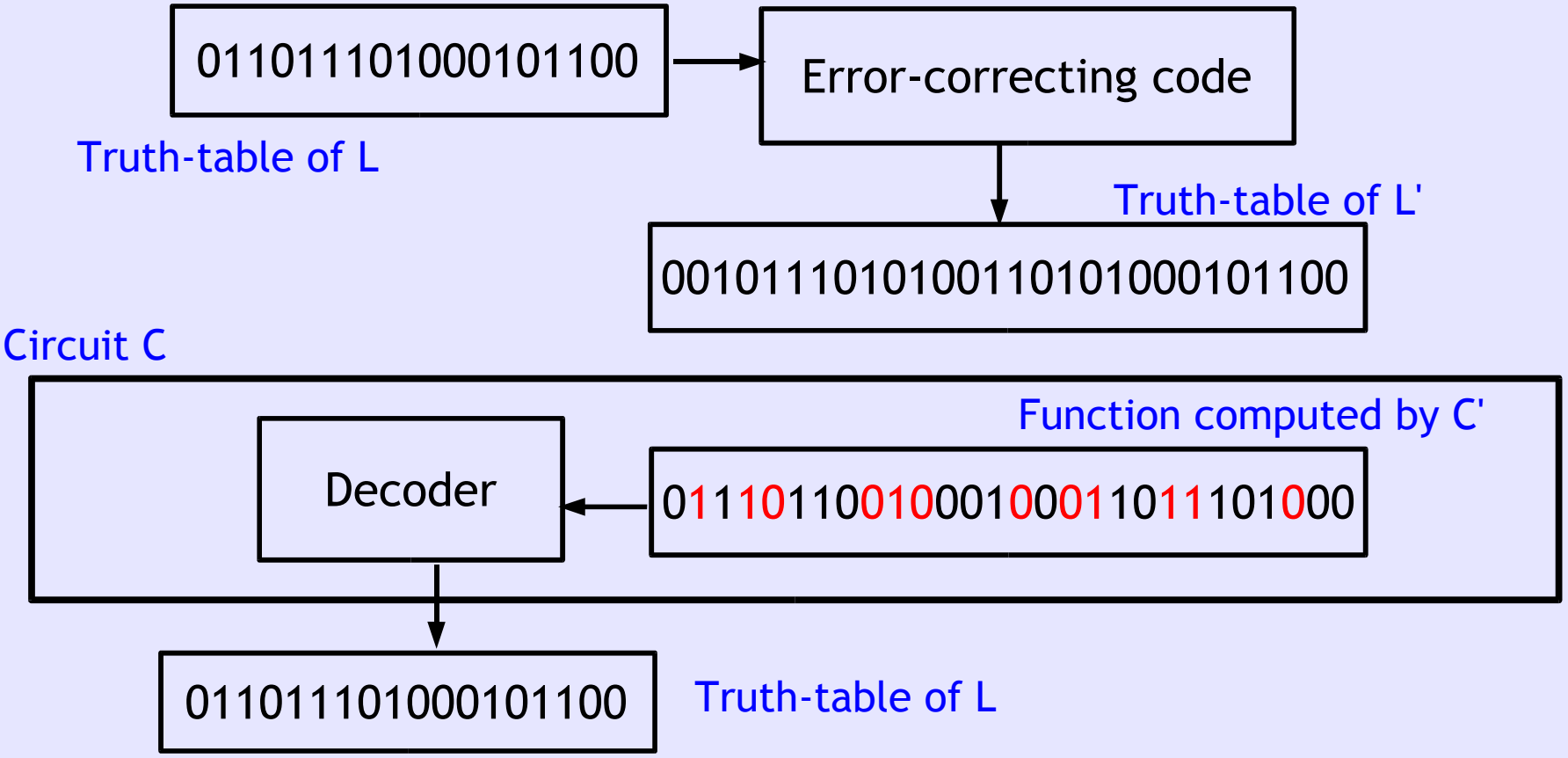
Average-Case vs Worst-Case

Example of Desired Result [IW'97,STV'99]:

- If EXP contains language L not solvable by sub-exp size circuits
- Then EXP contains language L' such that all sub-exp size circuits make mistakes on ~ ½ fraction of inputs

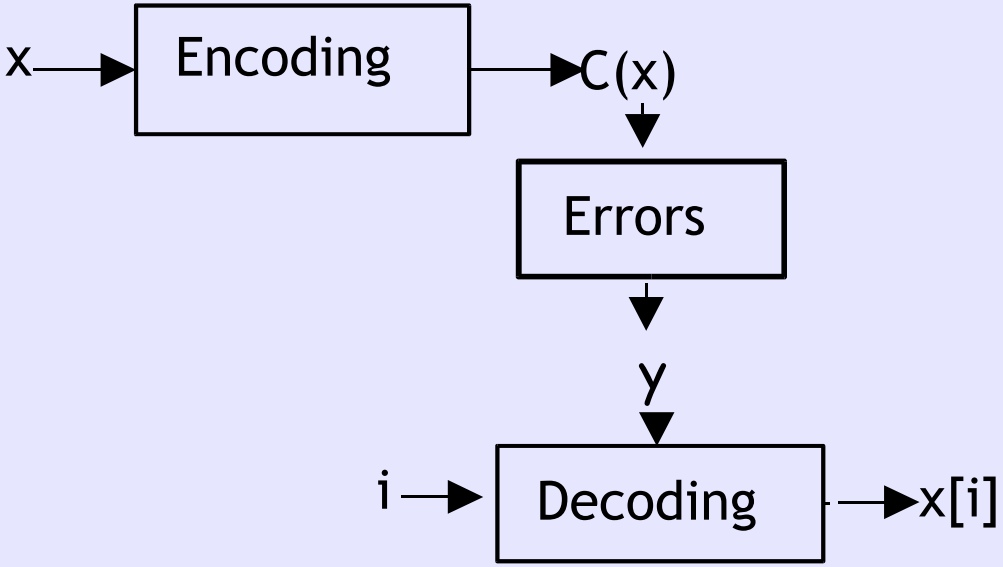


Average-Case vs Worst-Case



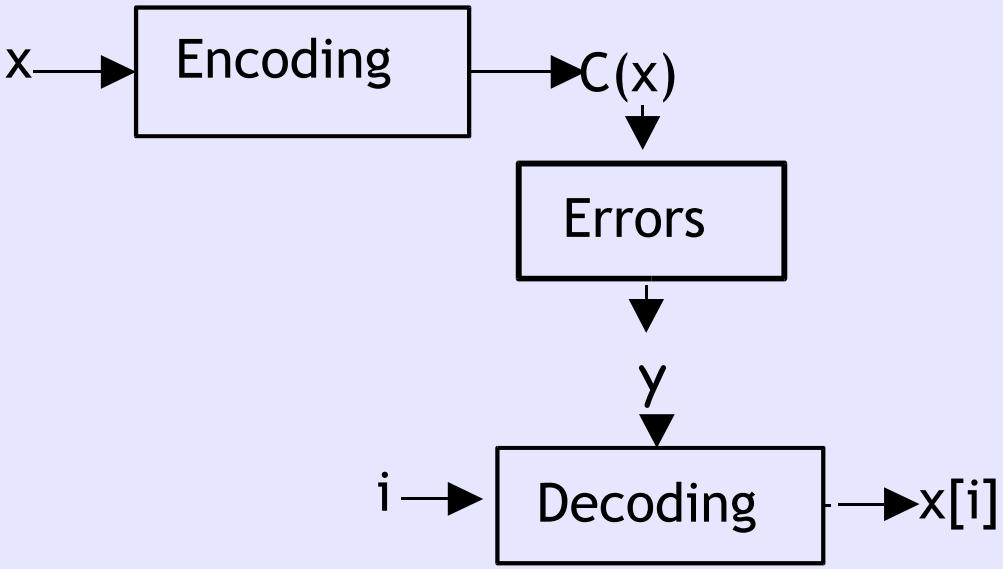
Sublinear Error-correction

- Decoder must work in time sub-linear in length of codeword
- Possible because we only want to decode one bit of message



Locally Decodable Codes

- In addition to all other parameters, we focus on complexity of decoder
- Proxy measure: query complexity



Decoder for Hadamard Code

- Suppose y in $\{0,1\}^n$ differs from H(x) in $< \delta n$ entries for some x
  - Then for all but $< \delta n$ vectors a, y[a]=a*x

- Given i, want to reconstruct x[i]
  - Pick at random a
  - Output $y[a+e_i] - y[a] \bmod 2$

Decoder for Hadamard Code

- Given i, want to reconstruct xi
  - Pick at random a
  - Output $y[a+e_i] - y[a] \bmod 2$

- Analysis [Blum-Luby-Rubinfeld]:
  - a, $a+e_i$ uniformly distributed
  - $(a+e_i)*x - a*x = e_i*x = x[i]$
  - $\Pr[y[a+e_i]=(a+e_i)*x \text{ AND } y[a]=a*x] > 1-2\delta$
  - $\Pr[\text{ algorithm outputs } x[i]] > 1-2\delta$

Decoding Reed-Muller Codes

Due to various authors
[L90,BF90,GLRSW91,BFNW93...]

Possible to set parameters to have
RM:$\{0,1\}^k$ -> $\{0,1\}^n$ with n=poly(k)
that is locally decodable with polylog n queries
after a ¼ - o(1) fraction of errors.

Open Question: n=O(k), polylogn queries,
constant fraction of errors. (My guess: no)

List-Decoding

Hadamard Code [Goldreich-Levin'89]

- Decoder runs in poly(k) = polylog n time,corrects ½ -o(1) fraction of errors
- Applications to cryptography, learning (!), ...

Reed-Muller Codes [Sudan-T-Vadhan'99]

- Construction with n=poly k, decoder works in polylog k = polylog n time, corrects ½ -o(1) fraction of errors
- Application to pseudorandomness, derandomization

Private Information Retrieval

Private Information Retrieval [CGKS'95]

- A cryptographic protocol:
  - User retrieves information from a server
  - Server does not know what information was retrieved by a user

  (Need server replication, . . .)
- Equivalent to construct locally decodable codes
- Interesting case: very small query complexity (2, 3, . . .). Large alphabet ok

2 Queries: Hadamard Code

$H: \{0,1\}^k \to \{0,1\}^n \quad n=2^k$

- Decoder makes 2 queries
- If fraction of errors < δ, then error prob < 2δ

Lower bounds

- If $C: \{0,1\}^k \to \{0,1\}^n$ is linear, decoder makes 2 queries, corrects $\Omega(1)$ frac of errors, error prob < $\frac{1}{2}-\Omega(1)$, then $n = 2^{\Omega(k)}$ [GKST, combinatorial]
- Same for general codes [KdW, quantum IT]

(Hadamard code best possible for 2-query, binary)

2 Queries, Larger Alphabet

$C: \{0,1\}^k \rightarrow (\Sigma)^n$

Construction:

- $n = \exp(O(k^{1/3}))$, $|\Sigma| = \exp(O(k^{1/3}))$ [CGKS'95]

Negative results:

- $n > \exp(\Omega(k/\Sigma))$ [GKST'02,KdW'03]
- $n > \Omega(k^2/\log |\Sigma|)$ [KT'99]

Open Question (one of many):

- Possible n=poly k, $|\Sigma|$= poly k? (probably not)

3 Queries, Binary Alphabet

$C: \{0,1\}^k \rightarrow \{0,1\}^n$

Construction:

- $n = \exp(O(k^{1/3}))$ [CGKS'95]

Negative results:

- $n > \Omega(k^2)$ [KdW'03]

Open Question (one of many):

- Possible n=poly k? (probably not)

3 Queries, Binary Alphabet

C: $\{0,1\}^k$ -> $\{0,1\}^n$ decodable with 3 queries

- Possible n=poly k? (probably not)

If C linear, equivalent to following question:

- Let $v_1, \ldots, v_n$ be vectors in $\{0,1\}^k$ such that for every i=1,...,k there are $\Omega(n)$ disjoint triples (a,b,c) such that

$$v_a + v_b + v_c = e_i$$

Is it possible that n=poly(k)?

q Queries, Binary Alphabet

$C: \{0,1\}^k \to \{0,1\}^n$

Construction:

- $n = \exp(O(k^{\log q/(q \log\log q)}))$ [BYKR'02]

Negative results:

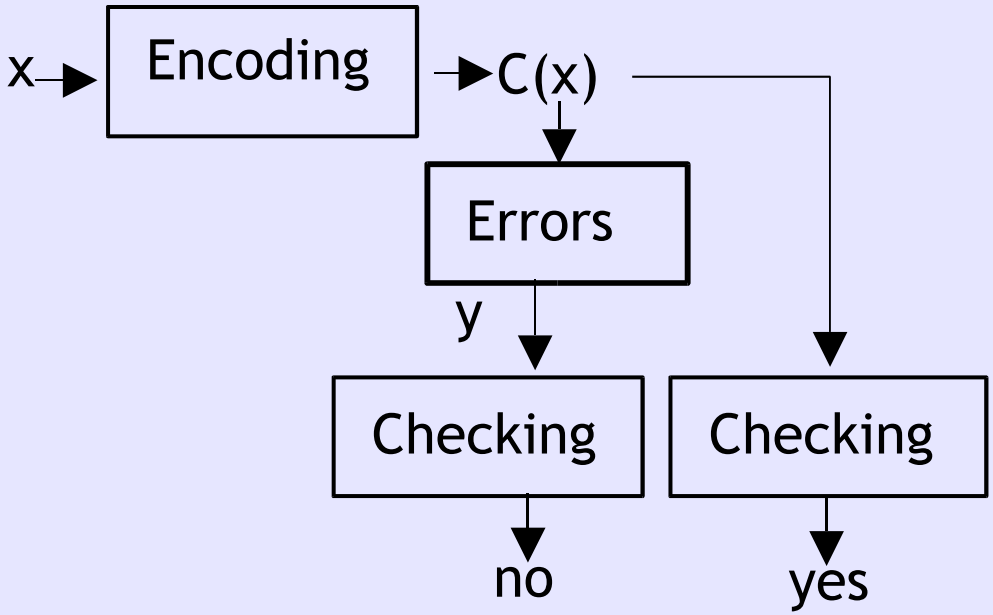
- $n > \Omega(k^{1+1/(q-2)})$ [KT'00,KdW'03]

Difficult to even conjecture right asymptotic.

$\exp(k^{\log q /q})$?

Locally Testable Codes and Probabilistically Checkable Proofs

Local Testability



Want low-complexity checking procedure.
Parameter of interest: query complexity

Tester for Hadamard Code

Algorithm [Blum-Luby-Rubinfeld]

- Pick a,b at random
- Accept if y[a]+y[b]=y[a+b]

Analysis

- [trivial] If y=H(x) for some x,
  then y[a]=x*a for every a, accept w/ prob 1
- [BLR] If y is δ-far from H(x) for all x,
  then test rejects w/ prob $\Omega(\delta)$
- [BCKHS] If ... then ... $\geq \delta$

Other Reed-Muller Codes

RM: $\mathbf{F}^k \to \mathbf{F}^n$ $|\mathbf{F}| := q$, $\mathbf{F}$ finite field, $n = \mathbf{F}^m$

- Message ~ m-variate polynomial of degree t (choose t so there are $|\mathbf{F}^k|$ such polynomials)
- Encoding ~ evaluation of polynomial over $\mathbf{F}^m$
- Minimum distance ~ $1 - t/|\mathbf{F}|$

- Testable with query complexity t+1 [GLRSW]
- Impossible to test with query complexity $\leq$ t
- Roughly, $n = \exp(k^{1/t})$. Impossible to have low query complexity and good rate with RM codes.

The "Bivariate Code"

- A degree-d bivariate polynomial **p:F x F -> F** can be represented as **2|F|** univariate degree-d polynomials (the "rows" and the columns")

$2x^2 + xy + y^2 + 1 \bmod 5$

1	2	0	0	2
3	0	4	0	3
4	2	2	4	3
4	3	4	2	2
3	3	0	4	0

Y^2+1	Y^2+y+3	Y^2+2y+4	Y^2+3y+4	Y^2+4y+3	$2x^2+1$	$2x^2+x+2$	$2x^2+2x$	$2x^2+3x$	$2x^2+4x+2$

Bivariate Low-Degree Test

- Pick a random row and a random column. Chek that they agree on intersection
- If $|F|$ is a constant factor bigger than d, then rejection probability is proportional to distance from code

  [Arora-Safra], [ALMSS],

  [Polishuck-Spielman]
- Only 2 queries
- Huge alphabet

Other Constructions

C: $\{0,1\}^k \to \{0,1\}^n$ tester that makes q queries

- . . .
- [Goldreich-Sudan'02]
  $n = k \cdot \exp((\log n)^{1/2})$ $q=O(1)$, non-constructive
- [BSSVW'03,BGHSV'04]
  $n = k \cdot \exp((\log n)^{\varepsilon})$ $q=O(1)$
- [BS'05,D'05] $n=k \cdot \text{polylog}(k)$, $q=\text{polylog } n$ (?)

- Recall: for decoding with q queries,
  $n = \Omega(k^{1+1/(q-2)})$

Main Open Question

C: $\{0,1\}^k \to \{0,1\}^n$ tester that makes q queries

- Is n=O(k), q=O(1) possible?
  For linear codes? q=3, n < 10k?

- Recall: for decoding with q queries,
  $n = \Omega(k^{1+1/(q-2)})$

In Summary (Decoding)

- Codes with sub-linear time decoders
  Applied to:
  - Average-case complexity/derandomization
  - Private information retrieval

  Current knowledge:
  - Good constructions with polylog k runnig time. (Average-case complexity)
  - k=2 case well understood (not for large alphabet)
  - Exponential length constructions k=O(1)
  - k>2 only slightly superlinear lower bounds

In Summary (Testing)

- Codes with sub-linear time testers. Applied to:
  - Probabilistically Checkable Proofs and approximability of optimization problems

  Current knowledge:
  - Good constructions with nearly linear encoding length
  - No lower bounds of any kind

Final Thoughts

Why so much work on these questions?

- Decoding
  Simple questions, surprisingly difficult to answer
  Something deep to be discovered while exploring these questions?

- Testing
  Probabilistically Checkable Proofs important and difficult
  Studying testable codes has helped to better understand and simplify PCP constructions