Coding Theory: Tutorial/Survey

Madhu Sudan
MIT

FOCS 2001,
October 14-17, 2001

Coding Theory

Basic goal:
Communication in the presence of noise.

In intersection of:

1. Statistics: What is information?

2. Combinatorics: What structures preserve information despite noise?

3. Algorithms & Complexity: How to deal with computational problems efficiently?

Information theory: Deals with (1).

Coding theory: deals with (2) and (3).

History

Origins

- Two seminal works:
  - [Shannon]: A Mathematical Theory of Communication.
  - [Hamming]: Error-detecting and error-correcting codes.

- Both went way beyond the immediate motivations and examined far-reaching subjects. (Shannon more so than Hamming?)

Development

Great progress over fifty+ years. Some sample results:

- 1950-1960: First families of codes. Algebraic coding theory.
  - Reed-Muller Codes.
  - Reed-Solomon Codes.
  - BCH Codes.
- 1960-1970: Algorithmic focus intensifies.
  - Peterson. Berlekamp-Massey.
  - Gallager - LDPC codes.
  - Forney - Concatenated codes.

Development (contd.)

- 1970-1980: Deep theories.
  - Linear Programming bound.
  - Lovasz on Shannon Capacity.
  - Justesen's codes.
- 1980-1990: Algebraic-Geometry codes. (started in mid 70's by Goppa). Better than random!
- 1990-2001: Algorithms:
  - Linear time decoding.
  - Approaching Shannon capacity in practice.
  - List-decoding.

Outline

- First part:
  - Shannon & Hamming.
  - Code constructions.
  - Bounds on performance.
- Second part:
  - Decoding algorithms.
  - Connections to CS.

Shannon's Theory

Model

- Noisy Channel modelled stochastically.
  - Input alphabet Σ, Output Γ.
  - Symbol transition - arbitrary distribution.
  - Correlations of errors: Independent, or Markovian.
  - Formally $P_1, \ldots, P_l$ are $\Sigma \times \Gamma$ stochastic matrices, with $l \times l$ Markov Chain M on states.
- Sender encodes $E : \Sigma^k \to \Sigma^n$,
- Receiver decodes $D : \Gamma^n \to \Sigma^k$.
- k/n is rate of transmission.

Main Results

Notions:

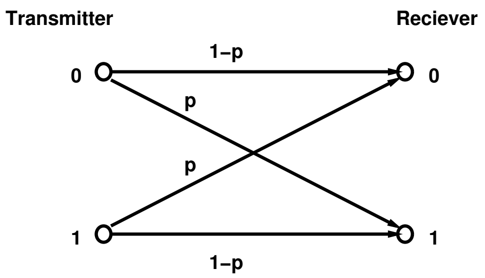
- Capacity of channel:
  Real number; function of $P_1, \ldots, P_l, M$.
- Probability of error:

$$\Pr_{m,\mathrm{Channel}} [D(\mathrm{Channel}(E(m))) \neq m]$$

Coding Theorem: Can transmit at rate $<$ capacity, while getting exponentially small error probability.

Converse Coding Theorem: Can't transmit at rate $>$ capacity.

Example: Binary Symmetric Channel



BSC(p):

- $\Sigma = \Gamma = \{0, 1\}$.
- Bits flipped independently w. probability p.
- Capacity $= 1 + p \log_2 p + (1-p) \log_2 (1-p)$.

Highlights

1. Stochastic modelling.
2. Highly non-constructive.
   Pick E at random.
   D is exhaustive search.
3. Focus on fleshing out mathematical underpinnings.
   Not "instant gratification".
4. In particular, note the "converse".
5. No mention of codes.

Hamming Theory

Model

- Adversary introduces upto t errors.
  - Can you detect errors?
  - Can you locate errors?

Definitions

- Hamming Distance on Σ^n.
  $\Delta(\mathbf{x},\mathbf{y}) = |\{i \in [n] | x_i \neq y_i\}|$.
- Code $C \subseteq \Sigma^n$.
- Distance of Code: $\Delta(C) = \min_{\mathbf{x}\neq\mathbf{y}\in C}\{\Delta(\mathbf{x},\mathbf{y})\}$.
- If $|\Sigma| = q$, $|C| = q^k$ and $\Delta(C) = d$
  then C is $(n,k,d)_q$ code.
- If $\Sigma = \mathbb{F}_q$ and C is linear subspace, then C is $[n,k,d]_q$ code.

Notation

- Hamming weight:
  $\mathrm{wt}(\mathbf{x}) = \#$ non-zero coordinates of $\mathbf{x}$.
- Hamming ball:
  $B(\mathbf{x},r) = \{\mathbf{y} \in \Sigma^n | \Delta(\mathbf{x},\mathbf{y}) \leq r\}$.
- Volume $\mathrm{Vol}_q(r,n) = |B(\mathbf{0},r)|$ in Σ^n,
  $q = |\Sigma|$. Omit subscript if $q = 2$.
- Fact: $\mathrm{Vol}(\delta n, n) \approx 2^{H(\delta)n}$ where
  $H(\delta) = -(\delta\log_2\delta + (1-\delta)\log_2(1-\delta))$.

Hamming's contributions

- Hamming Bound: $\exists[n,k,d]$-code $\Rightarrow$ $2^k\mathrm{Vol}((d-1)/2,n) \leq 2^n$.
- Hamming code: Codes with $d = 3$ meeting above bound.
- Also
  - Codes with $d = 1, 2, 4$.
  - Role of linearity.
  - Definition of model!

Shannon vs. Hamming

- Concurrent works!
- Non-constructive vs. Constructive.
- Probabilistic vs. Adversarial.

(Standard) Terminology

If $\mathcal{C} = (n, k, d)_q$ code, then:

- n = Block Length.
- k = Message Length.
- d = Distance.
- k/n = (Information) Rate.
- d/n = Distance (Rate).
- q = (Alphabet size).

(Words within parenthesis often omitted.)

Basic questions in Coding theory

- Given n, k, q, find $(n, k)_q$ code $\mathcal{C}$ that maximizes $\Delta(\mathcal{C})$.
- Given n, d, q, find $(n, k)_q$ code $\mathcal{C}$ with $\Delta(\mathcal{C}) \geq d$ that maximizes k.
- Given k, δ, q, find $(n, k)_q$ code $\mathcal{C}$ with $\Delta(\mathcal{C}) \geq \delta n$ that minimizes n. (Better phrasing of algorithmic question.)
- Given n, k, d, find $(n, k)_q$ code $\mathcal{C}$ with $\Delta(\mathcal{C}) \geq d$ that ???imizes q. (Actually a very nice perspective.)

Code constructions

Reed-Solomon Codes

Reed-Solomon Codes:
$[n, k, n-k+1]_q$ code for $q \geq n$.

- Fix distinct $x_0, \ldots, x_{n-1} \in \mathbb{F}_q$.
- Message: Coefficients of polynomials $\langle m_0, \ldots, m_{k-1} \rangle \cong M(x) = \sum_{i=0}^{k-1} m_i x^i$
- Encoding: Evaluations of polynomials $\langle M(x_0), \ldots, M(x_{n-1}) \rangle$
- Distance follows from fact that # of roots of a non-zero poly $\leq$ its degree.

Reed-Muller Codes

Codes from m-variate polynomials.

Degree $\leq r$.

CS version: $q > r$, $[n, k, d]_q$ code

$$n = q^m; k = \binom{m+r}{r}; d = q^m - rq^{m-1}$$

Notes:

- Distance from DeMillo-Lipton/Schwartz/Zippel Lemma.
- Smaller alphabet than Reed-Solomon.
- Poorer rate.

Hadamard codes

- Reed-Muller codes of degree 1.
- Can pick $q = 2$!
- Gives $[2^m, m+1, 2^{m-1}]_2$-code.

Very special objects.

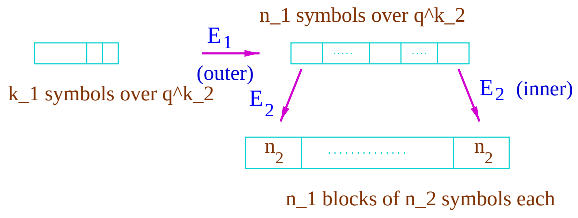
Summary

- Reed-Solomon codes have optimal tradeoff between distance and rate, but alphabet is large.
- Hadamard codes have maximum possible distance for binary code, but rate is terrible.
- Random linear codes: can get positive rate and positive relative distance simultaneously over any alphabet, but not verifiable/decodable (in poly time).

Concatenated codes [Forney]

To get explicit constructions:

Let $C_1 = [n_1, k_1, d_1]_{q^{k_2}}$-code and $C_2 = [n_2, k_2, d_2]_q$-code.



$[n_1, k_1, d_1]_{q^{k_2}} \circ [n_2, k_2, d_2]_q$
$\quad \to \quad [n_1 n_2, k_1 k_2, d_1 d_2]_q$.

- [Forney] RS $\circ$ Random Code
- [Justesen] RS $\circ$ Random Ensemble

Others

Many other codes in literature.

Hamming Already mentioned.

BCH Extend Hamming codes to $d = 5$ etc. Generalize and specialize Reed-Solomon codes!

Algebraic-Geometry codes Another generalizatio of RS codes. Do better than random for large constant alphabet sizes.

Graph-based codes Started with Gallager. Continued with Tanner, Spielman (+ Sipser), Luby et al. Practical codes with fast decoding.

Limitations

Upper bounds on rate

- An extremely elegant stream of work in coding theory is techniques used to bound the number of codewords that a code of a give distance must have.
- Started with Hamming's & Shannon's works.
- Famed bounds: Hamming, Plotkin, Singleton, Elias-Bassalygo, LP bound.
- Related works: Johnson bound, MacWilliams Identities, Non-existence of perfect codes, Bounds on Self-dual codes, Lovasz's theta function, Zeta-function on codes.

Hamming bound

Simple packing bound.

Observation: $(n, k, d)_2$ code corrects $(d-1)/2$ errors, i.e., Balls of radius $(d-1)/2$ around codewords are disjoint.

Conclusion:

$$2^k \mathrm{Vol}((d-1)/2, n)) \leq 2^n.$$

Elias-Bassalygo bound

Defn: Code C is (t, ℓ)-error-correcting if $\forall \mathbf{x} \in \Sigma^n$, there exists at most ℓ codewords $\mathbf{c} \in C$ with $\Delta(\mathbf{x}, \mathbf{c}) \leq t$.

Hamming Observation: $[n, k, d]$-code is $(t, 1)$-error-correcting.

Generalized Conclusion: If $[n, k, d]$-code is (t, ℓ)-error-correcting then

$$2^k \cdot \mathrm{Vol}(t, n) \leq \ell \cdot 2^n$$

Hope: Can use $t > d/2$ while fixing ℓ to constant or polynomial in n.

Johnson bound: $[n, k, d]$-code is (t, n)-error-correcting for $t = \frac{n}{2}(1 - \sqrt{1 - 2d/n})$.

LP bound

- Elias-Bassalygo bound has right endpoints. Implies Rate = 0, if Rel. Distance $\geq \frac{1}{2}$.
- Behavior at rate = 1, is also correct. But not at Rel. Dist. $= \frac{1}{2}$.
- Better bound: the LP bound. Gets this behavior right. Best known bound to date. Deep result.

Algorithmic questions

Basic tasks

Will stick to linear codes.

Encoding: Given message in $\mathbb{F}_q^k$, compute its encoding in $\mathbb{F}_q^n$.

- Easy, in $O(n^2)$ time.
- In linear time? [Spielman].

Decoding: Informally, given corrupted codeword, compute message.

- For any reasonable formalism, becomes a search problem - brute force takes exponential time.

This talk: Stick to decoding.

Formalizing Decoding

- Code - part of input or fixed?
  - Part of input: No positive results known.
  - Fixed & well-known: $\exists$ Positive results.
- What defines $\mathbf{m}$?
  - Most likely under some distribution on errors - Maximum Likelihood Decoding.
  - Most likely under q-ary symmetric channel - Nearest Codeword Problem.
- How many errors t?
  - Unlimited: Problem seems hard.
  - $t \leq d/2$: Unambiguous decoding.
  - $t > d/2$, but report all codewords in Hamming ball of radius t: List decoding.

Algebraic decoding

Reed-Solomon decoding

Given:

- $x_1, \ldots, x_n \in F$ distinct.
- $r_1, \ldots, r_n \in F$.
- Integers k, e

Task: Find a poly p of deg. $k-1$ s.t.
$p(x_i) \neq r_i$
for at most e values of $i \in \{1, \ldots, n\}$.

Decoding Reed Solomon Codes

[Peterson60, Berlekamp66, Massey66]
[Welch-Bkmp86, Gemmell-S.92]

Key concept: Error locator polynomial

$$Y(x) \text{ s.t. } Y(x_i) = 0 \text{ if } p(x_i) \neq r_i$$

1. Y has low-degree ($\leq e$)

2. $Z = Y.p$ has low-degree ($\leq e + k - 1$)

3. $\forall i, \quad Z(x_i) = Y(x_i).p(x_i) = Y(x_i).r_i$

Main Idea: Ignore all references to p above and look for Y, Z.

Decoding RS Codes (contd.)

I. Find (Y, Z) s.t.

- $Y \not\equiv 0$
- $\deg Y \leq e$
- $\deg Z \leq e + k - 1$
- $\forall i, \quad Z(x_i) = Y(x_i).r_i$

II. Output $\frac{Z(x)}{Y(x)}$.

Demystifying Step I: Just linear algebra!

Why does it work?

Claim 1: Pair of polynomials Y, Z satisfying the requirements of Step I do exist!

(In fact we just proved the existence.)

Claim 2: Linear Algebra can find one such pair.

(But pair may not be unique. How do we guarantee Y is the error-locator?)

Claim 3: If Y, Z and Y', Z' both satisfy conditions of Step I, then $Z/Y \equiv Z'/Y'$.

Proof of Claim 3

Consider the polynomials $Y' \cdot Z$ and $Y \cdot Z'$.

- Both have deg. $\leq 2e + k - 1$.
- For every $i \in \{1, \ldots, n\}$, $Z(x_i) = Y(x_i) \cdot r_i$ and $Y'(x_i) r_i = Z'(x_i)$.
- Multiplying and cancelling r_i's: $(Y' \cdot Z)(x_i) = (Y \cdot Z')(x_i)$.
- But above happens for n points, while degrees are smaller than n!
- So $Y' \cdot Z \equiv Y \cdot Z'$

Thm: Alg. works if $e \leq \frac{n-k}{2}$.

(As given, runs in time $O(n^3)$ time. Best implementations take $O(n \text{poly} \log n)$.)

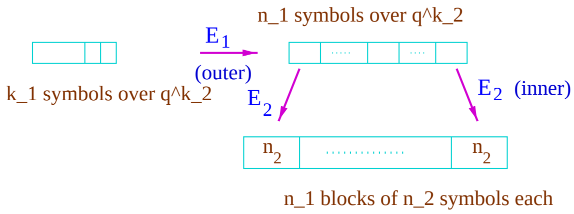
Musings

- Algorithm essentially in [Peterson'60]. Before "polytime" was formalized.
- Magic of algebra! Also a warning shot! Beware if you intend to base cryptography on algebra ...
- Roots of the specific algorithm. CS literature: [Berlekamp-Welch'86]. All ideas are there, but not the exposition. Exposition is from [Gemmell-S.'92].
- But equally simple exposition well-known in coding theory (from around 1988). [Pellikaan,Kotter,Duursma].

Decoding Concatenated Codes

Recall concatenation:

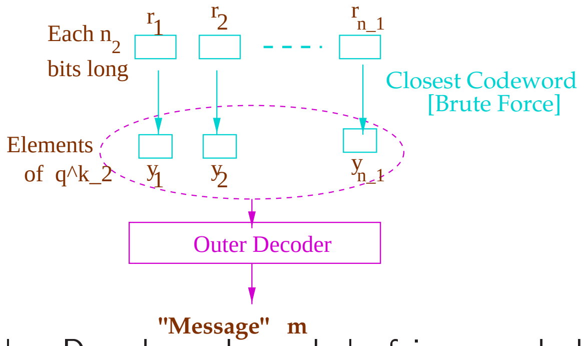
$$[n_1, k_1, d_1]_{q^{k_2}} \circ [n_2, k_2, d_2]_q$$



[Forney'66]: Also gave decoding algorithms.

Simple decoding

Prop: If outer code decodable up to e_1 errors (in poly time), then concatenated code is decodable up to $e_1 \cdot \frac{d_2}{2}$ errors in poly $+O(n_1 q^{k_2})$ time.



Alg: Decode each symbol of inner code by Brute force. Then decode the "received word" corr. to outer code.

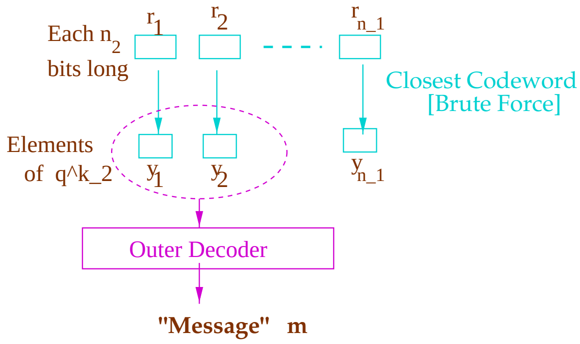
Generalized Min. Dist. Decoding

More sophisticated decoding. Stronger assumptions. Stronger result. [Forney].

Assumption: Outer code has error and erasure decoder. Decodes if $2e + s < d_1$, where $e = \#$ errors, $s = \#$ erasures.

Consequence: Concat. code can be decoded for up to $d_1 d_2/2$ errors ($=$ half the minimum distance).

GMD Algorithm



Alg:

- Let $w_i = \min\{\Delta(r_i, y_i), d_1/2\}$.
- W.l.o.g. $w_1 \leq w_2 \leq \cdots \leq w_{n_1}$.
- For $i = 1$ to n_1 do
  - Declare $\{i, \ldots, n_1\}$ to be erasures.
  - Decode prefix.

GMD Analysis

- Let m be s.t. $\Delta(E_2(E_1(m)), r) < d_1 d_2/2$.
  Let $\langle z_1, \ldots, z_{n_1}\rangle = E_1(m)$.
  Let $l_i = \Delta(z_i, r_i)$.
  Let $b_i = 1$ if $z_i \neq y_i$.
- Assume decoding unsuccessful. Then following inequalities hold:
  (1) $\forall j, \quad (n_1 - j) + 2 \cdot \sum_{i=1}^{j} b_i \geq d_1$
  (2) $\forall i, \quad l_i \geq \max\{w_i, b_i(d_2 - w_i)\}$
  (3) $\forall i, \quad w_i \leq w_{i+1} \leq d_2/2$
- Above imply:
  $\sum_{i=1}^{n_1} l_i \geq \frac{d_1 d_2}{2}$

Analysis (details)

(2) $\quad \Rightarrow \quad l_i \geq w_i + b_i(d_2 - 2w_i)$

So suffices to show:
$\sum_i w_i/d_2 + \sum_i b_i(1 - 2(w_i/d_2)) \geq d_1/2$.

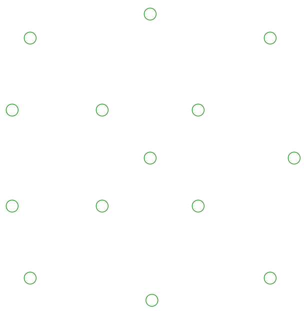
- Let $x_i = 1 - 2w_i/d_2$.
- Then x_i's are non-increasing, with $0 \leq x_i \leq 1$.
- Suffices to show:
  $\sum_i (1 - x_i/2) + \sum_i b_i x_i \geq d_1/2$,
  given $(n_1 - j)/2 + \cdot \sum_{i=1}^{j} b_i \geq d_1/2$
- Above follows if the vector
  $\langle x_1, \ldots, x_{n_1}, -\sum_i x_i\rangle$
  is in the convex hull of the vectors
  $v_1, \ldots, v_{n_1}$, where $v_j = \langle 1^j 0^{n_1 - j}, (-j)\rangle$.
- Last is easily verified.

Algebraic list-decoding

List decoding of Reed Solomon Codes

Given: t, d; $\{(x_1, y_1), \ldots, (x_n, y_n)\}$.
Task: Find all polynomials p of degree $\leq d$
that go through at least t points.

Example Instance



$$n = 14\ ; \quad d = 1\ ; \quad t = 5$$

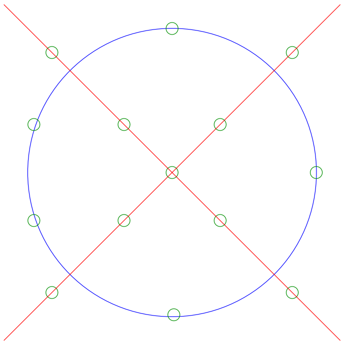
Solution Idea

- Somehow need to explain both the curve $y = x$ and $y = -x$. Further there is the mess of points (which fortunately lie on a circle $y^2 + x^2 = 1$. Need an error-locator for this!
- How to explain all these points together? Algebraically?
- Previously: Gave "functional" explanation. $y_i = R(x_i)$ for some rational function R.

Solution Idea (contd.)

- New idea: Try relational explanation.
- I.e. Find non-zero $Q(x, y)$ such that $Q(x_i, y_i) = 0$ for every i. (Ideally degree of Q small in both variables.)
- Hopefully p emerges in picture!
- Formally $(y - p(x)) \mid Q(x, y)$.

A Quartic Fit: All the Lines Emerge



$$Q(x,y) = y^4 - x^4 + y^2 - x^2$$

After factoring

$$Q(x,y) = (y^2 + x^2 - 1)(y + x)(y - x)$$

Finding $Q(x,y)$ with low degree

- Q1: How to find Q of low degree?
- Q2: Why does it even exist?

- A1: Solve linear system. Let $Q(x,y) = \sum_{j,l} q_{jl} x^j y^l$.
  Unknowns: $\{q_{jl} : 0 \le j, l \le D\}$.
  Constraints: $\sum_{j,l} q_{jl}\ x_i^j\ y_i^l = 0;\ i \in \{1, \ldots, n\}$.

- A2: Non-zero solution exists if

$$\#\text{ unknowns} > \#\text{ constraints}$$
$$(D+1)^2 > n$$

Conditions for divisibility of Q

- $\deg_x(Q), \deg_y(Q)$ small.
- $\deg(p)$ small.
- # pts. s.t. $Q(x_i, y_i) = y_i - p(x_i) = 0$ large.

Lemma: $\deg_x(Q), \deg_y(Q) \le D$, $\deg(p) \le d$, $t \ge D(d+1)$ implies $y - p(x)$ divides $Q(x,y)$.

Why? Similar to the fact that two degree k polynomials can not agree in $k+1$ places.

Proof

- Let $S = \{i | y_i = p(x_i)\}$.
- Let $g(x) = Q(x, p(x))$.
- $\deg(g) \le (d+1)D$.
- But $g(x_i) = Q(x_i, p(x_i)) = Q(x_i, y_i) = 0$ for every $i \in S$.
- $|S| > (d+1)D$ implies $g \equiv 0$.
- Thus $p(x)$ is a root of $Q(x,y)$ (or $y - p(x)$ divides $Q(x,y)$.

Summary

Algorithm:

Step 1: Find $Q \not\equiv 0$ of degree $\sqrt{n}$ s.t. $Q(x_i, y_i) = 0$ for every $i \in \{1, \ldots, n\}$.
(Solve a linear system.)

Step 2: Factor $Q(x, y)$ and report all p such that $y - p(x)$ divides Q.
(Factoring easy [Kaltofen,Grigoriev,Lenstra].

Theorem: Can find every p with agreement t, provided $t \geq (d+1)\sqrt{n}$.

Improvements:
– Careful choice of $Q \Rightarrow t \geq \sqrt{2dn}$.
(Can do even better [Guruswami-S.])

Other list decoding results

- AG codes. [SW'98].
- Chinese Remainder Codes [GRS'99].
- Reed-Muller codes. [STV'99].
- Concatenated codes. [GS'00].
- Extractor codes. [TSZ'01].
- Expander codes. [GI'01].

Non-algebraic decoding

Low-density parity check (LDPC) codes

Basic Idea: Pick parity check matrix of code carefully. If it is has low density and "pseudo-random" then a simple iterative decoding algorithm corrects constant fraction of errors.

- [Gallager], [Tanner]:
  - Introduced & developed Basic Idea.
- [Sipser-Spielman], [Spielman]:
  - Explicit codes;
  - Linear-time encoding & decoding.
- [Luby et al.], [Richardson et al.]:
  - Approach Shannon capacity with linear time algorithms.

Connections to CS Theory

Why we care?

- Sustained connections between coding theory and computer science over time.
- Most natural reason: Computation in the presence of noise.
- Other factors:
  - Source of ideas.
  - Right formalisms, language.
  - Common researchers.

Connections to unambiguous decoding

Secret sharing [Shamir].
Distribute secret among n parties so that any k can find secret, but not any $k-1$.
Soln: Take any $[n, k, n-k+1]$-code.

Probabilistic communication complexity
Question: Do probabilistic protocols offer advantage over deterministic ones?
Answer: Yes. Testing identity of k bits strings takes k bits deterministically. but only $O(\log k)$ bits probabilisitically, using any decent code.

ℓ-wise independence Construct small space of ℓ-wise independent bits.
Soln: [Alon,Babai,Itai]. Take code of distance $\ell+1$. Codewords of dual code give sample space.

List decoding connections

- Hardcore predicates Given hard function f find hard Boolean function B. ($B(x)$ easy if $f(x)$ is known, hard if not.)

  Soln: [Goldreich-Levin,Impagliazzo].
  Take a binary error-correcting code C with good list-decoding algorithm and let $B(x)$ be the ith bit of $C(f(x))$.

- Hardness amplification If EXP has a function f that is "hard" in the worst case, does is also have a hard on the average function?

  Answer: [IW,STV]. Yes. Interpret truth table of f as string and encode the string. The new string is the truth table of a really hard function, if the code is really good.

Other usages

- Literature on PCPs abounds with ECCs. List-decoding is a subliminal theme in optimal results.
- The new thrust in extractors has revolved around connections to list-decoding. [Trevisan] constructs extractors from good codes. [Ta-Shma+Zuckerman] construct good codes from extractors.
- Significant amounts of work in quantum computing revolve around (quantum) error-correcting codes and quantum information theory.

What Next?

Computer Scientists ought to be able to:

- Achieve Shannon Capacity. Primarily algorithmic questions.
- Move from random errors to worst case errors. Our forte!
- Unify many parts of extremal combinatorics (Codes, Extractors, PRGs, Expanders, Universal Traversal Sequences, Ramsey graphs.) Need more reductions.