A Crash Course on Coding Theory

Madhu Sudan
MIT

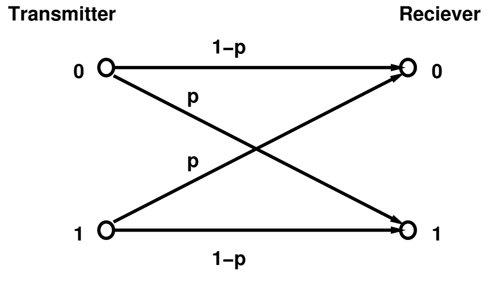
Topic: Linear time decoding - Part II

This lecture will focus on

- Simple and fast decoding algorithms
- Rate of noise close to optimal
- But noise is random, not adversarial.

Recall Shannon Capacity

Binary symmetric channel



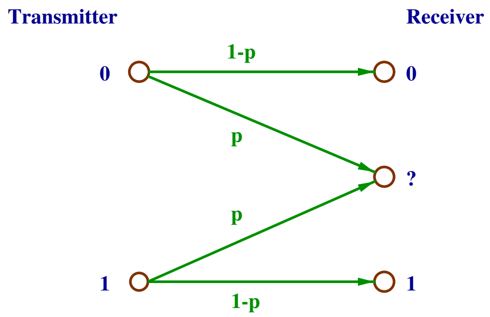
Capacity $= 1 - H(p)$, i.e.,

$\forall\ \epsilon > 0$, $\exists$ code of rate $1 - H(p) - \epsilon$,
s.t. if we transmit using code,
and decode from $\approx (p+\epsilon)$-fraction errors,
then recover message w.p. $1 - \exp(-n)$.

How to decode so much?

Other channels

Erasure channel



Capacity $= 1 - p$

Other channels (contd.)

AWGN Channel

(Additive White Gaussian Noise)

- Transmitted sequence in $\{-1, +1\}^n$.
- Received coordinates in $\Re^n$
  ith rec'd element equals
  Transmitted number $+ e_i$
  Where e_i is Gaussian r.v.
  with mean 0 and variance σ^2.

Note: σ replaces the parameter p.

Today's topics

- Polynomial time encoding and decoding up to capacity on Binary Symmetric Channel. (Original motivation of [Forney].)
- Linear time encoding and decoding up to capacity on Binary Symmetric Channel. (Using [Spielman].)
- A simple linear time encoding and decoding algorithm for the erasure channel. (Due to [Luby, Mitzenmacher, Shokrollahi, Spielman, Stemann].)

Forney Codes

Fix BSC parameter p and $\epsilon > 0$.

The code

- Let C_1 be $[n, (1-\epsilon)n, \epsilon n]_n$ RS code.
- Let C_2 be $[\ell, (1-H(p+\epsilon))\ell, (p+\epsilon)\ell]_2$ code with n messages. (i.e., $\ell \approx \log n$.)
- Let $C = C_1 \circ C_2$ be their concatenation.
- Transmit messages using C.

Its Parameters

- Block length $N = nl$.
- Rate $= (1-\epsilon)(1-H(p+\epsilon)) \approx (1-H(p))$.
- Distance Rate $= \epsilon p$.

Decoding Forney Codes

Simple Decoding algorithm

Step 1 Decode each inner block using Brute Force in time 2^ℓ.

Step 2 Decode outer code using RS decoder.

- But distance is pathetic!
- Why is this any good?
- (Work in Inf. Th. $\Rightarrow$ Know your probability.)

Analysis: Motivation

- Errors are random, so they are evenly distributed.
- Most blocks contain only p-fraction errors.
- Most errors caught by inner decoder.
- Outer decoder comes in to clean up.

Analysis: Formally

1. For any fixed inner block:
   $\Pr[\ \#\ \text{errors} \geq (p+\epsilon)\ell] \leq 2^{-\delta\ell}$.
   Call the bad event above a decoding failure.

2. $\Pr[\#\ \text{decoding failures} \geq \epsilon n/2] \leq 2^{-\gamma N}$.

   ($\gamma > 0$ depends on ϵ and is called the error exponent.)

3. If event in (2) doesn't happen, then decode successfully!

Thm: $\exists$ codes with rate $1 - H(p) - \epsilon$
with polytime encoders and decoders,
with decoding error prob. $\exp(-n)$
on the BSC with parameter p.

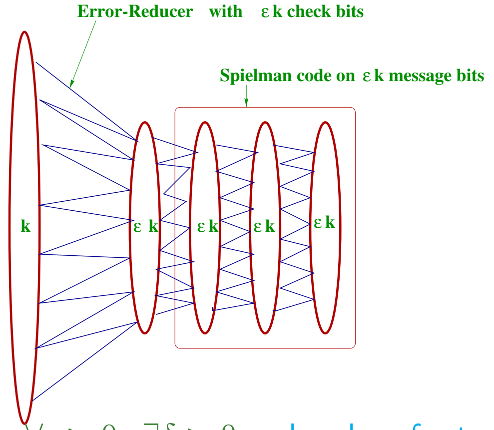
Moving on

- [Forney]'s work is from 1966.
- Introduced all the above ideas (concatenation, decoding, error analysis) and more.
- In fact also introduced GMD - why? To improve the error exponent!
- Very careful analysis needed to see why GMD helps!

Next: Linear time encoding and decoding.

Aside: High-rate Spielman codes

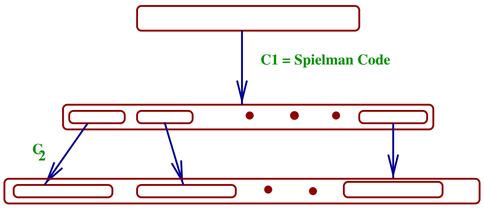
As described [Spielman] codes had rate $1/4$.
But can also get codes of high rates.



Thm: $\forall \epsilon > 0,\ \exists \delta > 0$ and codes of rate $1 - \epsilon$ that are linear-time encodable and decodable up to δ fraction errors.

Linear time encoding and decoding

Encoding:



- Given k message bits.
- Encode using Spielman codes of rate $1-\epsilon$.
- Partition into blocks with $\approx \frac{1}{\epsilon^2}$ bits.
- Encode blocks using random inner code of rate $\approx 1 - H(p)$.

Decoding + Analysis

- As usual decode inner blocks and then decode outer block.
- Prob. of inner decoding failure is small constant.
- Prob. that # of inner decoding failures is twice the expectation is exponentially small.

Thm: $\exists$ codes with rate $1 - H(p) - \epsilon$ with linear-time encoders and decoders, with decoding error prob. $\exp(-n)$ on the BSC with parameter p.

Towards Practice

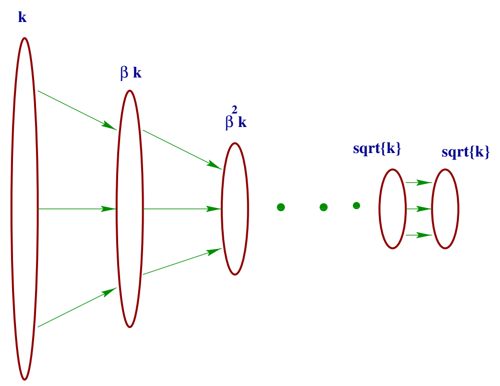
- Last theorem seems best possible theoretically.
- Not so good for practice.
- Needs large block lengths!
- Running time is actually $O(2^{\frac{1}{\epsilon^2}} \cdot n)$.
- While, can hope for running time of $O(n \cdot \text{poly}\log\frac{1}{\epsilon})$!

Recent developments

- Turbo codes + decoding:
  Simple codes + decoding algorithms, giving good results in simulations. [Benedetto, Montorsi, Thitijsima]. But no analysis?
- Low-density Parity Check Codes:
  - Provably good performance.
  - Reach capacity on erasure channel [LMSSS].
  - Come close on error channel. [LMSS, Richardson+Urbanke, ...].

Cascade codes [LMSSS]

Binary erasure channel with parameter β.



Main idea:

- Cascade seq. of "Error-reduction" codes.
- This helps correct the check bits first.
- Then correct message bits.

Cascade codes (contd.)

The construction

- Fix seq. of bipartite graphs $G_1, G_2, \ldots$.
- G_i has $\beta^{i-1}k$ left nodes and $\beta^i k$ right nodes.
- Identify right vertices of G_{i-1} with left vertices of G_i.
- Terminate when # vertices $\approx \sqrt{k}$
- Truncate with $O(n^2)$-time decodable code.

Encoding

- Message sets values of k left nodes of G_1.
- Encode left to right setting vertices to parity of their left neighbours.

Cascade codes (contd.)

Decoding

- Decode right to left.
- First decode final layer.
- Then, assuming all checkbits known for G_i, decode for "message" bits of G_i.
- Claim: Each layer fails with exponentially small probability.

Unspecified

- How are the graphs $G_1, G_2, \ldots$, picked?
- How to decode them?

The decoding algorithm

Will explain for G_1.

- Assume all checkbits known.
- Delete vertices corr. to message bits that are not erased, and incident edges.
- Iterate the following steps:
  - If $\exists$ edge (m, c) in residual graph, with c having degree one, then
  - Set m to be parity of c with ngbrs of c (in original graph).
  - Delete m and c from residual graph.
- Stop when no such vertex exists.

Properties of Cascade codes

- Rate $= 1 - \beta$.
- If graphs have linear number of edges, then encodable and decodable in linear time.
- Correct from β-fraction erasures, with all but exponentially small error probability, assuming the bipartite graphs can be constructed.

The bipartite graphs

- Option 1: Go the [Sipser+Spielman] route. (c, d)-regular graph with expansion $> c/2$.

  This is good to correct small # fraction of errors, but not close to capacity.
- Regular graphs seem to be no good!
- Irregular degree graphs work!
  Key innovation of [LMSSS].

The bipartite graphs (contd.)

- Pick a degree sequence $\{\lambda_i\}_i$, $\{\rho_i\}_i$, where λ_i (resp. ρ_i) denotes fraction of edges of left (resp. right) degree i.
- Let G_i's be random graphs with this degree pattern on appropriate # of edges.
- Rate condition: Degree seq. must satisfy

$$\frac{\sum_{i \geq 1} \lambda_i / i}{\sum_{i \geq 1} \rho_i / i} = \beta$$

.

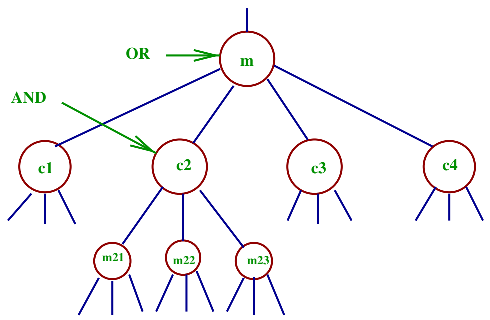
- Analyze as a function of the degree sequences.

Analysis via And-Or trees

- Say, decode in rounds: Delete all degree 1 edges simultaneously etc.
- Fix edge m, c. What is the prob. that this edge is not deleted by the ℓth round?

  1. m must be an erasure. AND
  2. $\exists$ check bit c_j s.t. for all m_{jk} adjacent to c_j (other than m), m_{jk} not deleted by round $\ell - 1$.
- Analysis leads to an "And-Or Tree" [LMS]. (assume no short cycles in graph).

And-Or trees



Let $q_\ell =$ Prob. of failure after ℓ rounds. Then

$$q_\ell \approx \beta\left(1 - \sum_i \lambda_i \left(1 - \sum_j \rho_j q_{\ell-1}^{j-1}\right)^{i-1}\right)$$

(Above informal: formal analysis hairier.)

Analysis (contd.)

Some compact notation:

Represent degree sequences by polynomials
$\lambda(x) = \sum_{i\geq 1} \lambda_i x^{i-1}$
and $\rho(x) = \sum_{i\geq 1} \rho_i x^{i-1}$.

Then $q_\ell = \beta(1 - \lambda(1 - \rho(q_{l-1})))$

When is decoding going to be successful?
If $q_\ell < q_{\ell-1}$.

Happens if
$\beta(1 - \lambda(1 - \rho(x))) < x, \quad \forall x \in (0, \beta)$.

Degree sequences?

- Given a degree sequence, can tell if it is good enough by previous analysis.
- How to find one?
  [LMSSS] give good sequence:
  λ_i proportional to $1/i$,
  up to max degree D.
  ρ_i's give Poisson distribution,
  with mean adjusted so as to
  satisfy rate condition.
- Note: Analysis only works for constant # of rounds. To finish off, add a Sipser-Spielman like analysis.

Theorem: Have linear time ($O(n \ln \frac{1}{\epsilon})$) encodable and decodable codes acheiving capacity on binary erasure channel.

Extending to BSC

- For the Binary Symmetric Channel, decoding algorithm has to change:
- Use a "Belief-Propagation" algorithm.
  - Maintain estimate (on edges) of prob. that incident message bit is 1.
  - On even rounds average the edges at the message end.
  - On odd rounds update the probability on the edges based on check bits.
- [LMSS], Richardson+Urbanke] prove that some degree sequences do very well.
- No analytic forms known on degrees. Numerically results come close to capacity (but not arbitrarily close.)

Conclusion

- Linear time decoding is an important feature in practice.
- Theoretically good analysis has resulted in good influence on practice.