Fundamental problems of coding theory

- Given a code, find/estimate its parameters.
  - Length - hopefully obvious.
  - Rate - also obvious for linear codes.
  - Distance!

- For a code $C$, solve the decoding problem.
  - Is the code given/official/well-known?
  - Is the distance of $C$ known or not?
  - Is the received word close to $C$ or not?

- Usually restrict to linear codes. Why?

Hardness of decoding

Maximum likelihood decoding (MLD):

Given: Generator matrix $G$ of linear code $C$.
Received vector $y$, error bound $e$.
Decide: $\exists c \in C$ s.t. $\Delta(c, y) \leq e$?

Thm: [Berlekamp, McEliece, van Tilborg '78]
MLD is NP-hard.

(MLD also called Nearest Codeword Problem in the CS literature.)
Proof

[Modern folklore] Reduction from Max CUT.

Max CUT:

Given: Graph $H = (V, E)$ and integer $k$.
Decide: $\exists S \subseteq V$ s.t. 
# edges from $S$ to $\overline{S}$ is at least $k$?

The reduction:

- Let generator $G$ be incidence matrix of $H$. 
  $([m, n, ?]-code, \text{where } n = |V|, m = |E|)$.
- Let $r = 1^m$ and $e = m - k$.

Analysis:

- Code = characteristic vectors of cuts.
- Codeword closest to $1^m$ is the one with largest # of edges crossing cut.

Approximability

- Ok - so can't find nearest codeword.
- Can you even find a nearby codeword?
- Or even approximate the distance to nearest codeword?
- Approximability: General modern day concern.

Nearest Codeword Problem (NCP)

Given: Generator $G$, received vector $y$.
Goal: Find codeword $c \in C_G$ nearest to $y$.

Defn: An $\alpha$-approximation algorithm to NCP is a polytime algorithm that, on input $(G, y)$, 
outputs $c' \in C_G$ that satisfies 
$\Delta(c', y) \leq \alpha \Delta(c, y), \forall c \in C_y$

Theorem: $\forall \epsilon > 0$, NCP is not $2^{\log^{1-\epsilon} n}$-approximable, if P $\neq$ NP.

- Theorem combines: 
  [Arora,Babai,Stern,Sweedyk '93] 
  + [Dinur,Kindler,Safra '99].
- Our proof: Uses stronger assumptions 
  Follows [Stern '93].

Proof

- Starting point: Know Max Cut is hard to approximate to within some $\alpha > 1$.
- Consequence: NCP is hard to approximate to within $\alpha > 1$, if P $\neq$ NP.
- Boosting the gap: Powering construction.
  Given $[n, k, ?]-code$ $C$ s.t. $\Delta(1^n, C) = \epsilon$.
  Can construct $[n^2, k(n+1), ?]$ code $C^2$ 
  s.t. $\Delta(1^{n^2}, C^2) = \epsilon^2$
- Conclude: NCP is not $\alpha$-approximable, for any $\alpha < \infty$, if NP $\neq$ P, and 
  is inapproximable to larger factors under stronger assumptions.
**Powering**

Codewords of $C^2$ are $n \times n$ matrices, constructed as follows.

For any collection of codewords $c, c_1, \ldots, c_n$ of $C$, $C^2$ contains the codewords $c^2$ drawn below:

![Diagram of $C^2$ construction]

**Analysis:** To pick codeword of largest weight, pick codeword $c$ of large weight in $C$ and let $c_i = c$, if $(c)_i = 0$ and $c_i = 0$ otherwise.

**Alternate routes**

- [ABSS]+[DKS]: Go deeper into “PCPs” to get the hardness result. (“Tailormade” PCPs.)
- [Hastad]: Celebrated result on inapproximability of Max 3SAT actually goes through the NCP! Yields weaker result, but in many senses cleaner.

**Thoughts**

- So something is hard! But what?
- If I throw a surprise code at you and ask you to decode, it will be hard! In fact, can even make the code linear, otherwise it will be a lot of effort to throw!
- Not in the usual spirit of problems we talk about.
- Still gives a useful application (inspiration?) — the McEliece cryptosystem.

**The McEliece Cryptosystem**

**Public-Key Cryptosystem**, inspired by the hardness of the NCP.

**Key generation:**
- Pick an $[n, k, d]_q$-AG code $C'$, with $t$-error-locating pair $A, B$.
- Pick random permutation $\pi \in \{0, 1\}^{n \times n}$
- Pick random non-singular $R \in \mathbb{F}_q^{k \times k}$.

**Private Key:** $(A, B, \pi, R)$.

**Public Key:**
- Let $G' \in \mathbb{F}_q^{k \times n}$ generate $C$.
- Let $G' = RG\pi$. ($G'$ generates $C$ with coordinates permuted by $\pi$).
- $G'$ is public key.
McEliece Cryptosystem (contd.)

Encryption: (of message $m \in \mathbb{F}_q^k$)
- Pick $\eta \in \mathbb{F}_q^n$ of weight $\leq \frac{n-k}{2}$
- Let $mG'' + \eta$ be its encryption.

Decryption:
- Given $r$, decode from $r\pi^{-1}$ using $(A, B)$.

Belief: Hard to decode, without knowledge of $\pi, R$.

Many ifs. Will return to this.

Decoding in the 80's

Which code? Distance $d$ vs. Errors $e$?

<table>
<thead>
<tr>
<th>Which code?</th>
<th>Known</th>
<th>Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e &lt; \frac{d}{2}$</td>
<td>RS</td>
<td>BCH</td>
</tr>
<tr>
<td>$e &gt; d$</td>
<td>?</td>
<td>[BMV]</td>
</tr>
</tbody>
</table>

Decoding vs. Preprocessing

- Positive results: For specific codes, $\exists$ algorithm that decodes efficiently (up to a limit on $\#$ errors).
- Negative results: There exists no algorithm that decodes all linear codes.
- Can we invert the quantifiers? “There exists a code, for which there is no efficient algorithm”?
- [Bruck & Naor ’90] addressed this problem.

Updates from the 90's

Added new coordinates.

<table>
<thead>
<tr>
<th>Which code?</th>
<th>Known</th>
<th>Fixed</th>
<th>Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e &lt; \frac{d}{2}$</td>
<td>RS</td>
<td>BCH</td>
<td>?</td>
</tr>
<tr>
<td>$\frac{d}{2} \leq e &lt; d$</td>
<td>RS</td>
<td>BCH</td>
<td>?</td>
</tr>
<tr>
<td>$e &gt; d$</td>
<td>?</td>
<td>[BN]</td>
<td>[BMV]</td>
</tr>
</tbody>
</table>
Decoding with preprocessing

Model:
- Allowed to preprocess the code.
  Preprocessing is computationally unbounded.
- But should not allow table lookup
- So preprocessing produces polysize circuit that decodes.

Challenge:
- Prev. NP-hardness had no “complexity” in received word - everything in code.
- Now we can’t do the same.
- How to transfer the complexity?

Decoding with preprocessing (contd).

Analysis:
- Codewords identical on twin coordinates.
- If $r$ has different values, that amounts to saying “don’t care”.

Theorem: There exists a code $C$ s.t. if it has a polynomial sized circuit decoding it, then $NP = P/poly$.

Warning: Does not preserve approximations.

Decoding with preprocessing (contd).

Reduction from Max CUT.

The Code:
- Let $C_1$ be code generated by incidence matrix of clique on $n$ vertices.
- Let $C$ be two-fold repetition of $C_1$.
  (Every edge of clique has two coordinates in the code - we call these “twins”.)

Received word
- Map $H$ to $r \in \{0,1\}^{n(n-1)}$.
- For every pair of vertices $i, j$ do
  If $(i, j)$ is an edge of $H$, then
    the twin pairs of $r$ are equal to 1.
  else they are unidentical.

Decoding up to the minimum distance

- Positive results decode up to a certain bound on number of errors; and at least assume $e < d$.
- Negative results don’t really mention distance of code!
- “These are certainly linear, but are they error-correcting codes?”
- Considered by [Dumer,Micciancio, S. ’99]

Diameter Bounded Decoding (DBD):

Given: Generator $G$, vector $r$, integers $e < d$.
Promise: Code has distance at least $d$.
Decide: Is $\Delta(r, C_G) \leq e$?
Diameter Bounded Decoding

Theorem: DBD is NP-hard under randomized reductions.

Comments:

- Proof adaptation of proof of [Ajtai] (and its simplification due to [Micciancio]) of NP-hardness of Shortest Lattice Vector Problem.
- Proof only uses instances with $\Delta(r, C) \leq e$ or $\Delta(r, C) > d$ and yields $e = d/(2 - \epsilon)$.

Review Ajtai-Micciancio (AM) proof

1. Combinatorial Step: Construct
   - $C = [n_1, k, d]_q$ code.
   - Vector $\vec{x} \in \mathbb{F}_q^{n_1}$ s.t.
     \[ \forall \vec{c} \in C, \quad \Delta(\vec{x}, \vec{c}) \leq \frac{d}{1.99} \]

2. Starting Point:
   Hard instance of Nearest Codeword Problem [ABSS]
   - $(B, \vec{v}, \leq \frac{d}{1.99} + \frac{d}{100}, > d)$.
   - $(B = [n_2, k, d']_q$ code, $\vec{v}$).

3. Endpoint:
   Paste to get hard instance of decoding.
   - $(C \circ B, \vec{x} \circ \vec{v}, \leq \frac{d}{1.99} + \frac{d}{100}, > d)$
   - $C \circ B$ has distance at least $d$.

Pasting

1. Strings = simple concatenation.

2. Codes = also concatenation .

\[ C \circ B \text{ has matrix } \begin{bmatrix} C \\ -B \end{bmatrix} \]

Codewords of $C \circ B$ are concatenations of codewords from $C$ and $B$.

\[ [n_1, k, d_1]_q \times [n_2, k, d_2]_q \Rightarrow [n_1 + n_2, k, d_1 + d_2]_q \]

Combinatorial Step: Details

Not Possible.

1. Combinatorial Step’: Construct
   - $C = [n_1, l, d]_q$ code.
   - Vector $\vec{x} \in \mathbb{F}_q^{n_1}$ s.t.
     for many $\vec{c} \in C$, $\Delta(\vec{x}, \vec{c}) \leq \frac{d}{1.99}$
   - Further construct $A \in \mathbb{F}_q^{k \times n_1}$ s.t.
     $A(S) = \mathbb{F}_q^k$.
     (where $S = \{ \vec{c} \in C \text{ s.t. } \Delta(\vec{c}, \vec{x}) \leq \frac{d}{1.99} \}$)

2. Endpoint’: Output
   $(C \circ (BAC), \vec{x} \circ \vec{v}, \leq \frac{d}{1.99} + \frac{d}{100}, > d)$
Good Case:
- $\exists \tilde{v} \in \mathbb{F}_q^k$ s.t. $\Delta(\tilde{v}, B\tilde{z}) \leq \frac{d}{100}$.
- $\exists \tilde{y} \in \mathbb{F}_q^n$ s.t. $\Delta(\tilde{x}, C\tilde{y}) \leq \frac{d}{1.99}$ and $AC\tilde{y} = \tilde{z}$.
- Then $\Delta((C \circ (BAC)) \cdot \tilde{y}, \tilde{x} \circ \tilde{v}) \leq \frac{d}{100} + \frac{d}{1.99}$.

Bad Case: Second part of codewords of $C \circ BAC$ are still codewords of $B$ and hence not close to $\tilde{v}$.

Lower bound for list decoding

- Recall new goal: Construct $C$ of distance $\geq d$.
- Code $C$ with exponentially many codewords of $C$ at distance $\frac{d}{1.99}$ from it.
- I.e., want List-Decode($C, \tilde{x}, \frac{d}{1.99}$) to have exponential output size.
- Possible?

The linear transform $A$

- Recall goal:
  - Have large set $S \subseteq \mathbb{F}_q^n$.
  - Want $A : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^k$ s.t. $A(S) = \mathbb{F}_q^k$.
  - Familiar problem in complexity.
  - Most natural idea: Pick $A \in \mathbb{F}_q^{n \times k}$ at random and hope it works.
  - Simple application of Chebycheff shows it works w.h.p. if $|S| \geq q^{2k}$.

How to get $C, \tilde{x}$?

- Adleman-Ajtai Lattice: Too number-theoretic.
- Random $C$ and $\tilde{x}$: Distance of $\tilde{x}$ to $C$ should be same as distance between two distinct vectors of $C$.
- Random $C$ and carefully chosen $\tilde{x}$: Unclear.
- Arbit $r$ and $C$ chosen carefully wrt $r$: Unclear.
Picking $C, \bar{x}$

- Pick a code $C$ that does better than random code!
  - Example Reed-Solomon Code.
  - Gives $[n, k, n-k]_q$ code, for any $k \leq n \leq q$.
  - Random code weaker. E.g. gives only $[n, n-n^c, \frac{1}{2-n^c}]_q$ code.

- Pick vector $\bar{x}$ at random.

- Expected number of vectors at distance at most $\frac{1}{2-e/2} n^c$ is exponentially large!

Done? Not yet.

---

An Inverted Markov Inequality

- Positive r.v. Expectation large. Want to sample so that prob. of finding small value is small.

- Markov’s bounds r.v. from above!

- Graph-theoretic formulation: Bipartite graph.
  - Left vertices = codewords
  - Right vertices = all vectors (space of $\bar{x}$)
  - Edge between $\bar{c}$ and $\bar{x}$ if they are within distance $\frac{d}{1.99}$.

- Expectation bound $\Rightarrow$ Average right degree large $D$.

---

Summary

Given: instance $(B, \bar{v}, \leq \frac{d}{100}, > d)$ of Nearest Codeword Problem

Pick Reed Solomon code $C$ of distance $d$ and $n \approx d^{100}$.

Pick random vector $\bar{x}$ at distance $\frac{d}{1.99}$ from $0^n$.

Output $(C \circ B A C, \bar{x} \circ \bar{v}, \leq \frac{d}{1.99} + \frac{d}{100}, > d)$.

Thm: DBD is NP-hard.
Minimum distance

• So far only focussed on the decoding question.

• What about the Distance of the code?

• Complexity undetermined till late 90’s.

• Finally resolved [Vardy ’97] - NP-complete indeed.

• Subsequently embelished with inapproximability [DMS ’99]. (Reduction from DBD.)

MinDist

Given: Generator $G$. Task: Find distinct codewords $c_1, c_2 \in C_G$ that minimize $\Delta(c_1, c_2)$.

Reducing DBD to MinDist

• Take a hard instance of DBD, i.e., $(C, r)$ s.t. $\Delta(r, C) \leq 2d/3$ or $\Delta(r, C) \geq d$.

• Consider $C' = C + r$.
  Either $\Delta(C') \leq 2d/3$ or $\Delta(C') \geq d$.

• NP-hard to distinguish.

Theorem: MinDist is hard to approximate to within a factor of 3/2, unless $NP = RP$.

But can now take tensor products of the code with itself and boost hardness result.

Theorem: MinDist is hard to approximate to within any constant factor, unless $NP = RP$. (Stronger results possible under stronger assumptions.)

Open Questions

Still - more open than closed!

• Can we certify just the good codes?

• Can we decode just the good codes?

• Show hardness of decoding RS Code?

• Hardness of decoding up to half the minimum distance?

• Hardness of decoding up to the minimum distance for a fixed code.

• Is there a worst-case to average-case connection here?

• Security of the McEliece Cryptosystem (implies all of the above?)

Topics we did not cover

• Convolutional codes. (Also Tree codes, and trellises.)

• Quantum error-correcting codes.

• Cyclic codes.

• Additive codes.

• Stuff that I don’t know about.
Acknowledgments

Amin Shokrollahi + Venkatesan Guruswami.