Today

- Complexity results for coding problems
  - (Might as well restrict to linear codes).
  - Hardness of the nearest codeword problem (NCP).
  - Approximation variants.
  - Decoding with preprocessing.
  - Decoding Relatively Near Codeword.
  - Minimum distance problem.
- What is not (known to be) hard?

Hardness of Nearest Codeword

- Given code with generator matrix G and received vector $\mathbf{r}$, find $\mathbf{x}$ that minimizes $\Delta(\mathbf{x}G, \mathbf{r})$.
- Hard even when $\mathbf{r} = \mathbf{1}$.
- Let G be incidence matrix of graph.
  - Rows = vertices
  - Columns = edges
  - 1 if edge incident to vertex.
- Messages = subset S of vertices; Codewords = characteristic vectors of cuts (1 if edge $S \to \overline{S}$).
- Nearest codeword to $\mathbf{1}$ is Max Cut!

Approximations

Search question: Given G and $\mathbf{r}$ compute $\mathbf{x}'$ such that $\Delta(\mathbf{x}'G, \mathbf{r}) \leq \alpha\Delta(\mathbf{x}G, \mathbf{r})$ for any $\mathbf{x}$.

Estimation question: Compute $t \in [\Delta(\mathbf{x}G, \mathbf{r}), \alpha\Delta$

Gap decision problem: Given $(G, \mathbf{r}, e)$ *promise* that $\tau = \min_{\mathbf{x}} \Delta(\mathbf{x}G, \mathbf{r}) \notin [t, \alpha t]$ decide if $\tau \leq t$ or not.

Note: Problems are provably no harder as we go down.

Analogous definitions for maximization problems.

Approximating NCP

- Know: Max Cut hard to approximate to within some $\alpha > 1$.
- Conclude? NCP hard to approximate?
  - Not immediate: If $X \in \{0, \ldots, m\}$ hard to approximate, is $m - X$ also hard? Not necessarily. E.g., if X actually in $\{0, \ldots, \sqrt{m}\}$, then m is a GOOD approximation to $m - X$!
  - Fortunately, in our case, we know $X \in \{m/2. \ldots, m\}$.
  - Can conclude: α approximation to $m - X$ gives $\alpha' = 1/(2 - \alpha)$ approximation to X. (Not useful if $\alpha \geq 2$. Why? But as $\alpha \to 1$, $\alpha' \to 1$ also!)

- Conclude: NCP hard to approximate to some $\alpha > 1$.

Approximating NCP (contd.)

- Self-improving problem: Given G of length n can construct a "product" $G^{(2)}$ of length n^2 such that G has vector of weight $n - w$ iff G has vector of weight $n^2 - w^2$.
- Conclude α-approx. hard implies α^2 approximation is hard implies any constant approximation is hard.
- The actual product:
  - Codewords of $G^{(2)}$ have n blocks of length n.
  - Any codeword of $G^{(2)}$ labels blocks as $0/1$. 0 blocks contain codewords of G, 1 blocks contain their complement.

  $0/1$ labelling of blocks corresponds to codeword of G.
  - Exercise: Show how to construct such a linear code.

First round of criticisms

- Code shouldn't be part of input.
  - After all we should be given lots of time to devise decoding algorithm.
- But how is this code "error-correcting".
  - To make sense, should be trying to correct less errors than minimum distance of code.
- What about Reed-Solomon codes (or substitute your favorite codes here)?

Decoding a fixed family of codes

uck-Naor : Can "inject" generator of code into received vector, while fixing code.

- Works whenever generator is a-sparse, i.e., has a 1s (even more general, actually).
- Basic idea: a-code C: Generator matrix has $2\binom{k}{a}$ columns, two for every column of a 1s.
- Now suppose have code B and received vector $\mathbf{r}$ as instance of NCP. Construct new received vector $\mathbf{r}'$ as follows: if a twin-pair of columns of C not in B, then put a $0, 1$ in corresponding coordinates of $\mathbf{r}'$. If twin-pair is in B, then duplicate corresponding entry in $\mathbf{r}$.

- Claim: $\Delta(\mathbf{x}C, \mathbf{r}') = N/2 - n + 2\Delta(\mathbf{x}B, \mathbf{r})$ where N is block length of C and n is block length of B.
- Conclude: Can't compute NCP exactly in for code C.
- Hardness of approximating in this setting: [Feige-Micciancio,Regev].

Addressing other complaint

io-Sudan : Can "boost" distance of code without altering the problem at hand (by much).

- Idea: Suppose finding nearest codeword to code generated by A is hard to approximate (to within factor of 100).
- Specifically, have $A, \mathbf{r}, d$ such that telling if $\tau > d$ or $\tau \leq d/100$ is hard.
- Attach to A, a matrix B which is generator of code of distance d.
- How to generate $\mathbf{r}'$? Details skipped...

A related problem

- Can we even compute minimum distance?
- Hardness of RNC above implies NO!
- Suppose G generates code of distance d with $(G, \mathbf{r}, d)$ being hard instance of NCP. Then code $G' = G + \mathbf{r}$ (with codewords being codewords of G translated by some multiple of $\mathbf{r}$ has distance $< d$ iff orig. instance is a YES instance.
- Implies Min Dist is hard to approximate to within some constant.
- Self-improvability (why?) implies hard to approximate to within any constant.

Open questions

- Solved problems raise more questions than resolve.
- Potentially polynomial-time solvable problems:
  - Exists a single decoding algorithm decoding all codes upto half the minimum distance.
  - Exists a minimum distance lower-bounding algorithm with guarantee that if rel. distance is $1 - \frac{1}{q} - \epsilon$, its lower bound is at least $1 - \frac{1}{q} - \sqrt{\epsilon}$.
  - NCP for Reed-Solomon (or your favorite) codes can be solved in polynomial time.
- Another general question: Decoding is a property of code? or the generator?