6.897 Algorithmic Introduction to Coding Theory	November 25, 2002
Lecture 20	
Lecturer: Madhu Sudan	*Scribe: Amit J. Deshpande*

1 Overview

In this lecture, we will see some complexity results for coding problems - known hardness results and some open questions.

- Hardness of the Nearest Codeword Problem (NCP)
- Approximation variants
- Decoding with preprocessing
- Decoding with relatively near codeword
- Minimum distance problem

2 Nearest Codeword Problem

The problem of finding out the nearest codeword (or maximum likelihood decoding) to a given received vector has been of crucial importance in the theory of error-correcting codes. Since in the general case, where the code is described by an encoding circuit, the problem of finiding a message corresponding to a given codeword is already hard so we might as well restrict our attention only to the linear codes. So given the code by its generator matrix and a received vector, find out a codeword nearest to it. We will formalize this as follows -

Definition 1 (Nearest Codeword Problem - NCP) *Given a code with generator matrix G and received vector r, find x that minimizes $\Delta(xG, r)$.*

How hard is it to solve NCP ? We will show that NCP is hard even for the special case when $r = \bar{1}$. This is done by a reduction from Max Cut (which is a well-known NP-hard problem).

Definition 2 (Max Cut Problem) *Given graph $H = (V, E)$ find $S \subseteq V$ such that, the number of edges between S and $\bar{S}$ is maximum.*

The reduction goes as follows - Let G be the incidence matrix of a graph $H = (V, E)$ with $|V| = k$ and $|E| = n$. So our message x corresponds to the subset of V specified by the 1's in it and codewords correspond to those edges e which give 1 after multiplication by x. i.e. both the 1's in e cannot be in S or $\bar{S}$. So e must be a crossing edge. Thus the codewords correspond to cuts and finding max cut is equivalent to finding the maximum weight codeword (meaning, nearest to $\bar{1}$).

3 Approximation variants of NCP

There are three important variants of the approximation problems. For a given instance (G, r) we will define $\tau = \min_x\{\Delta(xG, r)\}$, and $\alpha > 1$ be our approximation parameter.

Definition 3 (Search question): *Find x' such that $\tau \leq \Delta(x'G, r) \leq \alpha \cdot \tau$.*

Definition 4 (Estimation question): *Estimate t such that $\tau \leq t \leq \alpha \cdot \tau$.*

Definition 5 (Gap decision problem):("promise" problem) *Given (G, r, t) with the promise that $\tau \notin [t, \alpha t]$ decide if $\tau \leq t$ or not.*

And it's easy to observe that a solution to search problem gives a solution to estimation problem, and a solution to estimation problem gives a solution to Gap decision problem. Also as α becomes closer and closer to 1 the problems get harder. Analogous definitions can be made for the maximization versions of these problems.

4 Hardness of approximating NCP

A critical question would be - is it hard even to find an approximately nearest codeword ?

We know that Max Cut is hard to approximate to within some $\alpha < 1$. So we can use this fact to show the hardness for NCP. Elementary probability (first moment method) gives that every graph has at least a cut of size $|E|/2$, where $|E|$ is the number of edges. And the reduction that we used for showing the NCP is NP-hard says that finding a Max Cut of size x corresponds to getting a codeword of weight x. i.e. a codeword within distance $n - x$ from $\bar{1}$. But since we know that $x \geq n/2$. This alongwith an β-approximation to NCP within $n - x$, $n - x \leq n - x' \leq \beta(n - x)$, gives that $\frac{1}{(2-\beta)}x \leq x' \leq x$. And thus a $\alpha = 1/(2-\beta)$-approximation to Max Cut. And $\alpha \to 1$ as $\beta \to 1$. But we already know that Max Cut cannot be approximated within $\alpha < 1$ for some α, which implies the corresponding hardness result for NCP as -

Theorem 6 *NCP is hard to approximate to within some $\beta > 1$.*

Moreover, we can prove something stronger as this problem has a self-improving property.

Theorem 7 *β-approximation to NCP is hard implies that β^2-approximation is also hard. And using this repetitively we get, any constant approximation to NCP is hard.*

Proof The proof involves a clever construction - given G generator matrix of a code of length n, we can construct a "product" $G^{(2)}$ generator matrix of a code of length n^2 such that G has a codeword of weight $n - w$ iff $G^{(2)}$ has a codeword of weight $n^2 - w^2$.

A codewords of $G^{(2)}$ is an $n \times n$ matrix with columns labelled by a codeword of G. Each column is a codeword of G or its complement according to the label 0 or 1, respectively. To our surprise, this happens to be a linear code.

So if G has a code of weight $n - w$ then we can cosider the codeword in $G^{(2)}$ that has $\bar{1}$ in all the columns labelled by 1's and the $n - w$ weight code in G in all the columns labelled by 0's. And the labelling also corresponds to the $n - w$ weight codeword of G. This gives a codeword of in $G^{(2)}$ of weight $n^2 - w^2$. And that's the maximum you can do to stuff your matrix with more and more 1's.

This clearly implies that if there is a β^2-approximation algorithm for the code $G^{(2)}$ then it should give a β-approximation for code G. And thus β-approximation hardness for G translates into β^2-approximation hardness, too. ■

5 Criticism

There has been a lot of criticism on this which gives rise to the following problems -

- Code shouldn't be part of the input and we should be given a lot of preprocessing time to devise the decoding algorithm.
- How do these results relate to the error-correction property ? To make sense, we should be trying to correct less errors than the minimum distance of the code.
- The codes we saw here had a very low-density generator matrix as it was corresponding to the incidence matrix of a graph. But we want hardness results for better codes. e.g. Reed-Solomon codes, algebraic geometry codes, LDPC codes, Turbo codes (any of your favourite codes).

We will analyze some results that try to address these questions.

6 More hardness results addressing the criticism

6.1 Hardness of decoding a fixed family of codes [Bruck-Naor]

The first criticism regarding sparse generator matrix was addressed by Bruck-Naor [1] and the idea was to "inject" the generator of the code into received vector, while fixing the code. Let G be the incidence matrix of a graph. For every pair of vertices (u, v), have twin-pair of columns. So such a code C has a generator matrix with $2\begin{pmatrix} k \\ a \end{pmatrix}$ columns. Now suppose that we have code B and received vector r as an instance of NCP. Construct a new received vector as follows: if edge (u, v) is in G then duplicate the entry of r in the corresponding coordinate of r', and otherwise put $0, 1$.

Now note that, $\Delta(xC, r') = N/2 - n - 2\Delta(xB, r)$ where N and n are the block lengths of C and B, respectively. So the minimum distances are related and we cannot compute NCP exactly for the code C.

This method also works when the generator matrix is a-sparse (in fact, more generally). Hardness of approximating in this setting is studied in Feige-Micciancio [2].

6.2 Decoding codes upto min distance [Dumer-Micciancio-Sudan]

This addressed the other criticism regarding hardness results for asymptotically good codes. Dumer-Micciancio-Sudan [3] show that we can "boost" the distance of the code without altering the problem too much. This was shown by showing a hardness result for a version of Gap Decision Problem for the minimum distance.

Suppose that finding the nearest codeword to code generated by A is hard to approximate (to within factor of 100, say). Then we specifically have A, r, d such that telling if $\tau > d$ or $\tau \leq d/100$ with high probability is hard. The trick is to attach to A a generator matrix B of a code of distance d, and getting an appropriate r'.

Dumer-Micciancio-Sudan [3] show that decoding codes of minimum distance d for upto less than d errors is NP-hard.

7 Open questions

All these still raise a few more open questions -

- Can you solve NCP is polytime for some asymptotically good family of codes ? Reed-Solomon ? or your favourite code ?
- Does there exist a single decoding algorithm decoding all codes upto half the minimum distance ?
- Does there exist an algorithm giving a lower bound for minimum distance which guarantees that if the relative distance is $1 - \frac{1}{q} - \epsilon$ then the lower bound given by the algorithm is at least $1 - \frac{1}{q} - \epsilon^2$?

References

[1] J.Bruck, M.Naor, *The hardness of decoding linear code with preprocessing*, IEEE Transaction on Information Theory, pp. 381-385, May 1990.

[2] U.Feige, D.Micciancio, *The inapproximability of lattice and coding problems with preprocessing*, Proceedings of IEEE Conference on Computational Complexity, pp. 44-52, 2002.

[3] I.Dumer, D.Micciancio, M.Sudan, *Hardness of approximating the minimum distance of a linear code*, IEEE Transaction on Information Theory, Jan 2003 (to appear).