CSE522/Math581EB Coding Theory **Fall 2001**

Lecture 11: November 13

Lecturer: Elchanan Mossel *Scribe: Erik Vee*

11.1 Hardness of Problems in Coding

We show that two important problems in coding theory are at least as hard as *MAX-CUT*, an *NP-hard* problem. Recall the definition of *MAX-CUT*. Given a graph, $G = (V, E)$, along with any subset of the vertices, $U \subseteq V$, we let

$$\partial_E(U) = \{(u, v) \in E : u \in U, v \notin U\}$$

(This is commonly called a *cut* with respect to U and $V - U$.) The output of *MAX-CUT* is the $U' \subseteq V$ that maximizes $\partial_E(U')$ over all possible subsets. In the decision version of *MAX-CUT* we are given a positive integer k. The output is 1 if there exists a subset $U' \subseteq V$ such that $|\partial_E(U')| \geq k$, and is 0 otherwise.

Proposition 11.1 *Both MAX-CUT and the decision version of MAX-CUT are NP-hard*

In subsection 11.1.2, we will use the fact that even approximating *MAX-CUT* is hard.

Proposition 11.2 *Approximating MAX-CUT to within a factor of* $\alpha_0 = 16/15$ *is NP-hard.*

11.1.1 Maximum Likelihood Decoding

We first define the *Maximum Likelihood Decoding* problem [*MLD*].

Definition 11.3 *Given a generator,* $\mathcal{G}$*, for a linear code* C*, a minimum distance,* r*, and a received word,* y*, then the output of MLD is* 1 *if there exists a codeword* $c \in C$ *such that* $\Delta(c, y) \leq r$*, and* 0 *otherwise.*

We will now show that for any graph, G, there is a linear code, C, such that solving *MLD* on C will allow us to find a maximum size cut in G. But first, we need one lemma to help show that the code we construct is in fact linear. Let $\oplus$ denote both the bitwise XOR of two vectors over F_2, and the symmetric difference between two sets.

Lemma 11.4 *Let* $G = (V, E)$ *be a graph, and let* $U_x, U_y \subseteq V$*. Then*

$$\partial_E(U_x) \oplus \partial_E(U_y) = \partial_E(U_x \oplus U_y)$$

Proof: First suppose there is an edge $(u, v) \in \partial_E(U_x \oplus U_y)$. Without loss of generality, we may assume that $u \in U_x \oplus U_y = (U_x \cap \overline{U}_y) \cup (\overline{U}_x \cap U_y)$, and $v \in \overline{U_x \oplus U}_y = (U_x \cap U_y) \cup (\overline{U}_x \cap \overline{U}_y)$. So we have four cases to consider.

(i) $u \in U_x \cap \overline{U}_y, v \in U_x \cap U_y$:

Then $(u, v) \in \partial_E(U_y) - \partial_E(U_x) \subseteq \partial_E(U_x) \oplus \partial_E(U_y)$.

(ii) $u \in \overline{U}_x \cap U_y, v \in U_x \cap U_y$:
Then $(u,v) \in \partial_E(U_x) - \partial_E(U_y) \subseteq \partial_E(U_x) \oplus \partial_E(U_y)$.
(iii) $u \in U_x \cap \overline{U}_y, v \in \overline{U}_x \cap \overline{U}_y$:
Then $(u,v) \in \partial_E(U_x) - \partial_E(U_y) \subseteq \partial_E(U_x) \oplus \partial_E(U_y)$.
(iv) $u \in \overline{U}_x \cap U_y, v \in \overline{U}_x \cap \overline{U}_y$:
Then $(u,v) \in \partial_E(U_y) - \partial_E(U_x) \subseteq \partial_E(U_x) \oplus \partial_E(U_y)$.

This shows that $\partial_E(U_x) \oplus \partial_E(U_y) \supseteq \partial_E(U_x \oplus U_y)$. The other direction is proved similarly. ■

Theorem 11.5 ([BmEvT78]) *MLD is NP-complete.*

Proof: Clearly, $MLD \in NP$. We will show that *MLD* is *NP-hard* by a reduction from *MAX-CUT*.

Let $G = (V, E)$ be a graph. We construct a code, $C \subseteq \{0,1\}^E$. For every $E' \subseteq E$, let $\chi(E') \in \{0,1\}^E$ denote the vector whose e-th coordinate is 1 if $e \in E'$, and 0 otherwise. Then let $C = \{\chi(\partial_E(U)) : U \subseteq V\}$. Notice that there is a one-to-one correspondence between codewords in C cuts in the graph G.

It is easy to that C is linear using Lemma 11.4. For any $x, y \in C$, there are $U_x, U_y \subseteq V$ such that $x = \chi(\partial_E(U_x))$ and $y = \chi(\partial_E(U_y))$. Hence,

$$\begin{aligned} x \oplus y &= \chi(\partial_E(U_x)) \oplus \chi(\partial_E(U_y)) \\ &= \chi(\partial_E(U_x) \oplus \partial_E(U_y)) \\ &= \chi(\partial_E(U_x \oplus U_y)) \in C \end{aligned}$$

In fact, the generator for C is simply $\mathcal{G} = \langle \chi(\partial_E(\{v\})) : v \in V \rangle$.

Now, the max-cut in G corresponds to the codeword in C that is closest to $\chi(E)$. So any algorithm solving *MLD* also solves *MAX-CUT*. That is, *MLD* is *NP-hard*. ■

11.1.2 Nearest Codeword Problem

The *Nearest Codeword Problem* [*NCP*] is defined as follows: Given the generator, $\mathcal{G}$, for a linear code C and a received word y, return the codeword $c \in C$ such that $\Delta(c, y)$ is minimized (that is, $\Delta(c, y) = \Delta(C, y)$). It is clear from the previous section that *NCP* is *NP-hard*. We show in this section that even approximating it is hard.

Definition 11.6 *An α-approximation of NCP is a polytime algorithm that given a generator,$\mathcal{G}$, for a linear code C and a received word y, returns a codeword $c' \in C$ such that $\Delta(c', y) \leq \alpha\Delta(C, y)$.*

Note that Proposition 11.2 immediately shows that there is no polynomial α-approximation for *NCP* if $\alpha \leq \alpha_0 = 16/15$, assuming that $P \neq NP$. The following theorem shows the same is true for any constant α. The proof shows that given an α-approximation, we can also find a $\alpha^{\frac{1}{2}}$-approximation.

Theorem 11.7 ([S93]) *There is no α-approximation of NCP for α constant, unless $P = NP$.*

Proof: Suppose there is an α-approximation of *NCP* for some α constant. As we have already noted, if $\alpha < \alpha_0 = 16/15$, then we have a contradiction. We now show that using a powering construction, we can construct an $\alpha^{\frac{1}{2}}$-approximation. Since α is constant, we can repeat the powering construction a constant number of times to produce an algorithm giving better than an α_0-approximation, contradicting Proposition 11.2.

Given a graph $G = (V, E)$, construct the linear code C as in the proof of Theorem 11.5. Note that if the all ones vector, $\vec{1}$, is in C, then our α-approximation finds the true max cut, so we may assume that $\vec{1} \notin C$.

Let $n = |E|$. Then for $w_1, w_2, \ldots, w_n \in \{0,1\}^n$, let $M(w_1, w_2, \ldots, w_n)$ denote the $n \times n$ matrix whose i-th row is w_i. Similarly, $M^T(w_1, w_2, \ldots, w_n)$ is the matrix whose i-th column is w_i. Let

$$C^2 = \left\{ M^T(w_0, w_0, \ldots, w_0) + M(w_1, w_2, \ldots, w_n) : w_0, w_1 \ldots, w_n \in C \right\}$$

It is easy to check that C^2 is linear and its generator is polytime-constructible from the generator of C, so we may run our α-approximation on it. The key property of C^2 is that its distance from $M(\vec{1}, \vec{1}, \ldots, \vec{1})$ is exactly the square of the distance of C from $\vec{1}$.

To see this, define $f : C \to C^2$ for $c \in C$ by $f(c) = M^T(c, c, \ldots, c) + M(w_1, w_2, \ldots, w_n)$, where

$$w_i = \begin{cases} c & \text{if the } i\text{-th coordinate of } c \text{ is } 0 \\ \vec{0} & \text{otherwise} \end{cases}$$

For any $w \in C^2$, there are $w_0, w_1 \ldots, w_n \in C$ such that $w = M^T(w_0, w_0, \ldots, w_0) + M(w_1, w_2, \ldots, w_n)$. Let w_{min} be the vector from $w_0, w_1, \ldots, w_n \in C$ that minimizes $\Delta(w_i, \vec{1})$. Let S be the set of coordinates on which w_{min} is 0, and let T be the set of coordinates on which w_{min} is 1. Further, let S_0 be the set of coordinates on which w_0 is 0, and let T_0 be the set of coordinates on which w_0 is 1. Note that $|S| \leq |S_0|$. Then

$$\begin{aligned} \Delta(f(w_{min}), M(\vec{1}, \ldots, \vec{1})) &= \sum_{i \in S} \Delta(w_{min}, \vec{1}) + \sum_{i \in T} \Delta(\vec{1}, \vec{1}) \\ &= |S|(\Delta(w_{min}, \vec{1})) \\ &\leq |S_0|(\Delta(w_{min}, \vec{1})) \\ &= \sum_{i \in S_0} \Delta(w_{min}, \vec{1}) + \sum_{i \in T_0} \Delta(\vec{1}, \vec{1}) \\ &\leq \sum_{i \in S_0} \Delta(w_i, \vec{1}) + \sum_{i \in T_0} \Delta(\vec{1}, \vec{1}) = \Delta(w, M(\vec{1}, \ldots, \vec{1})) \end{aligned}$$

Hence, $\Delta(C^2, M(\vec{1}, \ldots, \vec{1})) = \Delta(f(C), M(\vec{1}, \ldots, \vec{1}))$. Furthermore, since $|S| = \Delta(w_{min}, \vec{1})$, we see that $\Delta(f(w_{min}), M(\vec{1}, \ldots, \vec{1})) = (\Delta(w_{min}, \vec{1}))^2$.

Now, we can run the α-approximation on C^2 using received word $M(\vec{1}, \ldots, \vec{1})$. Suppose the algorithm returns $m \in C^2$, so we know that $\Delta(m, M(\vec{1}, \ldots, \vec{1})) \leq \alpha \Delta(C^2, M(\vec{1}, \ldots, \vec{1}))$.

Furthermore, there exist $m_0, m_1, \ldots, m_n \in C$ such that $m = M^T(m_0, \ldots, m_0) + M(m_1, \ldots, m_n)$. Note that m is itself a $n \times n$ matrix, and the i-th row is either m_i or $\vec{1} \oplus m_i$. Since $\vec{1} \notin C$ and C is linear, we know that $\vec{1} \oplus m_i \notin C$. Hence, by checking whether or not each row of m is in C, we can compute the value of m_0 and $m_1, \ldots, m_n$ in polytime. Let m_{min} be the vector from $\{m_1, \ldots, m_n\}$ that minimizes $\Delta(m_i, \vec{1})$. From our argument above, we see

$$\begin{aligned} (\Delta(w_{min}, \vec{1}))^2 = \Delta(f(w_{min}, M(\vec{1}, \ldots, \vec{1})) &\leq \Delta(m, M(\vec{1}, \ldots, \vec{1})) \\ &\leq \alpha \Delta(C^2, M(\vec{1}, \ldots, \vec{1})) = \alpha(\Delta(C, \vec{1}))^2 \end{aligned}$$

That is,

$$\Delta(w_{min}, \vec{1}) \leq \alpha^{\frac{1}{2}} \Delta(C, \vec{1})$$

So we have produced a $\alpha^{\frac{1}{2}}$-approximation, as we wanted.

■

In fact, it is possible to prove an even stronger form of the above theorem. Its proof is beyond the scope of these notes.

Theorem 11.8 ([ABS97]) *For all $\epsilon > 0$, there is no α-approximation of NCP for $\alpha = 2^{\log^{1-\epsilon} n}$, unless $P = NP$.*

11.1.3 Hard to Decode Linear Codes

We have seen earlier that there is no polytime algorithm for *MLD* unless $P = NP$. We now show that solving *MLD* even for a single family of codes may be difficult.

Theorem 11.9 ([BN90]) *There exists a family of codes $C_1, C_2, \ldots$ with $C_i \subseteq \{0,1\}^i$ for all $i \geq 1$ such that if there is a polytime algorithm solving MLD for all codes in the family, then $P = NP$. Similarly, if it is possible to solve MLD for the codes in the family using polynomial circuits (i.e. if it is possible to solve MLD for the codes in $P/poly$), then $P/poly = NP$.*

Proof: We will define a code C_i for every i that can be written as $i = n(n-1)$. (We may choose any codes we like for i not of this form.) Let $C' \subseteq \{0,1\}^{n(n-1)/2}$ be the code described in Theorem 11.5 for the complete graph on n vertices. As before, we see that any $c' \in C'$ corresponds to a set of edges in the graph. Let $C = C' \times C'$.

We now show that any algorithm that solves *MLD* on C can also be used to solve *MAX-CUT* for any graph. Let $G = (V, E)$ be a graph, and set $y = \vec{1} \circ \chi(E)$. Let $\mathcal{E}$ denote the set of edges in the complete graph. For any $c \in C$, there is a set of edges, $\mathcal{E}' \subseteq \mathcal{E}$, such that $c = (\chi(\mathcal{E}'), \chi(\mathcal{E}'))$ and $\mathcal{E}'$ is the set of edges for some cut in the complete graph. Let $E' = E \cap \mathcal{E}'$. Note that E' is a set of edges for some cut in G.

We have

$$\begin{aligned}
\Delta(c, y) &= |\mathcal{E} \oplus \mathcal{E}'| + |E \oplus \mathcal{E}'| \\
&= |(E \cup \overline{E}) \oplus \mathcal{E}'| + |E \cap \overline{\mathcal{E}'}| + |\overline{E} \cap \mathcal{E}'| \\
&= 2|E \cap \overline{\mathcal{E}'}| + |\overline{E} \cap \overline{\mathcal{E}'}| + |\overline{E} \cap \mathcal{E}'| \\
&= 2|E - E'| + |\overline{E}|
\end{aligned}$$

Thus, it follows that finding a $c \in C$ with minimal distance from y is the same as finding a cut E' of maximal size. ■

References

[1] [ABS97] S. Arora., L. Babai., J. Stern and Z. Sweedyk (1997) The hardness of approximate optima in lattices, codes, and systems of linear equations. **34th** Annual Symposium on Foundations of Computer Science (Palo Alto, CA, 1993). *J. Comput. System Sci.* **54 no. 2, part 2**, 317–331.

[2] J. Bruck and M. Naor (1990) The hardness of decoding linear codes with preprocessing. *IEEE Trans. Inform. Theory* **36, no. 2**, 381–385.

[3] [S93] J. Stern (1993) Approximating the number of error locations within a constant ratio is NP-complete (1993) *Applied algebra, algebraic algorithms and error-correcting codes (San Juan, PR)* 325–331, in Lecture Notes in Comput. Sci., 673, Springer, Berlin, 1993.

[4] [BmEvT78] E. R. Berlekamp., R. J. McEliece and H. C. A. van Tilborg (1978) On the inherent intractability of certain coding problems. *IEEE Trans. Information Theory IT-24* (1978), **no. 3,** 384–386.