Hardness Of Approximation

Lecture 3: Proving Hardness of gap$(0.99, 1-\delta)$-3LIN, Subcode Covering

Instructor: Irit Dinur and Amey Bhangale *Scribe: Tal Herman*

1 Reminder from last week

This week we finish the proof we started last week, showing that gap$(0.99, 1-\delta)$-3LIN is $\mathcal{NP}$-hard. Recall that last week we showed a reduction from gap$(\epsilon, 1)$-`LabelCover` to gap$(0.99, 1-\delta)$-`3LIN`. Before returning to the analysis, let us first review the construction of the reduction once more.

1.1 Reducing gap$(\epsilon, 1)$-`LabelCover` to gap$(0.99, 1-\delta)$-`3LIN`

Given a *biregular, projective* instance $G = (U, V, E, \Pi)$ of `LabelCover`, we create $2^{|\Sigma_w|}$ variables for each $w \in U \cup V$. Instead of explicitly describing the set of equations over these variables, we define a probabilistic verifier of these assertions. The system of equations can be obtained by enumerating over the random coins of the verifier. We proceed with details of the verifier: given access to an assignment over the variables, the verifier runs as follows:

- Sample a uniform edge $(u, v) \in E$. Denote the assignment to the 2^{Σ_u} variables created from u by $f_u : \{0,1\}^{\Sigma_u} \to \{0,1\}$, and similarly, denote by $g_v : \{0,1\}^{\Sigma_v} \to \{0,1\}$ the assignment to the 2^{Σ_v} variables obtained form v. Next, perform one of the following tests, each with probability 1/3:

  (T1) Run the long code test with parameter δ (algorithm 2.3 in previous lecture) on f_u.

  (T2) Run the long code test with parameter δ on g_v.

  (T3) Consistency test: check that the labels (allegedly) encoded by f_u and g_v are consistent with the constraint $\Pi_{u,v}$: uniformly sample $x \in \{0,1\}^{\Sigma_v}$, and $y \in \{0,1\}^{\Sigma_u}$, and check:

  $$f_u(y) + f_u(\tilde{x} + y) = g_v(x)$$

  Where for each $\sigma \in \Sigma_u$, the σ'th coordinate of $\tilde{x}$ is $\tilde{x}_\sigma = x_{\Pi_{u,v}(\sigma)}$.

The value of δ as a function of ϵ will be determined during the proof.

2 Analysis of the reduction

The reduction presented above has a slight flaw in it - the *all-zeros* assignment passes all tests, but it is meaningless and provides no information about the original instance. Therefore, we first correct the reduction enforcing the condition that for every $u \in U$:

$$f_u(x) = \overline{f_u(\overline{x})},$$

where the bar represents flipping the binary value. In other words, we demand that f_u will satisfy the property that negating all the bits in the input will result with the negation of the output. Clearly, this condition does not hold for the *all-zeros* assignment.

2.1 Proving Completeness

Lemma 2.1. *If the* `LabelCover` *instance is satisfiable, then the value of the* `3LIN` *instance produced by the above reduction (i.e. the largest possible fraction of simultaneously satisfied equations) is at least* $1 - \delta$.

Proof. Let $G = (U, V, E, \Pi)$ be a completely satisfiable instance of `LabelCover`, where in a satisfying labelling the label of vertex $u \in U$ is a_u, and the label of vertex $v \in V$ is b_v. In the produced `3LIN` instance assign $f_u = Dict_{a_u}$, and $g_v = Dict_{b_v}$, for all $u \in U$ and all $v \in V$. It is left to show that the value of this assignment to the `3LIN` instance is indeed greater than $1 - \delta$. Note that both test `T1` and test `T2` will pass with probability of at least $1 - \delta$, since for every $u \in U$ $f_u(x) = x_a$, $f_u(y) = y_a$, and $f_u(x + y + u) = x_a + y_a + \eta_a$. Therefore:

$$f_u(x) + f_u(y) + f_u(x + y + \eta) = x_a + y_a + x_a + y_a + \eta_a = \eta_a$$

And by definition of η, $\eta_a = 0$ with probability greater than $1 - \delta$. The same logic applies to g_v as well. As for test `T3`, consider u and v as above, with the relevant dictator functions f_u and g_v associated with them. Then, for randomly selected $y \in \{0,1\}^{\Sigma_u}$, and $x \in \{0,1\}^{\Sigma_v}$:

$$f_u(y) + f_u(\tilde{x} + y) + g_v(x) = y_a + \tilde{x}_a + y_a + x_b = \tilde{x}_a + x_b$$

Recall that under the condition $\tilde{x}_a + x_b = 0$, the test succeeds. Thus, the probability of success is $\Pr(\tilde{x}_a + x_b = 0)$, but this probability is in fact 1, if labels a and b are indeed aligned with the constraints $\Pi_{u,v}$, since by definition, $\tilde{x}_a = x_b$, as $\Pi_{u,v}(a) = b$. □

2.2 Proving Soundness

Lemma 2.2. *If the* `LabelCover` *instance has value of at most* ϵ*, then produced* `3LIN` *instance has value of at most* 0.99

In order to prove this lemma, we first prove several claims. These claims will help us assert that any `3LIN` instance produced by the reduction with value of at least 0.99, must have too many vertices that can be labelled according to the restrictions of the original instance, and therefore, the instance must have a value of more than ϵ. Thus, from now on, let us consider an assignment of a produced `3LIN` instance with value 0.99.

Claim 2.3. *For at least* $\frac{1}{4}$*-fraction of the edges, under such assignment, the edge passes all three tests* `T1,T2,T3` *with probability of at least* 0.9*. Call this section of edges "good".*

Proof. Assume by way of contradiction that at most a quarter of the edges are *good*. Namely, that they pass each of the three tests with probability greater than 0.9, and consider the value of the relevant `3LIN`

instance:

$$\begin{aligned}\Pr_{(u,v)\in E}\left[(u,v)\ passes\ test\right] &\leq \Pr_{(u,v)}\left[(u,v)\ passes\ test \mid (u,v)\ is\ "good"\right]\cdot \Pr_{(u,v)}\left[(u,v)\ is\ "good"\right] + \\ &+ \Pr_{(u,v)}\left[(u,v)\ passes\ test \mid (u,v)\ is\ not\ "good"\right]\cdot \Pr_{(u,v)}\left[(u,v)\ is\ not\ "good"\right] \leq \\ &\leq \frac{3}{4}\left(\frac{1}{3}\cdot 1+\frac{1}{3}\cdot 1+\frac{1}{3}\cdot 0.9\right)+\frac{1}{4}\cdot 1 = \frac{8.7}{12}+\frac{3}{12} = \frac{11.7}{12} < 0.99,\end{aligned}$$

where the first inequality on the last line uses the assumption that the *good* set is upper bounded by $\frac{1}{4}$, as well as bounding from above the conditioned probabilities - inside the good set, by 1, and outside the good set, assuming one test passes with probability at most 0.9. We conclude that the probability that an edge passes the general test is lesser than 0.99, which implies the value of the instance is smaller than 0.99, and we reached contradiction. □

Recall from last lecture that we defined the Fourier coefficient to be $\hat{f}_u(k) = \langle f_u, \chi_k\rangle = \mathbb{E}_x\left[f_u(x)\chi_k(x)\right]$ for $k \in \{0,1\}^{\Sigma_u}$ (equivalently, for $k \subseteq \Sigma_u$), where $\chi_k = (-1)^{\sum_{i=1}^{|\Sigma_u|} k_i x_i}$. The next two claims give us possible *good* labels for a vertex incident at a good edge, by reducing Σ_u and Σ_v to smaller subsets of labels which are likely to be coherent with the labelling restrictions of the original `LabelCover` instance. This will later help us define a labelling that achieves a large number of satisfied edges. Also, from now onwards we consider the multiplicative version of the relevant functions and tests, rather than the additive ones used thus far.

Claim 2.4. *If (u,v) is a good edge, then:*

1. *$\exists S \subseteq \Sigma_u$, $0 < |S| \leq \frac{1}{\delta}$ such that $\hat{f}_u(S) \geq 0.8$*
2. *$\exists T \subseteq \Sigma_v$, $0 < |T| \leq \frac{1}{\delta}$ such that $\hat{g}_v(T) \geq 0.8$*

Proof. We proved last lecture (claim 2.5 in the notes) that the long code is sound, meaning that if a function passes the test with probability non-trivially higher than $\frac{1}{2}$, then it is correlated with a sparse linear function. In this case, it implies that for all $\delta \in [0,1]$, if $\Pr(f_u\ passes\ test) \geq \frac{1}{2}+\epsilon$, then $\max_S \hat{f}_u(S)(1-\delta)^{|S|} \geq 2\epsilon$. By assumption, on good edges, the test passes with probability of at least 0.9, therefore, setting $\epsilon = 0.4$, we get:

$$\max_S \hat{f}_u(S)(1-\delta)^{|S|} \geq 0.8$$

Similarly, for g_v:

$$\max_T \hat{g}_v(T)(1-\delta)^{|T|} \geq 0.8$$

Note that for all $|S| > \frac{1}{\delta}$ it must be then that $(1-\delta)^{\frac{1}{\delta}} < \frac{1}{e}$. Therefore, if we assume that $S_0 = \text{argmax}_S \hat{f}_u(S)$ satisfies $|S_0| > \frac{1}{\delta}$, this will imply that $\hat{f}_u(S_0) \geq 0.8 \cdot e > 1$, which is a contradiction. Therefore, it must be that $|S_0| \leq \frac{1}{\delta}$. □

Consider such T and S to represent restrictions on possible labels for vertices v and u respectively.

Claim 2.5. *Let T and S be defined as in claim 2.4, then $T \subseteq \Pi_{u,v}(S)$.*

Proof. Suppose by way of contradiction that there exists $j \in \Sigma_v$ such that $j \in T \setminus \Pi_{u,v}(S)$. Recall that the edge (u,v) is a *good* edge, therefore it passes the consistency test, `T3`, with probability of at least 0.9. Following immediately from the definition of the multiplicative version of `T3`:

$$\Pr\left[\texttt{T3}\ passes\right] = \frac{1}{2}+\frac{1}{2}\mathbb{E}\left[f_u(y)f_x(t+\tilde{x})g_v(x)\right] = (\star) \tag{2.1}$$

We wish to bound this value **from above** by 0.9, which will contradict the fact that the chosen edge is *good*, and thus reach contradiction. From claim 2.4 we know that f_u and g_v are highly correlated with χ_S and χ_T respectively. Namely:

$$\Pr\left[f_u(y) = \chi_S(y)\right] = \frac{1+\hat{f}(S)}{2} \geq 0.9$$

$$\Pr\left[g_v(x) = \chi_T(x)\right] = \frac{1+\hat{g}(T)}{2} \geq 0.9$$

Consider the event of the functions "disagreeing" on relevant inputs. By the above observations and the use of union bound:

$$\Pr\left[f_u(y) \neq \chi_S(y) \vee f_u(y+\tilde{x}) \neq \chi_S(y+\tilde{x}) \vee g_v(x) \neq \chi_T(x)\right] \leq 0.1 + 0.1 + 0.1 = 0.3$$

We use this to bound expression 2.1 from above, splitting the product of the functions inside the expectancy into two parts - conditioned on agreeing on all evaluated points with the characteristic functions (happens in probability of at least 0.7), and conditioned on having at least one function in disagreement (which happens in probability of at most 0.3). In the latter case, we also bound the conditional expectancy achieved by 1. this yields:

$$\Pr\left[\texttt{T3 } \textit{passes}\right] = \frac{1}{2} + \frac{1}{2}\mathbb{E}\left[f_u(y)f_x(t+\tilde{x})g_v(x)\right] \leq \frac{1}{2} + \frac{1}{2}\left[0.7 \cdot \mathbb{E}\left[\chi_S(y)\chi_S(y+\tilde{x})\chi_T(x)\right] + 0.3 \cdot 1\right] \tag{2.2}$$

$$= \frac{1}{2} + \frac{1}{2}\left[\mathbb{E}\left[\chi_S(\tilde{x})\chi_T(x)\right] + 0.3\right] \tag{2.3}$$

Where the last equality is due to $\chi_S(y+\tilde{x}) = \chi_S(y)\chi_S(\tilde{x})$, and the fact that $\chi_S(y) \cdot \chi_S(y) = 1$. We are left with evaluating the expression $\mathbb{E}\left[\chi_S(\tilde{x})\chi_T(x)\right]$. Indeed:

$$\begin{aligned}
\mathbb{E}\left[\chi_S(\tilde{x})\chi_T(x)\right] &= \mathbb{E}\left[(-1)^{\sum_{i\in S}\tilde{x}_i} \cdot (-1)^{\sum_{k\in T} x_k}\right] = \\
&= \mathbb{E}\left[(-1)^{\sum_{i\in S}\tilde{x}_i} \cdot (-1)^{\sum_{k\in T\setminus\{j\}} x_k} \cdot (-1)^{x_j}\right] = \\
&= \mathbb{E}\left[(-1)^{\sum_{i\in S}\tilde{x}_i} \cdot (-1)^{\sum_{k\in T\setminus\{j\}} x_k}\right] \cdot \mathbb{E}\left[(-1)^{x_j}\right] = 0,
\end{aligned}$$

where the first equality on the last line stems from the fact that x_j is independent by assumption from the distribution of the product $(-1)^{\sum_{i\in S}\tilde{x}_i} \cdot (-1)^{\sum_{k\in T\setminus\{j\}} x_k}$. And since by itself $\mathbb{E}\left[(-1)^{x_j}\right] = 0$, the above result follows. Plugging this result back in equation 2.3 yields:

$$\Pr\left[\texttt{T3 } \textit{passes}\right] \leq \frac{1}{2} + \frac{1}{2}\left[\mathbb{E}\left[\chi_S(\tilde{x})\chi_T(x)\right] + 0.3\right] \leq 0.5 + 0.15 = 0.65$$

and the contradiction follows from the reasoning presented above. □

We now turn to wrap up the proof of the soundness of the reduction, namely, lemma 2.2:

Proof of lemma 2.2. Assume by way of contradiction that an instance of `LabelCover` with value smaller than ϵ produced through the reduction an instance of `3LIN` with value of at least 0.99. We can apply claim 2.3, and conclude that at least a quarter of the edges are *"good"*. Fix a good edge (u, v). Consider the following randomized labelling of the original instance, $\mathcal{R}$: For u, if $\exists S$ such that $\hat{f}_u(S) > 0.8$ and $|S| < \frac{1}{\delta}$, pick $a \in S$ and assign it to u. Similarly, for v, if $\exists T$ such that $\hat{g}_v(T) > 0.8$, and pick uniformly at random $b \in T$ and assign it to v. Now:

$$\Pr_{(u,v)}\left[(u,v) \text{ is satisfied by } \mathcal{R}\right] \geq \frac{1}{4}\Pr\left[(u,v) \text{ is satisfied by } \mathcal{R} \mid (u,v) \text{ is good}\right] \geq \frac{\delta}{4},$$

where the last inequality is due to the fact that upon choosing a label a for u from S, we need to pick its proper image under $\Pi_{u,v}$, and from claim 2.5, it follows that $T \subseteq \Pi_{u,v}(S)$, and at worst, T contains $\frac{1}{\delta}$ elements (when the projection constraint over the edge is "injective"), thus the probability of choosing the label corresponding to $\Pi_{u,v}(a)$ is at least δ. Now, Plugging $\delta = 5\epsilon$ we get:

$$\Pr_{(u,v)}\left[(u,v) \text{ is satisfied by } \mathcal{R}\right] > \epsilon$$

which is a contradiction, since we assumed the original instance had value smaller than ϵ, and if the random labelling succeeded with higher probability, it implies there exists some labelling with value larger than ϵ. □

Note that the last article also defines the correct value to be assigned to the parameter δ for the reduction to hold.

3 Subcode Covering

We now wish to improve the previous result and show that gap$(\frac{1}{2}+\delta, 1-\delta)$-`3LIN` is also $\mathcal{NP}-hard$. In order to show this, we introduce the idea of subcode covering.

Note that in the previous proof, when picking an edge under a certain assignment, we needed to check the *consistency* of functions defined by the assignment. Due to the choice of encoding, namely, the *long code*, one side ended up being much larger than the other. For example, if we had for some edge (u,v), $|\Sigma_u| = 3k$, while $|\Sigma_v| = 3k-1$, then after the reduction, an assignment on u will be represented by $f_u : \{0,1\}^{\Sigma_u} \to \{0,1\}$, which means that the amount of possible different assignments over u is a set of size $2^{2^{|\Sigma_u|}} := 2^d$ (denoting $d = 2^{3k}$). For the same reasoning, the amount of possible assignments on v will be $2^{2^{|\Sigma_v|}} = 2^{2^{3k-1}} = 2^{\frac{d}{2}} = \sqrt{2^d}$. We would therefore need $\sqrt{2^d}$ small codes to *cover* the larger one (when, in reality, each vertex v only has $2^{\mathcal{O}(k)} = \text{poly}(d)$ neighbours).

This can be put in other words to mean that the long code of k bits, containing all boolean functions on k bits, doesn't look like the long code on $k-1$ bits, or, even more precisely, a *typical* Boolean function over k bits doesn't look like a typical Boolean function over $k-1$ bits. This can be seen by observing that a typical Boolean function over k bits, $f(x_1, x_2, \ldots, x_k)$, is not likely to give the same value for all pairs of strings $(\alpha_1, \alpha_2, \ldots, \alpha_{k-1}, 0), (\alpha_1, \alpha_2, \ldots, \alpha_{k-1}, 1)$ (i.e. all pairs of strings that only differ on a single entry).

Interestingly, in contract, a typical ***linear*** Boolean function over k bits **does** looks like a typical linear function over $k-1$ bits. Let f be some random linear function on k bits - $f(x_1, x_2, \ldots, x_k) = \sum_{i=1}^{k} a_i x_i$ for some $(a_1, a_2, \ldots, a_k) \in \{0,1\}^k$. In a typical such function, half of the a_i are 0. This prompts us to produce a slightly biased random linear function in the following way, ensuring at least one entry is indeed 0:

1. Choose $i \in [k]$.

2. Set $a_i = 0$.

3. Choose a_j independently from the set $\{0,1\}$, for $j \neq i$.

Claim 3.1. *Choosing a linear function f over k bits in the manner presented above is "almost" like choosing a function defined by a totally randomly chosen k-bit string.*

This claim stems from the fact that forcing only one entry to be 0 results with only a slight bias towards 0 (formally, the statistical distance between the distribution produced by the above procedure and the uniform distribution over the k bit linear functions is negligible). In fact, even when choosing multiple random indices (provided that the amount of indices chosen $<< \sqrt{k}$) the claim persists.

3.1 Approaching $\mathcal{NP} - hardness$ of gap$(\frac{1}{2}+\delta, 1-\delta)$-3LIN

The notions presented above promote the idea of replacing the *long code* by the **Hadamard code** with the hope that by doing so, the consistency test will be made simpler. With this in mind, we show the following theorem:

Theorem 3.2. *gap$(\epsilon, 1-\epsilon)$-LabelCover that satisfies the following conditions:*

- *Smoothness.*
- *"Linear" constraints*

is $\mathcal{NP} - hard$

The two conditions presented in the statement - *smoothness* and *linear constraints* will be explored and explained during the proof, that will be presented in the next lecture.

From this theorem we will derive the following theorem:

Theorem 3.3. *For any small constant $\delta > 0$, gap$(\frac{1}{2}+\delta, 1-\delta)$-3LIN is $\mathcal{NP} - hard$.*

We give here a sketch of the construction of the reduction underlying the proof of Theorem 3.2:
The reduction starts with an instance of gap$(0.99, 1-\delta)$-3LIN. Choose $k << \frac{1}{\epsilon}$, and $t << \sqrt{k}$. Define:

$$U = \text{All } k\text{-tuples of clauses}$$
$$V = \text{All } k\text{-tuples with } k-t \text{ clauses and } t \text{ variables}$$
$$E = \{(u,v) \mid for\ each\ i \in [k],\ u_i = v_i\ or\ v_i \in u_i\}$$

Next week we present the analysis to this construction, as well as explore more deeply the concept of subcode covering.