1 (Notes from the ITCSC Summer research program in Hong Kong)

1.1 Problem Statement and Definition

Suppose someone gives you a source $X \in \{0,1\}^n$ of n unbiased iid coin flips. Imagine we want to somehow process X and stretch it out to give a new bitstream $Y \in \{0,1\}^{n'}$, where each bit of Y is approximately an iid coin flip with bias $\frac{1}{4}$. This is easy to do when $n' = \frac{n}{2}$: for each of the $n/2$ consecutive blocks of X bit pairs $x_i x_{i+1}$, output $x_i \wedge x_{i+1}$. However, we want n' to be close to the information theoretic limit, say $n' \approx n/H(\frac{1}{4})$ (where H is the binary entropy function).

What we want to do is find a map to compute $Y := f(X)$ which reshapes the distribution of X into the target distribution of Y. We know how to do this with methods like arithmetic coding, computing the inverse Cumulative Distribution Function, and other compression/decompression methods (there's a nice construction using the leftover hash lemma), but all these methods are "non-local": ideally, to compute a bit Y_i, we should only need to look at b bits of X. It's possible to come up with such a construction using block decompression techniques for $b = \log(n)^2$; a more interesting question to ask is whether we can get $b = \mathcal{O}(1)$. If we're trying to look for a b local f to solve this problem, then one line of inquiry is to study how the entropy of $f(X)$ is distributed between the bits of $f(X)$ given that f is b-local. If the entropy of $Y = f(X)$ is always concentrated on a small fraction of bits when f is $\mathcal{O}(1)$ local, then we know that such a construction is impossible.

Definition 1. *Let $S = \{s_1 < s_2 < ... < s_k\} \subset [n]$, and $X = x_0...x_{n-1}$ a variable taking values in $\{0,1\}^n$. We write $X_S := x_{s_1}...x_{s_k}$ to denote the variable X whose bits are restricted to the set S.*

Definition 2. *We say a function $f : \{0,1\}^n \to \{0,1\}^{n'}$ is b-local if for each index $i \in [n']$, the ith bit $f(X)_i$ can be written as a function of at most b bits of X. In particular, $\forall i \in [n']$, $\exists S_i \subset [n]$ with $|S| = b$ and a function $f_i : \{0,1\}^b \to \{0,1\}$ such that $f(X)_i = f_i(X_{S_i})$.*

Question: Let $X \sim \{0,1\}^n$ be a uniformally drawn random variable, and let H return the entropy of a random variable. For any constants $b \in \mathbb{N}, \epsilon > 0$, given a b-local function $f : \{0,1\}^n \to \{0,1\}^{n'}$, does there always exist a subset $S \subset [n']$ with $|S| = \mathcal{O}(n)$ such that $H(f(X)|_S) \geq (1-\epsilon)H(f(X))$? Here we think of $n' >> n$, say $n' = \theta(n^2)$ (if $n' = \mathcal{O}(n)$ the question is trivial). Stated informally, given a b-local function f which stretches out an iid bit source, is most of the entropy of $f(X)$ concentrated on only a small fraction of its bits, or can the entropy also be "spread out" between all of its bits?

1.2 Solution of the problem for b=2

Lemma 3. *Let $Y_1, ..., Y_k, \tilde{Y}_1, ..., \tilde{Y}_k$ be random variables. Suppose that $\forall i \in [k]$, $H(Y_i|\tilde{Y}_i) \leq \epsilon H(Y_i)$. Then $H(Y_1, ..., Y_k|\tilde{Y}_1, ..., \tilde{Y}_k) \leq k\epsilon H(Y_1, ..., Y_k)$*

Proof.

$$
\begin{aligned}
H(Y_1, ..., Y_k|\tilde{Y}_1, ..., \tilde{Y}_k) &\leq \sum_{i\in[k]} H(Y_i|\tilde{Y}_1, ..., \tilde{Y}_k) && (1)\\
&\leq \sum_{i\in[k]} H(Y_i|\tilde{Y}_i) && (2)\\
&\leq \epsilon \sum_{i\in[k]} H(Y_i) && (3)\\
&\leq k\epsilon H(Y_1, ..., Y_k) && (4)
\end{aligned}
$$

□

Remark 4. *Note that in the proof, we actually have $H(Y_1, ..., Y_k|\tilde{Y}_1, ..., \tilde{Y}_k) \leq k^*\epsilon H(Y_1, ..., Y_k)$, where k^* is k minus the number of times the relation $H(Y_i|\tilde{Y}_i) = 0$ holds.*

Theorem 5. *Let X be uniform over $\{0,1\}^n$, and let $Y \in \{0,1\}^{n'}$ be a function of X such that each bit Y_i depends on at most $b = 2$ bits of X. Then there exists a set $S \subset [n']$ with $|S| \leq C_\epsilon n$ such that $H(Y|_S) \geq (1-\epsilon)H(Y)$.*

Proof intuition The intuition for the proof is as follows: note that each bit Y_i is a potentially different function $f_i : \{0,1\}^b \to \{0,1\}$ of some subset of the bits of X. We can use the previous lemma to partition the bits of Y into groups depending on which of the (at most) 2^{2^b} functions they are associated with. If we can solve the problem assuming the bits of Y are all computed by the same local function, we can use the previous lemma to stitch our solutions for each partition together and only pay a constant factor in entropy loss. It turns out that the only tricky subproblem to consider is $f(x_i, x_j) = x_i \wedge x_j$. The intuition for this problem is as follows: if there are a constant number $y_1, ..., y_c$ of Y bits which depend on x_i, then we can capture most of the entropy of x_i by looking at these Y bits: if $x_i = 0$, then all of $y_1, ..., y_c$ will be 0. If $x_i = 1$, then we expect at least one of $y_1, ..., y_c$ to be 1 with overwhelming probability. If x_i has fewer than c Y bits which depend on it, we can afford to simply include all the Y bits which depend on x_i into our set S, and we capture "all of the information of x_i" which is contained in Y. We can use this observation to take at most a constant number of Y bit "samples" for each X bit x_i; afterwards, we should have captured most of the entropy Y. The following proof makes these ideas formal:

Proof. First, consider the case where each bit of Y is a single fixed function f of pairs of bits of X. The claim is easily seen to be true when f is constant, linear, or the identity (modulo negation) in one of its arguments (in each case, we need at most n elements to capture exactly all of the entropy of Y by using guassian elimination).

The only remaining case to consider (up to negations) is when each bit of Y is an AND of bits of X, i.e. $f(x_i, x_j) = x_i \wedge x_j$. Construct a graph $\mathcal{G}$ with n vertices, where we have an edge $(a, b) \in E[\mathcal{G}]$ if $\exists i \in [n']$ s.t. $Y_i = X_a \wedge X_b$. Associate with each vertex a in $\mathcal{G}$ with its corresponding bit X_a, and each edge (a, b) of $\mathcal{G}$ with its corresponding bit $Y_i = X_a \wedge X_b$.

Now iteratively do the following: start with an empty set E_l. While there exists a vertex v with $0 < deg(v) < C$, add the edges incident v to E_l and remove these edges from $E[\mathcal{G}]$. Let E_h be the remaining edges after this process terminates, and let V_h be the vertices incident to E_h. Then we have $|E_l| \leq C(n - |V_h|)$ and that every vertex $v \in V_h$ has at least C vertices in E_h. For each $v \in V_h$, arbitrarily pick C edges in E_h which are adjacent to v, and add them to a new set $\tilde{E}_h$. First, we claim $H(V_h|\tilde{E}_h) \leq |V_h|(C+2)2^{-C-1}$. This follows since

$$\begin{aligned}
H(V_h|\tilde{E}_h) &\leq \sum_{v \in V_h} H(v|\tilde{E}_h) && (5)\\
&\leq \sum_{v \in V_h} H(v|E_v) && (6)\\
&= \sum_{v \in V_h} \sum_{x \in \{0,1\}, y \in \{0,1\}^C} Pr[v = x, E_v = y] \log \left(\frac{Pr[E_v = y]}{Pr[v = x \wedge E_v = y]} \right) && (7)\\
&= \sum_{v \in V_h} \sum_{y \in \{0,1\}^C} Pr[v = 1, E_v = y] \log \left(\frac{Pr[E_v = y]}{Pr[v = 1 \wedge E_v = y]} \right) && (8)\\
&+ \sum_{v \in V_h} Pr[v = 0, E_v = 0] \log \left(\frac{Pr[E_v = 0]}{Pr[v = 0 \wedge E_v = 0]} \right) && (9)\\
&= \sum_{v \in V_h} \left[\sum_{y \in \{0,1\}^C, y \neq \vec{0}} Pr[v = 1, E_v = y] \log \left(\frac{Pr[E_v = y \wedge v = 1] + Pr[E_v = y \wedge v = 0]}{Pr[v = 1, E_v = y]} \right) \right] && (10)\\
&+ \sum_{v \in V_h} \left[2^{-C-1} \log \left(2^{C+1} Pr[E_v = 0] \right) + \frac{1}{2} \log \left(2(\frac{1}{2} + 2^{-C-1}) \right) \right] && (11)\\
&= \sum_{v \in V_h} \left[2^{-C-1} \log \left(2^{C+1}(\frac{1}{2} + 2^{-C-1}) \right) + \frac{1}{2} \log \left(2(\frac{1}{2} + 2^{-C-1}) \right) \right] && (12)\\
&\leq \sum_{v \in V_h} (C+1)2^{-C-1} + 2^{-C-1} && (13)\\
&= |V_h|(C+2)2^{-C-1} && (14)\\
& && (15)
\end{aligned}$$

Where for each $v \in V_h$ we denote $E_v \subset \tilde{E}_h$ to be a set of C vertices adjacent to v. Note that (10) vanishes because $Pr[E_v = y \neq 0 \wedge v = 0] = 0$. Now, we have

$$H(\tilde{E}_h) \geq (1 - (C+2)2^{-C-1})|V_h| \tag{16}$$

(17)

And hence

$$
\begin{aligned}
H(E_h|\tilde{E}_h) &= H(E_h) - H(\tilde{E}_h) && (18)\\
&\leq H(E_h) - (1-(C+2)2^{-C-1})|V_n| && (19)\\
&\leq H(E_h) - (1-(C+2)2^{-C-1})H(E_n) && (20)\\
&= (C+2)2^{-C-1}H(E_n) && (21)\\
& && (22)
\end{aligned}
$$

The first equality follows from the fact that $\tilde{E}_h$ is a function of V_h so $H(\tilde{E}_h) = H(V_h) - H(V_h|\tilde{E}_h)$, and $H(V_h) = |V_h|$. The second equality follows because $\tilde{E}_h$ is a function of E_h, and the third inequality follows because E_n is a function of V_n.

Lastly, note that $|\tilde{E}_h| + |E_l| \leq C|V_h| + C(n - |V_h|) = Cn$ and

$$
\begin{aligned}
H(Y|\tilde{E}_h, E_l) &= H(E_h, E_l|\tilde{E}_h, E_l) && (23)\\
&\leq (C+2)2^{-C-1}H(E_h, E_l) = (C+2)2^{-C-1}H(Y) && (24)
\end{aligned}
$$

by lemma 3 and remark 4, so

$$
\begin{aligned}
H(\tilde{E}_h, E_l) &= H(Y) - H(Y|\tilde{E}_h, E_l) && (25)\\
&\geq (1-(C+2)2^{-C-1})H(Y) && (26)
\end{aligned}
$$

and we can set $S = \tilde{E}_h \cup E_l$ for the AND case.

Finally, consider the full problem. We apply lemma 3, and partition the bits of Y into sets $W_1, ..., W_{2^{2^b}}$ so that each set W_i only contains bits which are a single fixed function f_i of pairs of X. Now we construct approximation sets S_i for each of the sets W_i as before. Notice that we have $k^* = 8$ in remark 2 (we only lose entropy on the functions which are ANDs modulo negation). Thus there exists a set S' of size at most $8Cn + 8n \leq 8(C+1)n$ such that $H(Y|_{S'}) \geq (1 - 8(C+2)2^{-C-1})H(Y)$. Setting $C, C_\epsilon = \max(\mathcal{O}(\log(\frac{1}{\epsilon})), 4)$ gives us what we want.

□

1.3 Greedy is optimal, and a combinatorial interpretation

Theorem 6. *(Greedy is optimal up to factors in ϵ): Suppose there exists a set $S = \{S_1, ..., S_m\}$ such that $H(Y|_S) \geq (1-\epsilon)H(Y)$. Let $S' = \{S'_1, ..., S'_k\}$ be the set chosen as follows: at step i, find the index j such that $H(Y_j|Y_{S'_1}, ..., Y_{S'_{i-1}})$ is maximal, and add j to S'. Terminate when $H(Y|_{S'}) \geq (1-\epsilon)(1-\epsilon')H(Y)$. Then we have $k = \mathcal{O}\left(m \log\left(\frac{1}{\epsilon'}\right)\right)$.*

Proof. We claim that at step i, we have

$$
\max_j H(Y_j|Y_{S'_1}, ..., Y_{S'_{i-1}}) \geq \frac{(1-\epsilon)H(Y) - H(Y_{S'_1}, ..., Y_{S'_{i-1}})}{m} \tag{27}
$$

Suppose this did not hold. Then in particular we have

$$
\begin{aligned}
H(Y_{S_1}, ..., Y_{S_m}) &\leq H(Y_{S_1}, ..., Y_{S_m}|Y_{S'_1}, ..., Y_{S'_{i-1}}) + H(Y_{S'_1}, ..., Y_{S'_{i-1}}) && (28)\\
&\leq \sum_{i \in [m]} H(Y_{S_i}|Y_{S'_1}, ..., Y_{S'_{i-1}}) + H(Y_{S'_1}, ..., Y_{S'_{i-1}}) && (29)\\
&< (1-\epsilon)H(Y) - H(Y_{S'_1}, ..., Y_{S'_{i-1}}) + H(Y_{S'_1}, ..., Y_{S'_{i-1}}) && (30)\\
&= (1-\epsilon)H(Y) && (31)
\end{aligned}
$$

which gives a contradiction. Now we have

$$
\begin{aligned}
H(Y_{S'_1}, ..., Y_{S'_i}) &= H(Y_{S'_i}|Y_{S'_1}, ..., Y_{S'_{i-1}}) + H(Y_{S'_1}, ..., Y_{S'_{i-1}}) && (32)\\
&\geq \frac{(1-\epsilon)H(Y)}{m} + \left(1 - \frac{1}{m}\right) H(Y_{S'_1}, ..., Y_{S'_{i-1}}) && (33)\\
& && (34)
\end{aligned}
$$

Solving $f(i) = \frac{a}{m} + (1 - \frac{1}{m})f(i-1)$ and $f(0) = 0$ gives $f(k) = a(1 - (1 - \frac{1}{m})^k)$, so we have

$$H(Y_{S'_1}, ..., Y_{S'_k}) \geq (1-\epsilon)(1 - (1 - \frac{1}{m})^k)H(Y) \tag{35}$$

picking $k = \mathcal{O}\left(m \log\left(\frac{1}{\epsilon'}\right)\right)$ ensures that the LHS is $\geq (1-\epsilon)(1-\epsilon')H(Y)$

□

Corollary 7. *For any $\epsilon > 0$, there exists a set S of size $\mathcal{O}(C(\epsilon)n)$ such that $H(Y|_S) \geq (1-\epsilon)H(Y)$ iff for any $\epsilon > 0$ there exists a greedy set S' of size $\mathcal{O}(C'(\epsilon)n)$ such that $H(Y|_{S'}) \geq (1-\epsilon)H(Y)$*

Definition 8. *Let a weighted vertex graph $\mathcal{S}$ be a set of pairs (p_i, S_i), where $p_i \in [0,1], S_i \subset \{0,1\}^n$. We view each p_i as a weight for the subset of vertices S_i. At all times we have $\sum_i p_i = 1$ and that $\{S_i\}_i$ forms a partition of $\{0,1\}^n$.*

Definition 9. *Given a conditioning set $R_j \subset \{0,1\}^n$, where $R_j := \{x \in \{0,1\}^n | Y_j(x) = 1\}$, we define the weighted vertex graph $\mathcal{S} - R_j$ conditioned on R_j as follows: for each $(p_i, S_i) \in \mathcal{S}$, let $S_{i,0} = S_i \cap R_j^c$, $p_{i,0} = p_i \frac{|S_{i,0}|}{|S_i|}$ and $S_{i,1} = S_i \cap R_i$, $p_{i,1} = p_i \frac{|S_{i,1}|}{|S_i|}$. Now replace each (p_i, S_i) by the sets $(p_{i,0}, S_{i,0}), (p_{i,1}, S_{i,1})$ to form $\mathcal{S} - R_j$.*

Remark 10. *Let $\mathcal{S} = \Lambda := \{(1, \{0,1\}^n)\}$. Then we can view the distribution $X|Y_{i_1}, ..., Y_{i_k}$ as being "represented" by the weighted vertex graph $\mathcal{S} - R_{i_1} - ... - R_{i_k}$. Indeed, for each distribution $X|Y_{i_1} = y_{i_1}, ..., Y_{i_k} = y_{i_k}$, there exists an element $(p_j, S_j) \in \mathcal{S}$ such that $p_i = P(Y_{i_1} = y_{i_1}, ..., Y_{i_k} = y_{i_k})$ and $S_j = \{x \in \{0,1\}^n | Y_{i_1}(x) = y_{i_1}, ..., Y_{i_k}(x) = y_{i_k}\}$. We can draw a sample from the distribution of X by picking a set S_i with probability p_i, and then drawing uniformally from S_i.*

Lemma 11. *Let $\mathcal{S} = \{(p_i, S_i)\}_i = \Lambda - R_{i_1} - ... - R_{i_k}$. Then $H(X|Y_{i_1}, ..., Y_{i_k}) = \sum_i p_i \log(|S_i|)$ and $H(Y_{i_{k+1}}|Y_{i_1}, ..., Y_{i_k}) = \sum_i p_i \left[\frac{|S_{i,0}|}{|S_i|} \log\left(\frac{|S_{i,0}|+|S_{i,1}|}{|S_{i,0}|}\right) + \frac{|S_{i,1}|}{|S_i|} \log\left(\frac{|S_{i,0}|+|S_{i,1}|}{|S_{i,1}|}\right)\right]$. In particular, the remaining entropy $H(X|Y_{i_1}, ..., Y_{i_k})$ is the weighted average of the logarithm of the size of each group.*

Proof.

$$H(X|Y_{i_1}, ..., Y_{i_k}) = \sum_{y_{i_1}, ..., y_{i_k}} P(Y_{i_1} = y_{i_1}, ..., Y_{i_k} = y_{i_k}) \times \tag{36}$$

$$\left[\sum_x P(x|Y_{i_1} = y_{i_1}, ..., Y_{i_k} = y_{i_k}) \log\left(\frac{1}{P(x|Y_{i_1} = y_{i_1}, ..., Y_{i_k} = y_{i_k})}\right)\right] \tag{37}$$

$$= \sum_i p_i [\log(|S_i|)] \tag{38}$$

And

$$H(Y_{i_{k+1}}|Y_{i_1}, ..., Y_{i_k}) = H(X|Y_{i_1}, ..., Y_{i_k}) - H(X|Y_{i_1}, ..., Y_{i_{k+1}}) \tag{39}$$

$$= \sum_i (p_{i,0} + p_{i,1}) \log(|S_{i,0}| + |S_{i,1}|) - \sum_i (p_{i,0}) \log(|S_{i,0}|) - \sum_i (p_{i,1}) \log(|S_{i,1}|) \tag{40}$$

$$= \sum_i \left[p_{i,0} \log\left(\frac{|S_{i,0}| + |S_{i,1}|}{|S_{i,0}|}\right) + p_{i,1} \log\left(\frac{|S_{i,0}| + |S_{i,1}|}{|S_{i,1}|}\right)\right] \tag{41}$$

$$= \sum_i p_i \left[\frac{|S_{i,0}|}{|S_i|} \log\left(\frac{|S_{i,0}| + |S_{i,1}|}{|S_{i,0}|}\right) + \frac{|S_{i,1}|}{|S_i|} \log\left(\frac{|S_{i,0}| + |S_{i,1}|}{|S_{i,1}|}\right)\right] \tag{42}$$

$$= \sum_i p_i \left[\frac{|S_i \cap R^c_{i_{k+1}}|}{|S_i|} \log\left(\frac{|S_i \cap R^c_{i_{k+1}}| + |S_i \cap R_{i_{k+1}}|}{|S_i \cap R^c_{i_{k+1}}|}\right)\right] \tag{43}$$

$$+ \sum_i p_i \left[\frac{|S_i \cap R_{i_{k+1}}|}{|S_i|} \log\left(\frac{|S_i \cap R^c_{i_{k+1}}| + |S_i \cap R_{i_{k+1}}|}{|S_i \cap R_{i_{k+1}}|}\right)\right] \tag{44}$$

□

Remark 12. *Suppose each Y_i depends on at most b bits. Then each R_i is of the form $\{*...*i_1*...*i_{i_2}*...*...*i_b*...*\}$, i.e. the set of all n bit strings where each $*$ varies over $\{0,1\}$, and $(i_1, ..., i_b)$ takes values in a set $F \subset \{0,1\}^b$. In particular, if Y_i is not constant, then $2^{n-b} \leq |R_i|, |R_i^c| \leq 2^n - 2^{n-b}$.*

Moreover, we know from a previous lemma that we can assume all the Y_i's consist of the same function f on b bits of X. In this particular case, the forms of the R_i's are even simpler: for each i, j, there exists a function $\sigma_{i,j} : \{0,1\}^n \to \{0,1\}^n$ which permutes at most b bit positions, such that $x \in R_i$ iff $\sigma_{i,j}(x) \in R_j$.

Corollary 13. *Suppose we have random variables $Y_1, ..., Y_w$ with corresponding conditioning sets $R_1, ..., R_w$, where each Y_i is a function of X. Then the condition that $H(Y_{i_1}, ..., Y_{i_k}) \geq (1-\epsilon)H(Y_1, ..., Y_w)$ is equivalent to*

$$\sum_i p_i^{S'} \log(|S_i'|) \leq \epsilon n + (1-\epsilon) \sum_i p_i^S \log(|S_i|) \tag{45}$$

where $S = \{(p_i^S, S_i)\}_i = \Lambda - R_1 - ... - R_w$ and $S' = \{(p_i^{S'}, S_i')\}_i = \Lambda - R_{i_1} - ... - R_{i_k}$.

Proof. Write $Y := (Y_1, ..., Y_w)$, $\tilde{Y} := (Y_{i_1}, ..., Y_{i_k})$. Then we have the following sequence of equivalent conditions

$$\sum_i p_i^{S'} \log(|S_i'|) \leq \epsilon n + (1-\epsilon) \sum_i p_i^S \log(|S_i|) \tag{46}$$
$$H(X|\tilde{Y}) \leq \epsilon H(X) + (1-\epsilon)H(X|Y) \qquad \text{(lemma 11)} \tag{47}$$
$$H(X|\tilde{Y}) - H(X|Y) \leq \epsilon\,[H(X) - H(X|Y)] \tag{48}$$
$$(H(X) - H(\tilde{Y})) - (H(X) - H(Y)) \leq \epsilon\,[H(X) - (H(X) - H(Y))] \tag{49}$$
$$H(Y) - H(\tilde{Y}) \leq \epsilon H(Y) \tag{50}$$
$$H(\tilde{Y}) \geq (1-\epsilon)H(Y) \tag{51}$$

□

Remark 14. *The previous corollary shows that we can think of $\tilde{Y} = (Y_{i_1}, ..., Y_{i_k})$ as being a "covering" of $Y = (Y_1, ..., Y_w)$ which achieves a score $\sum_i p_i^{S'} \log(|S_i'|)$ (which we are trying to minimize). Likewise, Y achieves a score of $\sum_i p_i^S \log(|S_i|)$. If the score of $\tilde{Y}$ is close enough to the score of Y, then we know $H(\tilde{Y}) \geq (1-\epsilon)H(Y)$.*

Task: show that greedy always gives an $\epsilon H(Y)$ approximation of $H(Y)$ (using corollary 13) with a $C'(\epsilon)n$ sized covering, or give a counter example. Take note of remark 12: we can assume that each bit Y_i is a fixed function f of some b bits of X, and so the sets $\{R_i\}$ have a particularly simple form. My intuition is that we want to use the fact that the size of each R_i is large, and only "fixes" a finite number of coordinates. We then probably want to make some kind of "we can always decrease the entropy $H(Y|Y_{i_1}, ..., Y_{i_k})$ by a constant factor" argument, similar to the proof in theorem 6. We know by corollary 7 and theorem 5 that this claim holds when $b = 2$. My intuition is that a proof for greedy using the vertex covering language will not use the fact that $b = 2$, and so will hopefully generalize to $b = \mathcal{O}(1)$.

2 Converting bit sources locally

This is a more direct way of approaching the motivating problem in section 1.1: how does one locally convert an iid unbiased bit source X into a source of n' bits Y, where Y is statistically close to a collection of iid coin flips with bias say $\frac{1}{4}$, and n' is close to the optimal $n/H(\frac{1}{4})$ in expectation.

Lemma 15. *There exists an algorithm E which takes in $bH(p) + 1$ (in expectation) unbiased bits at a time, and returns b bits. Moreover, $KL(D(E), Bern(b,p)) \leq 1$, where $Bern(b,p)$ are b i.i.d. Bernoulli random variables with bias p, and $D(E)$ is the distribution of an output block of E.*

Proof. Let $q_x = P_{X \sim Bern(b,p)}[X = x]$. For each $x \in \{0,1\}^b$, let $l_x = \lceil -\log_2(q_x) \rceil$. The lengths $\{l_x\}_x$ satisfy the kraft inequality, so there exists a prefix code with these lengths where each codeword c_x of length l_x maps to x via $E(c_x) = x$. Let $p_x = 2^{-l_x}$ be the probability of drawing codeword c_x. We have

$$KL(D(E), Bern(b,p)) = \sum_x p_x \log\left(\frac{p_x}{q_x}\right) \tag{52}$$
$$= \sum_x p_x \left[\log(p_x) - \log(q_x)\right] \tag{53}$$
$$= \sum_x p_x \left[\lceil \log(q_x) \rceil - \log(q_x)\right] \tag{54}$$
$$\leq 1 \tag{55}$$

A similar calculation shows that $E[\text{codeword length}] = \sum_x p_x l_x = \sum_x 2^{-\lceil -\log_2(q_x) \rceil} \lceil \log\left(\frac{1}{q_x}\right) \rceil \leq H(q) + 1 = bH(p) + 1$.

□

Remark 16. *The +1 bound in the KL divergence might be overly pessimistic: this really depends on how tightly we can partition subsets of* $\{0,1\}^n$ *such that the subsets of the partition are close to (negative) powers of 2 in probability. It might be worth exploring explicit examples to see how tight we can get this (I think looking at things this low-level could also lead to bounds in statistical difference).*

Remark 17. *Another thing to try (instead of trying to make optimal symbol codes directly) is to consider a kind of truncated geometric distribution, i.e.* $P[x = 0...01] \propto \left(\frac{3}{4}\right)^{\textit{number of zeroes}} \frac{1}{4}$ *when the number of zeroes is less than say* $\mathcal{O}(\log(n))$*, otherwise x will simply be* $\mathcal{O}(\log(n))$ *zeroes. Then we can think of generating our sequence of biased coins by continually drawing i.i.d. values for x and concatenating them together. I'm not really sure whether this has any advantages over the previous approach, but it might allow us to partition the x's in a way which is more easy to analyze/break into powers of two.*

2.1 l2 entropy bound using fourier analysis

Here, we wanted to try and see whether specific construction would work as a candidate for converting an iid bit source X into a bit source Y.

The construction is as follows: for each $i \in [n']$, randomly pick indices $G(i,1), G(i,2), G(i,3), G(i,4), G(i,5), G(i,6) \in [n]$ and then set $Y_i := (x_{G(i,1)} + x_{G(i,2)} + x_{G(i,3)})(x_{G(i,4)} + x_{G(i,5)} + x_{G(i,6)})$. This gives the right expectation: Y_i is 1 with probability $\frac{1}{4}$. The hope was because this has a kind of "expander graph" like property, the bits would also be reasonably independent. The strategy for determining whether this would work is to try and upper bound the $l1$ norm of Y and the target distribution by the $l2$ norm. The $l2$ norm is easier to analyze with fourier analysis tricks. We were (unfortunately) able to show that this analysis approach doesn't work.

Let μ be the $\frac{1}{4}$ biased distribution on m bits, and v_G the distribution we construct from a graph G. To draw y_i, we first draw $x \sim \{0,1\}^n$. We then let $y_i = (x_{G(i,1)} + x_{G(i,2)} + x_{G(i,3)})(x_{G(i,4)} + x_{G(i,5)} + x_{G(i,6)})$. If we fix x and choose $G(i,j)$ uniformally at random, then each $x_{G(i,j)}$ are iid bernoulli random variables with mean equal to the mean number of 1's in the fixed x. We have

$$\sum_{y\in\{0,1\}^m} (P_v(y) - P_\mu(y))^2 = 2^m E_{y\sim\{0,1\}^m}[(P_v(y) - P_\mu(y))^2] \tag{56}$$
$$= 2^m \sum_{S\subset[m]} [(\widehat{P}_v(S) - \widehat{P}_\mu(S))^2] \tag{57}$$
$$= 2^m \sum_{S\subset[m]} [(E_{y\sim\{0,1\}^m}[(-1)^{\sum_{i\in S} y_i} P_v(y)] - E_{y\sim\{0,1\}^m}[(-1)^{\sum_{i\in S} y_i} P_\mu(y)])^2] \tag{58}$$
$$= 2^m \sum_{S\subset[m]} [(2^{-m} E_{y\sim v}[(-1)^{\sum_{i\in S} y_i}] - 2^{-m} E_{y\sim\mu}[(-1)^{\sum_{i\in S} y_i}])^2] \tag{59}$$
$$= 2^{-m} \sum_{S\subset[m]} [(E_{x\sim\{0,1\}^n}[(-1)^{\sum_{i\in S} y_i(x)}] - E_{y\sim\mu}[(-1)^{\sum_{i\in S} y_i}])^2] \tag{60}$$

Fix an $S \subset [m]$. We have that

$$E_{y\sim\mu}[(-1)^{\sum_{i\in S} y_i}] = \left(\frac{1}{2}\right)^{|S|} \tag{61}$$

And

$$E_G\left[(E_{x\sim\{0,1\}^n}[(-1)^{\sum_{i\in S} y_i(x)}] - E_{y\sim\mu}[(-1)^{\sum_{i\in S} y_i}])^2\right] = E_G\left[E_{x\sim\{0,1\}^n}[(-1)^{\sum_{i\in S} y_i(x)}]^2\right] - 2E_G\left[E_{x\sim\{0,1\}^n}[(-1)^{\sum_{i\in S} y_i(x)}]\right]\left(\frac{1}{2}\right)^{|S|} + \left(\frac{1}{2}\right)^{2|S|} \tag{62}$$

$$E_G E_{x\sim\{0,1\}^n}[(-1)^{\sum_{i\in S} y_i(x)}] = E_{x\sim\{0,1\}^n} E_G[(-1)^{\sum_{i\in S} y_i(x)}] \tag{63}$$

$$= \sum_{a=0}^{n} P[x \text{ has } a \text{ 1's}] E_G[(-1)^{\sum_{i\in S} y_i(x)} | x \text{ has } a \text{ 1's}] \tag{64}$$

$$= \sum_{a=0}^{n} P[x \text{ has } a \text{ 1's}] E_G[(-1)^{y_1(x)} | x \text{ has } a \text{ 1's}]^{|S|} \tag{65}$$

$$= 2^{-n} \sum_{a=0}^{n} \binom{n}{a} \tau\left(\frac{a}{n}\right)^{|S|} \tag{66}$$

where $\tau(p) := 1 - 18p^2 + 72p^3 - 120p^4 + 96p^5 - 32p^6$, or equivalently $\tau(\frac{1}{2} + d) = 1/2 - 8d^3 - 32d^6$.

Quick aside

Using $E[x^2] \geq E[x]^2$, we get that

$$(62) \geq \left[2^{-n} \sum_{a=0}^{n} \binom{n}{a} \tau\left(\frac{a}{n}\right)^{|S|} - \left(\frac{1}{2}\right)^{|S|}\right]^2 \tag{67}$$

$$\tag{68}$$

So that

$$E_G \sum_{y\in\{0,1\}^m} (P_v(y) - P_\mu(y))^2 \geq 2^{-m} \sum_{S\subset[m]} \left[2^{-n} \sum_{a=0}^{n} \binom{n}{a} \tau\left(\frac{a}{n}\right)^{|S|} - \left(\frac{1}{2}\right)^{|S|}\right]^2 \tag{69}$$

$$= 2^{-m} \sum_{k=0}^{m} \binom{m}{k} \left[2^{-n} \sum_{a=0}^{n} \binom{n}{a} \tau\left(\frac{a}{n}\right)^{k} - \left(\frac{1}{2}\right)^{k}\right]^2 \tag{70}$$

This is a lower bound on the l_2 norm squared. If we assumed this bound were tight (the optimistic case), the upper bound on the l_1 norm we would get is

$$2^{m/2} \left[2^{-m} \sum_{k=0}^{m} \binom{m}{k} \left[2^{-n} \sum_{a=0}^{n} \binom{n}{a} \tau\left(\frac{a}{n}\right)^{k} - \left(\frac{1}{2}\right)^{k}\right]^2\right]^{\frac{1}{2}} \tag{71}$$

$$= \left[\sum_{k=0}^{m} \binom{m}{k} \left[2^{-n} \sum_{a=0}^{n} \binom{n}{a} \tau\left(\frac{a}{n}\right)^{k} - \left(\frac{1}{2}\right)^{k}\right]^2\right]^{\frac{1}{2}} \tag{72}$$

Here are some plots of this bound for different relations between n and m, which show that this bound diverges for the parameter ranges we care about.

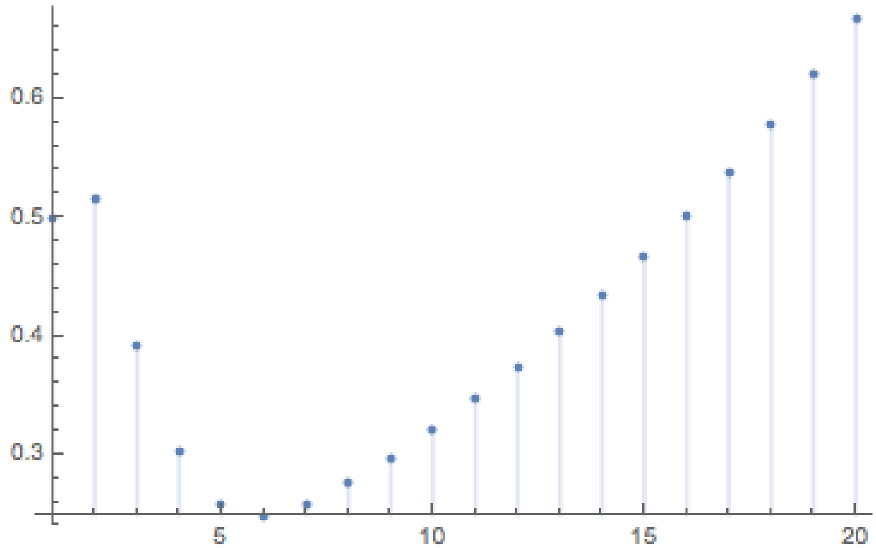

Figure 1: Plot of bound for $m = n$

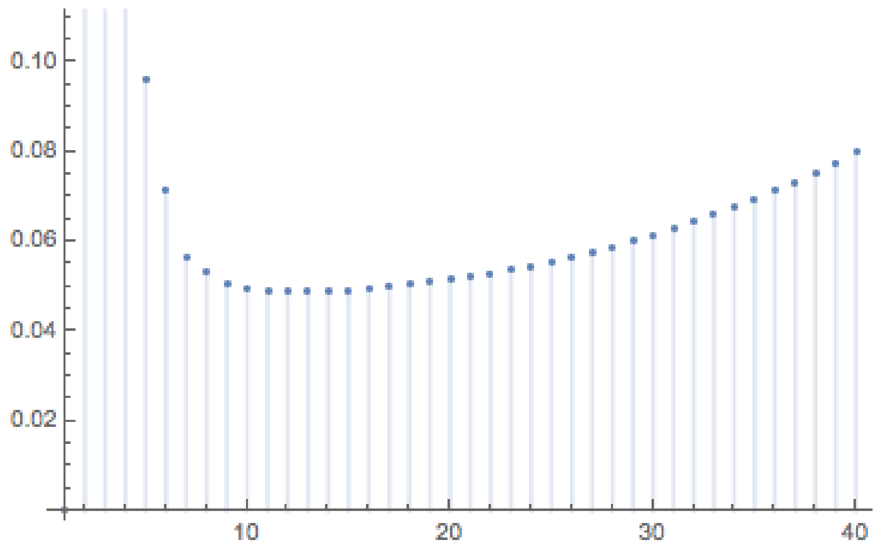

Figure 2: Plot of bound for $m = n/2$

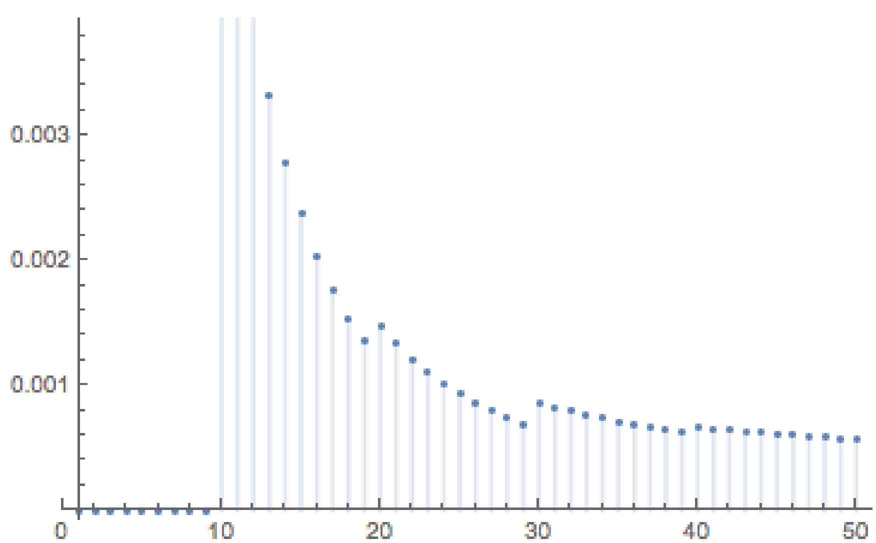

Figure 3: Plot of bound for $m = n/10$