1. Extractors

(a) In this question you are asked to show that randomness extraction is possible only from sources that are statistically close to sources with high min-entropy; thus, high min-entropy is both sufficient and necessary for randomness extraction. More formally, let $X$ be any distribution over $\{0,1\}^n$ and let $Ext : \{0,1\}^n \times \{0,1\}^d \rightarrow \{0,1\}^m$. Suppose that $Ext(X,U_d)$ is $\epsilon$-close to the uniform distribution $U_m$. Prove that $X$ is $O(\epsilon)$-close to some $k$-source $X'$ where $k \geq m - d - 1$. (Hint: Consider the set $A$ of strings $z \in \{0,1\}^m$ that get assigned probability more than $2 \times 2^{-m}$ by $Ext(X,U_d)$. Argue that the set $A$ gets the probability at most $2^\epsilon$ with respect to the distribution $Ext(X,U_d)$. Fix the seed $y \in \{0,1\}^d$ so that $Pr[Ext(X,y) \in A] \leq 2\epsilon$. Argue that for every $x$ such that $Ext(x,y) \notin A$, we have $Pr[X = x] \leq 2^{-m-d-1}$. Conclude that there exists a $k$-source $X'$ such that $\Delta(X,X') \leq 2\epsilon$.)

Solution: Following the hint, we get by the definition of extractor that $Pr[Ext(X,U_d) \in A] \leq \mu(A) + \epsilon$, where $\mu(A) = |A|/2^m$ is the density of the set $A$. On the other hand, by the definition of $A$, $Pr[Ext(X,U_d) \in A] \geq |A|2 \times 2^{-m} = 2\mu(A)$. Combining these two inequalities we get $\mu(A) \leq \epsilon$ and hence $Pr[Ext(X,U_d) \in A] \leq 2\epsilon$.

By averaging, there exists a string $y$ such that $Pr[Ext(X,y) \in A] \leq 2\epsilon$. Let $x$ be any string such that $z = Ext(x,y) \notin A$. If $Pr[X = x] > 2^{-m-d-1}$, then the pair $(x,y)$ gets the probability greater than $2^{-m-d-1} \times 2^d = 2 \times 2^{-m}$ according to $(X,U_d)$. This means that $z = Ext(x,y)$ gets probability greater than $2 \times 2^{-m}$ according to $Ext(X,U_d)$, and hence, by the definition of $A$, $z$ must be in $A$, which contradicts our earlier choice of $z \notin A$. Thus, for every such string $x$, we must have $Pr[X = x] \leq 2^{-m-d-1}$.

Finally, define the distribution $X'$ to be an arbitrary flat $k$-source for $k = m - d - 1$. We have $\Delta(X,X') = Pr[X \in T] - Pr[X' \in T] \leq Pr[X \in T]$ for $T = \{z \mid Pr[X = z] > Pr[X' = z]\}$ (this is easy to verify using the definition of the statistical distance $\Delta(X,X') = \max_T\{Pr[X \in T] - Pr[X' \in T]\}$). Now, observe that in our case, $T \subset \{x \mid Ext(x,y) \in A\}$. Hence, $Pr[X \in T] \leq Pr[Ext(X,y) \in A] \leq 2\epsilon$, and so, $\Delta(X,X') \leq 2\epsilon$.

(b) Show that every $k$-source $X$ over $\{0,1\}^n$, for large $k$, can be viewed as a block source $X = YZ$. More precisely, let $X = YZ$ be an $(n - \Delta)$-source, for some $\Delta$, where $Y$ is a distribution over $\ell$-bit strings for any $\ell \leq n$, and $Z$ is a distribution over strings of length $m = n - \ell$. Prove that $Y$ is a $(\ell - \Delta)$-source. Prove also that, for every $\epsilon > 0$, with probability at least $(1 - \epsilon)$ over the choice of $y$ according to the distribution $Y$, the conditional distribution $Z | Y = y$ is an $(m - \Delta - \log(1/\epsilon))$-source.

Solution: For any $y \in \{0,1\}^\ell$, we have $Pr[Y = y] = \sum_{x \in \{0,1\}^n} Pr[X = yz] \leq 2^m 2^{-(n-\Delta)} = 2^{-(\ell-\Delta)}$, where the last inequality is due to the condition that $X$ is an $(n - \Delta)$-source. Thus, we know that $Y$ is an $(\ell - \Delta)$-source.
Now, suppose that for more than $\epsilon$ of $ys$ according to $Y$, we have the existence of $z$ such that $Pr[Z|Y=z] > 2^\Delta/(\epsilon 2^m)$. Let $B$ denote the set of all such $ys$. Since the set $B$ gets probability greater than $\epsilon$ in $Y$, there must exist at least one $y_0 \in B$ such that $y_0$ gets probability greater than $\epsilon/|B| \geq \epsilon/2^\ell$ (the last inequality is due to $|B| \leq 2^\ell$). Let $z_0$ be such that $Pr[Z|Y=y_0 = z_0] > 2^\Delta/(\epsilon 2^m)$. Then $Pr[X = y_0 z_0] = Pr[Y = y_0] Pr[Z = z_0 | Y = y_0] > (\epsilon/2^\ell) 2^\Delta/(\epsilon 2^m) = 2^{-(n-\Delta)}$, contradicting the assumption that $X = YZ$ is an $(n-\Delta)$-source.

2. Error reduction in BPP algorithms, using extractors Let $A$ be any BPP algorithm that on input of length $\ell$ uses $m$ random bits, and has some constant error probability (say, 1/4). Using a $(k, \epsilon)$ extractor $Ext : \{0,1\}^n \times \{0,1\}^d \rightarrow \{0,1\}^m$ for appropriate parameters $n, k, d$, show how to reduce the error probability in the algorithm $A$ to less than $2^{-t}$, for any $t = poly(\ell)$, while using at most $m + t$ random bits. Your new randomized algorithm should still run in polytime. Conclude that that every BPP algorithm $A$ has an equivalent BPP algorithm $A'$ using $r$ random bits such that $A'$ errs on at most $2^{\sqrt{r}}$ of all $r$-bit random strings.

(Hint: Your algorithm $A'$ will pick a string $z \in \{0,1\}^n$ uniformly at random, and output the majority decision of $A$ when $A$ uses the strings $Ext(z, s)$ instead of its random strings, over all seeds $s \in \{0,1\}^d$. Analyze the error probability of this algorithm $A'$, and pick the extractor parameters $n, k, d, \epsilon$ appropriately.)

Solution: For this problem, we need to assume that an optimal extractor can be explicitly constructed. Let $Ext' : \{0,1\}^n \times \{0,1\}^d \rightarrow \{0,1\}^{m'}$ be an optimal extractor for min-entropy $k = m, n = m + t$, the error $\epsilon = 1/5$, and $d$ and $m'$ determined by the chosen parameters $k, n, \epsilon$. Note that, for an optimal extractor, we will have $d = log(n-k)+2log(1/\epsilon)+O(1) = log t + O(1)$ and $m' = k + d - 2log(1/\epsilon) - O(1) = m + d - O(1)$ which is at least $m$ for $d$ large enough (i.e., for $t$ large enough).

Assuming that $t$ is sufficiently large, we define $Ext(x, s) = Ext'(x, s)|_{1...m}$; that is, we define the output of $Ext$ to be the first $m$ bits of the output of $Ext'$. Observe that if $Ext'$ was a $(k, \epsilon)$-extractor, then $Ext$ is also a $(k, \epsilon)$-extractor. (Otherwise, a statistical test $T \subseteq \{0,1\}^m$ $\epsilon$-distinguishing an output of $Ext$ from uniform would immediately yield a statistical test $T' \subseteq \{0,1\}^{m'}$ $\epsilon$-distinguishing the output of $Ext'$ from uniform, where $T' = \{xy \mid x \in T, y \in \{0,1\}^{m'-m}\}$.) This $(m, 1/5)$-extractor $Ext : \{0,1\}^{m+t} \times \{0,1\}^d \rightarrow \{0,1\}^m$ will now be used to reduce the error in our BPP algorithm $A$ using the algorithm $A'$ described in the Hint. Note that for any $t = poly(\ell)$, we have $d = O(log \ell)$, and hence $A'$ will run in time polynomial in $\ell$.

Fix an arbitrary input $x$ to the algorithm $A$. Let $B$ be the set of all those $m$-bit strings $r$ where $A(x, r)$ is incorrect. By our assumption on the error of $A$, we have $\mu(B) \leq 1/4$, where $\mu(B) = |B|/2^m$ is the density of $B$. Let $C$ be the set of all those $n$-bit strings $z$ such that more than half of the seeds $s$ of the extractor $Ext$ result in a string $Ext(z, s) \in B$. In other words, $C$ is the set of strings where the algorithm $A'$ described in the Hint will make an error. We’ll show that $|C| < 2^k$.

Indeed, suppose that $|C| \geq 2^k$. By definition of $C$, we have $Pr[Ext(C, U_d) \in B] \geq 1/2$. On the other hand, by the definition of $Ext$ (since the uniform distribution on $C$ is a flat source with at least $k$ bits of min-entropy), we know that $Pr[Ext(C, U_d) \in B] \leq \mu(B) + \epsilon \leq 1/4+1/5 < 1/2$, which is a contradiction.

So, the probability that $A'$ will make an error is at most $|C|/2^m < 2^{-(n-k)}$, which is $2^{-t}$ by our choice of parameters in $Ext$. 


3. Error-correcting codes based on expanders

Let $G = (L, R, E)$ be a bipartite $(\alpha n, (1 - \epsilon)d)$-expander on $(n, m)$ vertices for $m < n$, with left degree $d$, for a constant $d$; that is, every set $S \subseteq L$ of size at most $\alpha n$ is expanded by a factor $(1 - \epsilon)d$. (Such “lossless” expanders can be constructed explicitly, using the “extractor technology”.) Assume that $\epsilon < 1/12$. The graph $G$ defines a binary error-correcting code $C \subset \{0,1\}^n$ as follows: A string $c \in \{0,1\}^n$ is a codeword if, for every node $i \in R$ with neighbours $j_1, \ldots, j_k \in L$, we have $c_{j_1} \oplus \cdots \oplus c_{j_k} = 0$, where $\oplus$ is addition modulo 2. That is, we view the nodes in $L$ as positions in an $n$-bit string $c$, and nodes in $R$ as parity-check constraints, where the constraint corresponding to vertex $i \in R$ checks $c$ in the positions determined by the neighbours of $i$ in $L$; a string $c$ is a codeword if all $m$ parity check constraints are satisfied.

(a) Consider a codeword $c \in C$ of minimum Hamming weight (i.e., with minimum number of 1’s). Show that the Hamming weight of this codeword $c$ is greater than $\alpha n$ (and hence, the minimum relative distance of the code $C$ is greater than $\alpha$). (Hint: Prove and then use the following fact: every set $S \subseteq L$ of size at most $\alpha n$ has at least $(1 - 2\epsilon)d|S|$ unique neighbours, where a vertex $v \in R$ is a unique neighbour for $S$ if $v$ is connected by an edge to exactly one node in $S$.)

Solution: First we show that every set $S$ of size at most $\alpha n$ has at least $(1 - 2\epsilon)d|S|$ unique neighbours. Consider all $d|S|$ neighbours of $S$, with some vertices possibly repeated. Since the graph has expansion factor $(1 - \epsilon)d$, we get that at most $\epsilon d$ fraction of neighbours of $S$ can be repeats. Since the size of $N(S)$ is at least $(1 - \epsilon)d|S|$, we get that at least $(1 - \epsilon)d|S| - \epsilon d|S| = (1 - 2\epsilon)d|S|$ nodes in $N(S)$ are unique neighbours.

Now, suppose that some codeword $c$ of weight $w < \alpha n$ exists. Then, by the just proved, there will be $(1 - 2\epsilon)dw$ (which is bigger than 0 for $\epsilon < 1/2$) constraints with exactly one variable falling among the $w$ 1s of the vector $c$ (and the other variables falling on the 0 components of $c$). But this means that this constraint equals 1, and hence is not satisfied. This contradicts the assumption that $c$ is a codeword. Thus, the minimum weight of any nonzero codeword $c$ must be greater than $\alpha n$.

(b) Prove that the following decoding algorithm for $C$ corrects $(1 - 3\epsilon)\alpha n$ errors in $O(n \log n)$ time.

Let $m \in \{0,1\}^n$ be a received message. Label the nodes in $L$ with the bits of the string $m$ (so that node $i \in L$ gets the label $m_i$). Until all the parity checks are satisfied, repeat the following: in parallel, each node $i \in L$ flips its value if the number of unsatisfied parity checks among its $d$ neighbours is at least $2d/3$; otherwise, the node $i$ retains its old value.

You should fill in the details in the proof outline below.

i. Let $S \subseteq L$ be a set of error positions at the beginning of a parallel round. Let $N(S) \subseteq R$ be the set of neighbours of $S$. If $|S| \leq \alpha(1 - 3\epsilon)n$, then $S$ has at least $(1 - 2\epsilon)d|S|$ unique neighbours in $R$ (by the previous question). By an averaging argument, show that at least $(1 - 6\epsilon)$ fraction of nodes in $S$ will have at least $2d/3$ unique neighbours in $R$. Conclude that at least $(1 - 6\epsilon)|S| \geq |S|/2$ nodes in $S$ will correct their labels.

Solution: Let $\alpha$ be the fraction of nodes in $S$ that have fewer than $2d/3$ unique neighbours. Then the total number of unique neighbours of $S$ is at most $\alpha |S| 2d/3 + \alpha n \leq \alpha n |S|/2$, as required.
(1 – α)|S|d = (1 – α/3)|S|d. On the other hand, this number must be at least
(1 – 2ε)|S|d. From the inequality 1 – α/3 ≥ 1 – 2ε, we get that α ≤ 6ε. Hence, each of at least 1 – 6ε fraction of nodes in S has a at least 2d/3 unique neighbours.
For each such node with 2d/3 unique neighbours, our algorithm will flip its value since a constraint associated with a unique neighbour of S is unsatisfied. Thus, for ε < 1/12, at least (1 – 6ε) ≥ 1/2 of error positions S will get corrected after one parallel round.

ii. Let T ⊆ L \ S be the set of positions outside S that have correct labels before the parallel round, but then incorrectly flip their values during the round. Prove that each node in T has at least 2d/3 of its neighbours inside the set N(S).

Solution: Consider any node v ∈ T. Let u ∈ R be its neighbour, which is associated with some parity check constraint. If all neighbours of u in the set L are outside of S, then the constraint associated with u is satisfied. On the other hand, we know that the node v must have at least 2d/3 of its neighbour constraints unsatisfied (since it was flipped by our algorithm). This means that at least 2d/3 of the neighbours of v must have neighbours in the set S.

iii. Using the result of the previous item, show that |T| ≤ 3|S|/12. (Hint: The idea is that if T is large, then T ∪ S should expand significantly. But, since a lot of neighbours of T are already in N(S), no large expansion of T ∪ S is possible.)

Solution: Our proof is by contradiction. Suppose that |T| > 3|S|/12. Take any subset T′ ⊆ T of size exactly 3|S|/12. Note that, since |S| ≤ (1 – 3ε)αn, we get by the above that |T′| ≤ 3αn, and so |S ∪ T′| ≤ αn.

Since |S ∪ T′| ≤ αn, we get by the expansion property of our graph that |N(S ∪ T′)| ≥ (1 – ε)d(|S ∪ T′|) = (1 – ε)d(|S| + |T′|) (the last equality is due to disjointness of S and T′).

On the other hand, since at least 2d/3 neighbours of each node in T is already in N(S), we get that T′ has at most d|T′|/3 of its neighbours outside of N(S), and so |N(S ∪ T′)| ≤ |N(S)| + d|T′|/3 ≤ |S| + d|T′|/3 = d(|S| + |T′|/3). Combining these upper and lower bounds on |N(S ∪ T′)|, we get (1 – ε)(|S| + |T′|) ≤ |S| + |T′|/3, which solves to |T′| ≤ 3|S|/2 – 3ε.

Finally, recalling that |T′| = 3|S|/12, we get that 3|S|/12 ≤ 3|S|/2 – 3ε, which is a contradiction. So, we must have |T| ≤ 3|S|/12, and since ε < 1/12 we get |T| ≤ 3|S|/12 = |S|/3.

iv. Conclude that each parallel round increases the number of correct positions of the message m by at least |S|/6, and so after O(log n) steps all incorrect positions will be eliminated.

Solution: Suppose we have |S| errors before a parallel round. By item (i) above, we know that the number of “old” errors after one round is at most |S|/2. By item (iii), the number of “new” errors after the same round is at most |S|/3. Thus, the total number of errors after one parallel round is at most 5|S|/6. Since |S| ≤ n, after at most O(log n) rounds the number of errors will drop to 0.