Carnegie Mellon University
15-859: Information Theory and Applications in TCS Spring 2013

PROBLEM SET 3
Due by Thursday, March 28

INSTRUCTIONS

- You are allowed to collaborate with up to two other students taking the class in solving problem sets. But here are some rules concerning such collaboration:
  1. You should think about each problem by yourself for at least 30 minutes before commencing any collaboration.
  2. Collaboration is defined as discussion of the lecture material and solution approaches to the problems. Please note that *you are not allowed to share any written material and you must write up solutions on your own without any "collaboration notes" as an aid.*
  3. You must clearly acknowledge your collaborator(s) in the write-up of your solutions.
  4. Of course, if you prefer, you can also work alone (see the last bullet item for some "credit" for doing so).
- Solutions typeset in LaTeX are strongly preferred.
- You should *not* search for solutions on the web. More generally, you are urged to try and solve the problems without consulting *any* reference material other than the course notes and what we cover in class. If for some reason you feel the need to consult some source, *please acknowledge the source* and try to articulate the difficulty you couldn't overcome before consulting the source and how it helped you overcome that difficulty. Alternatively, before turning to any such material, we encourage you to ask us for hints or clarifications.
- Please start work on the problem set early. The problem set has **four** problems and is worth a total of 100 points. As a rather rough guess/estimate, scoring around 80% of the points, or 70% of the points if you work by yourself, might suffice for an A on this problem set.

---

1. *No bit reversal please* (20 points)

   The polarization property from class stated the following. If U_0^{N-1} is a uniform N-bit string for $N = 2^n$, $X_0^{N-1} = G_2^{\otimes n} B_n U_0^{N-1}$, and Y_0^{N-1} denotes the outputs of a binary input symmetric memoryless channel W on inputs X_0^{N-1} is, then $\forall \epsilon > 0$,

   $$\lim_{n\to\infty} \Pr_i[H(U_i|U_0^{i-1}, Y_0^{N-1}) \in (\epsilon, 1-\epsilon)] = 0 \,.$$

   Here $G_2 = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$ and B_n is the permutation matrix for the n-bit bit reversal permutation.

   Prove that the same claim holds even without the bit reversal permutation in the encoding, i.e., when $X_0^{N-1} = G_2^{\otimes n} U_0^{N-1}$

   Hint: Prove that $G_2^{\otimes n}$ and B_n commute, and then argue why this implies the claim.

2. *Squaring and Doubling Bhattacharyya* (25 points)

   Recall the 2-dimension linear transformation $(X_0, X_1) \to (U_0, U_1)$ defined in lecture: $U_0 = X_0 \oplus X_1$ and $U_1 = X_1$. Let Y_i be a random variable correlated with X_i such that (X_0, Y_0) and (X_1, Y_1) are i.i.d (in our setting, Y_i is the output of some memoryless channel W when input bit is X_i).

   For a correlated random variables (A, B) with A taking values in $\{0, 1\}$ and joint distribution $p_{AB}(a, b)$, define the quantity

$$Z(A \mid B) = 2 \sum_{b \in \mathrm{supp}(B)} \sqrt{p_{AB}(0, b) p_{AB}(1, b)}$$

   where $\mathrm{supp}(B)$ is the support of B.

   Prove that

   (a) $Z(U_0 \mid (Y_0, Y_1)) \leq 2Z(X_0 \mid Y_0)$.

   (b) $Z(U_1 \mid (Y_0, Y_1, U_0)) = Z(X_0 \mid Y_0)^2$.

   Note: These bounds provide the basis for the generalization of our analysis for the BEC case to general channels. Instead of tracking the probabilility that say U_i is not determined given Y_0^{N-1} and U_0^{i-1}, we track the value $Z(U_i \mid Y_0^{N-1}, U_0^{i-1})$ which evolve recursively following above constraints.

3. *Moser re-analyzed* (30 points)

   In this problem, we will revisit the result about satisfiability of bounded-overlap k-SAT from lecture, and obtain a quantitatively better bound. Specifically, you will prove that the following algorithm will always terminate in expected polynomial time and find a satisfying assignment for any instance $\mathcal{I}$ of k-SAT (assume for convenience that each clause has exactly k literals) in which each clause intersects at most $\Delta < 2^k/e$ clauses (including itself). Note that this is sharper than the $2^{k-O(1)}$ upper bound on overlap we argued in class.

   (a) Pick a random assignment to the variables by setting each variable to be 0 or 1 independently with probability $1/2$ each.

   (b) While there is some clause that is not satisfied:
   - Pick an arbitrary clause that is not satisfied, and assign fresh random values to all its variables.

   To analyze the algorithm, consider the sequence of unsatisfied clauses visited by the algorithm, $C_1, C_1, \ldots$ (where C_i, C_j for $i \neq j$ may refer to the same clause). We visualize part of the execution of the algorithm for t steps as a tree T_t labeled by clauses. The tree $T_t = T_t^{(1)}$ is constructed iteratively as follows: For $i = t$ down to 1:

   - $T_t^{(t)}$ consists of the single node C_t
   - For $1 \leq i < t$, $T_t^{(i)}$ is derived from $T_t^{(i+1)}$ as follows: If C_i does not intersect any of the clauses in $T_t^{(i+1)}$, then $T_t^{(i)} = T_t^{(i+1)}$. Otherwise, C_i is appended to the clause C_j deepest in the tree $T_t^{(i+1)}$ sharing a variable with C_i.

Imagine the random bits used by the algorithm as arranged in a matrix with M rows (think of M as very large) and n columns indexed by the variables. Each time the algorithm needs a fresh random value for a variable, it uses the first unused entry in the corresponding column.

(i) Argue that if two clauses C_i and C_j with $i < j$ are at the same depth in T_t, then they must be disjoint.

(ii) Prove that the tree T_t determines from which *locations* in R the algorithms picks random values for the variables in C_i, for all clauses C_i in T_t.

We now make two definitions:

- Let us say that a legally labeled tree T_t is *consistent* with R if all clauses in T_t are not satisfied by the respective values in R.
- Let us say that a rooted tree T is *legally labeled* if it is labeled by clauses of the k-SAT instance $\mathcal{I}$, every two adjacent nodes are labeled by overlapping clauses, and no two nodes in the same level are labeled by overlapping clauses.

(iii) Given R and a positive integer M, if the algorithm runs for mM steps, then there must be a legally labeled tree with at least M clauses that is consistent with R.

(iv) Prove that the number of legally labeled trees with M nodes is at most $m(\Delta e)^M$, where m denotes the number of clauses in the original k-SAT instance. (Hint: Show the upper bound $m\binom{\Delta M}{M-1}$.)

(v) Prove that the probability over R that there is a legally labeled tree of size $\geq M$ that is consistent with R is at most $m(\Delta 2^{-k}e)^M/(1-\Delta 2^{-k}e)$.

(vi) Combine the above three parts to conclude that when $\Delta < 2^k/e$, the algorithm finds a satisfying assignment in expected $O(m \log m)$ steps (the constant in the big-Oh can depend on k).

4. *Wozencraft and Justesen Codes* (25 points)

From the theory of finite fields, it is possible to define addition and multiplication operations over the space $\{0,1\}^n$, where addition is the bit-wise XOR and multiplication of $\underline{u}, \underline{v} \in \{0,1\}^n$ satisfies the following natural properties:

- $\underline{u} \cdot \underline{v} \in \{0,1\}^n$,
- $\underline{u} \cdot (\underline{v} + \underline{w}) = \underline{u} \cdot \underline{v} + \underline{u} \cdot \underline{w}$,
- $\underline{u} \cdot \underline{v} = \underline{0} \Leftrightarrow \underline{u} = \underline{0}$ or $\underline{v} = \underline{0}$,

for all $\underline{w} \in \{0,1\}^n$ and $\underline{0} := (0, \ldots, 0)$. The *Wozencraft ensemble*, $\mathcal{W} := \{\mathcal{C}_\alpha : \alpha \in \{0,1\}^n \setminus \underline{0}\}$, is an ensemble of codes of length $2n$, where each code $\mathcal{C}_\alpha$ in the ensemble is parametrized by a vector $\alpha \in \{0,1\}^n \setminus \underline{0}$ and is defined by the encoding function

$$\underline{x} \in \{0,1\}^n \mapsto (\underline{x}, \underline{\alpha} \cdot \underline{x}) \in \{0,1\}^{2n}.$$

Thus, the rate of each code in this ensemble is $1/2$. In this exercise, we show that most codes in this ensemble are capacity achieving (we will focus on the erasure channel).

(a) Show that every non-zero vector $v \in \{0,1\}^{2n}$ belongs to exactly one code in $\mathcal{W}$.

(b) Suppose $\mathcal{C}_\alpha$ has been used for communication over BEC_p, for some fixed erasure probability $p < 1/2$. Let $J \subseteq [2n]$ denote a particular erasure pattern induced by the channel (that is, the channel happens to erase exactly the positions picked by J). Show that the erased bits can be recovered (regardless of the sent message) if and only if $\mathcal{C}_\alpha$ contains no non-zero codewords entirely supported on J. You may consider a decoder that tries to find a unique codeword matching the unerased positions and fails if no unique solution is found (*Hint:* First, observe that $\mathcal{C}_\alpha$ is a linear code).

(c) Now, fix an erasure pattern J of size $n(1-\gamma)$, for any fixed constant $\gamma > 0$. Show that, for large enough n, the fraction of codes in $\mathcal{W}$ that are unable to correct the erasure pattern induced by J (for one or more possible messages) is at most 2^{-cn}, where $c > 0$ is a constant depending on γ.

(d) Conclude that the error probability of all but at most a $2^{-c'n}$ fraction of the codes in $\mathcal{W}$ over BEC_p ($p < 1/2$) is at most $2^{-c'n}$, where $c' > 0$ is a constant depending on p, and n is large enough. (*Hint:* Use the result in the previous part combined with Markov and Chernoff bounds.)

(e) Now we describe Justesen's code, which is a concatenated code with varying inner codes. Consider the concatenated coding scheme as described in the lecture, where the block size for the inner codes is set to be $b := n$ (that is, each inner code encodes b bits into $b' = 2n$ bits). Moreover, suppose the number of inner blocks is set to be $2^n - 1$; that is, the final concatenated code is of length $N := b'(2^n - 1) = 2n(2^n - 1)$ bits and rate $R = (1-\epsilon)/2$, where the rate of the outer code is $1 - \epsilon$ for some sufficiently small $\epsilon > 0$ depending on p (as described in the lecture). Instead of using brute force for the right choice of the inner blocks, Justesen's concatenation encodes each inner block with a distinct code $\mathcal{C}_\alpha$ from the Wozencraft's ensemble. Show that for large enough n, and appropriate choice of the parameter ϵ, Justesen's explicit construction achieves arbitrarily small error probability over BEC_p. It suffices to sketch how the analysis of this construction differs from the original Forney's construction that was presented in the class.

Side note: The fact that the inner block size is logarithmic in N in Justesen's construction makes it possible to use classical algebraic codes such as the Reed-Solomon code for the outer code.