CS 225 - Pseudorandomness Prof. Salil Vadhan

Problem Set 4

Harvard SEAS - Fall 2016 *Due: Fri. Oct. 28, 2016 (5pm sharp)*

Your problem set solutions must be typed (in e.g. LaTeX) and submitted electronically to `cs225-hw@seas.harvard.edu`. You are allowed 12 late days for the semester, of which at most 5 can be used on any individual problem set. (1 late day = 24 hours exactly). Please name your file `ps4-`lastname`.*`.

The problem sets may require a lot of thought, so be sure to start them early. You are encouraged to discuss the course material and the homework problems with each other in small groups (2-3 people). Identify your collaborators on your submission. Discussion of homework problems may include brainstorming and verbally walking through possible solutions, but should not include one person telling the others how to solve the problem. In addition, each person must write up their solutions independently, and these write-ups should not be checked against each other or passed around.

Strive for clarity and conciseness in your solutions, emphasizing the main ideas over low-level details. Do not despair if you cannot solve all the problems! Difficult problems are included to stimulate your thinking and for your enjoyment, not to overwork you. *ed problems are extra credit.

Problem 5.1 (Limits of List Decoding)

Show that if there exists a q-ary code $\mathcal{C} \subset \Sigma^{\hat{n}}$ of rate ρ that is (δ, L) list-decodable, then $\rho \leq 1 - H_q(\delta, \hat{n}) + (\log_q L)/\hat{n}$.

Problem 5.2 (Concatenated Codes)

For codes $\mathrm{Enc}_1 : \{1, \ldots, N\} \to \Sigma_1^{n_1}$ and $\mathrm{Enc}_2 : \Sigma_1 \to \Sigma_2^{n_2}$, their *concatenation* $\mathrm{Enc} : \{1, \ldots, N\} \to \Sigma_2^{n_1 n_2}$ is defined by

$$\mathrm{Enc}(m) = \mathrm{Enc}_2(\mathrm{Enc}_1(m)_1)\mathrm{Enc}_2(\mathrm{Enc}_1(m)_2) \cdots \mathrm{Enc}_2(\mathrm{Enc}_1(m)_{n_1}).$$

This is typically used as a tool for reducing alphabet size, e.g. with $\Sigma_2 = \{0, 1\}$.

1. Prove that if Enc_1 has minimum distance δ_1 and Enc_2 has minimum distance δ_2, then Enc has minimum distance at least $\delta_1\delta_2$.

2. Prove that if Enc_1 is $(1 - \varepsilon_1, \ell_1)$ list-decodable and Enc_2 is (δ_2, ℓ_2) list-decodable, then Enc is $((1 - \varepsilon_1\ell_2) \cdot \delta_2, \ell_1\ell_2)$ list-decodable.

3. By concatenating a Reed–Solomon code and a Hadamard code, show that for every $n \in \mathbb{N}$ and $\varepsilon > 0$, there is a (fully) explicit code $\mathrm{Enc} : \{0,1\}^n \to \{0,1\}^{\hat{n}}$ with blocklength $\hat{n} = O(n^2/\varepsilon^2)$ with minimum distance at least $1/2 - \varepsilon$. Furthermore, show that with blocklength $\hat{n} = \mathrm{poly}(n, 1/\varepsilon)$, we can obtain a code that is $(1/2 - \varepsilon, \mathrm{poly}(1/\varepsilon))$ list-decodable in *polynomial time.* (Hint: the inner code can be decoded by brute force.)

Problem 5.6 (Improved list decoding of Reed–Solomon Codes)

1. Show that there is a polynomial-time algorithm for list decoding the Reed-Solomon codes of degree d over $\mathbb{F}_q$ up to distance $1-\sqrt{2d/q}$, improving the $1-2\sqrt{d/q}$ bound from Theorem 5.19. (Hint: do not use fixed upper bounds on the individual degrees of the interpolating polynomial $Q(Y, Z)$, but rather allow as many monomials as possible.)

2. (*) Improve the list-decoding radius further to $1 - \sqrt{d/q}$ by using the following "method of multiplicities". First, require the interpolating polynomial $Q(Y, Z)$ to have a zero of multiplicity s at each point $(y, r(y))$ — that is, the polynomial $Q(Y + y, Z + r(y))$ should have no monomials of degree smaller than s. Second, use the fact that a univariate polynomial $R(Y)$ of degree t can have at most t roots, counting multiplicities.

Problem 5.7 (Twenty Questions)

In the game of 20 questions, an oracle has an arbitrary secret $s \in \{0,1\}^n$ and the aim is to determine the secret by asking the oracle as few yes/no questions about s as possible. It is easy to see that n questions are necessary and sufficient. Here we consider a variant where the oracle has two secrets $s_1, s_2 \in \{0,1\}^n$, and can adversarially decide to answer each question according to either s_1 or s_2. That is, for a question $f : \{0,1\}^n \to \{0,1\}$, the oracle may answer with either $f(s_1)$ or $f(s_2)$. Here it turns out to be impossible to pin down either of the secrets with certainty, no matter how many questions we ask, but we can hope to compute a small list L of secrets such that $|L \cap \{s_1, s_2\}| \neq 0$. (In fact, $|L|$ can be made as small as 2.) This variant of twenty questions apparently was motivated by problems in Internet traffic routing.

1. Let $\mathrm{Enc} : \{0,1\}^n \to \{0,1\}^{\hat{n}}$ be a code such that every two codewords in Enc *agree* in at least a $1/2 - \varepsilon$ fraction of positions and that Enc has a polynomial-time $(1/4 + \varepsilon, \ell)$ list-decoding algorithm. Show how to solve the above problem in polynomial time by asking the $\hat{n}$ questions $\{f_i\}$ defined by $f_i(x) = \mathrm{Enc}(x)_i$.

2. Taking Enc to be the code constructed in Problem 5.2.3, deduce that $\hat{n} = \mathrm{poly}(n)$ questions suffices.