Carnegie Mellon University Instructor: Venkatesan Guruswami
15-859Y: Coding Theory Term: Fall 2014

PROBLEM SET 3
Due by Monday, November 17

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*. You must clearly acknowledge your collaborator(s) in the write-up of your solutions.
  3. Of course, you are also more than welcome to also work alone.
- You should not search for solutions on the web. More generally, you should try and solve the problems without consulting any reference material other than the course notes and what we cover in class. Ask the instructor for hints or clarifications if this does not seem to work for you on some problems. However, note that you may use references to brush up on the underlying math skills needed to solve some of the problems, such as linear algebra, matrix theory, number theory, finite fields, etc.
- Please start work on the problem set early. The problem set has **seven** problems worth a total of 100 points.

(The notation $\mathbb{F}[X]_{\leq k}$ stands for polynomials over the field $\mathbb{F}$ of degree at most k. We denote $[n] = \{1, 2, \ldots, n\}$.)

1. (15 points) We saw in class that the $[n, k+1, n-k]_q$ Reed-Solomon code encoding polynomials in $\mathbb{F}_q[X]_{\leq k}$ by their evaluations at n distinct elements $a_1, a_2, \ldots, a_n \in \mathbb{F}_q$ can be "list recovered" in the following sense:

   > Let $\ell < \frac{n}{k}$.
   > Given sets $S_i \subset \mathbb{F}_q$, $|S_i| \leq \ell$, for each $i \in [n]$, there are at most $O(n^2)$ polynomials $f \in \mathbb{F}_q[X]_{\leq k}$ such that $f(a_i) \in S_i$ for every $i \in [n]$, and the list of such polynomials can be found in polynomial time.

   In this exercise, you will show that this result is tight, in the sense that when $\ell = \lceil \frac{n}{k} \rceil$, there are settings where there are super-polynomially many (i.e., $n^{\omega(1)}$) polynomials.

   Let r be a fixed prime power. Let $n = q = r^m$ and $k = \frac{r^m - 1}{r-1}$. Prove that there are at least r^{2^m} polynomials $f \in \mathbb{F}_q[X]_{\leq k}$ such that $f(a) \in \mathbb{F}_r$ for every $a \in \mathbb{F}_q$. Deduce that the Reed-Solomon list recovery algorithm cannot be improved to work for $\ell = \lceil \frac{n}{k} \rceil$ in general.

   Hint: For $x \in \mathbb{F}_{r^m}$, $x^{\frac{r^m-1}{r-1}}$ always belongs to the subfield $\mathbb{F}_r$ (why?). So the polynomials $f_\beta(X) := (X+\beta)^{\frac{r^m-1}{r-1}}$ for $\beta \in \mathbb{F}_{r^m}$ take values in $\mathbb{F}_r$ on evaluation points in $\mathbb{F}_{r^m}$. Find 2^m of these polynomials that are linearly independent over $\mathbb{F}_r$.

2. (12 points) In the game of 20 questions, an oracle has an arbitrary secret $s \in \{0,1\}^k$ 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 k questions are necessary and sufficient. Here we consider a variant where the oracle has two secrets $s_1, s_2 \in \{0,1\}^k$ and can adversarially decide to answer each question according to either s_1 or s_2. That is, for a question $f : \{0,1\}^k \to \{0,1\}$, 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 set S of secrets, of a fixed size independent of k, such that $S \cap \{s_1, s_2\} \neq \emptyset$. (In fact, $|S|$ can be made as small as 2.) This variant of twenty questions apparently arose from Internet routing algorithms used by Akamai.

   (i) Let C be a binary code of block length n and 2^k codewords such that (a) every two codewords of C *agree* in at least a fraction $(1/2 - \epsilon)$ of positions and (b) C can be efficiently list decoded from a fraction $(1/4+\epsilon)$ of errors with list size ℓ independent of k. Show how to solve the above problem in polynomial time by asking questions based on the code C.

   (ii) Briefly describe how to construct a code with the properties spelled out in (i) above, and deduce that $n \leq \mathrm{poly}(k)$ questions suffice for the above variant of 20 questions. (In fact, $n = O(k)$ questions suffice, and feel free to show this stronger bound.)

3. (12 points) In lecture we saw that an s-folded Reed-Solomon code where the field element $\gamma \in \mathbb{F}_q^*$ used for folding has order s admits a clean linear-algebraic list decoder for correcting a $\frac{s}{s+1}(1-R)$ fraction of errors, where R is the code rate. Unfortunately to bound the list size by a polynomial our arguments required γ to have large order. In this exercise, you will show that this was for an inherent reason.

   (i) Prove that if the order of γ is r, then there exist some choice of polynomials $A_0, A_1, \ldots, A_s \in \mathbb{F}_q[X]$, not all zero, such that there are $q^{k/r}$ polynomials $f \in \mathbb{F}_q[X]_{\leq k}$ satisfying the condition

$$A_0(X) + A_1(X)f(X) + A_2(X)f(\gamma X) + \cdots + A_s(X)f(\gamma^{s-1}X) = 0\,.$$

   (ii) Let C be a s-folded Reed-Solomon code of length $N = (q-1)/s$ and rate R that is based on folding with a $\gamma \in \mathbb{F}_q^*$ of order s. Prove that if C is efficiently (ρ, L)-list decodable, then there is in fact a Reed-Solomon code (no folding needed!) of length N and rate R that is also efficiently (ρ, L)-list decodable.
   (This means that if we get an improvement over the Johnson radius $1-\sqrt{R}$ for a folded RS code using a small order element for folding, then we will also get an improvement for Reed-Solomon codes themselves.)

4. (13 points) Let p be a prime and let $1 \leq k < p$. For prime fields $\mathbb{F}_p$ and $m \mid (p-1)$, we can also define m-folded Reed-Solomon codes based on *additive* folding, namely the map $\mathbb{F}_p[X]_{\leq k} \to (\mathbb{F}_p^m)^{(p-1)/m}$ defined by

$$f(X) \mapsto \left(\begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ f(m-1) \end{bmatrix}, \begin{bmatrix} f(m) \\ f(m+1) \\ \vdots \\ f(2m-1) \end{bmatrix}, \ldots, \begin{bmatrix} f(p-1-m) \\ f(\gamma^{n-m+1}) \\ \vdots \\ f(p-2) \end{bmatrix} \right)\,. \tag{1}$$

   Prove that if $A_0, A_1, \ldots, A_s \in \mathbb{F}_p[X]$ are not all zero, then there are at most p^{s-1} polynomials $f \in \mathbb{F}_p[X]_{\leq k}$ that obey the condition

$$A_0(X) + A_1(X)f(X) + A_2(X)f(X+1) + \cdots + A_s(X)f(X+s-1) = 0\,.$$

(As with multiplicative folding based on a primitive element γ, the above implies that the folded RS code defined in (1) can be list decoded from an error fraction $\frac{s}{s+1}\left(1-\frac{mR}{m-s+1}\right)$, for any s, $1 \le s \le m$.)

5. (15 points) For constant k, we will see in lecture a construction of a k-query locally decodable code encoding n message bits into codewords of length $2^{O(n^{1/(k-1)})}$ over an alphabet of size $O(1)$. (Here we allow the big-Oh to suppress constant factors depending on k, as we are thinking of k as a constant in this exercise.)

   In this exercise, you are asked to give a k-query locally decodable code for n-bit messages with a better encoding length of $2^{O(n^{1/(2k-1)})}$ but over a much bigger alphabet, also of size $2^{O(n^{1/(2k-1)})}$. (This means each message bit can recovered by reading k symbols from the encoding, each consisting of $O(n^{1/(2k-1)})$ bits.)

   Hint: One approach is to encode the message via a low-degree polynomial as in class, except now use a degree $(2k-1)$ polynomial in $\approx n^{1/(2k-1)}$ variables, and include the evaluations of the polynomial as well as its first order partial derivatives in the encoding.

6. ($6+5+3+2+2+2=20$ points) We have mentioned objects called algebraic-geometric codes, that generalize Reed-Solomon codes and have some amazing properties, a few times in the course. The objective of this exercise is to construct one such AG code, and establish its rate vs distance trade-off.

   Let p be a prime and $q = p^2$. Consider the equation

   $$Y^p + Y = X^{p+1} \tag{2}$$

   over $\mathbb{F}_q$.

   (a) Prove that there are exactly p^3 solutions in $\mathbb{F}_q \times \mathbb{F}_q$ to (2). That is, if $S \subseteq \mathbb{F}_q^2$ is defined as

   $$S = \{(\alpha, \beta) \in \mathbb{F}_q^2 \mid \beta^p + \beta = \alpha^{p+1}\} \tag{3}$$

   then $|S| = p^3$.

   (b) Prove that the polynomial $f(X,Y) = Y^p + Y - X^{p+1}$ is irreducible over $\mathbb{F}_q$.
   (**Suggestion**: One approach is to use the Eisenstein criterion (feel free to look this up), considering $f(X,Y)$ as a polynomial in X with coefficients from $\mathbb{F}_q[Y]$.)

   (c) Let $n = p^3$. Consider the evaluation map $\mathrm{ev} : \mathbb{F}_q[X,Y] \to \mathbb{F}_q^n$ defined by

   $$\mathrm{ev}(f) = (f(\alpha,\beta) : (\alpha,\beta) \in S)$$

   where S is the set defined in (3).
   Argue that if $f \neq 0$ and is not divisible by $Y^p + Y - X^{p+1}$, then $\mathrm{ev}(f)$ has Hamming weight at least $n - \deg(f)(p+1)$, where $\deg(f)$ denotes the *total* degree of f.
   (**Hint**: You are allowed to use ***Bézout's theorem***, which states that if $f, g \in \mathbb{F}_q[X,Y]$ are nonzero polynomials *with no common factors*, then they have at most $\deg(f)\deg(g)$ common zeroes.)

   (d) For an integer parameter $\ell \ge 1$, consider the set $\mathcal{F}_\ell$ of bivariate polynomials

   $$\mathcal{F}_\ell = \{f \in \mathbb{F}_q[X,Y] \mid \deg(f) \le \ell, \deg_X(f) \le p\}$$

   where $\deg_X(f)$ denotes the degree of f in X.
   Argue that $\mathcal{F}_\ell$ is an $\mathbb{F}_q$-linear space of dimension $(\ell+1)(p+1) - \frac{p(p+1)}{2}$.

(e) Consider the code $C \subseteq \mathbb{F}_q^n$ for $n = p^3$ defined by

$$C = \{\mathrm{ev}(f) \mid f \in \mathcal{F}_\ell\} .$$

Prove that C is a linear code with minimum distance at least $n - \ell(p+1)$.

(f) Deduce a construction of an $[n,k]_q$ code with distance $d \geq n - k + 1 - p(p-1)/2$.
(**Remark:** Reed-Solomon codes have $d = n - k + 1$, whereas these codes are off by $p(p-1)/2$ from the Singleton bound. However they are much longer than RS codes, with a block length of $n = q^{3/2}$, and the deficiency from the Singleton bound is only $o(n)$.)

7. (13 points) Recent applications in distributed storage of massive amounts of data have motivated the study of codes where every codeword symbol can be recovered from few other codeword symbols, so as to enable recovery from the failure of any single node in a distributed system.

More formally, we are interested in codes that produce an n-symbol codeword from k information symbols and, for any symbol of the codeword, there exist at most t other symbols such that the value of the symbol can be recovered from them. Here we think of $t \ll n$.

(i) Prove that the rate of such a code is at most $\frac{t}{t+1}$.

(ii) Prove that the minimum distance of such a code is at most $n - k - \lceil \frac{k}{t} \rceil + 2$. Which classical coding bound does this generalize?

Suggestion: One approach is to use the fact that if a code $C \subset \Sigma^n$ has distance d, then $n - d$ equals the largest size of a subset $T \subseteq \{1, 2, \ldots, n\}$ such that $|C_T| < |C|$, where $C_T \subset \Sigma^T$ is the code C projected onto coordinates in T.