CS 151 Complexity Theory **Spring 2017**

Problem Set 4

Out: April 26 *Due:* **May 3**

Reminder: you are encouraged to work in groups of two or three; however you must turn in your own write-up and note with whom you worked. You may consult the course notes and the text (Papadimitriou). The full honor code guidelines can be found in the course syllabus.

Please attempt all problems. **To facilitate grading, please turn in each problem on a separate sheet of paper and put your name on each sheet. Do not staple the separate sheets.**

1. Define $\widetilde{\mathbf{ZPP}}$ to be the class of all languages decided by a probabilistic Turing Machine running in *expected* polynomial time. That is, for every language $L \in \widetilde{\mathbf{ZPP}}$ there is a probabilistic Turing Machine M (with two read-only tapes — the first tape containing the input, and the second tape containing a random bit in every tape square) with the following behavior: on input $x \in L$, M always accepts, on input $x \notin L$, M always rejects, and for every input x,

   $$\mathrm{E}[\#\text{ steps before } M \text{ halts}] = |x|^{O(1)}.$$

   Show that $\widetilde{\mathbf{ZPP}} = \mathbf{ZPP}$.

2. List-decoding of the binary Hadamard code. Throughout this problem $\mathbb{F}_2$ is the field with 2 elements (addition and multiplication are performed modulo 2). Given a k-bit message m, the associated Hadamard codeword $C(m)$ is described by first producing a linear multivariate polynomial $p_m(x_0, x_1, \ldots, x_{k-1}) = \sum_{i=0}^{k-1} m_i x_i$, and then evaluating that polynomial at all vectors in the space $\mathbb{F}_2^k$: $C(m) = (p_m(w))_{w \in \mathbb{F}_2^k}$. Thus the codeword has $n = 2^k$ bits, and the w-th bit is the inner product mod 2 of the k-bit vectors m and w. The bits of a codeword $C = C(m)$ are naturally indexed by $\mathbb{F}_2^k$; we write C_w (with $w \in \mathbb{F}_2^k$) to mean the w-th coordinate, which is just $p_m(w)$.

   Since the distance of the Hadamard code is $(1/2)n$ (by Schwartz-Zippel), unique decoding is only possible from up to $(1/4)n$ errors. In this problem you will show that efficient *list-decoding* is possible from a received word R that has suffered up to $(1/2 - \epsilon)n$ errors.

   You may need to use Chebyshev's Inequality: for a random variable X and any $k > 0$,

   $$\Pr[|X - \mathrm{E}[X]| \geq k] \leq \frac{\mathrm{Var}[X]}{k^2}.$$

   (a) Consider the following probabilistic procedure. Pick ℓ vectors $v_1, v_2, \ldots, v_\ell \in \mathbb{F}_2^k$ independently and uniformly at random. For a subset $S \subseteq \{1, 2, 3, \ldots, \ell\}$ define $u_S = \sum_{i \in S} v_i$. Show that for every non-empty set S and every $\alpha \in \mathbb{F}_2^k$, we have $\Pr[u_S = \alpha] = 2^{-k}$. Then show that for every pair of non-empty subsets S and T with $S \neq T$, and every $\alpha, \beta \in \mathbb{F}_2^k$, we have:

   $$\Pr[u_S = \alpha \wedge u_T = \beta] = \Pr[u_S = \alpha] \Pr[u_T = \beta].$$

In other words, the set of vectors u_S are pairwise independent random variables uniformly distributed on $\mathbb{F}_2^k$.

(b) Suppose that the received word R agrees with $C = C(m)$ in at least a $1/2 + \epsilon$ fraction of its n bits. Verify for yourself that $C_w + C_{w+e_i} = m_i$ (here e_i is the i-th elementary vector in $\mathbb{F}_2^k$ – the vector with 1 in the i-th position and zeros elsewhere). Of course, in our decoding algorithm we do not have access to C to find C_w and C_{w+e_i}. So, we will replace C_w with a "guess" (for now imagine it is always correct), and C_{w+e_i} with R_{w+e_i} (which may or may not be correct). Show that for all i:

$$\Pr\left[|\{S \neq \emptyset : C_{u_S} + R_{u_S+e_i} = m_i\}| \leq \frac{2^\ell - 1}{2}\right] \leq \frac{1}{4\epsilon^2(2^\ell - 1)}$$

Hint: define indicator random variables for the event $R_{u_S+e_i} = C_{u_S+e_i}$. Use the fact that for pairwise independent random variables, the variance of the sum is the sum of the variances.

(c) Describe a probabilistic procedure A that has the following behavior:

- it has random access to a word R that agrees with $C = C(m)$ in at least a $1/2 + \epsilon$ fraction of its n bits, and
- it runs in time $\text{poly}(k, \epsilon^{-1})$, and
- with probability at least $3/4$ it outputs a list of $L = O(k\epsilon^{-2})$ "candidate messages" $m_1, m_2, \ldots, m_L$ that includes the original message m.

Hint: argue that it takes surprisingly few bits to describe $(C_{u_S})_{S \neq \emptyset}$.

(d) Prove the Goldreich-Levin Theorem: for every function family $f_n : \{0,1\}^n \to \{0,1\}^n$,

$$GL(x, y) = \sum_{i=1}^{n} x_i y_i$$

is a *hard bit* for the function family $f'_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{2n}$ defined by $f'(x, y) = (f(x), y)$. That is, if there is a circuit family $\{C_n\}$ of size $s(n) \geq n$ that achieves:

$$\Pr_{x,y}[C_n(x, y) = GL(f'_n(x, y))] \geq 1/2 + \epsilon(n) \tag{4.1}$$

then there is a circuit family $\{C'_n\}$ of size $s'(n)$ that achieves:

$$\Pr_{x,y}[C'_n(x, y) = f'_n(x, y)] \geq \epsilon'(n),$$

with $s'(n) = (s(n)/\epsilon(n))^{O(1)}$ and $\epsilon'(n) = (\epsilon(n)/n)^{O(1)}$. Hint: It may be useful to observe that by an averaging argument, (**??**) implies that for at least an $\epsilon(n)/2$ fraction of the x's:

$$\Pr_{y}[C_n(x, y) = GL(f'_n(x, y))] \geq 1/2 + \epsilon(n)/2.$$

3. List-decoding of Reed-Solomon codes. Throughout this problem $\mathbb{F}_q$ is the field with q elements. Given a k-bit message m, the associated Reed-Solomon codeword $C(m)$ is described by first producing a degree $k-1$ polynomial $p_m(x) = \sum_{i=0}^{k-1} m_i x^i$, and then evaluating that polynomial at all of the elements in the field $\mathbb{F}_q$: $C(m) = (p_m(w))_{w \in \mathbb{F}_q}$. You should think of q as being polynomial in k.

   Now, we are given a received word R that has suffered e errors. We know that if $e > q/2$ unique decoding is impossible, since the distance of this code is *a priori* at most q. In this problem you will show that efficient *list-decoding* is possible when e is as large as $q - \sqrt{2kq}$. If we choose, say, $q = k^2$, then this implies that we can recover from up to a $1 - o(1)$ fraction of errors!

   You may need the following fact about polynomials: $(x - \alpha)$ divides a polynomial $p(x)$ iff α is a root of p.

   (a) We view the received word R as a function $R : \mathbb{F}_q \to \mathbb{F}_q$. Show that one can efficiently find a polynomial $Q(x, y) \not\equiv 0$ with x-degree at most $\sqrt{q}$ and y-degree at most $\sqrt{q}$ for which $Q(w, R(w)) = 0$ for all $w \in \mathbb{F}_q$. Hint: elementary linear algebra is sufficient here.

   (b) Let $p : \mathbb{F}_q \to \mathbb{F}_q$ be a polynomial of degree at most $k-1$ for which $|\{w \in F_q : p(w) \neq R(w)\}| \leq q - t$. That is, p is a RS codeword that agrees with the received word R in at least t locations. Show that if $t > k\sqrt{q}$ then $(y - p(x))$ divides $Q(x, y)$. (Hint: view Q as a *univariate* polynomial in y with coefficients in $\mathbb{F}_q[x]$.) Conclude that using an efficient algorithm for factoring multivariate polynomials (which is known), we can find all codewords which have agreement $t > k\sqrt{q}$ with R.

   (c) The $(1, k-1)$-weighted degree of a polynomial $Q(x, y) = \sum_{i,j} q_{i,j} x^i y^j$ is defined to be $\max\{i + (k-1)j : q_{i,j} \neq 0\}$. Refine your analysis in parts (a) and (b) to show that one can find $Q(x, y)$ with the required properties *and with* $(1, k-1)$*-weighted degree at most* $\sqrt{2kq}$, and therefore can tolerate agreement only $t > \sqrt{2kq}$.

4. List decoding of Parvaresh-Vardy codes. The setting is similar to the one in the previous problem. Throughout this problem $\mathbb{F}_q$ is the field with q elements. We have an integer parameter $c \geq 1$, and we fix a polynomial $E(x)$ of degree k, that is irreducible over $\mathbb{F}_q$. Set $h = \lceil (q+1)^{1/(c+1)} \rceil$.

   Given a k-bit message m, the associated Parvaresh-Vardy codeword $C(m)$ is described as follows:

   - produce the degree $k-1$ polynomial $p_m(x) = \sum_{i=0}^{k-1} m_i x^i$
   - define $p_m^{(i)}(x)$ to be the polynomial

     $$(p_m(x))^{h^i} \bmod E(x),$$

     for $i = 0, 1, 2, \ldots, c-1$. Note that $p_m^{(0)}(x)$ is just $p_m(x)$.
   - the codeword $C(m)$ is the following sequence of symbols in $\mathbb{F}_q^c$:

     $$\left([p_m^{(0)}(w), p_m^{(1)}(w), \ldots, p_m^{(c-1)}(w)] \right)_{w \in F_q}.$$

Now, we are given a received word R that has suffered e errors. You will show that *list-decoding* is possible for Parvaresh-Vardy codes when e is as large as $q - (c+1)kh$. Note, for example, that when taking $c = \log q$, we require agreement only $\approx k \log q$, in contrast to the $\approx \sqrt{kq}$ achievable with Reed-Solomon codes (typically, q is a large polynomial in k).

(a) We view the received word R as a function $R : \mathbb{F}_q \to \mathbb{F}_q^c$. Show that one can efficiently find a polynomial $Q(x, y_0, y_1, \ldots, y_{c-1}) \not\equiv 0$ with individual variable degrees at most $h-1$ for which $Q(w, R(w)_0, \ldots R(w)_{c-1}) = 0$ for all $w \in \mathbb{F}_q$. Hint: elementary linear algebra is sufficient here.

(b) Let $p : \mathbb{F}_q \to \mathbb{F}_q$ be a polynomial of degree at most $k-1$ for which

$$|\{w \in F_q : [p^{(0)}(w), p^{(1)}(w), \ldots, p^{(c-1)}(w)] \neq R(w)\}| \leq q - t.$$

That is, p is a Parvaresh-Vardy codeword that agrees with the received word R in at least t locations. Show that if $t > (c+1)kh$ then the polynomial

$$Q(x, p^{(0)}(x), p^{(1)}(x), \ldots, p^{(c-1)}(x))$$

is the zero polynomial.

(c) Recall that $\mathbb{F}_q[x]/E(x)$ – the set of univariate polynomials with coefficients in $\mathbb{F}_q$ and with multiplication and addition performed modulo $E(x)$ – is a field. Define the following polynomial with coefficients in this field:

$$Q^*(z) = Q(x, z, z^h, z^{h^2}, \ldots, z^{h^{c-1}}) \bmod E(x).$$

Assuming the existence of an efficient algorithm that outputs all of the roots of a given univariate polynomial with coefficients in $\mathbb{F}_q[x]/E(x)$, and making reference to Q^*, argue that one can efficiently find all Parvaresh-Vardy codewords which have agreement $t > (c+1)kh$ with R.