6.897 Algorithmic Introduction to Coding Theory	November 5, 2001
Lecture 14	
Lecturer: Madhu Sudan	*Scribe: Daniel Preda*

Topics for today:

- Ideal codes
- Abstract list-decoding
- Decoding of CRT codes

1 Ideal Error-Correcting Codes

Let us recall the following definition of an ideal error-correcting code from the last lecture.

Definition 1 *An* ideal error-correcting code C *is given by an integral domain* R*, a sequence of* n *ideals* $I_1, \ldots, I_n \subseteq R$ *(and associated sets of representatives* $\Sigma_1, \ldots, \Sigma_n$*), and a* message space $M \subseteq R$*. The canonical encoding function for* C *maps the message* $m \in M$ *to the sequence* $\langle m(I_1), \ldots, m(I_n)\rangle$*.*

We will show to abstract the list-decoding algorithm for Reed-Solomon codes in this framework, and then apply it to the CRT codes. First lets recall how Reed-Solomon and CRT codes fit in this framework.

Reed-Solomon codes

We saw how to express Reed-Solomon codes in this framework:

- The ring $R = \mathbb{F}_q[x]$.
- The ideals $I_1, \ldots, I_n$ being given by $I_i = (x - \alpha_i)$, where $(x - \alpha_i)$ represents the ideal of polynomials that are multiples of $x - \alpha_i$, i.e., $(x - \alpha_i) = \{p(x) \cdot (x - \alpha_i) \mid p(x) \in \mathbb{F}_q[x]\}$. The quotient ring $R/I_i \cong \mathbb{F}_q$ where the representatives are the set of polynomials of degree zero.
- The message space $M \subseteq R$ is the set of polynomials of degree less than k.

The ith coordinate of the encoding of a message polynomial m is the unique polynomial $m_i(x)$ of degree zero that is contained in the coset $m(x) + \{p(x) \cdot (x - \alpha_i) \mid \alpha_i \in \mathbb{F}_q\}$, which turns out to be the constant $m(\alpha_i)$. Thus this definition is consistent with our usual definition of Reed-Solomon codes.

CRT Codes

In the case of the CRT codes we have:

- R is *the* integral domain, i.e., the integers.
- $I_1, \ldots, I_n$ are the ideals $p_1\mathbb{Z}, \ldots, p_n\mathbb{Z}$ where $p_1, \ldots, p_n$ are n relatively prime integers, with Σ_i being $\mathbb{Z}_{p_i}$ the set of non-negative integers of value less than p_i.
- The message space $M \subseteq \mathbb{Z}$, is the set $M = \{0, 1, ..., K-1\}$ (where for convenience we pick $K = \prod_{i=1}^{k} p_i$).

Once again a message $m \in \mathbb{Z}_K$ is mapped to the vector $\langle m \pmod{p_1}, \ldots, m \pmod{p_n}\rangle$.

Size and distance of ideal error-correcting codes

While the abstraction so far describes how to construct the ideal error-correcting codes, it doesn't shed light on the minimum distance of the codes constructed this way. Here we give an informal discussion of a notion that is used to prove distance properties (as also specify decoding algorithms). To argue about distance properties, we associate with elements of the ring, a "size". E.g. the size of an integer will simply be its absolute value; the size of a polynomial of degree k over $\mathbb{F}_q$ will be q^{k+1}, etc. In general the size function must satisfy some axioms such as $\text{size}(a+b) \leq \text{size}(a) + \text{size}(b)$; and $\text{size}(ab) = \text{size}(a)\text{size}(b)$ etc. Here we won't go over this concept rigorously — though it can be done [5]. Instead we will keep this property in mind loosely and become precise in the case of Reed-Solomon codes and CRT codes. In the rigorous case, one also associates a size with ideals in R — the size of I being the size of the smallest non-zero element in I. If the size of the ideal also satisfies some size axioms, then minimum distance of the code can be established formally.

2 Abstracting the Reed-Solomon decoding algorithm

We start by recalling the (first) list-decoding algorithm for decoding Reed-Solomon codes and slowly translating it so it sounds like a generic decoding algorithm for ideal error correcting codes. The table below is best read row-by-row.

	RS decoder	Abstraction
Given:	$\alpha_1, \ldots, \alpha_n$ $y_1, \ldots, y_n i \in \mathbb{F}_q$	$I_1, \ldots, I_n$ $y_1, \ldots, y_n$, with $y_i \in \Sigma_i$
Find:	All polynomials p of degree less than k such that $p(x_i)$ for many values of $i \in [n]$	All messages $m \in M$ such that $m \pmod{I_i} = y_i$ for many values of i.
Step 1:	Find $Q(x, y) \in \mathbb{F}_q[x, y]$ such that	Since $R = \mathbb{F}_q[x]$, it makes sense to interpret $F_q[x, y]$ as $R[y]$ and thus the sentence on the left translates to "Find $Q(y) \in R[y]$ such that"
	Q non-zero	"Q non-zero"
	$\deg_y(Q)$ small	"$\deg_y(Q)$ small"
	$\deg_x(Q)$ small	Notice that R is not necessarily a ring over some variable; so we can't use $\deg_x$; instead we should ask that the coefficients of Q be small. Thus we get "Coefficients of Q have small size".
	$Q(\alpha_i, y_i) = 0$ for all i	This is a little trickier — but guessing that α_i is somehow related to I_i and y_i remains as it is, we come up with "$Q(y_i) \in I_i$ for all i" — this guess can be verified to be right.
Step 2:	Factor Q and report all polynomials $p(x) \in M$ such that $y - p(x)$ divides Q	The translation seems obvious: "Factor Q and report all messages $m \in M$ such $y - m$ divides Q." The important thing to note is that it is a fortunate coincidence that this is actually well-defined. In particular $R[y]$ is a unique factorization domain (upto multiplications by elements of R). (This need not have been the case if R were not an integral domain, or so we believe.)

We summarize the resulting "high-level algorithm" for decoding ideal codes below. In the explanation, we make a slight syntactic change: Instead of saying $Q(y_i) \in I_i$, we define a new ideal $J_i \subseteq R[y]$ consisting of polynomials Q such that $Q(y_i) \in I_i$. It can be verified that J_i is just the sum of the ideals I_i (viewed as an ideal of $R[y]$) and $(y - y_i)$. While the change is syntactic, it allows us to take a few extra steps in these notes (which we didn't manage in the lecture).

Ideal List-Decoder:

Given: n relatively prime[1], ideals $I_1, \ldots .I_n$ in a ring R, a message space $M \subseteq R$ and n elements $y_1, \ldots, y_n \in R$.

[1]Ideals I and J are relatively prime if $I \cap J = IJ$.

Unspecified parameters: ℓ and B, whose choices depend on the ring R, its size structure, the message space M, etc.

Step 0: Define ideals $J_1, \ldots, J_n \subseteq R[y]$ to be $J_i = (I_i) + (y - y_i)$.

Step 1: Find non-zero $Q \in R[y]$ such that $\deg(Q) \leq \ell$ and every coefficient of Q is has size at most B such that $Q \in \cap_{i=1}^{n} J_i$.

Step 2: Factor Q and output all $m \in M$ such that $y - m$ divides Q.

To specify the parameters, and to show how to achieve the steps above effectively, we need to know more about the ring R. Below we give an example of how we can apply the above structure to get a CRT list-decoder. We will still end up doing a lot of work to prove correctness and analyze bounds. But before going on to the case of the CRT codes, lets describe a "minor" twist to the above algorithm. Suppose the ideals $I_1, \ldots, I_n$ had not been relatively prime? What would the right definition have been? The one place where we use the relative primality (of J's but they inherit it from the relative primality of the I's) is in saying the Q should lie in the intersection of the ideals $J_1, \ldots, J_n$. If relative primality doesn't hold, we should probably revert to what seems to be the more correct statement to insist on — i.e., Q should lie in the *product* of $J_1, \ldots, J_n$. Thus we get the following modified algorithm for decoding ideal codes.

Ideal List-Decoder-2:

Given: n ideals $I_1, \ldots .I_n$ in a ring R, a message space $M \subseteq R$ and n elements $y_1, \ldots, y_n \in R$.

Unspecified parameters: ℓ and B, whose choices depend on the ring R, its size structure, the message space M, etc.

Step 0: Define ideals $J_1, \ldots, J_n \subseteq R[y]$ to be $J_i = (I_i) + (y - y_i)$.

Step 1: Find non-zero $Q \in R[y]$ such that $\deg(Q) \leq \ell$ and every coefficient of Q is has size at most B such that $Q \in \prod_{i=1}^{n} J_i$.

Step 2: Factor Q and output all $m \in M$ such that $y - m$ divides Q.

So what does this "minor change" buy us? Well, it turns out we get the weighted list-decoding algorithm immediately. Since we don't make any restrictions on the I_i's, we can repeat each one w_i times if needed. The change will be in the analysis, in which we need to argue that low-degree polynomials of small degree still exist in the ideal $\prod_{i=1}^{n} J_i$. We didn't argue this generically even in the simpler case; We certainly won't do it now! But in specific examples it can be done!

We now specialize the simpler algorithm above for the case of CRT codes and analyze it.

3 Decoding CRT codes

The algorithm above, applied to the case of CRT codes, becomes the following:

CRT-decoding-1:

Given: n relatively prime integers $\langle p_1 < \cdots < p_n \rangle$, residues $\langle r_1, \ldots, r_n \rangle$ with $r_i \in \mathbb{Z}_{p_i}$, and parameters k and t.

Goal Output all $m \in \mathbb{Z}_K$, where $K = \prod_{i=1}^{k} p_i$, such that $m = r_i \pmod{p_i}$.

Parameters: B, ℓ.

Step 1: Find $Q \in \mathbb{Z}[y]$ such that:

- $Q \not\equiv 0$.
- $\deg(Q) \leq \ell$.
- Coefficients of Q are at most B in absolute value.
- $Q(r_i) = 0 \pmod{p_i}$ for every $i \in [n]$.

Step 2: Factor Q and report all integers m such that $y - m$ divides Q and $m = r_i \pmod{p_i}$ for at least t values of $i \in [n]$.

The conditions $Q(r_i) = 0 \pmod{p_i}$ are not so nice to deal with. So, we modify them to an equivalent form which is easier to deal with: Let $N = \prod_{i=1}^{n} p_i$ and let R be the unique element of $\mathbb{Z}_N$ satisfying $R = r_i \pmod{p_i}$ for every $i \in [n]$ (as guaranteed to exist by the CRT theorem). Note that

$$Q(r_i) = 0 \pmod{p_i}, \quad \forall i \in [n], \quad \Leftrightarrow \quad Q(R) = 0 \pmod{N}.$$

Thus our goal in Step 1 is to find a polynomial with small coefficients that is zero modulo N at just one point. We will show below that this problem can be reduced to a "shortest vector problem (SVP)" in integer lattices. We define SVP below.

> **Shortest Vector Problem (SVP):** Approximation version with parameter $\alpha : \mathbb{Z}^+ \to \mathbb{Z}+$
> GIVEN: Integer $d \times n$ matrix $\mathbf{B}$ over the integers, integer bound L, with the promise that there exists a vector $\mathbf{x}' \in \mathbb{Z}^d$ such that $\|\mathbf{x}'\mathbf{B}\| \leq L/\alpha(d)$.
> FIND: A $\mathbf{x} \in \mathbb{Z}^d$ such that $\|\mathbf{x}\mathbf{B}\| \leq L$.

The set of vectors $\mathbf{xB}$ form an integer "lattice" in d-dimensional space. The task of finding short vectors in such lattices is a classical and well-studied problem. It was long conjectured to be NP-hard to solve exactly, i.e., when $\alpha(\cdot) = 1$. Finally, in a breakthrough result, shown to be true under randomized reductions by Ajtai [1]. Subsequently Micciancio [8] showed that the approximation version with $\alpha(\cdot) = \alpha < \sqrt{2}$ is also NP-hard under randomized reductions. Note that the problem becomes potentially easier as α becomes larger. We are interested in positive results. A seminal result by Lenstra, Lenstra, and Lovasz [6] showed that this problem is tractable for some finite function $\alpha(\cdot)$ (i.e., $\alpha(d) < \infty$ for every d). Their theorem is given below.

Theorem 2 ([6]) *The SVP problem with approximation parameter $\alpha(d) = 2^d$ is solvable in polynomial time in the length of the input.*

We now show how to find a polynomial as required in Step 1, using Theorem 2.

Lemma 3 *Given t-bit integers R and N and a degree bound ℓ and coefficient bound B, a non-zero polynomial Q of degree at most ℓ and coefficients at most B in absolute value satisfying $Q(R) = 0$ (mod N) can be found in polynomial time (in t and ℓ) provided such a polynomial with coefficients at most $B/(\ell 2^\ell)$ exists.*

Proof We will set up a lattice generated by the matrix $\mathbf{B}$ such that short vectors in this lattice correspond to polynomials Q such that $Q(R) = 0 \pmod{N}$. The intuition behind why this should be feasible is the following alternate characterization of a lattice: A lattice in $\mathbb{Z}^d$ is a subset $L \subseteq \mathbb{Z}^d$ that is closed under scalar multiplication and addition, i.e., if $\mathbf{x}, \mathbf{y} \in L$ and $\lambda \in \mathbb{Z}$ then $\lambda\mathbf{x}, \mathbf{x} + \mathbf{y} \in L$. If we consider the set of coefficient vectors $\langle q_0, \ldots, q_\ell$ of polynomials Q such that $Q(R) = 0 \pmod{N}$, then indeed such a set is closed under addition and scalar multiplication. Thus we see that the set of such vectors forms a lattice in $\mathbb{Z}^{\ell+1}$ and it is only a matter of perseverance to find an explicit basis. Below we give an explicit basis, using the fact that any such polynomial can be expressed as $p(x)(x-R)+c\cdot N$, where $p(x)$ has degree at most $\ell-1$. Consider the $(\ell+1)\times(\ell+1)$ matrix

$$\mathbf{B} = \begin{bmatrix} N & 0 & 0 & 0 & \ldots & 0 \\ -R & 1 & 0 & 0 & \ldots & 0 \\ 0 & -R & 1 & 0 & \ldots & 0 \\ 0 & 0 & -R & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & -R & 1 \end{bmatrix}.$$

We leave it to the reader to verify that the set of vectors $\langle q_0, \ldots, q_\ell \rangle$ that can be expressed as $\mathbf{xB}$ for $\mathbf{x} \in \mathbb{Z}^{\ell+1}$ is the set of coefficients of all polynomials $Q(y) = \sum_{i=0}^{\ell} q_i y^i$ such that $Q(R) = 0 \pmod N$. By Theorem 2, a vector of ℓ_2-norm, and thus all coefficients, at most B can be found in polynomial time in this lattice, provided a vector of ℓ_2 norm at most $B/2^{\ell+1}$ exists in this lattice. In turn such a vector does exist provided a vector all of whose coordinates are at most $B/((\ell+1)2^{\ell+1})$ exists. in this lattice. The lemma is thus proven. ■

We now move towards the efficiency of the second step. Once again this follows from the same seminal work of Lenstra, Lenstra, and Lovasz [6] mentioned above, but actually is a completely different result within this work (which uses the algorithm from Theorem 2 as a subroutine).

Theorem 4 *Given a polynomial $Q \in \mathbb{Z}[y]$ of degree ℓ with t-bit coefficients, Q can be factored over integers (modulo the factorization of the leading term) in time polynomial in ℓ and t.*

As a consequence we have that Step 2 of the algorithm above can also be executed in polynomial time. We only need to show that a good solution exists to Step 1 and then reason about the parameters for which the decoding algorithm works.

In the next lemma we prove that a very short solution to Step 1 exists, and thus Theorem 2 can find a moderately short solution.

Lemma 5 *A polynomial Q satisfying the conditions of Step 1 with coefficients of absolute value at most $B' \leq N^{1/\ell}$ exists.*

Proof Consider the function $f : \mathbb{Z}_{B'}^{\ell+1} \to \mathbb{Z}_N$ given by $f(q_0, \ldots, q_\ell) = \sum_{i=0}^{\ell} q_i R^i \pmod N$. The range of f is finite and has smaller cardinality than the domain (i.e., $N < (B')^{\ell+1}$). Thus there must exist distinct vectors $\mathbf{a}, \mathbf{b} \in \mathbb{Z}_{B'}^{\ell+1}$ such that $f(\mathbf{a}) = f(\mathbf{b})$. We conclude that $f(\mathbf{a}-\mathbf{b}) = 0 \pmod N$. Thus we have a non-zero polynomial $Q(y) = \sum_{i=0}^{\ell}(a_i - b_i)y^i$ such that $Q(R) = 0 \pmod N$. Since $a_i, b_i \in \mathbb{Z}_{B'}$, we have $|a_i - b_i| \leq B'$. ■

Next, we prove the usual next lemma — that any solution to Step 1 suffices.

Lemma 6 *For relatively prime integers $p_1 < \cdots < p_n$, let $N = \prod_{i=1}^{n} p_i$, and $T = \prod_{i=1}^{t} p_i$. If $Q(y)$ is any polynomial of degree ℓ with coefficients at most B such that $Q(R) = 0 \pmod N$, and $m \leq K$ is any integer such that $m = R \pmod{p_i}$ for t values of i and $(\ell+1)BK^\ell < T$, then $y - m$ divides $Q(y)$.*

Proof The lemma turns out to be harder to state than to prove! Let $S = \{i | m = R \pmod{p_i}\}$. Let $T' = \prod_{i \in S} p_i$. Note that $T' \geq T$. By the conditions on Q, we have for every $i \in S$,

$$Q(m)(\bmod p_i) = Q(R)(\bmod p_i) = 0(\bmod p_i).$$

We conclude that $Q(m) = 0(\bmod T')$. But on the other hand we have $|Q(m)| \leq \sum_{i=0}^{\ell} BK^i \leq (\ell+1)BK^\ell < T \leq T'$. Combining the two conditions, we get $Q(m) = 0$, and thus $y - m$ divides $Q(y)$. ■

Now we seem to have all the ingredients — all we need is to find out what they really mean - the number of parameters flowing around is huge and it is not clear if all the above combine to give any error-correction! To make sense of the above, lets us try using an inappropriate choice of parameters. Let us suppose $p_1 = p_2 = \cdots = p_n = p$. (Ahem! We this is somewhat inconsistent with the hypothesis that p_i's are relatively prime. But this is the best way to get a sense of the idea.) Then $N = p^n$, $T = p^t$, $K = p^k$. Fix some ℓ. We can apply Lemma 5 if we let $B' = N^{1/\ell} = p^{n/\ell}$, and then Lemma 3 can be applied if $B \approx 2^\ell p^{n/\ell}$. Lemma 6 can then be applied if $(\ell+1)BK^\ell < T$, which is well approximated by $p^{n/\ell + k\ell} < p^t$. So with all these outrageous approximations and assumptions, we see we are in a situation similar to the Reed-Solomon decoding case two lectures back: We can decode provided $n/\ell + k\ell < t$. The expression on the LHS is minimized by setting $\ell = \sqrt{n/k}$ and this gives decoding from $2\sqrt{nk}$ agreements. The following theorem is what can be obtained from a rigorous analysis of the above after some minor optimizations in the bounds on the coefficients of the polynomials, and is proven by Goldreich, Ron, and Sudan [3]. The main difference is a $\sqrt{\log p_n / \log p_1}$ factor in the amount of agreements required. This comes up because the p_i's are, unfortunately, not all equal.

Theorem 7 ([3]) *There exists a list-decoder for CRT codes that on input an instance with relatively prime integers $p_1, \ldots, p_n$, and information parameter k and agreement parameter t, recovers all messages with agreement t provided $t > \sqrt{2kn \log p_n / \log p_1}$ in polynomial time.*

Using the ideas of multiplicities one can (almost) get rid of the factor $\sqrt{2}$ in the above agreement; and this was done by Boneh [2]. Boneh uses the same multiplicities on each coordinate. Using multiplicities that vary as a function of the p_i's, gives further improvements and one can now decode from $t > \sqrt{(1+\epsilon)kn}$ error in time polynomial in $1/\epsilon$, n and $\log p_n$. This final result was given by Guruswami, Sahai, and Sudan [5].

Bibliographic notes

We never mentioned an unambiguous decoding algorithm for CRT codes, but one certainly exists. The first such algorithm was given by Mandelbaum [7]. Surprisingly this algorithm does not decode up to half the minimum distance, at least not in polynomial time. To get a decoder decoding up to half the minimum distance, one needs to combine this algorithm with a GMD decoder.[2] This was done by [5].

The first list-decoding algorithm for CRT codes was due to Goldreich et al.[4]. Boneh [2] and Guruswami et al. [5] improved the error-correction bounds subsequently. Guruswami et al. [5] also developed the machinery of ideal codes.

References

[1] Miklos Ajtai. The shortest vector problem is NP-hard for randomized reductions. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 10–19, Dallas, Texas, 23–26 May 1998.

[2] Dan Boneh. Finding smooth integers in short intervals using CRT decoding. *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 265–272, 2000.

[3] Oded Goldreich, Dana Ron, and Madhu Sudan. Chinese remaindering with errors. *IEEE Transactions on Information Theory*, 46(5):1330–1338, July 2000. Extended version appears as ECCC Technical Report TR98-062 (Revision 4), `http://www.eccc.uni-trier.de/eccc`.

[2]This could be a good, healthy, workout. The Mandelbaum algorithm is what you would get if you set $\ell = 1$ above and picked optimal bounds on the coefficients of the constant and degree one term in the Q-polynomial. Compute the number of errors this algorithm can recover from. Then show how to do a GMD like algorithm + analysis to recover from $(n-k)/2$ errors.

[4] Oded Goldreich, Ronitt Rubinfeld, and Madhu Sudan. Learning polynomials with queries: The highly noisy case. *SIAM Journal on Discrete Mathematics*, 13(4):535–570, November 2000.

[5] Venkatesan Guruswami, Amit Sahai, and Madhu Sudan. Soft-decision decoding of Chinese Remainder codes. In *Proceedings of the 41st IEEE Symposium on Foundations of Computer Science*, pages 159–168, Redondo Beach, California, 12-14 November 2000.

[6] Arjen K. Lenstra, Hendrik W. Lenstra, and Lászlo Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982.

[7] David M. Mandelbaum. On a class of arithmetic codes and a decoding algorithm. *IEEE Transactions on Information Theory*, 21(1):85–88, January 1976.

[8] Daniele Micciancio. The shortest vector in a lattice is hard to approximate to within some constant. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, pages 92–98, Palo Alto, California, 8–11 November 1998.