6.S897 Algebra and Computation	February 29, 2012
Lecture 7	
Lecturer: Madhu Sudan	*Scribe: Eric Price*

Today we enter the fun part of the course: factoring polynomials. Today we discuss univariate polynomials over finite fields; we'll move on later to the multivariate case, then discuss factoring over integers. It's actually really surprising than we can do this at all: a priori, there doesn't seem to be any reason we can factor in polynomial time. It seems hard to factor integers, so why should it be efficient for polynomials? Nevertheless, we shall see today that it is possible.

The approach we describe is due to Berlekamp. The plan is the following:

1. Factoring quadratics
2. Factoring polynomials with linear factors
3. General: $\text{poly}(n, \log q)$
4. Deterministic: $\text{poly}(n, p, \log q)$.

Throughout this talk, we will be factoring over $\mathbb{F}_q$, where $q = p^t$ for prime p. We will assume that $p \neq 2$ except in Section 4, where we describe how to modify the algorithms to handle that case.

1 Factoring univariate polynomials

1.1 Quadratics

Consider factoring a quadratic $f(x) = x^2 + ax + b = (x - \alpha)(x - \beta)$ over $\mathbb{F}_q$. Given (a, b) the task is to find (α, β).

First, we show how to factor when α is a quadratic residue and β is not. Then $\alpha^{(q-1)/2} = 1$ and $\beta^{(q-1)/2} \neq 1$, so $(x-\alpha) \mid x^{(q-1)/2}-1$ and $(x-\beta) \nmid x^{(q-1)/2}-1$. This implies that $\gcd(x^{(q-1)/2}-1, f) = x - \alpha$.

Note that we don't run the gcd algorithm on $x^{(q-1)/2} - 1$ as a vector of dimension $(q-1)/2$; instead, we compute $g(x) := x^{(q-1)/2} \bmod f$, and run $\gcd(g(x)-1, f)$. To compute $g(x)$, we compute $x^{2^i} \bmod f$ for $i \in [\log_2 q]$ via repeated squaring, and multiply together the appropriate terms mod f. Then we can compute $\gcd(g(x) - 1, f)$. Overall, this takes $\text{poly}(n, \log q)$ time.

Now let's consider the general quadratic, where we don't assume that exactly one root is a quadratic residue but we do assume $\alpha \neq \beta$. Given $c, d \in \mathbb{F}_q$, we consider $c^2 f(\frac{x-d}{c})$, which has roots $(\alpha c + d, \beta c + d)$. If $c \in \mathbb{F}_q^*$ and $d \in \mathbb{F}_q$ are chosen uniformly at random, then the new roots are uniformly at random from $\mathbb{F}_q^2$. Hence there is a one-half chance that exactly one root is a quadratic residue, in which case the previous algorithm will find it. So we can repeat until this occurs.

1.2 Linear products of distinct terms

In general, we have $f(x) = \sum_{i=0}^n f_i x^i = \prod_{i=1}^n (x - \alpha_i)$, with $f_n = 1$.

First, we assume the α_i are distinct. Then with the affine transform $\alpha_i \to \alpha_i c + d$, for each pair (α_i, α_j) there is a one-half chance that exactly one new root $\alpha_i c + d$ is a quadratic residue. Hence if $f' = c^2 f(\frac{x-d}{c})$ and $g = \gcd(x^{(q-1)/2} - 1, f')$, there is a one-half chance that $\alpha_i c + d$ and $\alpha_j c + d$ split over g and f'/g. We then repeat to factor g and f'/g.

Since each pair (i, j) splits with constant probability in each round, after $O(\log n)$ rounds each pair is split with $1 - 1/n^3$ probability. Hence all pairs are split with $1 - 1/n$ probability, in which case every polynomial is a monomial $x - \alpha_i$. So this lets us factor f if all the α_i are distinct.

1.3 Repeated roots

To find repeated roots, we apply a technique that generalize throughout the course. The idea is to find repeated factors via the *derivative* f': we compute $\gcd(f, f')$ to get a smaller polynomial containing the repeated factors, and recurse.

The derivative in finite fields is defined as you would expect:

Definition 1 *We define the* derivative f' *by*

- $(\alpha x^d)' = d\alpha x^{d-1}$
- $(f+g)' = f' + g'$

Claim 2 *The derivative obeys the following rules:*

1. ***Product rule.*** *(fg)' = f'g + g'f*
2. *If $g^i \mid f$, then $g^{i-1} \mid f'$.*

Then f' has smaller degree than f, and for any repeated root $(x-\alpha)^d$, $x - \alpha \mid \gcd(f, f')$. Then $\gcd(f, f')$ contains all the repeated factors and has smaller degree than f (assuming it is not zero, a case we discuss later). So we can recursively factor $\gcd(f, f')$ to get the repeated factors, divide them out, and factor the remainder as a product of non-repeated linear tems.

The aforementioned caveat is that $\gcd(f, f')$ can be zero, if $f' = 0$. But this only happens if $f(x) = \sum c_i x^{ip} = (\sum c_i^{1/p} x^i)^p$, where $c_i^{1/p}$ can be computed by $c_i^{1/p} = c_i^{p^{t-1}}$ for $q = p^t$. So if $f' = 0$, we can reduce to factoring $\sum c_i^{1/p} x^i$.

Combining the results so far, we can factor any polynomial that splits into linear factors. What if f has irreducible factors? We could extend the field and factor over the extension. This is kind of complicated, so we'll instead give an explicit method for finding degree d irreducible factors. However, before we do so, we will first give a deterministic algorithm to check if a polynomial f is irreducible. This has the benefit of turning our randomized factoring algorithm from a Monte Carlo algorithm (which has a tiny chance of giving the wrong output) into a Las Vegas algorithm (which always outputs the correct answer, but has a tiny chance of taking a long time).

2 Deterministic Reducibility Testing

Recall that $x^q - x = \prod_{\alpha \in \mathbb{F}_q} (x - \alpha)$. Therefore $f(x) \mid x^q - x$ if and only if f splits into distinct linear factors. Similarly, recall that if $f(x)$ is irreducible and degree d then $f(x) \mid x^{q^d} - x$. Then if $\gcd(f, x^{q^d} - x)$ is non-trivial, f must have an irreducible factor of degree dividing d. So to check if f is irreducible, we check these gcds for $d = 2, 3, \ldots, n$.

3 Higher degree irreducible factors

Suppose $q(x)$ has degree 2^n, but only has t coefficients, and $a(x)$ has degree d. Then we can compute $q(x) \mod a(x)$ in $\text{poly}(t, d, n)$ time via repeated squaring.

Then to factor f, we

- Eliminate repeated factors by gcd
- $f_0 \leftarrow f$
- For $i = 1$ to n

- $g_i \leftarrow \gcd(f_{i-1}, x^{q^i} - x)$
- $f_{i+1} = f_i / g_i$
- Factor g_i into degree i factors using the method described below.

- Output all factors recovered.

The essential case is to factor $g = g_1 \cdot g_2$, where $g_1 \neq g_2$ and both are irreducible of degree d. We consider the two fields $\mathbb{F}_q[x]/g_1$ and $\mathbb{F}_q[x]/g_2$. We want to find an irreducible h such that $h(x)$ is a quadratic residue in one field but not in the other one. The Chinese Remainder Theorem says that $\mathbb{F}_q[x]/(g_1 g_2)$ is isomorphic to $\mathbb{F}_q[x]/g_1 \times \mathbb{F}_q[x]/g_2$. So we choose a random $h \in \mathbb{F}_q[x]/(g_1 g_2)$, which makes ($h \bmod g_1, h \bmod g_2$) uniform from the product space. With 1/4 probability, h is a quadratic residue modulo g_1 and is not a quadratic residue modulo g_2.[1] We can check this by verifying that $\gcd(h(x)^{(q^d-1)/2} - 1, g) = g_1$.

In general, for $g = \prod g_i$, then we get that $g' = \gcd(h(x)^{(q^d-1)/2} - 1, g)$ is a nontrivial, smaller polynomial with good probability. So we can repeat on g' and g/g' until we reach degree d factors, which are the desired irreducible factors of f.

For the running time: we've shown polynomial time. Efforts have been made to speed it up, with the best not being quite linear—more like $n^{1.5}$.

4 $p = 2$

What if $q = 2^t$? Then $(q-1)/2$ is not an integer, so $\gcd(f, x^{(q-1)/2} - 1)$ doesn't make sense. However, in this case we have

$$x^{2^t} - x = \mathrm{Tr}(x)(\mathrm{Tr}(x) - 1)$$

where the *trace* $\mathrm{Tr}\colon \mathbb{F}_{2^t} \to \mathbb{F}_2$ satisfies $\mathrm{Tr}(x+y) = \mathrm{Tr}(x) + \mathrm{Tr}(y)$ and has degree 2^{t-1}. Explicitly, it is $\mathrm{Tr}(x) = x + x^2 + \ldots + x^{2^{t-1}}$. We can check $\mathrm{Tr}(x+y) = \mathrm{Tr}(x+y)^2 = \sum x^{2i} + y^{2i} = \mathrm{Tr}(x) + \mathrm{Tr}(y)$ to verify the claim.

Then $(x - \alpha) \mid \mathrm{Tr}(x)$ for half the $\alpha \in \mathbb{F}_q$, so in the previous sections we simply replace $\gcd(f, x^{(q-1)/2} - 1)$ with $\gcd(f, \mathrm{Tr}(x))$ to factor into linear factors. To factor into irreducibles, we replace $\gcd(g, h(x)^{(q-1)/2})$ by XXX.

5 Deterministic factoring

The above algorithms involve randomly shifting the roots so they split. Can we factor without this randomness? We will see that the answer is yes in fields of small characteristic – $\mathrm{poly}(n, p, \log q)$ time.

Consider the problem of factoring g. Our goal will be to find $h(x)$ such that

$$h^p - h = 0 \mod g. \tag{1}$$

Two obvious solutions exist: $h = g$ and $h = \alpha \in \mathbb{F}_p$. We will require $\deg h < \deg g$ and $\deg h \geq 1$.

First, we show why this is sufficient. We know that

$$h^p - h = \prod_{\alpha \in \mathbb{F}_p} (h - \alpha).$$

Thus we will have $g \mid \prod_\alpha (h - \alpha)$, so there exists an α with $\gcd(g, h - \alpha)$ non-trivial. Since h has smaller degree than g, we make progress.

[1] Excluding the cases where h divides one of g_1 and g_2. But these cases are rare, with q^{-d} probability; even if they weren't rare, one could test $\gcd(h, g)$.

So it suffices to find such an h. Note that this is a *linear system*: if h_1 and h_2 satisfy (1), then $(h_1 + h_2)^p = h_1^p + h_2^p = h_1 + h_2$ by the characteristic and by (1). We don't have time to go into details, but linear systems can be solved in polynomial time.