6.S897 Algebra and Computation	March 14, 2012

Lecture 11

Lecturer: Madhu Sudan *Scribe: TB Schardl*

This lecture concludes our discussion of bivariate polynomial factorization. We first focus on a couple remaining key ingredients to make the algorithm work. We then turn our attention to multivariate polynomial factorization. First, we introduce the notion of representing the polynomial as a black box. Using this representation, we then sketch the high level ideas of the factorization algorithm. Our goal today is to wrap up the discussion of polynomial factorization and move on to new topics by next lecture.

1 Factoring bivariate polynomials

Let us begin by reviewing the algorithm for factoring bivariate polynomials.

Split($f \in \mathbb{F}[x, y], \deg(f) = d$):

0. Preprocess f to ensure it has no repeated factors.
   If $\frac{\partial f}{\partial x} = 0$ and $\frac{\partial f}{\partial y} = 0$, then $f = g^p$ for some $g \in \mathbb{F}$, and thus return g. To compute g, notice that, if $f = \sum_{ij} c_{ij} x^i y^j$, then $g = \sum_{ij} c_{ij}^{1/p} x^{i/p} y^{j/p}$. Because the exponents i/p and j/p will be integers whenever c_{ij} is non-zero, g can be computed by calculating $c_{ij}^{1/p}$ for each i, j. Furthermore, each $c_{ij}^{1/p}$ can be computed using the equation $c_{ij}^{1/p} = c_{ij}^{q/p}$.

   Otherwise, if $\frac{\partial f}{\partial x} = 0$, then evaluate Split($f(y, x), d$), thereby swapping the variables x and y. Finally, if $g = \gcd(f, \frac{\partial f}{\partial x}) \neq 1$, then return $(g, f/g)$ for reasons previously discussed.

1. Pick some $\beta \in \mathbb{F}$ such that $f(x, \beta)$ has no repeated factors, and set $f(x, y) \leftarrow f(x, y + \beta)$.

2. Factor $f = g_1 \cdot g_2 \cdot \ldots \cdot g_k \pmod{y}$. Notationally, let $g = g_1$ and $h = g_2 \cdot \ldots \cdot g_k$. Make sure g is irreducible and monic.

3. Lift $f = g^{(t)} \cdot h^{(t)} \pmod{y^t}$, where t is chosen to be sufficiently large, i.e. $t > d^2$.

4. Use $g^{(t)}$ to get information for an irreducible factor of f by jumping $g^{(t)} \to \tilde{g}$. This jump is done by solving $\tilde{g} = g^{(t)} \cdot \tilde{h} \pmod{y^t}$ such that $\deg(\tilde{g}) \leq d$ and $\deg_x(\tilde{g})$ is minimal.

5. Return $(\tilde{g}, f/\tilde{g})$.

We now justify step 5, or in particular, that $\tilde{g}$ divides f. To prove this property, we argue that $\tilde{g}$ is one of the factors of f through a sequence of small claims.

Before arguing this sequence of claims, let us establish some notation. First, we write $f = f_1 \cdot f_2 \cdot \ldots \cdot f_\ell$, where f_i is irreducible for $1 \leq i \leq \ell$. As previous lectures have shown, after computing $f \bmod y$, each f_i may split further into factors $f_i = f_{i1} \cdot f_{i2} \cdot \ldots \cdot f_{in_i} \pmod{y}$, where each $f_{ij} \in \mathbb{F}[x]$ is irreducible.

With this notation, we start making claims to help us show the validity of step 5. First, we argue that the g computed from factoring f mod y is a factor of one of the f_i's.

Claim 1 *The factor $g = f_{ij}$ for some i, j.*

Proof This claim follows from unique factorization. ■

Next, we argue that the $\tilde{g}$ term computed from the jump step is one of the factors of f.

Claim 2 *If $g = f_{ij}$ for some i, j, then $\tilde{g} = f_i$ for the same i.*

We argue this claim through a sequence of smaller steps. In particular, we consider a hypothetical Hensel lifting, and we first argue that the lift of the factor f_i of f is closely related to the lift of f.

Claim 3 *Suppose we lift $f_i = f_{ij} \cdot \prod_{m \neq j} f_{im} \pmod{y}$. Let $g = f_{ij}$ and $h_0 = \prod_{m \neq j} f_{im}$, so $f_i = g \cdot h_0 \pmod{y}$. After lifting, we have $f_i = g_0^{(t)} \cdot h_0^{(t)} \pmod{y^t}$. Then there exists some polynomial $u \in \mathbb{F}[x, y]$ such that $g^{(t)} = g_0^{(t)}(1 + u \cdot y^{t/2})$, or equivalently, $g^{(t)}(1 - u \cdot y^{t/2}) = g_0^t \pmod{y^t}$.*

Proof The proof follows from the uniqueness of Hensel liftings (see last lecture). We know that $f = g^{(t)} \cdot h^{(t)} \pmod{y^t}$. Furthermore, we have

$$\begin{aligned} f &= \prod_m f_m \\ &= f_i \cdot \prod_{m \neq i} f_m \\ &= g_0^{(t)} \cdot h_0^{(t)} \cdot \prod_{m \neq i} f_m \pmod{y^t} . \end{aligned}$$

■

We now argue that $\tilde{g}$ is related to the factor f_i. This argument uses a common paradigm in properties of these algorithms. In this paradigm, we first show that *some* solution to the problem is valid, and then we show that there is little room to maneuver around the valid solution.

Using this paradigm, let us first show that there is some solution to the jumping problem such that $\tilde{g} = f_i$.

Claim 4 *There exists some $\tilde{h_0}$ such that $(f_i, \tilde{h_0})$ is a valid solution to the jump problem, ignoring minimality.*

Proof We have that $f_i = g_0^{(t)} \cdot h_0^{(t)} \pmod{y^t}$. From the previous claim, this can be rewritten as $f_i = g^{(t)} \cdot (1 - u \cdot y^{t/2}) h_0^{(t)} \pmod{y^t}$. Letting $\tilde{g} = g^{(t)}$ and $\tilde{h_0} = (1 - u \cdot y^{t/2}) h_0^{(t)}$ proves the claim. ■

Finally, we show that $\tilde{g}$ and f_i share a common factor. Because f_i is irreducible, this implies $\tilde{g} \sim f_i$ as desired.

Claim 5 *Suppose both $(f_i, \tilde{h_0})$ and $(\tilde{g}, \tilde{h})$ are both valid solutions (with small degree) to the jump problem. Then f_i and $\tilde{g}$ share a common factor.*

Proof We show this claim by examining the resultant $\mathrm{Res}_x(f_i, \tilde{g})$ and showing that it must be 0, which implies that f_i and $\tilde{g}$ share a factor. To show that $\mathrm{Res}_x(f_i, \tilde{g}) = 0$, we assume the contrapositive in order to arrive at a contradiction.

Suppose that f_i and $\tilde{g}$ have no common factor. As a result, their resultant $R(y) = \mathrm{Res}_x(f_i, \tilde{g})$ is nonzero, has degree at most d^2, and is in the ideal of $(\tilde{g}, f_i)$. Consequently, there exist polynomials $A, B \in \mathbb{F}[x, y]$ such that $R = A \cdot f_i + B \cdot \tilde{g}$. Substituting in $f_i = g^{(t)} \cdot \tilde{h_0} \pmod{y^t}$ and $\tilde{g} = g^{(t)} \cdot \tilde{h} \pmod{y^t}$ and rearranging terms produces the equation

$$R = g^{(t)}(A \cdot \tilde{h_0} + B \cdot \tilde{h}) \pmod{y^t} .$$

We now notice two things. First, the polynomial $g^{(t)}$ is a monic polynomial in x. Second, because the highest degree term in $(A \cdot \tilde{h_0} + B \cdot \tilde{h})$ must be nonzero, it must contain a highest degree term in x.

Consequently, the highest degree x term cannot be eliminated to get $R(y)$, and thus this scenario cannot happen. ■

As an aside, notice that computing the resultant eliminates a variable, which is a useful property for a variety of algebraic computations. For example, suppose we wish to find common 0's between two bivariate polynomials. One method to find such 0's is to take the resultant of those polynomials, find solutions where the resultant is 0, and then use those solutions in y to look for 0's in x. We shall use and expand on this idea in future lectures.

2 Factoring polynomials over the integers

We now briefly consider the problem of factoring a polynomial f over the integers. This problem is soluble using a similar algorithm to Split for factoring bivariate polynomials, substituting a chosen prime p in place of y. The modified algorithm is summarized below.

Split-Z($f \in \mathbb{Z}[x], \deg(f) = d$):

0. Preprocess f to ensure it has no repeated factors.
1. Pick a prime $p \in \mathbb{Z}$ such that f has no repeated factors modulo p. Factor $f = g \cdot h \pmod{p}$.
2. Lift $f = g^{(t)} \cdot h^{(t)} \pmod{p^t}$, where t is chosen to be sufficiently large, i.e. $t > d^2$.
3. Use $g^{(t)}$ to get information for an irreducible factor of f by jumping $g^{(t)} \to \tilde{g}$. This jump is done by solving $\tilde{g} = g^{(t)} \cdot \tilde{h} \pmod{p^t}$ such that $\deg(\tilde{g}) \leq d$ and $\deg_x(\tilde{g})$ is minimal.
4. Return $(\tilde{g}, f/\tilde{g})$.

One question regarding the validity of this algorithm is, "How large must p^t be?" The answer to this question depends on the size of the coefficients of the original polynomial f. Once we bound the size of these coefficients, the resultant will behave as expected, and the rest of the algorithm follows.

We now sketch the arguments for bounding the size of the coefficients of the factors, thereby bounding the size of p^t, in terms of the coefficients of f. Consider a polynomial $f \in \mathbb{Z}[x]$, written as $f = \sum_i f_i x^i$, and suppose that $|f_i| \leq 2^b$ for all i. For another polynomial g that divides f, we bound the size of the coefficients of g by justifying two claims. First, we argue that the complex roots are "small."

Claim 6 *All complex roots of f are bounded by $n \cdot 2^b$.*

Sketch of Proof Suppose that some root α of f is large, or formally, suppose $|\alpha| > n \cdot 2^b$. Then the first term in the expression of f is $f_n(n \cdot 2^b)^n > \sum_{i=0}^{n-1} f_i(n \cdot 2^b)^i$. ■

Next, we argue that, if the complex roots of g are small, then the coefficients of g are bounded in terms of the coefficients of f.

Claim 7 *If the complex roots of g are bounded, then so are the coefficients of g.*

Sketch of Proof Assume that g is monic, and let $g \in \mathbb{Q}[x]$. Suppose g is split into complex terms. In order to transform the factors g into factors of $f \in \mathbb{Z}[x]$, we must multiply these factors by some integer, whose size is bounded by the largest term in f. Hence, the coefficients of g are bounded in terms of the coefficients of f. ■

3 Factoring multivariate polynomials

To conclude, let us turn our attention to the problem of factoring polynomials of more than 2 variables. One idea to approach this problem is apply a similar algorithm to SPLIT, using step 2 to eliminate one variable of the polynomial at a time until we are left with factoring a univariate polynomial. This scheme introduces blowup, however, in both time and the representation of the polynomials involved for each variable, and while this idea works for polynomials over a constant number of variables, for polynomials over more variables this blowup can be problematic. It turns out that several alternative schemes for factoring multivariate polynomials introduce similarly large blow-ups in time and representation, because these algorithms explicitly represent the terms of the polynomial.

An alternative representation of multivariate polynomials as a black box allows us to factor more efficiently. In this representation, a polynomial is represented by some black box P, which takes as input some assignment $(\alpha_1, \ldots, \alpha_n)$ of the n variables and produces $P(\alpha_1, \ldots, \alpha_n)$. With this representation, the goal of factoring the polynomial P is to produce the set of black boxes for the factors of P.

Assuming we use a black box representation of polynomials, the rough idea for factoring multivariate polynomials works as follows. Consider a polynomial $P(y_1, \ldots, y_n)$ that is the product of k irreducible factors $P = P_1 \cdot P_2 \cdot \ldots \cdot P_k$. Suppose the polynomial is "nice" in that $P(y_1, 0, \ldots, 0) = \prod_i P_i(y_1, 0, \ldots, 0)$, where the $P_i(y_1, 0, \ldots, 0)$'s are irreducible and pairwise distinct. While this univariate polynomial is more convenient to work with, it does not immediately allow us to compute $P(\alpha_1, \ldots, \alpha_n)$. To address this issue, we instead work with the bivariate polynomial $\tilde{P}(t_1, t_2) = P(t_1 + \alpha_1 t_2, \alpha_2 t_2, \alpha_3 t_2, \ldots, \alpha_n t_2)$, which we can use to compute either $P(y_1, 0, \ldots, 0)$ or $P(\alpha_1, \ldots, \alpha_n)$ by choosing t_1 and t_2 appropriately. Notice that, because P splits into k factors, $\tilde{P}$ also splits into k factors. Furthermore, $\tilde{P}$ does not split into more factors, since setting $t_2 = 0$ produces a polynomial with k factors.

Assuming P is a nice polynomial, we can factor the black box polynomial P as follows.

0. In preprocessing, factor $P(y_1, 0, \ldots, 0) = \prod_i P_i(y_1, 0, \ldots, 0)$. Notice that $P(y_1, 0, \ldots, 0)$ can be represented explicitly, because it is a univariate polynomial.
1. Compute $\tilde{P}(t_1, t_2) = P(t_1 + \alpha_1 t_2, \alpha_2 t_2, \ldots, \alpha_n t_2)$ by interpolation.
2. Factor $\tilde{P}$ into factors $\tilde{P} = Q_1 \cdot Q_2 \cdot \ldots \cdot Q_\ell$.
3. Find j such that $Q_j(t_1, 0) = P_1(y_1, 0, \ldots, 0)$. Exactly one such Q_j exists, since all pairwise P_i's are pairwise distinct and irreducible. Return $Q_j(0, 1)$.

This procedure relies on P being a nice polynomial, i.e. a polynomial whose factors can be discovered by considering a single line. A natural question to ask is, "What are the chances that we get such a nice polynomial?" Unfortunately, there are some polynomials that are irreducible but can be factored along any line. Consequently, this approach seems doomed to failure.

It turns out, however, that we can use a similar approach to factoring multivariate polynomials by considering a plane instead of a line. According to the ***Hilbert Irreducibility Theorem*** (which is really due to Kaltofen), if $P \in \mathbb{F}[y_1, \ldots, y_n]$ is irreducible, then we have

$$\Pr_{\bar{\alpha}, \bar{\beta}, \bar{\gamma} \in \mathbb{F}^n} \left\{ P_{\bar{\alpha}, \bar{\beta}, \bar{\gamma}}(t_1, t_2) = P(\bar{\alpha} + t_1 \bar{\beta} + t_2 \bar{\gamma}) \text{ is reducible} \right\} \leq \deg(P)^4 / |\mathbb{F}| \ ,$$

where $\left\{ \bar{\alpha} + t_1 \bar{\beta} + t_2 \bar{\gamma} \mid t_1, t_2 \in \mathbb{F} \right\}$ represents the surface. Applying this theorem, we can adapt our earlier technique by preprocessing the black box polynomial P along a random plane to produce an explicitly represented trivariate polynomial, and then applying our previous SPLIT algorithm to find factors.