CS 812
Lecture 39
Monday 4/27/20

Factoring Polynomials With Rational Coefficients.

Reference: See Lecture 38

Background

Polynomial factoring is a key task in algebra. As taught in high school, it is not particularly algorithmic. Rather, the student is taught some heuristics, with the implicit assurance that these will always work for any problem assigned by the teacher. However, Kronecker had proved in the 19th century that polynomials with integer coefficients could be factored by an algorithm. (See van der Waerden, Modern Algebra, 1953, pp. 77 ff.)

In the late 1960s, a practically suitable method was published by Zassenhaus (J. Number Theory, 1969). The basic idea was to use Hensel's lemma to factor the polynomial over the p-adic integers, and then recover the "true" factors by multiplying together one of the (finitely many) subsets of the p-adic factors. In the 1970's this was refined and analyzed by George Collins and his student David Musser. In particular, Musser proved that it did not have a worst-case polynomial running time.

Therefore, the publication by LLL in 1982 of a polynomial-time algorithm for factoring in $\mathbf{Z}[X]$ was a major event. In this lecture, we'll present a related algorithm by Kannan, Lenstra, and Lovasz that exploits algebraic number inference.

Mathematical Background

Let $f(X)$ be a polynomial with rational coefficients that we want to factor. Normalize f by clearing denominators, so that $f(X) = a_0 + a_1X + ... + a_nX^n$ with $a_i \in \mathbf{Z}$. We can assume that the gcd of the a_i's is 1. In this case, we say that f is primitive.

Warning: don't confuse this with primitive polynomials over finite fields, it means something completely different.

Gauss's Lemma: If a primitive polynomial is irreducible in $\mathbf{Z}$[X], it's irreducible in $\mathbf{Q}[X]$. More generally, the factorization of any primitive polynomial in $\mathbf{Z}[X]$ is the same as its factorization over $\mathbf{Q}[X]$. For proof, see, e.g. S. Lang, Algebra, 1965, p. 127.

This means that it will suffice to factor f (which we have cooked to be primitive) over the integers. Allowing rational coefficients won't give us any more factors.

To keep the complexity of this lecture within bounds, we'll assume an exact arithmetic model.

The KLL Polynomial Factoring Procedure

Input: f in $\mathbf{Z}[X]$, primitive.

1. Compute a zero α of $f(X)$ (might be complex)
2. Use the LLL algorithm to find a $g \in \mathbf{Z}[X]$ of lowest degree such that $g(\alpha) = 0$.
3. Conclude that g divides f.

Actually, this is a splitting procedure. If the algorithm returns a constant multiple of f for g, the algorithm stops and declares f irreducible.

We must check that $g \mid f$. The set $I = \{h \in \mathbf{Q}[X] : h(\alpha) = 0\}$ is an ideal in $\mathbf{Q}[X]$.

This means it's closed under $+, -$ and multiplication by arbitrary elements of $\mathbf{Q}[X]$. You can readily check that I has these properties.

It is known that any ideal in $\mathbf{Q}[X]$ is principal, that is, the set of multiples of one element. (See S. Lang, Algebra, p. 120.) g is an element of least degree for which $g(\alpha) = 0$, hence it must be a constant multiple of the generator of the ideal. In $\mathbf{Q}[X]$, then, g divides f. Appealing to Gauss's lemma, we see that g (more precisely, a primitive multiple of it) divides f in $\mathbf{Z}[X]$.

Step 2 is similar to what was done in the last lecture. Since we don't know the degree d of f, we will try $d = 1, 2, \ldots, n$. One potential problem is that α can be complex. Write $\alpha = \beta + i\gamma$, and let

$$\alpha^j = \beta_j + i\gamma_j, \qquad j = 0, ..., n.$$

Choose a suitably large C and feed the vectors

$$\begin{array}{lcccccccc}
b_0 & = & (C\beta_0 & C\gamma_0 & 1 & 0 & 0 & \ldots & 0) \\
b_1 & = & (C\beta_1 & C\gamma_1 & 0 & 1 & 0 & \ldots & 0) \\
b_2 & = & (C\beta_2 & C\gamma_2 & 0 & 0 & 1 & \ldots & 0) \\
 & \cdots & & & & & & & \\
b_d & = & (C\beta_d & C\gamma_d & 0 & 0 & 0 & \ldots & 1)
\end{array}$$

into the basis reduction algorithm. If B is a bound on the absolute value of g's coefficients, we proved in the last lecture that value of C with

$$\log C = O(d^2 + d \log B)$$

is sufficient to recover a degree d polynomial g with $g(\alpha) = 0$, should any exist.

How big does B need to be? It follows from work of Mignotte that the largest coefficient of g is no more than

$$\sqrt{n+1}\, 2^n F,$$

where F is the largest (in absolute value) coefficient of f. (A good review of coefficient bounds for factors of polynomials appears in J. Abbott, J. Symbolic Computation, 2013.)

How Do We Find a Zero of f?

Step 1. Make sure f is squarefree

Replace f by $f/\gcd(f', f)$.

Step 2. Bound the zeros of f.

By the Gershgorin circle theorem (see, e.g., Johnson and Reiss, Numerical Analysis), any root α of $X^n + \cdots + a_0 = 0$ must satisfy

$$|\alpha| \leq 1 + \max_{1 \leq i < n} |a_i|.$$

Step 3. Look for real roots, then (if this fails), look for complex ones.

a) The classic method to locate the real roots of a polynomial is Sturm's theorem. (Reference: van der Waerden, op. cit., p. 220.) Let $f_0 = f$, $f_1 = f'$, then apply the Euclidean algorithm:

$$\begin{aligned} f_0 &= q_1 f_1 - f_2 \\ f_1 &= q_2 f_2 - f_3 \\ &\cdots \\ f_{r-1} &= q_r f_r \end{aligned}$$

(Note the minus signs!) If $a < b$ are not zeroes of f then f has

$$\#\text{ of sign changes in}(f_0(a), f_1(a), \ldots, f_r(a))$$

$$-\#\text{ of sign changes in}(f_0(b), f_1(b), \ldots, f_r(b))$$

real roots in $[a, b]$. Here a sign change means a pattern of the form $+0^*-$ or -0^*+, and we count a multiple root only once. (Using regular expression notation here.)

We start with $a = -R$, $b = R$, where R is the Gershgorin circle bound plus 1. If f has no real roots, we find this out at the first step. Otherwise, we use binary search (combined with Sturm's theorem) to locate intervals containing exactly one real root of f. It is of interest to ask how many steps of binary search we will need. Using a result of S. M. Rump [Math. Comp. 1979], it can be shown that if $\alpha < \beta$ are real zeroes of a degree d polynomial in $\mathbf{Z}[X]$ whose coefficients are bounded by F in absolute value, then

$$-\log(\beta - \alpha) = O(d \log(dF)).$$

This shows that polynomially many bisections suffice to isolate the roots (we have, in effect, a binary tree with $\leq d$ leaves, and all root-leaf paths $O(d \log(dF))$ in length). Once the roots are isolated, we can further approximate each one by binary search. (Since f has no multiple roots, an isolating interval $[a, b]$ must have $f(a)f(b) < 0$.)

If f has odd degree, of course, we can skip all this and just find a root by binary search.

b) Finding complex zeroes.

There are two ways to do this.

1. Use resultants to reduce to finding real roots. (We follow Uspensky, Theory of Equations, p. 290.) Let f be a polynomial with real coefficients and no real roots. By Taylor's theorem,

$$f(x+iy) = f(x) + if'(x)y - 1/2f''(x)y^2 + \cdots = 0$$

implies

$$\begin{aligned} f(x) - 1/2!f''(x)y^2 + \cdots = 0, \\ f'(x) - 1/3!f'''(x)y^2 + \cdots = 0. \end{aligned} \qquad (*)$$

Using the resultant, eliminate y^2 from these equations to obtain a polynomial $r(x)$ (of degree $\binom{n}{2}$). Each pair of complex zeroes $x \pm yi$ of f leads to a real zero of r, which can be found using Sturm's theorem. Substituting these real zeroes back into (*), we obtain the corresponding values of y. One potential difficulty with this method is that the resultant might vanish identically. (I do not know if it can.) An actual difficulty is that the resultant has, in general, degree $n(n-1)/2$. Another problem is that r can have multiple roots, so it must be "purified" by another gcd computation. [Possible reference: T. B. Sprague, Trans. Roy. Soc. Edinburgh v. 30 (1882), part II.]

2. Use the "argument principle" from complex analysis. If f is a polynomial, and B is a rectangle with no zeroes of f on its boundary D, then the number of zeroes f has inside R equals

$$\frac{1}{2\pi i}\int_D \frac{f'(z)}{f(z)}dz.$$

We start by taking a box big enough to contain all zeroes of f, then do binary search, by dividing B horizontally, then vertically, then horizontally again, etc. The only difficulty with this method is that it requires a numerical determination of an integral, albeit not a very accurate one.

One can, however, use an algebraic version of the argument principle. By considering the integral of f'/f over the real axis, one can prove the following result.

Theorem: Let $f(X) = f_0(X) + if_1(X)$, where f_0 and f_1 have real coefficients. Let the Euclidean algorithm for f_0, f_1 yield

$$\begin{aligned} f_0 &= q_1 f_1 - f_2 \\ f_1 &= q_2 f_2 - f_3 \\ &\cdots \\ f_{r-1} &= q_r f_r \end{aligned}$$

(again, note the minus signs). If f has no real roots, and f_1 is not identically 0, then

$$\#\text{ of roots above the real axis } - \#\text{ of roots below the real axis}$$

equals

$$\#\text{ of sign changes at } +\infty \text{ of } f_0, f_1, \ldots, f_r$$

$$-\ \#\text{ of sign changes at } -\infty \text{ of } f_0, f_1, \ldots, f_r.$$

The sign of a polynomial at $\pm\infty$ can be ascertained from its degree and leading coefficient.

[The idea of using Sturm's theorem in this way is due to Routh. See M. Marden, The Geometry of the Zeroes of a Complex Polynomial, p. 132.]

We can use this in a binary search process to obtain imaginary parts of the roots of f. For the degenerate case where $f_1 = 0$, the polynomial f has equally many roots above and below the real axis, since its coefficients are real.