

1 Polynomial Reconstruction

1.1 Overview

The general reconstruction problem is as follows. Let F be a field (not necessarily finite). Given two sequences $\{x_1, \dots, x_n\} \in F$ and $\{y_1, \dots, y_n\} \in F$, we want to find a polynomial $f(x) \in F[x]$ such that $f(x_i) = y_i$ for “enough” values of i .

The notable difference here between interpolation and reconstruction is that we do not specify the particular values of i for which $f(x_i) = y_i$ and by doing so we avoid the $\binom{n}{\deg f}$ running time of the naïve algorithm which interpolates every possible subset.

1.2 Specialized Polynomial Reconstruction

By restricting the degree of $f(x)$ and setting a lower bound on the number of values of i for which $f(x_i) = y_i$, we can narrow the set of possible solutions down to either one or zero solutions and obtain a simple algorithm to find a solution if it exists. Specifically, given a field F , $x_1, \dots, x_n \in F$ distinct and $y_1, \dots, y_n \in F$ choose a $k \in \mathbb{N}, k < n$ and seek f such that

1. $\deg f(x) < k$
2. $\#\{i : f(x_i) \neq y_i\} \leq \frac{n-k}{2}$

then the algorithm will either give a unique $f(x)$ which satisfies the conditions or will signal that no such $f(x)$ exists.

1.3 Algorithm for Polynomial Reconstruction

To find an $f \in F[x]$ which satisfies the above restrictions, this algorithm finds polynomials $E, N \in F[x]$ such that

1. E monic with $\deg E \leq \frac{n-k}{2}$
2. $\deg N \leq \frac{n+k}{2} - 1$
3. $N(x_i) = y_i E(x_i) \forall i = 1, \dots, n$

If $E \mid N$ and $\deg N/E < k$, then N/E is a polynomial of degree $< k$ for which $N(x_i)/E(x_i) = y_i$ for at most $\deg E \leq \frac{n-k}{2}$ values. Hence $f = N/E$ is a solution to the original problem.

If $E \nmid N$ or $\deg N/E \geq k$, then (as we shall see later), there is no solution to the original problem.

1.3.1 Description of Algorithm

To find $N, E \in F[x]$, note that condition (3) implies that

$$N(x_i) = \sum_{j=0}^{\frac{n+k}{2}-1} n_j x_i^j = y_i \sum_{j=0}^{\frac{n-k}{2}} e_j x_i^j = y_i E(x_i)$$

for all (x_i, y_i) . But this is a linear system of n equations in $\frac{n+k}{2} + \frac{n-k}{2} + 1 = n+1$ unknowns, hence there is at least one solution and, by dividing by the leading nonzero coefficient of E , we can assume that E is monic.

Note also that there is a solution with $E \neq 0$ since, if $E = 0$, then $N = 0$ because $\deg N < n$ and $N(x_i) = 0$ for all $1 \leq i \leq n$.

1.3.2 Proof of Uniqueness

Although the solution to the linear system above is not unique, we show that N/E is unique.

Lemma 1. *If N_1, E_1 and N_2, E_2 are two sets of solutions to conditions (1), (2), and (3) above with $E_i \mid N_i$ then $N_1/E_1 = N_2/E_2$.*

Proof. Note that, by condition (3), $(N_1(x_i)E_2(x_i) - N_2(x_i)E_1(x_i))y_i = 0$ for all i . If $y_i \neq 0$, we get $N_1(x_i)E_2(x_i) - N_2(x_i)E_1(x_i) = 0$. If $y_i = 0$, we get $N_1(x_i) = N_2(x_i) = 0$ and it follows that, again, $N_1(x_i)E_2(x_i) - N_2(x_i)E_1(x_i) = 0$. Since $\deg(N_1E_2 - N_2E_1) \leq \frac{n-k}{2} + \frac{n+k}{2} - 1 = n-1$ we conclude that $N_1E_2 = N_2E_1$ and so $N_1/E_1 = N_2/E_2$ as required. \square

1.3.3 Proof of Sufficiency

We need to know whether the algorithm will always produce a solution if a solution to the original problem exists. Suppose f is a solution to the original problem. Define E as

$$E(x) := \prod_{\substack{i=1 \\ f(x_i) \neq y_i}}^n (x - x_i)$$

and define $N = fE$ so that $N/E = f$.

Now we check that N, E satisfy the three conditions. Since $\#\{i : f(x_i) \neq y_i\} \leq \frac{n-k}{2}$ we have $\deg E \leq \frac{n-k}{2}$ and E is monic by construction so condition (1) holds. Furthermore, since $\deg f < k$ and $\deg E \leq \frac{n-k}{2}$ we have that $\deg N = \deg f + \deg E \leq k-1 + \frac{n-k}{2} = \frac{n+k}{2} - 1$ and so condition (2) holds. Finally, $N(x_i) = f(x_i)E(x_i)$ so $N(x_i)$ is equal to $y_i E(x_i)$ when $f(x_i) = y_i$. If $f(x_i) \neq y_i$ then $E(x_i) = N(x_i) = 0$ so again $N(x_i)$ is equal to $y_i E(x_i)$.

By the previous section, we know that for any N, E which satisfy the three conditions and for which $E \mid N$, then N/E is unique. Hence if the algorithm does not find a suitable N, E , then no such pair exists and so there is no solution to the original problem.

1.4 Reconstruction Example

To see how the algorithm works in detail, let us consider the relatively simple case where $n = 3$ and $k = 1$. Then we are looking for a constant polynomial $f \in F$ such that $f = y_i$ for at least two values of i . Clearly, this can only occur when two of the y_i are the same.

The above algorithm finds polynomials $N(x) = n_0 + n_1x$ and $E(x) = e_0 + e_1x$ such that $N(x_i) = y_iE(x_i)$. For simplicity, let $x_i = i$ and then, to find the coefficients n_0, n_1, e_0, e_1 we must solve the following equation

$$\begin{pmatrix} 1 & 1 & -y_1 & -y_1 \\ 1 & 2 & -y_2 & -2y_2 \\ 1 & 3 & -y_3 & -3y_3 \end{pmatrix} \begin{pmatrix} n_0 \\ n_1 \\ e_0 \\ e_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

which has the general solution

$$\begin{pmatrix} n_0 \\ n_1 \\ e_0 \\ e_1 \end{pmatrix} = c \begin{pmatrix} 3y_1y_2 - 4y_1y_3 + y_2y_3 \\ -y_1y_2 + 2y_1y_3 - y_2y_3 \\ -y_1 + 4y_2 + 3y_3 \\ y_1 - 2y_2 + y_3 \end{pmatrix}$$

for any $c \in F$ except when $y_1 = y_2 = y_3$ in which case a solution is $N(x) = y_1x$ and $E(x) = x$ or $N(x) = y_1$ and $E(x) = 1$.

To see what happens when two of the y_i are the same, suppose that $y_1 = y_2 \neq y_3$, then

$$\begin{aligned} N(x) &= 3y_1(y_1 - y_3) + y_1(y_3 - y_1)x \\ E(x) &= 3(y_1 - y_3) + (y_3 - y_1)x \\ N(x)/E(x) &= y_1 \end{aligned}$$

and so $N/E = y_1$ is a solution to the original problem.

To see what happens when all the y_i are distinct, note that the general case is rather complicated so take $y_1 = 1, y_2 = 2, y_3 = 4$ to be an illustrative example. Then $N(x) = -2 - 2x$ and $E(x) = -5 + x$ so that $N(1) = E(1) = -4$, $N(2) = 2E(2) = -6$ and $N(3) = 4E(3) = -8$, but $E \nmid N$. In this case, the algorithm signals that there is no solution.