Vector Rational Number Reconstruction

Curtis Bright   Arne Storjohann

Abstract

We develop an efficient deterministic algorithm for performing rational number reconstruction on a vector of images modulo some modulus. When used with output sensitive p-adic lifting or Chinese remaindering for linear system solving, about half of the lifting steps can be saved.

Rational Number Reconstruction

- Given an integer residue $a \pmod{M}$, find a rational number n/d such that $a \equiv n/d \pmod{M}$.
- Would like the solution n/d to be unique, so we require the solution pair (n, d) be small:
$$|n| \leq N, \quad 0 < d \leq N$$
for a given bound N.
- If $M > 2N^2$ then the solution (if any) is unique.

The Vector Version

- Given a vector $\boldsymbol{a} \in \mathbb{Z}^n$ of images modulo M and a target length N, find a vector $\boldsymbol{n}/d \in \mathbb{Q}^n$ such that
$$\boldsymbol{a} \equiv \boldsymbol{n}/d \pmod{M}, \quad 0 < \left\| \left[\, d \mid \boldsymbol{n} \,\right] \right\|_2 \leq N.$$
- In general $M > 2N^2$ is still required for uniqueness, but often we still have uniqueness for smaller M.
- The problem can be solved using scalar reconstruction n times, but this requires $M > 2N^2$ even if uniqueness holds for smaller M.
- Our algorithm finds the unique solution (if one exists) which only requires $M > 2^{(c+1)/2} N^{1+1/c}$ for a small integer constant c, e.g., $c \in \{2, 3, 4, 5\}$.

Rewriting as a Lattice Problem

- Find vectors with length shorter than N in the lattice $\mathcal{L}$ generated by the rows of the matrix
$$\boldsymbol{L} = \begin{bmatrix} & & & M \\ & & \iddots & \\ & M & & \\ 1 & a_1 & \cdots & a_n \end{bmatrix} \in \mathbb{Z}^{(n+1)\times(n+1)},$$
because short vectors in such a lattice have the general form $\left[\, d \mid d\boldsymbol{a} \bmod M \,\right] = \left[\, d \mid \boldsymbol{n} \,\right]$.

LLL Lattice Basis Reduction

- The famed LLL reduction algorithm can be used to find short vectors in lattices, but is too costly to run on lattices of dimension $n + 1$ when n is large.
- Instead, we gradually reduce $\boldsymbol{L}$ by iteratively reducing truncated sublattices of $\mathcal{L}$, à la [1].
- By discarding vectors which cannot contribute to short vectors we can ensure we never have to reduce lattices of dimension more than $c + 1$.

Example

- Find a vector of size at most $N = 1000$ which gives a reconstruction of
$$\begin{bmatrix} -11431 & 5719 & -16455 \end{bmatrix} \bmod 40009.$$
- Scalar reconstruction would require $M > 2 \cdot 10^6$, but our algorithm with $c = 3$ succeeds.
- LLL-reduce the lower-left 2×2 submatrix of $\boldsymbol{L}$:
$$\begin{bmatrix} 0 & 40009 \\ 1 & -11431 \end{bmatrix} \xrightarrow{\text{LLL}} \begin{bmatrix} -7 & -1 \\ 802 & -5601 \end{bmatrix}.$$

Now, any vector which includes the last row must be longer than N, so the last row is discarded.

- Add a column and row and LLL-reduce:
$$\begin{bmatrix} 0 & 0 & 40009 \\ -7 & -1 & -40033 \end{bmatrix} \xrightarrow{\text{LLL}} \begin{bmatrix} -7 & -1 & -24 \\ -10738 & -1534 & 3193 \end{bmatrix}$$
Once again, the last row may be discarded.
- Add a column and row and LLL-reduce:
$$\begin{bmatrix} 0 & 0 & 0 & 40009 \\ -7 & -1 & -24 & 115185 \end{bmatrix} \xrightarrow{\text{LLL}} \begin{bmatrix} -231 & -33 & -792 & 250 \\ 175 & 25 & 600 & 1023 \end{bmatrix}$$
Once again, the last row may be discarded.
- We find the unique vector reconstruction is
$$\begin{bmatrix} 33/231 & 792/231 & -250/231 \end{bmatrix}.$$

Optimizations

- When $c \in O(1)$ the algorithm just demonstrated has cost $O(n^2(\log M)^3)$.
- Two primary optimizations:
  - Only store the first basis column during computations and reconstruct the rest of the basis at the end.
  - Apply the L^2 algorithm [2], a variant of LLL which works with the Gramian matrix $\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$.
- Taking these modifications into account, the algorithm has cost $O(n(\log M)^2)$.

References

[1] M. van Hoeij and A. Novocin. Gradual sub-lattice reduction and a new complexity for factoring polynomials. *LATIN 2010: Theoretical Informatics.*

[2] P. Q. Nguyen and D. Stehlé. An LLL algorithm with quadratic complexity. *SIAM Journal on Computing*, 2009.

University of Waterloo, Ontario, Canada N2L 3G1