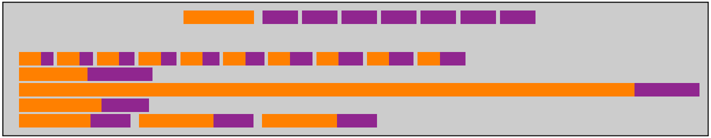
Output-sensitive algorithms for sumset and sparse polynomial multiplication



Andrew Arnold

Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada

Daniel S. Roche

Computer Science Department
United States Naval Academy
Annapolis, Maryland, USA

ISSAC 2015
Bath, UK
July 8, 2014

Our Result

We can multiply *any* polynomial
(sparse or dense)
in linear time
in the size of the input.*

*This statement is false.

Our Result

We can multiply *any* polynomial
(sparse or dense)
in linear time
in the sizes of the input **and output.***

*This statement is false.

Our Result

We can multiply *any* polynomial
(sparse or dense)
in **softly**-linear time
in the sizes of the input and output.*

Note: $\tilde{O}(\phi)$ means $O\left(\phi \log^{O(1)} \phi\right)$.

*This statement is false.

Our Result

We can multiply *any* polynomial
(sparse or dense)
in softly-linear time
in the **"structural"** sizes of the input and output.

Note: $\tilde{O}(\phi)$ means $O\left(\phi \log^{O(1)} \phi\right)$.

Dense multiplication

How to multiply?

$$65x^3 + 20x^2 + 26x + 16$$

$$\times$$

$$60x^2 + 78x - 48$$

Dense multiplication

How to multiply?

$$65x^3 + 20x^2 + 26x + 16$$

$$60x^2 + 78x - 48$$

$$=$$

$$3900x^5 + 6270x^4 + 2028x^2 - 768$$

- Direct “school” method. Quadratic complexity.

Dense multiplication

How to multiply?

$$65x^3 + 20x^2 + 26x + 16 \longrightarrow 65002000260016$$

$$\times$$

$$60x^2 + 78x - 48 \longrightarrow 6000779952$$

$$= \qquad\qquad =$$

$$3900x^5+6270x^4+2028x^2-768 \longleftarrow 390062700000202799999232$$

- Direct "school" method. Quadratic complexity.
- Indirect method, using FFT. Softly-linear complexity.

Sparse Multiplication

How to multiply?

$$65x^{31}y^{36} + 20x^{13}y^{49} + 26x^{38}y^{12} + 16x^{20}y^{25}$$

$$\times$$

$$60x^{16}y^{43} + 78x^{41}y^{6} - 48x^{23}y^{19}$$

Sparse Multiplication

How to multiply?

$$65x^{31}y^{36} + 20x^{13}y^{49} + 26x^{38}y^{12} + 16x^{20}y^{25}$$

$$60x^{16}y^{43} + 78x^{41}y^{6} - 48x^{23}y^{19}$$

- Direct “school” method. Quadratic complexity

Sparse Multiplication

How to multiply?

$$65x^{31}y^{36} + 20x^{13}y^{49} + 26x^{38}y^{12} + 16x^{20}y^{25}$$

$$60x^{16}y^{43} + 78x^{41}y^{6} - 48x^{23}y^{19}$$

- Direct “school” method. Quadratic complexity
- Geobuckets (Yan ’98)

Sparse Multiplication

How to multiply?

$$65x^{31}y^{36} + 20x^{13}y^{49} + 26x^{38}y^{12} + 16x^{20}y^{25}$$

$$60x^{16}y^{43} + 78x^{41}y^{6} - 48x^{23}y^{19}$$

- Direct “school” method. Quadratic complexity
- Geobuckets (Yan ’98)
- Heaps (Johnson ’74, Monagan & Pearce ’07. . .)

Output-Sensitive Sparse Multiplication

Quadratic-time already defeated in many cases:

- Recursive dense
- Chunky, equal spaced (R. '11)
- Blockwise dense (van der Hoeven & Lecerf '12)
- Homogeneous dense (Gastineau & Laskar '13)
- Support on a lattice (van der Hoeven, Lebreton, Schost '13)
- Support is given (van der Hoeven & Lecerf '13)

What about sparse intepolation?

Idea: Evaluate at $T \gg \#(fg)$ points, multiply, interpolate the product

What about sparse intepolation?

Idea: Evaluate at $T \gg \#(fg)$ points, multiply, interpolate the product

“Big prime” algorithms

Computation is performed modulo p, $p \gg \deg(fg)$.

But one evaluation needs $\tilde{O}(T \log \deg(fg))$ ops modulo p;
hence at least $\tilde{O}(T \log^2 \deg(fg))$ bit complexity

What about sparse intepolation?

Idea: Evaluate at $T \gg \#(fg)$ points, multiply, interpolate the product

"Big prime" algorithms

Computation is performed modulo p, $p \gg \deg(fg)$.

But one evaluation needs $\tilde{O}(T \log \deg(fg))$ ops modulo p;
hence at least $\tilde{O}(T \log^2 \deg(fg))$ bit complexity

"Small primes" algorithms

Computations performed modulo small primes p.

But all algorithms still need $\tilde{O}(T \log^2 \deg(fg))$ operations.

Observe: The trouble is in the degree!

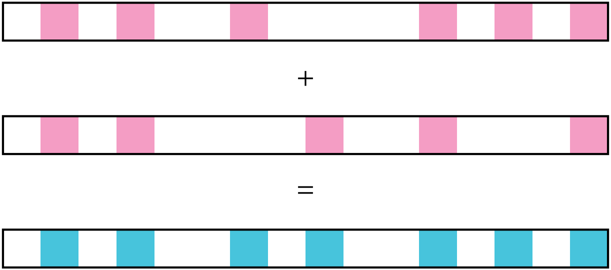
Two kinds of sparsity

Consider the following sparse addition problem:

$+$

Two kinds of sparsity

Consider the following sparse addition problem:



- Structural sparsity is 7.

Two kinds of sparsity

Consider the following sparse addition problem:

	−2		5			−5					−3		8		1

$$+$$

	6		−5					−2			3				7

$$=$$

	4					−5		−2					8		8

- Structural sparsity is 7.
- Arithmetic sparsity is 5.

What to notice

2 Building Blocks

- Dense polynomial arithmetic
- Sparse polynomial interpolation

What to notice

2 Building Blocks

- Dense polynomial arithmetic
- Sparse polynomial interpolation

2 Techniques

- Multiple reduction and relaxation
- Coefficient ratios without derivatives

What to notice

2 Building Blocks

- Dense polynomial arithmetic
- Sparse polynomial interpolation

2 Techniques

- Multiple reduction and relaxation
- Coefficient ratios without derivatives

2 Useful Subroutines

- Computing sumset
- Sparse interpolation with known support

Running Example

The Problem

$$f = 65x^{31}y^{36} + 20x^{13}y^{49} + 26x^{38}y^{12} + 16x^{20}y^{25}$$
$$g = 60x^{16}y^{43} + 78x^{41}y^{6} - 48x^{23}y^{19}$$

What is the product $h = fg$?

Running Example

The Problem

$$f = 65x^{31}y^{36} + 20x^{13}y^{49} + 26x^{38}y^{12} + 16x^{20}y^{25}$$
$$g = 60x^{16}y^{43} + 78x^{41}y^{6} - 48x^{23}y^{19}$$

What is the product $h = fg$?

Overview of approach

1. Estimate structural sparsity
2. Compute structural support
3. Compute arithmetic support (i.e., the actual exponents)
4. Compute the coefficients

Step 0: Substitutions

Given

$$f = 65x^{31}y^{36} + 20x^{13}y^{49} + 26x^{38}y^{12} + 16x^{20}y^{25}$$
$$g = 60x^{16}y^{43} + 78x^{41}y^{6} - 48x^{23}y^{19}$$

Kronecker Substitution

$$f_K = f(z, z^{100}) = 20z^{4913} + 65z^{3631} + 16z^{2520} + 26z^{1238}$$
$$g_K = g(z, z^{100}) = 60z^{4316} - 48z^{1923} + 78z^{641}$$

Note: h completely determined from $f_K g_K$.

Step 0: Substitutions

Given

$$f = 65x^{31}y^{36} + 20x^{13}y^{49} + 26x^{38}y^{12} + 16x^{20}y^{25}$$
$$g = 60x^{16}y^{43} + 78x^{41}y^{6} - 48x^{23}y^{19}$$

Kronecker Substitution

$$f_K = f(z, z^{100}) = 20z^{4913} + 65z^{3631} + 16z^{2520} + 26z^{1238}$$
$$g_K = g(z, z^{100}) = 60z^{4316} - 48z^{1923} + 78z^{641}$$

Note: h completely determined from $f_K g_K$.

Coefficient removal

$$f_S = z^{4913} + z^{3631} + z^{2520} + z^{1238}$$
$$g_S = z^{4316} + z^{1923} + z^{641}$$

Note: *structural* support of h determined from $f_S g_S$.

Step 1: Estimate structural sparsity

Given

$$f_S = z^{4913} + z^{3631} + z^{2520} + z^{1238}$$
$$g_S = z^{4316} + z^{1923} + z^{641}$$

How sparse is the product $h_S = f_S \cdot g_S$?

1. Choose primes $p = 211, \quad p' = 5$
2. Compute $\left((f_S \cdot g_S)^{\bmod p}\right)^{\bmod p'}$
   $= 2z^4 + 3z^3 + 3z^2 + 2z + 2$
3. Less than half-dense? No

Step 1: Estimate structural sparsity

Given

$f_S = z^{4913} + z^{3631} + z^{2520} + z^{1238}$
$g_S = z^{4316} + z^{1923} + z^{641}$

How sparse is the product $h_S = f_S \cdot g_S$?

1. Choose primes $p = 211, \quad p' = 11$
2. Compute $\left((f_S \cdot g_S)^{\bmod p}\right)^{\bmod p'}$
$= 3z^9 + 2z^8 + z^7 + 2z^4 + z^3 + 3z^2$
3. Less than half-dense? No

Step 1: Estimate structural sparsity

Given

$f_S = z^{4913} + z^{3631} + z^{2520} + z^{1238}$

$g_S = z^{4316} + z^{1923} + z^{641}$

How sparse is the product $h_S = f_S \cdot g_S$?

1. Choose primes $p = 211, \quad p' = 17$
2. Compute $\left((f_S \cdot g_S)^{\bmod p}\right)^{\bmod p'}$
   $= z^{16} + z^7 + z^6 + 2z^4 + 3z^3 + z^2 + z + 2$
3. Less than half-dense? Yes
   Means structural sparsity is close to 8.

First technique: Multiple Reduction and Relaxation

$$f_S = z^{4913} + z^{3631} + z^{2520} + z^{1238}$$

$$f_S^{\bmod 211} = z^{199} + z^{183} + z^{60} + z^{44}$$

$$\left(f_S^{\bmod 211}\right)^{\bmod 17} = z^{13} + z^{12} + z^{10} + z^{9}$$

What's going on?

- First reduce exponents modulo p
- Now treat that as an ordinary polynomial
- Then reduce further!
- Each reduction introduces a factor-2 in the error estimation.

First building block

How to compute $\left((f_S \cdot g_S)^{\bmod p}\right)^{\bmod p'}$?

- This polynomial never gets very sparse
- Its degree is *linear* in the actual structural sparsity

First building block

How to compute $\left((f_S \cdot g_S)^{\bmod p}\right)^{\bmod p'}$?

- This polynomial never gets very sparse
- Its degree is *linear* in the actual structural sparsity
- So we can use dense polynomial arithmetic!

Papers: (Karatsuba '58), (Toom & Cook '63), (Schönhage & Strassen '71), (Cantor & Kaltofen '91), (Fürer '07), (DKSS '08), . . .

Software: GMP, NTL, FLINT, Singular, Maple,. . .

Step 2: Compute structural support

Given

$f_S = z^{4913} + z^{3631} + z^{2520} + z^{1238}$
$g_S = z^{4316} + z^{1923} + z^{641}$

$\#(f_S \cdot g_S) \approx 8$

What are the exponents of $h_S = f_S \cdot g_S$?

- Use the same prime $p = 211$ as before.
- Compute $h_1 = \left(f_S^{\bmod p} \cdot g_S^{\bmod p}\right)^{\bmod p}$
  $= 2z^{207} + z^{191} + z^{156} + z^{140} + 2z^{84} + 3z^{68} + z^{52} + z^{12}$

Step 2: Compute structural support

Given

$f_S = z^{4913} + z^{3631} + z^{2520} + z^{1238}$
$g_S = z^{4316} + z^{1923} + z^{641}$

$\#(f_S \cdot g_S) \approx 8$

What are the exponents of $h_S = f_S \cdot g_S$?

- Use the same prime $p = 211$ as before.
- Set $\ell \gg \deg(h) = 16000$
- Compute $f_2 = \sum(e\ell + 1)z^{e \bmod p}$
  $= (4913{\cdot}16000{+}1)z^{4913 \bmod 211} + (3631{\cdot}16000{+}1)z^{3631 \bmod 211} + \cdots$
  $= 40320001z^{199} + 19808001z^{183} + 78608001z^{60} + 58096001z^{44}$
- Compute g_2 similarly.
- Compute $h_2 = (f_2 \cdot g_2)^{\bmod p} \bmod \ell^2$

$= 101152002z^{207} + 30064001z^{191} + 147664001z^{156} + 127152001z^{140} + 218752002z^{84} + 266592003z^{68} + 68352001z^{52} + 71088001z^{12}$

Step 2: Compute structural support

Given

$f_S = z^{4913} + z^{3631} + z^{2520} + z^{1238}$
$g_S = z^{4316} + z^{1923} + z^{641}$

$\#(f_S \cdot g_S) \approx 8$

What are the exponents of $h_S = f_S \cdot g_S$?

- $p = 211, \quad , \ell = 16000$
- $h1 = 2z^{207} + z^{191} + z^{156} + z^{140} + 2z^{84} + 3z^{68} + z^{52} + z^{12}$
- $h2 = 101152002z^{207} + \cdots + 68352001z^{52} + \cdots$
- Take coefficient ratios: $\dfrac{\frac{c_2}{c_1} - 1}{\ell}$
- Structural support:
  $1879, \ 3161, \ 4272, \ 4443, \ 5554, \ 6836, \ 7947, \ 9229$

Did you notice the first technique again?

Did you notice the first technique again?

$$(f_2 \cdot g_2)^{\bmod p} \bmod \ell^2$$

Multiple levels of reduction/relaxation here!

Second technique: Coefficient ratios

The polynomials f_2, g_2, h_2 have their exponents *encoded in the coefficients*.

The encoding is *additive* modulo ℓ^2:
$(a\ell + 1)(b\ell + 1) \bmod \ell^2 = (a + b)\ell + 1$

Allows recovering the *actual exponents* from the coefficients of the degree-reduced product.

Second building block

How to compute $h_2 = f_2 \cdot g_2$?

- This polynomial is *kind of* sparse.
- It has huge coefficients!

Second building block

How to compute $h_2 = f_2 \cdot g_2$?

- This polynomial is *kind of* sparse.
- It has huge coefficients!
- We can use sparse polynomial interpolation!
- **Requirement**: Linear-time in the sparsity bound, poly-logarithmic in the degree.

Papers: (Prony '95), (Blahut '79), (Ben-Or & Tiwari '88), (Kaltofen '10), (Kaltofen & Lee '03), (A., Giesbrecht, Roche '14), ...

Software: Mathemagix, Maple (maybe), ???

Step 3: Trim down to the arithmetic support

Given

$f_K = f(z, z^{100}) = 20z^{4913} + 65z^{3631} + 16z^{2520} + 26z^{1238}$
$g_K = g(z, z^{100}) = 60z^{4316} - 48z^{1923} + 78z^{641}$
$\operatorname{supp}(f_K \cdot g_K) \subseteq S =$
$\{1879,\ 3161,\ 4272,\ 4443,\ 5554,\ 6836,\ 7947,\ 9229\}$

What are the *actual* exponents of $f_K \cdot g_K$?

1. Choose $\quad p = 23, \quad q = 47 \quad$ (note $p|(q-1)$)
2. Compute $S \bmod p = \{16, 10, 17, 4, 11, 5, 12, 6\}$
3. Compute $h_{p,q} = (f_K \cdot g_K)^{\bmod p} \bmod q$
   $= 41z^{17} + 7z^{16} + 46z^{12} + 25z^6 + 31z^4$

Step 3: Trim down to the arithmetic support

Given

$f_K = f(z, z^{100}) = 20z^{4913} + 65z^{3631} + 16z^{2520} + 26z^{1238}$
$g_K = g(z, z^{100}) = 60z^{4316} - 48z^{1923} + 78z^{641}$
$\mathrm{supp}(f_K \cdot g_K) \subseteq S =$
$\{\mathbf{1879},\ 3161,\ \mathbf{4272},\ \mathbf{4443},\ 5554,\ 6836,\ \mathbf{7947},\ \mathbf{9229}\}$

What are the *actual* exponents of $f_K \cdot g_K$?

1. Choose $p = 23, \quad q = 47$ (note $p|(q-1)$)
2. Compute $S \bmod p = \{\mathbf{16}, 10, \mathbf{17}, \mathbf{4}, 11, 5, \mathbf{12}, \mathbf{6}\}$
3. Compute $h_{p,q} = (f_K \cdot g_K)^{\bmod p} \bmod q$
   $= 41z^{17} + 7z^{16} + 46z^{12} + 25z^6 + 31z^4$
4. Identify support from nonzero terms

(Of course you saw the first technique again.)

$$(f_K \cdot g_K)^{\bmod p} \bmod q$$

Twist on second building block

How to compute $(f_K \cdot g_K)^{\bmod p} \bmod q$?

- This polynomial is *kind of* sparse.
- An advantage: this time we know the support!

Twist on second building block

How to compute $(f_K \cdot g_K)^{\operatorname{mod} p} \bmod q$?

- This polynomial is *kind of* sparse.
- An advantage: this time we know the support!
- Use the coefficient-finding step of sparse interpolation!
- Because $p|(q-1)$, we can evaluate at pth roots of unity and solve a transposed Vandermonde system.

Papers: (Kaltofen & Lakshman ’89), (van der Hoeven & Lecerf ’13)

Step 4: Compute the coefficients

Given

$f_K = f(z, z^{100}) = 20z^{4913} + 65z^{3631} + 16z^{2520} + 26z^{1238}$
$g_K = g(z, z^{100}) = 60z^{4316} - 48z^{1923} + 78z^{641}$
$\mathrm{supp}(f_K \cdot g_K) = S' = \{1879,\ 4272,\ 4443,\ 7947,\ 9229\}$

What are the coefficients of $f_K \cdot g_K$?

1. Choose $p = 11, \quad q = 23$ (note $p|(q-1)$)
2. Compute $S' \bmod p = \{9, 4, 10, 5, 0\}$
3. Compute $h_{p,q} = (f_K \cdot g_K)^{\bmod p} \bmod q$
   $= 14z^{10} + 4z^9 + 13z^5 + 10z^4 + 4$
4. Group like terms for Chinese Remaindering

Step 4: Compute the coefficients

Given

$f_K = f(z, z^{100}) = 20z^{4913} + 65z^{3631} + 16z^{2520} + 26z^{1238}$
$g_K = g(z, z^{100}) = 60z^{4316} - 48z^{1923} + 78z^{641}$
$\mathrm{supp}(f_K \cdot g_K) = S' = \{1879,\ 4272,\ 4443,\ 7947,\ 9229\}$

What are the coefficients of $f_K \cdot g_K$?

1. Choose $p = 11, \quad q = 67$ (note $p|(q-1)$)
2. Compute $S' \bmod p = \{9, 4, 10, 5, 0\}$
3. Compute $h_{p,q} = (f_K \cdot g_K)^{\bmod p} \bmod q$
   $= 36z^{10} + 18z^{9} + 14z^{5} + 45z^{4} + 61$
4. Group like terms for Chinese Remaindering

Step 4: Compute the coefficients

Given

$f_K = f(z, z^{100}) = 20z^{4913} + 65z^{3631} + 16z^{2520} + 26z^{1238}$
$g_K = g(z, z^{100}) = 60z^{4316} - 48z^{1923} + 78z^{641}$
$\mathrm{supp}(f_K \cdot g_K) = S' = \{1879,\ 4272,\ 4443,\ 7947,\ 9229\}$

What are the coefficients of $f_K \cdot g_K$?

1. Choose $p = 11, \quad q = 89$ (note $p|(q-1)$)
2. Compute $S' \bmod p = \{9, 4, 10, 5, 0\}$
3. Compute $h_{p,q} = (f_K \cdot g_K)^{\bmod p} \bmod q$
   $= 33z^{10} + 70z^{9} + 73z^{5} + 86z^{4} + 43$
4. Group like terms for Chinese Remaindering

Step 4: Compute the coefficients

Given

$f_K = f(z, z^{100}) = 20z^{4913} + 65z^{3631} + 16z^{2520} + 26z^{1238}$
$g_K = g(z, z^{100}) = 60z^{4316} - 48z^{1923} + 78z^{641}$
$\mathrm{supp}(f_K \cdot g_K) = S' = \{1879,\ 4272,\ 4443,\ 7947,\ 9229\}$

What are the coefficients of $f_K \cdot g_K$?

1. Choose $p = 11, \quad q =$ 23, 67, 89
2. Compute $S' \bmod p = \{9, 4, 10, 5, 0\}$
3. Compute $h_{p,q} = (f_K \cdot g_K)^{\bmod p} \bmod q$
5. Apply CRT and undo the Kronecker map:
$h = 3900x^{47}y^{79} + 1200x^{29}y^{92} + 5070x^{72}y^{42} + 2028x^{79}y^{18} - 768x^{43}y^{44}$

Complexity Overview

Non-toy example

1000 terms, 8 variables, 64-bit coefficients, 32-bit exponents

Structural sparsity 10000, arithmetic sparsity 1000

Complexity Overview

Non-toy example

1000 terms, 8 variables, 64-bit coefficients, 32-bit exponents

Structural sparsity 10000, arithmetic sparsity 1000

Steps of the algorithm

1. Estimate structural sparsity

Complexity Overview

Non-toy example

1000 terms, 8 variables, $\boxed{\text{64-bit coefficients}}$, $\boxed{\text{32-bit exponents}}$

Structural sparsity 10000, arithmetic sparsity 1000

Steps of the algorithm

1. Estimate structural sparsity

2. Compute structural support

Complexity Overview

Non-toy example

1000 terms, 8 variables, 64-bit coefficients, 32-bit exponents

Structural sparsity 10000, arithmetic sparsity 1000

Steps of the algorithm

1. Estimate structural sparsity

2. Compute structural support

3. Trim to arithmetic support

Complexity Overview

Non-toy example

1000 terms, 8 variables, 64-bit coefficients, 32-bit exponents

Structural sparsity 10000, arithmetic sparsity 1000

Steps of the algorithm

1. Estimate structural sparsity
2. Compute structural support
3. Trim to arithmetic support
4. Compute coefficients

Summary

C = |largest coefficient|	S = structural sparsity
D = max degree	T = arithmetic sparsity

Theorem

Given $f, g \in \mathbb{Z}[x]$, our Monte Carlo algorithm computes $h = fg$ with $\tilde{O}(S \log C + T \log D)$ bit complexity.

Extends to softly-linear time algorithms for

- Multivariate polynomials
- Laurent polynomials
- Modular rings, finite fields, exact rationals

Two useful subroutines

Sumset

Given sets $A, B \subset \mathbb{Z}$, compute
$S = \{a + b | a \in A, b \in B\}$.

Sparse multiplication with known support

Given $f, g \in \mathbb{Z}[x]$ and the exponents of $f \cdot g$,
compute the coefficients of $f \cdot g$.

We provide softly linear-time solutions to both problems.
(They correspond to steps 1-2 and steps 3-4, resp.)

What's left to do? (Lots!)

- Make an efficient (parallel) implementation
- Decrease randomness (Las Vegas? Deterministic?)
- Make cost dependent on *arithmetic* sparsity
- Start worrying about the log factors
- Apply improvements to other problems
  (division, interpolation, . . .)