Polynomial Factorization: Recent advances, and challenges

Pranjal Dutta
School of Computing, NUS

10^{th} July, 2023

Algebraic Complexity Theory Workshop @ ICALP 2023

Table of Contents

1. Multivariate Polynomial Factoring: Background

2. Classical Factoring results

3. Recent advances

4. Conclusion

Multivariate Polynomial Factoring: Background

- Polynomial factoring is encountered in high school!

- Polynomial factoring is encountered in high school!
- Polynomials can be factored in polynomial time.

Factoring Univariates

- Polynomial factoring is encountered in high school!
- Polynomials can be factored in polynomial time.
- Factor $f(x) \in \mathbb{Q}[x]$ using LLL algorithm in deterministic polynomial time.

Factoring Univariates

- Polynomial factoring is encountered in high school!
- Polynomials can be factored in polynomial time.
- Factor $f(x) \in \mathbb{Q}[x]$ using LLL algorithm in deterministic polynomial time.
- Factor $f(x) \in \mathbb{F}_q[x]$ using Berlekamp's algorithm.

Complexity of Multivariate Factoring

- The polynomial ring $\mathbb{F}[x_1, \ldots, x_n]$ is UFD (Unique Factorization Domain).

- The polynomial ring $\mathbb{F}[x_1, \ldots, x_n]$ is UFD (Unique Factorization Domain).

Factorization of a polynomial

Let f be a polynomial of degree d that has 'size' s in some class $\mathcal{C}$. Let $f(\boldsymbol{x}) = \prod_{i=1}^{m} f_i^{e_i}$, where the polynomials f_i are its irreducible factors over $\mathbb{F}$. Output each f_i, in some related class $\mathcal{D}$.

Complexity of Multivariate Factoring

- The polynomial ring $\mathbb{F}[x_1, \ldots, x_n]$ is UFD (Unique Factorization Domain).

Factorization of a polynomial

Let f be a polynomial of degree d that has 'size' s in some class $\mathcal{C}$. Let $f(\boldsymbol{x}) = \prod_{i=1}^{m} f_i^{e_i}$, where the polynomials f_i are its irreducible factors over $\mathbb{F}$. Output each f_i, in some related class $\mathcal{D}$.

- Factor size bound: Do all its factors have $\mathrm{poly}(s, d)$ size in $\mathcal{D}$?

Complexity of Multivariate Factoring

- The polynomial ring $\mathbb{F}[x_1, \ldots, x_n]$ is UFD (Unique Factorization Domain).

Factorization of a polynomial

Let f be a polynomial of degree d that has 'size' s in some class $\mathcal{C}$. Let $f(\boldsymbol{x}) = \prod_{i=1}^{m} f_i^{e_i}$, where the polynomials f_i are its irreducible factors over $\mathbb{F}$. Output each f_i, in some related class $\mathcal{D}$.

- Factor size bound: Do all its factors have $\mathrm{poly}(s, d)$ size in $\mathcal{D}$?
- Efficient algorithm: Design an 'efficient' algorithm to compute the irreducible factors.

Complexity of Multivariate Factoring

- The polynomial ring $\mathbb{F}[x_1, \ldots, x_n]$ is UFD (Unique Factorization Domain).

Factorization of a polynomial

Let f be a polynomial of degree d that has 'size' s in some class $\mathcal{C}$. Let $f(\boldsymbol{x}) = \prod_{i=1}^{m} f_i^{e_i}$, where the polynomials f_i are its irreducible factors over $\mathbb{F}$. Output each f_i, in some related class $\mathcal{D}$.

- Factor size bound: Do all its factors have $\mathrm{poly}(s, d)$ size in $\mathcal{D}$?
- Efficient algorithm: Design an 'efficient' algorithm to compute the irreducible factors.
- Factor of a polynomial can be more "complex" than the polynomial itself.

- The polynomial ring $\mathbb{F}[x_1, \ldots, x_n]$ is UFD (Unique Factorization Domain).

FACTORIZATION OF A POLYNOMIAL

Let f be a polynomial of degree d that has 'size' s in some class $\mathcal{C}$. Let $f(\boldsymbol{x}) = \prod_{i=1}^{m} f_i^{e_i}$, where the polynomials f_i are its irreducible factors over $\mathbb{F}$. Output each f_i, in some related class $\mathcal{D}$.

- FACTOR SIZE BOUND: Do all its factors have $\text{poly}(s, d)$ size in $\mathcal{D}$?
- EFFICIENT ALGORITHM: Design an 'efficient' algorithm to compute the irreducible factors.
- Factor of a polynomial can be more "complex" than the polynomial itself.
- For example, $\prod_{i=1}^{n}(x_i^d - 1)$ has sparsity 2^n. But its factor $\prod_{i=1}^{n}(1 + x_i + \ldots + x_i^{d-1})$ has sparsity $d^n = (2^n)^{\log d}$.

- The polynomial ring $\mathbb{F}[x_1, \ldots, x_n]$ is UFD (Unique Factorization Domain).

Factorization of a polynomial

Let f be a polynomial of degree d that has 'size' s in some class $\mathcal{C}$. Let $f(\boldsymbol{x}) = \prod_{i=1}^{m} f_i^{e_i}$, where the polynomials f_i are its irreducible factors over $\mathbb{F}$. Output each f_i, in some related class $\mathcal{D}$.

- Factor size bound: Do all its factors have $\mathrm{poly}(s, d)$ size in $\mathcal{D}$?
- Efficient algorithm: Design an 'efficient' algorithm to compute the irreducible factors.
- Factor of a polynomial can be more "complex" than the polynomial itself.
- For example, $\prod_{i=1}^{n}(x_i^d - 1)$ has sparsity 2^n. But its factor $\prod_{i=1}^{n}(1 + x_i + \ldots + x_i^{d-1})$ has sparsity $d^n = (2^n)^{\log d}$.
- When $\mathcal{C} = \mathcal{D}$, then $\mathcal{C}$ is *closed under factoring*.

Multivariate to Univariate Factoring

- Multivariate factoring $f(\boldsymbol{x}) = g(\boldsymbol{x}) \cdot h(\boldsymbol{x})$can be reduced to univariate factoring via *Kronecker* substitution:

- Multivariate factoring $f(\boldsymbol{x}) = g(\boldsymbol{x}) \cdot h(\boldsymbol{x})$can be reduced to univariate factoring via *Kronecker* substitution:
  - Let the degree of each variable in f is $\leq d$. Apply Kronecker substitution $\phi : x_i \mapsto z^{(d+1)^{i-1}}$.

- Multivariate factoring $f(\boldsymbol{x}) = g(\boldsymbol{x}) \cdot h(\boldsymbol{x})$can be reduced to univariate factoring via *Kronecker* substitution:
  - Let the degree of each variable in f is $\leq d$. Apply Kronecker substitution $\phi : x_i \mapsto z^{(d+1)^{i-1}}$.
  - Each monomial in f uniquely maps to a monomial in $\phi(f)$.

- Multivariate factoring $f(\boldsymbol{x}) = g(\boldsymbol{x}) \cdot h(\boldsymbol{x})$can be reduced to univariate factoring via *Kronecker* substitution:
  - Let the degree of each variable in f is $\leq d$. Apply Kronecker substitution $\phi : x_i \mapsto z^{(d+1)^{i-1}}$.
  - Each monomial in f uniquely maps to a monomial in $\phi(f)$.
  - Factorize $\phi(f)$ into univariate irreducible factors.

- Multivariate factoring $f(\boldsymbol{x}) = g(\boldsymbol{x}) \cdot h(\boldsymbol{x})$can be reduced to univariate factoring via *Kronecker* substitution:
  - Let the degree of each variable in f is $\leq d$. Apply Kronecker substitution $\phi : x_i \mapsto z^{(d+1)^{i-1}}$.
  - Each monomial in f uniquely maps to a monomial in $\phi(f)$.
  - Factorize $\phi(f)$ into univariate irreducible factors.
  - Though g is irreducible, $\phi(g)$ *may not* be irreducible.

- Multivariate factoring $f(\boldsymbol{x}) = g(\boldsymbol{x}) \cdot h(\boldsymbol{x})$can be reduced to univariate factoring via *Kronecker* substitution:
  - Let the degree of each variable in f is $\leq d$. Apply Kronecker substitution $\phi : x_i \mapsto z^{(d+1)^{i-1}}$.
  - Each monomial in f uniquely maps to a monomial in $\phi(f)$.
  - Factorize $\phi(f)$ into univariate irreducible factors.
  - Though g is irreducible, $\phi(g)$ *may not* be irreducible.
  - Product of a subset of the factors of $\phi(f)$ would correspond to $\phi(g)$.

- Multivariate factoring $f(\boldsymbol{x}) = g(\boldsymbol{x}) \cdot h(\boldsymbol{x})$can be reduced to univariate factoring via *Kronecker* substitution:
  - Let the degree of each variable in f is $\leq d$. Apply Kronecker substitution $\phi : x_i \mapsto z^{(d+1)^{i-1}}$.
  - Each monomial in f uniquely maps to a monomial in $\phi(f)$.
  - Factorize $\phi(f)$ into univariate irreducible factors.
  - Though g is irreducible, $\phi(g)$ *may not* be irreducible.
  - Product of a subset of the factors of $\phi(f)$ would correspond to $\phi(g)$.
  - Try *all subsets*. Apply inverse Kronecker and check if the polynomial divides f. [Check by Resultant].

- Multivariate factoring $f(\boldsymbol{x}) = g(\boldsymbol{x}) \cdot h(\boldsymbol{x})$can be reduced to univariate factoring via *Kronecker* substitution:
    - Let the degree of each variable in f is $\leq d$. Apply Kronecker substitution $\phi : x_i \mapsto z^{(d+1)^{i-1}}$.
    - Each monomial in f uniquely maps to a monomial in $\phi(f)$.
    - Factorize $\phi(f)$ into univariate irreducible factors.
    - Though g is irreducible, $\phi(g)$ *may not* be irreducible.
    - Product of a subset of the factors of $\phi(f)$ would correspond to $\phi(g)$.
    - Try *all subsets*. Apply inverse Kronecker and check if the polynomial divides f. [Check by Resultant].
    - Time complexity: Exponential in degree in worst-case (even for bivariates).

Classical Factoring results

Efficient multivariate factorization

- Let us fix algebraic circuit as the model and $\mathrm{size}_{\mathrm{Circuit}}$ denotes the circuit size.

EFFICIENT CIRCUIT FACTORING [Kaltofen 1986]

$g \mid f \implies \mathrm{size}_{\mathrm{Circuit}}(g) \leq \mathrm{poly}(\mathrm{size}_{\mathrm{Circuit}}(f), \deg(f)).$

- Let us fix algebraic circuit as the model and $\mathrm{size}_{\mathrm{Circuit}}$ denotes the circuit size.

Efficient Circuit Factoring [Kaltofen 1986]

$g \mid f \implies \mathrm{size}_{\mathrm{Circuit}}(g) \le \mathsf{poly}(\mathrm{size}_{\mathrm{Circuit}}(f), \mathsf{deg}(f))$. Moreover, there is a randomized $\mathsf{poly}(\mathrm{size}_{\mathrm{Circuit}}(f), \mathsf{deg}(f))$-time algorithm that outputs every irreducible factor.

- Let us fix algebraic circuit as the model and $\mathrm{size}_{\mathrm{Circuit}}$ denotes the circuit size.

Efficient Circuit Factoring [Kaltofen 1986]

$g \mid f \implies \mathrm{size}_{\mathrm{Circuit}}(g) \leq \mathrm{poly}(\mathrm{size}_{\mathrm{Circuit}}(f), \deg(f))$. Moreover, there is a randomized $\mathrm{poly}(\mathrm{size}_{\mathrm{Circuit}}(f), \deg(f))$-time algorithm that outputs every irreducible factor.

- VP is *closed under factoring*.

- Let us fix algebraic circuit as the model and $\text{size}_{\text{Circuit}}$ denotes the circuit size.

Efficient Circuit Factoring [Kaltofen 1986]

$g \mid f \implies \text{size}_{\text{Circuit}}(g) \leq \mathsf{poly}(\text{size}_{\text{Circuit}}(f), \mathsf{deg}(f))$. Moreover, there is a randomized $\mathsf{poly}(\text{size}_{\text{Circuit}}(f), \mathsf{deg}(f))$-time algorithm that outputs every irreducible factor.

- VP is *closed under factoring*.

- Tools: Newton iteration/ Hensel lifting, Linear System Solving.

- Let us fix algebraic circuit as the model and $\mathsf{size}_{\text{Circuit}}$ denotes the circuit size.

Efficient Circuit Factoring [Kaltofen 1986]

$g \mid f \implies \mathsf{size}_{\text{Circuit}}(g) \leq \mathsf{poly}(\mathsf{size}_{\text{Circuit}}(f), \deg(f))$. Moreover, there is a randomized $\mathsf{poly}(\mathsf{size}_{\text{Circuit}}(f), \deg(f))$-time algorithm that outputs every irreducible factor.

- VP is *closed under factoring*.

- Tools: Newton iteration/ Hensel lifting, Linear System Solving.

- Goal: Extend Kaltofen's result for formulas, constant depth circuits, algebraic branching programs (ABPs), high-degree regime etc.

- Let us fix algebraic circuit as the model and $\mathrm{size}_{\mathrm{Circuit}}$ denotes the circuit size.

Efficient Circuit Factoring [Kaltofen 1986]

$g \mid f \implies \mathrm{size}_{\mathrm{Circuit}}(g) \le \mathrm{poly}(\mathrm{size}_{\mathrm{Circuit}}(f), \deg(f))$. Moreover, there is a randomized $\mathrm{poly}(\mathrm{size}_{\mathrm{Circuit}}(f), \deg(f))$-time algorithm that outputs every irreducible factor.

- VP is *closed under factoring*.
- Tools: Newton iteration/ Hensel lifting, Linear System Solving.
- Goal: Extend Kaltofen's result for formulas, constant depth circuits, algebraic branching programs (ABPs), high-degree regime etc.
- What happens if we only care about just the query/blackbox complexity?

- Let us fix algebraic circuit as the model and $\mathsf{size}_{\text{Circuit}}$ denotes the circuit size.

Efficient Circuit Factoring [Kaltofen 1986]

$g \mid f \implies \mathsf{size}_{\text{Circuit}}(g) \leq \mathsf{poly}(\mathsf{size}_{\text{Circuit}}(f), \mathsf{deg}(f))$. Moreover, there is a randomized $\mathsf{poly}(\mathsf{size}_{\text{Circuit}}(f), \mathsf{deg}(f))$-time algorithm that outputs every irreducible factor.

- VP is *closed under factoring*.
- Tools: Newton iteration/ Hensel lifting, Linear System Solving.
- Goal: Extend Kaltofen's result for formulas, constant depth circuits, algebraic branching programs (ABPs), high-degree regime etc.
- What happens if we only care about just the query/blackbox complexity?
- Application: *Hardness versus randomness* in algebraic complexity [KI'03, Agrawal'05]; *possible separation* of complexity classes.

- [Kabanets-Impagliazzo 2003]: Exponential lower bound for Permanent (i.e. VNP exponentially far from VP) $\implies$ Quasi-poly blackbox *deterministic PIT* for circuits.

- [Kabanets-Impagliazzo 2003]: Exponential lower bound for Permanent (i.e. VNP exponentially far from VP) $\implies$ Quasi-poly blackbox *deterministic PIT* for circuits.

- [Possible separation]: If $\mathcal{C}$ is *not closed under factoring*, then $\mathcal{C} \neq \mathsf{VP}$.

Applications

- [Kabanets-Impagliazzo 2003]: Exponential lower bound for Permanent (i.e. VNP exponentially far from VP) $\implies$ Quasi-poly blackbox *deterministic PIT* for circuits.
- [Possible separation]: If $\mathcal{C}$ is *not closed under factoring*, then $\mathcal{C} \neq \mathsf{VP}$.
- Can we show that $\mathsf{VP} \neq \mathsf{VNP}$, VBP, VF via factoring?!

- [Kabanets-Impagliazzo 2003]: Exponential lower bound for Permanent (i.e. VNP exponentially far from VP) $\implies$ Quasi-poly blackbox *deterministic PIT* for circuits.

- [Possible separation]: If $\mathcal{C}$ is *not closed under factoring*, then $\mathcal{C} \neq \mathsf{VP}$.

- Can we show that $\mathsf{VP} \neq \mathsf{VNP}$, VBP, VF via factoring?!

- [Hardness of multiples]: If factors of $\mathcal{C}$ are in class $\mathcal{D}$, and f is hard for $\mathcal{D}$, all its nonzero multiples of f are *hard* for $\mathcal{C}$!

- [Kabanets-Impagliazzo 2003]: Exponential lower bound for Permanent (i.e. VNP exponentially far from VP) $\implies$ Quasi-poly blackbox *deterministic PIT* for circuits.

- [Possible separation]: If $\mathcal{C}$ is *not closed under factoring*, then $\mathcal{C} \neq \mathsf{VP}$.

- Can we show that $\mathsf{VP} \neq \mathsf{VNP}$, VBP, VF via factoring?!

- [Hardness of multiples]: If factors of $\mathcal{C}$ are in class $\mathcal{D}$, and f is hard for $\mathcal{D}$, all its nonzero multiples of f are *hard* for $\mathcal{C}$!
  - Take $\mathcal{C} = \mathcal{D} = \mathsf{VP}$. If $\mathsf{VP} \neq \mathsf{VNP}$, *any polynomial-degree multiple* of perm_n is *also hard* for VP.

- [Kabanets-Impagliazzo 2003]: Exponential lower bound for Permanent (i.e. VNP exponentially far from VP) $\implies$ Quasi-poly blackbox *deterministic PIT* for circuits.
- [Possible separation]: If $\mathcal{C}$ is *not closed under factoring*, then $\mathcal{C} \neq \mathsf{VP}$.
- Can we show that $\mathsf{VP} \neq \mathsf{VNP}$, VBP, VF via factoring?!
- [Hardness of multiples]: If factors of $\mathcal{C}$ are in class $\mathcal{D}$, and f is hard for $\mathcal{D}$, all its nonzero multiples of f are *hard* for $\mathcal{C}$!
  - Take $\mathcal{C} = \mathcal{D} = \mathsf{VP}$. If $\mathsf{VP} \neq \mathsf{VNP}$, *any polynomial-degree multiple* of perm_n is *also hard* for VP.
- [KSS'14]: Derandomizing circuit-factoring *is equivalent to* derandomizing circuit-PIT.

Blackbox factoring

- [Kaltofen-Trager 1991]: Given a black box computing a multivariate polynomial f, black boxes of the irreducible factors of f can be computed in randomized polynomial time.

- [Kaltofen-Trager 1991]: Given a black box computing a multivariate polynomial f, black boxes of the irreducible factors of f can be computed in randomized polynomial time.
    - Dimension reduction: Randomly project to bivariates.

- [Kaltofen-Trager 1991]: Given a black box computing a multivariate polynomial f, black boxes of the irreducible factors of f can be computed in randomized polynomial time.
  - Dimension reduction: Randomly project to bivariates.
  - This works due to an effective version of **Hilbert's irreducibility theorem**.

- [Kaltofen-Trager 1991]: Given a black box computing a multivariate polynomial f, black boxes of the irreducible factors of f can be computed in randomized polynomial time.
  - Dimension reduction: Randomly project to bivariates.
  - This works due to an effective version of **Hilbert's irreducibility theorem**.
  - If $f(x, z_1, \ldots, z_n)$ is irreducible, then $f(x, \beta_1 + \alpha_1 y, \ldots, \beta_n + \alpha_n y)$ is irreducible with high probability if β_i, α_i picked at random.

- [Kaltofen-Trager 1991]: Given a black box computing a multivariate polynomial f, black boxes of the irreducible factors of f can be computed in randomized polynomial time.
  - Dimension reduction: Randomly project to bivariates.
  - This works due to an effective version of **Hilbert's irreducibility theorem**.
  - If $f(x, z_1, \ldots, z_n)$ is irreducible, then $f(x, \beta_1 + \alpha_1 y, \ldots, \beta_n + \alpha_n y)$ is irreducible with high probability if β_i, α_i picked at random.
  - Currently, derandomization of this theorem for sparse polynomials reduces to ABP PIT.

Recent advances

Some more closure results

Some more closure results

- [Oliveira'15]: The class $C =$ is *closed under factoring*, where $C =$ constant depth circuits with constant individual degree.

Some more closure results

- [Oliveira'15]: The class C = is *closed under factoring*, where C = constant depth circuits with constant individual degree.
  - $\deg_{x_i} f(\boldsymbol{x}) \le r$, for each $i \in [n]$, $\text{size}_{\text{Circuit}}(f) \le s$, and depth Δ, and if $g \mid f$, then $\text{size}_{\text{Circuit}}(g) \le \text{poly}(r^r, s)$, and depth $\Delta + 5$.

Some more closure results

- [Oliveira'15]: The class $\mathcal{C}$ = is *closed under factoring*, where $\mathcal{C}$ = constant depth circuits with constant individual degree.
  - $\deg_{x_i} f(\boldsymbol{x}) \leq r$, for each $i \in [n]$, $\text{size}_{\text{Circuit}}(f) \leq s$, and depth Δ, and if $g \mid f$, then $\text{size}_{\text{Circuit}}(g) \leq \text{poly}(r^r, s)$, and depth $\Delta + 5$.

- [Dutta'18]: $f \in \text{VP}_{\text{constant}} \implies \text{size}_{\text{Circuit}}(f) \leq \text{poly}(n)$, and $\deg_{x_i}(f) \leq r$, for some constant r. Then, $\text{VP}_{\text{constant}}$ is *closed under factoring*.

Some more closure results

- [Oliveira'15]: The class C = is *closed under factoring*, where C = constant depth circuits with constant individual degree.
  - $\deg_{x_i} f(\boldsymbol{x}) \leq r$, for each $i \in [n]$, $\text{size}_{\text{Circuit}}(f) \leq s$, and depth Δ, and if $g \mid f$, then $\text{size}_{\text{Circuit}}(g) \leq \mathsf{poly}(r^r, s)$, and depth $\Delta + 5$.

- [Dutta'18]: $f \in \mathsf{VP}_{\text{constant}} \implies \text{size}_{\text{Circuit}}(f) \leq \mathsf{poly}(n)$, and $\deg_{x_i}(f) \leq r$, for some constant r. Then, $\mathsf{VP}_{\text{constant}}$ is *closed under factoring*. Same for $\text{VBP}_{\text{constant}}, \mathsf{VNP}_{\text{constant}}$.

- [Oliveira'15]: The class $C =$ is *closed under factoring*, where $C =$ constant depth circuits with constant individual degree.
  - $\deg_{x_i} f(\boldsymbol{x}) \leq r$, for each $i \in [n]$, $\text{size}_{\text{Circuit}}(f) \leq s$, and depth Δ, and if $g \mid f$, then $\text{size}_{\text{Circuit}}(g) \leq \text{poly}(r^r, s)$, and depth $\Delta + 5$.

- [Dutta'18]: $f \in \mathsf{VP}_{\text{constant}} \implies \text{size}_{\text{Circuit}}(f) \leq \text{poly}(n)$, and $\deg_{x_i}(f) \leq r$, for some constant r. Then, $\mathsf{VP}_{\text{constant}}$ is *closed under factoring*. Same for $\text{VBP}_{\text{constant}}, \mathsf{VNP}_{\text{constant}}$.

- [Dutta-Saxena-Sinhababu'18]: $g \mid f$, and $\deg(f) = d$, then $\text{size}_{\text{ABP}}(g) \leq \text{poly}(\text{size}_{\text{ABP}}(f), d^{O(\log d)})$.
  - Same for VF, VNP.

Some more closure results

- [Oliveira'15]: The class C = is *closed under factoring*, where C = constant depth circuits with constant individual degree.
  - $\deg_{x_i} f(\boldsymbol{x}) \leq r$, for each $i \in [n]$, $\text{size}_{\text{Circuit}}(f) \leq s$, and depth Δ, and if $g \mid f$, then $\text{size}_{\text{Circuit}}(g) \leq \text{poly}(r^r, s)$, and depth $\Delta + 5$.

- [Dutta'18]: $f \in \text{VP}_{\text{constant}} \implies \text{size}_{\text{Circuit}}(f) \leq \text{poly}(n)$, and $\deg_{x_i}(f) \leq r$, for some constant r. Then, $\text{VP}_{\text{constant}}$ is *closed under factoring*. Same for $\text{VBP}_{\text{constant}}, \text{VNP}_{\text{constant}}$.

- [Dutta-Saxena-Sinhababu'18]: $g \mid f$, and $\deg(f) = d$, then $\text{size}_{\text{ABP}}(g) \leq \text{poly}(\text{size}_{\text{ABP}}(f), d^{O(\log d)})$.
  - Same for VF, VNP.
  - So, quasipolynomial-VBP (similarly for formula and VNP) are closed under factoring.

Some more closure results

- [Oliveira'15]: The class C = is *closed under factoring*, where C = constant depth circuits with constant individual degree.
  - $\deg_{x_i} f(\boldsymbol{x}) \leq r$, for each $i \in [n]$, $\text{size}_{\text{Circuit}}(f) \leq s$, and depth Δ, and if $g \mid f$, then $\text{size}_{\text{Circuit}}(g) \leq \mathsf{poly}(r^r, s)$, and depth $\Delta + 5$.

- [Dutta'18]: $f \in \mathsf{VP}_{\text{constant}} \implies \text{size}_{\text{Circuit}}(f) \leq \mathsf{poly}(n)$, and $\deg_{x_i}(f) \leq r$, for some constant r. Then, $\mathsf{VP}_{\text{constant}}$ is *closed under factoring*. Same for $\text{VBP}_{\text{constant}}$, $\mathsf{VNP}_{\text{constant}}$.

- [Dutta-Saxena-Sinhababu'18]: $g \mid f$, and $\deg(f) = d$, then $\text{size}_{\text{ABP}}(g) \leq \mathsf{poly}(\text{size}_{\text{ABP}}(f), d^{O(\log d)})$.
  - Same for VF, VNP.
  - So, quasipolynomial-VBP (similarly for formula and VNP) are closed under factoring.

- [Chou-Kumar-Solomon'18]: VNP is closed under factoring.

- [Oliveira'15]: The class $C =$ is *closed under factoring*, where $C =$ constant depth circuits with constant individual degree.
  - $\deg_{x_i} f(\boldsymbol{x}) \le r$, for each $i \in [n]$, $\text{size}_{\text{Circuit}}(f) \le s$, and depth Δ, and if $g \mid f$, then $\text{size}_{\text{Circuit}}(g) \le \text{poly}(r^r, s)$, and depth $\Delta + 5$.

- [Dutta'18]: $f \in \text{VP}_{\text{constant}} \implies \text{size}_{\text{Circuit}}(f) \le \text{poly}(n)$, and $\deg_{x_i}(f) \le r$, for some constant r. Then, $\text{VP}_{\text{constant}}$ is *closed under factoring*. Same for $\text{VBP}_{\text{constant}}$, $\text{VNP}_{\text{constant}}$.

- [Dutta-Saxena-Sinhababu'18]: $g \mid f$, and $\deg(f) = d$, then $\text{size}_{\text{ABP}}(g) \le \text{poly}(\text{size}_{\text{ABP}}(f), d^{O(\log d)})$.
  - Same for VF, VNP.
  - So, quasipolynomial-VBP (similarly for formula and VNP) are closed under factoring.

- [Chou-Kumar-Solomon'18]: VNP is closed under factoring.

- [Sinhababu-Thierauf'21]: VBP is closed under factoring.

Factor complexity for border classes

- One can ask what happens when $C = \overline{\mathsf{VP}}$.

- One can ask what happens when $\mathcal{C} = \overline{\mathsf{VP}}$. In particular, if $g \mid f$, and $f \in \overline{\mathsf{VP}}$, then $g \in \overline{\mathsf{VP}}$?

- One can ask what happens when $C = \overline{\mathsf{VP}}$. In particular, if $g \mid f$, and $f \in \overline{\mathsf{VP}}$, then $g \in \overline{\mathsf{VP}}$?
- [Bürgisser 03]: $\overline{\mathsf{VP}}$ is *closed under factoring*.

- One can ask what happens when $C = \overline{\mathsf{VP}}$. In particular, if $g \mid f$, and $f \in \overline{\mathsf{VP}}$, then $g \in \overline{\mathsf{VP}}$?
- [Bürgisser 03]: $\overline{\mathsf{VP}}$ is *closed under factoring*.
- [Dutta-Saxena-Sinhababu'18]: Quasipoly-$\overline{\mathsf{VBP}}$, Quasipoly-$\overline{\mathsf{VNP}}$, Quasipoly-$\overline{\mathsf{VF}}$ are *closed under factoring*.

- One can ask what happens when $C = \overline{\mathsf{VP}}$. In particular, if $g \mid f$, and $f \in \overline{\mathsf{VP}}$, then $g \in \overline{\mathsf{VP}}$?
- [Bürgisser 03]: $\overline{\mathsf{VP}}$ is *closed under factoring*.
- [Dutta-Saxena-Sinhababu'18]: Quasipoly-$\overline{\mathsf{VBP}}$, Quasipoly-$\overline{\mathsf{VNP}}$, Quasipoly-$\overline{\mathsf{VF}}$ are *closed under factoring*.
- $\overline{\mathsf{VNP}}$ is *closed under factoring* (implicit in [Chou-Kumar-Solomon'18]).

Factor complexity for border classes

- One can ask what happens when $C = \overline{\mathsf{VP}}$. In particular, if $g \mid f$, and $f \in \overline{\mathsf{VP}}$, then $g \in \overline{\mathsf{VP}}$?
- [Bürgisser 03]: $\overline{\mathsf{VP}}$ is *closed under factoring*.
- [Dutta-Saxena-Sinhababu'18]: Quasipoly-$\overline{\mathsf{VBP}}$, Quasipoly-$\overline{\mathsf{VNP}}$, Quasipoly-$\overline{\mathsf{VF}}$ are *closed under factoring*.
- $\overline{\mathsf{VNP}}$ is *closed under factoring* (implicit in [Chou-Kumar-Solomon'18]).
- $\overline{\mathsf{VBP}}$ is *closed under factoring* (implicit in [Sinhababu-Thierauf'21]).

High degree regime

High degree regime

- [Kaltofen'87] If $f = g^e$, then $\mathrm{size}_{\mathrm{Circuit}}(g) \leq \mathrm{poly}(\mathrm{size}_{\mathrm{Circuit}}(f), \deg(\mathbf{g}))$.

High degree regime

- [Kaltofen'87] If $f = g^e$, then $\mathrm{size}_{\mathrm{Circuit}}(g) \leq \mathrm{poly}(\mathrm{size}_{\mathrm{Circuit}}(f), \deg(\mathbf{g}))$.
  - e can be as large as 2^s, where $s = \mathrm{size}_{\mathrm{Circuit}}(f)$!

High degree regime

- [Kaltofen'87] If $f = g^e$, then $\mathrm{size}_{\mathrm{Circuit}}(g) \leq \mathsf{poly}(\mathrm{size}_{\mathrm{Circuit}}(f), \mathsf{deg}(\mathbf{g}))$.
  - e can be as large as 2^s, where $s = \mathrm{size}_{\mathrm{Circuit}}(f)$!
  - First result which depends on $\mathsf{deg}(g)$ instead of $\mathsf{deg}(f)$!

High degree regime

- [Kaltofen'87] If $f = g^e$, then $\mathrm{size}_{\mathrm{Circuit}}(g) \leq \mathsf{poly}(\mathrm{size}_{\mathrm{Circuit}}(f), \mathsf{deg}(\mathbf{g}))$.
  - e can be as large as 2^s, where $s = \mathrm{size}_{\mathrm{Circuit}}(f)$!
  - First result which depends on $\mathsf{deg}(g)$ instead of $\mathsf{deg}(f)$!
- [Factor Conjecture, Bürgisser 03]: If $g \mid f$, then $\mathrm{size}_{\mathrm{Circuit}}(g) \leq \mathsf{poly}(\mathrm{size}_{\mathrm{Circuit}}(f), \mathsf{deg}(g))$.

High degree regime

- [Kaltofen'87] If $f = g^e$, then $\mathrm{size}_{\text{Circuit}}(g) \leq \mathsf{poly}(\mathrm{size}_{\text{Circuit}}(f), \mathsf{deg}(\mathbf{g}))$.
  - e can be as large as 2^s, where $s = \mathrm{size}_{\text{Circuit}}(f)$!
  - First result which depends on $\mathsf{deg}(g)$ instead of $\mathsf{deg}(f)$!
- [Factor Conjecture, Bürgisser 03]: If $g \mid f$, then $\mathrm{size}_{\text{Circuit}}(g) \leq \mathsf{poly}(\mathrm{size}_{\text{Circuit}}(f), \mathsf{deg}(g))$.
- [Bürgisser 03:] Factor conjecture is *true*, when one replaces $\mathrm{size}_{\text{Circuit}}$ by $\overline{\mathrm{size}}_{\text{Circuit}}$!

- [Kaltofen'87] If $f = g^e$, then $\mathsf{size}_{\text{Circuit}}(g) \leq \mathsf{poly}(\mathsf{size}_{\text{Circuit}}(f), \mathsf{deg}(\mathbf{g}))$.
  - e can be as large as 2^s, where $s = \mathsf{size}_{\text{Circuit}}(f)$!
  - First result which depends on $\mathsf{deg}(g)$ instead of $\mathsf{deg}(f)$!
- [Factor Conjecture, Bürgisser 03]: If $g \mid f$, then $\mathsf{size}_{\text{Circuit}}(g) \leq \mathsf{poly}(\mathsf{size}_{\text{Circuit}}(f), \mathsf{deg}(g))$.
- [Bürgisser 03:] Factor conjecture is *true*, when one replaces $\mathsf{size}_{\text{Circuit}}$ by $\overline{\mathsf{size}}_{\text{Circuit}}$!
- Can we extend [Kaltofen'87] to $f = g_1^{e_1} g_2^{e_2}$, where both $\mathsf{deg}(g_i)$ are polynomially bounded?

- [Kaltofen'87] If $f = g^e$, then $\mathrm{size}_{\mathrm{Circuit}}(g) \leq \mathsf{poly}(\mathrm{size}_{\mathrm{Circuit}}(f), \mathsf{deg}(\mathbf{g}))$.
  - e can be as large as 2^s, where $s = \mathrm{size}_{\mathrm{Circuit}}(f)$!
  - First result which depends on $\mathsf{deg}(g)$ instead of $\mathsf{deg}(f)$!
- [Factor Conjecture, Bürgisser 03]: If $g \mid f$, then $\mathrm{size}_{\mathrm{Circuit}}(g) \leq \mathsf{poly}(\mathrm{size}_{\mathrm{Circuit}}(f), \mathsf{deg}(g))$.
- [Bürgisser 03:] Factor conjecture is *true*, when one replaces $\mathrm{size}_{\mathrm{Circuit}}$ by $\overline{\mathrm{size}}_{\mathrm{Circuit}}$!
- Can we extend [Kaltofen'87] to $f = g_1^{e_1} g_2^{e_2}$, where both $\mathsf{deg}(g_i)$ are polynomially bounded?

Improved Kaltofen [Dutta-Saxena-Sinhababu'18]:

Let $\mathrm{rad}(f)$ denotes the *square-free part* of f, i.e. $f = \prod g_i^{e_i}$, then $\mathrm{rad}(f) = \prod_i g_i$. If $g \mid f$, then $\mathrm{size}_{\mathrm{Circuit}}(g) \leq \mathsf{poly}(\mathrm{size}_{\mathrm{Circuit}}(f), \mathsf{deg}(\mathrm{rad}(f)))$.

- [Oliveira 2016, Dutta-Saxena-Sinhababu'18]: Factoring $\leq$ root approximation in power series.

- [Oliveira 2016, Dutta-Saxena-Sinhababu'18]: Factoring $\leq$ root approximation in power series.
- $p(\boldsymbol{x}, y)$ has factor $y - q(\boldsymbol{x}) \iff p(\boldsymbol{x}, q(\boldsymbol{x})) = 0$.

- [Oliveira 2016, Dutta-Saxena-Sinhababu'18]: Factoring $\leq$ root approximation in power series.
- $p(\boldsymbol{x}, y)$ has factor $y - q(\boldsymbol{x}) \iff p(\boldsymbol{x}, q(\boldsymbol{x})) = 0$.
- Approximate root via Newton iteration

$$y_{t+1} = y_t - \frac{p(\boldsymbol{x}, y_t)}{p'(\boldsymbol{x}, y_t)}$$

- [Oliveira 2016, Dutta-Saxena-Sinhababu'18]: Factoring $\leq$ root approximation in power series.
- $p(\boldsymbol{x}, y)$ has factor $y - q(\boldsymbol{x}) \iff p(\boldsymbol{x}, q(\boldsymbol{x})) = 0$.
- Approximate root via Newton iteration

$$y_{t+1} = y_t - \frac{p(\boldsymbol{x}, y_t)}{p'(\boldsymbol{x}, y_t)}$$

- $\log d$ iterations, since precision doubles everytime!

- [Oliveira 2016, Dutta-Saxena-Sinhababu'18]: Factoring $\leq$ root approximation in power series.
- $p(\boldsymbol{x}, y)$ has factor $y - q(\boldsymbol{x}) \iff p(\boldsymbol{x}, q(\boldsymbol{x})) = 0$.
- Approximate root via Newton iteration

$$y_{t+1} = y_t - \frac{p(\boldsymbol{x}, y_t)}{p'(\boldsymbol{x}, y_t)}$$

- $\log d$ iterations, since precision doubles everytime!
- A random shift $\phi : x_i \mapsto \alpha_i y + x_i + \beta_i$, makes

$$\phi(f(\boldsymbol{x})) = \prod_i (y - q_i(\boldsymbol{x})) ,$$

where q_i are power series.

- [Oliveira 2016, Dutta-Saxena-Sinhababu'18]: Factoring $\leq$ root approximation in power series.
- $p(\boldsymbol{x}, y)$ has factor $y - q(\boldsymbol{x}) \iff p(\boldsymbol{x}, q(\boldsymbol{x})) = 0$.
- Approximate root via Newton iteration

$$y_{t+1} = y_t - \frac{p(\boldsymbol{x}, y_t)}{p'(\boldsymbol{x}, y_t)}$$

- $\log d$ iterations, since precision doubles everytime!
- A random shift $\phi : x_i \mapsto \alpha_i y + x_i + \beta_i$, makes

$$\phi(f(\boldsymbol{x})) = \prod_i (y - q_i(\boldsymbol{x})) \,,$$

  where q_i are power series.
- $\mathbb{F}[[x_1, \ldots, x_n]]$ is UFD!

Related results

- [Bhargava-Saraf-Volkovich 20]: If $\mathsf{sp}(f) \leq s$, with individual degrees bounded by r, and $g \mid f$, then $\mathsf{sp}(g) \leq s^{O(r^2 \log n)}$.

- [Bhargava-Saraf-Volkovich 20]: If $\mathsf{sp}(f) \leq s$, with individual degrees bounded by r, and $g \mid f$, then $\mathsf{sp}(g) \leq s^{O(r^2 \log n)}$. This lead to an $s^{\mathsf{poly}(r) \log n}$-time algorithm for factoring sparse polynomials.

- [Bhargava-Saraf-Volkovich 20]: If $\mathsf{sp}(f) \leq s$, with individual degrees bounded by r, and $g \mid f$, then $\mathsf{sp}(g) \leq s^{O(r^2 \log n)}$. This lead to an $s^{\mathrm{poly}(r) \log n}$-time algorithm for factoring sparse polynomials.
- [Koiran-Ressyare'18]: Randomized polynomial-time algorithm to test if $f(x_1, \ldots, x_n)$ is of the form $f(x) = \ell_1(x)^{\alpha_1} \cdots \ell_n(x)^{\alpha_n}$, and if yes, outputs the linear factors.

- [Bhargava-Saraf-Volkovich 20]: If $\mathsf{sp}(f) \leq s$, with individual degrees bounded by r, and $g \mid f$, then $\mathsf{sp}(g) \leq s^{O(r^2 \log n)}$. This lead to an $s^{\mathsf{poly}(r) \log n}$-time algorithm for factoring sparse polynomials.
- [Koiran-Ressyare'18]: Randomized polynomial-time algorithm to test if $f(x_1, \ldots, x_n)$ is of the form $f(x) = \ell_1(x)^{\alpha_1} \cdots \ell_n(x)^{\alpha_n}$, and if yes, outputs the linear factors.
- [Dutta-Sinhababu-Thierauf, 202X]: If $f = \prod g_i^{e_i}$, where $\mathsf{deg}(g_i) \leq r$, and $\text{size}_{\text{Circuit}}(f) = s$. Then there is a *determinstic* $\mathsf{poly}(s^r)$-time algorithm that outputs g_i.

Conclusion

Open questions

Open questions

- Given an n-variate degree d polynomial of sparsity $\leq s$, test if it is irreducible in deterministic $\mathsf{poly}(n, s, d)$ time.

- Given an n-variate degree d polynomial of sparsity $\leq s$, test if it is irreducible in deterministic $\mathsf{poly}(n, s, d)$ time.
  - Challenge: Currently, it requires PIT for symbolic Determinants.

Open questions

- Given an n-variate degree d polynomial of sparsity $\leq s$, test if it is irreducible in deterministic $\mathrm{poly}(n, s, d)$ time.
  - Challenge: Currently, it requires PIT for symbolic Determinants.
- Given two n-variate degree d polynomial of sparsity $\leq s$, test if they are coprime in deterministic $\mathrm{poly}(n, s, d)$ time.

Open questions

- Given an n-variate degree d polynomial of sparsity $\leq s$, test if it is irreducible in deterministic $\mathsf{poly}(n, s, d)$ time.
  - Challenge: Currently, it requires PIT for symbolic Determinants.
- Given two n-variate degree d polynomial of sparsity $\leq s$, test if they are coprime in deterministic $\mathsf{poly}(n, s, d)$ time.
  - Challenge: The resultant of two sparse polynomials may not be sparse.

Open questions

- Given an n-variate degree d polynomial of sparsity $\leq s$, test if it is irreducible in deterministic $\mathrm{poly}(n, s, d)$ time.
  - Challenge: Currently, it requires PIT for symbolic Determinants.
- Given two n-variate degree d polynomial of sparsity $\leq s$, test if they are coprime in deterministic $\mathrm{poly}(n, s, d)$ time.
  - Challenge: The resultant of two sparse polynomials may not be sparse.
- Show VF is closed under factoring, or come up with candidate counter example!

Open questions

- Given an n-variate degree d polynomial of sparsity $\leq s$, test if it is irreducible in deterministic $\mathrm{poly}(n, s, d)$ time.
  - Challenge: Currently, it requires PIT for symbolic Determinants.
- Given two n-variate degree d polynomial of sparsity $\leq s$, test if they are coprime in deterministic $\mathrm{poly}(n, s, d)$ time.
  - Challenge: The resultant of two sparse polynomials may not be sparse.
- Show VF is closed under factoring, or come up with candidate counter example!
  - Challenge: Determinant *does not* have small arithmetic formulas!

Thank you! Questions?