Polynomial semantics of probabilistic circuits

Oliver Broadrick, Honghua Zhang, and Guy Van den Broeck

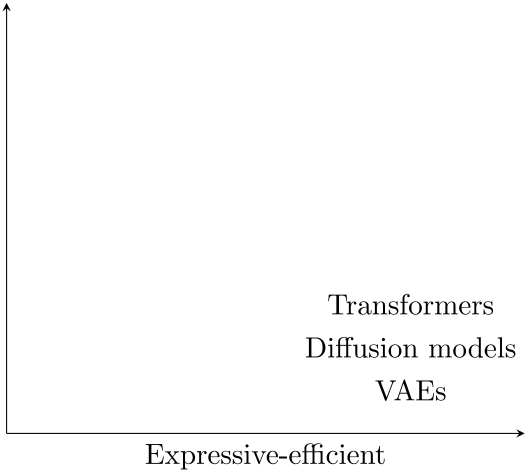
University of California, Los Angeles

Probabilistic Models

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

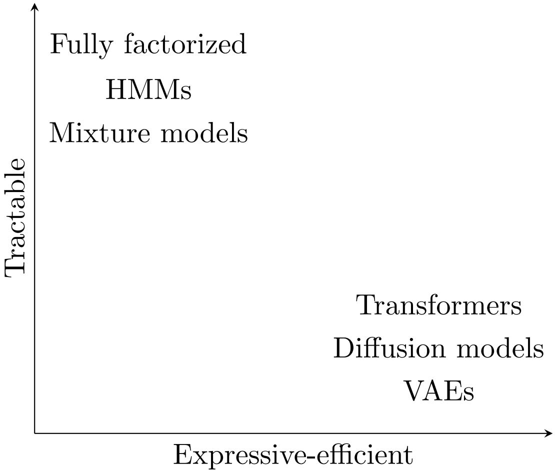
Probabilistic Models

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4



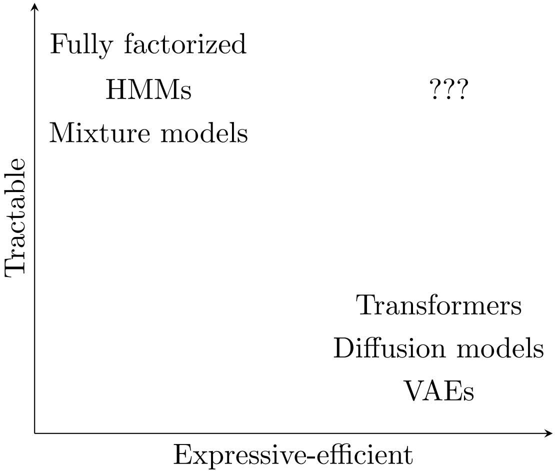
Probabilistic Models

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4



Probabilistic Models

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4



Marginal Inference

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

$$\begin{aligned}\Pr[X_1 = 1] &= \Pr[X_1 = 1, X_2 = 0] + \Pr[X_1 = 1, X_2 = 1] \\ &= 0.3 + 0.4 \\ &= 0.7\end{aligned}$$

Marginal Inference

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

$$\begin{aligned}\Pr[X_1 = 1] &= \Pr[X_1 = 1, X_2 = 0] + \Pr[X_1 = 1, X_2 = 1] \\ &= 0.3 + 0.4 \\ &= 0.7\end{aligned}$$

Goal: Find maximally expressive-efficient models that support marginal inference in time polynomial in the model size.

Approaches

Bayesian Networks (of bounded treewidth)
Determinantal Point Processes
Characteristic Circuits
Multi-Linear Representations
Probabilistic Generating Circuits
Sum-Product Networks
...

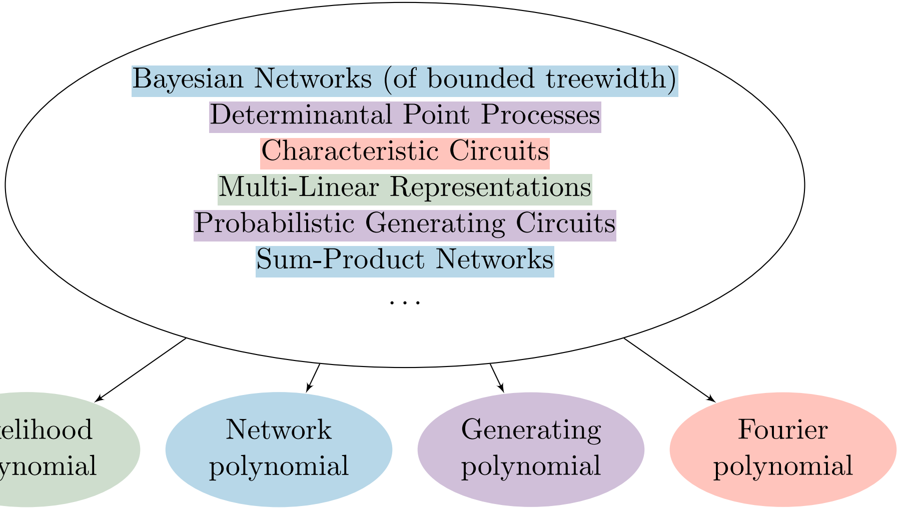
Approaches

Bayesian Networks (of bounded treewidth)
Determinantal Point Processes
Characteristic Circuits
Multi-Linear Representations
Probabilistic Generating Circuits
Sum-Product Networks
...

↓

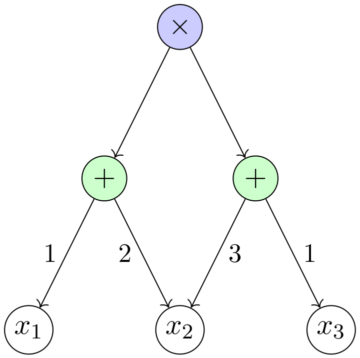
Polynomials!

Approaches



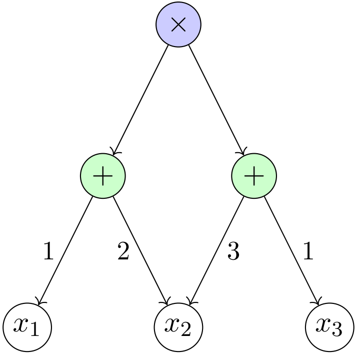
Bayesian Networks (of bounded treewidth)
Determinantal Point Processes
Characteristic Circuits
Multi-Linear Representations
Probabilistic Generating Circuits
Sum-Product Networks
...
Likelihood polynomial
Network polynomial
Generating polynomial
Fourier polynomial

Circuits represent polynomials succinctly



$$3x_1x_2 + x_1x_3 + 6x_2^2 + 2x_2x_3$$

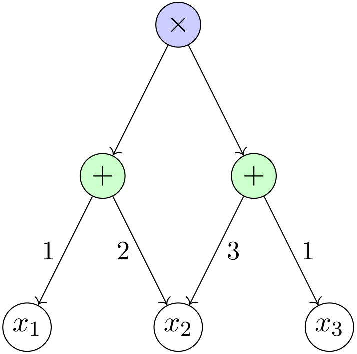
Circuits represent polynomials succinctly



$$3x_1x_2 + x_1x_3 + 6x_2^2 + 2x_2x_3$$

Circuits are *fully expressive*

Circuits represent polynomials succinctly

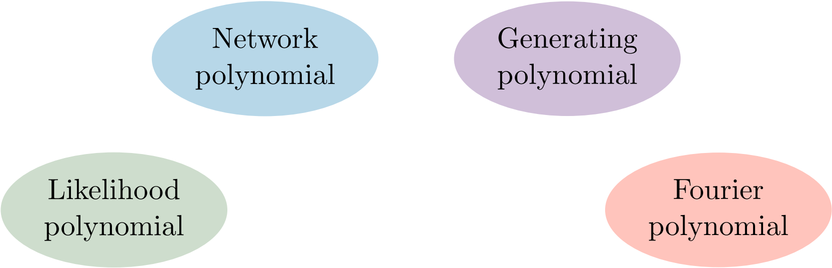


$$3x_1x_2 + x_1x_3 + 6x_2^2 + 2x_2x_3$$

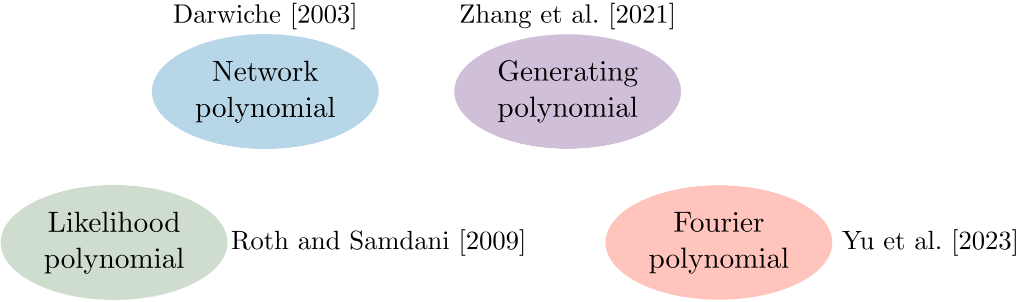
Circuits are *fully expressive*

They can also be *expressive-efficient*

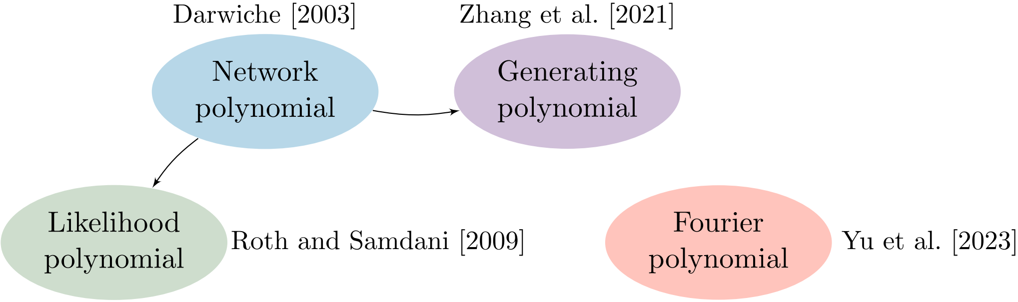
Polynomial Semantics



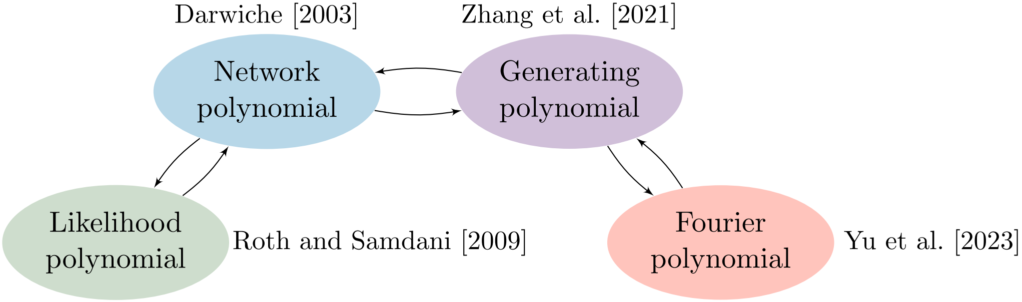
Polynomial Semantics



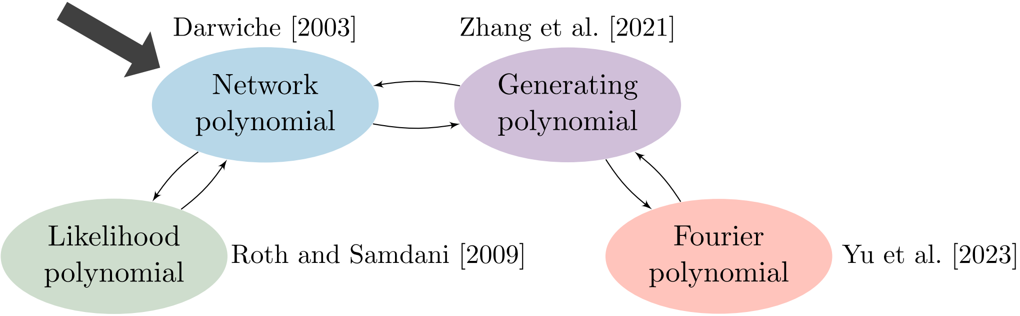
Polynomial Semantics



Polynomial Semantics



Polynomial Semantics



Network polynomial

$$p(x_1, x_2, \bar{x}_1, \bar{x}_2) = .1\bar{x}_1\bar{x}_2 + .2\bar{x}_1 x_2 + .3 x_1 \bar{x}_2 + .4 x_1 x_2$$

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

Network polynomial

$$p(x_1, x_2, \bar{x}_1, \bar{x}_2) = .1\bar{x}_1\bar{x}_2 + .2\bar{x}_1 x_2 + .3 x_1 \bar{x}_2 + .4 x_1 x_2$$

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

Network polynomial

$$p(x_1, x_2, \bar{x}_1, \bar{x}_2) = .1\bar{x}_1\bar{x}_2 + .2\bar{x}_1 x_2 + .3 x_1 \bar{x}_2 + .4 x_1 x_2$$

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

$\Pr[X_1 = 1]$

Network polynomial

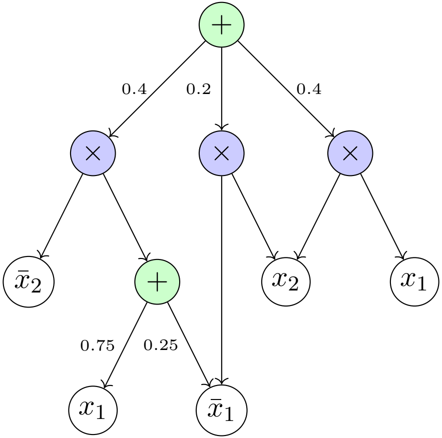
$$p(x_1, x_2, \bar{x}_1, \bar{x}_2) = .1\bar{x}_1\bar{x}_2 + .2\bar{x}_1 x_2 + .3 x_1 \bar{x}_2 + .4 x_1 x_2$$

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

$$\begin{aligned}\Pr[X_1 = 1] &= p(1, 1, 0, 1) \\ &= .1(0)(1) + .2(0)(1) + .3(1)(1) + .4(1)(1) \\ &= 0 + 0 + .3 + .4 \\ &= .7\end{aligned}$$

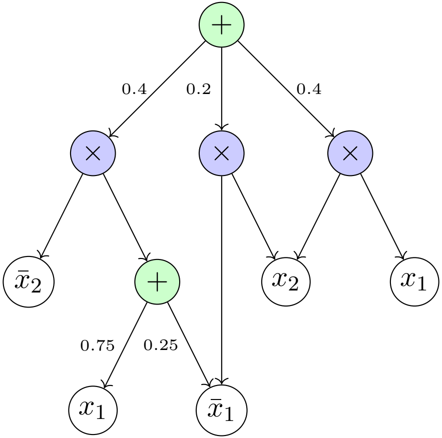
Network polynomial

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4



Network polynomial

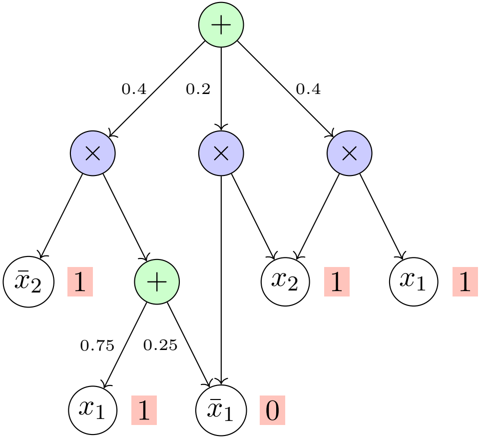
X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4



$\Pr[X_1 = 1]$?

Network polynomial

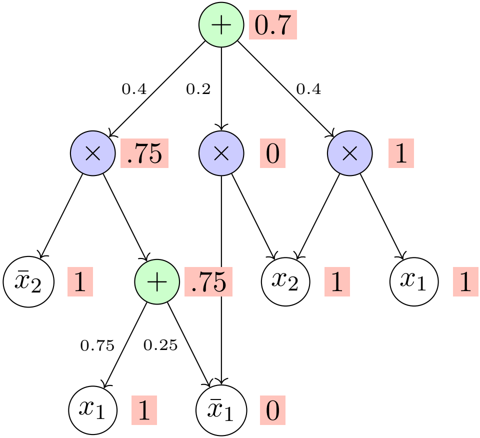
X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4



$\Pr[X_1 = 1]$?

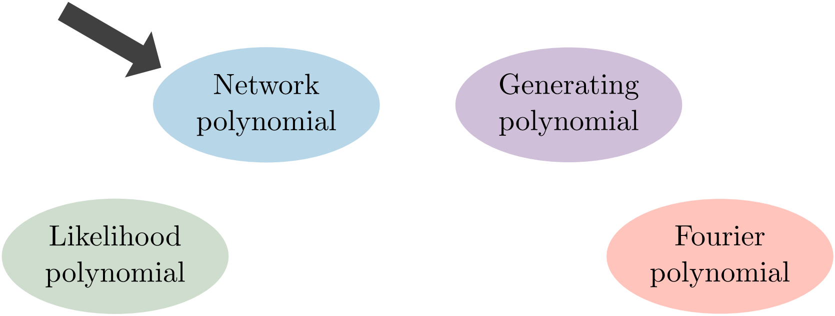
Network polynomial

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

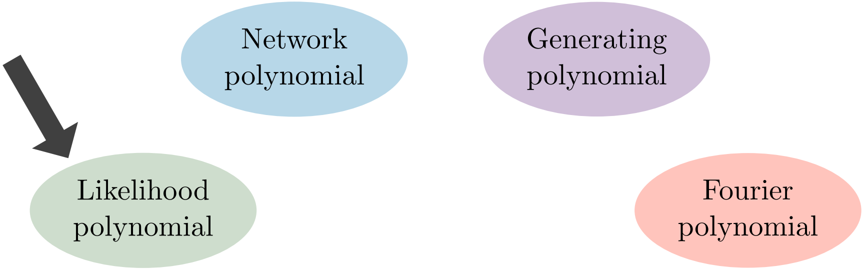


$\Pr[X_1 = 1]$?

Progress Update



Progress Update



Likelihood polynomial

$$p(x_1, x_2) = .2x_1 + .1x_2 + .1$$

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

Likelihood polynomial

$$p(x_1, x_2) = .2x_1 + .1x_2 + .1$$

Marginal inference?

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

Likelihood polynomial

$$p(x_1, x_2) = .2x_1 + .1x_2 + .1$$

Marginal inference?
Relation to network polynomial?

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

Likelihood polynomial

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

$$p(x_1, x_2) = .2x_1 + .1x_2 + .1$$

Marginal inference?
Relation to network polynomial?

Transform network to likelihood:

$$p(x, \bar{x}) = .1\bar{x}_1\bar{x}_2 + .2\bar{x}_1x_2 + .3x_1\bar{x}_2 + .4x_1x_2$$

- Replace $\bar{x}_i$ with $1 - x_i$

Likelihood polynomial

Transform likelihood to network:

$$p(x_1, x_2) = .2x_1 + .1x_2 + .1$$

Likelihood polynomial

Transform likelihood to network:

$$p(x_1, x_2) = .2x_1 + .1x_2 + .1$$

$$(x_1 + \bar{x}_1)(x_2 + \bar{x}_2)\left(.2\frac{x_1}{x_1 + \bar{x}_1} + .1\frac{x_2}{x_2 + \bar{x}_2} + .1\right)$$

Likelihood polynomial

Transform likelihood to network:

$$p(x_1, x_2) = .2x_1 + .1x_2 + .1$$

$$(x_1 + \bar{x}_1)(x_2 + \bar{x}_2)\left(.2\frac{x_1}{x_1 + \bar{x}_1} + .1\frac{x_2}{x_2 + \bar{x}_2} + .1\right)$$

$$= .2x_1(x_2 + \bar{x}_2) + .1x_2(x_1 + \bar{x}_1) + .1(x_1 + \bar{x}_1)(x_2 + \bar{x}_2)$$

Likelihood polynomial

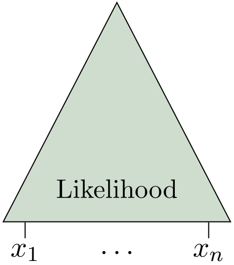
Transform likelihood to network:

$$p(x_1, x_2) = .2x_1 + .1x_2 + .1$$

$$(x_1 + \bar{x}_1)(x_2 + \bar{x}_2)\left(.2\frac{x_1}{x_1 + \bar{x}_1} + .1\frac{x_2}{x_2 + \bar{x}_2} + .1\right)$$
$$= .2x_1(x_2 + \bar{x}_2) + .1x_2(x_1 + \bar{x}_1) + .1(x_1 + \bar{x}_1)(x_2 + \bar{x}_2)$$
$$= p(x_1, x_2, \bar{x}_1, \bar{x}_2)$$

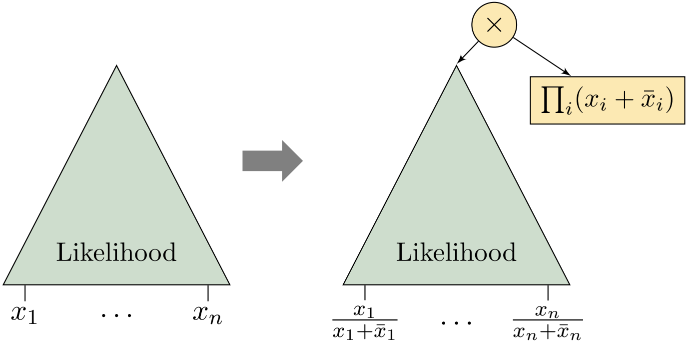
Likelihood polynomial

Transform likelihood to network:



Likelihood
polynomial

Transform likelihood to network:

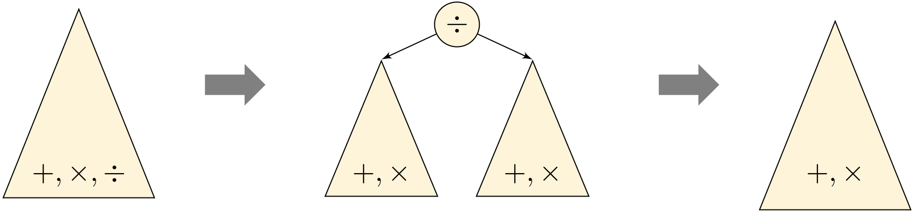


Removing Divisions

Theorem (Strassen [1973]). *You can remove divisions in polynomial time!*

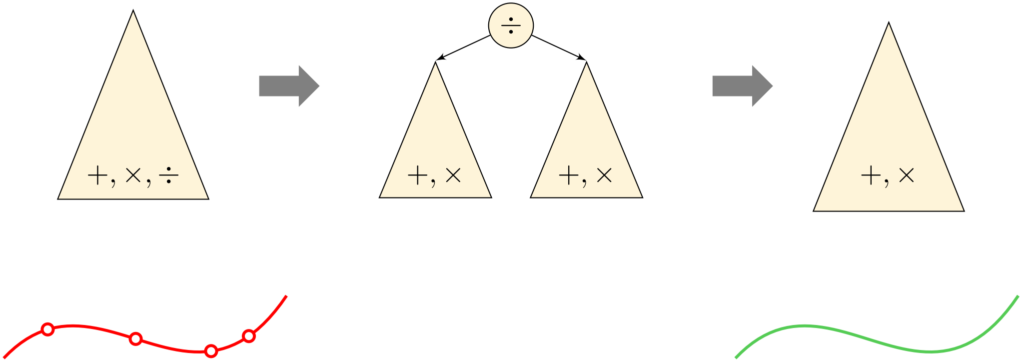
Removing Divisions

Theorem (Strassen [1973]). *You can remove divisions in polynomial time!*



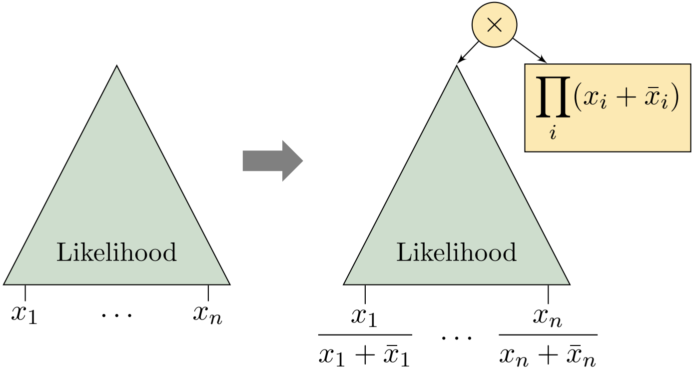
Removing Divisions

Theorem (Strassen [1973]). *You can remove divisions in polynomial time!*



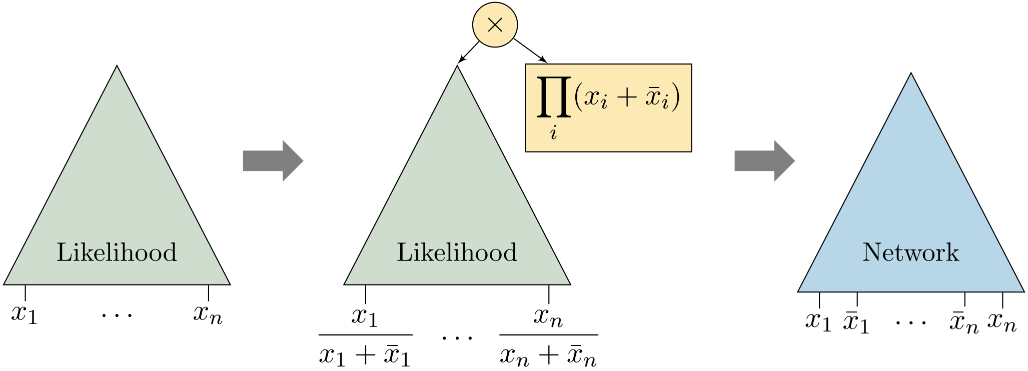
Likelihood polynomial

Transform likelihood to network:

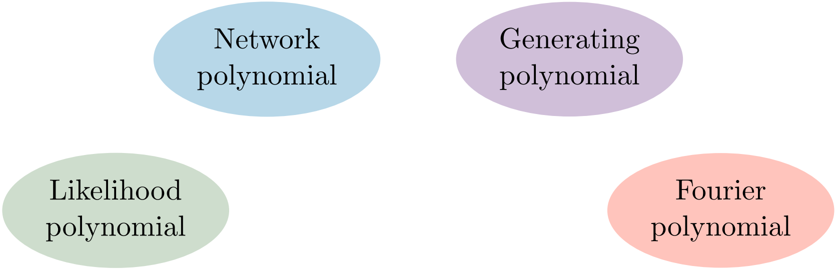


Likelihood polynomial

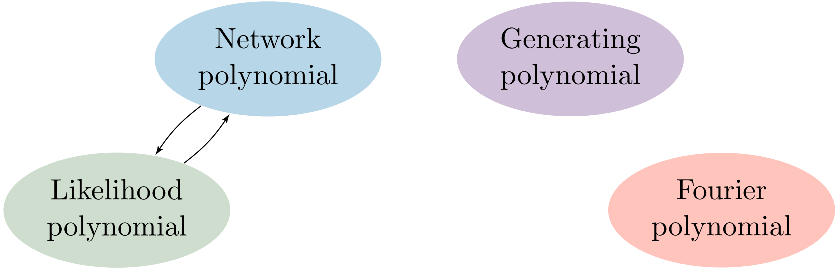
Transform likelihood to network:



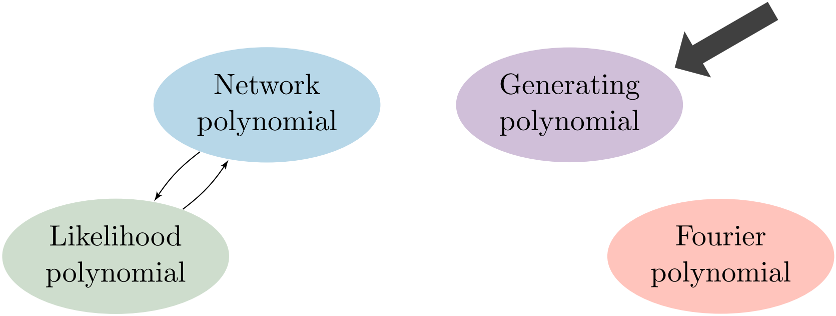
Progress Update



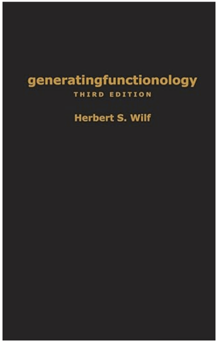
Progress Update



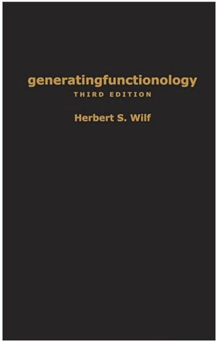
Progress Update



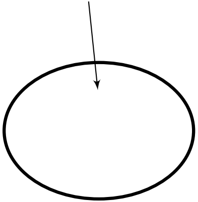
Generating polynomial



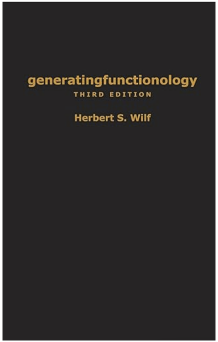
Generating polynomial



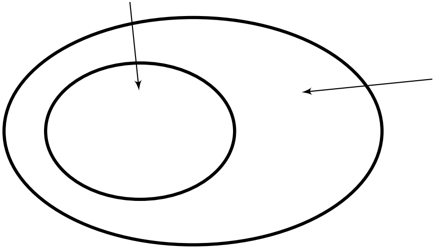
Monotone, decomposable circuits computing network polynomials (SPNs, PCs)



Generating polynomial

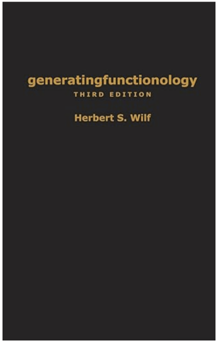


Monotone, decomposable circuits computing network polynomials (SPNs, PCs)

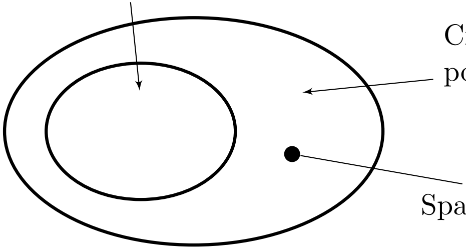


Circuits computing generating polynomials

Generating polynomial



Monotone, decomposable circuits computing network polynomials (SPNs, PCs)



Circuits computing generating polynomials

Spanning tree distribution[a]

[a] Martens and Medabalimi [2015], Zhang et al. [2021]

Generating polynomial

$$g(x) = .1 + .2x_2 + .3x_1 + .4x_1x_2$$

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

Generating polynomial

$$g(x) = .1 + .2x_2 + .3x_1 + .4x_1x_2$$

Marginal inference: ✓ [Zhang et al., 2021]

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

Generating polynomial

$$g(x) = .1 + .2x_2 + .3x_1 + .4x_1x_2$$

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

Marginal inference: ✓ [Zhang et al., 2021]
Relation to network polynomial?

Generating polynomial

$$g(x) = .1 + .2x_2 + .3x_1 + .4x_1x_2$$

Marginal inference: ✓ [Zhang et al., 2021]
Relation to network polynomial?

X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4

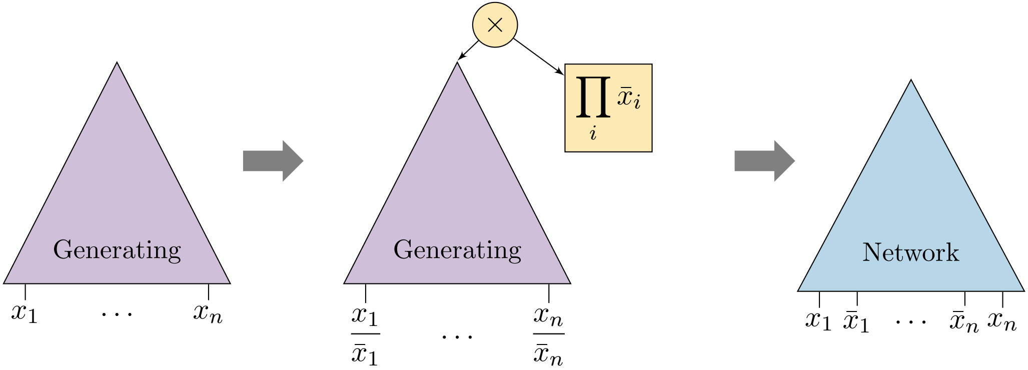
Transform network to generating:

$$p(x_1, x_2, \bar{x}_1, \bar{x}_2) = .1\bar{x}_1\bar{x}_2 + .2\bar{x}_1x_2 + .3x_1\bar{x}_2 + .4x_1x_2$$

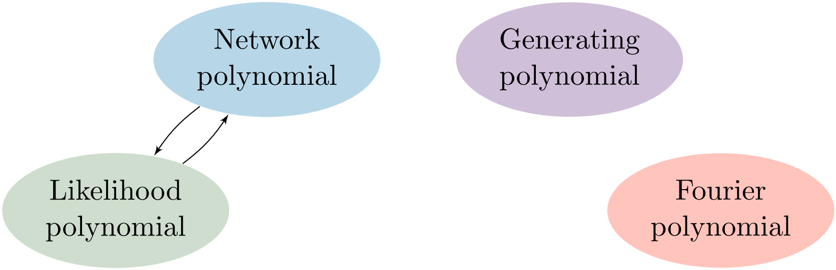
- Replace $\bar{x}_i$ with 1

Generating polynomial

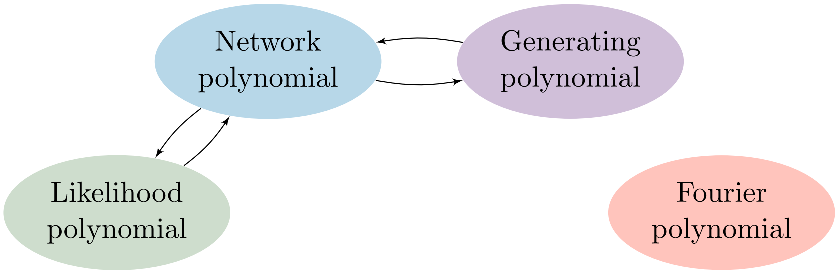
Transform generating to network:



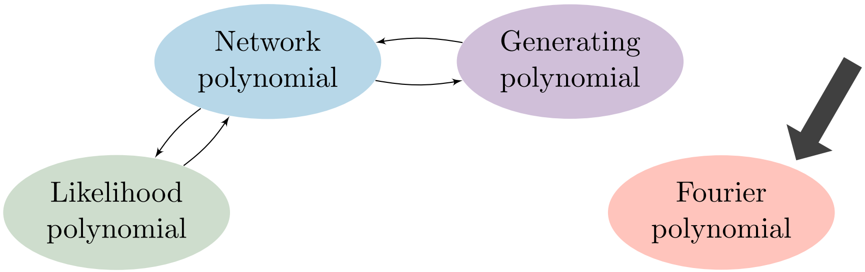
Progress Update



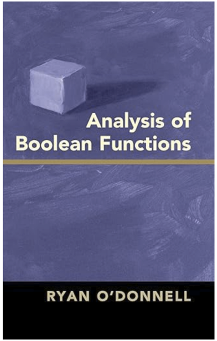
Progress Update



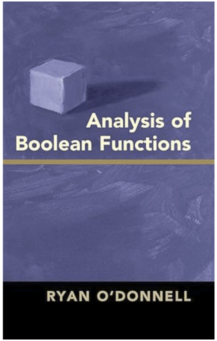
Progress Update



Fourier Polynomial

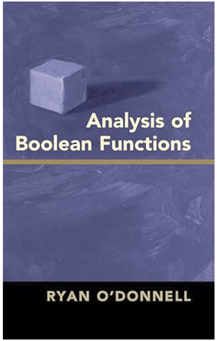


Fourier Polynomial



Fourier transform of the probability mass function

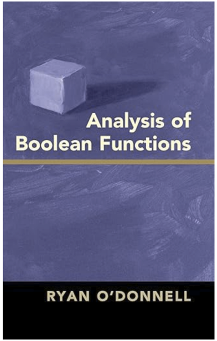
Fourier Polynomial



Fourier transform of the probability mass function

- Graphical model approximate inference
- Characteristic Circuits

Fourier Polynomial

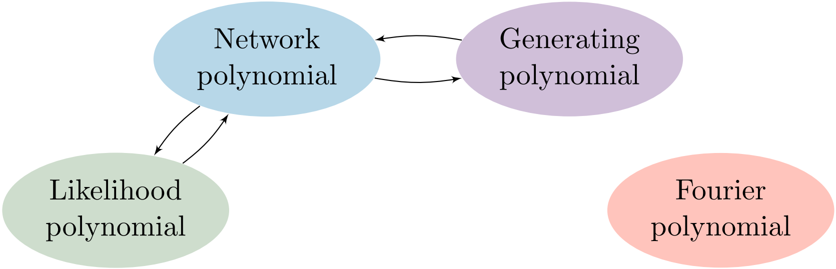


Fourier transform of the probability mass function

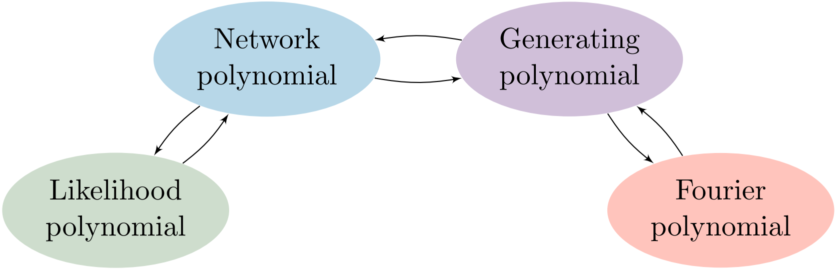
- Graphical model approximate inference
- Characteristic Circuits

Proposition. *Generating polynomials and Fourier polynomials compute* ***the same function*** *on respective domains* $\{-1, 1\}^n$ *and* $\{0, 1\}^n$*.*

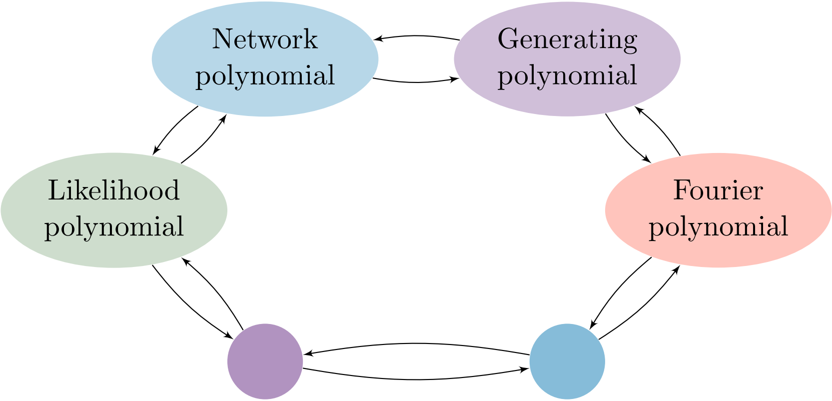
Progress Update



Progress Update



Some New Semantics



Non-binary variables?

X_1	X_2	Pr
0	1	.1
1	3	.3
3	2	.2
$\vdots$	$\vdots$	$\vdots$

Non-binary variables?

Literature: just use a binary encoding

X_1	X_2	Pr
0	1	.1
1	3	.3
3	2	.2
⋮	⋮	⋮

Non-binary variables?

Literature: just use a binary encoding

X_1	X_2	Pr
0	1	.1
1	3	.3
3	2	.2
⋮	⋮	⋮

$$g(x) = .1x_2 + .3x_1x_2^3 + .2x_1^3x_2^2 + \dots$$

Generating polynomial

Non-binary variables?

Literature: just use a binary encoding

X_1	X_2	Pr
0	1	.1
1	3	.3
3	2	.2
$\vdots$	$\vdots$	$\vdots$

$$g(x) = .1x_2 + .3x_1x_2^3 + .2x_1^3x_2^2 + \dots$$

Generating polynomial

Theorem. *For $|K| \geq 4$, computing likelihoods on a circuit for $g(x)$ is #P-hard.*

Approach: Reduction from $0,1$-permanent.

Conclusion

What we've done:

- Shown several distinct circuit models are equally expressive-efficient
- Unified existing (and one new) inference algorithms
- Inference is #P-hard in generating polynomials circuits for $k \geq 4$ categories

Conclusion

What we've done:

- Shown several distinct circuit models are equally expressive-efficient
- Unified existing (and one new) inference algorithms
- Inference is #P-hard in generating polynomials circuits for $k \geq 4$ categories

What's next?

- How can this theoretical progress be leveraged in practice?
- Are there more expressive-efficient tractable representations?

Conclusion

What we've done:

- Shown several distinct circuit models are equally expressive-efficient
- Unified existing (and one new) inference algorithms
- Inference is #P-hard in generating polynomials circuits for $k \geq 4$ categories

What's next?

- How can this theoretical progress be leveraged in practice?
- Are there more expressive-efficient tractable representations?

Thank you! Questions?