Polynomial semantics of probabilistic circuits

Oliver Broadrick, UCLA

May, 2024

Based on joint work with Honghua Zhang and Guy Van den Broeck

Outline

- Tractable probabilistic models

Outline

- Tractable probabilistic models
- Probabilistic circuits

Outline

- Tractable probabilistic models
- Probabilistic circuits
- Several circuit semantics in the literature ...

Outline

- Tractable probabilistic models
- Probabilistic circuits
- Several circuit semantics in the literature ...
- are equivalent! (for binary random variables)

Outline

- Tractable probabilistic models
- Probabilistic circuits
- Several circuit semantics in the literature ...
- are equivalent! (for binary random variables)
- And, don't all extend to non-binary variables

Probabilistic Models

Probabilistic Models

How we think about the world: models with uncertainty

Probabilistic Models

How we think about the world: models with uncertainty

- Will I make it to the SNAIL talk on time, if I leave home at 2:30pm?

Probabilistic Models

How we think about the world: models with uncertainty

- Will I make it to the SNAIL talk on time, if I leave home at 2:30pm?
- Did I pass that final exam?

Probabilistic Models

How we think about the world: models with uncertainty

- Will I make it to the SNAIL talk on time, if I leave home at 2:30pm?
- Did I pass that final exam?

AI research

Probabilistic Models

How we think about the world: models with uncertainty

- Will I make it to the SNAIL talk on time, if I leave home at 2:30pm?
- Did I pass that final exam?

AI research

- Deep learning and formal methods: “neuro-symbolic AI”

Probabilistic Models

How we think about the world: models with uncertainty

- Will I make it to the SNAIL talk on time, if I leave home at 2:30pm?
- Did I pass that final exam?

AI research

- Deep learning and formal methods: “neuro-symbolic AI”
- Applications: images, language, audio, medicine, science, economics, etc.

The problem

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

The problem

- Expressive-efficient representation

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

The problem

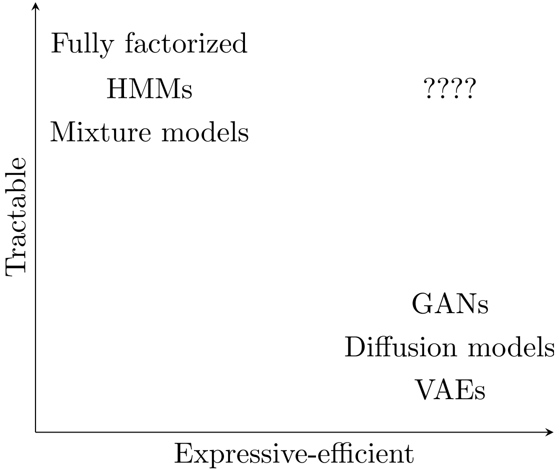
- Expressive-efficient representation
- Tractable inference

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

The problem

- Expressive-efficient representation
- Tractable inference

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

Marginal inference

If $\boldsymbol{X} = \boldsymbol{Y} \sqcup \boldsymbol{Z}$, then what is $\Pr[\boldsymbol{Y} = \boldsymbol{y}]$?

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

If $\boldsymbol{X} = \boldsymbol{Y} \sqcup \boldsymbol{Z}$, then what is $\Pr[\boldsymbol{Y} = \boldsymbol{y}]$?

In general:

$$\Pr[\boldsymbol{Y} = \boldsymbol{y}] = \sum_{\boldsymbol{z}} \Pr[\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{Z} = \boldsymbol{z}]$$

Marginal inference

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

If $\boldsymbol{X} = \boldsymbol{Y} \sqcup \boldsymbol{Z}$, then what is $\Pr[\boldsymbol{Y} = \boldsymbol{y}]$?

In general:

$$\Pr[\boldsymbol{Y} = \boldsymbol{y}] = \sum_{\boldsymbol{z}} \Pr[\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{Z} = \boldsymbol{z}]$$

For example:

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

Marginal inference

If $\boldsymbol{X} = \boldsymbol{Y} \sqcup \boldsymbol{Z}$, then what is $\Pr[\boldsymbol{Y} = \boldsymbol{y}]$?

In general:

$$\Pr[\boldsymbol{Y} = \boldsymbol{y}] = \sum_{\boldsymbol{z}} \Pr[\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{Z} = \boldsymbol{z}]$$

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

For example:

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

Goal: Find a model of polysize that supports marginal inference in polytime, for as large a set of probability distributions as possible.

Approaches

Approaches

- Bayesian Networks (of bounded treewidth) (BNs)

Approaches

- Bayesian Networks (of bounded treewidth) (BNs)
- Determinantal Point Processes (DPPs)

Approaches

- Bayesian Networks (of bounded treewidth) (BNs)
- Determinantal Point Processes (DPPs)
- Probabilistic Sentential Decision Diagrams (PSDDs)
- ...

Approaches

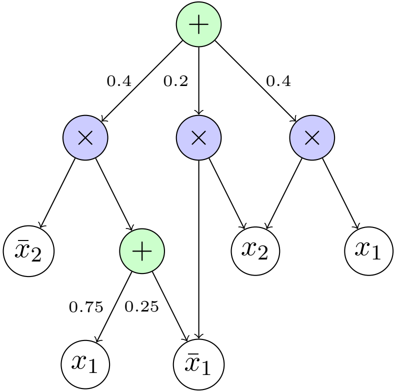
- Bayesian Networks (of bounded treewidth) (BNs)
- Determinantal Point Processes (DPPs)
- Probabilistic Sentential Decision Diagrams (PSDDs)
- ...
- Probabilistic Circuits!

Probabilistic Circuits

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

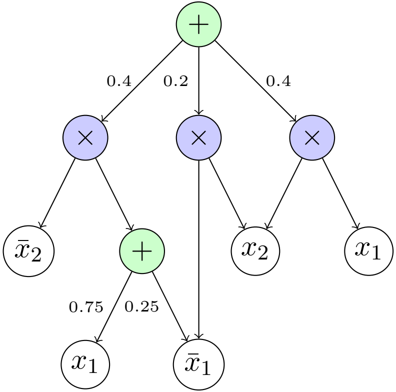
Probabilistic Circuits

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



Probabilistic Circuits

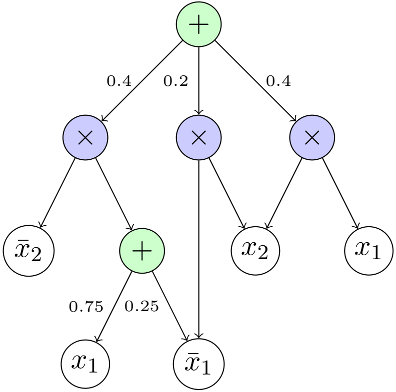
X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4



Marginal Inference:

Probabilistic Circuits

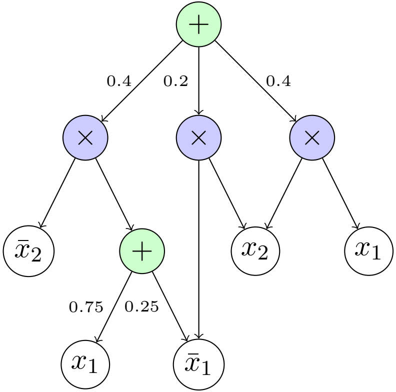
X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4



Marginal Inference:
If $X_i = b$ for $b \in \{0, 1\}$, set $x_i = b$ and $\bar{x}_i = 1 - b$.

Probabilistic Circuits

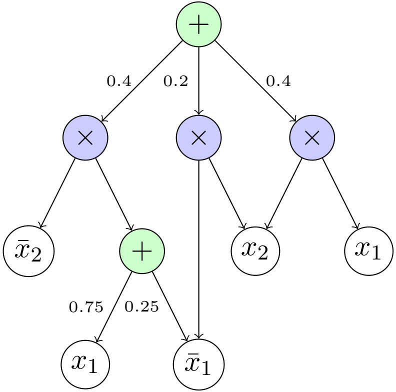
X_1	X_2	Pr
0	0	.1
0	1	.2
1	0	.3
1	1	.4



Marginal Inference:
If $X_i = b$ for $b \in \{0, 1\}$, set $x_i = b$ and $\bar{x}_i = 1 - b$.
If X_i is not assigned, set $x_1 = 1$ and $\bar{x}_1 = 1$.

Probabilistic Circuits

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

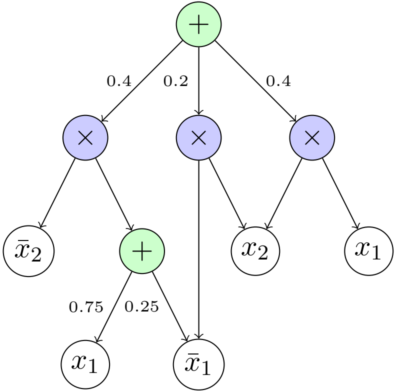


Marginal Inference:
If $X_i = b$ for $b \in \{0, 1\}$, set $x_i = b$ and $\bar{x}_i = 1 - b$.
If X_i is not assigned, set $x_1 = 1$ and $\bar{x}_1 = 1$.

$\Pr[X_1 = 1]$?
Set $x_1 = 1$, $\bar{x}_1 = 0$, $x_2 = 1$, $\bar{x}_2 = 1$

Probabilistic Circuits

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



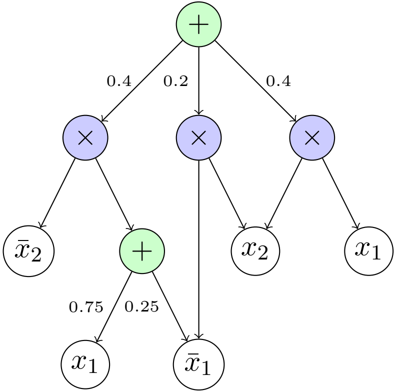
Marginal Inference:
If $X_i = b$ for $b \in \{0, 1\}$, set $x_i = b$ and $\bar{x}_i = 1 - b$.
If X_i is not assigned, set $x_1 = 1$ and $\bar{x}_1 = 1$.

$\Pr[X_1 = 1]$?
Set $x_1 = 1$, $\bar{x}_1 = 0$, $x_2 = 1$, $\bar{x}_2 = 1$

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

Probabilistic Circuits

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



Marginal Inference:
If $X_i = b$ for $b \in \{0, 1\}$, set $x_i = b$ and $\bar{x}_i = 1 - b$.
If X_i is not assigned, set $x_1 = 1$ and $\bar{x}_1 = 1$.

$\Pr[X_1 = 1]$?
Set $x_1 = 1$, $\bar{x}_1 = 0$, $x_2 = 1$, $\bar{x}_2 = 1$

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

"Network polynomial"

Circuit Semantics

Polynomial	Notation	Inference	Citation
Network polynomial	$p(x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n)$	✓	Darwiche [2003]

Circuit Semantics

Polynomial	Notation	Inference	Citation
Network polynomial	$p(x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n)$	✓	Darwiche [2003]
Likelihood polynomial	$p(x_1, \ldots, x_n)$	?	Roth and Samdani [2009]

Circuit Semantics

Polynomial	Notation	Inference	Citation
Network polynomial	$p(x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n)$	✓	Darwiche [2003]
Likelihood polynomial	$p(x_1, \ldots, x_n)$	?	Roth and Samdani [2009]
Generating function	$g(x_1, \ldots, x_n)$	✓	Zhang et al. [2021]

Circuit Semantics

Polynomial	Notation	Inference	Citation
Network polynomial	$p(x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n)$	✓	Darwiche [2003]
Likelihood polynomial	$p(x_1, \ldots, x_n)$	?	Roth and Samdani [2009]
Generating function	$g(x_1, \ldots, x_n)$	✓	Zhang et al. [2021]
Fourier transform	$\hat{p}(x_1, \ldots, x_n)$	✓	Yu et al. [2023]

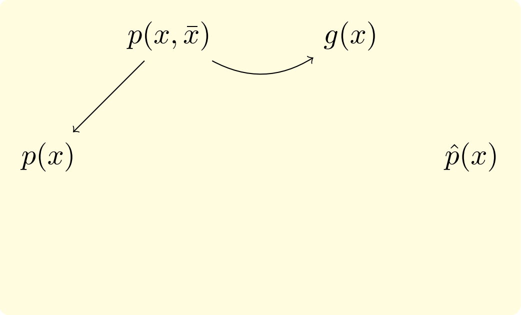
How do they relate?

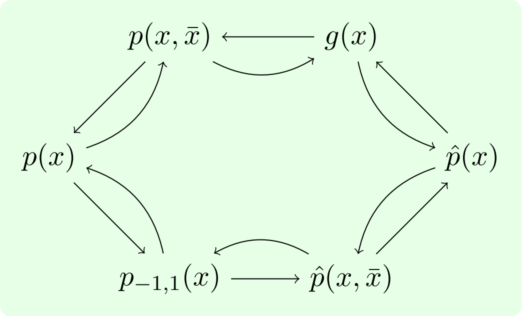
$p(x, \bar{x})$	Network polynomial
$p(x)$	Likelihood polynomial
$g(x)$	Generating function
$\hat{p}(x)$	Fourier transform

$$\begin{tikzcd}
 & p(x, \bar{x}) \arrow[dl] \arrow[r, bend right] & g(x) & \\
p(x) & & & \hat{p}(x)
\end{tikzcd}$$

How do they relate?

$p(x, \bar{x})$	Network polynomial
$p(x)$	Likelihood polynomial
$g(x)$	Generating function
$\hat{p}(x)$	Fourier transform





Likelihood polynomials

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

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

Likelihood polynomials

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

Can we do marginal inference?

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

Likelihood polynomials

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

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

Can we do marginal inference?
Relation to network polynomial?

Likelihood polynomials

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

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

Can we do marginal inference?
Relation to network polynomial?

Transformation from network to a likelihood:
$$p(x_1, x_2, \bar{x}_1, \bar{x}_2) = .1\bar{x}_1\bar{x}_2 + .2\bar{x}_1 x_2 + .3x_1\bar{x}_2 + .4x_1 x_2$$

Likelihood polynomials

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

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

Can we do marginal inference?
Relation to network polynomial?

Transformation from network to a likelihood:

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

$$p(x_1, x_2, 1 - x_1, 1 - x_2)$$

Likelihood polynomials

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

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

Can we do marginal inference?
Relation to network polynomial?

Transformation from network to a likelihood:

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

$$p(x_1, x_2, 1 - x_1, 1 - x_2)$$
$$= .1(1 - x_1)(1 - x_2) + .2(1 - x_1)x_2 + .3x_1(1 - x_2) + .4x_1x_2$$

Likelihood polynomials

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

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

Can we do marginal inference?
Relation to network polynomial?

Transformation from network to a likelihood:

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

$$p(x_1, x_2, 1 - x_1, 1 - x_2)$$

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

$$= .2x_1 + .1x_2 + .1$$

Transformation from likelihood to network

Theorem 1. *Let* Pr *be a probability distribution on* n *binary random variables. Then a circuit of size* s *computing the likelihood polynomial for* Pr *can be transformed to a circuit of size* $O(sn^2)$ *computing the network polynomial for* Pr.

Transformation from likelihood to network

Theorem 1. *Let* Pr *be a probability distribution on n binary random variables. Then a circuit of size s computing the likelihood polynomial for* Pr *can be transformed to a circuit of size $O(sn^2)$ computing the network polynomial for* Pr.

Idea: $\left(\prod_{i=1}^{n}(x_i + \bar{x}_i)\right) p\left(\frac{x_1}{x_1 + \bar{x}_1}, \ldots, \frac{x_n}{x_n + \bar{x}_n}\right)$

Transformation from likelihood to network

Theorem 1. *Let* Pr *be a probability distribution on n binary random variables. Then a circuit of size s computing the likelihood polynomial for* Pr *can be transformed to a circuit of size $O(sn^2)$ computing the network polynomial for* Pr.

Idea: $\left(\prod_{i=1}^{n}(x_i + \bar{x}_i)\right) p\left(\frac{x_1}{x_1 + \bar{x}_1}, \ldots, \frac{x_n}{x_n + \bar{x}_n}\right)$

Example: Starting with $p(x_1, x_2) = .2x_1 + .1x_2 + .1$, we form

Transformation from likelihood to network

Theorem 1. *Let* Pr *be a probability distribution on n binary random variables. Then a circuit of size s computing the likelihood polynomial for* Pr *can be transformed to a circuit of size $O(sn^2)$ computing the network polynomial for* Pr.

Idea: $\left(\prod_{i=1}^{n}(x_i + \bar{x}_i)\right) p\left(\frac{x_1}{x_1 + \bar{x}_1}, \ldots, \frac{x_n}{x_n + \bar{x}_n}\right)$

Example: Starting with $p(x_1, x_2) = .2x_1 + .1x_2 + .1$, we form

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

Transformation from likelihood to network

Theorem 1. *Let* $\Pr$ *be a probability distribution on* n *binary random variables. Then a circuit of size* s *computing the likelihood polynomial for* $\Pr$ *can be transformed to a circuit of size* $O(sn^2)$ *computing the network polynomial for* $\Pr$.

Idea: $\left(\prod_{i=1}^{n}(x_i+\bar{x}_i)\right) p\left(\frac{x_1}{x_1+\bar{x}_1},\ldots,\frac{x_n}{x_n+\bar{x}_n}\right)$

Example: Starting with $p(x_1,x_2)=.2x_1+.1x_2+.1$, we form

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

Transformation from likelihood to network

Theorem 1. *Let* Pr *be a probability distribution on n binary random variables. Then a circuit of size s computing the likelihood polynomial for* Pr *can be transformed to a circuit of size $O(sn^2)$ computing the network polynomial for* Pr.

Idea: $\left(\prod_{i=1}^{n}(x_i + \bar{x}_i)\right) p\left(\frac{x_1}{x_1 + \bar{x}_1}, \ldots, \frac{x_n}{x_n + \bar{x}_n}\right)$

Example: Starting with $p(x_1, x_2) = .2x_1 + .1x_2 + .1$, we form

$$
\begin{aligned}
&(x_1 + \bar{x}_1)(x_2 + \bar{x}_2)\left(.2\frac{x_1}{x_1 + \bar{x}_1} + .1\frac{x_2}{x_2 + \bar{x}_2} + .1\right) \\
&= .2x_1(x_2 + \bar{x}_2) + .1x_2(x_1 + \bar{x}_1) + .1(x_1 + \bar{x}_1)(x_2 + \bar{x}_2) \\
&= .2x_1x_2 + .2x_1\bar{x}_2 + .1x_1x_2 + .1\bar{x}_1x_2 + .1x_1x_2 + .1\bar{x}_1x_2 + .1x_1\bar{x}_2 + .1\bar{x}_1\bar{x}_2
\end{aligned}
$$

Transformation from likelihood to network

Theorem 1. *Let* Pr *be a probability distribution on n binary random variables. Then a circuit of size s computing the likelihood polynomial for* Pr *can be transformed to a circuit of size $O(sn^2)$ computing the network polynomial for* Pr.

Idea: $\left(\prod_{i=1}^{n}(x_i + \bar{x}_i)\right) p\left(\frac{x_1}{x_1+\bar{x}_1}, \ldots, \frac{x_n}{x_n+\bar{x}_n}\right)$

Example: Starting with $p(x_1, x_2) = .2x_1 + .1x_2 + .1$, we form

$$
\begin{aligned}
&(x_1+\bar{x}_1)(x_2+\bar{x}_2)\left(.2\frac{x_1}{x_1+\bar{x}_1} + .1\frac{x_2}{x_2+\bar{x}_2} + .1\right)\\
&= .2x_1(x_2+\bar{x}_2) + .1x_2(x_1+\bar{x}_1) + .1(x_1+\bar{x}_1)(x_2+\bar{x}_2)\\
&= .2x_1x_2 + .2x_1\bar{x}_2 + .1x_1x_2 + .1\bar{x}_1x_2 + .1x_1x_2 + .1\bar{x}_1x_2 + .1x_1\bar{x}_2 + .1\bar{x}_1\bar{x}_2\\
&= .1\bar{x}_1\bar{x}_2 + .2\bar{x}_1x_2 + .3x_1\bar{x}_2 + .4x_1x_2 = p(x_1, x_2, \bar{x}_1, \bar{x}_2)
\end{aligned}
$$

Divisions?

Wait a second... $\left(\prod_{i=1}^{n}(x_i + \bar{x}_i)\right) p\left(\frac{x_1}{x_1 + \bar{x}_1}, \ldots, \frac{x_n}{x_n + \bar{x}_n}\right)$

Divisions?

Wait a second... $\left(\prod_{i=1}^{n}(x_i + \bar{x}_i)\right) p\left(\frac{x_1}{x_1 + \bar{x}_1}, \ldots, \frac{x_n}{x_n + \bar{x}_n}\right)$

Theorem 2 (Strassen). *You can remove the divisions in polynomial time!*

Divisions?

Wait a second... $\left(\prod_{i=1}^{n}(x_i + \bar{x}_i)\right) p\left(\frac{x_1}{x_1 + \bar{x}_1}, \dots, \frac{x_n}{x_n + \bar{x}_n}\right)$

Theorem 2 (Strassen). *You can remove the divisions in polynomial time!*

Lemma 3. *For a circuit computing f of degree d, we can obtain circuits computing $H_0[f], H_1[f], \dots, H_d[f]$ the homogeneous parts of f, i.e. $H_i[f]$ has degree i and $f = \sum_i H_i[f]$.*

Divisions?

Wait a second... $\left(\prod_{i=1}^{n}(x_i + \bar{x}_i)\right) p\left(\frac{x_1}{x_1 + \bar{x}_1}, \ldots, \frac{x_n}{x_n + \bar{x}_n}\right)$

Theorem 2 (Strassen). *You can remove the divisions in polynomial time!*

Lemma 3. *For a circuit computing f of degree d, we can obtain circuits computing $H_0[f], H_1[f], \ldots, H_d[f]$ the homogeneous parts of f, i.e. $H_i[f]$ has degree i and $f = \sum_i H_i[f]$.*

Idea: Move divisions to the root using $a/b + c/d = (ad + bc)/bd$ and $a/b * c/d = ac/bd$.

Divisions?

Wait a second... $\left(\prod_{i=1}^{n}(x_i + \bar{x}_i)\right) p\left(\frac{x_1}{x_1+\bar{x}_1}, \ldots, \frac{x_n}{x_n+\bar{x}_n}\right)$

Theorem 2 (Strassen). *You can remove the divisions in polynomial time!*

Lemma 3. *For a circuit computing f of degree d, we can obtain circuits computing $H_0[f], H_1[f], \ldots, H_d[f]$ the homogeneous parts of f, i.e. $H_i[f]$ has degree i and $f = \sum_i H_i[f]$.*

Idea: Move divisions to the root using $a/b + c/d = (ad + bc)/bd$ and $a/b * c/d = ac/bd$.

Then for circuit a/b computing polynomial $f = a/b$ of degree d, assume $b(0) = 1$, and we have

$$H_i[f] = H_i[a(1 + (1-b) + (1-b)^2 + \ldots + (1-b)^d].$$

Progress update

$p(x,\bar{x})$	Network polynomial
$p(x)$	Likelihood polynomial
$g(x)$	Generating function
$\hat{p}(x)$	Fourier transform

$$
\begin{tikzcd}
 & p(x,\bar{x}) \arrow[dl] & g(x) & \\
p(x) & & & \hat{p}(x)
\end{tikzcd}
$$

Progress update

$p(x, \bar{x})$	Network polynomial
$p(x)$	Likelihood polynomial
$g(x)$	Generating function
$\hat{p}(x)$	Fourier transform

$$
\begin{tikzcd}
 & p(x, \bar{x}) \arrow[dl] & g(x) & \\
p(x) \arrow[ur, bend right] & & & \hat{p}(x)
\end{tikzcd}
$$

Progress update

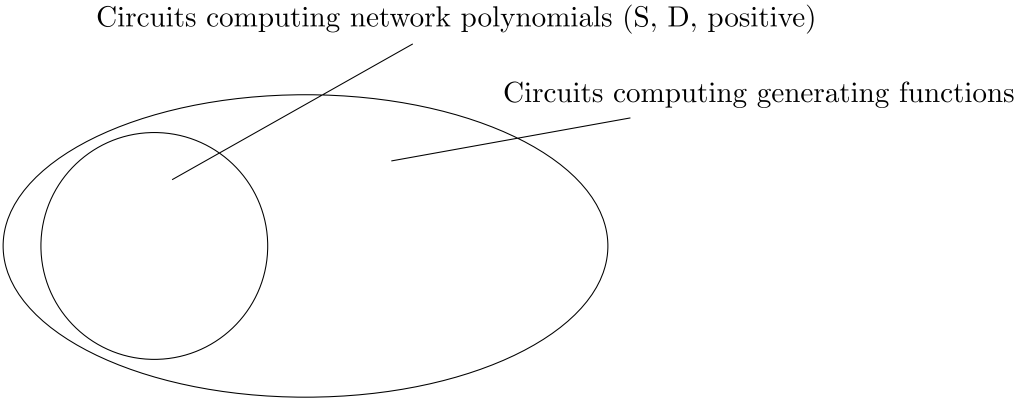
$p(x,\bar{x})$	Network polynomial
$p(x)$	Likelihood polynomial
$g(x)$	Generating function
$\hat{p}(x)$	Fourier transform

$$\begin{tikzcd}
 & p(x,\bar{x}) \arrow[dl] & g(x) & \\
p(x) \arrow[ur, bend right] & & & \hat{p}(x) \arrow[dl, bend right] \\
 & & \hat{p}(x,\bar{x}) \arrow[ur] &
\end{tikzcd}$$

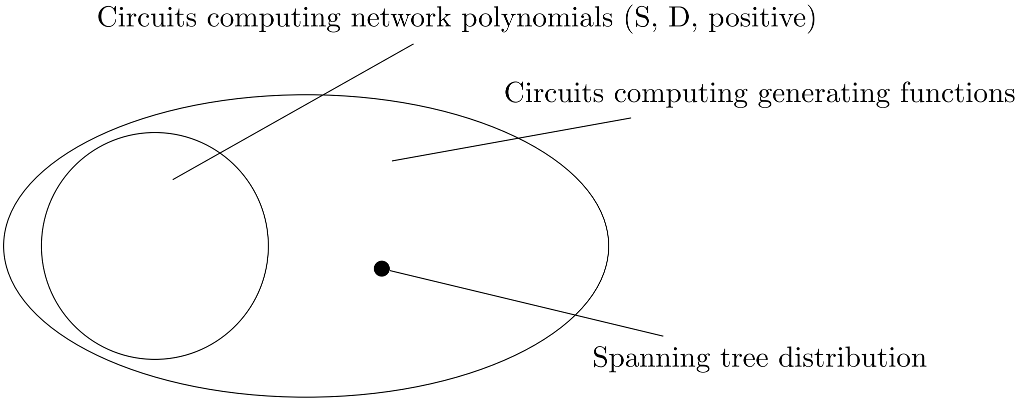
Generating functions:

Generating functions: Why?

Generating functions: Why?



Generating functions: Why?



Generating functions

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

Can we do marginal inference?

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

Generating functions

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

Can we do marginal inference?
For $X_i = 1$, set $x_i = t$

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

Generating functions

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

Can we do marginal inference?
For $X_i = 1$, set $x_i = t$
For $X_i = 0$, set $x_i = 0$

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

Generating functions

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

Can we do marginal inference?

For $X_i = 1$, set $x_i = t$

For $X_i = 0$, set $x_i = 0$

For $X_i = ?$, set $x_i = 1$

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

Generating functions

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

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

Can we do marginal inference?
For $X_i = 1$, set $x_i = t$
For $X_i = 0$, set $x_i = 0$
For $X_i = ?$, set $x_i = 1$

$\Pr[X_1 = 1]$?
$g(t, 1) = .1 + .2 + .3t + .4t = .3 + .7t$

Generating functions

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

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

Can we do marginal inference?
For $X_i = 1$, set $x_i = t$
For $X_i = 0$, set $x_i = 0$
For $X_i =?$, set $x_i = 1$

$\Pr[X_1 = 1]$?
$g(t, 1) = .1 + .2 + .3t + .4t = .3 + .7t$

Relation to network polynomial?

Generating functions

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

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

Can we do marginal inference?
For $X_i = 1$, set $x_i = t$
For $X_i = 0$, set $x_i = 0$
For $X_i =?$, set $x_i = 1$

$\Pr[X_1 = 1]$?
$g(t, 1) = .1 + .2 + .3t + .4t = .3 + .7t$

Relation to network polynomial?

Transformation from network to a generating:
$p(x_1, x_2, \bar{x}_1, \bar{x}_2) = .1\bar{x}_1\bar{x}_2 + .2\bar{x}_1x_2 + .3x_1\bar{x}_2 + .4x_1x_2$

Generating functions

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

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

Can we do marginal inference?
For $X_i = 1$, set $x_i = t$
For $X_i = 0$, set $x_i = 0$
For $X_i =?$, set $x_i = 1$

$\Pr[X_1 = 1]$?
$g(t, 1) = .1 + .2 + .3t + .4t = .3 + .7t$

Relation to network polynomial?

Transformation from network to a generating:
$p(x_1, x_2, \bar{x}_1, \bar{x}_2) = .1\bar{x}_1\bar{x}_2 + .2\bar{x}_1x_2 + .3x_1\bar{x}_2 + .4x_1x_2$
$p(x_1, x_2, 1, 1) = .1 + .2x_2 + .3x_1 + .4x_1x_2 = g(x_1, x_2)$

Transformation from generating to network

Transformation from generating to network

Theorem 4. *Let* Pr *be a probability distribution on* n *binary random variables. Then a circuit of size* s *computing the probability generating function for* Pr *can be transformed to a circuit of size* $O(sn^2)$ *computing the network polynomial for* Pr.

Transformation from generating to network

Theorem 4. *Let* Pr *be a probability distribution on n binary random variables. Then a circuit of size s computing the probability generating function for* Pr *can be transformed to a circuit of size $O(sn^2)$ computing the network polynomial for* Pr.

Idea: $\left(\prod_{i=1}^{n} \bar{x}_i\right) g\left(\frac{x_1}{\bar{x}_1}, \frac{x_2}{\bar{x}_2}, \ldots, \frac{x_n}{\bar{x}_n}\right)$

Transformation from generating to network

Theorem 4. *Let* Pr *be a probability distribution on n binary random variables. Then a circuit of size s computing the probability generating function for* Pr *can be transformed to a circuit of size $O(sn^2)$ computing the network polynomial for* Pr.

Idea: $\left(\prod_{i=1}^{n} \bar{x}_i\right) g\left(\frac{x_1}{\bar{x}_1}, \frac{x_2}{\bar{x}_2}, \ldots, \frac{x_n}{\bar{x}_n}\right)$

Example: Starting with $g(x) = .1 + .2x_2 + .3x_1 + .4x_1x_2$, we form

Transformation from generating to network

Theorem 4. *Let* Pr *be a probability distribution on n binary random variables. Then a circuit of size s computing the probability generating function for* Pr *can be transformed to a circuit of size $O(sn^2)$ computing the network polynomial for* Pr.

Idea: $\left(\prod_{i=1}^{n} \bar{x}_i\right) g\left(\frac{x_1}{\bar{x}_1}, \frac{x_2}{\bar{x}_2}, \ldots, \frac{x_n}{\bar{x}_n}\right)$

Example: Starting with $g(x) = .1 + .2x_2 + .3x_1 + .4x_1x_2$, we form

$$(\bar{x}_1\bar{x}_2)\left(.1 + .2\frac{x_2}{\bar{x}_2} + .3\frac{x_1}{\bar{x}_1} + .4\frac{x_1}{\bar{x}_1}\frac{x_2}{\bar{x}_2}\right)$$

Transformation from generating to network

Theorem 4. *Let* Pr *be a probability distribution on n binary random variables. Then a circuit of size s computing the probability generating function for* Pr *can be transformed to a circuit of size $O(sn^2)$ computing the network polynomial for* Pr.

Idea: $\left(\prod_{i=1}^{n} \bar{x}_i\right) g\left(\frac{x_1}{\bar{x}_1}, \frac{x_2}{\bar{x}_2}, \ldots, \frac{x_n}{\bar{x}_n}\right)$

Example: Starting with $g(x) = .1 + .2x_2 + .3x_1 + .4x_1x_2$, we form

$$\begin{aligned}
&(\bar{x}_1\bar{x}_2)\left(.1 + .2\frac{x_2}{\bar{x}_2} + .3\frac{x_1}{\bar{x}_1} + .4\frac{x_1}{\bar{x}_1}\frac{x_2}{\bar{x}_2}\right) \\
&= .1\bar{x}_1\bar{x}_2 + .2\bar{x}_1x_2 + .3x_1\bar{x}_2 + .4x_1x_2
\end{aligned}$$

Transformation from generating to network

Theorem 4. *Let* Pr *be a probability distribution on* n *binary random variables. Then a circuit of size* s *computing the probability generating function for* Pr *can be transformed to a circuit of size* $O(sn^2)$ *computing the network polynomial for* Pr.

Idea: $\left(\prod_{i=1}^{n} \bar{x}_i\right) g\left(\frac{x_1}{\bar{x}_1}, \frac{x_2}{\bar{x}_2}, \ldots, \frac{x_n}{\bar{x}_n}\right)$

Example: Starting with $g(x) = .1 + .2x_2 + .3x_1 + .4x_1x_2$, we form

$$\begin{aligned}
&(\bar{x}_1\bar{x}_2)\left(.1 + .2\frac{x_2}{\bar{x}_2} + .3\frac{x_1}{\bar{x}_1} + .4\frac{x_1}{\bar{x}_1}\frac{x_2}{\bar{x}_2}\right) \\
&= .1\bar{x}_1\bar{x}_2 + .2\bar{x}_1x_2 + .3x_1\bar{x}_2 + .4x_1x_2 \\
&= p(x, \bar{x})
\end{aligned}$$

Transformation from generating to network

Theorem 4. *Let* Pr *be a probability distribution on n binary random variables. Then a circuit of size s computing the probability generating function for* Pr *can be transformed to a circuit of size $O(sn^2)$ computing the network polynomial for* Pr.

Idea: $\left(\prod_{i=1}^{n} \bar{x}_i\right) g\left(\frac{x_1}{\bar{x}_1}, \frac{x_2}{\bar{x}_2}, \ldots, \frac{x_n}{\bar{x}_n}\right)$

Example: Starting with $g(x) = .1 + .2x_2 + .3x_1 + .4x_1x_2$, we form

$$
\begin{aligned}
&(\bar{x}_1\bar{x}_2)\left(.1 + .2\frac{x_2}{\bar{x}_2} + .3\frac{x_1}{\bar{x}_1} + .4\frac{x_1}{\bar{x}_1}\frac{x_2}{\bar{x}_2}\right) \\
&= .1\bar{x}_1\bar{x}_2 + .2\bar{x}_1x_2 + .3x_1\bar{x}_2 + .4x_1x_2 \\
&= p(x, \bar{x})
\end{aligned}
$$

Strassen to the rescue!

Progress update

$p(x, \bar{x})$	Network polynomial
$p(x)$	Likelihood polynomial
$g(x)$	Generating function
$\hat{p}(x)$	Fourier transform

$$
\begin{tikzcd}
 & p(x, \bar{x}) \arrow[dl] & g(x) & \\
p(x) \arrow[ur, bend right] & & & \hat{p}(x) \arrow[dl, bend right] \\
 & & \hat{p}(x, \bar{x}) \arrow[ur] &
\end{tikzcd}
$$

Progress update

$p(x,\bar{x})$	Network polynomial
$p(x)$	Likelihood polynomial
$g(x)$	Generating function
$\hat{p}(x)$	Fourier transform

$$\begin{tikzcd}
& p(x,\bar{x}) \arrow[dl] \arrow[r, bend right] & g(x) \arrow[l] & \\
p(x) \arrow[ur, bend right] & & & \hat{p}(x) \arrow[dl, bend right] \\
& & \hat{p}(x,\bar{x}) \arrow[ur] &
\end{tikzcd}$$

Progress update

$p(x, \bar{x})$	Network polynomial
$p(x)$	Likelihood polynomial
$g(x)$	Generating function
$\hat{p}(x)$	Fourier transform

$$
\begin{tikzcd}
 & p(x,\bar{x}) \arrow[dl] \arrow[r, bend right] & g(x) \arrow[l] & \\
p(x) \arrow[ur, bend right] & & & \hat{p}(x) \arrow[dl, bend right] \\
 & p_{-1,1}(x) \arrow[r] & \hat{p}(x,\bar{x}) \arrow[ur] \arrow[l, bend right] &
\end{tikzcd}
$$

Progress update

$p(x, \bar{x})$	Network polynomial
$p(x)$	Likelihood polynomial
$g(x)$	Generating function
$\hat{p}(x)$	Fourier transform

$$
\begin{tikzcd}
 & p(x,\bar{x}) \arrow[dl] \arrow[r, bend right] & g(x) \arrow[l] & \\
p(x) \arrow[ur, bend right] & & & \hat{p}(x) \arrow[dl, bend right] \\
 & p_{-1,1}(x) \arrow[r] & \hat{p}(x,\bar{x}) \arrow[l, bend right] \arrow[ur] &
\end{tikzcd}
$$

Proposition 1. *For binary random variables, probability generating functions $g(x)$ and Fourier polynomials $\hat{p}(x)$ are the same function(!), on respective domains $\{-1,1\}^n$ and $\{0,1\}^n$, up to the bijection $\phi : \{0,1\} \to \{-1,1\}$ given by $\phi(b) = (-1)^b$ applied bitwise.*

Progress update

$p(x, \bar{x})$	Network polynomial
$p(x)$	Likelihood polynomial
$g(x)$	Generating function
$\hat{p}(x)$	Fourier transform

$$\begin{tikzcd}
 & p(x, \bar{x}) \arrow[dl] \arrow[r, bend right] & g(x) \arrow[l] \arrow[dr, bend right] & \\
p(x) \arrow[ur, bend right] \arrow[dr] & & & \hat{p}(x) \arrow[ul] \arrow[dl, bend right] \\
 & p_{-1,1}(x) \arrow[ul, bend right] \arrow[r] & \hat{p}(x, \bar{x}) \arrow[l, bend right] \arrow[ur] &
\end{tikzcd}$$

Proposition 1. *For binary random variables, probability generating functions $g(x)$ and Fourier polynomials $\hat{p}(x)$ are the same function(!), on respective domains $\{-1, 1\}^n$ and $\{0, 1\}^n$, up to the bijection $\phi : \{0, 1\} \to \{-1, 1\}$ given by $\phi(b) = (-1)^b$ applied bitwise.*

Progress update

$p(x,\bar{x})$	Network polynomial
$p(x)$	Likelihood polynomial
$g(x)$	Generating function
$\hat{p}(x)$	Fourier transform

$$
\begin{tikzcd}
 & p(x,\bar{x}) \arrow[dl] \arrow[r, bend right] & g(x) \arrow[l] \arrow[dr, bend right] & \\
p(x) \arrow[ur, bend right] \arrow[dr] & & & \hat{p}(x) \arrow[ul] \arrow[dl, bend right] \\
 & p_{-1,1}(x) \arrow[ul, bend right] \arrow[r] & \hat{p}(x,\bar{x}) \arrow[l, bend right] \arrow[ur] &
\end{tikzcd}
$$

Proposition 1. *For binary random variables, probability generating functions $g(x)$ and Fourier polynomials $\hat{p}(x)$ are the same function(!), on respective domains $\{-1,1\}^n$ and $\{0,1\}^n$, up to the bijection $\phi : \{0,1\} \to \{-1,1\}$ given by $\phi(b) = (-1)^b$ applied bitwise.*

What have we done?

Progress update

$p(x,\bar{x})$	Network polynomial
$p(x)$	Likelihood polynomial
$g(x)$	Generating function
$\hat{p}(x)$	Fourier transform

$$
\begin{tikzcd}
 & p(x,\bar{x}) \arrow[dl] \arrow[r, bend right] & g(x) \arrow[l] \arrow[dr, bend right] & \\
p(x) \arrow[ur, bend right] \arrow[dr] & & & \hat{p}(x) \arrow[ul] \arrow[dl, bend right] \\
 & p_{-1,1}(x) \arrow[ul, bend right] \arrow[r] & \hat{p}(x,\bar{x}) \arrow[l, bend right] \arrow[ur] &
\end{tikzcd}
$$

Proposition 1. *For binary random variables, probability generating functions $g(x)$ and Fourier polynomials $\hat{p}(x)$ are the same function(!), on respective domains $\{-1,1\}^n$ and $\{0,1\}^n$, up to the bijection $\phi : \{0,1\} \to \{-1,1\}$ given by $\phi(b) = (-1)^b$ applied bitwise.*

What have we done?

- several distinct circuit-based models are equally succinct

Progress update

$p(x, \bar{x})$	Network polynomial
$p(x)$	Likelihood polynomial
$g(x)$	Generating function
$\hat{p}(x)$	Fourier transform

$$\begin{tikzcd}
& p(x,\bar{x}) \arrow[dl] \arrow[r, bend right] & g(x) \arrow[l] \arrow[dr, bend right] & \\
p(x) \arrow[ur, bend right] \arrow[dr] & & & \hat{p}(x) \arrow[ul] \arrow[dl, bend right] \\
& p_{-1,1}(x) \arrow[ul, bend right] \arrow[r] & \hat{p}(x,\bar{x}) \arrow[l, bend right] \arrow[ur] &
\end{tikzcd}$$

Proposition 1. *For binary random variables, probability generating functions $g(x)$ and Fourier polynomials $\hat{p}(x)$ are the same function(!), on respective domains $\{-1, 1\}^n$ and $\{0, 1\}^n$, up to the bijection $\phi : \{0, 1\} \to \{-1, 1\}$ given by $\phi(b) = (-1)^b$ applied bitwise.*

What have we done?

- several distinct circuit-based models are equally succinct
- distinct inference algorithms in a common framework

Non-binary distributions?

Non-binary distributions?

Let $\Pr : K^n \to \mathbb{R}$ be a probability mass function with $K = \{0, 1, 2, \ldots, k-1\}$. Then the probability generating polynomial of Pr is

$$g(x) = \sum_{(d_1, d_2, \ldots, d_n) \in K^n} \Pr(d_1, \ldots, d_n) x_1^{d_1} x_2^{d_2} \cdots x_n^{d_n}. \tag{1}$$

Non-binary distributions?

Let $\Pr : K^n \to \mathbb{R}$ be a probability mass function with $K = \{0, 1, 2, \ldots, k-1\}$.
Then the probability generating polynomial of Pr is

$$g(x) = \sum_{(d_1, d_2, \ldots, d_n) \in K^n} \Pr(d_1, \ldots, d_n) x_1^{d_1} x_2^{d_2} \cdots x_n^{d_n}. \tag{1}$$

Theorem 5. *For $k \geq 4$, computing likelihoods on a circuit for $g(x)$ is #P-hard.*

Proof idea: Reduce from $0, 1$-permanent.

Conclusion

What we did:

- Several distinct circuit-based models are equally succinct
- Distinct inference algorithms in a common framework
- Inference is hard in circuits computing generating functions for $k \geq 4$ categories

Conclusion

What we did:

- Several distinct circuit-based models are equally succinct
- Distinct inference algorithms in a common framework
- Inference is hard in circuits computing generating functions for $k \geq 4$ categories

What's next?

Conclusion

What we did:

- Several distinct circuit-based models are equally succinct
- Distinct inference algorithms in a common framework
- Inference is hard in circuits computing generating functions for $k \geq 4$ categories

What's next?

- Are there more succinct tractable representations? e.g., do we need multilinearity?

Conclusion

What we did:

- Several distinct circuit-based models are equally succinct
- Distinct inference algorithms in a common framework
- Inference is hard in circuits computing generating functions for $k \geq 4$ categories

What's next?

- Are there more succinct tractable representations? e.g., do we need multilinearity?
- Can we characterize *all* tractable marginal inference?

Conclusion

What we did:

- Several distinct circuit-based models are equally succinct
- Distinct inference algorithms in a common framework
- Inference is hard in circuits computing generating functions for $k \geq 4$ categories

What's next?

- Are there more succinct tractable representations? e.g., do we need multilinearity?
- Can we characterize *all* tractable marginal inference?
- How can theoretically more expressive models be learned/constructed in practice?