Building Tractable Generative Models

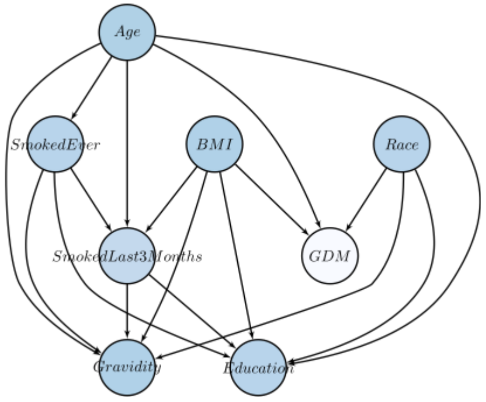
Day 1 | **Day 2** | Day 3

Probabilistic Circuits

Slides adapted from the PC tutorial by Prof. Guy Van Den Broeck's group and Einsum Network presentation

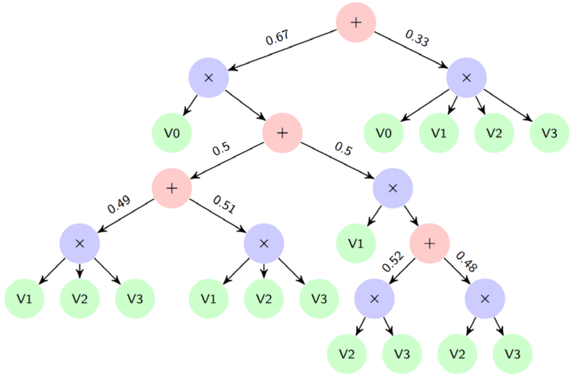
Joint distributions

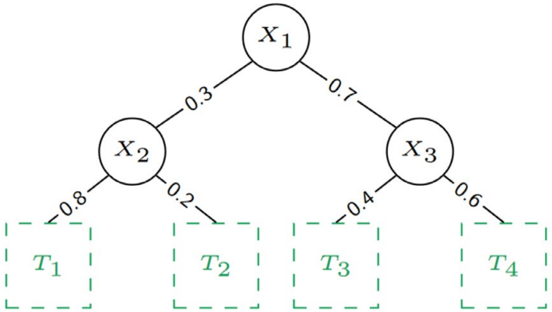
PGMs



Explainable

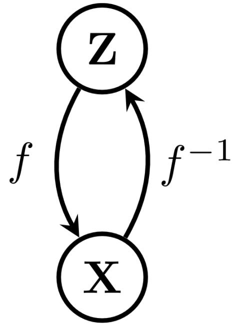
PCs





Tractable

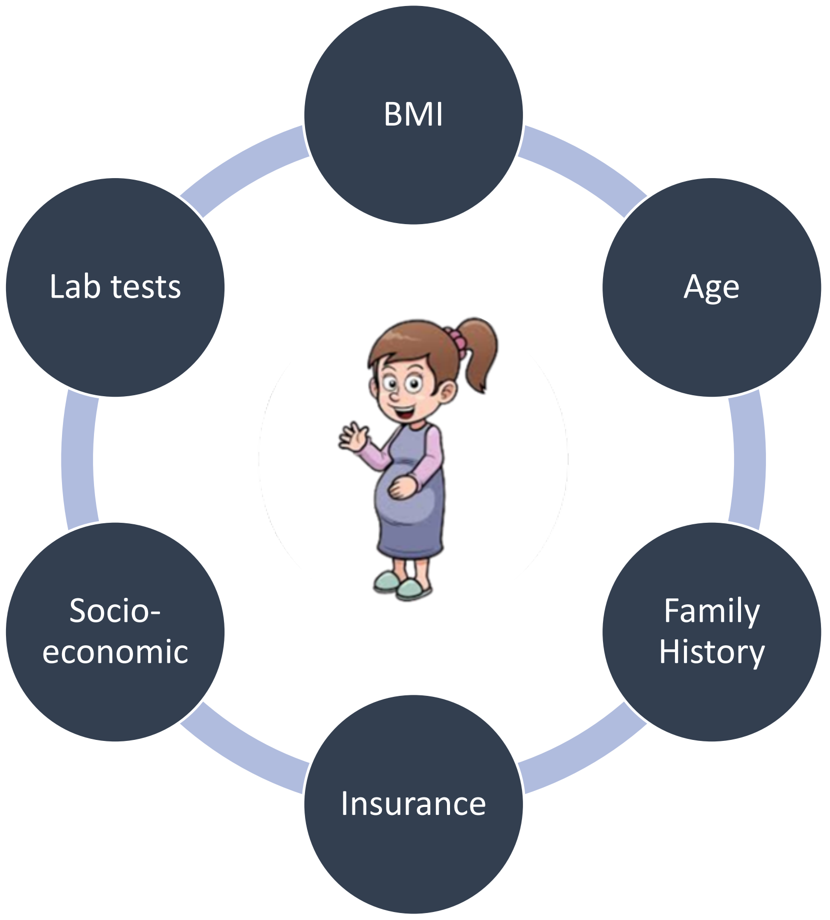
Normalizing flows



Expressive

Probabilistic Inference

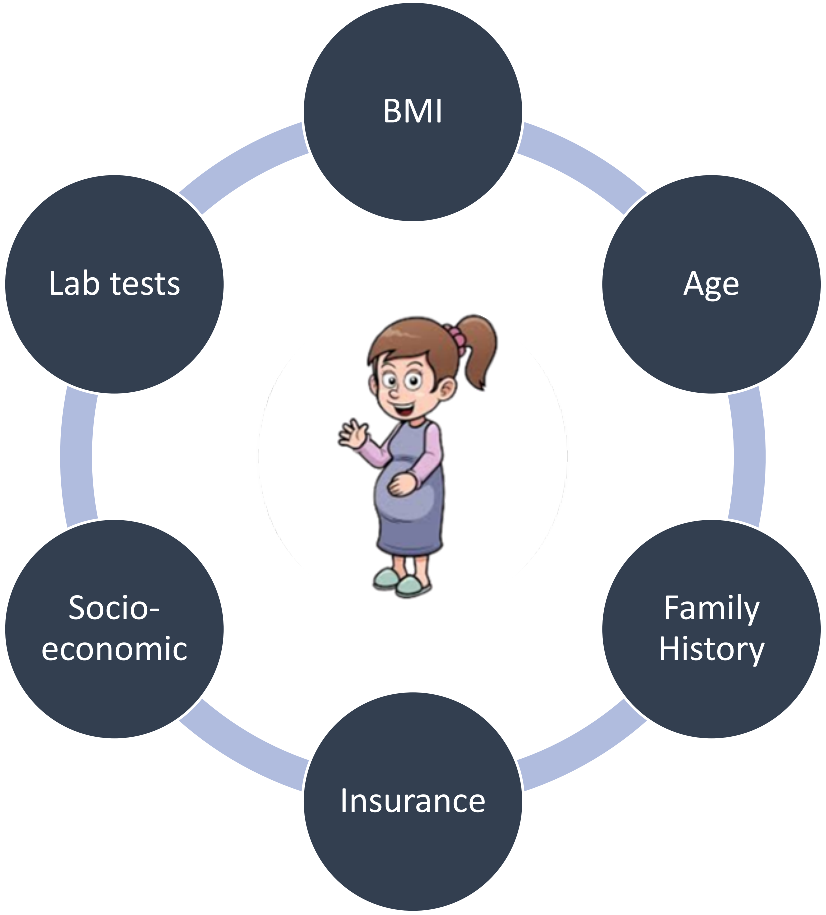
Q1: What is the likelihood of a pregnant woman being over the age of 30, having high BMI, a family history of diabetes and gestational diabetes?



Probabilistic Inference

Q1: What is the likelihood of a pregnant woman being over the age of 30, having high BMI, a family history of diabetes and gestational diabetes?

Q2: Women of what age are most likely to have gestational diabetes?

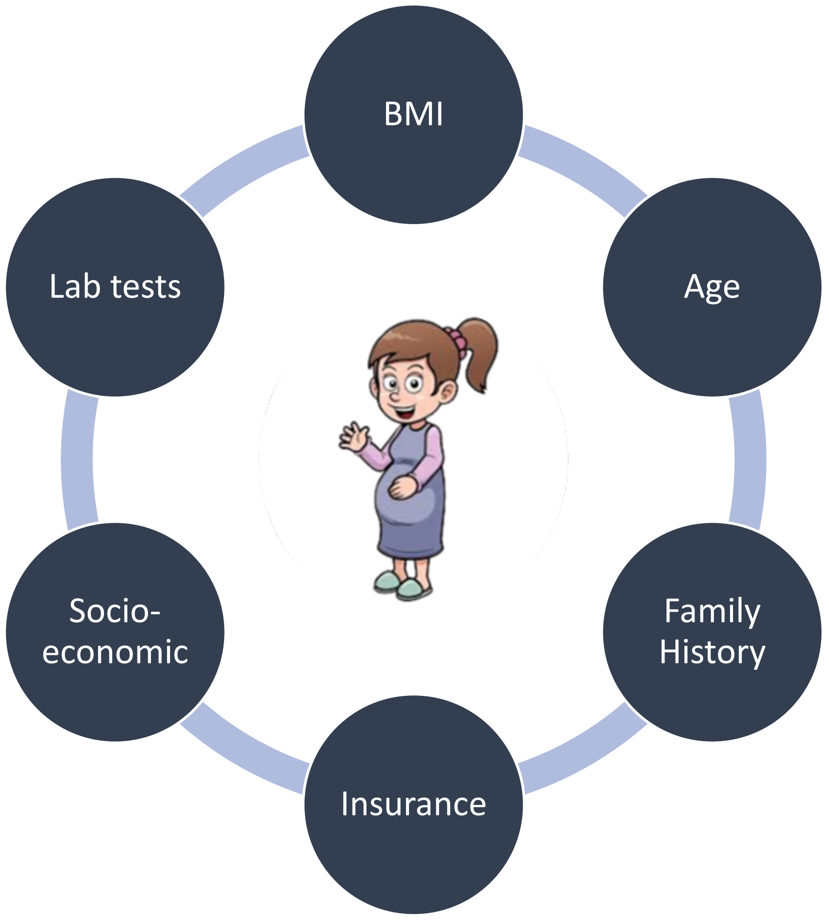


Probabilistic Inference

Q1: What is the likelihood of a pregnant woman being over the age of 30, having high BMI, a family history of diabetes and gestational diabetes?

Q2: Women of what age are most likely to have gestational diabetes?

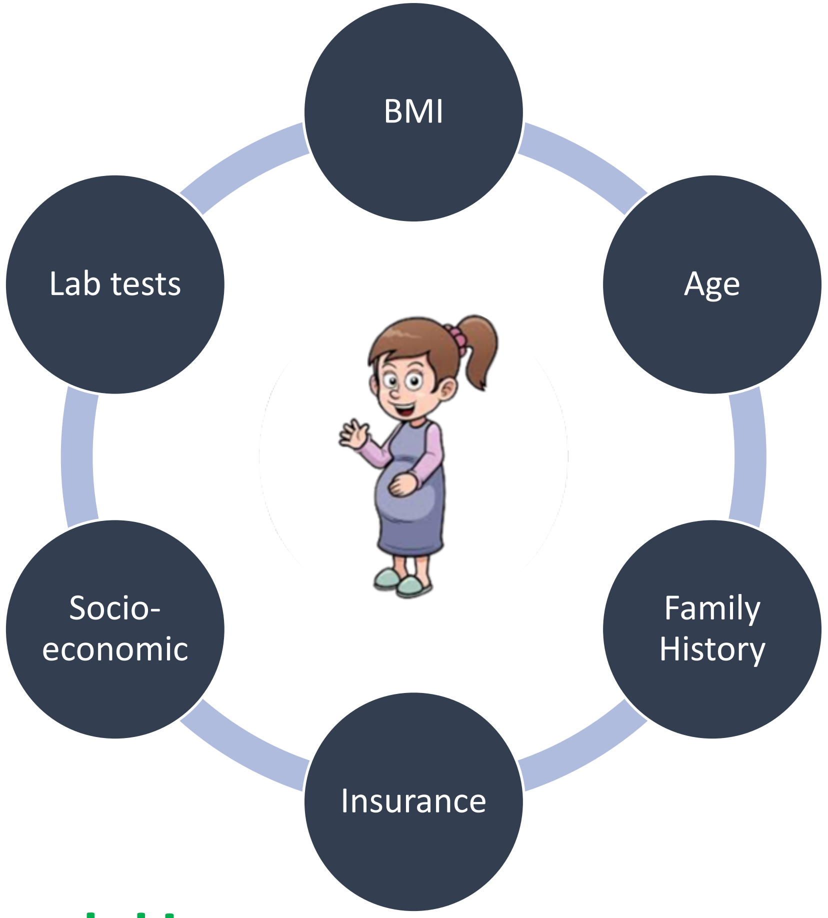
Train a predictive model!



Probabilistic Inference

Q1: What is the likelihood of a pregnant woman being over the age of 30, having high BMI, a family history of diabetes and gestational diabetes?

Q2: Women of what age are most likely to have gestational diabetes?



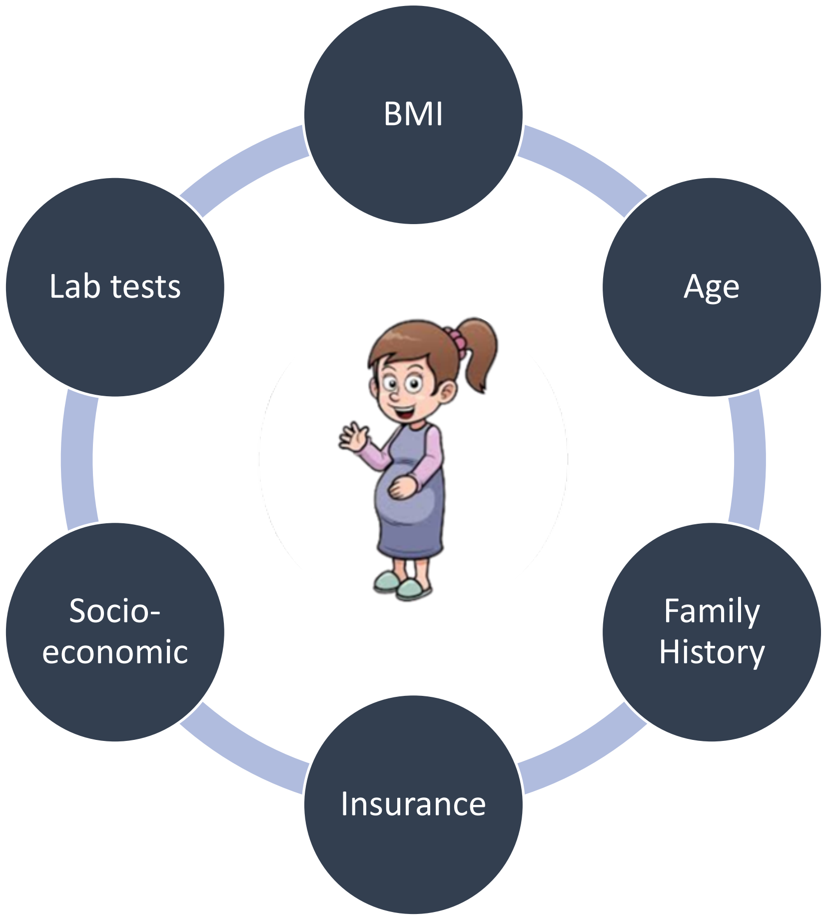
~~Train a predictive model!~~

Learn a probabilistic model of the world!

Probabilistic Inference

Q1: What is the likelihood of a pregnant woman being over the age of 30, having high BMI, a family history of diabetes and gestational diabetes?

$$P(BMI = High, Age = 30, FamilyHist = True, GestD = True)$$

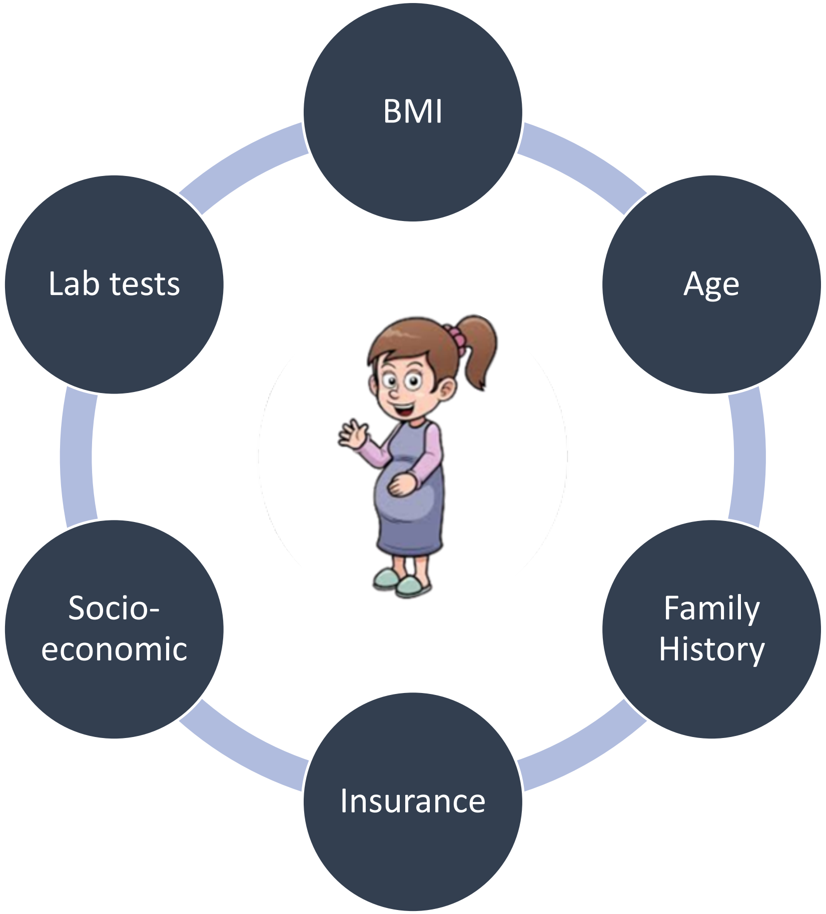


Probabilistic Inference

Q1: What is the likelihood of a pregnant woman being over the age of 30, having high BMI, a family history of diabetes and gestational diabetes?

$$P(BMI = High, Age = 30, FamilyHist = True, GestD = True)$$

Marginals

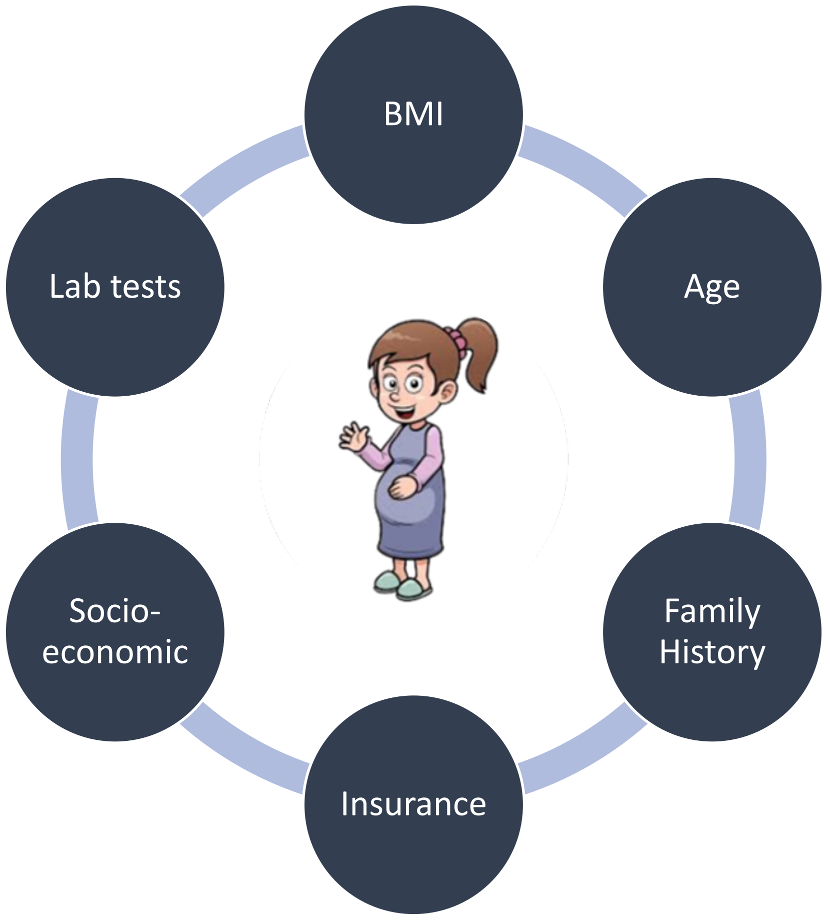


Probabilistic Inference

Q2: Women of what age are most likely to have gestational diabetes?

$$argmax_a P(Age = a, GestD = True)$$

MMAP



What is tractable probabilistic inference?

A class of queries Q is tractable on a family of probabilistic models M *iff for any query* $q \in Q$ and model $m \in M$ *exactly* computing $q(m)$ runs in time $O(poly|m|)$

What is tractable probabilistic inference?

A class of queries Q is tractable on a family of probabilistic models M *iff for any query* $q \in Q$ and model $m \in M$ *exactly* computing $q(m)$ runs in time $O(poly|m|)$

Why tractable inference?

What is tractable probabilistic inference?

A class of queries Q is tractable on a family of probabilistic models M *iff for any query* $q \in Q$ and model $m \in M$ *exactly* computing $q(m)$ runs in time $O(poly|m|)$

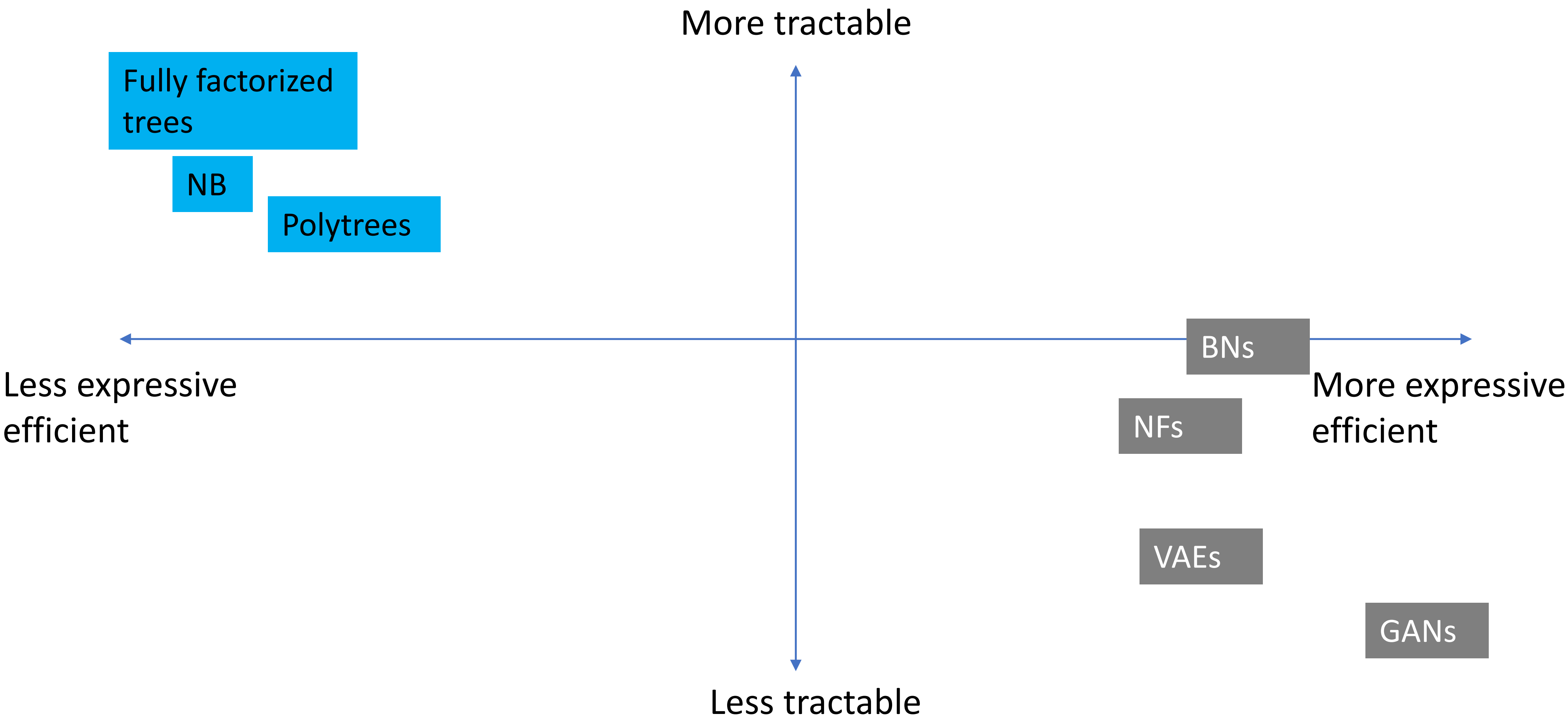
Why tractable inference?

Scalability

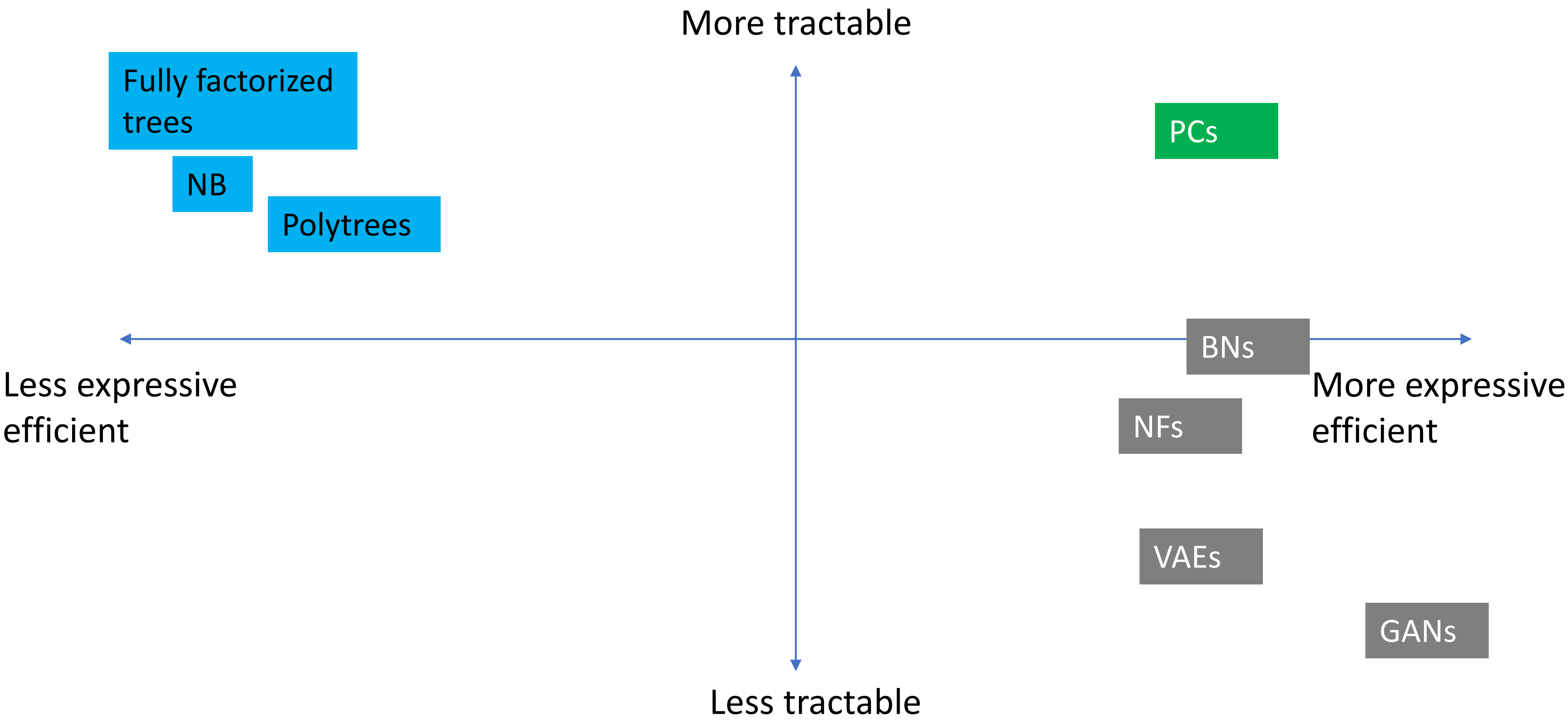
Real-time inference

No need to approximate

Expressive efficiency vs Tractability



Expressive efficiency vs Tractability



Probabilistic Circuits: Outline

1. **Representation**

2. Inference

3. Learning

Probabilistic circuits

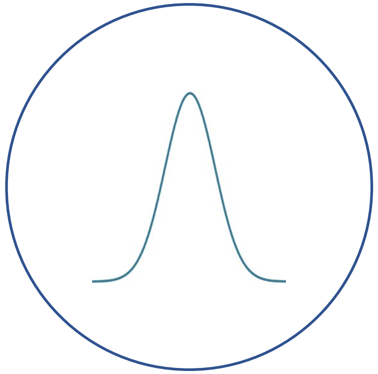
A probabilistic circuit $\boldsymbol{C}$ *over variables* $\boldsymbol{X}$ *is a computational graph encoding a probability distribution* $P(\boldsymbol{X})$

Probabilistic circuits

A probabilistic circuit $\mathcal{C}$ *over variables* $\boldsymbol{X}$ *is a computational graph encoding a probability distribution* $P(\boldsymbol{X})$

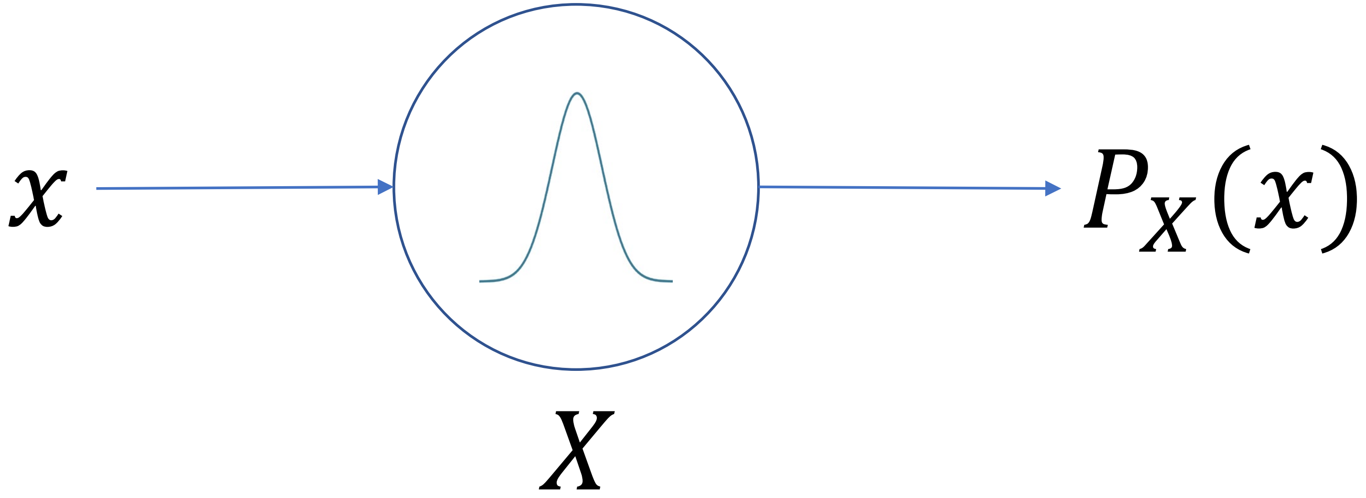
Which computations are allowed?

Leaf nodes



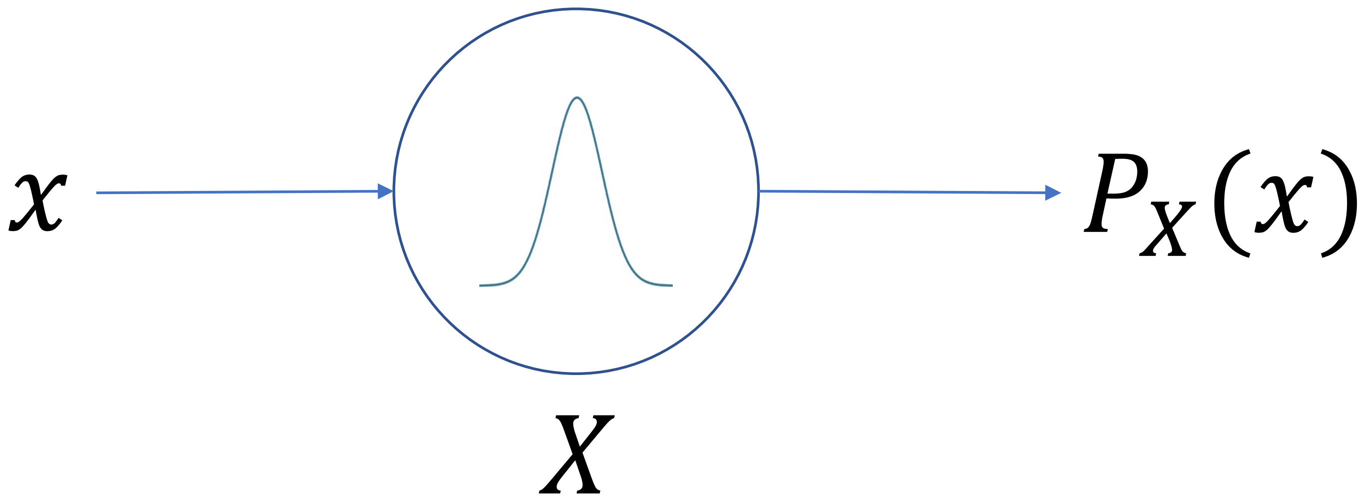
A single node encoding a distribution
E.g.: Gaussian, categorical etc.

Leaf nodes



A single node encoding a distribution
E.g.: Gaussian, categorical etc.

Leaf nodes

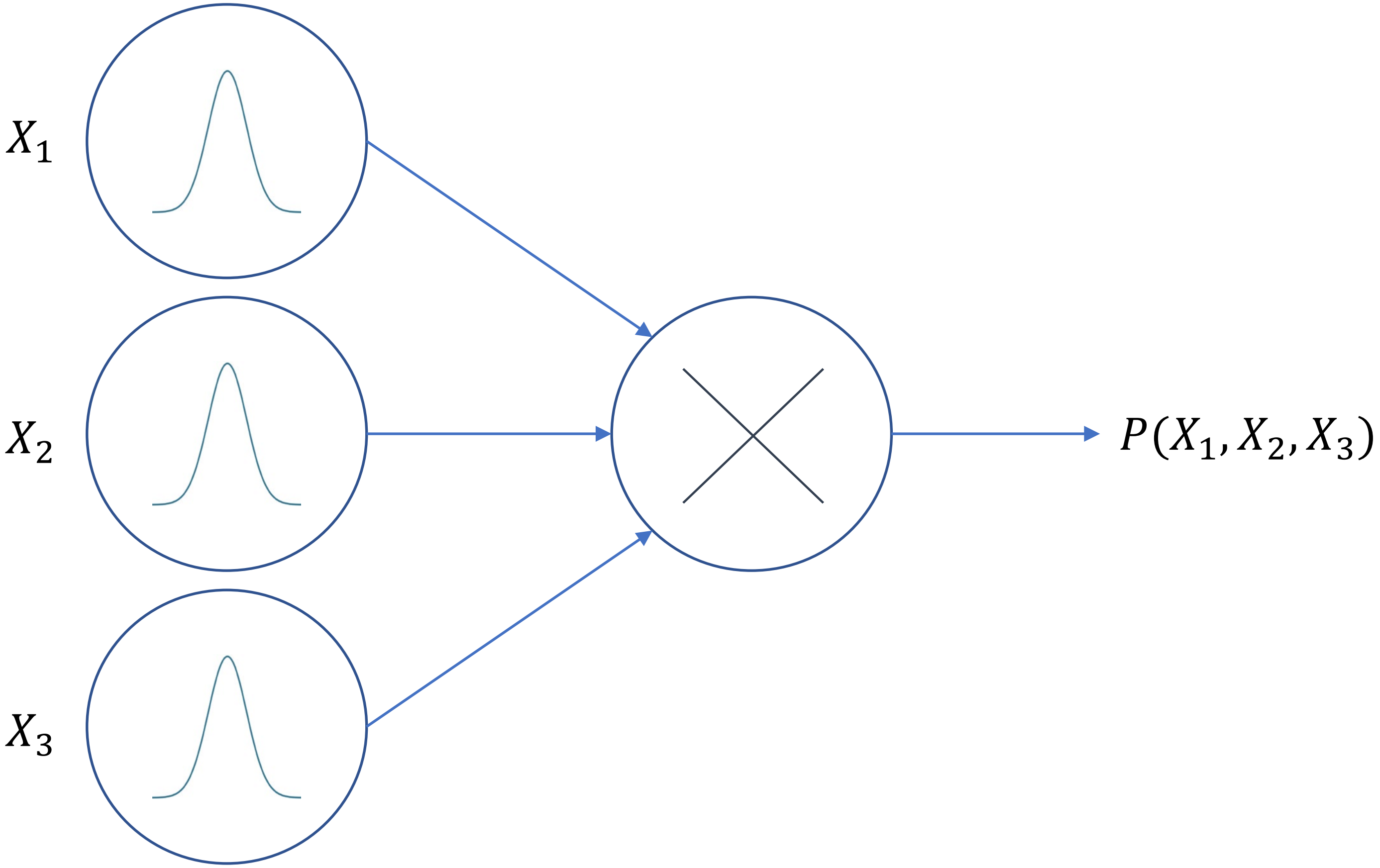


Such simple distributions allow for tractable:

- ❑ Likelihood of full of evidence (EVI)
- ❑ Marginals (MAR)
- ❑ MAP (MAP)

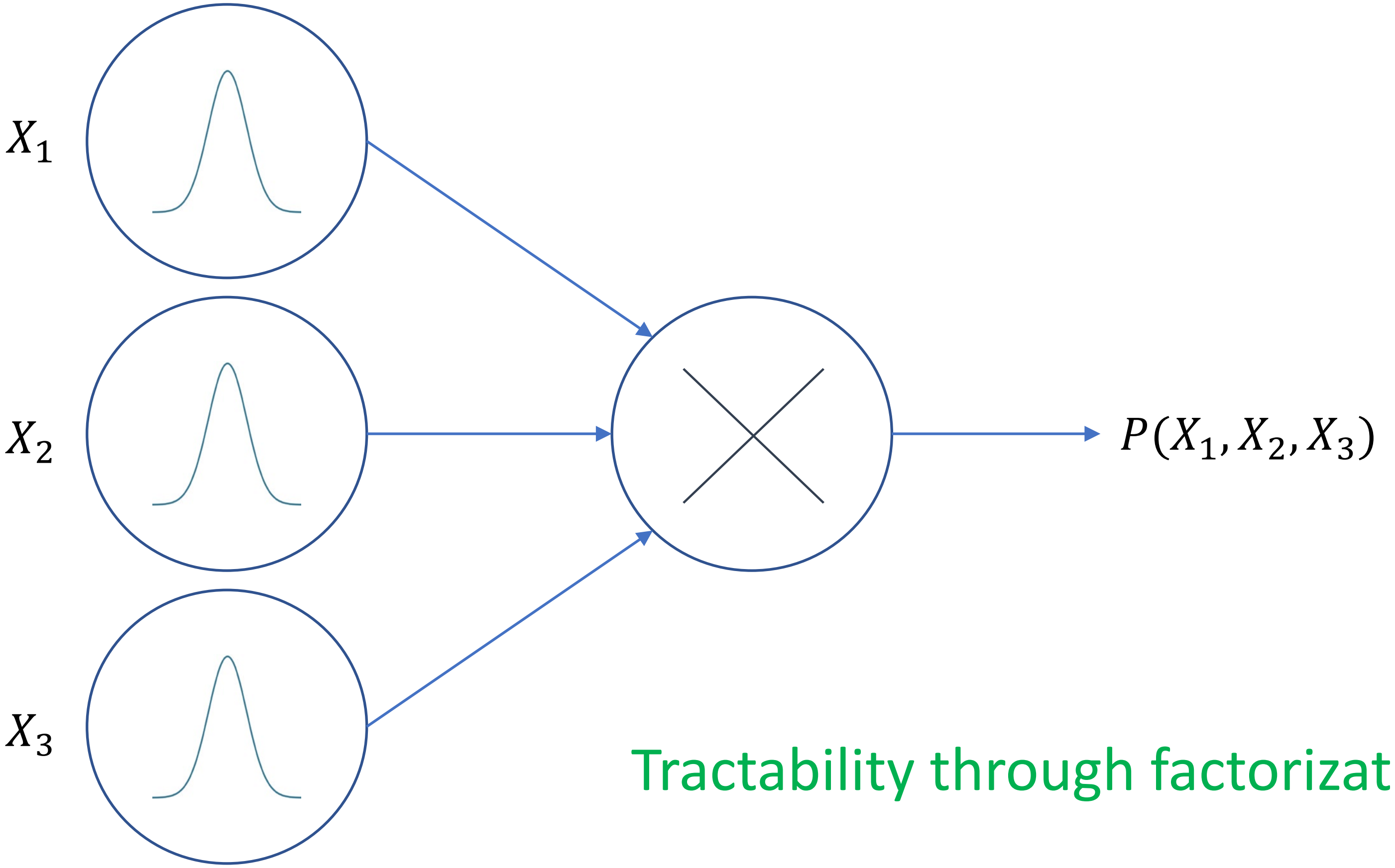
Product nodes

$$P(X_1, X_2, X_3) = P(X_1) * P(X_2) * P(X_3)$$



Product nodes

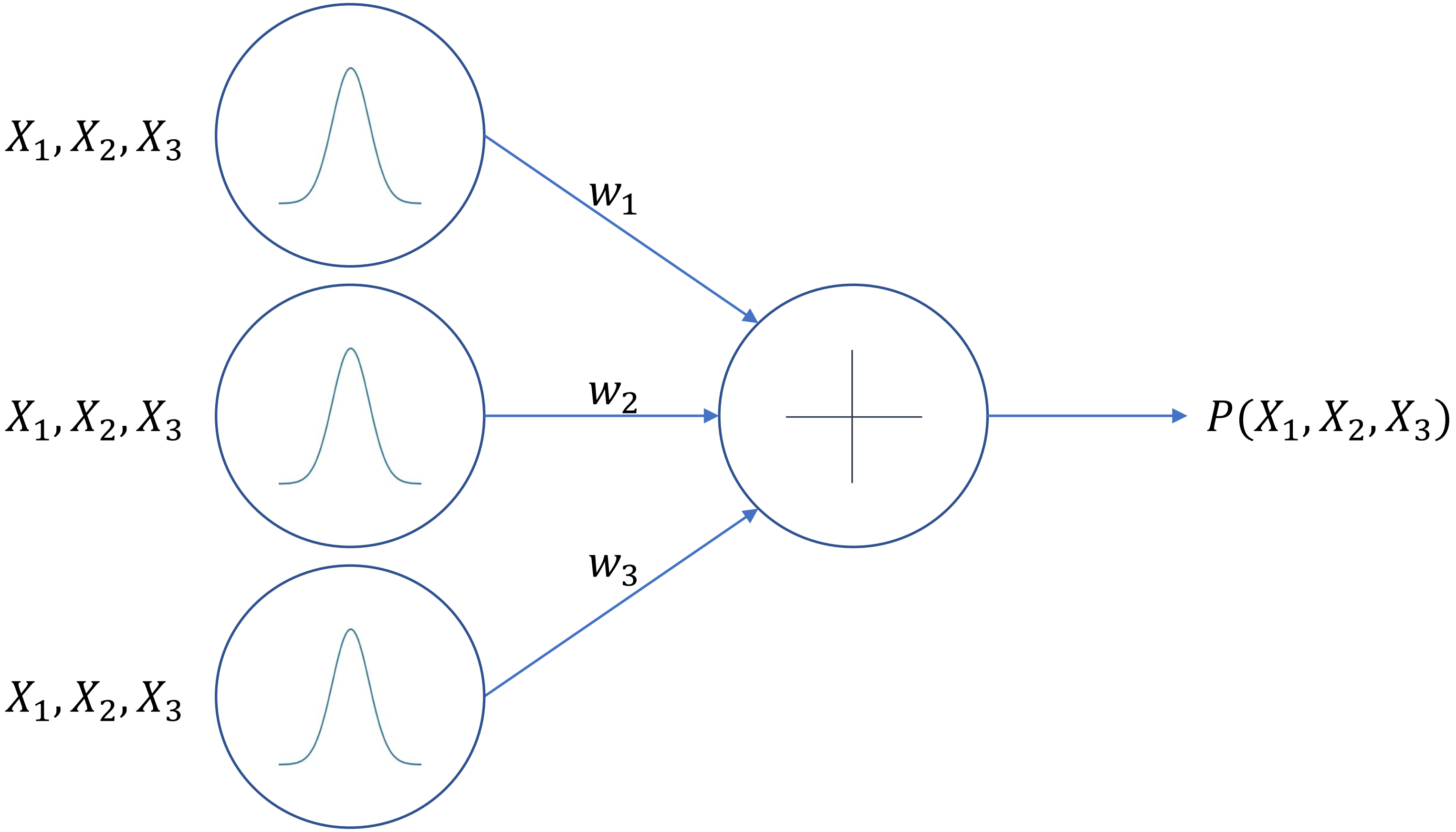
$$P(X_1, X_2, X_3) = P(X_1) * P(X_2) * P(X_3)$$



Tractability through factorization

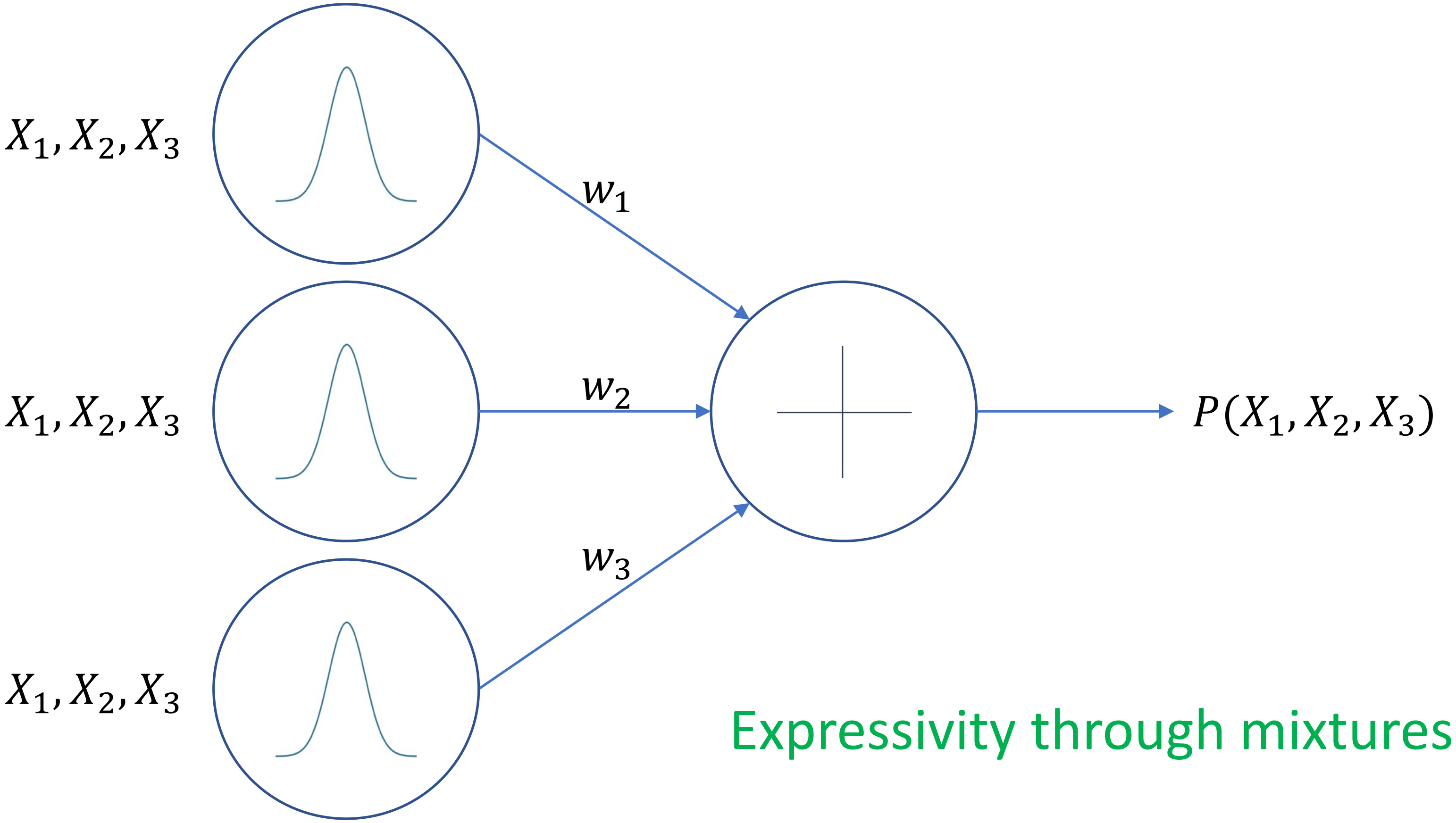
Sum nodes

$$P(X_1, X_2, X_3) = w_1 * P_1(X_1, X_2, X_3) + w_2 * P_2(X_1, X_2, X_3) + w_3 * P_3(X_1, X_2, X_3)$$



Sum nodes

$$P(X_1, X_2, X_3) = w_1 * P_1(X_1, X_2, X_3) + w_2 * P_2(X_1, X_2, X_3) + w_3 * P_3(X_1, X_2, X_3)$$



Expressivity through mixtures

Probabilistic circuits

A probabilistic circuit $\boldsymbol{C}$ *over variables* $\boldsymbol{X}$ *is a computational graph encoding a probability distribution* $P(\boldsymbol{X})$

Any computational graph composed of sums and products?

Probabilistic circuits

A probabilistic circuit $\boldsymbol{C}$ *over variables* $\boldsymbol{X}$ *is a computational graph encoding a probability distribution* $P(\boldsymbol{X})$

~~Any computational graph composed of sums and products?~~

Structural constraints to ensure tractability!

Probabilistic circuits

A probabilistic circuit C *over variables* $\boldsymbol{X}$ *is a computational graph encoding a probability distribution* $P(\boldsymbol{X})$

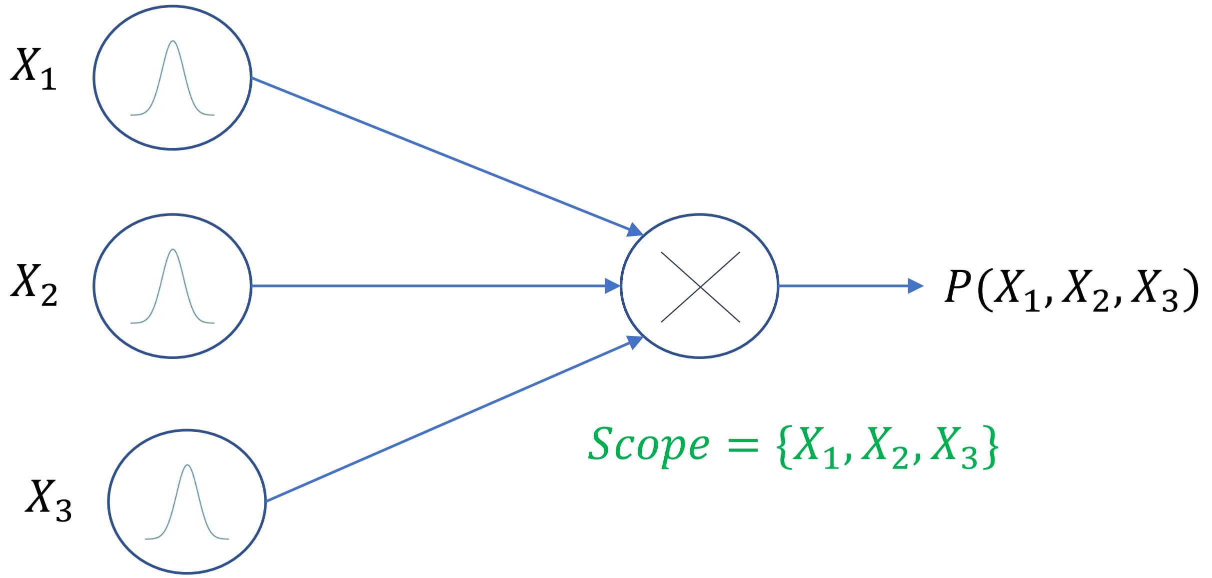
~~Any computational graph composed of sums and products?~~

Structural constraints to ensure tractability!

Which constraints?

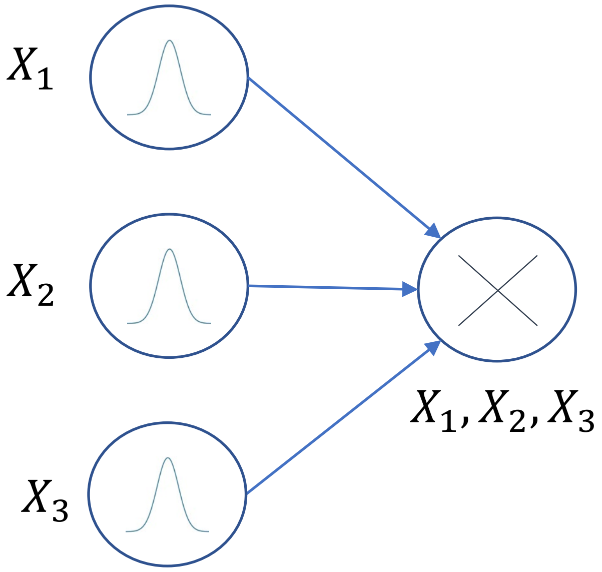
Definitions to build up to structural constraints

Scope: *The scope of a node is the set of all variables in the leaves of the subgraph rooted at that node*
Intuitively, the scope identifies the variables the output of a node "depends" on

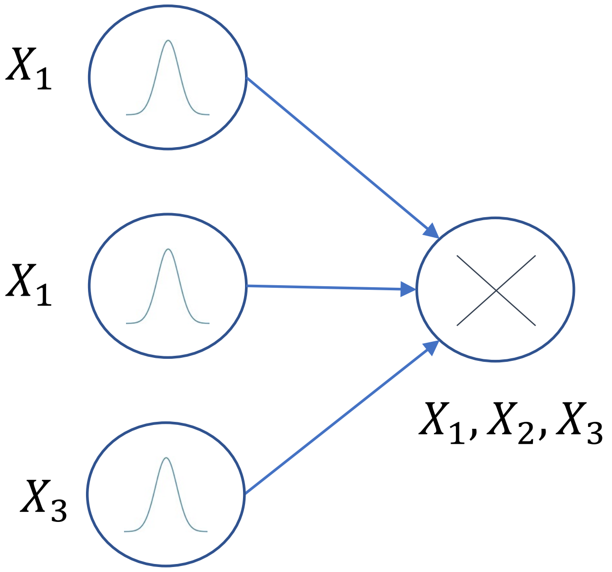


Decomposability

A product node is decomposable if the scope of its children are disjoint
A PC is decomposable if all its product nodes are decomposable



$Decomposable\ circuit$

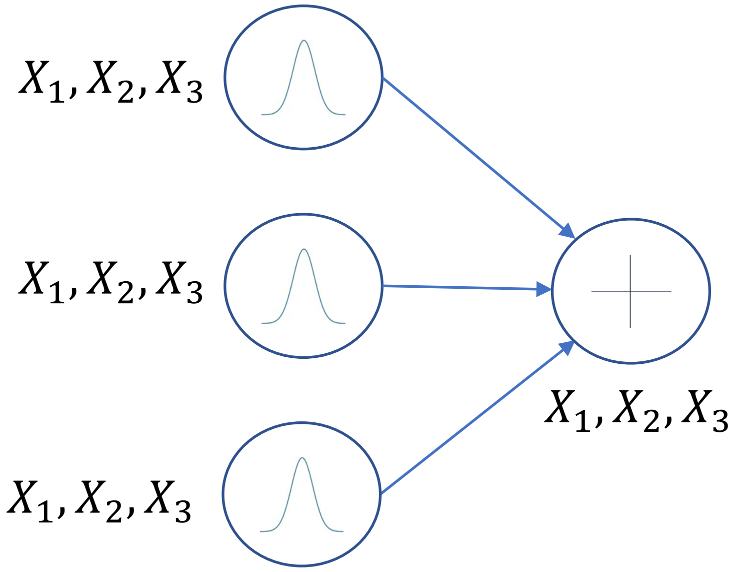


$Non-decomposable\ circuit$

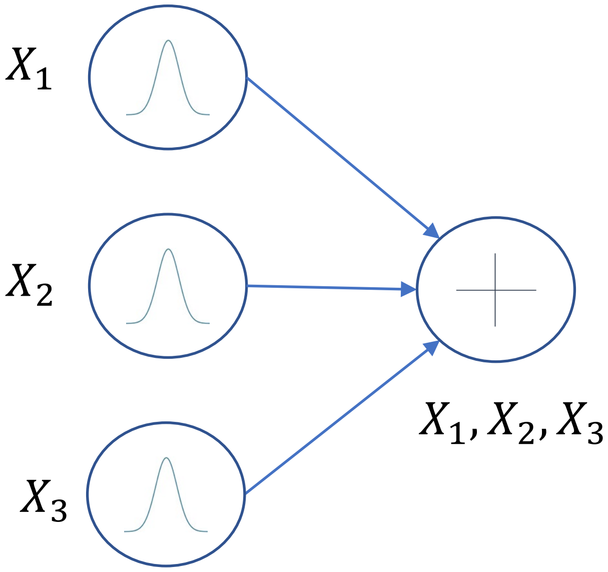
Smoothness

A sum node is smooth if its children have the same scope

A PC is smooth if all its sum nodes are smooth



$Smooth\ circuit$



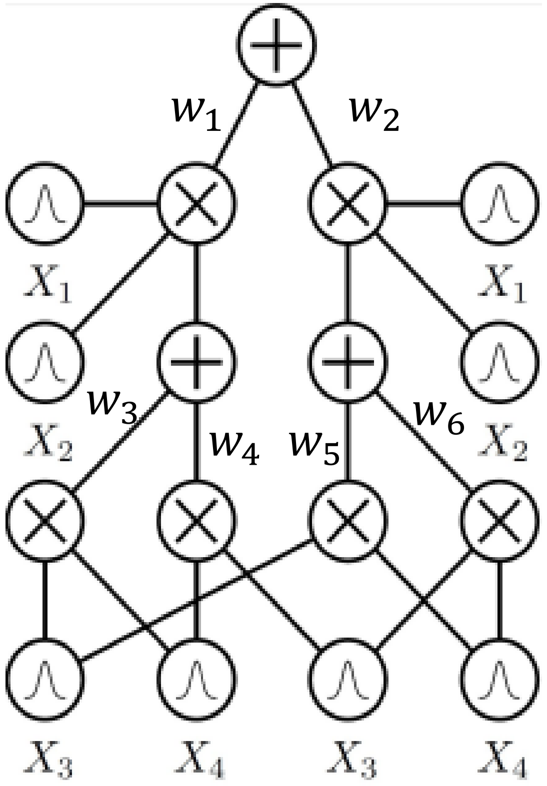
$Non - smooth\ circuit$

Smooth + Decomposable = Tractable MAR

$$P(\boldsymbol{x}) = \sum_i w_i P_i(\boldsymbol{x})$$

$$\int P(\boldsymbol{x}) d\boldsymbol{x} = \int \sum_i w_i P_i(\boldsymbol{x}) d\boldsymbol{x}$$

$$= \sum_i w_i \int P_i(\boldsymbol{x}) d\boldsymbol{x}$$



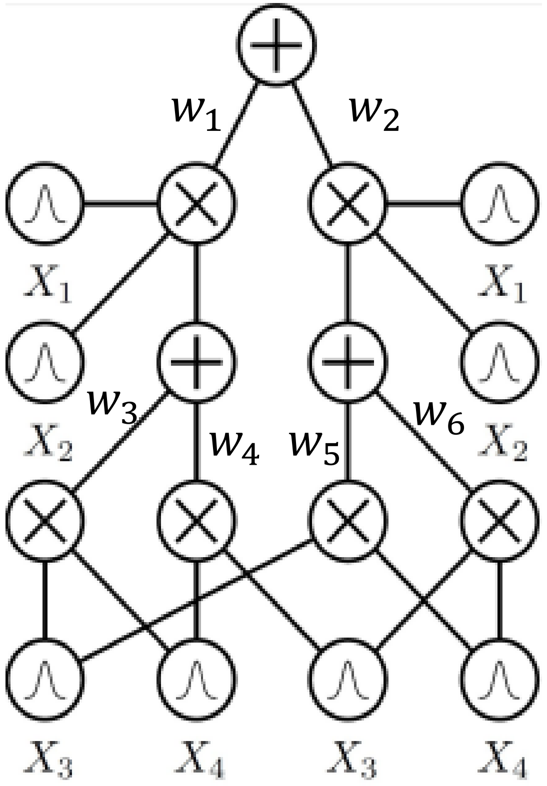
Smooth + Decomposable = Tractable MAR

$$P(\boldsymbol{x}) = \sum_i w_i P_i(\boldsymbol{x})$$

$$\int P(\boldsymbol{x}) d\boldsymbol{x} = \int \sum_i w_i P_i(\boldsymbol{x}) d\boldsymbol{x}$$

$$= \sum_i w_i \int P_i(\boldsymbol{x}) d\boldsymbol{x}$$

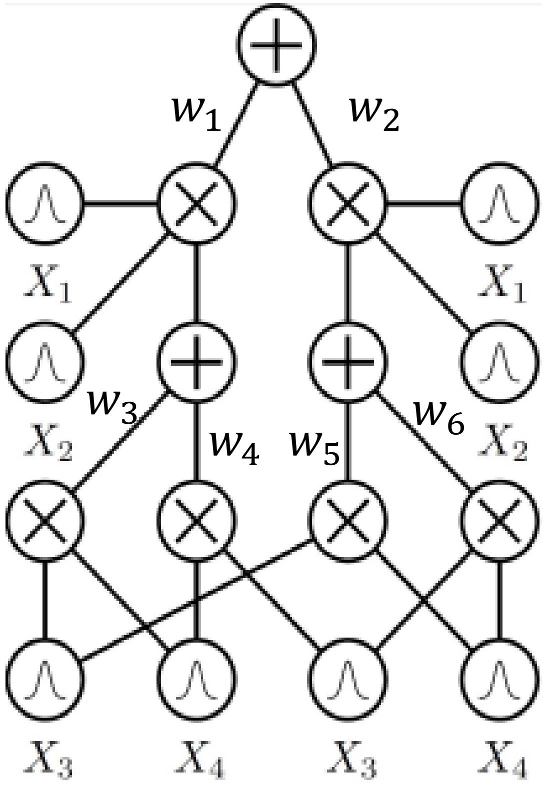
*Integrals at **smooth** sum nodes are "pushed down" to children*



Smooth + Decomposable = Tractable MAR

$$P(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = P(\boldsymbol{x}) * P(\boldsymbol{y}) * P(\boldsymbol{z})$$

$$\begin{aligned} &\int\int\int P(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) d\boldsymbol{x} d\boldsymbol{y} d\boldsymbol{z} \\ &= \int\int\int P(\boldsymbol{x}) P(\boldsymbol{y}) P(\boldsymbol{z}) d\boldsymbol{x} d\boldsymbol{y} d\boldsymbol{z} \\ &= \int P(\boldsymbol{x}) d\boldsymbol{x} \int P(\boldsymbol{y}) d\boldsymbol{y} \int P(\boldsymbol{z}) d\boldsymbol{z} \end{aligned}$$

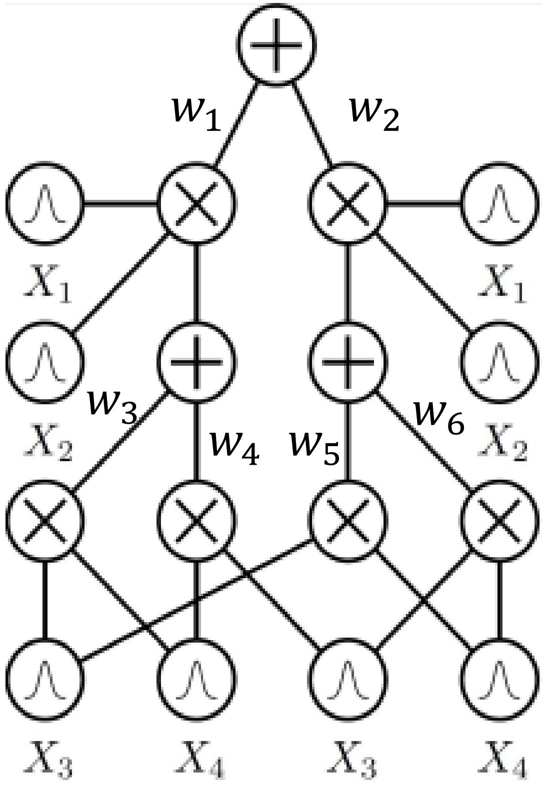


Smooth + Decomposable = Tractable MAR

$$P(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = P(\boldsymbol{x}) * P(\boldsymbol{y}) * P(\boldsymbol{z})$$

$$\begin{aligned} &\int\int\int P(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) d\boldsymbol{x} d\boldsymbol{y} d\boldsymbol{z} = \\ &= \int\int\int P(\boldsymbol{x}) P(\boldsymbol{y}) P(\boldsymbol{z}) d\boldsymbol{x} d\boldsymbol{y} d\boldsymbol{z} = \\ &= \int P(\boldsymbol{x}) d\boldsymbol{x} \int P(\boldsymbol{y}) d\boldsymbol{y} \int P(\boldsymbol{z}) d\boldsymbol{z} \end{aligned}$$

*Integrals at **decomposable** product nodes decompose into simpler ones*

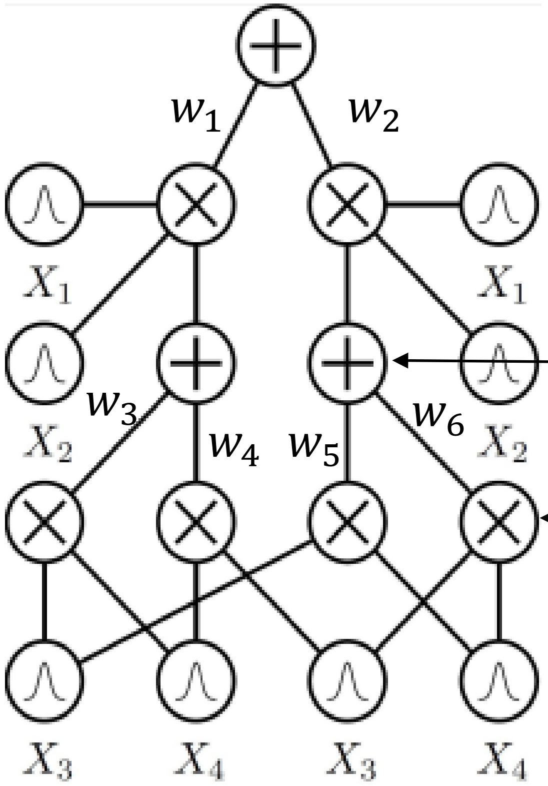


Probabilistic Circuits: Outline

1. Representation
2. **Inference**
3. Learning

Inference: evaluate the computational graph!

EVI: Evaluate the PC bottom-up



Q: What is the probability of the complete evidence $\{x_1, x_2, x_3, x_4\}$

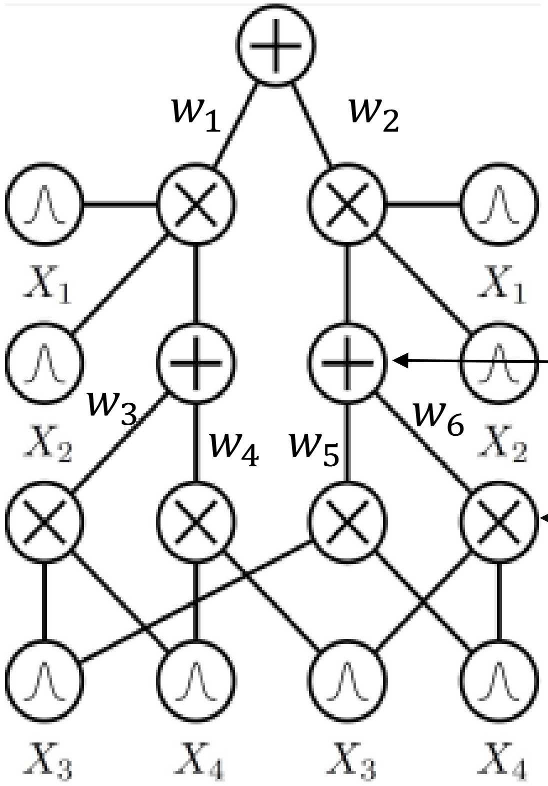
$$P_3(x_3, x_4) = w_5 * P_1(x_3, x_4) + w_6 * P_2(x_3, x_4)$$

$$P_2(x_3, x_4) = P(x_3) * P(x_4)$$

Linear in the size of the circuit!

Inference: evaluate the computational graph!

MAR: Set marginalized leaf distributions to 1 (assuming normalized distributions) and compute output of root node



Q: What is the marginal probability of $\{x_1, x_2, x_3\}$

$P_3(x_3) = w_5 * P_1(x_3) + w_6 * P_2(x_3)$

$P_2(x_3, x_4) = P(x_3) * \int P(x_4) dx_4 = P(x_3) * 1$

Linear in the size of the circuit!

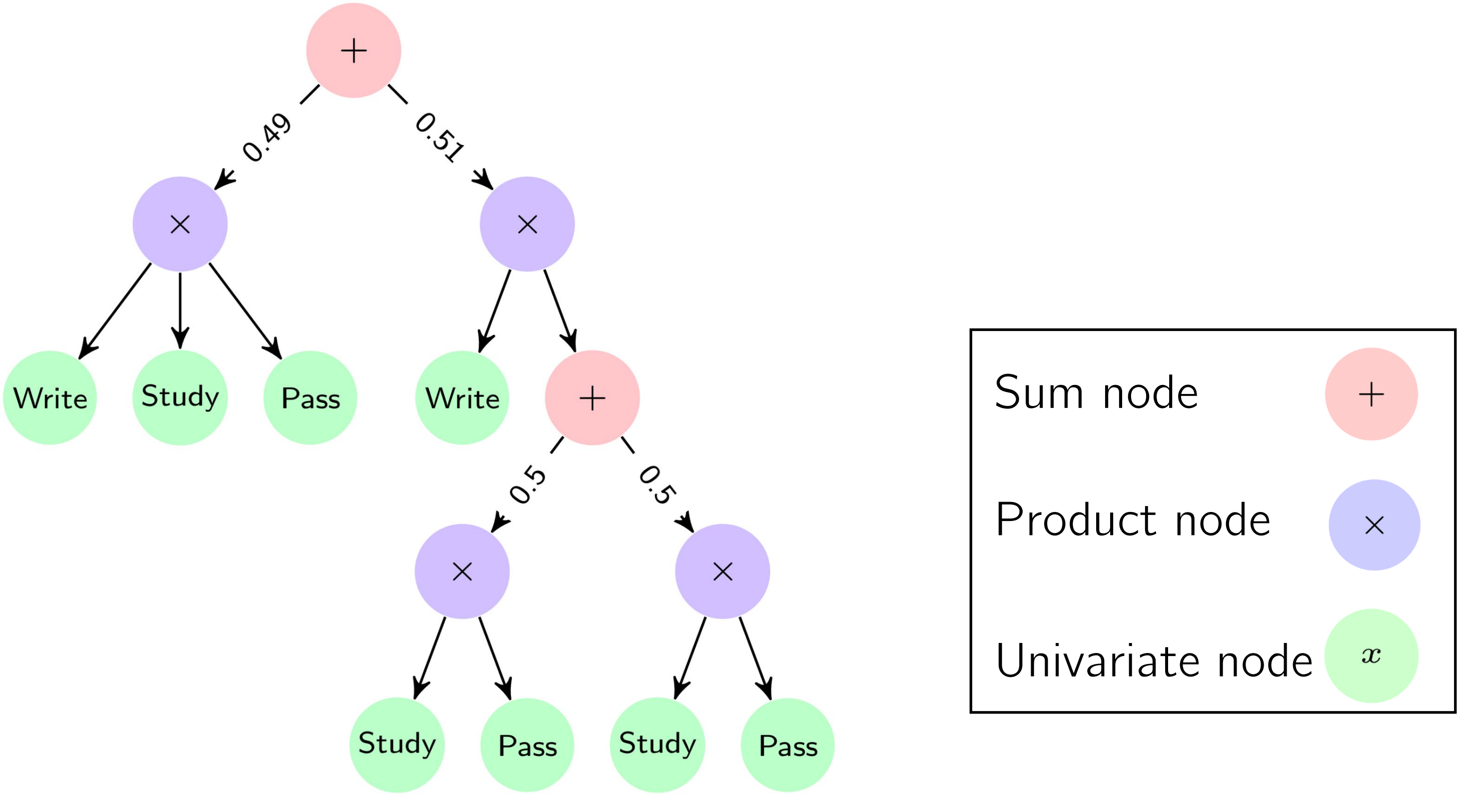
Probabilistic Circuits: Outline

1. Representation

2. Inference

3. **Learning**

Sum-product networks: Smooth & Decomposable PCs

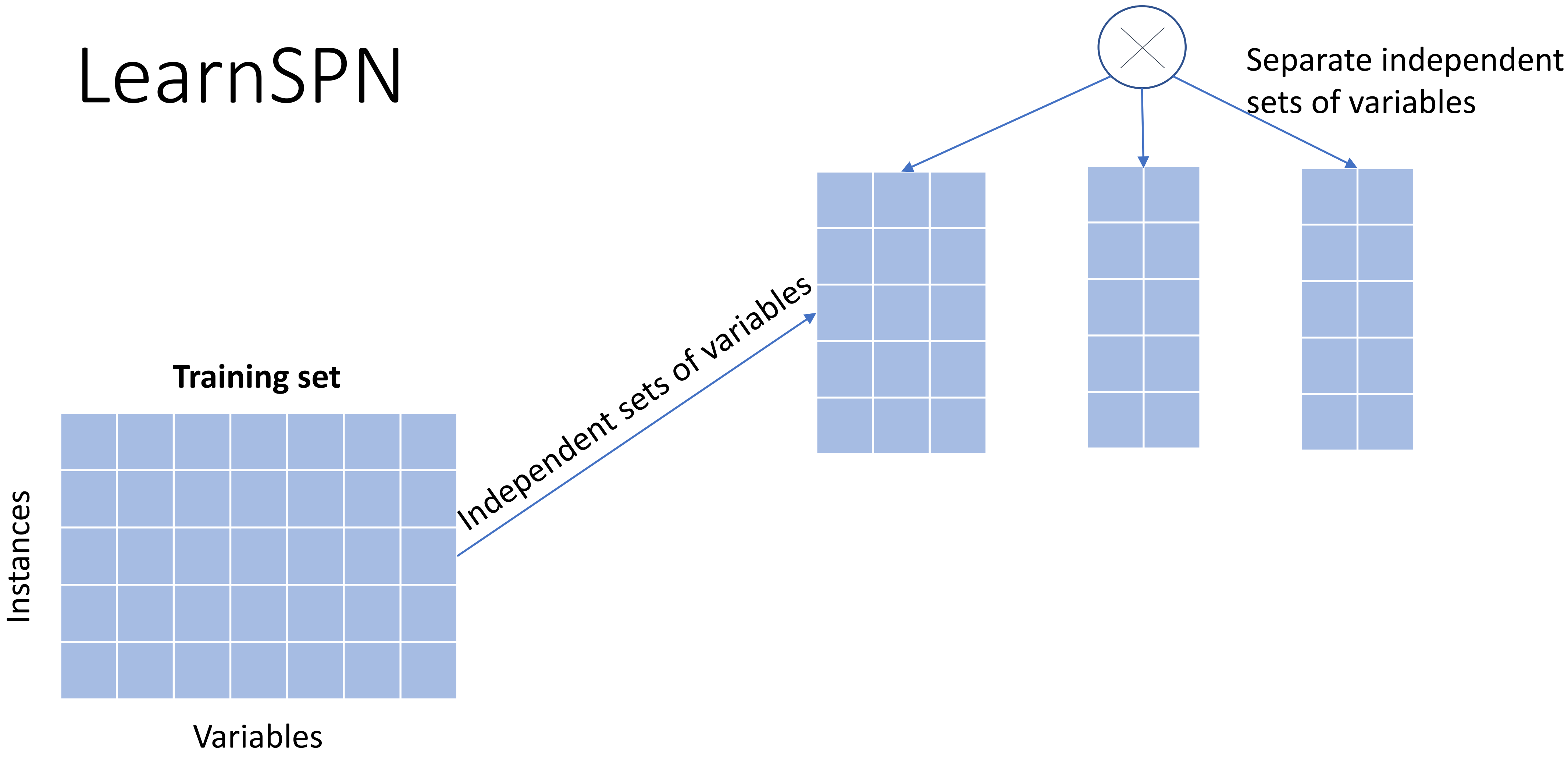


Hoifun Poon and Pedro Domingos, "Sum-product networks: A new deep architecture", Proceedings of the Twenty-Seventh international conference on Uncertainty in artificial intelligence. 2011

Sum-product networks: Learning

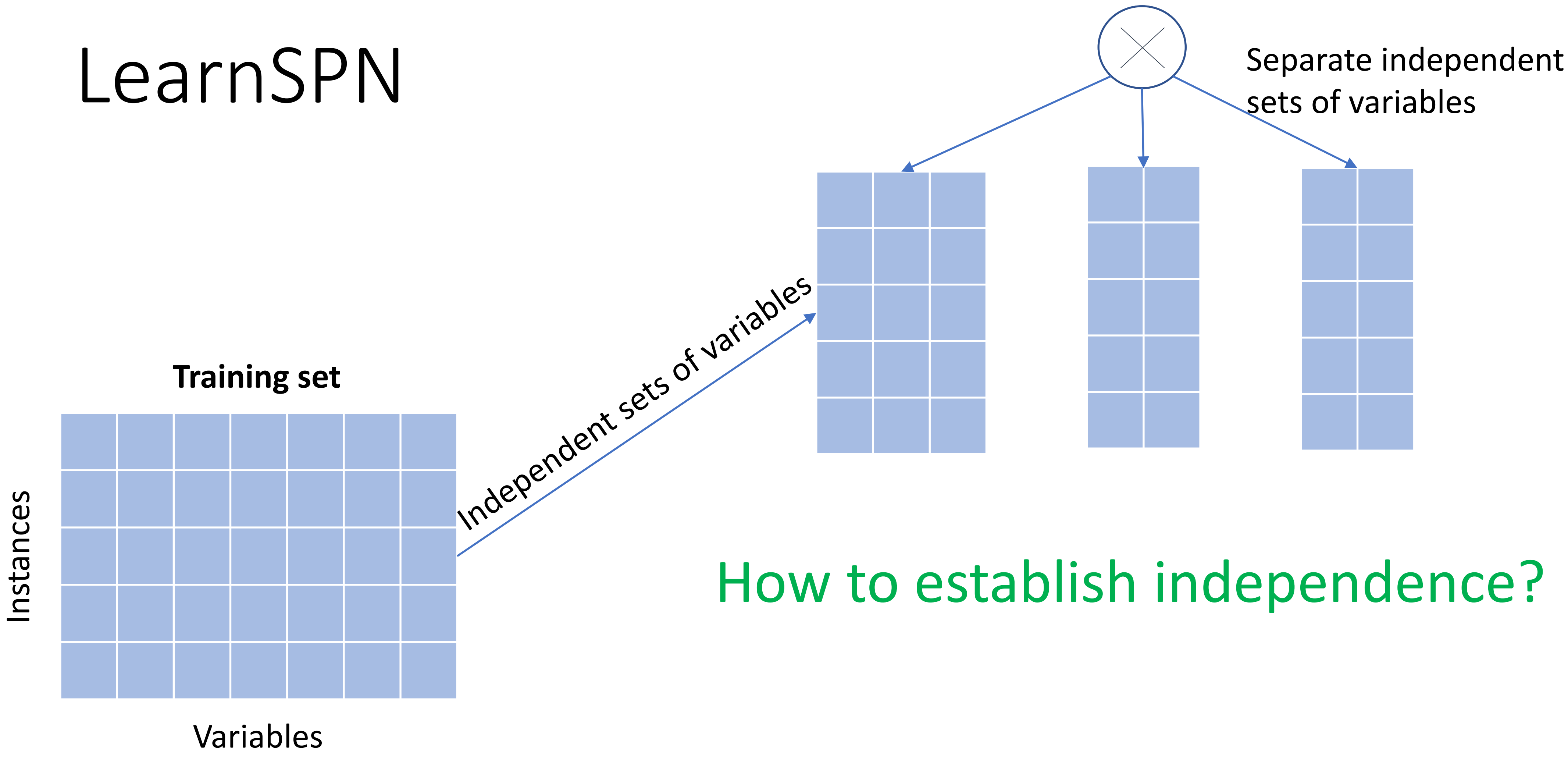
- SPN Learning
  - Parameter learning
    - MLE – SGD, hard GD — *RAT-SPNs, **Einsum Networks***
    - Expectation-Maximization — *Concave-convex procedure (CCCP), hard EM*
    - Approximate Bayesian learning — *Bayesian moment matching, collapsed variational Bayesian inference (CVB-SPN)*
  - Structure learning
    - Top-down learning — ***LearnSPN**, mixed-SPN learning, SVD-SPN*
    - Search & score — *SearchSPN, oSLRAU*
    - Other approaches — *ID-SPN, Bayesian learning of SPNs*

LearnSPN



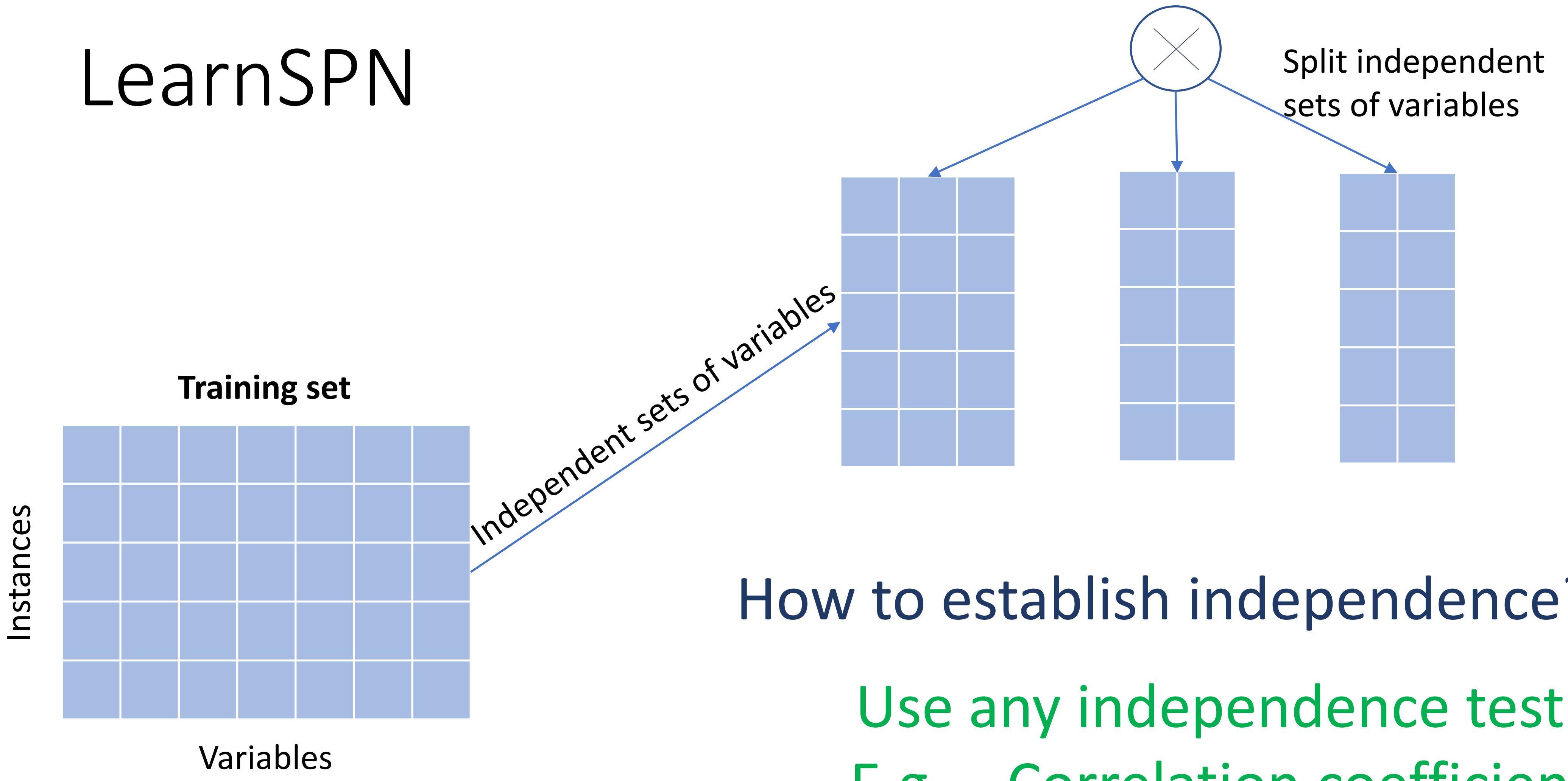
Separate independent sets of variables
Independent sets of variables
Training set
Instances
Variables

LearnSPN



How to establish independence?

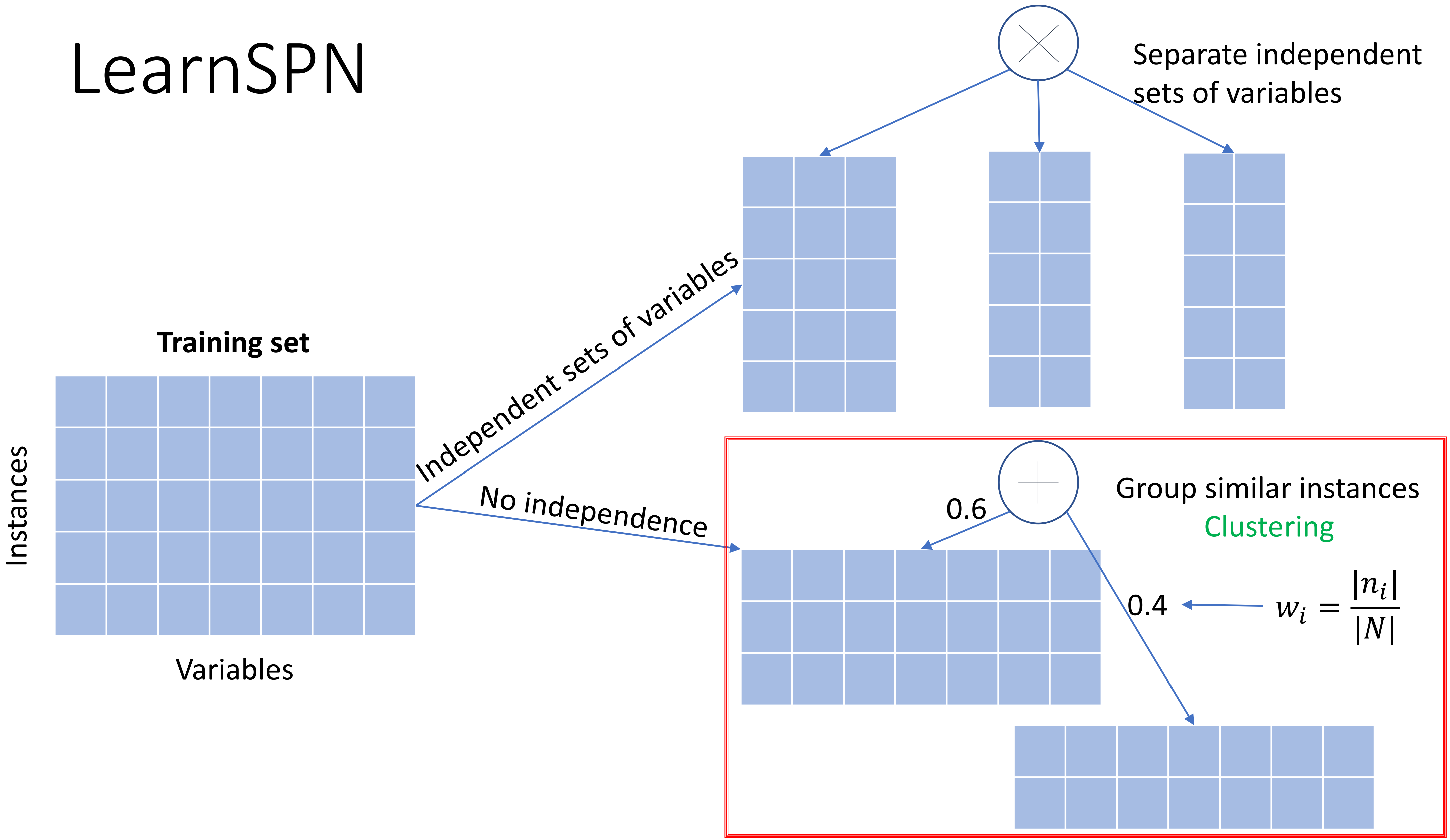
LearnSPN



How to establish independence?

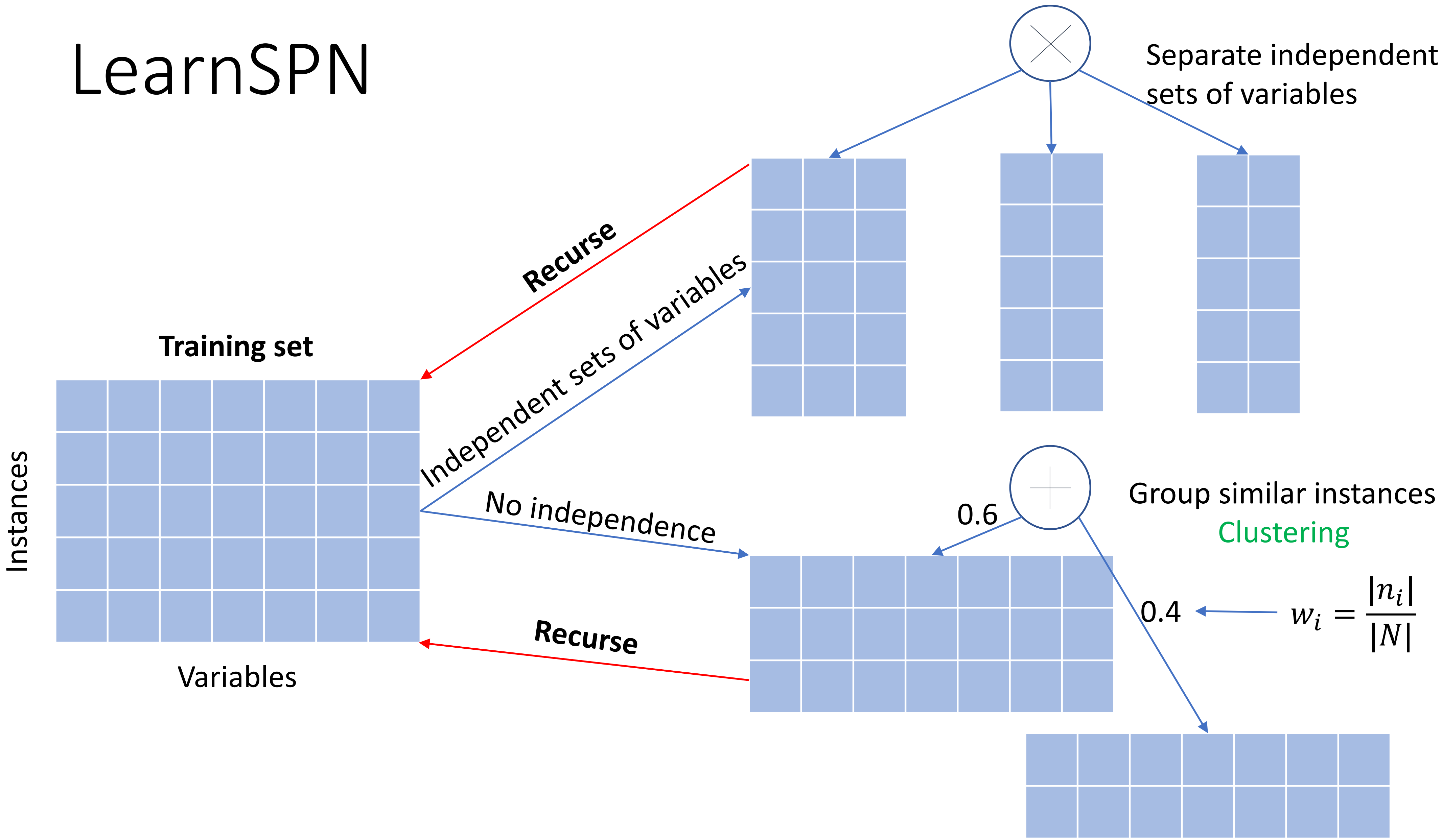
Use any independence test!
E.g. – Correlation coefficient threshold

LearnSPN



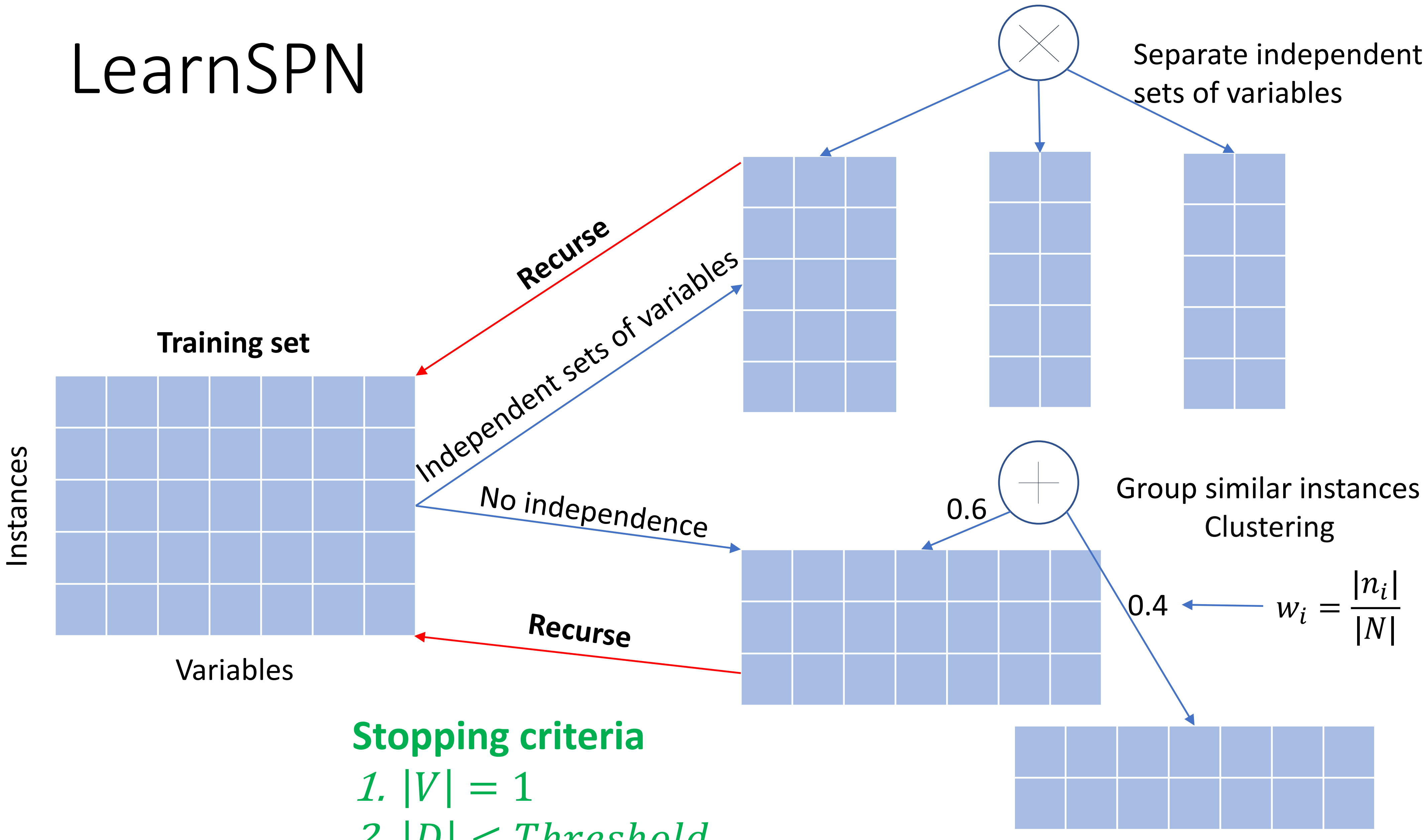
Separate independent sets of variables
Training set
Instances
Variables
Independent sets of variables
No independence
0.6
Group similar instances
Clustering
0.4
$w_i = \frac{|n_i|}{|N|}$

LearnSPN



Separate independent sets of variables
Recurse
Independent sets of variables
Training set
Instances
Variables
No independence
0.6
Group similar instances
Clustering
0.4
$w_i = \frac{|n_i|}{|N|}$
Recurse

LearnSPN



Stopping criteria

1. $|V| = 1$
2. $|D| < Threshold$

Einsum Networks

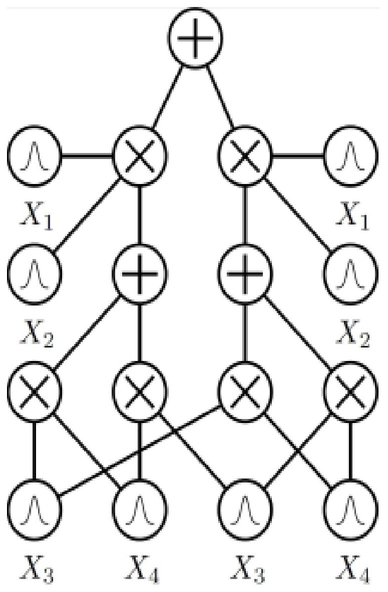
Weaknesses of PCs

- ❑ Highly sparse computational graphs
- ❑ ~ 50 times slower than neural net of comparable size

Einsum Networks

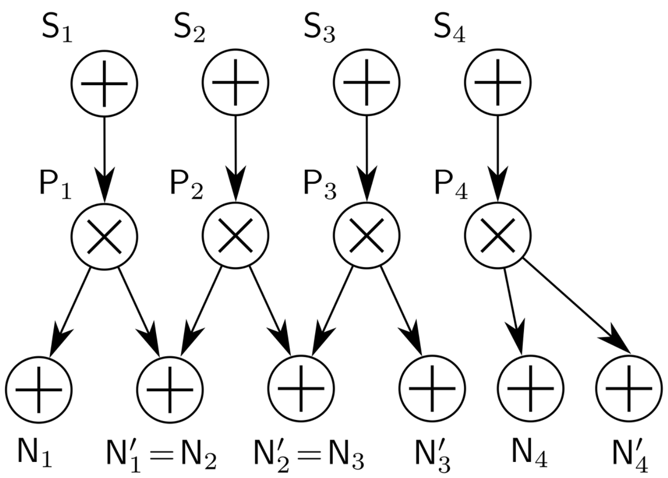
Weaknesses of PCs

- ❑ Highly sparse computational graphs
- ❑ ~ 50 times slower than neural net of comparable size

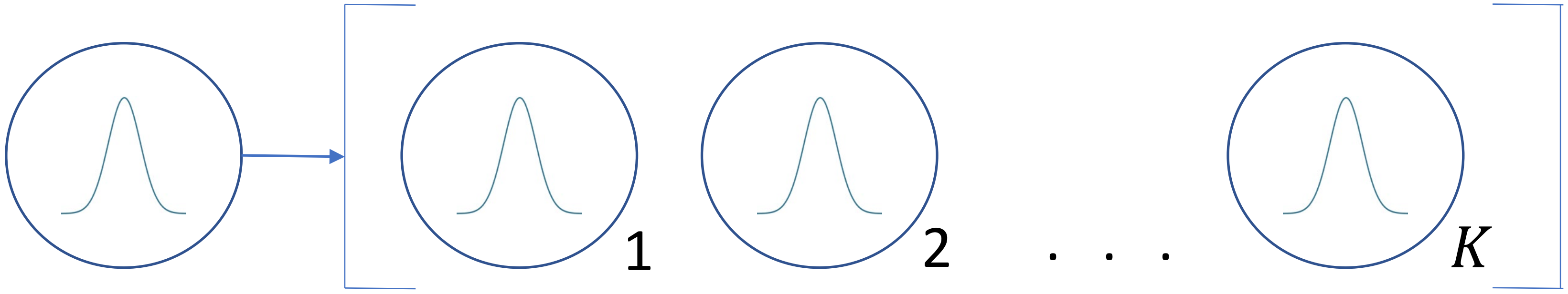


Einsum Networks propose

- ❑ New PC architecture using eisum operations
- ❑ Training and inference up to two orders of magnitude faster
- ❑ Scale PCs to large datasets (CelebA, SVHN)

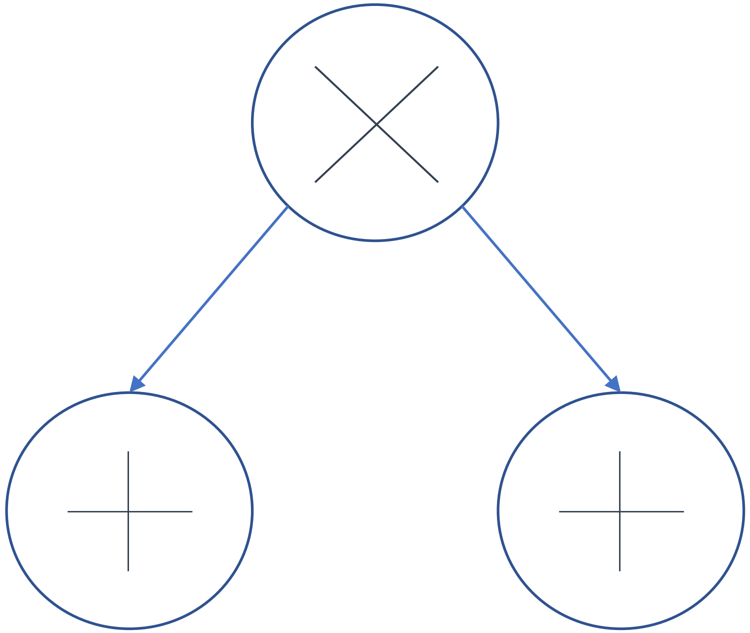


Vectorizing leaf nodes



K *parameterized distributions, each modeling the density of varibles in the scope of that leaf node*

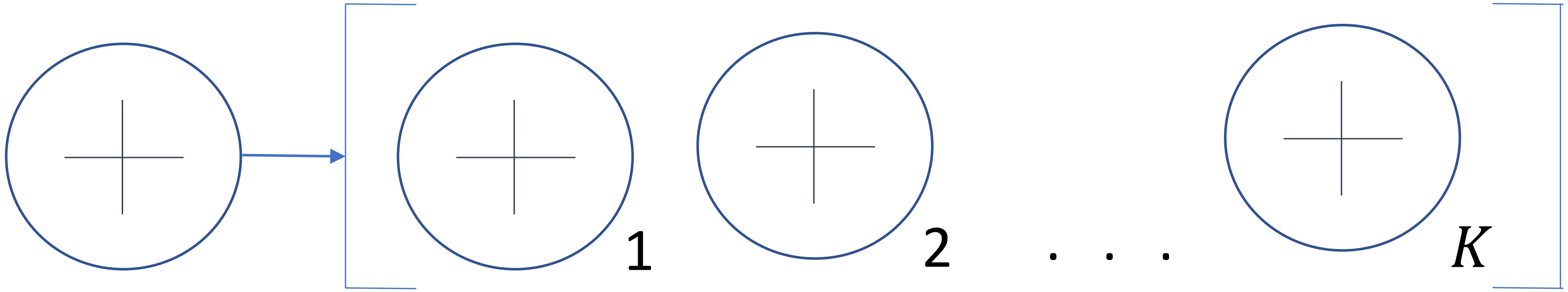
Redefining product nodes



$P \text{ is the outerproduct of its children}$

$P - \text{Matrix of dimension } K \text{ by } K$

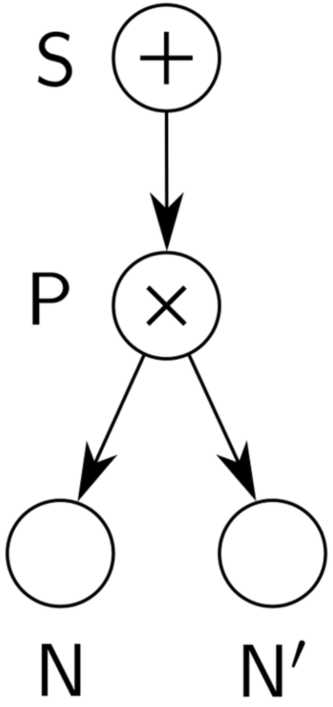
Vectorizing sum nodes

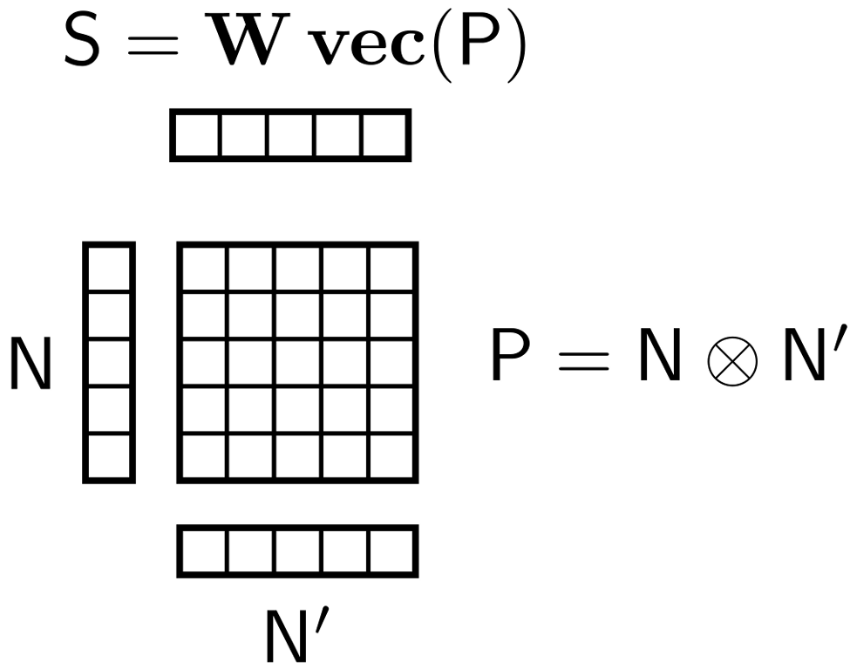


$$S = \boldsymbol{W} vec(P)$$

$$\boldsymbol{W} - Matrix\ of\ dimension\ K X K^2$$

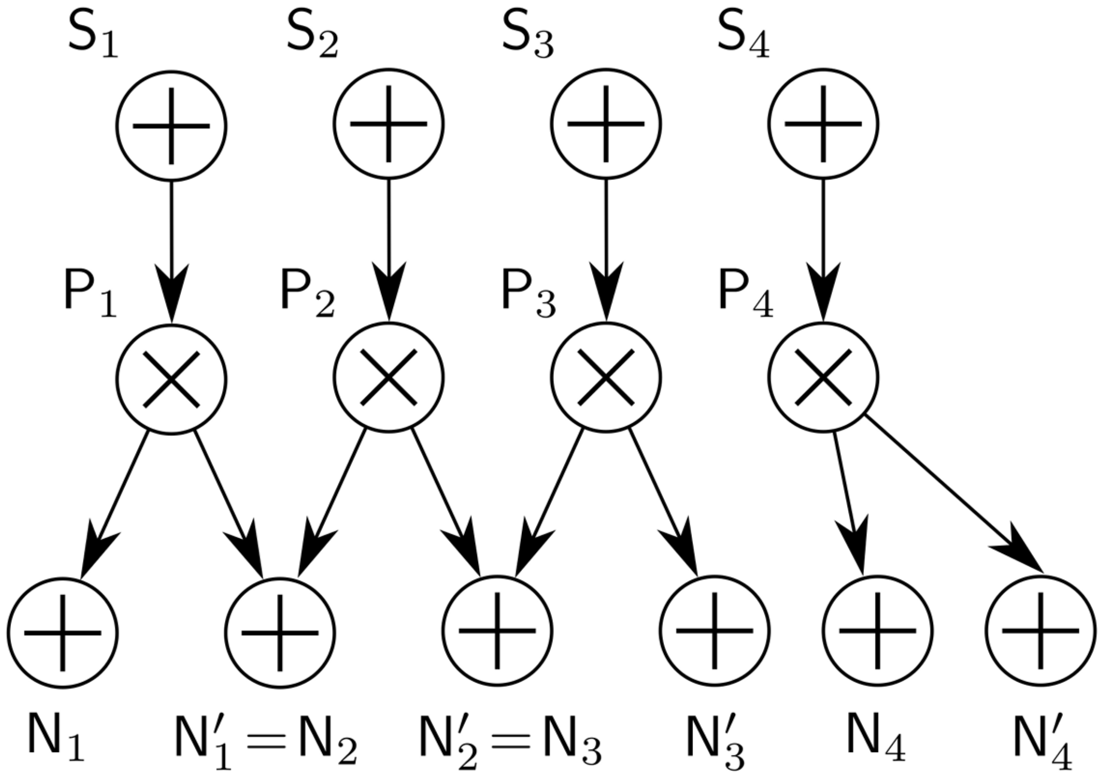
Basic Einsum Operation





$$S_k = \boldsymbol{W}_{kij} N_i N_j{}'$$

Einsum Layers



$\mathbf{S}_{lk} = \mathbf{W}_{lkij} \mathbf{N}_{li} \mathbf{N}'_{lj}$ single `einsum`-operation

Recap

Sum-product network (SPN)

- Nodes $\{ +, \times, x \}$
- Smooth, Decomposable
- $+$ introduces latent variables
- Universal density approximator
- MAR tractable
- Mixture is SPN

Hands-On Demo

bit.ly/tpm-day2-pc1

bit.ly/tpm-day2-pc2