Bayesian Learning of Sum-Product Networks

Martin Trapp, Robert Peharz, Hong Ge, Franz Pernkopf, Zoubin Ghahramani

Presented by Shuaijun Gao, ETH Zurich

Agenda

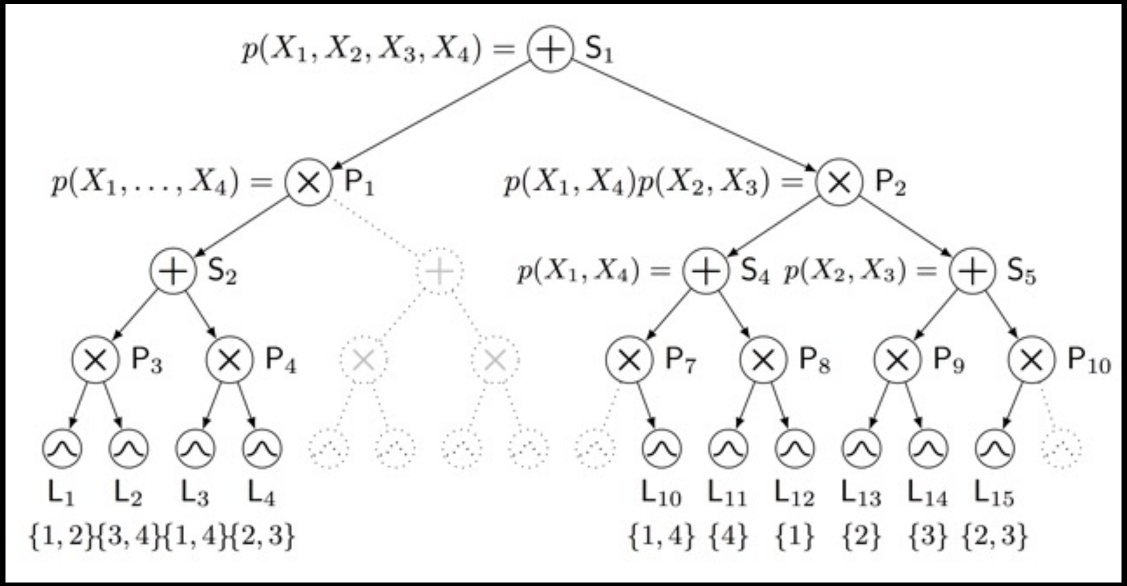
1. What is Sum-Product Network(SPN)?
   1. Motivation
   2. Features

2. Parameter Learning and Structure Learning
   1. Parameter Learning
   2. Structure Learning

3. Bayesian Learning of SPN
   1. Update parameters
   2. Update structure

4. Experiments

Motivation of SPN

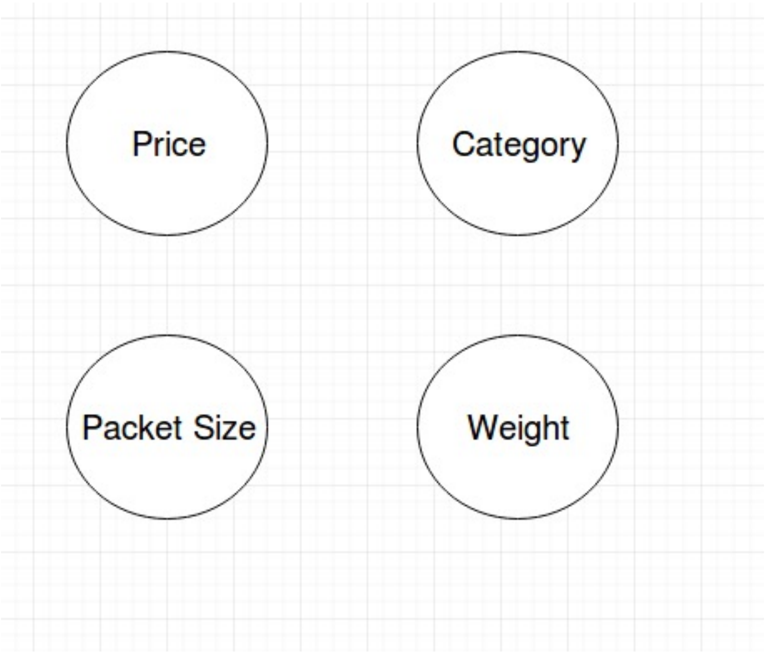


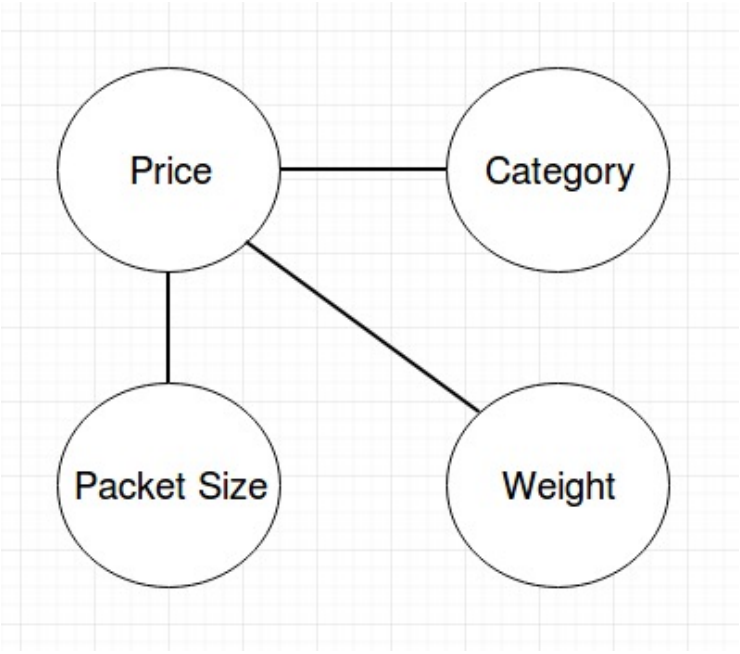
Problems that SPN try to solve

Name	Price	Category	Packet Size	Weight
Notebook	$$$	tech	m	heavy
Docking Station	$$	tech	s	light
Monitor	$$	tech	xl	heavy
Smartphone	$$	tech	s	light
Star Wars Shirt	$	clothes	m	light
Light sabor	$	stuff	s	light
Lego Star Wars	$$$	stuff	m	heavy

Name	Price	Category	Packet Size	Weight	P(Joe buys it)
Graphics Card	$$$	tech	m	light	???
Star Wars Fan Art	$$	stuff	xl	heavy	???

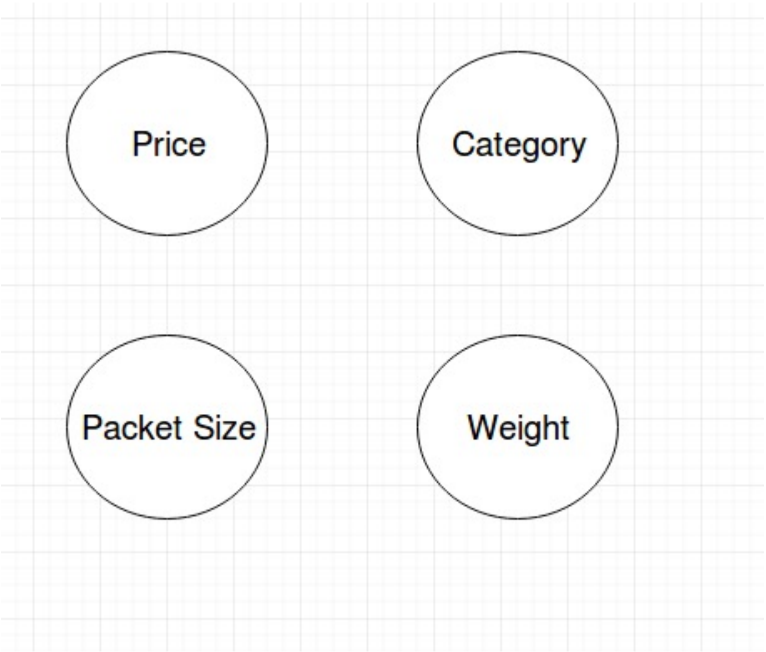
Problems that SPN try to solve

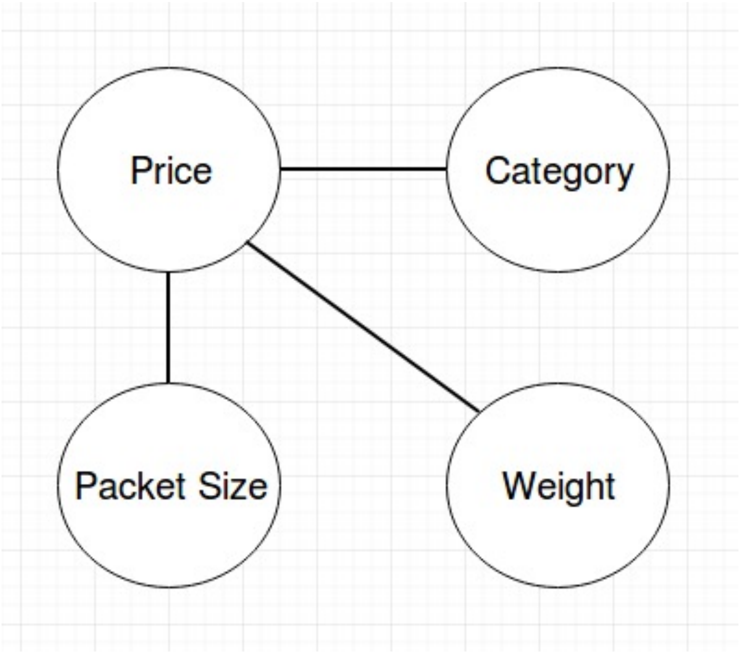




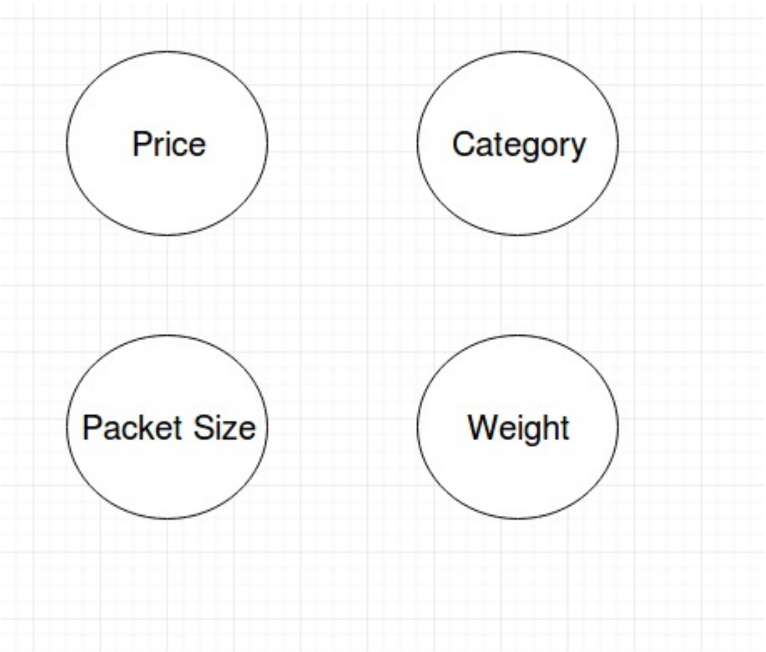
$$P(Buy|P = \$10, Cat = tech, PS = m, W = h) = \frac{P(P = \$10, Cat = tech, PS = m, W = h|Buy) * P(Buy)}{\sum P(P = \$10, Cat = tech, PS = m, W = h|\bullet) * P(\bullet)}$$

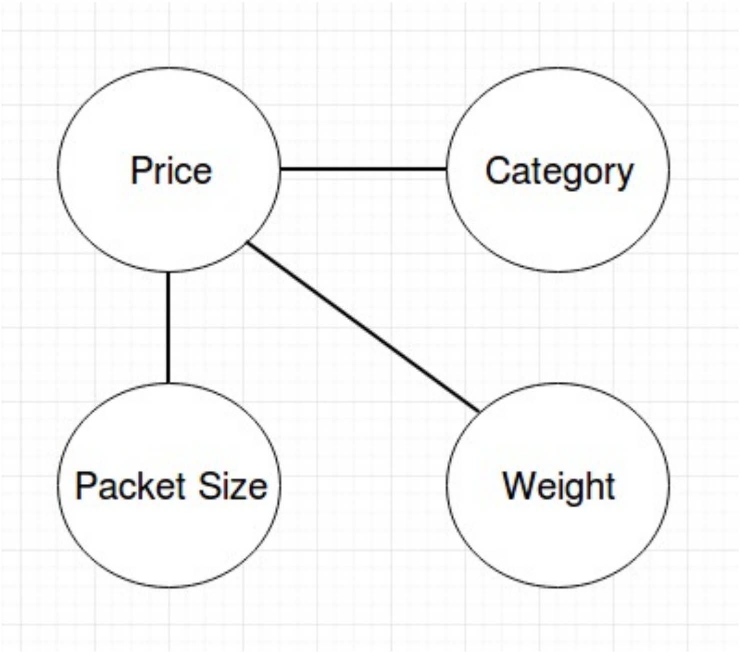
Problems that SPN try to solve





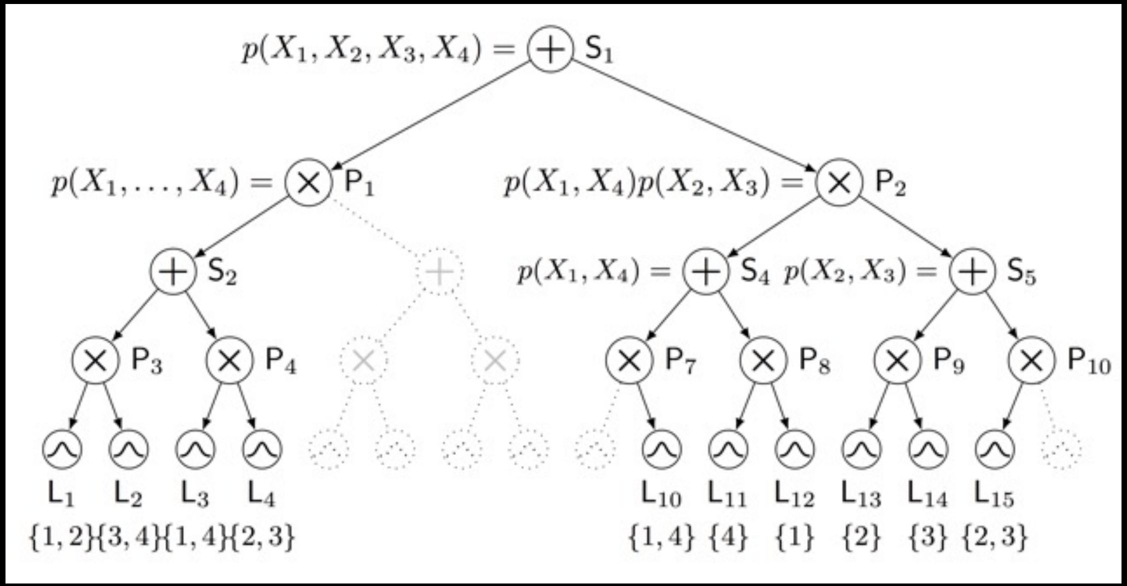
Problems that SPN try to solve





$$\Pr(R=T \mid G=T) = \frac{\Pr(G=T, R=T)}{\Pr(G=T)} = \frac{\sum_{x\in\{T,F\}} \Pr(G=T, S=x, R=T)}{\sum_{x,y\in\{T,F\}} \Pr(G=T, S=x, R=y)}$$

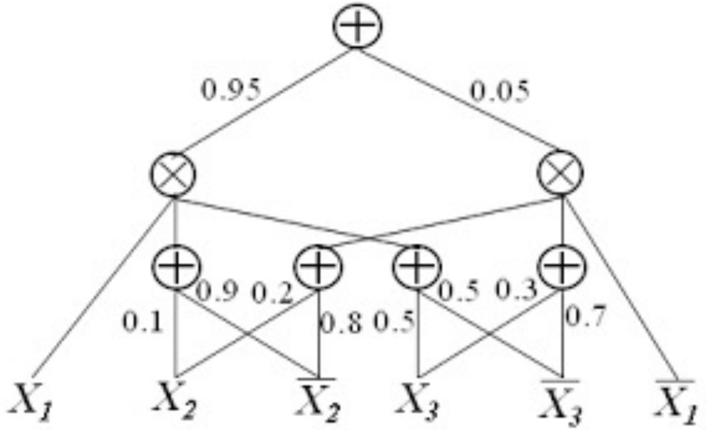
Features of SPN



What is a SPN?

1. SPN – a joint distribution of a set of random variables

2. Three components: sum nodes, product nodes and leaves



What is a SPN?

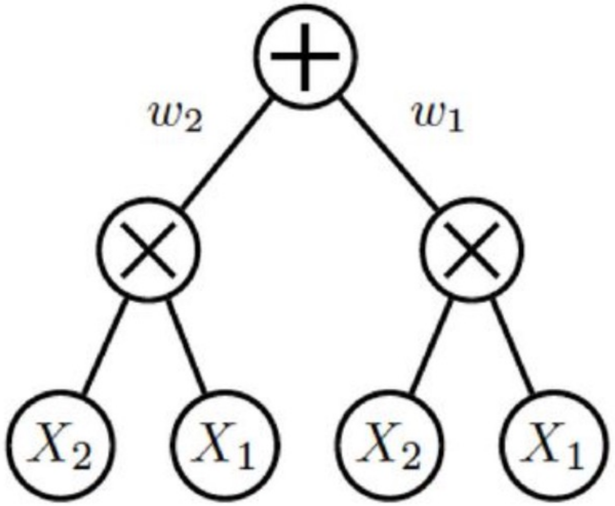
1. SPN – a joint distribution of a set of random variables

2. Three components: sum nodes, product nodes and leaves

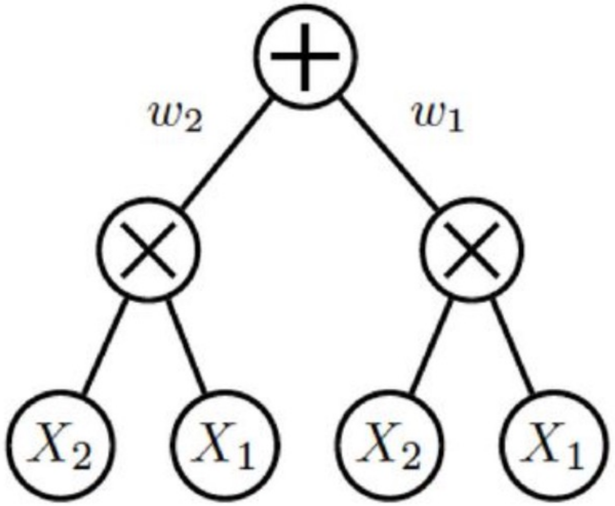
What is a SPN?

1. SPN – a joint distribution of a set of random variables

2. Three components: sum nodes, product nodes and leaves

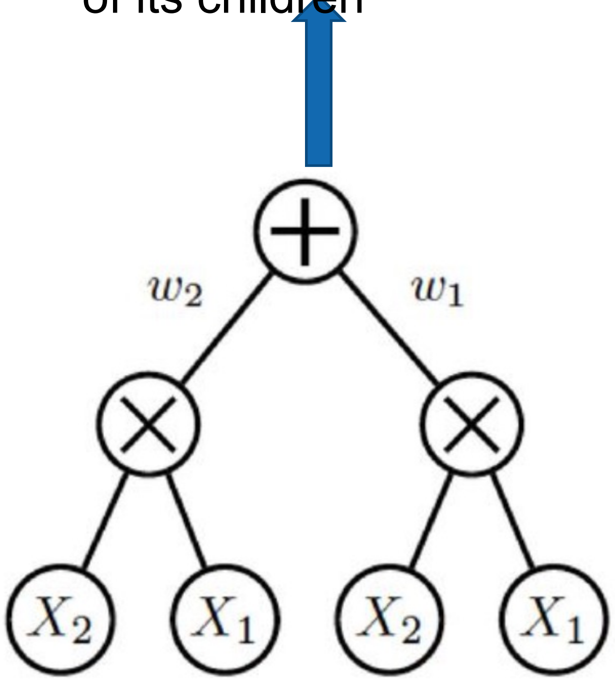


What is a SPN?



What is a SPN?

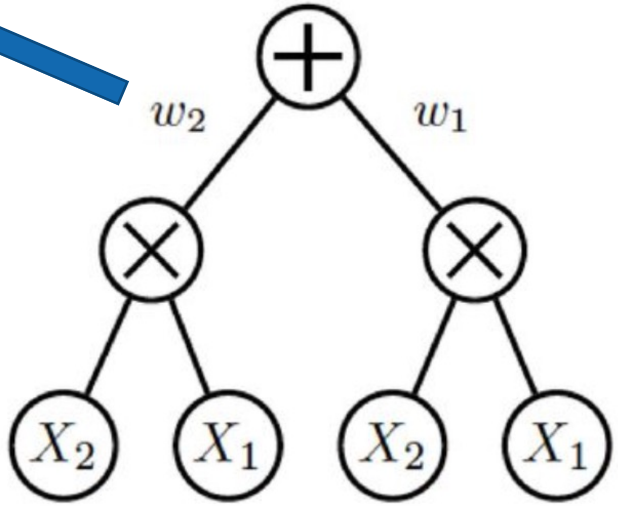
Sum of the values
of its children



What is a SPN?

Sum of the values
of its children

Non-negtive weight
associated with sum nodes

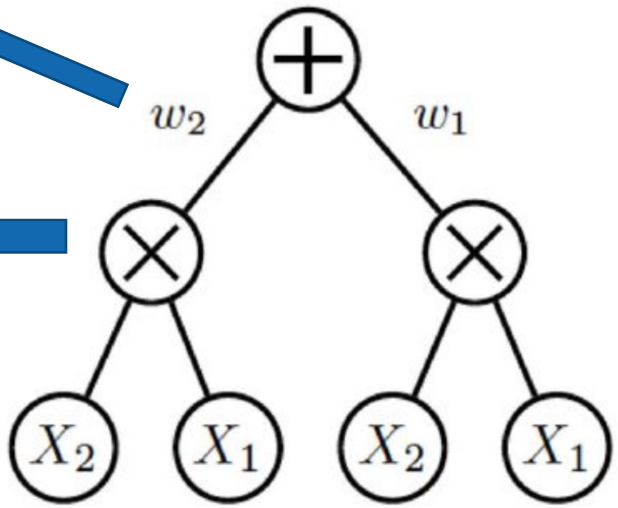


What is a SPN?

Sum of the values
of its children

Non-negtive weight
associated with sum nodes

Product of the values
of its children

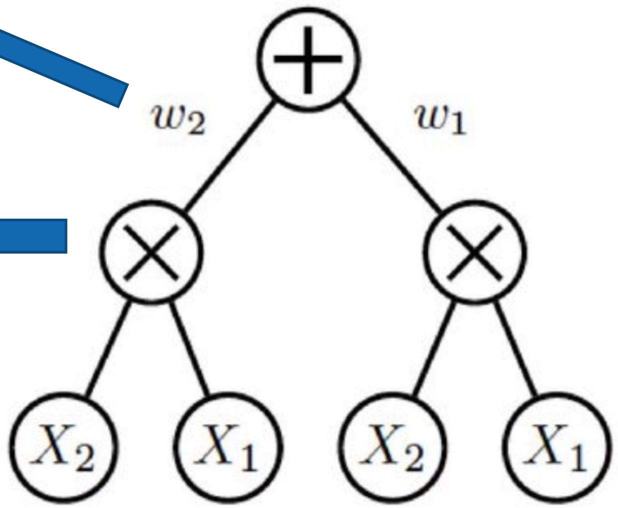


What is a SPN?

Sum of the values
of its children

Non-negtive weight
associated with sum nodes

Product of the values
of its children



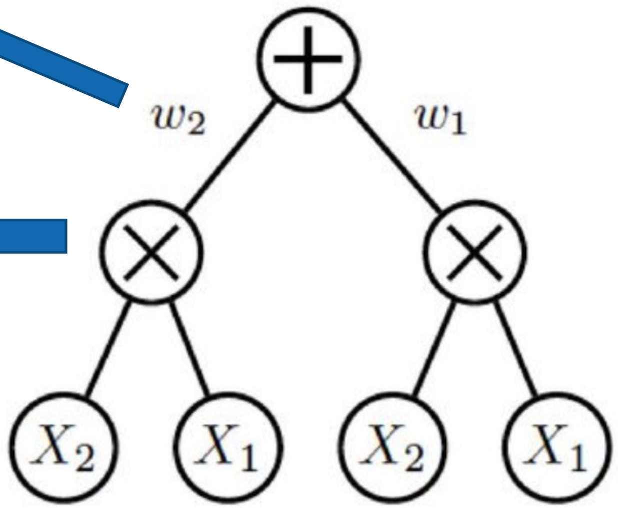
Unnormalized
univariate
distribution

What is a SPN?

Sum of the values
of its children

Non-negtive weight
associated with sum nodes

Product of the values
of its children



Unnormalized
univariate
distribution

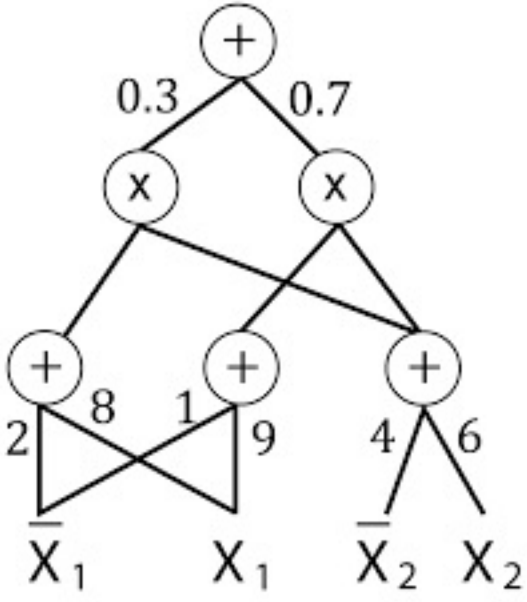
What is a valid SPN?

1. Decomposability: for each child of a Product node, they have disjoint scopes

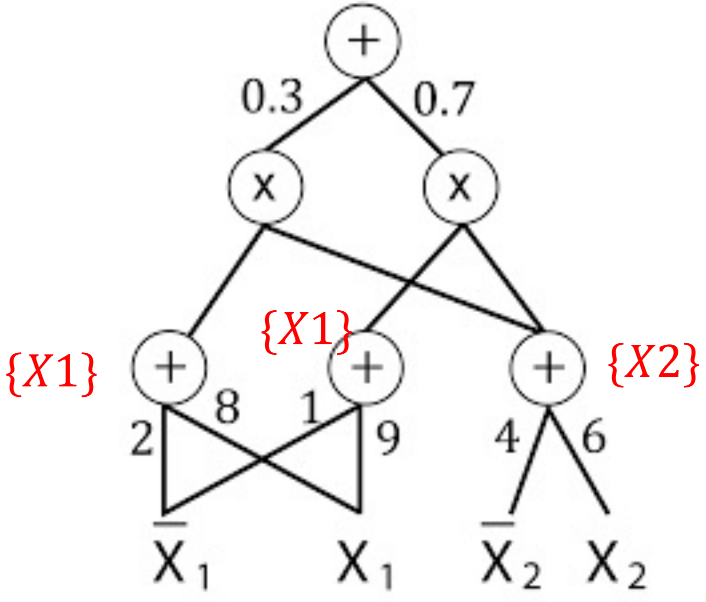
2. Completeness: for each child of a Sum node, they have identical scopes

What is a valid SPN?

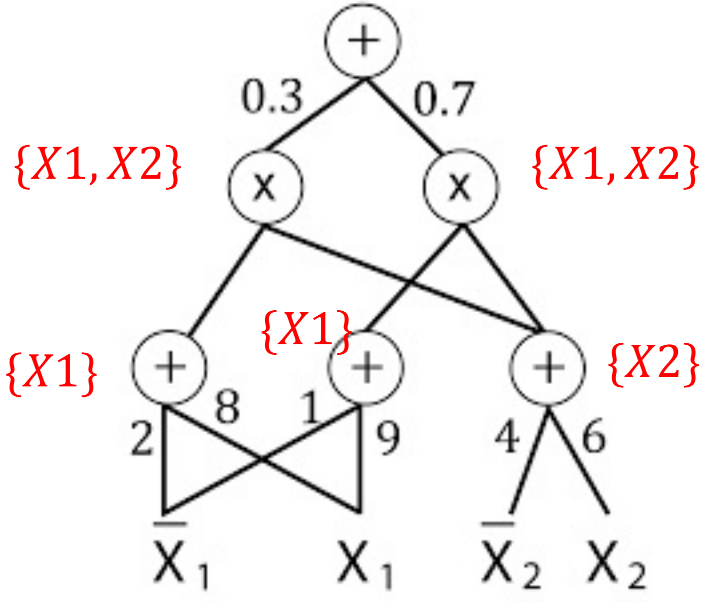
What is a valid SPN?



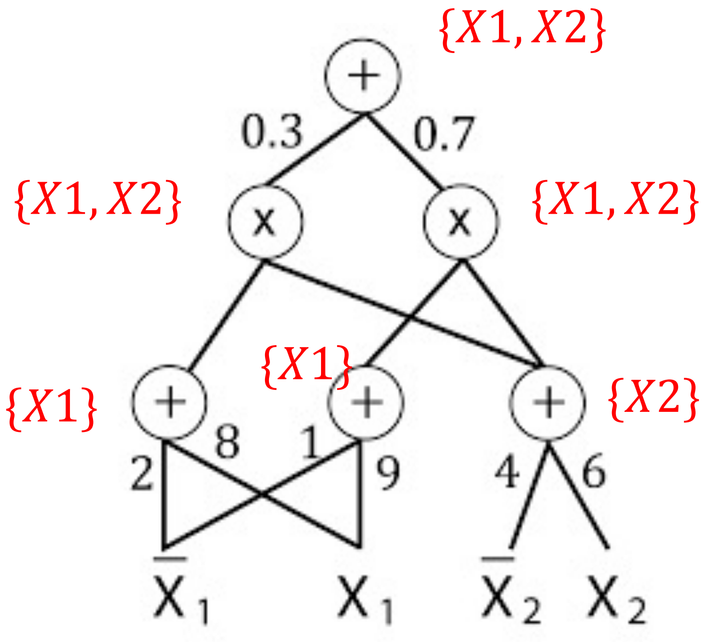
What is a valid SPN?



What is a valid SPN?



What is a valid SPN?

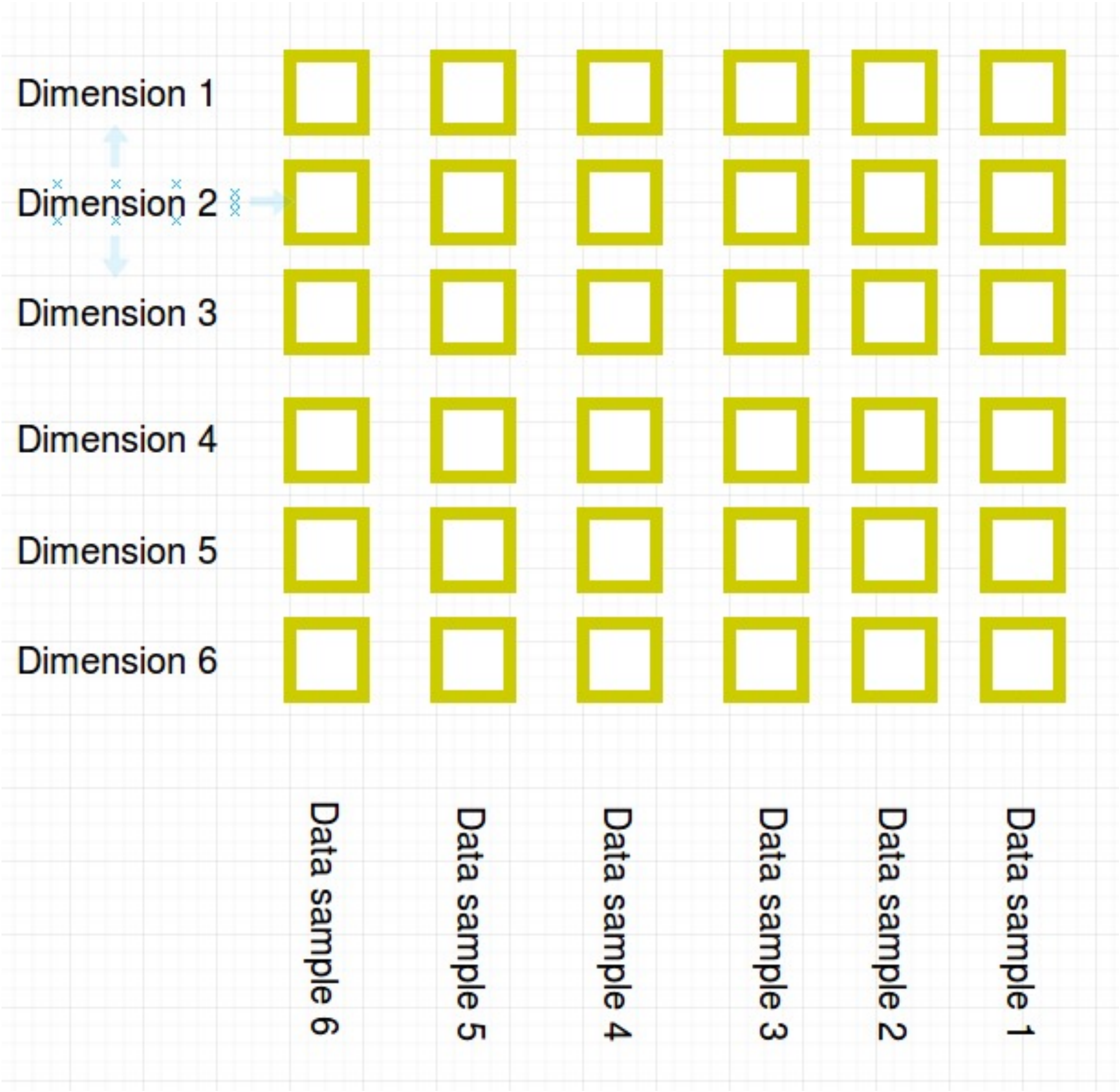


How does a SPN do inference in linear time?

1. What do Sum nodes and Product nodes mean?

How does a SPN do inference in linear time?

1. What do Sum nodes and Product nodes mean?

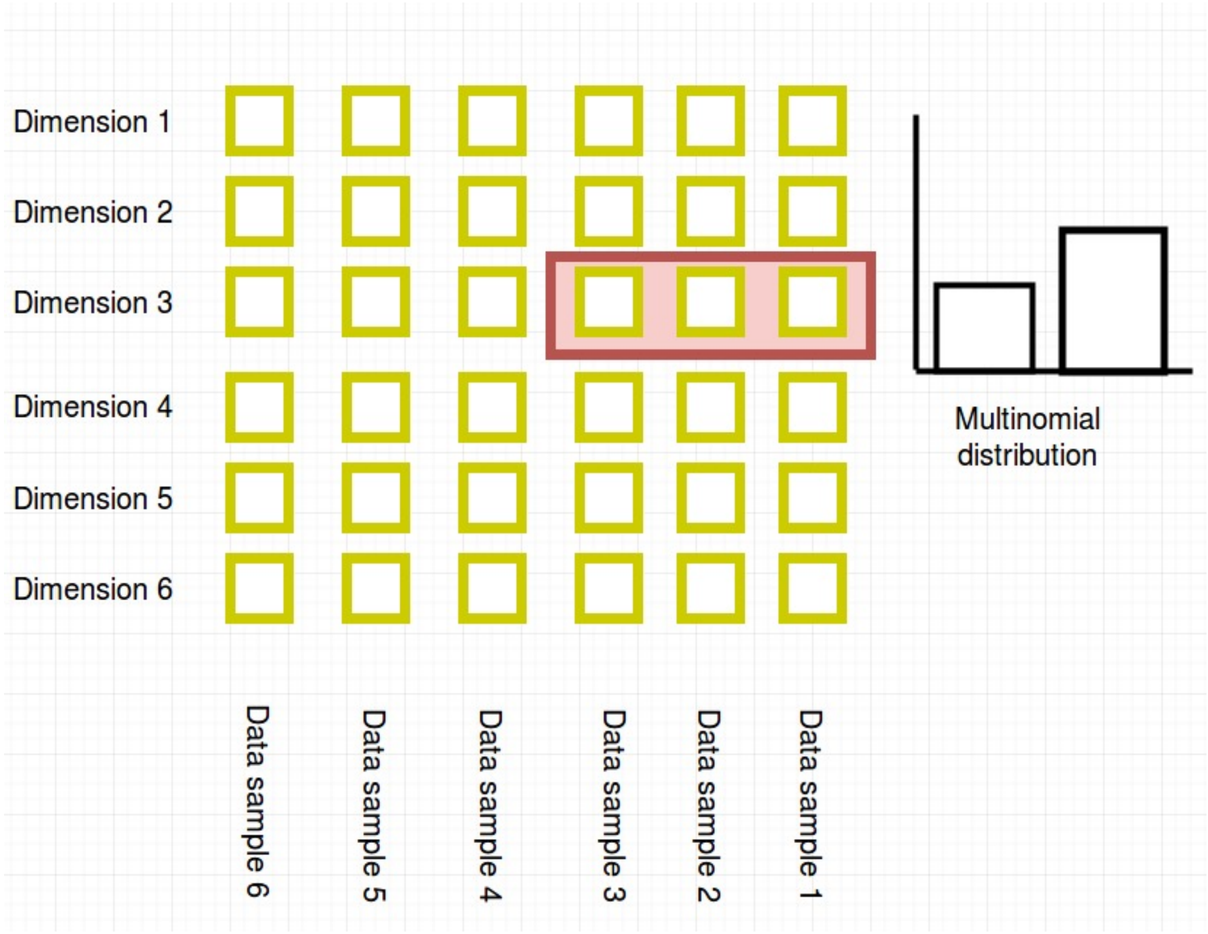


How does a SPN do inference in linear time?

1. What do Sum nodes and Product nodes mean?

How does a SPN do inference in linear time?

1. What do Sum nodes and Product nodes mean?

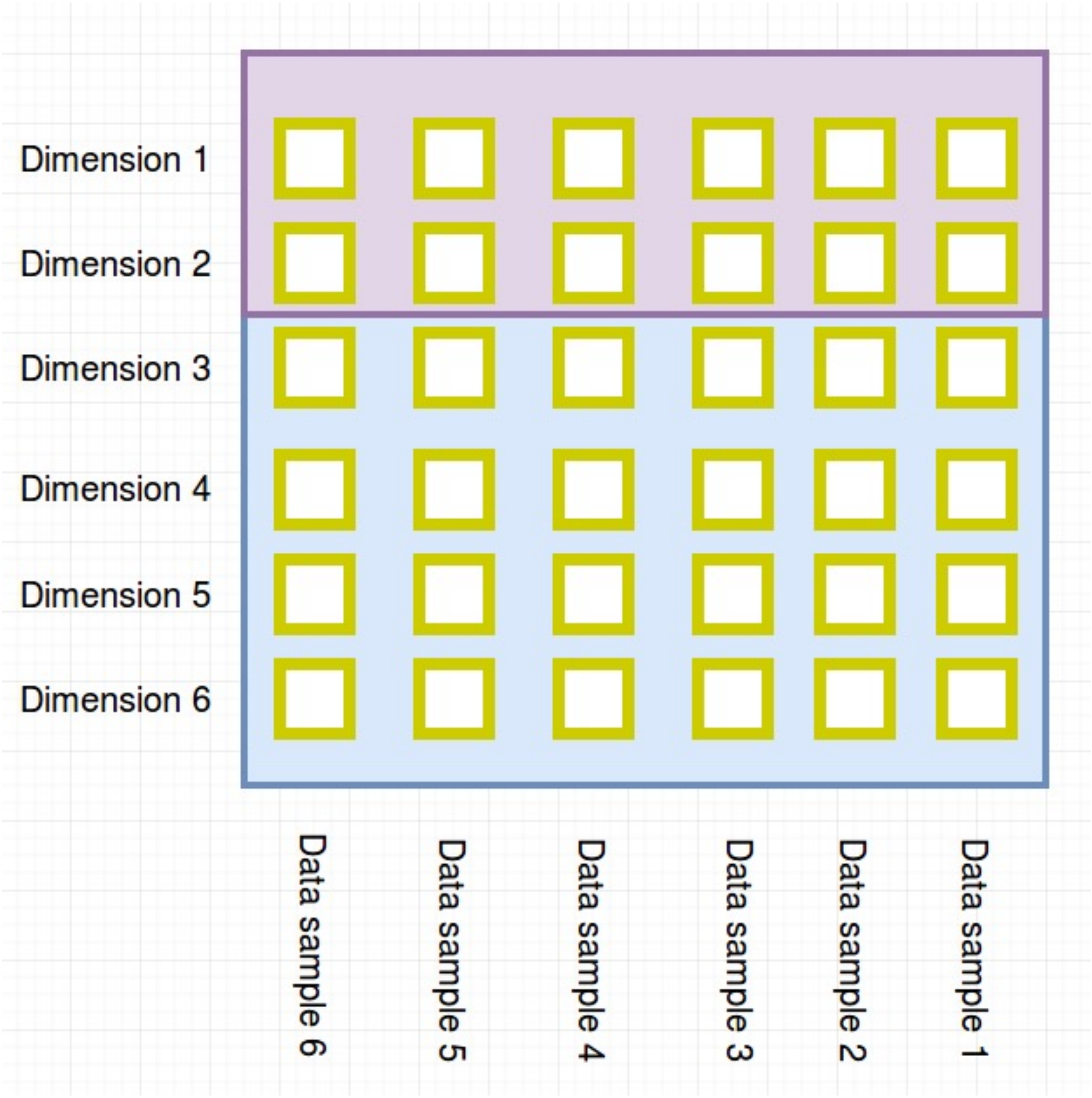


How does a SPN do inference in linear time?

1. What do Sum nodes and Product nodes mean?

How does a SPN do inference in linear time?

1. What do Sum nodes and Product nodes mean?

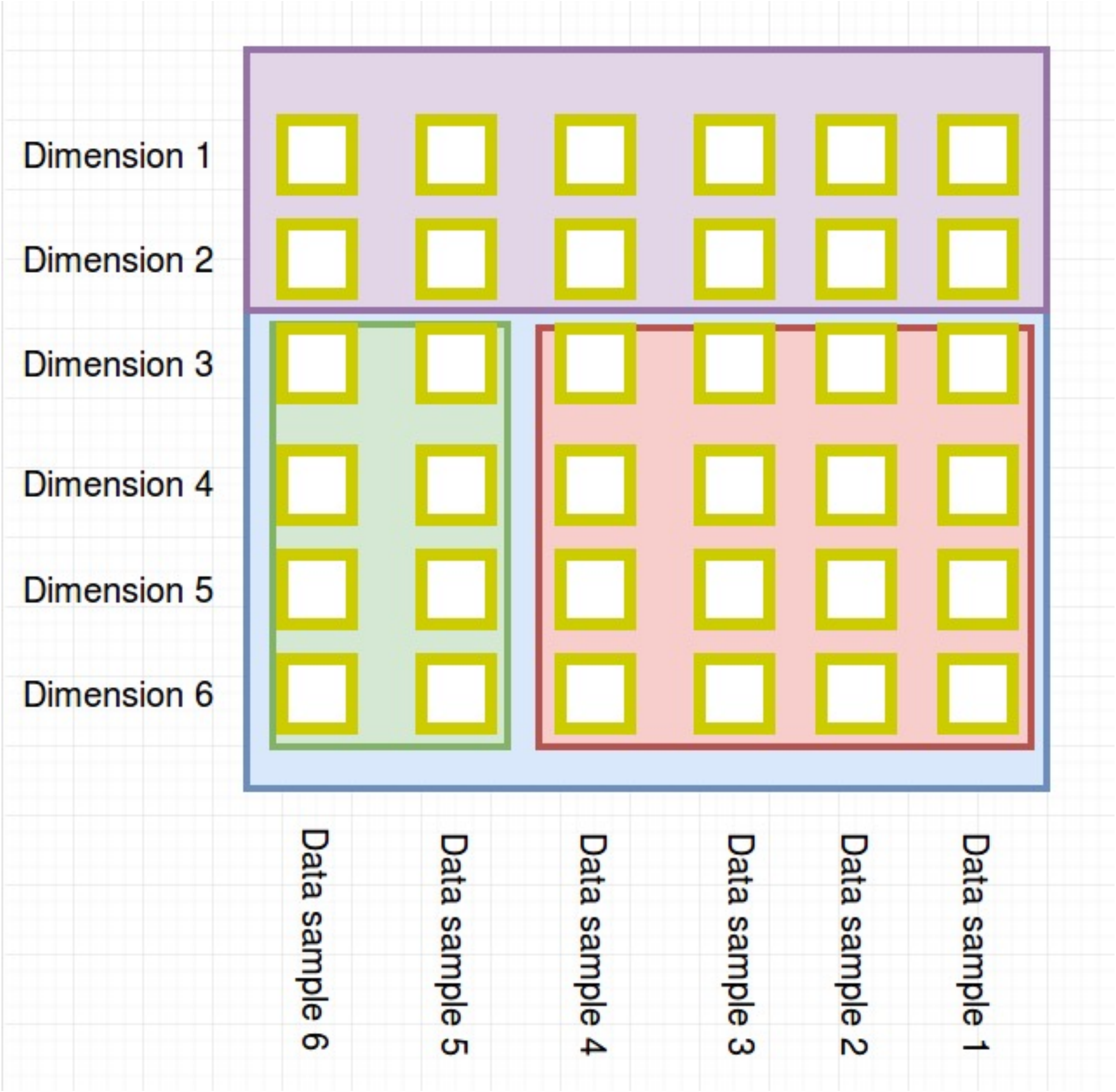


How does a SPN do inference in linear time?

1. What do Sum nodes and Product nodes mean?

How does a SPN do inference in linear time?

1. What do Sum nodes and Product nodes mean?

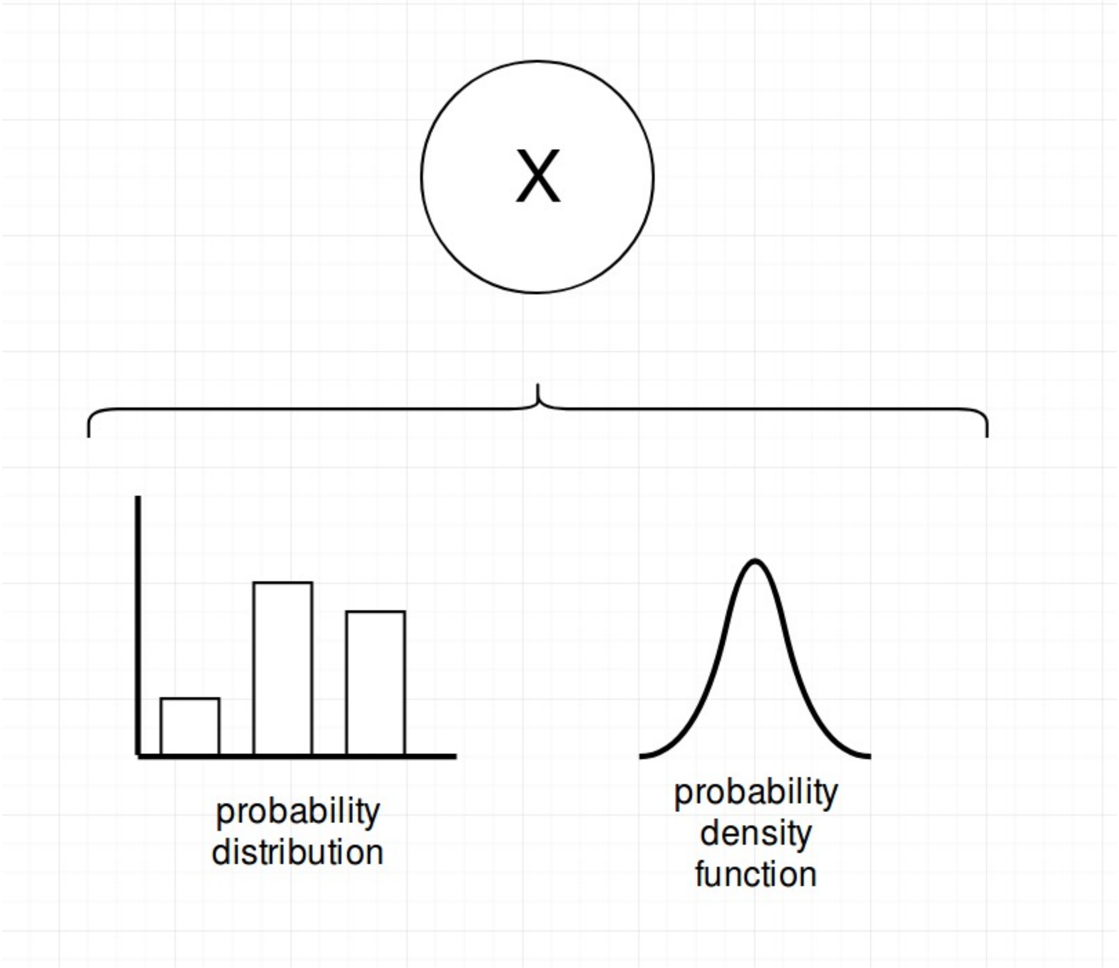


How does a SPN do inference in linear time?

1. What do Sum nodes and Product nodes mean?

How does a SPN do inference in linear time?

1. What do Sum nodes and Product nodes mean?

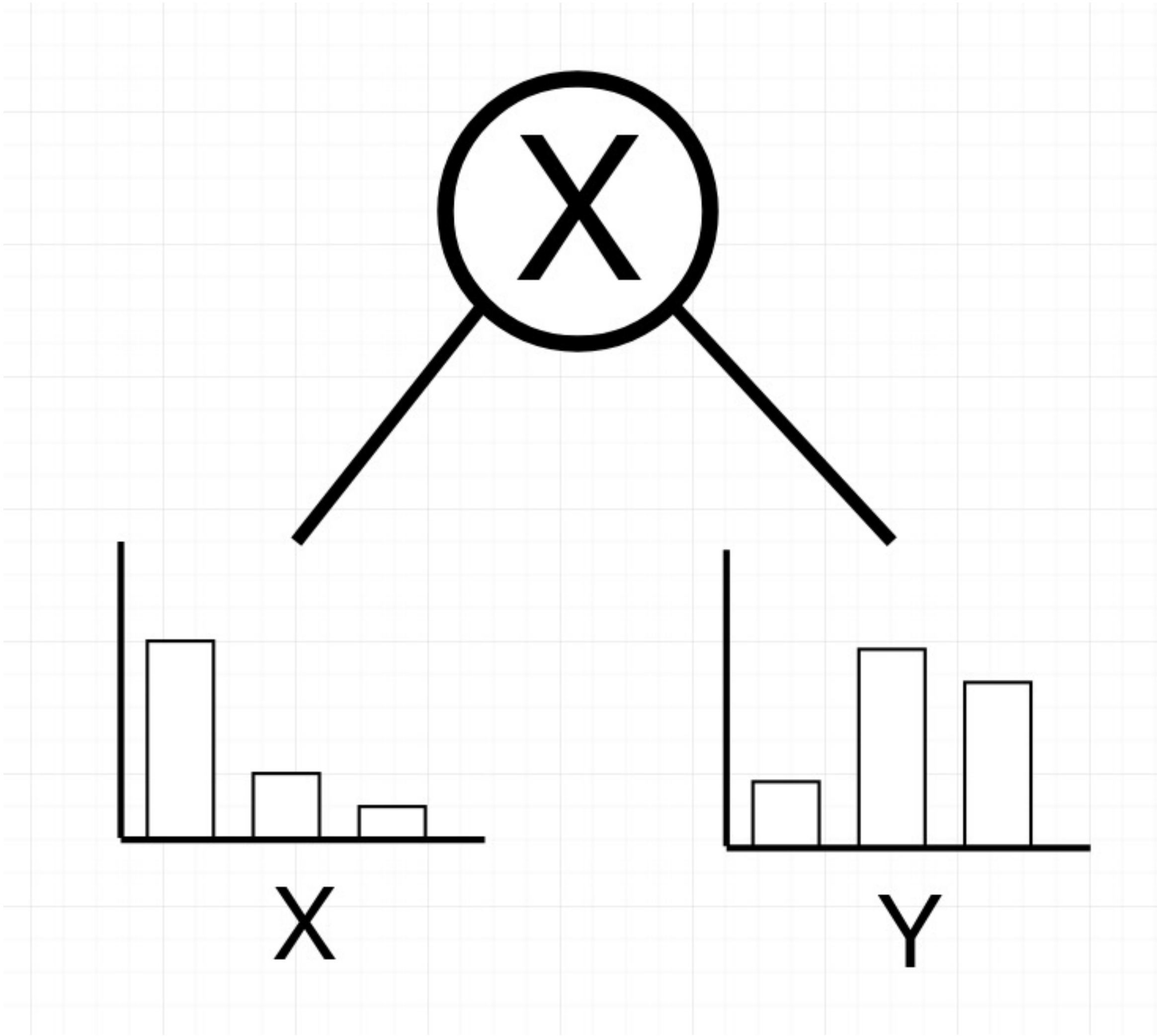


How does a SPN do inference in linear time?

1. What do Sum nodes and Product nodes mean?

How does a SPN do inference in linear time?

1. What do Sum nodes and Product nodes mean?

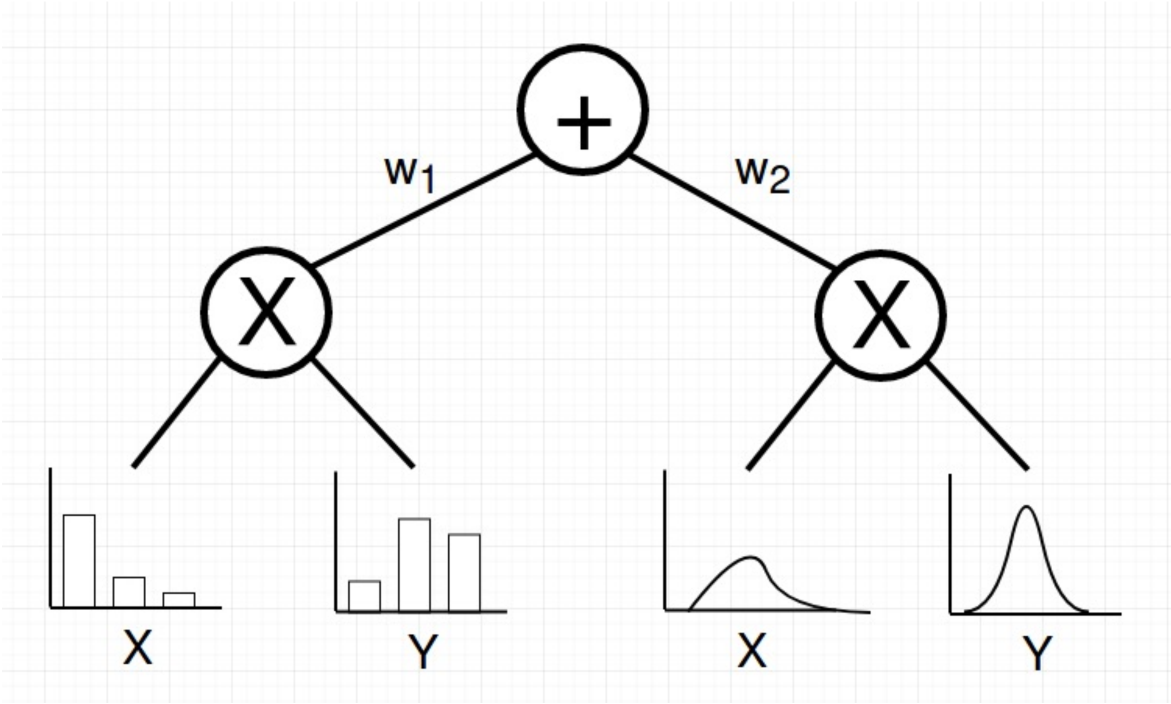


How does a SPN do inference in linear time?

1. What do Sum nodes and Product nodes mean?

How does a SPN do inference in linear time?

1. What do Sum nodes and Product nodes mean?



How does a SPN do inference in linear time?

1. What do Sum nodes and Product nodes mean?

How does a SPN do inference in linear time?

1. What do Sum nodes and Product nodes mean?

2. Computation Process

How does a SPN do inference in linear time?

1. What do Sum nodes and Product nodes mean?

2. Computation Process

How does a SPN do inference in linear time?

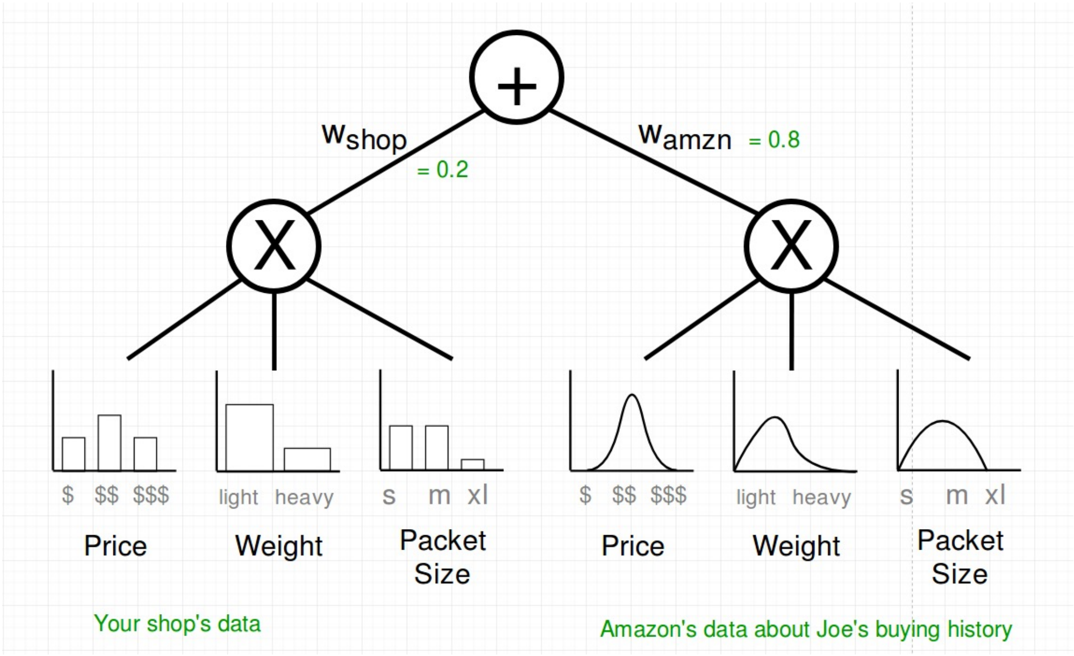
1. What do Sum nodes and Product nodes mean?

2. Computation Process

How does a SPN do inference in linear time?

1. What do Sum nodes and Product nodes mean?

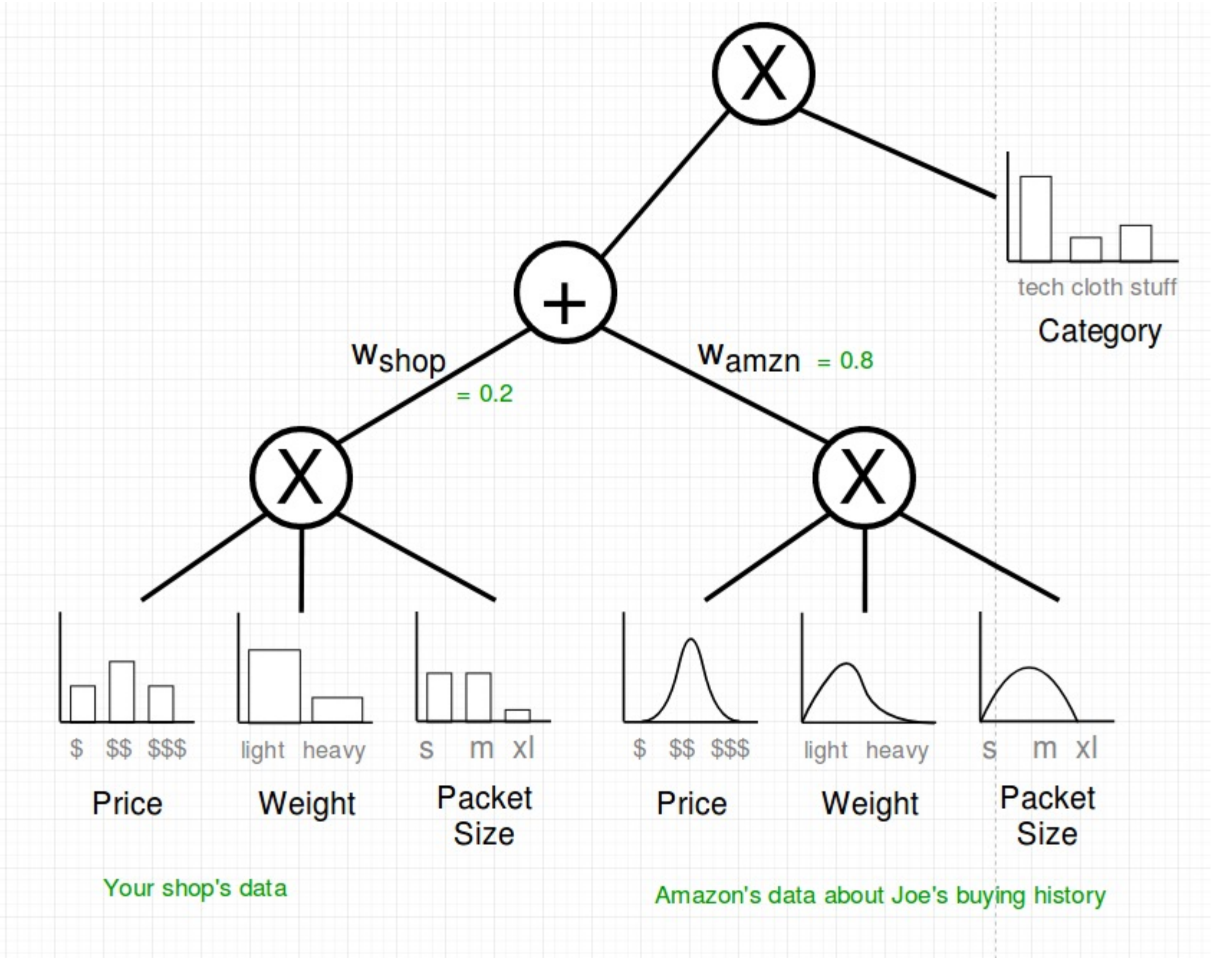
2. Computation Process



How does a SPN do inference in linear time?

1. What do Sum nodes and Product nodes mean?

2. Computation Process



How does a SPN do inference in linear time?

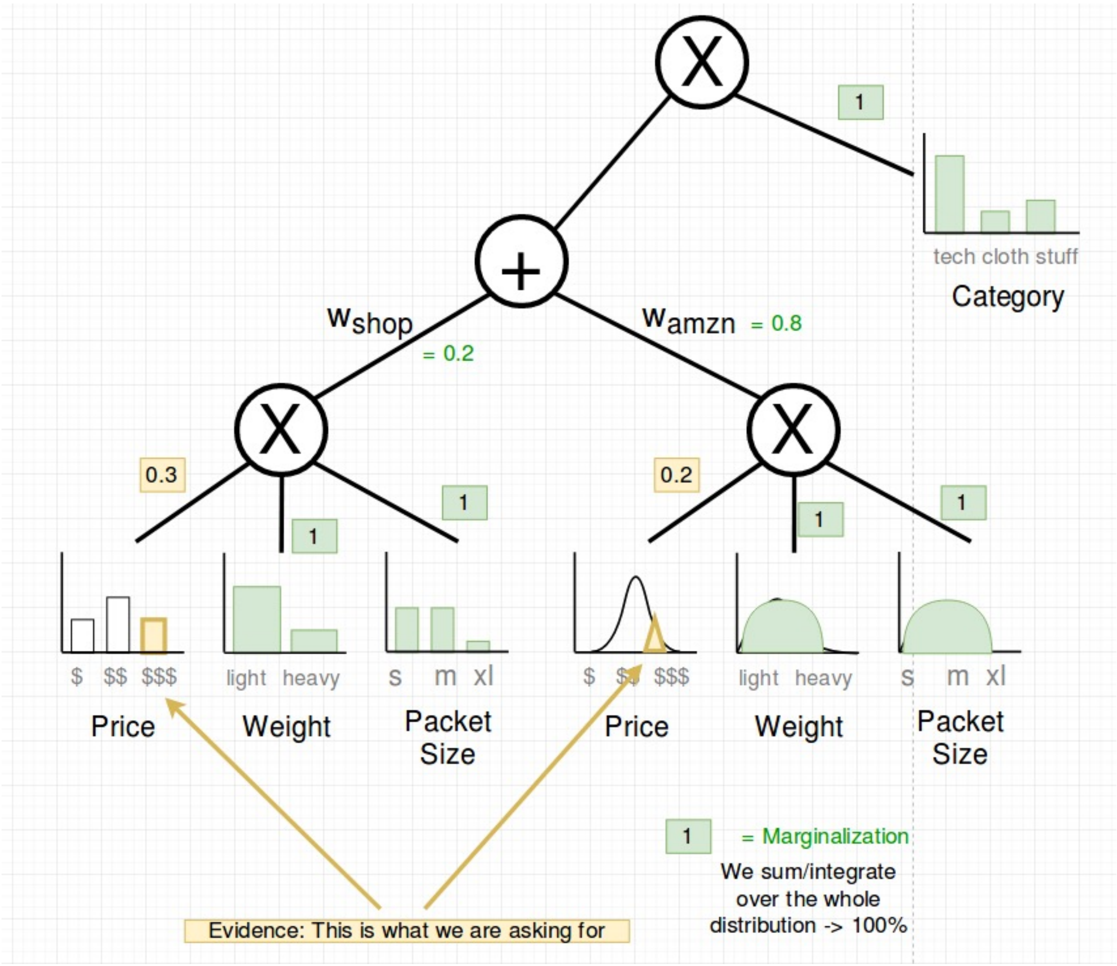
1. What do Sum nodes and Product nodes mean?

2. Computation Process

How does a SPN do inference in linear time?

1. What do Sum nodes and Product nodes mean?

2. Computation Process



How does a SPN do inference in linear time?

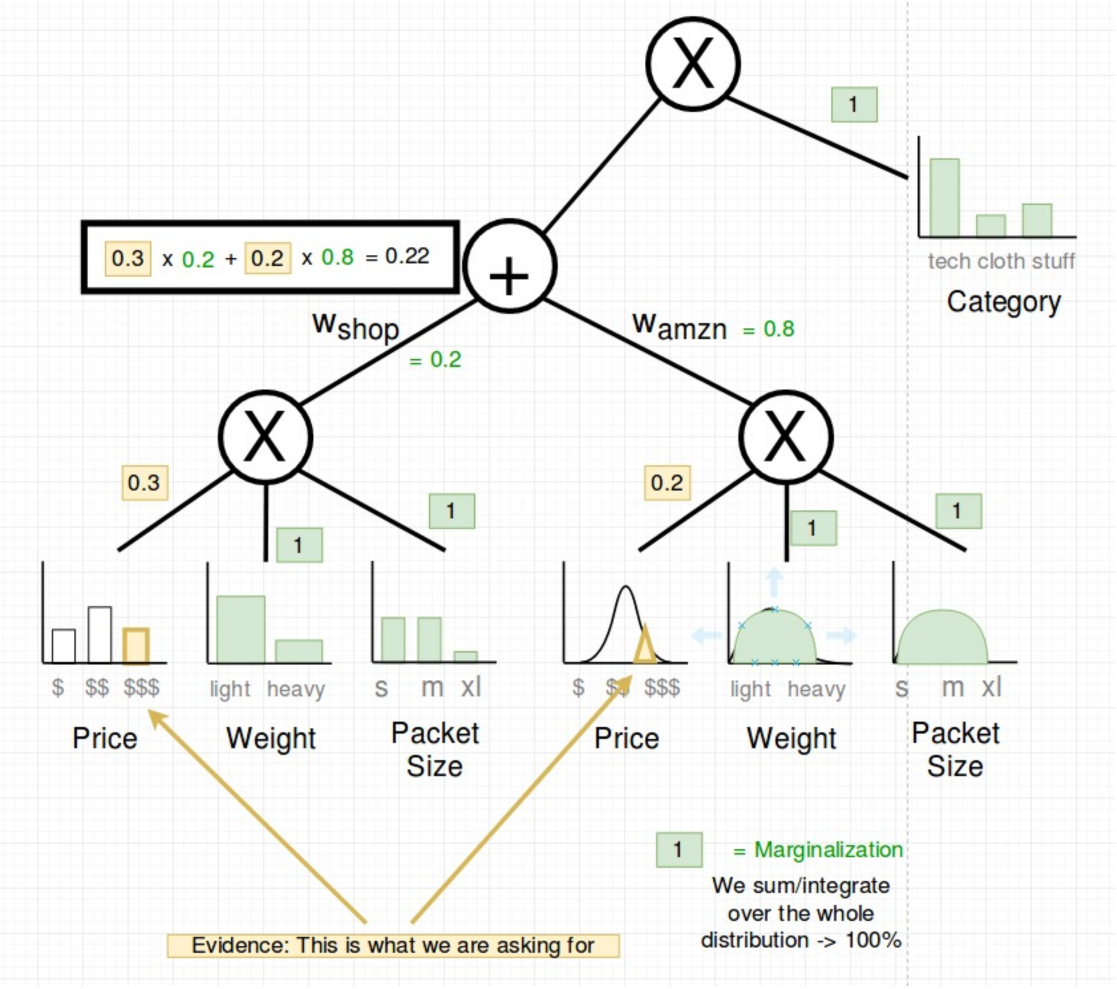
1. What do Sum nodes and Product nodes mean?

2. Computation Process

How does a SPN do inference in linear time?

1. What do Sum nodes and Product nodes mean?

2. Computation Process



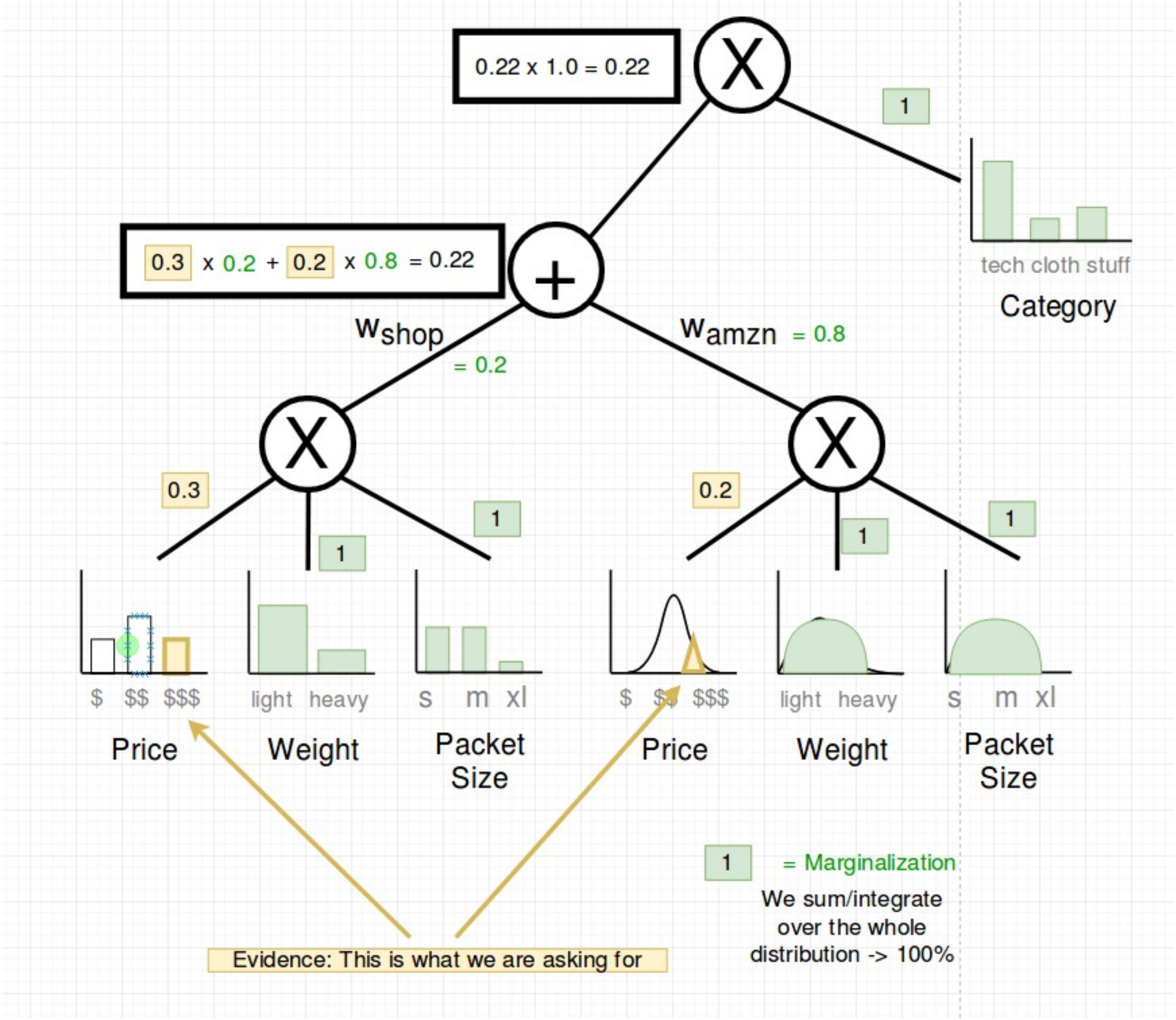
How does a SPN do inference in linear time?

1. What do Sum nodes and Product nodes mean?

2. Computation Process

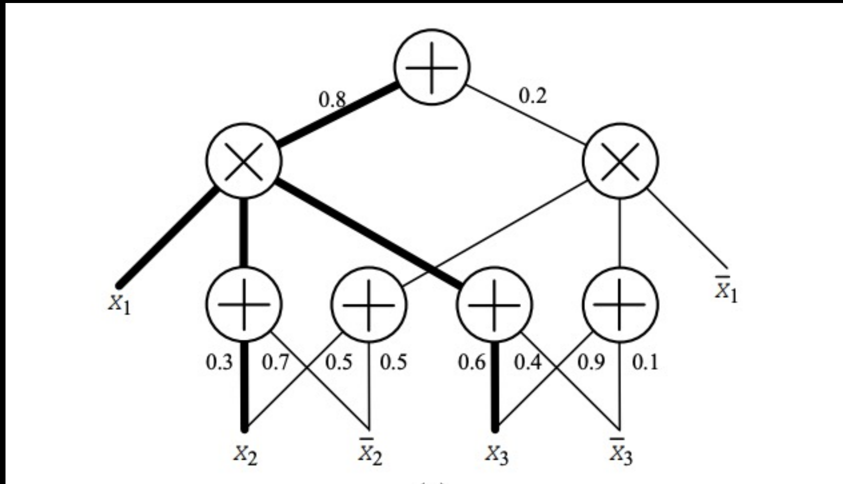
How does a SPN do inference in linear time?

1. What do Sum nodes and Product nodes mean?



2. Computation Process

Parameter Learning



(a)

Bayesian Parameter Learning in SPNs

1. Define SPN $= (\mathcal{G}, \psi, \boldsymbol{w}, \theta)$.
   1. $\mathcal{G}$ is a computational graph;
   2. ψ is a scope function.
   3. $\boldsymbol{w}$ is a set of sum-weights.
   4. θ is a set of leaf parameters.
2. $\mathcal{G}$ has few structural requirements
3. Learning ψ is challenging
4. Develop a parametrisation of ψ.

Learning Parameters $\boldsymbol{w}, \theta$ – Fixing Scope Function ψ

1. Any sum node S and assume that it has K_s children

2. Each data instance X_n and each S
   1. Latent variable $Z_{S,n}$ with K_s states and categorical distribution given by the weights $\boldsymbol{w_S}$ of S

3. Let $\boldsymbol{Z}_n = \{Z_{S,n}\}S \in \boldsymbol{S}.$

4. Induced tree
   1. for each sum $S \in \mathcal{G}$, delete all but one outgoing edge
   2. delete all nodes and edges which are now unreachable from the root.

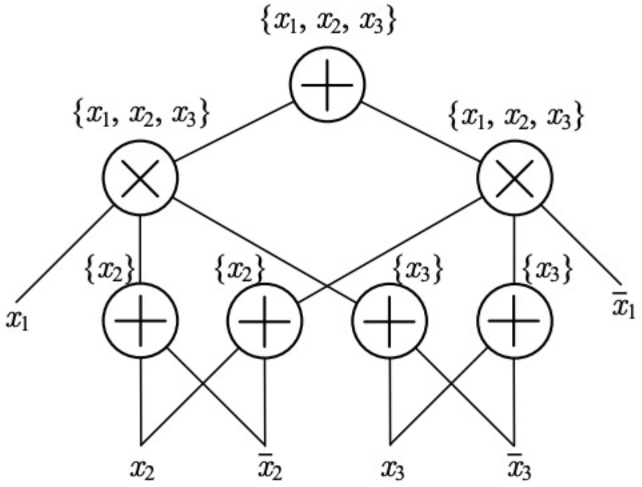
Learning Parameters $\boldsymbol{w}, \theta$ – Fixing Scope Function ψ

1. Any sum node S and assume that it has K_s children

2. Each data instance X_n and each S
   1. Latent variable $Z_{S,n}$ with K_s states and categorical distribution given by the weights $\boldsymbol{w_S}$ of S

3. Let $\boldsymbol{Z}_n = \{Z_{S,n}\} S \in \boldsymbol{S}$.

4. Induced tree
   1. for each sum $S \in \mathcal{G}$, delete all but one outgoing edge
   2. delete all nodes and edges which are now unreachable



Learning Parameters $\boldsymbol{w}, \theta$ – Fixing Scope Function ψ

1. Any sum node S and assume that it has K_s children

2. Each data instance X_n and each S
   1. Latent variable $Z_{S,n}$ with K_s states and categorical distribution given by the weights $\boldsymbol{w_S}$ of S

3. Let $\boldsymbol{Z}_n = \{Z_{S,n}\} S \in \boldsymbol{S}$.

4. Induced tree
   1. for each sum $S \in \mathcal{G}$, delete all but one outgoing edge
   2. delete all nodes and edges which are now unreachable from the root.

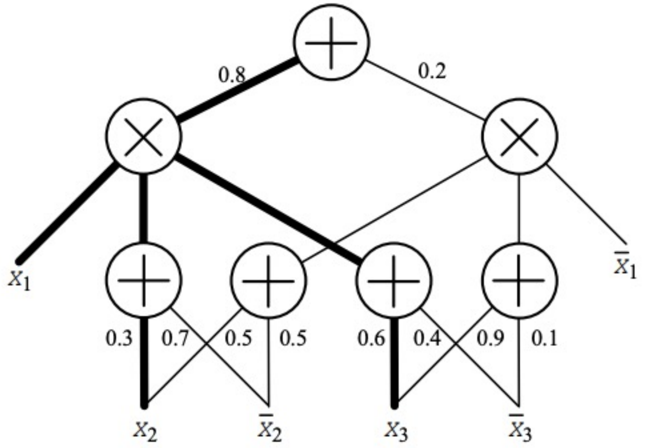
Learning Parameters $\boldsymbol{w}, \theta$ – Fixing Scope Function ψ

1. Any sum node S and assume that it has K_s children

2. Each data instance X_n and each S
   1. Latent variable $Z_{S,n}$ with K_s states and categorical distribution given by the weights $\boldsymbol{w_S}$ of S

3. Let $\boldsymbol{Z}_n = \{Z_{S,n}\}S \in \boldsymbol{S}$.

4. Induced tree
   1. for each sum $S \in \mathcal{G}$, delete all but one outgoing edge
   2. delete all nodes and edges which are now unreachable fror



(a)

Learning Parameters $\boldsymbol{w}, \theta$ – Fixing Scope Function ψ

1. Any sum node S and assume that it has K_s children

2. Each data instance X_n and each S
    1. Latent variable $Z_{S,n}$ with K_s states and categorical distribution given by the weights $\boldsymbol{w_S}$ of S

3. Let $\boldsymbol{Z}_n = \{Z_{S,n}\} S \in \boldsymbol{S}$.

4. Induced tree
    1. for each sum $S \in \mathcal{G}$, delete all but one outgoing edge
    2. delete all nodes and edges which are now unreachable from the root.

Learning Parameters $\boldsymbol{w}, \theta$ – Fixing Scope Function ψ

1. Rewrite the SPN distribution
   1. $S(\boldsymbol{x}) = \sum_{T \sim S} \prod_{(S,N) \in T} w_{S,N} \prod_{L \in T} L(X_L)$
2. Define $T(\boldsymbol{z})$: assigns to each value $\boldsymbol{z}$ of $\boldsymbol{Z}$ the induced tree determined by $\boldsymbol{z}$
   1. $\boldsymbol{z}$ indicates the kept sum edges in Induced tree definition
3. Partially Invertible:
   1. given an induced tree T, can perfectly retrieve the states of the (latent variables of) sum nodes in T
4. Conditional distribution: $p(\boldsymbol{x}|\boldsymbol{z}) = \prod_{L \in T} L(X_L)$ and prior $p(\boldsymbol{z}) = \prod_{S \in \mathcal{G}} w_{S,z_S}$
   1. $p(\boldsymbol{x}) = \sum_{\boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z}) = \sum_{\boldsymbol{z}} p(\boldsymbol{x}|\boldsymbol{z}) p(\boldsymbol{z})$

Learning Parameters $\boldsymbol{w}, \theta$ – Fixing Scope Function ψ

1. Rewrite the SPN distribution
   1. $S(\boldsymbol{x}) = \sum_{T\sim S} \prod_{(S,N)\in T} w_{S,N} \prod_{L\in T} L(X_L)$

2. Define $T(\boldsymbol{z})$: assigns to each value $\boldsymbol{z}$ of $\boldsymbol{Z}$ the induced tree determined by $\boldsymbol{z}$
   1. $\boldsymbol{z}$ indicates the kept sum edges in Induced tree definition

3. Partially Invertible:
   1. given an induced tree T, can perfectly retrieve the states

$$\sum_{\mathbf{z}} \prod_{\mathsf{S}\in\mathbf{S}} w_{\mathsf{S},z_{\mathsf{S}}} \prod_{\mathsf{L}\in T(\mathbf{z})} L(\mathbf{x}_{\mathsf{L}}) = \sum_{\mathcal{T}} \sum_{\mathbf{z}\in T^{-1}(\mathcal{T})} \prod_{\mathsf{S}\in\mathbf{S}} w_{\mathsf{S},z_{\mathsf{S}}} \prod_{\mathsf{L}\in T(\mathbf{z})} L(\mathbf{x}_{\mathsf{L}})$$

$$= \sum_{\mathcal{T}} \prod_{(\mathsf{S},\mathsf{N})\in\mathcal{T}} w_{\mathsf{S},\mathsf{N}} \prod_{\mathsf{L}\in\mathcal{T}} \mathsf{L}(\mathbf{x}_{\mathsf{L}}) \underbrace{\left(\sum_{\bar{\mathbf{z}}} \prod_{\mathsf{S}\in\bar{\mathbf{S}}_{\mathcal{T}}} w_{\mathsf{S},\bar{z}_{\mathsf{S}}}\right)}_{=1} = \mathcal{S}(\mathbf{x}),$$

Learning Parameters $\boldsymbol{w}, \theta$ – Fixing Scope Function ψ

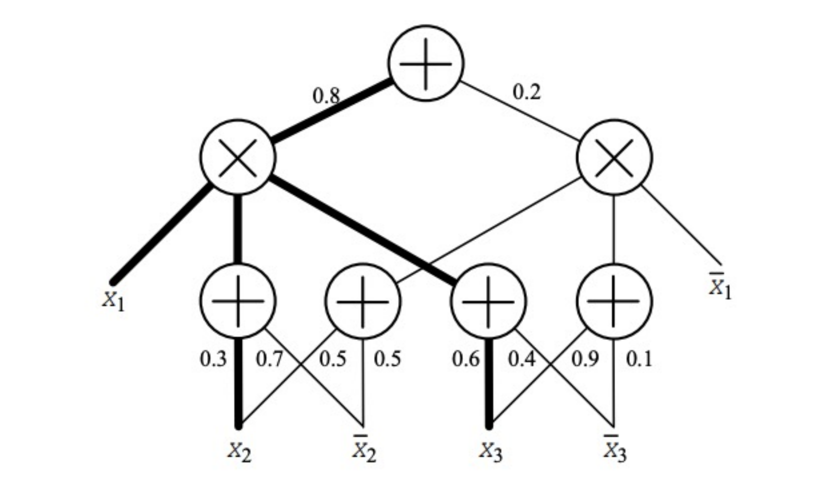
1. Extend the model to a Bayesian setting, by equipping the sum-weights w and leaf-parameters θ with suitable priors

Learning Parameters $\boldsymbol{w}, \theta$ – Fixing Scope Function ψ

1. Extend the model to a Bayesian setting, by equipping the sum-weights w and leaf-parameters θ with suitable priors

$$\mathbf{w}_\mathsf{S} \mid \alpha \sim \mathcal{D}ir(\mathbf{w}_\mathsf{S} \mid \alpha) \;\; \forall \mathsf{S}\,, \qquad z_{\mathsf{S},n} \mid \mathbf{w}_\mathsf{S} \sim \mathcal{C}at(z_{\mathsf{S},n} \mid \mathbf{w}_\mathsf{S}) \;\; \forall \mathsf{S}\, \forall n,$$
$$\theta_\mathsf{L} \mid \gamma \sim p(\theta_\mathsf{L} \mid \gamma) \;\; \forall \mathsf{L}\,, \qquad \mathbf{x}_n \mid \mathbf{z}_n, \theta \sim \prod_{\mathsf{L} \in T(\mathbf{z}_n)} \mathsf{L}(\mathbf{x}_{\mathsf{L},n} \mid \theta_\mathsf{L}) \;\; \forall n.$$

Structure Learning



(a)

Joint Learning $\boldsymbol{w}, \theta$ and ψ

1. Restrict to the class of SPN - $\mathcal{G}$ follows tree-shaped region graph

2. Tree-shaped region graph
    1. A region graph is a tuple (R, ψ): R is a DAG containing: regions (R) and partitions (P).
    2. Need to satisfy Decomposability and Completeness.
    3. A tree-shaped region graph (R, ψ): each node in R has at most one parent.

Joint Learning $\boldsymbol{w}, \theta$ and ψ

1. Induced Scope function: R is a tree-shaped region graph
   1. $\boldsymbol{Y} = \{Y_{P,d}\}_{P \in R, d \in \{1 \dots D\}}$
   2. $\psi_{\boldsymbol{y}}(Q) \coloneqq \{X_d \mid \prod_{P \in \prod} \mathbb{1}[R_{y_{P,d}} \in \Pi] = 1\}$

2. Incorporate Y in our model

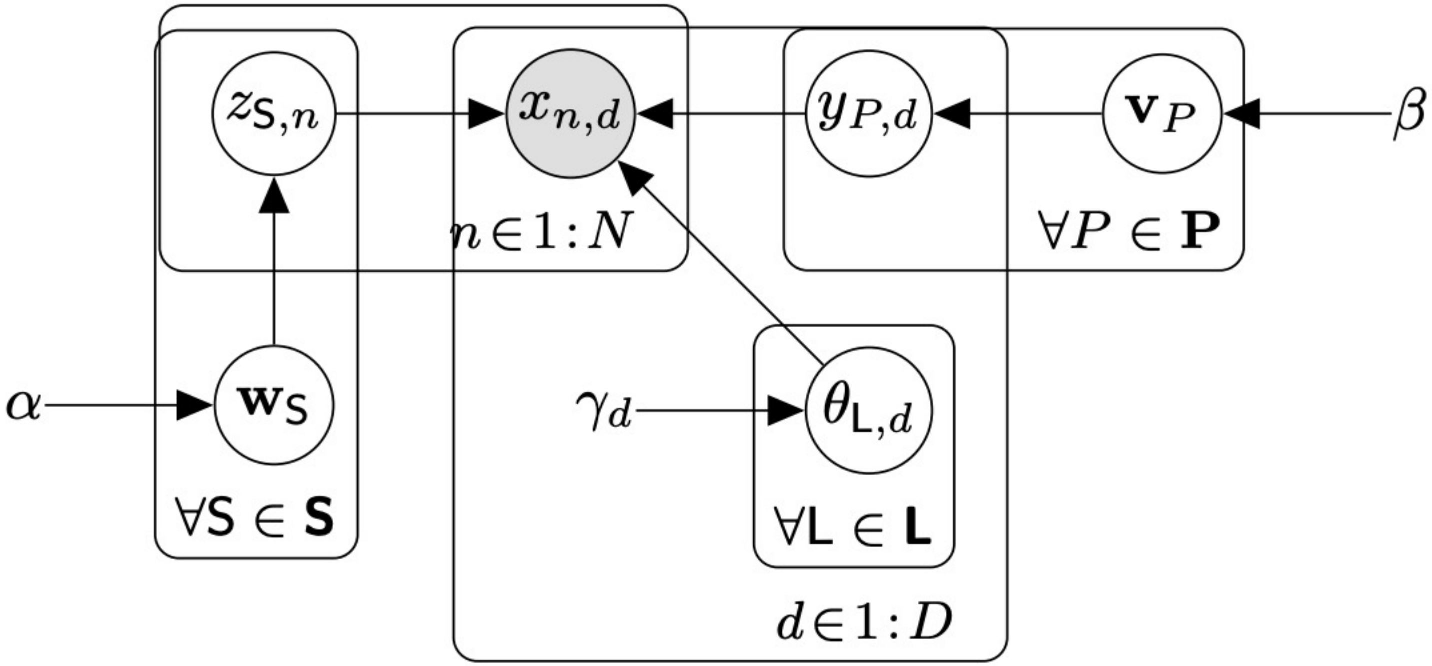
Joint Learning $\boldsymbol{w}, \theta$ and ψ

1. Induced Scope function: R is a tree-shaped region graph
   1. $\boldsymbol{Y} = \{Y_{P,d}\}_{P \in R, d \in \{1 \dots D\}}$
   2. $\psi_{\boldsymbol{y}}(Q) \coloneqq \{X_d \mid \prod_{P \in \prod} \mathbb{1}[R_{y_{P,d}} \in \Pi] = 1\}$

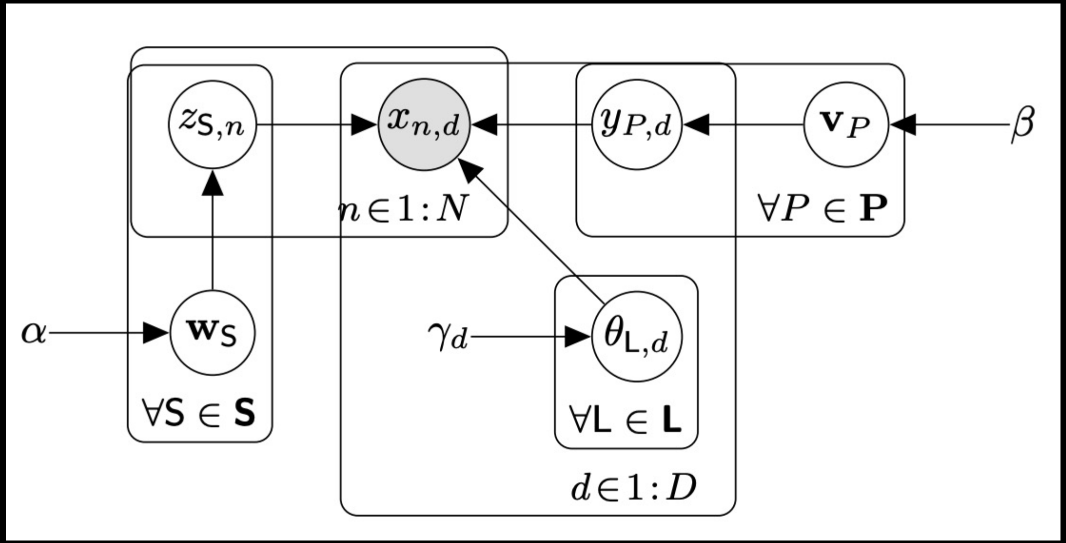
2. Incorporate Y in our model

$$
\begin{aligned}
\mathbf{w}_{\mathsf{S}} \,|\, \alpha &\sim \mathcal{D}ir(\mathbf{w}_{\mathsf{S}} \,|\, \alpha) \;\; \forall \mathsf{S}\,, & z_{\mathsf{S},n} \,|\, \mathbf{w}_{\mathsf{S}} &\sim \mathcal{C}at(z_{\mathsf{S},n} \,|\, \mathbf{w}_{\mathsf{S}}) \;\; \forall \mathsf{S}\, \forall n, \\
\mathbf{v}_P \,|\, \beta &\sim \mathcal{D}ir(\mathbf{v}_P \,|\, \beta) \;\; \forall P\,, & y_{P,d} \,|\, \mathbf{v}_P &\sim \mathcal{C}at(v_{P,d} \,|\, \mathbf{v}_P) \;\; \forall P\, \forall d, \\
\theta_{\mathsf{L}} \,|\, \gamma &\sim p(\theta_{\mathsf{L}} \,|\, \gamma) \;\; \forall \mathsf{L}, & \mathbf{x}_n \,|\, \mathbf{z}_n, \mathbf{y}, \theta &\sim \prod_{\mathsf{L} \in T(\mathbf{z}_n)} \mathsf{L}(\mathbf{x}_{\mathbf{y},n} \,|\, \theta_{\mathsf{L}}) \;\; \forall n.
\end{aligned}
$$

Joint Learning $\boldsymbol{w}, \theta$ and ψ



Update parameters

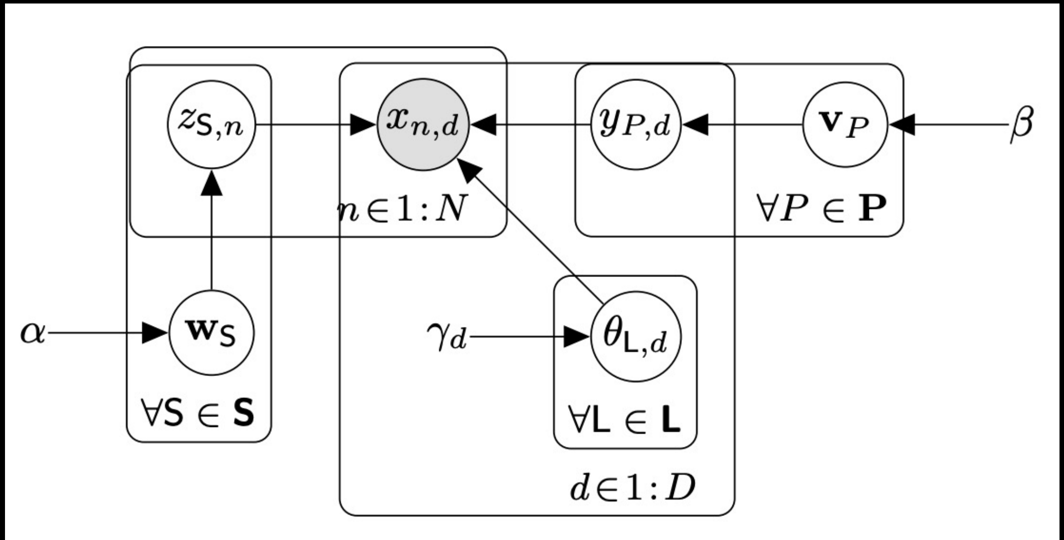


Update $\boldsymbol{w}, \theta$, fixed $\boldsymbol{y}$

1. $X = \{\boldsymbol{x}_n\}_{n=1}^{N}$: training set of N observations $\boldsymbol{x}_n$.
   1. Aim: draw posterior samples from the generative model given X

2. Perform Gibbs sampling!
   1. First update $\boldsymbol{w}, \theta$, fixed $\boldsymbol{y}$
      1. Sample $\boldsymbol{z}_n$ for all the sum latent variables $\boldsymbol{Z}_n$
      2. Sample sum weights from the posterior distributions of a Dirichlet
         1. $Dir(\alpha + c_{S,1}, \dots, \alpha + c_{S,K_S}), c_{S,K_S} = \sum_{n=1}^{N} \mathbb{1}[z_{S,n} = k]$

Update structure



Update $\boldsymbol{w}, \theta$, fixed $\boldsymbol{y}$

1. $X = \{\boldsymbol{x}_n\}_{n=1}^{N}$: training set of N observations $\boldsymbol{x}_n$.
    1. Aim: draw posterior samples from the generative model given X

2. Perform Gibbs sampling!
    1. Second update $\boldsymbol{y}$, fixed $\boldsymbol{w}, \theta$

Update $\boldsymbol{w}, \theta$, fixed $\boldsymbol{y}$

1. $X = \{\boldsymbol{x}_n\}_{n=1}^{N}$: training set of N observations $\boldsymbol{x}_n$.
   1. Aim: draw posterior samples from the generative model given X

2. Perform Gibbs sampling!
   1. Second update $\boldsymbol{y}$, fixed $\boldsymbol{w}, \theta$

$$p(y_{P,d} = k \mid \mathbf{y}_{P,\cancel{d}}, \mathbf{y}_{\mathbf{P}\setminus P,d}, \mathcal{X}, \mathbf{z}, \theta, \beta) = p(y_{P,d} = k \mid \mathbf{y}_{P,\cancel{d}}, \beta) p(\mathcal{X} \mid y_{P,d} = k, \mathbf{y}_{\mathbf{P}\setminus P,d}, \mathbf{z}, \theta)$$

Update $\boldsymbol{w}, \theta$, fixed $\boldsymbol{y}$

1. $X = \{\boldsymbol{x}_n\}_{n=1}^{N}$: training set of N observations $\boldsymbol{x}_n$.
    1. Aim: draw posterior samples from the generative model given X

2. Perform Gibbs sampling!
    1. Second update $\boldsymbol{y}$, fixed $\boldsymbol{w}, \theta$

$$p(y_{P,d} = k \,|\, \mathbf{y}_{P,\cancel{d}}, \mathbf{y}_{\mathbf{P}\setminus P,d}, \mathcal{X}, \mathbf{z}, \theta, \beta) = p(y_{P,d} = k \,|\, \mathbf{y}_{P,\cancel{d}}, \beta) p(\mathcal{X} \,|\, y_{P,d} = k, \mathbf{y}_{\mathbf{P}\setminus P,d}, \mathbf{z}, \theta)$$

$$p(y_{P,d} = k \,|\, \mathbf{y}_{P,\cancel{d}}, \beta) = \frac{\beta + m_{P,k}}{\sum_{j=1}^{|\mathbf{ch}(P)|} \beta + m_{P,k}}$$

Update $\boldsymbol{w}, \theta$, fixed $\boldsymbol{y}$

1. $X = \{\boldsymbol{x}_n\}_{n=1}^{N}$: training set of N observations $\boldsymbol{x}_n$.
   1. Aim: draw posterior samples from the generative model given X

2. Perform Gibbs sampling!
   1. Second update $\boldsymbol{y}$, fixed $\boldsymbol{w}, \theta$

$$p(y_{P,d} = k \,|\, \mathbf{y}_{P,\cancel{d}}, \mathbf{y}_{\mathbf{P}\setminus P,d}, \mathcal{X}, \mathbf{z}, \theta, \beta) = p(y_{P,d} = k \,|\, \mathbf{y}_{P,\cancel{d}}, \beta) p(\mathcal{X} \,|\, y_{P,d} = k, \mathbf{y}_{\mathbf{P}\setminus P,d}, \mathbf{z}, \theta)$$

$$p(y_{P,d} = k \,|\, \mathbf{y}_{P,\cancel{d}}, \beta) = \frac{\beta + m_{P,k}}{\sum_{j=1}^{|\mathbf{ch}(P)|} \beta + m_{P,k}}$$

$$m_{P,k} = \sum_{d \in \psi(P)\setminus d} \mathbb{1}[y_{P,d} = k]$$

Inference

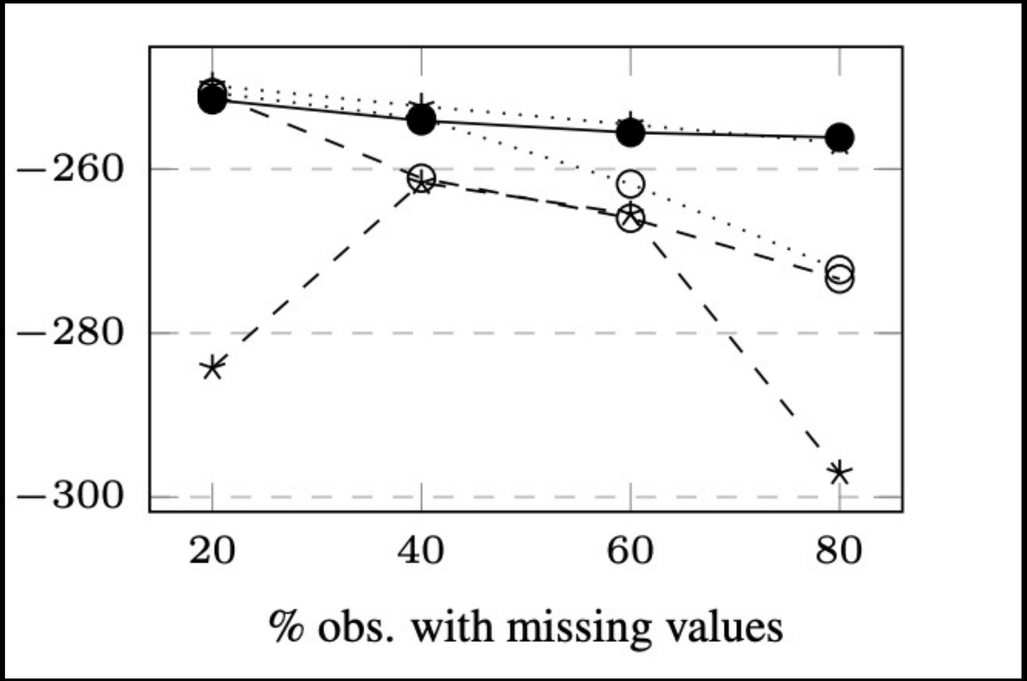
1. Given a set of T posterior samples

Inference

1. Given a set of T posterior samples

$$p(\mathbf{x}^* \mid \mathcal{X}) \approx \frac{1}{T} \sum_{t=1}^{T} \mathcal{S}(\mathbf{x}^* \mid \mathcal{G}, \psi_{\mathbf{y}^{(t)}}, \mathbf{w}^{(t)}, \theta^{(t)})$$

Experiments



Experiment results

Experiment results

Dataset	LearnSPN	RAT-SPN	CCCP	ID-SPN	ours	ours$^\infty$	BTD
NLTCS	−6.11	−6.01	−6.03	−6.02	**−6.00**	−6.02	−5.97
MSNBC	−6.11	−6.04	−6.05	−6.04	−6.06	**−6.03**	−6.03
KDD	−2.18	−2.13	−2.13	−2.13	**−2.12**	−2.13	−2.11
Plants	−12.98	−13.44	−12.87	**−12.54**	−12.68	−12.94	−11.84
Audio	−40.50	−39.96	−40.02	−39.79	**−39.77**	−39.79	−39.39
Jester	−53.48	−52.97	−52.88	−52.86	**−52.42**	−52.86	−51.29
Netflix	−57.33	−56.85	−56.78	−56.36	**−56.31**	−56.80	−55.71
Accidents	−30.04	−35.49	−27.70	**−26.98**	−34.10	−33.89	−26.98
Retail	−11.04	−10.91	−10.92	−10.85	−10.83	**−10.83**	−10.72
Pumsb-star	−24.78	−32.53	−24.23	**−22.41**	−31.34	−31.96	−22.41
DNA	−82.52	−97.23	−84.92	**−81.21**	−92.95	−92.84	−81.07
Kosarak	−10.99	−10.89	−10.88	**−10.60**	−10.74	−10.77	−10.52
MSWeb	−10.25	−10.12	−9.97	**−9.73**	−9.88	−9.89	−9.62
Book	−35.89	−34.68	−35.01	−34.14	**−34.13**	−34.34	−34.14
EachMovie	−52.49	−53.63	−52.56	−51.51	−51.66	**−50.94**	−50.34
WebKB	−158.20	−157.53	−157.49	**−151.84**	−156.02	−157.33	−149.20
Reuters-52	−85.07	−87.37	−84.63	**−83.35**	−84.31	−84.44	−81.87
20 Newsgrp	−155.93	−152.06	−153.21	**−151.47**	−151.99	−151.95	−151.02
BBC	−250.69	−252.14	**−248.60**	−248.93	−249.70	−254.69	−229.21
AD	−19.73	−48.47	−27.20	**−19.05**	−63.80	−63.80	−14.00

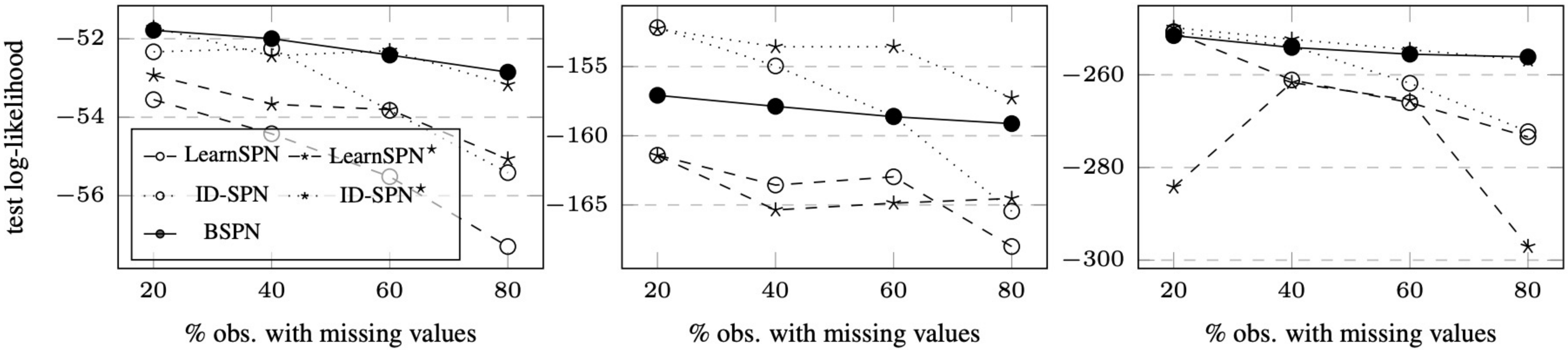
Experiment results

Experiment results

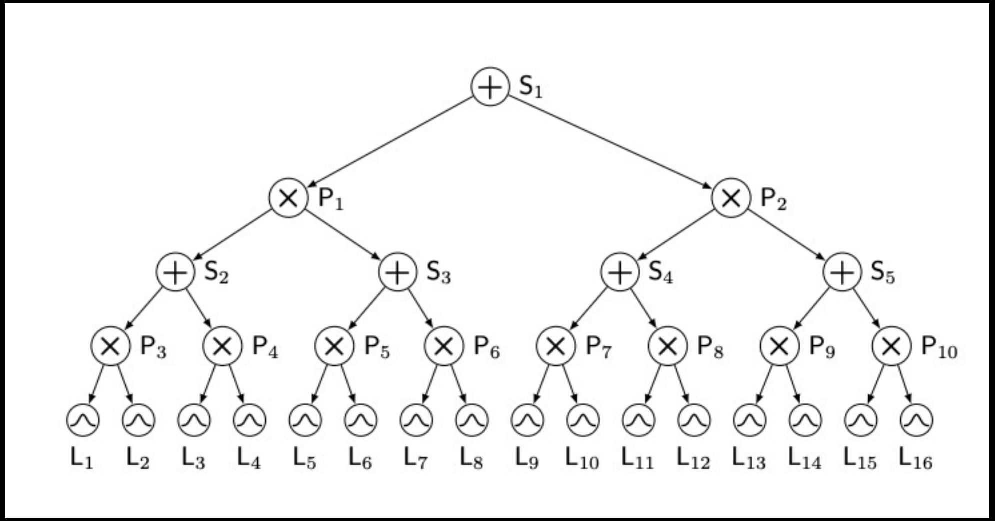
Dataset	MSPN	ABDA	ours	ours$^{\infty}$
Abalone	**9.73**	2.22	3.92	3.99
Adult	−44.07	−5.91	**−4.62**	−4.68
Australian	−36.14	**−16.44**	−21.51	−21.99
Autism	−39.20	−27.93	**−0.47**	−1.16
Breast	−28.01	−25.48	**−25.02**	−25.76
Chess	−13.01	−12.30	**−11.54**	−11.76
Crx	−36.26	**−12.82**	−19.38	−19.62
Dermatology	−27.71	−24.98	**−23.95**	−24.33
Diabetes	−31.22	**−17.48**	−21.21	−21.06
German	−26.05	**−25.83**	−26.76	−26.63
Student	−30.18	**−28.73**	−29.51	−29.9
Wine	**−0.13**	−10.12	−8.62	−8.65

Experiment results

Experiment results



Conclusion



S_1
P_1
P_2
S_2
S_3
S_4
S_5
P_3 P_4 P_5 P_6 P_7 P_8 P_9 P_{10}
L_1 L_2 L_3 L_4 L_5 L_6 L_7 L_8 L_9 L_{10} L_{11} L_{12} L_{13} L_{14} L_{15} L_{16}

Conclusion

1. Propose a novel and well-principled approach to SPN structure learning
    1. Decomposing the problem into finding a computational graph and learning a scope-function.

2. Propose a natural parametrisation for an important sub-type of SPNs
    1. Formulate a joint Bayesian framework simultaneously over structure and parameters

3. Bayesian SPNs are protected against overfitting
    1. Waiving the necessity of a separate validation set, which is beneficial for low data regimes

Thank you for your attention.