Bayesian Learning of Probabilistic Circuits with Domain Constraints

Athresh Karanam[*1] **Saurabh Mathur**[*1] **Sahil Sidheekh**[*1] **Sriraam Natarajan**[1]

[1]Erik Jonsson School of Engineering & Computer Science, University of Texas at Dallas, Richardson, Texas, USA

Abstract

Probabilistic Circuits (PCs) have emerged as an efficient framework for representing and learning complex probability distributions. However, existing research on PCs primarily focuses on data-driven parameter learning, with limited exploration of knowledge-intensive learning and structure learning. In this work, we propose to address these gaps by introducing a comprehensive approach to incorporating various kinds of domain knowledge into the learning of a PC's structure as well as its parameters.

1 INTRODUCTION

Recent developments in the field of tractable probabilistic models have enabled efficiently representing and learning complex probability distributions by parameterizing them in the form of computational graphs, commonly known as Probabilistic Circuits (PCs) [Choi et al., 2020b]. However, most of the works aimed at improving PCs focus on learning them using *data alone*. Further, they emphasize on building better algorithms and representations specifically tailored for *parameter learning* [Peharz et al., 2020b,a, Liu et al., 2023], assuming a fixed (often random) structure. While this has enabled building deep and expressive PCs, it has also made them data-hungry and susceptible to outliers Ventola et al. [2023].

However, in many real-world scenarios, the data is scarce or noisy. Incorporating domain knowledge into the learning process has been shown to be one effective strategy for learning better discriminative [Kokel et al., 2020, Odom et al., 2015] as well as generative models under such scenarios [Altendorf et al., 2005, de Campos et al., 2008, Yang and Natarajan, 2013]. Also note that when deploying PCs for decision-making in real life, several domains may inherently demand modeling constraints involving fairness and privileged information. However, knowledge-intensive learning of PCs is relatively less explored in the literature, and we aim to bridge this gap through this work.

When learning data distributions using probabilistic graphical models such as Bayesian Networks, or Markov Networks, a more accurate structure that encodes more valid independencies between variables is expected to be more data efficient [Koller and Friedman, 2009] than a random structure. Similarly, the underlying graph structure of a PC not only determines its expressive power, but also encodes factorizations of the joint distribution, which enables performing exact and tractable probabilistic inference over its random variables under appropriate constraints. Therefore, learning a better PC structure is another way to better model data distributions in the sparse regime.

However, compared to parameter learning, structure learning for PCs is a difficult and less explored problem. Most prevalent approaches to structure learning for PCs are based on heuristics [Gens and Pedro, 2013, Rooshenas and Lowd, 2014, Dang et al., 2020] and do not define a principled objective for structure learning [Adel et al., 2015, Peharz et al., 2013]. Principled approaches propose structure and parameter learning through a Bayesian approach [Trapp et al., 2019, Zhao et al., 2016]. Recently, Yang et al. [2023] proposed to use Bayesian structure scores for learning PC structure by marginalizing out its parameters from the joint likelihood under a Dirichlet prior. However, it was restricted to deterministic PCs, which constitute only a restricted subclass. In this work, we extend Bayesian structure learning to arbitrary PCs utilizing the notion of dropout Gal and Ghahramani [2016]. Nalisnick et al. [2019] demonstrated that dropout can be viewed as a structured prior over the model weights, Ventola et al. [2023] showed that the expected value of a PC node under the dropout distribution can be tractably computed. Thus, we propose to compute the Bayesian structure score for PCs by marginalizing out their parameters under a dropout induced posterior.

[*]Equal contribution

Accepted for the 6th Workshop on Tractable Probabilistic Modeling at UAI (TPM 2023).

Thus overall, in this work, we propose to integrate knowledge into the learning of PCs. To do so, we first develop a unified framework that allows encoding different types of domain knowledge. We then demonstrate how we can incorporate the encoded knowledge in the form of constraints to learn both the parameters as well as the structure of a PC, making them more adaptable to real-world scenarios with limited data and modeling constraints.

2 BACKGROUND

2.1 NOTATION

We use X to denote random variables and x to denote a value of X. Sets of random variables are denoted as $\mathbf{X}$ and their values as $\mathbf{x}$. We use $\mathcal{M} = (\theta, G)$ to denote a probabilistic circuit over variables $\mathbf{X} = \{X_1, \ldots, X_n\}$, having structure G and parameterized by θ. We use $P(\mathbf{X})$ to denote the joint probability distribution over $\mathbf{X}$.

2.2 PROBABILISTIC CIRCUITS

Probabilistic circuits (PCs, Choi et al. [2020a]) are models that represent probability distributions in the form of computational graphs. In this work, we consider a class of PCs called sum-product networks (SPNs, Poon and Domingos [2011]). The structure of an SPN consists of three kinds of nodes namely, sum nodes, product nodes, and leaf nodes. The sum nodes represent a mixture distribution of their children, the product nodes represent the factorized distribution over their children and the leaf nodes are univariate distributions.

In order for a PC to be a valid SPN, its structure must satisfy smoothness and decomposability conditions. Smoothness requires that the scope of each child of a sum node be identical. Decomposability requires that the scopes of the children of a product node be disjoint. These properties allow tractable computation of marginal and conditional probability queries in linear times in the size of the PC.

2.3 BAYESIAN INFERENCE IN PCs

We aim to learn the structure and parameters of a PC under the Bayesian framework. While this is tractable for deterministic PCs like cutset networks [Yang et al., 2023], it remains intractable for PCs in general.

Variational inference provides a fast approximation to Bayesian inference. Multiplicative noise like dropout induces a variational posterior distribution over the parameters of the PC. Recent work has shown that the posterior predictive distribution can be tractably computed for PCs with dropout[Ventola et al., 2023].

2.4 DOMAIN CONSTRAINTS

Real-world domains often require that models satisfy validity constraints. These constraints concisely encode information about general trends in the domain. As a result, they can function as an inductive bias, yielding more useful and more accurate probabilistic models, especially in noisy and sparse domains [Altendorf et al., 2005, Kokel et al., 2020, Towell and Shavlik, 1994, Yang and Natarajan, 2013]

While probabilistic constraints have been used to learn SPNs, they were limited to parameter learning [Papantonis and Belle, 2021]. Recently qualitative influence based constraints were used to learn the structure and parameters of cutset networks [Mathur et al., 2023]. However, since cutset networks have the additional property of determinism, the same method cannot be used for SPNs. As far as we are aware domain constraints have not been used to learn both structure and parameters of PCs.

Concretely, we consider 6 types of domain constraints namely, generalization, qualitative influence [Altendorf et al., 2005], context-specific independence [Boutilier et al., 1996], class imbalance [Yang et al., 2014], metric fairness[Dwork et al., 2012], and privileged information [Pasunuri et al., 2016].

3 LEARNING PCs WITH DOMAIN CONSTRAINTS

We aim to solve the following problem,

Given: Dataset $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$ over random variables $\mathbf{X}$ and a set of linear constraints over conditional queries C
To Do: Learn a probabilistic circuit $\mathcal{M}$ that accurately models $P(\mathbf{X})$

3.1 LINEAR CONSTRAINTS

In this work, we focus on constraints that can be expressed as linear functions of conditional probability queries. Formally, we denote a conditional probability query over the PC $\mathcal{M}$ as $P(f(\mathbf{x}_q) = 1 \mid g(\mathbf{x}_q) = 1)$. Here, $\mathbf{x}_q \in dom(\mathbf{X}_q),\ \mathbf{X}_q \subseteq \mathbf{X}$. f and g are boolean functions. We use the shorthand notation $P(f(\mathbf{x}_q) \mid g(\mathbf{x}_q))$ to refer to these queries. Linear functions of these queries can represent a wide range of constraints including monotonicity, synergy, class imbalance, and fairness. We group these constraints into two categories – equality constraints and inequality constraints.

3.1.1 Equality constraints

Equality constraints require that two conditional probability queries be equal. Specifically,

$$P(f_1(\mathbf{x}_q) \mid g_1(\mathbf{x}_q)) = P(f_2(\mathbf{x}'_q) \mid g_2(\mathbf{x}'_q))$$

We encode these as the penalty $\gamma(G, \theta) = \sum_x \delta(x)^2$ where,

$$\delta(x) = (P(f_1(\mathbf{x}_q) \mid g_1(\mathbf{x}_q)) - P(f_2(\mathbf{x}'_q) \mid g_2(\mathbf{x}'_q)).$$

Example The generalization constraint is an example of an equality constraint. It is defined as

$$P(x_i \mid x_{-i}) = P(x'_i \mid x'_{-i})$$
$$\forall \mathbf{x}, \mathbf{x}' \in \mathcal{D} \text{ s.t. } sim(\mathbf{x}, \mathbf{x}')$$

where $sim(\mathbf{x}, \mathbf{x}')$ is true if $\mathbf{x}$ and $\mathbf{x}'$ are known to be similar. This can be encoded using $f_1(\mathbf{x}) = f_2(\mathbf{x}) = \mathbb{I}[X_i = x_i]$ and $g_1(\mathbf{x}) = g_2(\mathbf{x}) = \mathbb{I}[X_{-i} = x_{-i}]$.

3.1.2 Inequality constraints

Inequality constraints require one or more conditional probability query to be greater than (or lesser than) others. Such constraints with two terms are of the form

$$P(f_1(\mathbf{x}_q) \mid g_1(\mathbf{x}_q)) > P(f_2(\mathbf{x}'_q) \mid g_2(\mathbf{x}'_q)).$$

We encode these as the penalty $\zeta^+(G, \theta) = \sum_x \max\{0, \delta^+(x)\}^2$ where,

$$\delta^+(x) = P(f_2(\mathbf{x}'_q) \mid g_2(\mathbf{x}'_q)) - P(f_1(\mathbf{x}_q) \mid g_1(\mathbf{x}_q)) + \epsilon$$
$$\exists\, \epsilon > 0$$

Example The positive monotonicity constraint is an example of an inequality constraint. The positive monotonic constraint $X_j \overset{M+}{\prec} X_i$ is defined as

$$P(X_i \leq x_i \mid x_j) > P(X_i \leq x'_i \mid x'_j)$$
$$\forall \mathbf{x}, \mathbf{x}' \text{ s.t. } x_i = x'_i,\ x'_j > x_j$$

This can be encoded using $f_1(x_i) = f_2(x_i) = \mathbb{I}[X_i \leq x_i]$ and $g_1(x_j) = g_2(x_j) = \mathbb{I}[X'_j = x_j]$.

Note that higher-order constraints like synergies [Yang and Natarajan, 2013] can be encoded similarly as they are still linear functions. For eg, $X_j, X_k \overset{S+}{\prec} X_i$ is expressed as

$$P(X_i \leq x'_i \mid x'_j, x_k) + P(X_i \leq x'_i \mid x_j, x'_k) >$$
$$P(X_i \leq x_i \mid x_j, x_k) + P(X_i \leq x_i \mid x'_j, x'_k)$$
$$\forall \mathbf{x}, \mathbf{x}' \text{ s.t. } x_i = x'_i,\ x'_j > x_j, x'_k > x_k$$

Tables 2 shows the way to encode the remaining constraints and Table 1 summarizes the domain sets over which the constraints are evaluated.

[1]context is defined by boolean function $K(x)$

3.2 LEARNING

The PC learning task consists of two subtasks, namely Parameter Learning and Structure Learning.

3.2.1 Structure Learning

Formally, the structure learning task given data set $\mathcal{D}$ and constraints C is

$$\arg\max_G \mathcal{S}(G; \mathcal{D}) \text{ s.t } C \tag{1}$$

where $\mathcal{S}(G; \mathcal{D})$ is the score of the structure G with respect to data set $\mathcal{D}$.

We approach the structure learning task under the Bayesian framework. Bayesian structure scores are a principled way to compare two structures without committing to specific parameter values. Formally, they are defined as the marginal likelihood $P(D \mid G) = \mathbb{E}_{\theta \sim p}[P(D, \theta \mid G)]$ where p is the prior distribution over the parameters θ.

Similarly, we define structure penalty functions γ_S and ζ_S over the structure of the PC by marginalizing the parameters θ.

$$\zeta_S(G) = \mathbb{E}_{\theta \sim p}[\zeta(G, \theta)]$$
$$\gamma_S(G) = \mathbb{E}_{\theta \sim p}[\gamma(G, \theta)].$$

This allows us to define a penalized structure score

$$\underbrace{P(D \mid G)}_{\text{Data}} - \lambda \underbrace{(\zeta_S(G) + \gamma_S(G))}_{\text{Knowledge}}. \tag{2}$$

Here, λ is a hyper-parameter that controls the weight of the penalty term.

Computing expectations Computing the structure score efficiently requires a tractable prior distribution over the parameters θ. The structure score is tractable for the specific case of smooth, decomposable, and deterministic PCs when the prior on the parameters is a Dirichlet distribution Yang et al. [2023]. However, it remains intractable for a broader class of PCs like SPNs which are smooth and decomposable but may not be deterministic. In order to compute the structure score efficiently, we use a variational approximation to the posterior. Specifically, we use the variational posterior induced by dropout because it allows tractable inference [Ventola et al., 2023]. Let this posterior be q. This allows us to compute the first term efficiently as

$$P(D \mid G) = \mathbb{E}_{\theta \sim p}[P(D, \theta \mid G)]$$
$$\approx \mathbb{E}_{\theta \sim q}[P(D, \theta \mid G)]$$

Since the second term is still intractable, we derive a lower

Name	Domain set
Monotonicity $X_i \overset{M+}{\prec} X_j$	$\{(\mathbf{x}, \mathbf{x}') \mid (x_j = x'_j) \wedge (x_i > x'_i) \wedge (x_{-ij} = C)\ \forall \mathbf{x}, \mathbf{x}' \in \text{Dom}(\mathbf{X})^2\}$
Class imbalance (FP)	$\{(\mathbf{x}, \mathbf{x}) \mid x_t = 0\ \forall \mathbf{x} \in \mathcal{D}\}$
Context-specific independence[1]	$\{(\mathbf{x}, \mathbf{x}') \mid K(x)\ \forall \mathbf{x}, \mathbf{x}' \in \text{Dom}(\mathbf{X})^2$
Privileged information	$\{(\mathbf{x}, \mathbf{x}) \mid\ \forall x \in \mathcal{D}\}$
Generalization	$\{(\mathbf{x}, \mathbf{x}') \mid sim(\mathbf{x}, \mathbf{x}')\ \forall \mathbf{x}, \mathbf{x}' \in \mathcal{D}^2\}$
Metric fairness	$\{(\mathbf{x}, \mathbf{x}') \mid sim(\mathbf{x}, \mathbf{x}')\ \forall \mathbf{x}, \mathbf{x}' \in \mathcal{D}^2\}$

Table 1: Domain sets for various penalties

Name	$f_1(\mathbf{x})$	$g_1(\mathbf{x})$	$f_2(\mathbf{x})$	$g_2(\mathbf{x})$
Monotonicity	$\mathbb{I}[X_i \leq x_i]$	$\mathbb{I}[X_j = x_j]$	$\mathbb{I}[X_i \leq x_i]$	$\mathbb{I}[X_j = x_j]$
Class imbalance (FP)	$\mathbb{I}[X_t = 1]$	$\mathbb{I}[X_{-t} = x_{-t}]$	$\mathbb{I}[X_t = 0]$	$\mathbb{I}[X_{-t} = x_{-t}]$
Context-specific independence[1]	$\mathbb{I}[X_i = x_i]$	$\mathbb{I}[X_j = x_j]K(x)$	$\mathbb{I}[X_i = x_i]$	$K(x)$
Privileged information	$\mathbb{I}[X_i = 1]$	$\mathbb{I}[X_{-i} = x_{-i}]$	$\mathbb{I}[X_i = 1]$	$\mathbb{I}[\mathbf{X}_{\text{obs.}} = \mathbf{x}_{\text{obs.}}]$
Generalization	$\mathbb{I}[X_i = 1]$	$\mathbb{I}[X_{-i} = x_{-i}]$	$\mathbb{I}[X_i = 1]$	$\mathbb{I}[X_{-i} = x_{-i}]$
Metric fairness (Acc.)	$\mathbb{I}[X_i = x_i]$	$\mathbb{I}[X_{-i} = x_{-i}]$	$\mathbb{I}[X_i = x_i]$	$\mathbb{I}[X_{-i} = x_{-i}]$

Table 2: Boolean functions f_1, g_1, f_2, g_2

bound for it using Jensen's inequality. Consider $\zeta_S(G)$

$$\zeta_S(G) = \sum_x \mathbb{E}_{\theta \sim q}[\max\{0, \delta(x)\}^2]$$
$$\geq \sum_x \max\{0, \mathbb{E}_{\theta \sim q}[\delta(x)]\}^2 = \hat{\zeta}_S(G).$$

Similarly, $\gamma_S(G)$ is lower bounded as

$$\gamma_S(G) \geq \sum_x \mathbb{E}_{\theta \sim q}[\delta(x)]^2 = \hat{\gamma}_S(G).$$

Hence, the penalized structure score can be approximated by the expression

$$\mathbb{E}_{\theta \sim q}[P(D, \theta \mid G)] - \lambda(\hat{\zeta}_S(G) + \hat{\gamma}_S(G)).$$

Structure learning using this score can be performed by a standard search algorithm like hill-climbing search.

3.2.2 Parameter Learning

Formally, the parameter learning task given structure G, data set $\mathcal{D}$ and constraints C is

$$\arg\max_\theta \mathcal{L}(\theta; G, \mathcal{D}) \text{ s.t } C \tag{3}$$

where $\mathcal{L}$ is a data-dependent objective function. When $\mathcal{L}$ is the likelihood, the above optimization problem becomes a constrained maximum likelihood problem. In this work, we set $\mathcal{L}$ to $\hat{\mathcal{L}}$, the lower bound of the KL divergence between the approximate dropout posterior q and the true posterior. Additionally, we use the penalty functions γ and ζ to write the above problem as

$$\arg\max_\theta \hat{\mathcal{L}}(\theta; G, \mathcal{D}) - \lambda(\zeta(G, \theta) + \gamma(G, \theta)) \tag{4}$$

Given a penalty weight $\lambda > 0$, this optimization problem can be solved using a standard gradient-based optimizer. Additionally, increasing the value of λ until the penalty term vanishes yields a solution to (3) in the limit [Bertsekas, 1996].

4 CONCLUSION

We propose a unified framework for integrating probabilistic domain constraints into PC structure and parameter learning. We show that different instantiations of our framework correspond to domain constraints represented as linear functions of conditional probability queries and consider 6 domain constraints and their corresponding instantiations. Furthermore, we present knowledge-intensive structure and parameter learning of PCs that incorporate the aforementioned domain constraints.

Our future work includes empirically validating the effectiveness of our proposed framework, incorporating more domain constraints, and finally, comparing our approach to existing knowledge-intensive learning methods. Additionally, a direction of future work is evaluating the impact of different priors on parameter and structure learning in our framework.

Acknowledgements

The authors acknowledge the support by the NIH grant R01HD101246 and ARO award W911NF2010224.

References

Tameem Adel, David Balduzzi, and Ali Ghodsi. Learning the structure of sum-product networks via an svd-based algorithm. In *Conference on Uncertainty in Artificial Intelligence*, 2015.

Eric Altendorf, Angelo C. Restificar, and Thomas G. Dietterich. Learning from sparse data by exploiting monotonicity constraints. In *UAI*, pages 18–26. AUAI Press, 2005.

Dimitri P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods (Optimization and Neural Computation Series)*. Athena Scientific, 1 edition, 1996. ISBN 1886529043.

Craig Boutilier, Nir Friedman, Moises Goldszmidt, and Daphne Koller. Context-specific independence in bayesian networks. In *UAI*, 1996.

YooJung Choi, Antonio Vergari, and Guy Van den Broeck. Lecture notes: Probabilistic circuits: Representation and inference. February 2020a. URL `http://starai.cs.ucla.edu/papers/LecNoAAAI20.pdf`.

YooJung Choi, Antonio Vergari, and Guy Van den Broeck. Probabilistic circuits: A unifying framework for tractable probabilistic models. 2020b.

Meihua Dang, Antonio Vergari, and Guy Van den Broeck. Strudel: Learning structured-decomposable probabilistic circuits. In *Proceedings of the 10th International Conference on Probabilistic Graphical Models*, volume 138 of *Proceedings of Machine Learning Research*, pages 137–148. PMLR, 23–25 Sep 2020.

Cassio P. de Campos, Yan Tong, and Qiang Ji. Constrained maximum likelihood learning of bayesian networks for facial action recognition. In *ECCV (3)*, volume 5304 of *Lecture Notes in Computer Science*, pages 168–181. Springer, 2008.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S. Zemel. Fairness through awareness. In *ITCS*, pages 214–226. ACM, 2012.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

Robert Gens and Domingos Pedro. Learning the structure of sum-product networks. In *International conference on machine learning*, pages 873–880. PMLR, 2013.

Harsha Kokel, Phillip Odom, Shuo Yang, and Sriraam Natarajan. A unified framework for knowledge intensive gradient boosting: Leveraging human experts for noisy sparse domains. In *AAAI*, pages 4460–4468. AAAI Press, 2020.

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.

Anji Liu, Honghua Zhang, and Guy Van den Broeck. Scaling up probabilistic circuits by latent variable distillation. In *The Eleventh International Conference on Learning Representations*, 2023.

Saurabh Mathur, Vibhav Gogate, and Sriraam Natarajan. Knowledge intensive learning of cutset networks. In *Conference on Uncertainty in Artificial Intelligence*, 2023.

Eric Nalisnick, José Miguel Hernández-Lobato, and Padhraic Smyth. Dropout as a structured shrinkage prior. In *International Conference on Machine Learning*, pages 4712–4722. PMLR, 2019.

Phillip Odom, Tushar Khot, Reid Porter, and Sriraam Natarajan. Knowledge-based probabilistic logic learning. In *AAAI*, pages 3564–3570. AAAI Press, 2015.

Ioannis Papantonis and Vaishak Belle. Closed-form results for prior constraints in sum-product networks. *Frontiers Artif. Intell.*, 4:644062, 2021.

Rahul Pasunuri, Phillip Odom, Tushar Khot, Kristian Kersting, and Sriraam Natarajan. Learning with privileged information: Decision-trees and boosting. In *Proc. Int. Joint Conf. Artif. Intell. Workshop*, pages 1–7, 2016.

Robert Peharz, Bernhard C. Geiger, and Franz Pernkopf. Greedy part-wise learning of sum-product networks. In Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Železný, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 612–627, 2013.

Robert Peharz, Steven Lang, Antonio Vergari, Karl Stelzner, Alejandro Molina, Martin Trapp, Guy Van den Broeck, Kristian Kersting, and Zoubin Ghahramani. Einsum networks: Fast and scalable learning of tractable probabilistic circuits. In *ICML*, 2020a.

Robert Peharz, Antonio Vergari, Karl Stelzner, Alejandro Molina, Xiaoting Shao, Martin Trapp, Kristian Kersting, and Zoubin Ghahramani. Random sum-product networks: A simple and effective approach to probabilistic deep learning. In *UAI*, 2020b.

Hoifung Poon and Pedro Domingos. Sum-product networks: A new deep architecture. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 689–690. IEEE, 2011.

Amirmohammad Rooshenas and Daniel Lowd. Learning sum-product networks with direct and indirect variable interactions. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 710–718, Bejing, China, 22–24 Jun 2014. PMLR.

Geoffrey G. Towell and Jude W. Shavlik. Knowledge-based artificial neural networks. *Artif. Intell.*, 70(1–2):119–165, oct 1994. ISSN 0004-3702.

Martin Trapp, Robert Peharz, Hong Ge, Franz Pernkopf, and Zoubin Ghahramani. Bayesian learning of sum-product networks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Fabrizio Ventola, Steven Braun, Zhongjie Yu, Martin Mundt, and Kristian Kersting. Probabilistic circuits that know what they don't know. *arXiv e-prints*, pages arXiv–2302, 2023.

Shuo Yang and Sriraam Natarajan. Knowledge intensive learning: Combining qualitative constraints with causal independence for parameter learning in probabilistic models. In *ECML/PKDD (2)*, volume 8189 of *Lecture Notes in Computer Science*, pages 580–595. Springer, 2013.

Shuo Yang, Tushar Khot, Kristian Kersting, Gautam Kunapuli, Kris Hauser, and Sriraam Natarajan. Learning from imbalanced data in relational domains: A soft margin approach. In *ICDM*, 2014.

Yang Yang, Gennaro Gala, and Robert Peharz. Bayesian structure scores for probabilistic circuits. In *International Conference on Artificial Intelligence and Statistics*, pages 563–575. PMLR, 2023.

Han Zhao, Tameem Adel, Geoff Gordon, and Brandon Amos. Collapsed variational inference for sum-product networks. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1310–1318, New York, New York, USA, 20–22 Jun 2016. PMLR. URL `https://proceedings.mlr.press/v48/zhaoa16.html`.