Knowledge Intensive Learning of Cutset Networks

Saurabh Mathur[1] **Vibhav Gogate**[1] **Sriraam Natarajan**[1]

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

Abstract

Cutset networks (CNs) are interpretable probabilistic representations that combine probability trees and tree Bayesian networks, to model and reason about large multi-dimensional probability distributions. Motivated by high-stakes applications in domains such as healthcare where (a) rich domain knowledge in the form of qualitative influences is readily available and (b) use of interpretable models that the user can efficiently probe and infer over is often necessary, we focus on learning CNs in the presence of qualitative influences. We propose a penalized objective function that uses the influences as constraints, and develop a gradient-based learning algorithm, KICN. We show that because CNs are tractable, KICN is guaranteed to converge to a local maximum of the penalized objective function. Our experiments on several benchmark data sets show that our new algorithm is superior to the state-of-the-art, especially when the data is scarce or noisy.

1 INTRODUCTION

Recently, there has been a growing interest in learning tractable probabilistic models (TPMs) or probabilistic circuits [Choi et al., 2020] from data. The key advantage of these models is that they admit tractable, and in most cases, linear time *exact* probabilistic inference as opposed to traditional probabilistic graphical models such as Bayesian and Markov networks (BNs and MNs) which require the use of approximate inference methods. We consider a restricted class of TPMs called *cutset networks* [Rahman et al., 2014] that are inspired from Pearl's cutset conditioning [Pearl, 1988, Bidyuk and Dechter, 2004]. These are essentially a combination of OR trees and tree BNs where the leaves of the OR tree are tree BNs. Cutset networks are tractable in that many reasoning queries such as computing the marginal probability over a subset of variables given observations and finding the most likely explanation for evidence can be solved in time that scales linearly in the size of the network. Another key virtue of cutset networks is that, unlike state-of-the-art TPMs such as arithmetic circuits [Darwiche, 2003] and sum-product networks [Poon and Domingos, 2011], they are also interpretable [Rahman et al., 2019]—another key property that is necessary for models used in high-stakes applications.

Currently, state-of-the-art algorithms for learning *tractable, interpretable* cutset networks [Rahman et al., 2014, 2019, Mauro et al., 2015] use training data alone. Recently, there has been a surge in developing systems that can effectively use human inputs that range from decision boundary constraints [Fung et al., 2002, Towell and Shavlik, 1994, Kunapuli et al., 2010, 2013], label preferences [Odom et al., 2015], misclassification costs [Yang et al., 2014], privileged information [Vapnik and Vashist, 2009, Sharmanska et al., 2013] and qualitative influences [Altendorf et al., 2005, Yang and Natarajan, 2013, Kokel et al., 2020]. Our key hypothesis in this work is that such knowledge can potentially allow for effective learning of cutset networks in settings where data is scarce or noisy.

Specifically, we consider a type of qualitative influence called *monotonicities*, as an *inductive bias* when learning cutset networks. We consider this task in the context of a real-world problem, that of modeling gestational diabetes from a clinical study [Haas et al., 2015], a high-stakes application where using tractable, interpretable models such as cutset networks is necessary. The monotonicities obtained from the domain expert (a physician in our case) will serve as constraints on the model and allow for learning a more robust model. We develop a novel learning framework, *Knowledge Intensive Learning of Cutset Networks* (KICN) that enforces these constraints during either structure or parameter learning. Consequently, we present two variations of our learning algorithm – one for parameter learning and one for structure learning.

Proceedings of the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023), PMLR 216:1380–1389.

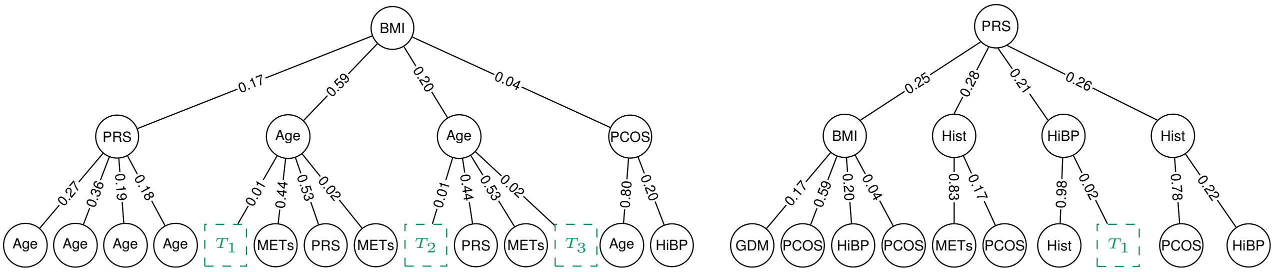

Figure 1: The first 3 levels of cutset networks (CNs) learned from the nuMoM2b-b dataset using LearnCNet (left) and using KICN (right). While Polygenic Risk Score (PRS) is a very important risk factor for Gestational Diabetes in the literature [Pagel et al., 2022], the CN learned from data has it in second and third levels. Combining data with monotonicities obtained from the domain expert allows the CN on the right to select PRS as the root node. Moreover, the CN on the right is more concise (and thus more interpretable).

To understand the impact of knowledge in learning, consider the learned models in Figure 1. The model on the left is learned from data alone while the one on the right is learned from data and knowledge for modeling gestational diabetes [Haas et al., 2015]. We observe that while the model learned from data alone uses BMI as the top feature in modeling gestational diabetes, the model learned using both data and domain knowledge from clinicians uses polygenic risk scores (PRS), a top risk factor for gestational diabetes according to literature [Pagel et al., 2022]. These are precisely the type of models that we aim to learn using our KICN algorithm.

We make the following key contributions: (1) As far as we are aware, we present the first work on employing rich qualitative information as inductive biases when learning tractable probabilistic models. (2) We develop an efficient learning framework (KICN) that uses this qualitative information as constraints during learning. (3) We outline a few variations of this framework based on where and how the constraints are enforced. (4) We perform extensive evaluation of the proposed framework on many standard data sets and demonstrate the superiority of the proposed algorithm on two evaluation measures: test set log-likelihood score and mean-squared error on conditional probability queries. Most importantly, we present results on a real, high-impact, gestational diabetes data set where these learned models allow for interpretability and hence, can result in building effective treatment plans.

2 BACKGROUND

2.1 CUTSET NETWORKS

Cutset networks [Rahman et al., 2014] are a class of tractable probabilistic models that compactly represent large multi-dimensional joint probability distributions. They combine two interpretable and tractable representations: OR probability trees and tree BNs.

Formally, given a set of variables $X = \{X_1, \ldots, X_n\}$, a cutset network is defined as a pair $\mathcal{M} = (O, T)$ where O is an OR tree having l leaves where each OR node is labeled with a variable $X_i \in X$ and $T = \{T_1, \ldots, T_l\}$ is a set of tree BNs such that $T_j \in T$ is associated with the j-th leaf node of O. Similar to decision trees, we assume that each variable $X_i \in X$ appears *at most once* on the path from the root to a leaf node in O. OR nodes in O represent conditioning and each OR node labeled with X_i has $|domain(X_i)|$ children, one for each value in $domain(X_i)$. Each edge from a parent OR node labeled with $X_i \in X$ and a child OR node labeled with $X_j \in X$ (or a leaf node T_k) is labeled with the conditional probability of X_i taking the corresponding value given the assignment from the root note to the parent. Each tree BN $T_j \in T$ represents the conditional probability distribution over all variables from the set X that are not included in the OR nodes on the path from the root node to T_j given the assignment on the path from the root node to T_j.

Given an assignment (data-point) $x = (x_1, \ldots, x_n)$ to all variables in the set X, let $z = l(x)$ be the leaf node corresponding to x, $path_O(z)$ be the path from the root of the OR tree O to the leaf z, V_z be the variables in X that are not included on the OR nodes in $path_O(z)$, and W_z be the set of conditional probability labels on the edges in $path_O(z)$. Then, the joint probability distribution induced by the cutset network $\mathcal{M}$ is

$$P_{\mathcal{M}}(x) = \left(\prod_{w \in W_z} w\right) P^z(x_{V_z}) \tag{1}$$

where x_{V_z} is the projection of x on the subset V_z of X. We assume that each tree BN $T_z \in T$ is defined by the parent map $\mathrm{Pa}^z : V_z \mapsto V_z$ and parameters θ^z as $T_z = (\mathrm{Pa}^z, \theta^z)$. Further, we use the shorthand Pa_i^z to refer to $\mathrm{Pa}^z(X_i)$. Then, the probability distribution at each leaf is

$$P^z(x_{V_z}) = \prod_{X_i \in V_z} P_i^z(x_i \mid x_{\mathrm{Pa}_i^z}) = \prod_{X_i \in V_z} \theta_{ijk}^z$$

where θ_{ijk}^z is the conditional probability that the random variable X_i at the leaf z has the value k given that its par-

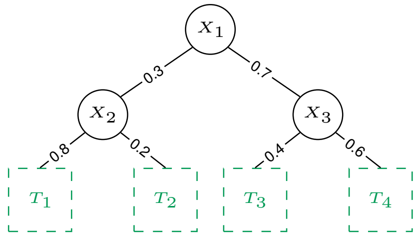

Figure 2: A Cutset network over 5 binary random variables $\{X_1, \ldots, X_5\}$. The children for each node are shown in ascending order of their corresponding values. Here, $T_1, \ldots, T_4$ are tree Bayesian Networks over the variables not included on the path from the root. Each edge label represents a conditional probability and can be interpreted as the proportion of data points belonging to the corresponding data partition.

ent Pa_i^z has the value j. Let W and θ denote the set of parameters associated with O and T respectively.

Cutset networks are learned using the LearnCNet algorithm [Rahman et al., 2014, Mauro et al., 2015]. It works by recursively splitting the given data on a heuristically selected variable until a termination condition is reached. For example, termination conditions could be defined in terms of the number of remaining variables or the amount of remaining data or both. When a termination condition is met, the Chow-Liu algorithm [Chow and Liu, 1968] is used to fit a tree BN. The most widely used heuristic for variable selection is the maximum pairwise mutual information score heuristic where we select a variable having the maximum sum pairwise mutual information.

Figure 2 shows a cutset network defined over a set of 5 binary valued random variables $\{X_1, \ldots, X_5\}$. The nodes of the OR tree are shown as circular nodes, while tree BNs are shown as dashed square nodes labeled $T_1, \ldots, T_4$. T_1 and T_2 represent conditional probability distribution over $\{X_3, X_4, X_5\}$ given the assignments $(X_1 = 0, X_2 = 0)$ and $(X_1 = 0, X_2 = 1)$, respectively. T_3 and T_4 represent the conditional probability distribution over $\{X_2, X_4, X_5\}$ given the assignments $(X_1 = 1, X_3 = 0)$ and $(X_1 = 1, X_3 = 1)$, respectively.

Cutset networks are tractable probabilistic models [Rahman et al., 2014]. Queries such as finding the most probable explanation and computing the marginal probability distribution at each variable given observations can be answered in time that scales linearly with the size (number of parameters) of the network. In addition to being tractable, Cutset networks are also interpretable [Rahman et al., 2019] because they consist of OR-trees and tree BNs. OR-trees are structured like probabilistic decision trees and hence inherit their interpretability. Tree BNs at the leaves of the OR-tree are interpretable because all of their nodes correspond to observed variables. The parameters of both the OR-tree and the tree BNs have probabilistic interpretations because they are conditional probabilities of observed variables. For example, in Figure 2, the path from the root node to T_1 can be interpreted as follows. The probability that X_1 takes the value 0 is 0.3; given $X_1 = 0$, the probability that X_2 takes the value 0 is 0.8; given $(X_1 = 0, X_2 = 0)$ the conditional distribution over the remaining variables is given by a tree BN T_1. These two properties – tractability and interpretability – make cutset networks **a natural fit for high-stakes domains like healthcare** that require the models to be interpretable while being able to answer certain queries exactly in order to build trust with the domain expert [Rudin, 2019].

2.2 QUALITATIVE INFLUENCES IN PROBABILISTIC MODELS

We approach the problem of learning interpretable and tractable probabilistic models for high-stakes domains by using qualitative influences given by a domain expert as an inductive bias. As far as we are aware, this is the first work on learning interpretable tractable probabilistic models using qualitative influences. Qualitative influences have been previously used for learning probabilistic models [Wellman, 1990]. For instance, they have been used to learn more accurate discriminative models in the presence of noisy and sparse data [Kokel et al., 2020, Odom et al., 2015]. However, their use in learning generative models has been limited to BNs [Altendorf et al., 2005, de Campos et al., 2008, Yang and Natarajan, 2013], where exact inference is intractable in general [Cooper, 1990].

Additionally, while generative models like Sum-Product Networks (SPNs) guarantee tractable inference [Poon and Domingos, 2011], they are hard to explain because their internal nodes do not correspond to any observed variables. The probabilistic prior constraints that have been used to learn SPNs [Papantonis and Belle, 2021] are harder to elicit from experts. In contrast, the structure of cutset networks makes it natural to encode a variety of domain knowledge such as conditional independences, context-specific independences, deterministic constraints, and quantitative constraints [Chavira and Darwiche, 2007, Gogate and Domingos, 2010, Rahman et al., 2014]. In this work, we propose to learn cutset networks using qualitative influences which are easier to elicit from experts in domains like healthcare.

Concretely, we consider a specific type of qualitative influence called *monotonic influence* [Altendorf et al., 2005]. A random variable X_j is said to positively monotonically influence another random variable X_i if an increase in the value of X_j increases the probability of higher values of X_i. We use $X_j \overset{M+}{\prec} X_i$ to denote a positive monotonic influence. Similarly, a negative monotonic influence $X_j \overset{M-}{\prec} X_i$ implies that an increase in the values of X_j decreases the probability of higher values of X_i.

The positive and negative influences can be expressed respectively using the following constraints:

$$P_{\mathcal{M}}(X_i \leq c \mid X_j = a) \leq P_{\mathcal{M}}(X_i \leq c \mid X_j = b) \quad \forall a > b;\ a, b \in domain(X_j);\ c \in domain(X_i) \tag{2}$$

$$P_{\mathcal{M}}(X_i \leq c \mid X_j = a) \geq P_{\mathcal{M}}(X_i \leq c \mid X_j = b) \quad \forall a > b;\ a, b \in domain(X_j);\ c \in domain(X_i) \tag{3}$$

This form of monotonic influence relation has been expressed in prior work [Altendorf et al., 2005] in the context of BNs for the case where X_j is a parent of X_i. This work on BNs was later extended to learn conditional distributions with causal independence and qualitative constraints [Yang and Natarajan, 2013]. These relations were used as margin constraints to learn conditional probability tables. In our work, instead of using monotonic influences as constraints on conditional distributions, we use them as constraints on the joint distribution.

3 KNOWLEDGE INTENSIVE LEARNING OF CUTSET NETWORKS

We hypothesize that knowledge in the form of monotonic influence statements integrates well with the patterns learned from data in a cutset network, producing more accurate and concise (and hence more interpretable) models. To test our hypothesis, we propose Algorithm KICN, which solves the following problem:

Given: Dataset $\mathcal{D} = \{x^{(i)}\}_{i=1}^{N}$ over random variables X and a set of qualitative influences C
To Do: Learn a cutset network $\mathcal{M}$

Mathematically, the above problem can be expressed as the following constrained optimization problem:

$$\arg\max_{\mathcal{M}} \mathcal{L}(\mathcal{M}, \mathcal{D}) \quad \text{s.t. constraints in } C \tag{4}$$

where $\mathcal{L}(\mathcal{M}, \mathcal{D})$ is the log-likelihood of $\mathcal{D}$ w.r.t. $\mathcal{M} = (O, T)$ and is given by

$$\mathcal{L}(\mathcal{M}, \mathcal{D}) = \sum_{x \in \mathcal{D}} \left(\log P^z(x_{V_z}) + \sum_{w \in W_z} \log w \right) \tag{5}$$

where $z = l(x)$ is the leaf node corresponding to x. An issue with the constrained optimization formulation given in Eq. (4) is that it may not have any feasible solutions. For example, since the qualitative influence statements are elicited from a domain expert, they are *not guaranteed to be consistent with each other*. As a result, exact constraint satisfaction *may not be possible*. To address this issue, we propose an algorithm to incorporate the qualitative influences in the parameters of Tree BNs. Finally, we propose the KICN framework which adapts the LearnCNet framework to leverage qualitative influences for both parameter learning and structure learning.

3.1 LEARNING TREE BN PARAMETERS

Consider a tree BN T_z having scope V_z. Let $\mathcal{D}_z$ be a dataset over V_z and C_z be a set of qualitative influences over V_z. Given the structure of T_z defined by Pa^z, we define the optimization problem for parameter learning using qualitative influences as

$$\arg\max_{\theta^z} \mathcal{L}(T_z, \mathcal{D}_z) \quad \text{s.t. constraints in } C_z \tag{6}$$

Here, $\mathcal{L}_z(T_z, \mathcal{D}_z)$ is the log likelihood function and is

$$\begin{aligned} \mathcal{L}_z(T_z, \mathcal{D}_z) &= \sum_{x \in \mathcal{D}_z} \log P^z(x_{V_z}) \\ &= \sum_{x \in \mathcal{D}_z} \sum_{X_i \in V_z} \log \theta^z_{ijk} \end{aligned} \tag{7}$$

where $k = x_i$ and $j = x_{pa^z_i}$.

Inspired by prior work of Altendorf et al. [2005], we define the following margin constraint for each positive monotonic influence $X_j \overset{M+}{\prec} X_i \in C$ (see Eq. (2))

$$\begin{aligned} \delta^{a,b,c}_{i,j,+} = &P(X_i \leq c \mid X_j = a) \\ &- P(X_i \leq c \mid X_j = b) + \epsilon \leq 0 \end{aligned} \tag{8}$$

where $\epsilon \geq 0$ is a user-defined margin parameter. Similarly, each negative monotonic influence $X_j \overset{M-}{\prec} X_i \in C$ can be encoded as margin constraint $\delta^{a,b,c}_{i,j,-} \leq 0$.[1] The key difference from prior work (see [Altendorf et al., 2005]) is that we interpret the monotonic influence as constraints on conditional distributions obtained by marginalizing other variables instead of fixing their values.

Using the notation given in Eq. (8), we can express the optimization problem given in Eq. (6) as

$$\begin{aligned} \arg\max_{\theta^z} \mathcal{L}(T_z, \mathcal{D}_z) \qquad & s.t. \\ \delta^{a,b,c}_{i,j,+} \leq 0 \quad & \forall X_j \overset{M+}{\prec} X_i \in C \\ \delta^{a,b,c}_{i,j,-} \leq 0 \quad & \forall X_j \overset{M-}{\prec} X_i \in C \end{aligned} \tag{9}$$

A standard approach for solving the above optimization task is to use Lagrangian relaxation (see, for example, [Bertsekas, 1996]). A better alternative is the *penalty method*, which we will use in this paper. This method relaxes the constrained optimization problem into an unconstrained one by adding a penalty term to the objective. The latter equals the product of a penalty parameter λ and a function that is zero when the constraints are satisfied and non-zero (i.e., it penalizes the objective) when they are violated. It then optimizes the value of the penalty parameter λ by progressively increasing it (e.g., by multiplying it by 10) until convergence. Several penalty functions have been proposed in the literature. In our work, we use the quadratic penalty.

[1] Essentially this is similar to the positive monotonic constraint with a and b reversed.

To simplify our notation, we define the penalty function for each pair X_i, X_j as $\zeta_{i,j} = \sum_c \sum_{a>b} \zeta_{i,j}^{a,b,c}$ where,

$$\zeta_{i,j}^{a,b,c} = \begin{cases} \mathbb{1}_{\delta_{i,j,+}^{a,b,c} \geq 0} (\delta_{i,j,+}^{a,b,c})^2 & \text{If } X_j \overset{M+}{\prec} X_i \in C \\ \mathbb{1}_{\delta_{i,j,-}^{a,b,c} \geq 0} (\delta_{i,j,-}^{a,b,c})^2 & \text{If } X_j \overset{M-}{\prec} X_i \in C \\ 0 & \text{Otherwise} \end{cases} \quad (10)$$

Using the penalty function given in Eq. (10), we can solve the optimization problem given in Eq. (6) using the following series (indexed by t) of penalized problems:

$$\arg\max_{\theta^z} \mathcal{L}_{\text{pl}}(T^z, \mathcal{D}_z, t) \quad (11)$$

where $\mathcal{L}_{\text{pl}}(T^z, \mathcal{D}_z, t)$ is the penalized log-likelihood and is

$$\mathcal{L}_{\text{pl}}(T^z, \mathcal{D}_z, t) = \mathcal{L}_z(T^z, \mathcal{D}_z, t) - \lambda_t \sum_{i,j} \zeta_{i,j}(T^z, C_z)$$

Here, t denotes the iteration number. At each iteration, we increase λ_t (e.g. by a factor of 10), solve the unconstrained problem given in Eq. (11), and use the values of θ^z as the initial guess for the next iteration. As we increase λ_t, the solution will eventually converge to the solution of the constrained optimization problem given in Eq. (6) [Luenberger and Ye, 2016].

3.1.1 Gradients

At each iteration t, the unconstrained optimization problem given in Eq. (11) can be solved in practice using a standard gradient ascent procedure. Since the objective is smooth, the gradient ascent will always converge to a local optimum. To complete the description of this gradient ascent procedure, we provide the expressions for gradients in this section.

To encode the constraints $0 \leq \theta_{ijk}^z \leq 1$ and $\sum_{k'} \theta_{ijk'}^z = 1$, we parameterize θ^z using the softmax function, S as $\theta_{ijk}^z = S(\mu_{ij}^z)_k$. The gradient of the tree BN distribution with respect to the parameter μ_{ijk}^z is

$$\frac{\partial P^z(x)}{\partial \mu_{ijk}^z} = \mathbb{1}_{\text{Pa}_{x_i}^z = j} \frac{P^z(x)}{\theta_{ijk'}^z} S'(\mu_{ij}^z)_{k'}, \quad (12)$$

where S' is the gradient of the softmax function.

Now, without loss of generality, the gradient of penalty term due to the positive monotonic influence $X_j \overset{M+}{\prec} X_i \in C$ is

$$\frac{\partial \zeta_{i,j}^{a,b,c}}{\partial \mu_{ijk}^z} = 2\mathbb{1}_{\delta_{i,j,+}^{a,b,c} \geq 0} \delta_{i,j,+}^{a,b,c} \frac{\partial \delta_{i,j,+}^{a,b,c}}{\partial \mu_{ijk}^z}, \quad (13)$$

where the gradient of margin constraint is

$$\begin{aligned} \frac{\partial \delta_{i,j,+}^{a,b,c}}{\partial \mu_{ijk}^z} =& \frac{\partial P^z(X_i \leq c \mid X_j = a)}{\partial \mu_{ijk}^z} \\ &- \frac{\partial P^z(X_i \leq c \mid X_j = b)}{\partial \mu_{ijk}^z} \end{aligned} \quad (14)$$

The gradient of each conditional distribution is

$$\begin{aligned} \frac{\partial P^z(X_i = x_i \mid X_j = a)}{\partial \mu_{ijk}^z} =& \frac{\frac{\partial P^z(X_i = x_i, X_j = a)}{\partial \mu_{ijk}^z} P^z(X_j = a)}{P^z(X_j = a)^2} \\ &- \frac{P^z(X_i = x_i, X_j = a) \frac{\partial P^z(X_j = a)}{\partial \mu_{ijk}^z}}{P^z(X_j = a)^2}, \end{aligned} \quad (15)$$

where the gradient of each marginal distribution over a set of variables Q can be computed using equation 12 as

$$\frac{\partial P^z(X_Q = x_Q)}{\partial \mu_{ijk}^z} = \sum_{\substack{x' \in domain(X) \\ s.t. x'_Q = x_Q}} \frac{\partial P^z(X = x')}{\partial \mu_{ijk}^z} \quad (16)$$

3.1.2 Parameter Learning Algorithm

We use these gradients to optimize the penalized loglikelihood over the tree BN distribution (Equation (11)) using the L-BFGS-B algorithm. We describe the procedure to use the monotonic influences as constraints for the penalized loglikelihoods in Algorithm 1. Here, we iteratively increase the value of λ value until the penalty term is 0. We use a parameter t_{max} to limit the number of such iterations.

Algorithm 1 can be used to learn the parameters of the leaf distributions of Cutset Networks. Specifically, this can be done by setting $\mathcal{D}_z$ to the set of datapoints x such that $l(x) = z$ and C_z to the subset of C such that for each $X_j \overset{M+}{\prec} X_i \in C_z$ and each $X_j \overset{M-}{\prec} X_i \in C_z$, both i and j are in the scope V_z of the leaf. Algorithm 2 describes the procedure to learn leaf distributions of a cutset network using monotonic influences. It selects the data points $\mathcal{D}_z$ and constraints C_z that are applicable to each leaf z using the procedures *SelectDatapointsByPath* and *SelectInfluencesByScope* before performing the optimization over the parameters which are specific to that leaf.

3.2 VARIABLE SELECTION HEURISTIC

A limitation of the above parameter learning approach is that it can only use the monotonic influences over variables that are present in the scope of leaf nodes. As a result, knowledge about the variables in the internal nodes of the OR tree cannot be incorporated. To address this issue, we propose a variable selection heuristic to incorporate monotonic influences in the OR-tree structure. At internal node n, the heuristic score for variable $X_m \in V_n$ is given as

$$\frac{\mathcal{L}(\mathcal{M'}_{nm}, \mathcal{D'}_n)}{|\mathcal{D'}_n|} - \log(|\mathcal{D'}_n|) \sum_{X_i, X_j \in V_m^2} \zeta_{i,j}(\mathcal{M'}_{nm}, C_n)$$

where $\mathcal{D'}_n$ is the set of data points at node n, $\mathcal{M'}_{nm}$ is a cutset network of depth 1, rooted at X_m, and $\zeta_{i,j}(\mathcal{M'}_{nm})$

Algorithm 1: FitParameters

input : Parent map for a Tree BN Pa : $X \mapsto X$,
Scope of the Chow-Liu Tree V,
Data $\mathcal{D}$,
Set of Monotonic Influences C,
Maximum number of tries $t_{\max}$

output : Parameters for Tree BN θ

initialize: $\mu = \arg\max_{\mu} \mathcal{L}(\mu, \text{Pa}, \mathcal{D}), t = 1$
▷ start with maximum likelihood solution
$\lambda_1 = 1$
while $\sum_{X_i, X_j \in V^2} \zeta_{i,j}(\mu, C) \neq 0$ *and* $t \leq t_{max}$ **do**
  ▷ while constraints are not satisfied
  $\mu = \arg\max_{\mu} \left(\mathcal{L}(\mu, pa, \mathcal{D}) - \lambda_t \sum_{i,j} \zeta_{i,j}(\mu, pa, C)\right)$
  $\lambda_{t+1} = \lambda_t \times 10$ ▷ increase penalty weight
  $t = t + 1$
end
$\theta_{ijk} = S(\mu_{ij})_k, \ \forall i, j, k$ ▷ map into probability space
return θ'

Algorithm 2: FitLeaves

input : Cutset Network $\mathcal{M} = (O, T)$,
Data $\mathcal{D}$,
Set of Monotonic Influences C,
Maximum number of tries $t_{\max}$

output : Cutset Network with updated leaf parameters $\mathcal{M}'$

initialize: $\mathcal{M}' = \mathcal{M}$
for z *in* $1, \ldots, |T|$ **do**
  L_z = GetPathToLeaf(O, z)
  $\mathcal{D}_z$ = SelectDatapointsByPath($\mathcal{D}$, L_z)
  V_z = GetScope(O, z)
  C_z = SelectInfluencesByScope(C, V_z)
  Pa^z = GetParentMap(T_z)
  θ^z = FitParameters($\text{Pa}^z, V_z, \mathcal{D}_z, C_z, t_{max}$)
  replace $T'_z \in \mathcal{M}'$ with (Pa^z, θ^z)
end
return θ

Algorithm 3: ScoreWithKnowledge

input : Variable X_m,
Scope V,
Data $\mathcal{D}$,
Set of Monotonic Influences C,
Maximum number of tries $t_{\max}$

output : Heuristic score for variable X_m

O = OR-tree of depth 1 defined over V, rooted at X_m
T = Chow-Liu Trees at each leaf of O
$\mathcal{M}_{\text{init}} = (O, T)$
$\mathcal{M}$ = FitLeaves($\mathcal{M}_{\text{init}}, \mathcal{D}_z, C_z, t_{\max}$)
MeanLL = $\frac{1}{|\mathcal{D}|}\mathcal{L}(\mathcal{M}, \mathcal{D})$
PenaltyTerm = $\sum_{i,j \in V^2} \zeta_{i,j}(\mathcal{M}, C)$
Score = MeanLL $- \log(|\mathcal{D}|) \cdot$ PenaltyTerm
return Score

is the penalty function (Equation 10) defined over the cutset network distribution and the subset of qualitative influences C_n which are applicable to scope V_n. Note that this score is the same as the penalized loglikelihood objective function from Equation (11) applied to a cutset network of depth 1 and setting the penalty weight λ_t to $|\mathcal{D}'_n| \log(|\mathcal{D}'_n|)$. Algorithm 3 describes the procedure to compute this variable selection heuristic score for a variable X_m.

3.2.1 Structure Learning Algorithm

The knowledge-based parameter learning and variable selection heuristics described above can be integrated into a generalized framework for learning the structure and the parameters of a cutset network using qualitative influences. Algorithm 4 describes the KICN algorithm which learns a cutset network recursively like LearnCNet but uses Algorithm 1 to learn leaf parameters and uses Algorithm 3 for the variable selection heuristic.

4 EMPIRICAL EVALUATION

We aim to answer the following questions explicitly:

(Q1) Are monotonicities useful in learning cutset networks from noisy and sparse data?

(Q2) Does KICN improve the accuracy of learned models?

(Q3) Does KICN learn an *interpretable, explainable yet accurate* probabilistic model in high-stakes, clinical settings?

To answer these questions, we compared the networks learned using KICN with networks learned using LearnCNet.

We used two types of data sets for our experiments – 15 standard data sets to study the properties of KICN and 4 data sets from *high-stakes medical domains* to understand the interpretability and explainability of the models.

Benchmark data sets: We used two types of benchmark data sets – UCI repository [Dua and Graff, 2017] and classic Bayes net (BN) data sets. For UCI data sets, we considered the Computer Hardware (cpu), Breast Cancer (Ljubljana), Haberman's Survival (haberman), Auto MPG (auto), Car Evaluation (car), Yeast (yeast), Wine quality (redwine and whitewine), Abalone (abalone) Heart disease (cleveland) and Pima Indians Diabetes (diabetes) data sets. Wherever discretization ranges were not available, we categorized

Algorithm 4: KICN

input : Data $\mathcal{D}$,
Scope V,
Set of Monotonic Influences C,
Maximum number of tries $t_{\max}$

output : Cutset Network with updated leaf parameters $\mathcal{M}'$

if *Termination condition is satisfied* **then**
  Pa = Structure of Tree BN using Chow-Liu algorithm
  $\theta = \text{FitParameters}(\text{Pa}, V, \mathcal{D}, C, t_{\max})$
  $T = (\text{Pa}, \theta)$
  return T
end
Select a variable X_m as $\underset{X_m \in V}{\arg\max}\ \text{ScoreWithKnowledge}(X_m, V, \mathcal{D}, C, t_{\max})$
Child = List, W = List
for i *in* $|domain(X_m)|$ **do**
  $\mathcal{D}_z = \{x : x \in \mathcal{D}, x_m = i\}$
  $W_i = \frac{\mathcal{D}_z}{\mathcal{D}}$
  $V_z = V \setminus X_m$
  C_z = SelectInfluencesByScope(C, V_z)
  $\text{Child}_i = \text{KICN}(\mathcal{D}_i, V_i, C_i, t_{\max})$
end
O = (Child, W)
return O

Data set	LearnCNet	KICN (P)	KICN
cpu	-509.67	-490.98	**-468.76**
ljubljana	-1,059.62	-1,053.26	**-1,026.08**
cleveland	-1,498.21	-1,486.55	**-1,475.27**
haberman	-670.85	-668.82	**-643.76**
diabetes	-2,121.14	-2,084.99	**-2,078.09**
auto	-1,239.30	-1,233.32	**-1,230.08**
yeast	-6,040.67	-5,927.33	**-5,864.03**
car	-7,518.63	-7,501.16	**-7,485.55**
redwine	-5,769.67	-5,722.72	**-5,659.97**
whitewine	-15,054.33	-15,017.73	**-14,985.78**
abalone	-13,461.99	-13,340.42	**-12,992.09**
sachs	-1,025.38	**-1,015.24**	-1,015.92
asia	-411.62	-397.13	**-389.92**
earthquake	-124.65	-121.25	**-116.94**
survey	-451.02	-450.04	**-449.93**
ppd	-717.13	-711.90	**-710.02**
adni	-907.67	-901.85	**-867.09**
numom2b-a	-14,102.51	-14,102.81	**-14,068.81**
numom2b-b	-10,535.51	-10,515.25	**-10,448.37**

Table 1: Test loglikelihood scores for cutset networks fit on UCI data sets with 30% noise (rows 1–11), data sampled from Bayesian Networks (rows 12–15), and data from medical domains (rows 16–19) using LearnCNet, KICN with only parameter learning (KICN(P)) and KICN with both structure and parameter learning. The scores are averaged over 10 bootstrap samples.

each non-boolean variable into 3 categories and split each data set into a 50:50 train-test split. Of these, Haberman's Survival, Heart disease, and Pima Indians Diabetes data sets had monotonic influences available in the literature [Altendorf et al., 2005, Kokel et al., 2020]. For all the other data sets, we employed the use of the Qualitative Knowledge Extraction (QuaKE) algorithm [Karanam et al., 2021] to generate monotonic influences. Since the QuaKE algorithm can work with any probabilistic model, we used cutset networks to infer the monotonic constraints.

Our key hypothesis is that these domain constraints are more useful in data-scarce and noisy domains. While 50:50 train-test split takes care of sparsity, we induced noise in the training data by replacing 30% of the data points for each positive monotonic influence $X_j \overset{M+}{\prec} X_i$ with $X_i = R_i - \lfloor X_j \frac{R_i}{R_j} \rfloor$ where R_i and R_j are the max values of X_i and X_j. Similarly, for each negative monotonic influence, we use $X_j \overset{M-}{\prec} X_i$, $X_i = \lfloor X_j \frac{R_i}{R_j} \rfloor$. The noisy examples computed using the above formulas encode the reverse monotonic influences in C.

Our second type of benchmark data sets come from the BN community. We used Earthquake [Korb and Nicholson, 2010], Asia [Lauritzen, 1988], Survey [Scutari and Denis, 2014] and Sachs [Sachs et al., 2005] BNs. We sampled 100 examples (for sparsity) for generating both training and testing data from the BNs. We added 30% noise to the sampled training data using the same formula as above. We computed the monotonicities using the QuaKE algorithm.

High-stakes medical-domains: To understand the advantage of cutset networks over other deeper models in issues of interpretability, we used data from 3 studies, namely, Post-Partum Depression Survey (PPD Natarajan et al. [2017]), Alzheimer's Disease Neuroimaging Initiative (ADNI), and Nulliparous Pregnancy Outcomes Study: Monitoring Mothers-to-Be (nuMoM2b Haas et al. [2015]).

While we selected subsets of variables based on prior work on the PPD and ADNI domains, we considered two sub-cohorts of the nuMoM2b data, focusing on risk factors for Gestational Diabetes [Pagel et al., 2022]. The first sub-cohort (nuMoM2b-a) had 7 variables, namely, Body Mass Index (BMI), exercise in Metabolic Equivalent of Task units (METs), Age at first visit (Age), family history of diabetes (Hist), Polycystic Ovary Syndrome (PCOS), high Blood Pressure (HiBP), and Gestational Diabetes Mellitus (GDM). After excluding participants that had missing data for any of these variables, we had data from 6,164 in this sub-cohort. We obtained the following set of qualitative influences from

Data set	LearnCNet	KICN (P)	KICN
sachs	0.0708	0.0685	**0.0663**
asia	0.1923	0.1766	**0.1696**
earthquake	0.1391	0.1296	**0.1221**
survey	0.0181	0.0185	**0.0165**
cleveland	0.2746	0.2655	**0.2477**
haberman	0.2121	0.2065	**0.1953**
diabetes	0.2298	0.2164	**0.2114**
ppd	0.2043	0.1974	**0.1963**
adni	0.1825	0.1713	**0.1636**
numom2b-a	0.0397	0.0390	**0.0383**
numom2b-b	0.0515	0.0490	**0.0445**

Table 2: Mean squared error (MSE) for conditional probability queries for cutset networks fit on data sampled from BNs, on UCI data sets with prior knowledge and on data from clinical studies using LearnCNet, KICN with only parameter learning (KICN(P)) and KICN with both structure and parameter learning. The MSE values are averaged over 10 bootstrap samples.

an Obstetrics and Gynecology expert, Dr. David Haas:

$$\{\text{BMI}\overset{M+}{\prec}\text{GDM},\ \text{METs}\overset{M-}{\prec}\text{GDM},\ \text{Age}\overset{M+}{\prec}\text{GDM},\\ \text{Hist}\overset{M+}{\prec}\text{GDM},\ \text{PCOS}\overset{M+}{\prec}\text{GDM},\ \text{HiBP}\overset{M+}{\prec}\text{GDM}\}$$

The second sub-cohort (nuMoM2b-b) had the Polygenic Risk Score (PRS) as an additional variable. Further, since PRS is applicable only to non-Hispanic white participants with European ancestry, we excluded all the other participants. As a result, we had data from 3,657 participants in this sub-cohort. We categorized the non-boolean variables, namely, BMI, METs, Age, and PRS into 4 categories each. Apart from the influences listed above, we used the additional influence that PRS$\overset{M+}{\prec}$GDM.

Methods We compared the following versions of KICN[2] –

(1) Parameter learning using knowledge (KICN(P)).

(2) Parameter and structure learning using knowledge.

For version (1), the structure is pre-learned using LearnCNet and only the leaf parameters are updated. On the other hand version (2) involves learning the structure and parameters of the cutset network. For both modes, we set the number of tries t_{max} to 10 and we set the margin parameter ϵ to 0.001.

Metrics We used two metrics in our evaluation: the log-likelihood of the cutset network on the test data (test loglikelihood), and the Mean Squared Error (MSE) for conditional probability queries.

[2]Code and supplementary material available at `github.com/saurabhmathur96/KIL-CN`

	Edge count		Parameter count	
Data set	LearnCNet	KICN	LearnCNet	KICN
ppd	**113.8**	114.1	205.7	**198.8**
adni	121.9	**57.8**	343.3	**246.4**
numom2b-a	179.4	**108.6**	422.2	**366.3**
numom2b-b	416.5	**220.9**	1,069.9	**905.7**

Table 3: The number of edges and the number of free parameters for cutset networks fit using LearnCNet and KICN on medical data sets, averaged over 10 bootstrap samples.

Results

(**Q1**) Table 1 presents the log-likelihood on the test set for standard data sets (rows 1–15). The training data for each of these domains had 30% noise. Overall, using domain knowledge improves generative performance, and using knowledge for structure learning results in better performance than using knowledge only for parameter learning. This allows us to answer Q1 affirmatively.

(**Q2**) Table 2 presents the mean squared error for conditional probability queries for the BN data sets (rows 1–4) and the UCI data sets with prior knowledge (rows 5–7). For the BN datasets, we compared queries of the form $P(X_i = x_i \mid X_j = x_j)$ for each positive monotonic influence $X_j \overset{M+}{\prec} X_i$ or negative monotonic influence $X_j \overset{M-}{\prec} X_i$ to the ground truth probabilities from the BN. For the UCI data sets, we used the conditional probability of the target given all the risk factors ($P(X_{\text{target}} \mid X \setminus X_{\text{target}})$) and compared it to the values of the target in the test data set. The use of monotonic influence results in a lower mean squared error and hence more accurate answers to the queries. Further, as with the log-likehood scores, knowledge-based structure learning methods perform better than those using only parameter learning. Thus, we can answer Q2 affirmatively.

(**Q3**) The last 4 rows of table 1 show the test log-likelihood for the PPD, ADNI, and the two nuMoM2b data sets. For all the data sets, KICN improves the test log-likelihood. The last five rows of table 2 show the mean squared error for conditional probability queries. As with the UCI data sets with prior knowledge, we compared the probability $P(X_{\text{target}} \mid X \setminus X_{\text{target}})$ to the values of the target variable in the test set. The models learned using KICN give more accurate answers for the conditional probability query. Finally, table 3 compares the edge count and free parameter count for the structures learned using LearnCNet and KICN. The structures learned using KICN are more concise than the ones learned using LearnCNet. Thus, Q3 is answered affirmatively.

Finally, recall that as shown in Figure 1, the learned model is not only interpretable[3] but follows published research [Pagel et al., 2022]. It should be mentioned that while KICN uses the monotonic constraints on both PRS and BMI, it correctly infers that PRS is more important than BMI. Moreover, for the low values of PRS, BMI is chosen indicating that while the person might have a low propensity risk of gestational diabetes, BMI can have a significant impact. Similarly, for high-risk scores, the history of gestational diabetes becomes more important than BMI. These not only reflect and validate current medical knowledge but enhances it by identifying specific combinations that can allow for corresponding treatment plans.

Discussion: One of the limitations of KICN is that the knowledge in the form of monotonic influences must be valid regardless of the context. That is, if an influence $X_i \overset{M+}{\prec} X_j$ is given, we assume that there does not exist any context $\{X_Q = x_Q\}$ where $X_Q \subseteq (X \setminus \{X_i, X_j\})$ and $x_Q \in domain(X_Q)$ such that $X_i \overset{M+}{\prec} X_j \mid X_q = x_q$ is false. To account for this limitation, we used influences that were either independent of other variables or had a positive synergistic effect with them.

5 CONCLUSION

We considered the problem of incorporating rich domain knowledge in the form of qualitative constraints when learning an *interpretable, tractable* probabilistic model, namely, cutset networks. We developed KICN to leverage qualitative constraints to learn the structure and parameters of cutset networks. Our experiments on benchmark data sets and medical data sets demonstrated the efficacy of the proposed approach. Extending this work to deeper tractable models is an interesting future direction. Incorporating different types of domain knowledge including synergistic information, preferences over conditional distributions, privileged information, and imbalance tradeoffs is another direction. Finally, generating global explanations using the structure of these networks, and instance-level explanations constructed from the differences in the reasoning paths of the different examples can allow clinicians to develop treatment plans that mitigate adverse pregnancy outcomes.

Acknowledgements

SN and SM acknowledge the support by the NIH grant R01HD101246 and ARO award W911NF2010224. VG was supported in part by the DARPA Perceptually enabled Task Guidance (PTG) Program under contract number HR00112220005 and by the National Science Foundation CAREER award IIS-1652835.

[3]See supplementary material for additional results.

References

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.

Bozhena Bidyuk and Rina Dechter. On finding minimal w-cutset. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 43–50, 2004.

Mark Chavira and Adnan Darwiche. Compiling bayesian networks using variable elimination. In *IJCAI*, pages 2443–2449, 2007.

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

Chao-Kong Chow and Chao-Ning Liu. Approximating discrete probability distributions with dependence trees. *IEEE Trans. Inf. Theory*, 14(3):462–467, 1968.

Gregory F. Cooper. The computational complexity of probabilistic inference using bayesian belief networks. *Artif. Intell.*, 42(2-3):393–405, 1990.

Adnan Darwiche. A differential approach to inference in bayesian networks. *J. ACM*, 50(3):280–305, 2003.

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.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Glenn Fung, Olvi L. Mangasarian, and Jude W. Shavlik. Knowledge-based support vector machine classifiers. In *NIPS*, pages 521–528. MIT Press, 2002.

Vibhav Gogate and Pedro M. Domingos. Formula-based probabilistic inference. In *UAI*, pages 210–219. AUAI Press, 2010.

David M Haas, Corette B Parker, et al. A description of the methods of the nulliparous pregnancy outcomes study: monitoring mothers-to-be (numom2b). *American journal of obstetrics and gynecology*, 2015.

Athresh Karanam, Alexander L. Hayes, Harsha Kokel, David M. Haas, Predrag Radivojac, and Sriraam Natarajan. A probabilistic approach to extract qualitative knowledge for early prediction of gestational diabetes. In *AIME*, volume 12721 of *Lecture Notes in Computer Science*, pages 497–502. Springer, 2021.

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.

Kevin B Korb and Ann E Nicholson. *Bayesian artificial intelligence*. CRC press, 2010.

Gautam Kunapuli, Kristin P. Bennett, Richard Maclin, and Jude W. Shavlik. The Adviceptron: Giving Advice to the Perceptron. In *Intelligent Engineering Systems through Artificial Neural Networks, Volume 20*. ASME Press, 01 2010. ISBN 9780791859599.

Gautam Kunapuli, Phillip Odom, Jude W Shavlik, and Sriraam Natarajan. Guiding autonomous agents to better behaviors through human advice. In *ICDM*, 2013.

Steffen L Lauritzen. Local computations with probabilities on graphical structures and their application to expert systems (with discussion). *J. Roy. Statis., Soc., B*, 50(2): 251–263, 1988.

David G. Luenberger and Yinyu Ye. Springer International Publishing, 2016. doi: 10.1007/978-3-319-18842-3_1.

Nicola Di Mauro, Antonio Vergari, and Floriana Esposito. Learning accurate cutset networks by exploiting decomposability. In *AI*IA*, volume 9336 of *Lecture Notes in Computer Science*, pages 221–232. Springer, 2015.

Sriraam Natarajan, Annu Prabhakar, Nandini Ramanan, Anna Bagilone, Katie Siek, and Kay Connelly. Boosting for postpartum depression prediction. In *2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, pages 232–240. IEEE, 2017.

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

Kymberleigh A Pagel, Hoyin Chu, Rashika Ramola, Rafael F Guerrero, Judith H Chung, Samuel Parry, Uma M Reddy, Robert M Silver, Jonathan G Steller, Lynn M Yee, et al. Association of genetic predisposition and physical activity with risk of gestational diabetes in nulliparous women. *JAMA network open*, 5(8):e2229158–e2229158, 2022.

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

Judea Pearl. *Probabilistic reasoning in intelligent systems - networks of plausible inference*. Morgan Kaufmann series in representation and reasoning. Morgan Kaufmann, 1988.

Hoifung Poon and Pedro Domingos. Sum-product networks: A new deep architecture. In *UAI*, 2011.

Tahrima Rahman, Prasanna V. Kothalkar, and Vibhav Gogate. Cutset networks: A simple, tractable, and scalable approach for improving the accuracy of chow-liu trees. In *ECML/PKDD (2)*, volume 8725 of *Lecture Notes in Computer Science*, pages 630–645. Springer, 2014.

Tahrima Rahman, Shasha Jin, and Vibhav Gogate. Look ma, no latent variables: Accurate cutset networks via compilation. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 5311–5320. PMLR, 2019.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.*, 1(5):206–215, 2019.

Karen Sachs, Omar Perez, Dana Pe'er, Douglas A Lauffenburger, and Garry P Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

Marco Scutari and Jean-Baptiste Denis. *Bayesian Networks: With Examples in R*. Chapman and Hall/CRC, 2014.

Viktoriia Sharmanska, Novi Quadrianto, and Christoph H. Lampert. Learning to rank using privileged information. In *ICCV*, pages 825–832. IEEE Computer Society, 2013.

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

Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural networks*, 2009.

Michael P. Wellman. Fundamental concepts of qualitative probabilistic networks. *Artif. Intell.*, 44(3):257–303, 1990.

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.