Learning Credal Conditional Probability Tables with Qualitative Knowledge

Saurabh Mathur[1], **Alessandro Antonucci**[2], **Sriraam Natarajan**[1]

[1]The University of Texas at Dallas
[2]Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA) - Lugano, Switzerland
SaurabhSanjay.Mathur@utdallas.edu, alessandro@idsia.ch, Sriraam.Natarajan@utdallas.edu

Abstract

Bayesian networks are a popular class of directed probabilistic graphical models that allow for closed-form learning of the local parameters if complete data are available. However, learning the parameters is challenging when the data are sparse, incomplete, and uncertain. In this work, we present an approach to this problem based on *credal networks*, a generalization of Bayesian networks based on set-valued local parameters. We derive an algorithm to learn such set-valued parameters from data using qualitative knowledge in the form of monotonic influence statements. Our preliminary empirical evaluation shows that using qualitative knowledge reduces uncertainty about the parameters without significant loss in accuracy.

Introduction

Bayesian networks (BNs) are a powerful tool for representing and reasoning under uncertainty. They have been successfully applied in a wide variety of domains (Daly, Shen, and Aitken 2011) including healthcare (Lucas, Van der Gaag, and Abu-Hanna 2004), weather forecasting (Abramson et al. 1996), software engineering (Pendharkar, Subramanian, and Rodger 2005) and risk management (Fan and Yu 2004). However, BNs require complete and accurate data to learn the network parameters from data. In many real-world applications, such data may not be available.

To overcome the limitations of noisy and sparse data, domain knowledge might be used to learn BNs. Domain knowledge can concisely determine the direction and strength of relationships between variables (Niculescu et al. 2006) and trends in these relationships (Wellman 1990). Incorporating domain knowledge has been studied more broadly in machine learning. Knowledge in the form of precision-recall trade-off (Yang et al. 2014), label preferences (Odom et al. 2015), privileged information and qualitative influence statements (Altendorf, Restificar, and Dietterich 2005; Yang and Natarajan 2013; Mathur et al. 2023; Mathur, Gogate, and Natarajan 2023) have been successfully used to learn more accurate and robust models. While these methods overcome the limitations of noisy and sparse data, they are still unable to deal with incomplete and uncertain data.

Credal networks (CNs) address this limitation by extending BNs to explicitly represent incompleteness and uncertainty about probability distributions (Mauá and Cozman 2020). They provide a more cautious approach to the specification of probabilistic models. This makes CNs especially useful for noisy, sparse, and possibly incomplete data domains. However, inducing them purely from data can make the model too "imprecise" and result in vacuous inferences.

In this work, we present a solution to the problem of learning accurate yet robust models in the presence of noisy, sparse, and possibly incomplete data by embedding domain knowledge in CNs. Specifically, we consider a subclass of qualitative influence statements called *monotonic influence statements* to make CNs more precise. The main contributions of this paper are the development of a learning method for CNs that effectively exploits monotonic influence relationships in the domain as knowledge and the preliminary empirical evaluation of the learning algorithm.

The rest of this paper is organized as follows. First, we provide some necessary background about CNs and qualitative influence statements. Then, we detail our method for learning CNs from data using domain knowledge and report our empirical evaluation before concluding the paper with a summary and a discussion on the central outlooks.

Background Concepts

Bayesian and Credal Networks. *Bayesian networks* (BNs, Koller and Friedman 2009) are probabilistic graphical models that compactly represent joint *probability mass functions* (PMFs). Formally, a BN over a set of variables $\boldsymbol{X} = \{X_1, \ldots, X_n\}$ is a pair $\langle \mathcal{G}, \theta \rangle$. Here, $\mathcal{G}$ is a directed acyclic graph such that each node corresponds to a random variable in $\boldsymbol{X}$ and θ is a set of conditional PMFs specified for each variable, given all the possible values of its parents $\mathrm{Pa}_X \subset \boldsymbol{X}$ according to $\mathcal{G}$. Graph $\mathcal{G}$ represents conditional independence relations according to the Markov condition. As a result, the joint PMF induced by the BN can be expressed as the following factorization:

$$P(\boldsymbol{x}) = \prod_{X \in \boldsymbol{X}} P(x|\mathrm{pa}_X)\,, \tag{1}$$

for each $\boldsymbol{x} \in \mathrm{Dom}(\boldsymbol{X})$, where the values pa_X and x are those consistent with $\boldsymbol{x}$.

Credal networks (CNs, Mauá and Cozman 2020) are a generalization of BNs that allows us to define sets of joint PMFs. A set of PMFs over X is called *credal set* (CS) and denoted as $K(X)$. CSs (Levi 1980; Augustin et al. 2014) allow us to explicitly represent incompleteness in uncertain specifications (e.g., a *vacuous* CS including all the possible PMFs over X, thus expressing a condition of complete ignorance about X).

In practice, the specification of a CN is the same as that of a BN except that each (conditional) PMF is replaced by a CS. The Markov condition can also be applied to CNs, provided that a suitable notion of independence is considered. Here we focus on the notion of *strong* independence, i.e., X and X' are independent according to CS $K(X, X')$ if they are independent in the stochastic sense for each PMF in the extreme points CS. This allows us to define a joint CS $K(\boldsymbol{X})$ as the convex closure of the set of all joint PMFs as in Eq. (1) such that the conditional PMFs are taken from the conditional CSs in the CN specification. Inferences in CSs are consequently intended w.r.t. such a joint CS.

In this work, we consider CNs defined using closed and convex CSs, that are finitely-generated, i.e., induced by a finite number of linear constraints on the PMFs $P(X)$ belonging to $K(X)$. This allows us to equivalently describe each conditional CS by listing its extreme points, whose number should be also finite.

Decision-making in CSs. Recall that decision-making in PMFs involves finding the state (decision) that minimizes a given loss function. For the 0-1 loss function, this corresponds to taking the state $x^* := \arg\max_{x \in \boldsymbol{X}} P(x)$. Decision-making in CSs can be done using *interval dominance* (Zaffalon 2002; Troffaes 2007). State $x \in \mathrm{Dom}(X)$ is said to interval-dominate another state $x' \in \mathrm{Dom}(X)$ according to the CS $K(X)$ if and only if:

$$\min_{P \in K} P(x) > \max_{P \in K} P(x') \,, \tag{2}$$

where the two optimizations can be computed w.r.t. the linear constraints in the CS specification. For closed convex CSs, this can be done by enumerating the extreme points. If a single state interval-dominates all other states, then that state can be selected as optimal for the decision. However, we might have more than one undominated state. In such cases, we can abstain from making a further decision and regard all the undominated states as optimal.

Learning closed convex CSs. The *imprecise Dirichlet model* (IDM, Walley 1996) is the most popular approach for learning closed convex CSs from categorical data. This is a generalization of the Bayesian approach of combining a multinomial likelihood with a Dirichlet prior distribution. Instead of a single Dirichlet prior, the IDM posits a set of priors, called the imprecise Dirichlet prior. Specifically, when learning from a data set $\mathcal{D}$ of observations of the random variable X, the set of Dirichlet priors is parameterized as $\mathrm{Dir}(st_X)$. Here, $s \in \mathbb{R}^+$ is a hyper parameter and $t_X := \{t_x\}_{x \in \mathrm{Dom}(X)}$ with $t_x \in [0, 1]$ and $\sum_x t_x = 1$. The probability induced by the IDM is therefore:

$$P(x) = \frac{N_x + st_x}{N + s} \,, \tag{3}$$

where N_x is the number of times $X = x$ occurs in data and N is the total number of observations in the $\mathcal{D}$. The bounds of the above expression are:

$$\underline{P}(x) := \min_{P \in K} P(x) = \min_{t_x \in [0,1]} \frac{N_x + st_x}{N + s} = \frac{N_x}{N + s} \,, \tag{4}$$

$$\overline{P}(x) := \max_{P \in K} P(x) = \max_{t_x \in [0,1]} \frac{N_x + st_x}{N + s} = \frac{N_x + s}{N + s} \,. \tag{5}$$

Those bounds induce linear constraints defining a CS $K(X)$. For data sets that are small w.r.t. the parameter s, these bounds can be quite broad. In the rest of the paper we discuss a procedure to shrink these bounds by using domain knowledge.

Domain knowledge as qualitative influence statements. *Qualitative influence statements* (QISs, Wellman 1990) describe the influence of one or more variables over another variable. These statements allow domain experts to concisely express a trend in the distribution without needing to specify precise values. Here we focus on learning CNs using a class of qualitative influence statements called *monotonic influence statements* (MISs, Altendorf, Restificar, and Dietterich 2005). MISs refer to ordinal, and hence also Boolean as a special case, variables. Given a variable Y and a joint variable $\boldsymbol{X}$ in a probabilistic model, we say that Y is *positively monotonically influenced* by parent $X \in \boldsymbol{X}$ if higher values of X stochastically result in higher values of Y, *ceteris paribus*. Such an influence is denoted as $X \overset{M+}{\prec} Y$ and corresponds to domain knowledge of the form "As X increases, Y also increases". We express this MIS as the inequality:

$$P(Y \leq y | x, \tilde{\boldsymbol{x}}) \geq P(Y \leq y | x', \tilde{\boldsymbol{x}}) \tag{6}$$

for each $x, x' \in \mathrm{Dom}(X)$ such that $x \leq x'$, $y \in \mathrm{Dom}(Y)$, and $\tilde{\boldsymbol{x}} \in \mathrm{Dom}(\tilde{\boldsymbol{X}})$, where $\tilde{\boldsymbol{X}} := \boldsymbol{X} \setminus \{X\}$. Negative influence can be defined analogously and denoted as $X \overset{M-}{\prec} Y$.

Related Work. QISs have been used to induce more accurate precise probabilistic models from noisy and sparse data for both discriminative (Kokel et al. 2020; Odom et al. 2015) and generative learning settings (van der Gaag, Bodlaender, and Feelders 2004; Altendorf, Restificar, and Dietterich 2005; de Campos, Tong, and Ji 2008; Yang and Natarajan 2013; Plajner and Vomlel 2020; Mathur et al. 2023; Mathur, Gogate, and Natarajan 2023). In this work, we deal with learning imprecise generative models from sparse, incomplete, and uncertain data. QISs have been previously used to make generative models more precise. Renooij and van der Gaag (2002) introduce influence-intervals and perform interval-propagation on Qualitative Probabilistic Networks to shrink the intervals. In contrast, our method maintains probabilistic semantics by dealing with closed, convex credal sets. QISs have also been used to learn conditional credal sets. de Campos and Cozman (2005) use qualitative influences as constraints on the imprecise Dirichlet prior. However, in the presence of prior-data conflicts (Evans and Moshonov 2006), this approach does not guarantee consistency with the qualitative knowledge. Our approach of directly constraining a credal set provides a more flexible solution to this problem. This also makes it independent of the way that the CS is initially computed.

Methodology

We consider the problem of integrating qualitative knowledge (and in particular MISs) in the statistical learning of an imprecise probabilistic model. We consequently focus on the following learning task.

> **Given:** Data set $\mathcal{D} := \{y^{(i)}, \boldsymbol{x}^{(i)}\}_{i=1}^{N}$ over variables $(Y, \boldsymbol{X})$ and a collection C of MISs as in Eq. (6).
> **To Do:** Learn a collection of conditional CSs over Y, say $\{K(Y|\boldsymbol{x})\}_{\boldsymbol{x}\in\mathrm{Dom}(\boldsymbol{X})}$, that are compatible with C.

Embedding a MIS in the prior specification can be more complicated as the relation in Eq. (6) induces a constraint between different conditional distributions. Thus, to solve the above problem, we obtain an initial set of CSs from $\mathcal{D}$ through the standard IDM learning and then derive a procedure to shrink the IDM bounds by eliminating PMFs that violate the MISs in C. We achieve that by jointly solving for all the maximum values that satisfy the monotonicity constraints. This corresponds to the following optimization:

$$\underset{\substack{\underline{P}_0(y|\boldsymbol{x})\leq q_{y|\boldsymbol{x}}\leq \overline{P}_0(y|\boldsymbol{x})\\ q_{y|\boldsymbol{x}}\models C\\ \boldsymbol{x}\in\mathrm{Dom}(\boldsymbol{X})\\ y\in\mathrm{Dom}(Y)}}{\arg\max} \sum_{\substack{\boldsymbol{x}\in\mathrm{Dom}(\boldsymbol{X})\\ y\in\mathrm{Dom}(Y)}} q_{y|\boldsymbol{x}}\,, \tag{7}$$

where, for each $y \in \mathrm{Dom}(Y)$ and $\boldsymbol{x} \in \mathrm{Dom}(\boldsymbol{X})$, $q_{y|\boldsymbol{x}}$ is an optimisation variable, $\underline{P}_0(y|\boldsymbol{x})$ and $\overline{P}_0(y|\boldsymbol{x})$ are the IDM constraints as in Eqs. (4) and (5), while $\models C$ denote compatibility between the optimization variables and the MIS constraints in C as stated by Eq. (6). Let us denote the set of all the optimization variables as $\boldsymbol{q}$. An analogous optimization can be considered for the lower bounds. Such linear programs are not guaranteed to have feasible solutions, because some constraints might be unsatisfiable under the IDM constraints. If this is the case we address the optimization using the *barrier penalty* method (Luenberger and Ye 2016). Specifically, we encode each MIS constraints $c \in C$ of the form $X_{\prec}^{M+}Y$ as $\delta_c(\boldsymbol{q}, \epsilon) \leq 0$ where:

$$\delta_c(\boldsymbol{q}, \epsilon) = \sum_{y'\leq y} q_{y'|x',\tilde{\boldsymbol{x}}} - \sum_{y''\leq y} q_{y''|x,\tilde{\boldsymbol{x}}} + \epsilon\,, \tag{8}$$

and we introduce a penalty $\max\{0, \delta_c(\boldsymbol{q}, \epsilon)\}^2$. Then, instead of Eq. (7), we solve a sequence of optimization problems of the form:

$$\underset{\substack{\underline{P}_0(y|\boldsymbol{x})\leq q_{y|\boldsymbol{x}}\leq \overline{P}_0(y|\boldsymbol{x})\\ \boldsymbol{x}\in\mathrm{Dom}(\boldsymbol{X})\\ y\in\mathrm{Dom}(Y)}}{\arg\max} \left[\mathcal{L}(\boldsymbol{q}) - \lambda \underbrace{\sum_{c\in C} \max\{0, \delta_c(\boldsymbol{q}, \epsilon)\}^2}_{\text{Penalty}}\right], \tag{9}$$

for $\lambda = 10^0, 10^1, 10^2, \ldots, 10^L$ until the penalty term vanishes, where $\mathcal{L}(\boldsymbol{q})$ is the objective function in Eq. (7). If a feasible solution exists, then this method is guaranteed to converge to a solution in the limit (Luenberger and Ye 2016). We analogously proceed for the minimization task.

Algorithm 1 details our procedure to obtain the consistent conditional CSs from the data set $\mathcal{D}$ and the MISs C. The algorithm begins by computing the IDM CSs from $\mathcal{D}$. It then uses the MISs C to shrink the CS bounds. It does so by finding the highest and lowest values in the initial CS that satisfy all the constraints in C. These values are obtained by constrained optimization based on the barrier penalty method. This is performed by sub-procedure detailed by Algorithm 2.

Algorithm 1: LearnConditionalCredalSetWithKnowledge

Input:
- $\mathcal{D}$ (data set over $\boldsymbol{X}$ and Y)
- C (MISs)
- $t_{\max}$ (maximum number of iterations)

Output:
- CS bounds

```
Initialize P̲(y|x) = P̲_0(y|x), P̄(y|x) = P̄_0(y|x) for each y ∈ Dom(Y) and x ∈ Dom(X)
{P̄(y|x)}_{y,x} = ConstrOpt(+1, P̲_0, P̄_0, C, t_max)
{P̲(y|x)}_{y,x} = ConstrOpt(−1, P̲_0, P̄_0, C, t_max)
return {[P̲(y|x), P̄(y|x)]}_{y∈Dom(Y), x∈Dom(X)}
```

Algorithm 2: ConstrOpt

Input:
- σ (+1 if maximize and -1 if minimize)
- $\{\underline{P}(y|\boldsymbol{x}), \overline{P}(y|\boldsymbol{x})\}_{y\in\mathrm{Dom}(Y),\boldsymbol{x}\in\mathrm{Dom}((\boldsymbol{X})}$ (CS bounds)
- C (MISs)
- $t_{\max}$ (maximum number of iterations)

Output:
- upper/lower CS bounds satisfying C

```
Initialize q = argmax_{P̲(y|x) ≤ q_{y|x} ≤ P̄(y|x), x ∈ Dom(X), y ∈ Dom(Y)} σL(q)
λ = 1
while Σ_{c∈C} max{0, δ_c(q, ε)}^2 > 0 and t ≤ t_max do
    q = argmax [σL(q) − λ Σ_{c∈C} max{0, δ_c(q, ε)}^2]
    λ = λ × 10
    t = t + 1
end while
return q
```

Experimental Evaluation

We aim to answer the following research question:

(Q) Does using monotonicities with the IDM improve its coverage without losing performance?

Data sets. For a preliminary validation, we consider five data sets from the UCI Machine Learning Repository. We use the same pre-processing and domain knowledge as in

Data set	$\|\mathcal{D}\|$	Y	$\boldsymbol{X}$
haberman	306	survive	nodes$^-$, year$^+$, age$^-$
diabetes	392	diabetes	Age$^+$, Pregnancies$^+$, BMI$^+$, PedigreeFunction$^+$
breast-cancer	277	recurrence	age$^+$, menopause$^+$, deg_malig$^+$, tumor_size$^+$, irradiat$^-$
thyroid	185	Hyperthyroid	TSH$^+$, TSH_diff$^+$, T3_resin$^+$, T3$^+$, T4$^+$
heart-disease	297	heart_disease	sex_male$^+$, age$^+$, trestbps$^+$, chol$^+$, diabetes$^+$

Table 1: Data sets used for empirical evaluation, the number of examples ($|\mathcal{D}|$), the target (Y) and feature variables ($\boldsymbol{X}$). A feature with the superscript + denotes a positive monotonic influence, and a feature with the superscript - denotes a negative monotonic influence.

Data set	**BN**	**CN-IDM**		**CN-IDM-MIS**	
	accuracy	**accuracy**	**uncertainty**	**accuracy**	**uncertainty**
haberman	0.73 ± 0.01	0.77 ± 0.01	0.09 ± 0.03	0.76 ± 0.02	0.05 ± 0.01
diabetes	0.68 ± 0.05	0.76 ± 0.06	0.35 ± 0.05	0.72 ± 0.03	0.04 ± 0.03
breast-cancer	0.69 ± 0.02	0.80 ± 0.05	0.31 ± 0.11	0.76 ± 0.04	0.08 ± 0.08
thyroid	0.94 ± 0.02	0.95 ± 0.02	0.01 ± 0.01	0.95 ± 0.02	0.01 ± 0.01
heart-disease	0.57 ± 0.03	0.67 ± 0.10	0.39 ± 0.14	0.63 ± 0.06	0.08 ± 0.05

Table 2: The accuracy and fraction of uncertain examples (*uncertainty*) in the test set for each method. The mean and standard deviation for the scores are computed by stratified five-fold cross-validation. Note that the BN method has by construction zero uncertainty.

prior work (Yang and Natarajan 2013). Table 1 details the size of the datasets, the Boolean target variables considered in our experiments, and the parent variables of the target together with the kind of monotonic influence they have on the target. The target variables (Y) in all the data sets are Boolean and the parents ($\boldsymbol{X}$) are ordinal variables.

Methods. We compare our algorithm (discussed in the previous section and denoted here as IDM+QI) against two baselines – (1) a precise BN estimator with a Dirichlet prior (denoted as BN), and (2) a CN estimator based on the pure IDM (denoted as CN-IDM-MIS). We set the prior parameter $s = 1$ for all data sets and models. Additionally, we set $\epsilon = 0.001$ for all the constraints. The Python code used for the experiments is freely available in a public repository.[1]

We perform inference in the CN models by interval-dominance. If neither value of the Boolean target interval-dominates the other, we mark that data point as uncertain and do not make an inference for it. For the BN model, we perform inference by thresholding the positive probability at ≥ 0.5.

Metrics. We evaluate the methods using two metrics – fraction of uncertain data points (*uncertainty*) and *accuracy* over non-uncertain data points. We compute these metrics by five-fold cross-validation.

Results.

(Q) Table 2 presents the accuracy and the number of uncertain examples for the three methods under consideration. The IDM method achieves the highest accuracies, but the price is being uncertain about a large fraction of the test examples (over 30% for diabetes, breast-cancer, and heart-disease). The proposed method (CN-IDM-MIS) reduces the number of uncertain examples relative to CN-IDM by 64.4% on average with an average relative decrease in accuracy of 3.4%. Hence the research question is answered affirmatively.

[1]https://github.com/saurabhmathur96/credal-cpd

Conclusions and Outlooks

We presented an IDM-based procedure to learn credal networks from data in a way that is also consistent with qualitative knowledge expressed by monotonic influence statements. This is achieved by an iterative procedure shrinking the IDM bounds. Our preliminary tests demonstrate that the proposed algorithm yields conditional credal sets that have higher coverage without losing much accuracy. For a deeper validation, more extensive experiments involving a sensitivity analysis with respect to the algorithm parameters should be considered. Additional future work includes extending the proposed method to support other qualitative influence statements like synergies. Moreover, we also intend to consider a more general setup where the qualitative influence statements are not restricted to parent-child relations.

Acknowledgements

The authors acknowledge the support by AFOSR award FA9550-23-1-0239 and ARO award W911NF2010224.

References

Abramson, B.; Brown, J.; Edwards, W.; Murphy, A.; and Winkler, R. L. 1996. Hailfinder: A Bayesian system for forecasting severe weather. *International Journal of Forecasting*, 12(1): 57–71.

Altendorf, E.; Restificar, A. C.; and Dietterich, T. G. 2005. Learning from Sparse Data by Exploiting Monotonicity Constraints. In *UAI*, 18–26. AUAI Press.

Augustin, T.; Coolen, F. P.; De Cooman, G.; and Troffaes, M. C. 2014. *Introduction to imprecise probabilities*. John Wiley & Sons.

Daly, R.; Shen, Q.; and Aitken, S. 2011. Learning Bayesian networks: approaches and issues. *The knowledge engineering review*, 26(2): 99–157.

de Campos, C. P.; and Cozman, F. G. 2005. Belief Updating and Learning in Semi-Qualitative Probabilistic Networks. In *UAI*, 153–160. AUAI Press.

de Campos, C. P.; Tong, Y.; and Ji, Q. 2008. Constrained Maximum Likelihood Learning of Bayesian Networks for Facial Action Recognition. In *ECCV (3)*, volume 5304 of *Lecture Notes in Computer Science*, 168–181. Springer.

Evans, M.; and Moshonov, H. 2006. Checking for prior-data conflict. *Bayesian Analysis*, 1(4): 893 – 914.

Fan, C.-F.; and Yu, Y.-C. 2004. BBN-based software project risk management. *Journal of Systems and Software*, 73(2): 193–203.

Kokel, H.; Odom, P.; Yang, S.; and Natarajan, S. 2020. A Unified Framework for Knowledge Intensive Gradient Boosting: Leveraging Human Experts for Noisy Sparse Domains. In *AAAI*, 4460–4468. AAAI Press.

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

Levi, I. 1980. *The enterprise of knowledge: An essay on knowledge, credal probability, and chance*. MIT press.

Lucas, P. J.; Van der Gaag, L. C.; and Abu-Hanna, A. 2004. Bayesian networks in biomedicine and health-care. *Artificial intelligence in medicine*, 30(3): 201–214.

Luenberger, D. G.; and Ye, Y. 2016. *Linear and Nonlinear Programming*, 1–8. Cham: Springer International Publishing.

Mathur, S.; Gogate, V.; and Natarajan, S. 2023. Knowledge Intensive Learning of Cutset Networks. In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, 1380–1389. PMLR.

Mathur, S.; Karanam, A.; Radivojac, P.; Haas, D. M.; Kersting, K.; and Natarajan, S. 2023. Exploiting Domain Knowledge as Causal Independencies in Modeling Gestational Diabetes. In *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, volume 28, 359–370.

Mauá, D. D.; and Cozman, F. G. 2020. Thirty years of credal networks: Specification, algorithms and complexity. *International Journal of Approximate Reasoning*, 126: 133–157.

Niculescu, R. S.; Mitchell, T. M.; Rao, R. B.; Bennett, K. P.; and Parrado-Hernández, E. 2006. Bayesian network learning with parameter constraints. *Journal of Machine Learning Research*, 7(7).

Odom, P.; Khot, T.; Porter, R.; and Natarajan, S. 2015. Knowledge-Based Probabilistic Logic Learning. In *AAAI*, 3564–3570. AAAI Press.

Pendharkar, P. C.; Subramanian, G. H.; and Rodger, J. A. 2005. A probabilistic model for predicting software development effort. *IEEE Transactions on software engineering*, 31(7): 615–624.

Plajner, M.; and Vomlel, J. 2020. Learning bipartite Bayesian networks under monotonicity restrictions. *Int. J. Gen. Syst.*, 49(1): 88–111.

Renooij, S.; and van der Gaag, L. C. 2002. From Qualitative to Quantitative Probabilistic Networks. In *UAI*, 422–429. Morgan Kaufmann.

Troffaes, M. C. 2007. Decision making under uncertainty using imprecise probabilities. *International journal of approximate reasoning*, 45(1): 17–29.

van der Gaag, L. C.; Bodlaender, H. L.; and Feelders, A. J. 2004. Monotonicity in Bayesian Networks. In *UAI*, 569–576. AUAI Press.

Walley, P. 1996. Inferences from multinomial data: learning about a bag of marbles. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1): 3–34.

Wellman, M. P. 1990. Fundamental Concepts of Qualitative Probabilistic Networks. *Artif. Intell.*, 44(3): 257–303.

Yang, S.; Khot, T.; Kersting, K.; Kunapuli, G.; Hauser, K.; and Natarajan, S. 2014. Learning from imbalanced data in relational domains: A soft margin approach. In *ICDM*.

Yang, S.; and Natarajan, S. 2013. 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*, 580–595. Springer.

Zaffalon, M. 2002. The naive credal classifier. *Journal of Statistical Planning and Inference*, 105(1): 5–21.