Hybrid Probabilistic Inference with Logical and Algebraic Constraints: a Survey

Paolo Morettin[1] , **Pedro Zuidberg Dos Martires**[1] , **Samuel Kolb**[1] , **Andrea Passerini**[2]
[1]KU Leuven,
[2]University of Trento
{paolo.morettin, pedro.zudo, samuel.kolb }@kuleuven.be, andrea.passerini@unitn.it

Abstract

Real world decision making problems often involve both discrete and continuous variables and require a combination of probabilistic and deterministic knowledge. Stimulated by recent advances in automated reasoning technology, hybrid (discrete+continuous) probabilistic reasoning with constraints has emerged as a lively and fast growing research field. In this paper we provide a survey of existing techniques for hybrid probabilistic inference with logic and algebraic constraints. We leverage weighted model integration as a unifying formalism and discuss the different paradigms that have been used as well as the expressivity-efficiency trade-offs that have been investigated. We conclude the survey with a comparative overview of existing implementations and a critical discussion of open challenges and promising research directions.

1 Introduction

Achieving general-purpose machine reasoning requires both the ability to jointly model discrete and continuous variables as well as the ability to integrate both probabilistic and deterministic knowledge. In real-world scenarios, hybrid domains are extremely common, from transport modelling and traffic forecasting [Hensher and Button, 2007] to probabilistic robotics [Thrun *et al.*, 2005] and cyber-physical systems [Lee, 2008]. Probabilistic graphical models (PGMs) [Koller and Friedman, 2009] offer a principled way for modeling uncertainty and have been extensively studied over the last three decades. Although initially restricted to purely discrete or purely continuous settings, they have been progressively extended to increasingly general hybrid settings, i.e., settings with discrete and continuous variables.

Early approaches to hybrid probabilistic modeling relied on *discretization* of the continuous variables [Friedman and Goldszmidt, 1996; Kozlov and Koller, 1997] or made strong *distributional assumptions*, such as conditional Gaussian (CG) models [Lauritzen, 1992], which do not allow to condition a discrete variable on continuous ones. These assumptions severely impact the expressiveness of the models. For instance, it is not possible to model whether a thermostat will trigger given the measured temperature. More recent approaches to hybrid probabilistic models include hybrid Markov random fields [Yang *et al.*, 2014] and hybrid sum-product networks [Molina *et al.*, 2018; Bueff *et al.*, 2018]. The former assumes that conditional distributions can be specified using heterogeneous univariate distributions from the exponential family, while the latter extends SPNs by allowing for piecewise polynomials over continuous variables.

A drawback of the approaches considered above is that they do not support deterministic relations, i.e., hard constraints, between variables. These constraints can either arise from the problem itself, such as laws of physics, or they could be instrumental in characterizing the desired behavior of the system, such as safety or fairness constraints. Consider, for example, planning a robust supply system for a chain of stores [Belle *et al.*, 2015b]. Probabilistic estimates of travel times between pairs of stores should be combined with hard constraints involving stores to be supplied, number of vehicles available and total travel time for a vehicle being the sum of the travel times for the different stores it supplies. Recent years have witnessed a growing research interest in modeling and solving problems with such deterministic relations, by developing inference algorithms for hybrid probabilistic models that support algebraic and logical constraints. This survey aims at characterizing the most prominent techniques and identifying open challenges in this field.

First, we introduce a canonical formulation for hybrid probabilistic inference tasks based on Weighted Model Integration (WMI) [Belle *et al.*, 2015b] (Section 2). Second, we characterize the different expressiveness-efficiency trade-offs that have been investigated to mitigate the computational cost associated with probabilistic inference (Section 3). Third, we present the different paradigms that have been used to address the two main components of hybrid probabilistic inference, combinatorial enumeration and the integration, and how they are used by solver implementations (Section 4). In Section 5 we discuss approaches to probabilistic inference with constrains in the probabilistic programming literature Finally, we discuss the most relevant open challenges as well as promising research directions that address them (Section 6).

2 A Canonical Formulation

For probabilistic graphical models, weighted model counting (WMC) has emerged as a canonical formulation of the probabilistic inference task [Chavira and Darwiche, 2008]. Instead of describing and implementing inference algorithms directly on the level of PGMs, WMC abstracts a query in a PGM to a computation on a propositional formula. Each literal in the propositional formula is given a weight and the answer to the query in the PGM is the weight of the propositional formula, also called the weighted model count.[1]

Definition 1 (Weighted Model Count)**.** *Given a set* $\mathbf{b}$ *of* M *Boolean variables, a weight function* $w : \mathbb{B}^M \to \mathbb{R}_{\geq 0}$*, and a propositional formula* ϕ *(called support) over* $\mathbf{b}$*, the weighted model count is*

$$\text{WMC}(\phi, w|\mathbf{b}) = \sum_{\mathbf{b}_\mathcal{I} \in \mathcal{I}_\mathbf{b}(\phi)} w(\mathbf{b}_\mathcal{I}) \qquad (1)$$

$\mathcal{I}_\mathbf{b}(\phi)$ *is the set of interpretations that satisfy* ϕ*.*

Traditionally, WMC is used when the weight function w factorizes as a product of weights of literals:

$$\text{WMC}(\phi, w|\mathbf{b}) = \sum_{\mathbf{b}_\mathcal{I} \in \mathcal{I}_\mathbf{b}(\phi)} \prod_{b_i \in \mathbf{b}_\mathcal{I}} w(b_i) \qquad (2)$$

This has the algorithmic advantage that dynamic programming techniques can be deployed [Bellman, 1957], which mitigates the computational hardness of computing the weighted model count (#P-hard), i.e., of performing probabilistic inference.

One obvious shortcoming of propositional logic formulas is their inability to express continuous random variables (uncountable sets). Consequently, WMC is unable to fully express inference over continuous variables, as well. In a seminal paper, Belle et al. [2015a] extended WMC to include continuous variables. They dubbed their canonical formulation *weighted model integration* (WMI). A key innovation was to replace propositional logical formulas with so-called (quantifier-free) satisfiability modulo theory (SMT) formulas [Barrett and Tinelli, 2018], a unifying language for expressing combinations of logical and algebraic constraints.

Definition 2 (SMT($\mathcal{LRA}$) formula)**.** *Let* $\mathbf{b}$ *be a set of* M *Boolean and* $\mathbf{x}$ *a set of* N *real variables. An* **atomic formula** *is an expression of the form* $\sum_i c_i \cdot x_i \bowtie c$*, where the* $x_i \in \mathbf{x}$*,* $c_i, c \in \mathbb{Q}$*, and* $\bowtie \in \{=, \neq, \geq, \leq, >, <\}$*. We then define SMT(*$\mathcal{LRA}$*) theories as Boolean combinations (by means of the standard Boolean operators* $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$*) of Boolean variables in* $\mathbf{b}$ *and of atomic formulas over* $\mathbf{x}$*.*

The acronym $\mathcal{LRA}$ in SMT($\mathcal{LRA}$) stands for *linear real arithmetics*, which already implies that the algebraic constraints on the real variables can only be of a linear form. While WMI has mostly been studied on SMT($\mathcal{LRA}$) formulas, variations have been proposed to handle non-linear constraints [Zuidberg Dos Martires *et al.*, 2019; Fuxjaeger and Belle, 2020; Afshar *et al.*, 2016] or integers instead of reals [Kolb *et al.*, 2018a].

[1]We refer the interested reader to [Chavira and Darwiche, 2008] on how to encode discrete PGMs as Boolean formulas.

In this survey we focus on SMT($\mathcal{LRA}$) and from now on omit the $\mathcal{LRA}$ specification.

Definition 3 (Weighted Model Integral)**.** *Given a set* $\mathbf{b}$ *of* M *Boolean variables,* $\mathbf{x}$ *of* N *real variables, a measurable weight function* $w : \mathbb{B}^M \times \mathbb{R}^N \to \mathbb{R}_{\geq 0}$*, and a support* ϕ *in the form of an SMT formula over* $\mathbf{b} \cup \mathbf{x}$*, the weighted model integral is*

$$\text{WMI}(\phi, w|\mathbf{x}, \mathbf{b}) = \sum_{\mathbf{b}_\mathcal{I} \in \mathcal{I}(\phi)} \int_{\mathcal{I}(\phi^{\mathbf{b}_\mathcal{I}})} w(\mathbf{x}_\mathcal{I}, \mathbf{b}_\mathcal{I}) d\mathbf{x}_\mathcal{I} \qquad (3)$$

The symbol $\mathcal{I}(\phi^{\mathbf{b}_\mathcal{I}})$ *denotes the set of satisfying interpretations where Boolean variables have been substituted by the values in the interpretation* $\mathbf{b}_\mathcal{I}$*. In the absence of continuous variables (*$\mathbf{x}=\emptyset$*) WMI reduces naturally to WMC*[2]*.*

Formulations of WMI such as in Definition 3 usually allow for combinatorics in the weight function by using indicator functions, for instance, $w(x) = [\![x > 0]\!]x + [\![x \leq 0]\!]x^2$ (for this weight function to be measurable we assume x to be bounded, for instance on the $[-10, 10]$ interval).

Following observations made by Kolb et al. [2019a], we reformulate weighted model integration using a weight function with no (hidden) combinatorics.[3]

Example 1. *Let us consider again the weight function* $w(x) = [\![x > 0]\!]x + [\![x \leq 0]\!]x^2$*. We can pull out the combinatorics of the weight function by introducing a fresh Boolean variable* $b_{x>0} \leftrightarrow (x > 0)$*. Now we conjoin the support* ϕ *with* $(b_{x>0} \leftrightarrow (x > 0))$ *and add a dependence on* $b_{x>0}$ *in the weight function:* $w(x, b_{x>0}) = x$ *and* $w(x, \neg b_{x>0}) = x^2$*.*

Pulling out the combinatorics of the weight function foreshadows the separation of the WMI problem into a combinatorics problem and an integration problem (cf. Section 4). While the combinatorics problem is also present in WMC, the integration problem is unique to WMI.

In order to explicitly indicate that we are working with a combinatorics-free weight function we will use the notation $w_\mathbf{b}(\mathbf{x})$ instead of $w(\mathbf{x}, \mathbf{b})$. The weighted model integral is now expressed as:

$$\text{WMI}(\phi, w|\mathbf{x}, \mathbf{b}) = \sum_{\mathbf{b}} \int [\![\phi(\mathbf{x}, \mathbf{b})]\!] w_\mathbf{b}(\mathbf{x}) d\mathbf{x} \qquad (4)$$

Note that we wrote the weighted model integral as a summation and integration over all values of the variables in $\mathbf{b}$ and $\mathbf{x}$, instead of the satisfying interpretations of ϕ. The non-satisfying interpretations are filtered out by the indicator function $[\![\phi(\mathbf{x}, \mathbf{b})]\!]$.

In Equation 4 we see the two sub-problems that make up WMI: 1) finding all disjoint regions encoded in the support $\phi(\mathbf{x}, \mathbf{b})$ and 2) integrating out the continuous variables over these regions. Kolb et al. [2019a] dubbed the region finding problem λ-SMT. In the case of exclusively linear constraints, the regions form convex polytopes.

[2]An other special case of WMI that has received some attention in its own right is the unweighted case – also referred to as #SMT (just as #SAT is the unweighted case of WMC) [Ma *et al.*, 2009; Phan and Malacaria, 2014; Chistikov *et al.*, 2015; Zhou *et al.*, 2015].

[3]Note that in practice many WMI solver do no pull out the combinatorics of the weight function as it might be detrimental to the run time of the solver. We perform this step for the sake of exposition.

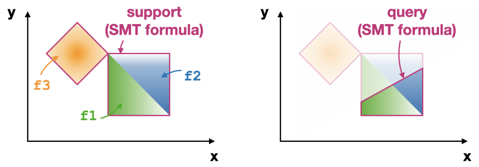

Figure 1: Example of a piecewise distribution with three pieces described by different weight functions f_1, f_2 and f_3. The conditions of the pieces and the feasible area are modeled by univariate and multivariate constraints in the support, as is the query.

Probabilities. In general, a weighted model integral returns a positive real number, which we can easily map to a probability by dividing the weighted model integral by a normalization constant $Z = \mathrm{WMI}(\sigma, w|\mathbf{x}, \mathbf{b})$, where σ denotes the support of the (unnormalized) distribution.

$$p_{w,\sigma}(\phi) = \frac{\mathrm{WMI}(\phi \wedge \sigma, w|\mathbf{x}, \mathbf{b})}{Z} \tag{5}$$

Conditional Probabilities. Let us assume we would like to query the probability of the SMT formula ϕ being satisfied given the SMT formula ψ (see Figure 1 for an example). We can express this conditional probability in terms of two weighted model integrals:

$$p_{w,\sigma}(\phi|\psi) = \frac{p_{w,\sigma}(\phi \wedge \psi)}{p_{w,\sigma}(\psi)} = \frac{\mathrm{WMI}(\phi \wedge \psi \wedge \sigma, w|\mathbf{x}, \mathbf{b})}{\mathrm{WMI}(\psi \wedge \sigma, w|\mathbf{x}, \mathbf{b})} \tag{6}$$

3 Expressiveness

The generic formulation introduced in Section 2 allows a wide range of problems to be modeled. However, performing inference using such an expressive formulation can be costly. Therefore, many existing approaches restrict the expressiveness of the problems they solve by imposing additional constraints on the formulation.

These simplifying assumptions commonly fall into one of three categories: 1) imposing constraints upon the co-occurrence of continuous variables within SMT literals; 2) assuming the support to be in *conjunctive normal form* (CNF) and imposing constraints upon the co-occurrence of variables within clauses[4]; and 3) restricting the parametric families used in the weight function.

3.1 Co-occurrence of Variables Within Literals

In propositional logic, literals consist of either a constant truth value, a Boolean variable or its negation. Literals thus contain at most one variable. In SMT, however, multiple continuous variables may co-occur within a literal. For example, the variables x and y both occur in the literal $x + 1.5y \leq 3$. These co-occurrences introduce dependencies between literals that complicate probabilistic inference.

Univariate Literals. To avoid any such complications, several approaches [Belle *et al.*, 2016; Molina *et al.*, 2018; Zuidberg Dos Martires *et al.*, 2019] disallow any co-occurrence of variables within literals, restricting themselves to problems with *univariate* inequalities of the form $\pm x \leq c$, where c is a constant. Recall that probabilistic inference requires two steps: solving the λ-SMT problem and integrating polytopes. By using univariate literals, the integration of a polytope can be simplified:

$$\mathrm{WMI}(\phi, w|\mathbf{x}, \mathbf{b}) = \sum_{\mathbf{b}} [\![\phi(\mathbf{b})]\!] \int \prod_{x \in \mathbf{x}} [\![\phi(x)]\!] w_{\mathbf{b}}(\mathbf{x}) d\mathbf{x} \tag{7}$$

By further restricting the weight function to be fully-factorizable, i.e., $w(\mathbf{x}) = \prod_{x \in \mathbf{x}} w_x(x)$, the integration step itself can, for many families of weight functions, be computed in polynomial time:

$$\mathrm{WMI}(\phi, w|\mathbf{x}, \mathbf{b}) = \sum_{\mathbf{b}} [\![\phi(\mathbf{b})]\!] \prod_{x \in \mathbf{x}} \int [\![\phi(x)]\!] w_{\mathbf{b}}(x) dx \tag{8}$$

Structured Multivariate Literals. Supporting only univariate literals allows for more efficient inference, however, it comes at a steep cost in expressiveness, as no multivariate dependencies or queries can be supported. If the co-occurrence of variables within literals fulfills certain properties, the inference task can still be simplified. One tool to analyze such properties is the Literal Interaction Graph (LIG) [Derkinderen *et al.*, 2020]. The nodes of the LIG are the continuous variables and there is an edge between any two variables that co-occur in a literal. If a formula has an LIG that contains multiple components C, the integration over the formula can be nested to integrate the variables of one component at a time. The integral to be solved in a WMI problem becomes:

$$\int \prod_{\mathbf{x}_c \in C} [\![\phi_c(\mathbf{x}_c)]\!] w_{\mathbf{b}}(\mathbf{x}) d\mathbf{x} \tag{9}$$

$$= \int [\![\phi_{c_1}(\mathbf{x}_{c_1})]\!] ... \int [\![\phi_{c_n}(\mathbf{x}_{c_n})]\!] w_{\mathbf{b}}(\mathbf{x}) d\mathbf{x}_{c_n} ... d\mathbf{x}_{c_1} \tag{10}$$

If, additionally, the weight function factorizes into the same components, all components can be integrated independently.

$$(10) = \prod_{\mathbf{x}_c \in C} \left(\int [\![\phi_c(\mathbf{x}_c)]\!] w_{\mathbf{b},c}(\mathbf{x}_c) d\mathbf{x}_c \right) \tag{11}$$

The complexity of integration will then depend on the maximal size of the components.

Exploiting additional properties such as the tree-width or the cardinalities of edges in the tree decomposition of the LIG is a topic of current research. In this line of work, restrictions are typically not imposed upfront, rather structure in problems is dynamically recognized and exploited [Kolb *et al.*, 2019b; Derkinderen *et al.*, 2020].

3.2 Co-occurrence of Variables Within Clauses

In addition to restricting the co-occurrence of variables within literals, several approaches based on message-passing [Zeng and Van den Broeck, 2019; Zeng *et al.*, 2020b; Zeng *et al.*, 2020a] assume the support to be in CNF and restrict the co-occurrence of variables within *clauses* of the support. Intuitively, dependencies between variables co-occurring in

[4]A clause is a disjunction of literals, e.g., $(x \leq y) \vee (x \leq z) \vee a$, and a formula in CNF consists of a conjunction of clauses.

clauses arise because the truth values of literals in a clause affect each other. By allowing at most two variables to co-occur within a clause, message-passing algorithms can aggregate messages using partial symbolic integrations over a single variable at a time. For these algorithms to converge, they additionally require the dependencies between variables to be acyclic, as is the case for discrete message passing algorithms.

3.3 Parametric Families for the Weight Function

The parametric family used to express the continuous densities has a large impact on the ability to efficiently integrate over it, especially for multivariate linear constraints. As a result, most approaches relying on exact integration use piecewise polynomial densities [Sanner and Abbasnejad, 2012; Belle *et al.*, 2015a; de Salvo Braz *et al.*, 2016]. This class of functions can approximate any density with arbitrary precision while being closed under integration over convex polytopes. Additionally, the integration problem has been extensively studied in literature and addressed with a variety of techniques (see Sec.4.2).

The adoption of piecewise densities was motivated by earlier work on mixtures of truncated exponentials (MTE) [Cobb and Shenoy, 2005; Cobb and Shenoy, 2017] and mixtures of piecewise polynomials (MOP) [Shenoy and West, 2011]. While MTE and MOP are limited to hyper-rectangular decompositions of the joint density, most of the piecewise-polynomial approaches considered here allow for arbitrary linear decompositions. When allowing only univariate conditions or using approximate integration, other types of functions can be used as well. Several approaches have opted for Gaussians or mixtures of Gaussians [Gutmann *et al.*, 2010a; Merrell *et al.*, 2017; Zuidberg Dos Martires *et al.*, 2019].

4 Paradigms and Solvers

Inference in this setting poses two main challenges: enumerating [5] the solutions of a combinatorial space and integrating the density function over convex regions. The first problem is well studied in the literature. While solving it exactly is #P-complete in general [Valiant, 1979], modern algorithms can scale up to thousands of variables. As described later, some successful ideas proposed in discrete inference were also applied in hybrid settings. The second problem has received less attention in the probabilistic inference field. Algebraic constraints among multiple variables pose considerable challenges. In this context, many known families are not closed under marginalization and the integration problem is typically computationally hard. In fact, even computing the volume of a convex polytope is #P-hard [Dyer and Frieze, 1988].

4.1 Combinatorial Enumeration

We now review ideas that have been used to address the combinatorial enumeration problem (λ-SMT) in WMI.

[5]For lack of better words, we use "enumeration" to describe the process of eliciting, all possible (countable or uncountable) solutions. Note that in practice, solvers never perform this elicitation exhaustively.

DPLL-based Inference. A popular approach for solving WMC is based on exhaustive DPLL search [Birnbaum and Lozinskii, 1999] and has been generalized to the hybrid setting. The PRAiSE system implements the DPLL procedure using a custom search engine called Probabilistic Inference Modulo Theories (PIMT) [de Salvo Braz *et al.*, 2016]. Alternatively, external oracles such as modern CDCL-based lazy SMT solvers have been used [Belle *et al.*, 2015a; Morettin *et al.*, 2017]. The latter family of techniques is well suited for problems with a heavy combinatorial component, where it can fully leverage the latest developments in SMT solving. For instance, WMI-PA [Morettin *et al.*, 2017] uses *predicate abstraction* [Lahiri *et al.*, 2006] instead of a block clause enumeration strategy, which lead to significant speed ups in run time, especially when density functions contain combinatorics themselves.

Under the simplifying assumption of univariate constraints, efficient DPLL-based techniques developed for solving WMC can be ported to WMI. For example, WMI-CC [Belle *et al.*, 2016] showed that component caching (CC) [Sang *et al.*, 2004] avoids redundant path enumerations and, hence, recomputing integrals.

Knowledge Compilation. Many state-of-the-art approaches to probabilistic inference for WMC are based on knowledge compilation [Darwiche and Marquis, 2002]. The idea is to split up inference in a computationally hard off-line step that yields an evaluable data structure and a tractable on-line step [Chavira and Darwiche, 2008]. Given a compiled structure, called arithmetic circuit, probabilistic inference can be performed cheaply and repeatedly for different parameters (weight functions).

Using circuit representations for hybrid probabilistic problems was pioneered by Sanner and Abbasnejad [2012]. They introduced the *symbolic variable elimination* (SVE) algorithm, which performs symbolic integration on a so-called *extended algebraic decision diagram* (XADD). Later on, Kolb et al. [2018a] improved upon SVE by implementing a caching scheme that reuses partial symbolic computations. An alternative circuit representation for WMI problems, based on *sentential decision diagrams* (SDDs) [Choi *et al.*, 2013] and called XSDDs, was introduced in a couple of recent papers [Zuidberg Dos Martires *et al.*, 2019; Kolb *et al.*, 2019b], which lead to the development of a set of inference algorithms: Symbo, Sampo, and F-XSDD. In an independent effort, compilation to XSDDs was also introduced by Fuxjaeger and Belle [2020] who also presented the solver WMISDD for separable non-linear constraints.

A key issue in probabilistic inference is variable ordering, i.e., determining the order in which one wants to marginalize out variables. Finding the optimal variable ordering that minimizes the size of a compiled circuits is an NP-hard problem in the context of hybrid inference. This issue has again been studied by adopting and extending techniques from the WMC setting [Derkinderen *et al.*, 2020].

Note that current knowledge compilation approaches use bottom-up compilation. A direction for future research could be to investigate top-down compilation, which is based on DPLL search. In the discrete setting, top-down compilation has been shown to outperform bottom-up compilation [Huang and Darwiche, 2004]. This would require the unification of ideas from extended decision diagram approaches [Sanner and Abbasnejad, 2012] and modern DPLL approaches [Morettin *et al.*, 2017].

AND/OR Search-Based Inference. Zeng et al. [2019] proposed a unique approach to WMI, based on the observation that a volume computation can be reduced to univariate integration of a specific polynomial. The algorithm, dubbed *search-based model integration* (SMI), first reduces WMI to a volume computation by introducing additional continuous variables, in order to incorporate both the discrete combinatorics and the weight function into a possibly non-convex volume. Then, it uses a variant of AND/OR search, which makes it closely related to DPLL based approaches, to find the target piecewise polynomial whose integral is equivalent to the resulting volume. Notably, the AND/OR search assumes a tree-shaped dependency graph, but the reduction to MI preserves the treewidth of the problem only if the weights are monomials, thus limiting its use to restricted settings.

Message Passing. Building upon SMI, a recent approach solves WMI by reducing it to message passing on a pairwise factor graph with bivariate factors. This approach assumes the support to be in CNF, allows for at most two variables to co-occur within a clause and requires the weight function to be factorizable into bivariate polynomials over variables that co-occur in the support (see Section 3.2). Under this assumption, message aggregations conveniently reduce to symbolic products and partial integrations over one variable at the time. This approach is implemented in MPWMI [Zeng *et al.*, 2020b], which is exact on acyclic factor graphs.

Inspired by the *relax, compensate then recover* (RCR) [Darwiche, 2010] algorithm for discrete approximate inference, ReCoIn (*RElax, COmpensate then INtegrate*) [Zeng *et al.*, 2020a], lifts the acyclicity assumption by first removing some edges in the factor graph to obtain an acyclic relaxation of the original problem. Pairwise dependencies $(x \bowtie y)$ are relaxed by first introducing a copy variable $(x = x') \wedge (x' \bowtie y)$ and relaxing (removing) the equality constraint. The algorithm compensates for the relaxed equalities by optimizing the parameters of newly introduced potentials in order to match the marginals of x and x'. Then, MPWMI can be used to compute exact inference on the acyclic relaxation of the original problem.

Sampling. A variety of sampling-based approaches have been proposed to deal with situations where an exact solution is out of reach. It is well known that traditional sampling techniques have poor performance in highly structured spaces. Several approaches that have been proposed to mitigate this problem for discrete probabilistic inference have extended to the hybrid case. Discrete approaches using hashing-based projections that rely on decision- [Meel *et al.*, 2016] or optimization oracles [Ermon *et al.*, 2013] have been extended to use SMT oracles to approximate the volume of a an SMT formula [Chistikov *et al.*, 2015] or its weighted model integral [Belle *et al.*, 2015b] for supports in CNF (ApproxWMI-CNF). While approximating the discrete WMC problem for supports in CNF is NP-hard, requiring the support to be in *disjunctive normal form* (DNF) makes WMC amenable to fully polynomial randomized approximation schemes (FPRAS). Abboud et al. [2020] have extended this approach in their ApproxWMI-DNF solve, proving that WMI for supports in DNF also admits FPRAS by building on FPRAS for volume estimation of unions of convex bodies [Dyer *et al.*, 1991; Bringmann and Friedrich, 2010].

4.2 Integration

Decomposition into Simplices. Early approaches to WMI used polynomial weight functions, which was driven by the availability of LattE [Baldoni *et al.*, 2011; De Loera *et al.*, 2013]. LattE is a high-performance integration library for exact integration of polynomials on convex polytopes. The algorithm works by decomposing the convex polytope into (signed) simplices and computes integrals by summing up the (signed) values of the integral in each simplex. The complexity scales exponentially in the number of dimensions, and the run-time for computing integrals starts slowing down considerably for $\approx$10-dimensional polytopes. VINCI [Büeler *et al.*, 2000] is a competitor of LattE, based on the same principle. Even though LattE is technically a symbolic algorithm, we consider it to be a numeric integration algorithm as it can only produce definite integrals.

Fourier-Motzkin Elimination. A second exact integration technique that has found use in probabilistic inference under algebraic constraints is symbolic integration. Symbolic integration is one of the core functionalities of any computer algebra system (e.g., SymPy [Meurer *et al.*, 2017]). Contrarily to the simplex decomposition approach, symbolic integration algorithms integrate out variables one by one, always using Fourier-Motzkin elimination [Imbert, 1990].[6] The advantage of symbolic integration is that it produces intermediate results after integrating out each variable in a functional form (e.g. symbolic expression tree). These functional intermediate results can be cached and used in a dynamic programming algorithm [Kolb *et al.*, 2018a; Zuidberg Dos Martires *et al.*, 2019] or in a message passing scheme [Zeng *et al.*, 2020b]. A disadvantage of symbolic integration with regards to the simplex decomposition approach is its poor scaling behavior: integrating polynomials in dimensions >3 over convex polytopes becomes practically infeasible.

Monte Carlo Approximation. As integration is inevitably a computationally hard problem, the only viable option to scale to high dimensions is approximation. Monte Carlo (MC) approximation is well-studied for unconstrained probabilistic inference but has received less attention in the constrained probabilistic inference literature. The problem is that Monte Carlo samplers that have been developed for the unconstrained setting will fail in high-dimensional constrained spaces: the sample rejection rate will simply explode. This is also true for Sampo, the first MC-based WMI algorithm presented in [Zuidberg Dos Martires *et al.*, 2019]. Investigating ad-hoc MC integration techniques is, however, a promising future research direction, especially as

[6]Fourier-Motzkin elimination is to a system of inequalities what Gaussian elimination is to a system of equalities.

Algorithm	Combinatorial Enumeration		Integration			Expressiveness	
	Exact	Method	Exact	Sym.	Method	Parametric form	Assumptions
WMI-CC	✓	DPLL-SMT	✓		LattE + CC	Mul. polynomial	UC
WMI-PA *	✓	DPLL-SMT	✓		LattE	Mul. polynomial	-
PRAiSE	✓	DPLL-PIMT	✓	✓	PIMT	Mul. polynomial	-
SVE	✓	KC-XADD	✓	✓	XADD	Mul. polynomial	-
BR *	✓	KC-XADD	✓	✓	XADD	Mul. polynomial	-
F-XSDD *	✓	KC-XSDD	✓	✓	XADD / PSI	Mul. polynomial	-
WMISDD	✓	KC-XSDD	✓		SciPy/ LattE	Mul. polynomial	-
Symbo *	✓	KC-XSDD	✓	✓	PSI	Mul. Gaussian	UC
Sampo	✓	KC-XSDD			MC	Mul. Gaussian	-
SMI	✓	MI + AND/OR search	✓	✓	univ. integration	Biv. monomials	BC, CNF, A
MPWMI *	✓	MP	✓	✓	SymPy	Biv. polynomial	BC, CNF, A
ReCoIn		MP	✓	✓	SymPy	Biv. polynomial	BC, CNF
ApproxWMI-CNF		hashing + SMT	✓		LattE	Mul. polynomial	CNF
ApproxWMI-DNF		FPRAS	✓		LattE	Mul. polynomial	DNF

Table 1: Overview of the existing implementations: whether they solve the combinatorial enumeration exactly or approximately and what method they use; whether they solve the integration exactly or approximately, symbolically or numerically and what method they use; their expressiveness expressed in terms of 1) the parametric form of the weight function; 2) whether they restrict the support to contain only acyclic (A), univariate (UC) or bivariate constraints (BC); and whether they require a certain form (CNF or DNF). Solvers marked with * are included in the pywmi suite [Kolb *et al.*, 2019a].

samples can be drawn in polytime from log-concave functions constrained by convex polytopes [Lovász and Vempala, 2006] (an implementation can be found in the VolEsti library [Emiris and Fisikopoulos, 2014]). First attempts in this direction have been presented in [Afshar *et al.*, 2016; Zuidberg Dos Martires and Kolb, 2020].

4.3 Solvers

We provide an overview of the existing implementations and report the tools that they are based on. The inference algorithms relying on an external SMT oracle either use MathSAT [Cimatti *et al.*, 2013] or Z3 [de Moura and Bjørner, 2008]. Numerical integration is solved exactly using LattE or approximately with MC estimates. Symbolic integration is either based on external tools like PSI [Gehr *et al.*, 2016], SciPy [Virtanen *et al.*, 2020] or SymPy [Meurer *et al.*, 2017], or it is performed via operations on XADDs. Compilation to XSDDs is built upon pysdd[7].

Table 1 illustrates how the available systems handle the combinatorial enumeration and integration. We report the parametric forms supported for the continuous distributions and the assumption made: univariate or bivariate constraints only, whether the constraints must be specified in CNF / DNF and whether the algorithm assumes an acyclic structure.

5 WMI and Probabilistic Programming

From a software perspective, weighted model integration solvers can be regarded as computer algebra systems (think (Matlab or SymPy) that perform probabilistic inference via symbol manipulations, e.g. simplifications of algebraic constraints or parametric functions. This is especially true for WMI solvers that perform exact inference. It should, hence, not come as a surprise that probabilistic programming languages that rely on exact symbolic inference have also started to support algebraic constraints.

[7]https://github.com/wannesm/PySDD

An early attempt of including algebraic constraints in probabilistic programming is the work of Gutmann et al. [2010b], who extended the probabilistic programming language ProbLog [Fierens *et al.*, 2015] with continuous variables by allowing for univariate constraints on normally distributed variables. Univariate constraints are actually a common constraint imposed on random variables and supported by many probabilistic programming languages. Stan [Carpenter *et al.*, 2017], for instance, supports univariately constrained variables by transforming them into unconstrained variables and performing inference in the unconstrained space.

Allowing for more expressive (linear multivariate) constraints necessitates more powerful symbolic inference engines to be included in the probabilistic inference engine. Probabilistic languages in this direction are (PCLP) [Michels *et al.*, 2015], FairSquare [Albarghouthi *et al.*, 2017], and PSI.

It is noteworthy that Michels et al. [2015] developed, independently of the WMI literature, a generalization of weighted model counting that enables them to perform probabilistic inference in the PCLP language [Michels *et al.*, 2013; Michels *et al.*, 2016]. Of interest to the WMI community might be their approximate inference algorithm [Michels *et al.*, 2016] that iteratively splits up the feasible space into mutually exclusive pieces and calculates bounds for each piece. This translates to iteratively tighter and tighter error bounds on the queried probabilities.

As noted by Zuidberg Dos Martires et al. [2019], probabilistic languages that handle algebraic constraints (PSI and PCLP) usually fail to handle well logical constraints, too. To tackle this issue, an interesting avenue of future research would therefore be the deployment of WMI solvers as the inference backend of probabilistic programming languages. First efforts in this direction have been presented in [Zuidberg Dos Martires *et al.*, 2018] and [Zuidberg Dos Martires, 2020].

6 Future Directions

Most of the research work on hybrid probabilistic inference with constraints has focused on the task of computing marginals. While many of the proposed approaches can be naively adapted to handle MAP inference by combining enumeration and maximization, support for this type of queries in existing implementations is still lacking, and further research is needed to make MAP inference with constraints efficient.

As mentioned in Section 2, first approaches that handle hybrid problems beyond SMT($\mathcal{LRA}$) have been proposed and are interesting research directions in terms of expressivity. However, given the complexity gap between linear and non-linear arithmetic, the gain in expressivity comes at a cost. For instance, non-linear arithmetic over integers is undecidable in general [Matiyasevich, 1993]. A promising direction for oracle-based approaches is the recently proposed incremental linearization approach [Cimatti *et al.*, 2018], that relies on an abstraction-refinement loop on top of solvers for linear arithmetic and uninterpreted functions.

Despite these forays into more expressive problems, efficiently computing query probabilities remains an issue to date and a large part of research efforts in this field remain dedicated to improving inference algorithms. Approximate inference algorithms can handle expressive models, while exact inference algorithms can deliver more accurate results. However, exact algorithms typically impose some restrictions on the model expressiveness to become more efficient. One interesting direction has been to eschew upfront restrictions on the expressiveness in favor of recognizing structure in problems on the fly and exploiting it when present [Kolb *et al.*, 2019b]. The challenge then is to both efficiently recognize structure and exploit it effectively. Another direction has been to identify subsets that can be solved efficiently or even in polynomial time and learn such representations from data rather than modeling them by hand [Molina *et al.*, 2018].

While this survey focuses on the probabilistic inference task, learning hybrid probabilistic models from data is its natural counterpart. On the one hand, most unconstrained hybrid models have been equipped with learning procedures [Murphy, 1998; Yang *et al.*, 2014; Molina *et al.*, 2018]. Constraint learning, on the other hand, has traditionally mostly focused on discrete domains [Beldiceanu and Simonis, 2012; Bessiere *et al.*, 2017] – for an overview see [De Raedt *et al.*, 2018]. Techniques for inducing SMT theories from data have been developed in the area of program synthesis [Gulwani *et al.*, 2017], or by casting learning itself as an SMT problem [Kolb *et al.*, 2018b]. Directly learning constrained hybrid models has received less attention. A simple but effective solution consists in separately learning the constraints and the unconstrained hybrid distribution, and renormalizing the latter to account for the former [Morettin *et al.*, 2020]. Nonetheless, much work is still to be done in order to develop generic, effective and robust learning strategies.

The techniques developed in this line of research already found application in maritime safety and security tasks [Velikova *et al.*, 2014], fairness verification of probabilistic programs [Albarghouthi *et al.*, 2017] and logistics [Morettin *et al.*, 2019]. Despite these promising applications, research mostly focused on theoretical aspects and the practical potential of these techniques is still vastly unexplored.

Acknowledgments

This research was partially supported by TAILOR, a project funded by EU Horizon 2020 research and innovation programme under GA No 952215 and from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. [694980] SYNTH: Synthesising Inductive Data Models). PZD is supported by KU Leuven Special Research Fund.

References

[Abboud *et al.*, 2020] R. Abboud, İ İlkan Ceylan, and R. Dimitrov. On the approximability of weighted model integration on DNF structures. In *KR*, 2020.

[Afshar *et al.*, 2016] H.M. Afshar, S. Sanner, and C. Webers. Closed-form gibbs sampling for graphical models with algebraic constraints. In *AAAI*, 2016.

[Albarghouthi *et al.*, 2017] A. Albarghouthi, L. D'Antoni, S. Drews, and A.V. Nori. Fairsquare: probabilistic verification of program fairness. In *OOPSLA*, 2017.

[Baldoni *et al.*, 2011] V. Baldoni, N. Berline, J. A. De Loera, M. Köppe, and M. Vergne. How to integrate a polynomial over a simplex. *Mathematics of Computation*, 80, 2011.

[Barrett and Tinelli, 2018] C. Barrett and C. Tinelli. Satisfiability modulo theories. In *Handbook of Model Checking*. Springer, 2018.

[Beldiceanu and Simonis, 2012] N. Beldiceanu and H. Simonis. A model seeker: Extracting global constraint models from positive examples. In *CP*, 2012.

[Belle *et al.*, 2015a] V Belle, A. Passerini, and G. Van den Broeck. Probabilistic inference in hybrid domains by weighted model integration. In *IJCAI*, 2015.

[Belle *et al.*, 2015b] V. Belle, G. Van den Broeck, and A. Passerini. Hashing-based approximate probabilistic inference in hybrid domains. In *UAI*, 2015.

[Belle *et al.*, 2016] V. Belle, G. Van den Broeck, and A. Passerini. Component caching in hybrid domains with piecewise polynomial densities. In *AAAI*, 2016.

[Bellman, 1957] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[Bessiere *et al.*, 2017] C. Bessiere, F. Koriche, N. Lazaar, and B. O'Sullivan. Constraint acquisition. *Artificial Intelligence*, 244, 2017.

[Birnbaum and Lozinskii, 1999] E. Birnbaum and E. L. Lozinskii. The Good Old Davis-Putnam Procedure Helps Counting Models. *JAIR*, 10, 1999.

[Bringmann and Friedrich, 2010] K. Bringmann and T. Friedrich. Approximating the volume of unions and intersections of high-dimensional geometric objects. *Computational Geometry*, 43, 2010.

[Bueff *et al.*, 2018] A. Bueff, S. Speichert, and V. Belle. Tractable querying and learning in hybrid domains via sum-product networks. In *KR Workshop on Hybrid Reasoning and Learning*, 2018.

[Büeler *et al.*, 2000] B. Büeler, A. Enge, and K. Fukuda. Exact volume computation for polytopes: a practical study. In *Polytopes—combinatorics and computation*, 2000.

[Carpenter *et al.*, 2017] B. Carpenter, A. Gelman, M. D Hoffman, D. Lee, B. Goodrich, M. Betancourt, M.A. Brubaker, J. Guo, P. Li, and A. Riddell. Stan: a probabilistic programming language. 2017.

[Chavira and Darwiche, 2008] M. Chavira and A. Darwiche. On probabilistic inference by weighted model counting. *AIJ*, 172, 2008.

[Chistikov *et al.*, 2015] D. Chistikov, R. Dimitrova, and R. Majumdar. Approximate counting in SMT and value estimation for probabilistic programs. In *TACAS*, 2015.

[Choi *et al.*, 2013] A. Choi, D. Kisa, and A. Darwiche. Compiling probabilistic graphical models using sentential decision diagrams. In *ECSQARU*, 2013.

[Cimatti *et al.*, 2013] A. Cimatti, A. Griggio, B.J. Schaafsma, and R. Sebastiani. In *TACAS*, 2013.

[Cimatti *et al.*, 2018] A. Cimatti, A. Griggio, A. Irfan, M. Roveri, and R. Sebastiani. Incremental linearization: A practical approach to satisfiability modulo nonlinear arithmetic and transcendental functions. In *SYNASC*, 2018.

[Cobb and Shenoy, 2005] B.R. Cobb and P. P. Shenoy. Hybrid bayesian networks with linear deterministic variables. In *UAI*, 2005.

[Cobb and Shenoy, 2017] B.R. Cobb and P.P. Shenoy. Inference in hybrid bayesian networks with nonlinear deterministic conditionals. *International Journal of Intelligent Systems*, 32, 2017.

[Darwiche and Marquis, 2002] A. Darwiche and P. Marquis. A knowledge compilation map. *JAIR*, 2002.

[Darwiche, 2010] A. Darwiche. Relax, compensate and then recover: A theory of anytime, approximate inference. In *European Conference on Logics in Artificial Intelligence*, 2010.

[De Loera *et al.*, 2013] J. A. De Loera, B. Dutra, M. Koeppe, S. Moreinis, G. Pinto, and J. Wu. Software for exact integration of polynomials over polyhedra. *Computational Geometry*, 46, 2013.

[de Moura and Bjørner, 2008] L. M. de Moura and N. Bjørner. Z3: an efficient SMT solver. In *TACAS*, 2008.

[De Raedt *et al.*, 2018] L. De Raedt, A. Passerini, and S. Teso. Learning constraints from examples. In *AAAI*, 2018.

[de Salvo Braz *et al.*, 2016] R. de Salvo Braz, C. O'Reilly, V. Gogate, and R. Dechter. Probabilistic inference modulo theories. In *IJCAI*, 2016.

[Derkinderen *et al.*, 2020] V. Derkinderen, E. Heylen, P. Zuidberg Dos Martires, S. Kolb, and L. De Raedt. Ordering variables for weighted model integration. In *UAI*, 2020.

[Dyer and Frieze, 1988] M. E. Dyer and A. M. Frieze. On the complexity of computing the volume of a polyhedron. *SIAM Journal on Computing*, 17, 1988.

[Dyer *et al.*, 1991] M. Dyer, A. Frieze, and R. Kannan. A random polynomial-time algorithm for approximating the volume of convex bodies. *JACM*, 38(1), 1991.

[Emiris and Fisikopoulos, 2014] I. Z. Emiris and V. Fisikopoulos. Efficient random-walk methods for approximating polytope volume. In *Annual symposium on Computational geometry*, 2014.

[Ermon *et al.*, 2013] S. Ermon, C. P. Gomes, A. Sabharwal, and B. Selman. Taming the curse of dimensionality: Discrete integration by hashing and optimization. In *ICML*, 2013.

[Fierens *et al.*, 2015] D. Fierens, G. Van den Broeck, J. Renkens, D.S. Shterionov, B. Gutmann, I. Thon, G. Janssens, and L. De Raedt. Inference and learning in probabilistic logic programs using weighted boolean formulas. *Theory Practice of Logic Programming*, 2015.

[Friedman and Goldszmidt, 1996] N. Friedman and M. Goldszmidt. Discretizing continuous attributes while learning bayesian networks. In *ICML*, 1996.

[Fuxjaeger and Belle, 2020] A. Fuxjaeger and V. Belle. Scaling up probabilistic inference in linear and non-linear hybrid domains by leveraging knowledge compilation. In *ICAART*, 2020.

[Gehr *et al.*, 2016] T. Gehr, S. Misailovic, and M. T. Vechev. PSI: exact symbolic inference for probabilistic programs. In S. Chaudhuri and A. Farzan, editors, *CAV*, 2016.

[Gulwani *et al.*, 2017] S. Gulwani, O. Polozov, R. Singh, et al. Program synthesis. *Foundations and Trends in Programming Languages*, 4, 2017.

[Gutmann *et al.*, 2010a] B. Gutmann, M. Jaeger, and L. De Raedt. Extending problog with continuous distributions. In *ILP*, 2010.

[Gutmann *et al.*, 2010b] B. Gutmann, M. Jaeger, and L. De Raedt. Extending problog with continuous distributions. In *ILP*, 2010.

[Hensher and Button, 2007] D. A. Hensher and K. J. Button. *Handbook of Transport Modelling*. Handbooks in Transport. 2007.

[Huang and Darwiche, 2004] J. Huang and A. Darwiche. Using DPLL for efficient OBDD construction. In *SAT*, 2004.

[Imbert, 1990] J. Imbert. About redundant inequalities generated by Fourier's algorithm. In *AIJ*. 1990.

[Kolb *et al.*, 2018a] S. Kolb, M. Mladenov, S. Sanner, V. Belle, and K. Kersting. Efficient symbolic integration for probabilistic inference. In *IJCAI*, 2018.

[Kolb *et al.*, 2018b] S. Kolb, S. Teso, A. Passerini, and L. De Raedt. Learning SMT(LRA) constraints using smt solvers. In *IJCAI*, 2018.

[Kolb *et al.*, 2019a] S. Kolb, P. Morettin, P. Zuidberg Dos Martires, F. Sommavilla, A Passerini, R. Sebastiani, and L. De Raedt. The pywmi framework and toolbox for probabilistic inference using weighted model integration. In *IJCAI*, 2019.

[Kolb *et al.*, 2019b] S. Kolb, P. Zuidberg Dos Martires, and L. De Raedt. How to exploit structure while solving weighted model integration problems. In *UAI*, 2019.

[Koller and Friedman, 2009] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[Kozlov and Koller, 1997] A.V. Kozlov and D. Koller. Nonuniform dynamic discretization in hybrid networks. In *UAI*, 1997.

[Lahiri *et al.*, 2006] S. K. Lahiri, R. Nieuwenhuis, and A. Oliveras. SMT techniques for fast predicate abstraction. In *CAV*, 2006.

[Lauritzen, 1992] S.L. Lauritzen. Propagation of probabilities, means, and variances in mixed graphical association models. *Journal of the American Statistical Association*, 87, 1992.

[Lee, 2008] E. A. Lee. Cyber physical systems: Design challenges. In *Proc. IEEE Symposium on Object Oriented Real-Time Distributed Computing*, 2008.

[Lovász and Vempala, 2006] L. Lovász and S. Vempala. Fast algorithms for logconcave functions: Sampling, rounding, integration and optimization. In *IEEE Symposium on Foundations of Computer Science*, 2006.

[Ma *et al.*, 2009] F. Ma, S. Liu, and J. Zhang. Volume computation for boolean combination of linear arithmetic constraints. In *CADE*, 2009.

[Matiyasevich, 1993] Y. V. Matiyasevich. *Hilbert's Tenth Problem*. MIT Press, 1993.

[Meel *et al.*, 2016] K. S. Meel, M. Y. Vardi, S. Chakraborty, D. J. Fremont, S. A. Seshia, D. Fried, A. Ivrii, and S. Malik. Constrained sampling and counting: Universal hashing meets SAT solving. In *Beyond NP workshop @ AAAI*, 2016.

[Merrell *et al.*, 2017] D. Merrell, A. Albarghouthi, and L. D'Antoni. Weighted model integration with orthogonal transformations. In *IJCAI*, 2017.

[Meurer *et al.*, 2017] A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, T. Rathnayake, S. Vig, B. E. Granger, R. P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M. J. Curry, A. R. Terrel, S. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, and A. Scopatz. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3, January 2017.

[Michels *et al.*, 2013] S. Michels, A. Hommersom, P.J.F. Lucas, M. Velikova, and P. Koopman. Inference for a new probabilistic constraint logic. In *IJCAI*, 2013.

[Michels *et al.*, 2015] S. Michels, A. Hommersom, P.J.F. Lucas, and M. Velikova. A new probabilistic constraint logic programming language based on a generalised distribution semantics. *Artificial Intelligence*, 2015.

[Michels *et al.*, 2016] S. Michels, A. Hommersom, and P.J.F. Lucas. Approximate probabilistic inference with bounded error for hybrid probabilistic logic programming. In *IJCAO*, 2016.

[Molina *et al.*, 2018] A. Molina, A. Vergari, N. Di Mauro, S. Natarajan, F. Esposito, and K. Kersting. Mixed sum-product networks: A deep architecture for hybrid domains. In *AAAI*, 2018.

[Morettin *et al.*, 2017] P. Morettin, A. Passerini, and R. Sebastiani. Efficient weighted model integration via smt-based predicate abstraction. In *IJCAI*, 2017.

[Morettin *et al.*, 2019] P. Morettin, A. Passerini, and R. Sebastiani. Advanced SMT techniques for weighted model integration. *AIJ*, 275, 2019.

[Morettin *et al.*, 2020] P. Morettin, S. Kolb, S. Teso, and A. Passerini. Learning weighted model integration distributions. In *AAAI*, 2020.

[Murphy, 1998] K. P. Murphy. Inference and learning in hybrid bayesian networks. Technical report, USA, 1998.

[Phan and Malacaria, 2014] Quoc-Sang Phan and Pasquale Malacaria. Abstract model counting: a novel approach for quantification of information leaks. In *Symposium on Information, computer and communications security*, 2014.

[Sang *et al.*, 2004] T. Sang, F. Bacchus, P. Beame, H. A. Kautz, and T. Pitassi. Combining component caching and clause learning for effective model counting. In *SAT*, 2004.

[Sanner and Abbasnejad, 2012] S. Sanner and E. Abbasnejad. Symbolic variable elimination for discrete and continuous graphical models. In *AAAI*, 2012.

[Shenoy and West, 2011] P. P. Shenoy and J. C. West. Inference in hybrid bayesian networks using mixtures of polynomials. *IJAR*, 52, 2011.

[Thrun *et al.*, 2005] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 2005.

[Valiant, 1979] L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8, 1979.

[Velikova *et al.*, 2014] M. Velikova, P. Novák, B. Huijbrechts, J. Laarhuis, J. Hoeksma, and S. Michels. An integrated reconfigurable system for maritime situational awareness. In *ECAI*, 2014.

[Virtanen *et al.*, 2020] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, I. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17, 2020.

[Yang *et al.*, 2014] E. Yang, Y. Baker, P. Ravikumar, G.I. Allen, and Z. Liu. Mixed graphical models via exponential families. In *AISTATS*, 2014.

[Zeng and Van den Broeck, 2019] Z. Zeng and G. Van den Broeck. Efficient search-based weighted model integration. In *UAI*, 2019.

[Zeng *et al.*, 2020a] Z. Zeng, P. Morettin, F. Yan, A. Vergari, and G. Van den Broeck. Probabilistic inference with algebraic constraints: Theoretical limits and practical approximations. In *NeurIPS*, 2020.

[Zeng *et al.*, 2020b] Z. Zeng, P. Morettin, F. Yan, A. Vergari, and G. Van den Broeck. Scaling up hybrid probabilistic inference with logical and arithmetic constraints via message passing. In *ICML*, 2020.

[Zhou *et al.*, 2015] M. Zhou, F. He, X. Song, S. He, G. Chen, and M. Gu. Estimating the volume of solution space for satisfiability modulo linear real arithmetic. *Theory of Computing Systems*, 2015.

[Zuidberg Dos Martires and Kolb, 2020] P. Zuidberg Dos Martires and S. Kolb. Monte carlo anti-differentiation for approximate weighted model integration. In *StarAI @ AAAI*, 2020.

[Zuidberg Dos Martires *et al.*, 2018] P. Zuidberg Dos Martires, A. Dries, and L. De Raedt. Knowledge compilation with continuous random variables and its application in hybrid probabilistic logic programming. StarAI @ IJCAI, 2018.

[Zuidberg Dos Martires *et al.*, 2019] P. Zuidberg Dos Martires, A. Dries, and L. De Raedt. Exact and approximate weighted model integration with probability density functions using knowledge compilation. In *AAAI*, 2019.

[Zuidberg Dos Martires, 2020] P. Zuidberg Dos Martires. From atoms to possible worlds: Probabilistic inference in the discrete-continuous domain. *PhD thesis, KU Leuven*, 2020.