Symbolic Variable Elimination for Discrete and Continuous Graphical Models

Anonymous

Abstract

Probabilistic reasoning in the real-world often requires inference in continuous variable graphical models, yet there are few methods for *exact, closed-form* inference when joint distributions are non-Gaussian. To address this inferential deficit, we introduce SVE – a *symbolic* extension of the well-known *variable elimination* algorithm to perform exact inference in an *expressive* class of mixed discrete and continuous variable graphical models whose conditional probability functions can be well-approximated as piecewise combinations of polynomials with bounded support. Using this representation, we show that we can compute all of the SVE operations *exactly and in closed-form*, which crucially includes *definite integration w.r.t. nonlinear piecewise boundary constraints*. To aid in the efficient computation and compact representation of this solution, we use an extended algebraic decision diagram (XADD) data structure that supports all SVE operations. We provide illustrative results for SVE on probabilistic inference queries inspired by robotics localization and tracking applications that use complex continuous distributions; this represents the first time a general closed-form exact solution has been proposed for this expressive class of discrete/continuous graphical models.

Introduction

Real-world probabilistic reasoning is rife with uncertainty over continuous random variables with complex (often nonlinear) relationships, e.g., estimating the position and pose of entities from measurements in robotics, or radar tracking applications with asymmetric stochastic dynamics and complex mixtures of noise processes. While closed-form exact solutions exist for inference in some continuous variable graphical models, such solutions are largely limited to relatively well-behaved cases such as joint Gaussian models. On the other hand, the more complex inference tasks of tracking with real-world sensors involves underlying distributions that are beyond the reach of current closed-form exact solutions. Take, for example, the robot sensor model in Figure 1 motivated by (Thrun et al. 2000). Here $d \in \mathbb{R}$ is a variable representing the distance of a mobile robot to a wall

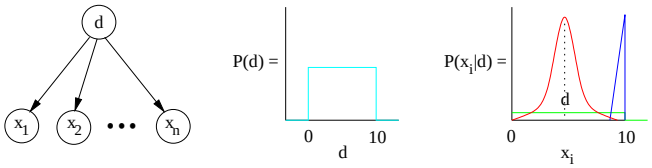

Figure 1: The *robot localization* graphical model and all conditional probabilities.

and x_i are the observed measurements of distance (e.g., using a laser range finder). The prior distribution $P(d)$ is uniform $U(d; 0, 10)$ while the observation model $P(x_i|d)$ is a *mixture* of three distributions: (red) is a truncated Gaussian representing noisy measurements x_i of the actual distance d within the sensor range of $[0, 10]$; (green) is a uniform noise model $U(d; 0, 10)$ representing random anomalies leading to any measurement; and (blue) is a triangular distribution peaking at the maximum measurement distance (e.g., caused by spurious deflections of laser light or the transparency of glass).

While our focus in this paper is on exact inference in *general* discrete and continuous variable graphical models and not purely on this robotics localization task, this example clearly motivates the real-world need for reasoning with complex distributions over continuous variables. However, in contrast to previous work that has typically resorted to (sequential) Monte Carlo methods (Thrun et al. 2000; Doucet, De Freitas, and Gordon 2001) to perform inference in this model (and dynamic extensions) via *sampling*, our point of departure in this work is to seek *exact, closed-form solutions* to general probabilistic inference tasks in these graphical models, e.g., arbitrary conditional probability queries or conditional expectation computations.

To achieve this task, we focus on an expressive class of mixed discrete and continuous variable Bayesian networks whose conditional probability functions can be expressed as piecewise polynomials with bounded support (where nonlinear piece boundaries themselves may consist of inequalities over polynomials). In practice, this representation is sufficient to represent or arbitrarily approximate a wide range of common distributions such as the following: uniform (piecewise constant), triangular (piecewise linear), truncated normal distributions[1] (piecewise quadratic or quartic), as

[1] In practice, measurements have natural ranges that make trun-

well as all mixtures of such distributions as exemplified in $P(x_i|d)$ above (since a sum of piecewise polynomials is still piecewise polynomial). While polynomials directly support the integrals we will need to compute *variable elimination* (Zhang and Poole 1996) in closed-form, computing the definite integrals w.r.t. arbitrary polynomial piece boundaries is a much more difficult task achieved through novel *symbolic* methods that underlie the key contribution of *symbolic variable elimination (SVE)* that we make in this paper.

In the following sections, we present our graphical model framework using a *case* representation of probability distributions followed by a description of our SVE procedure. After this, we introduce the *extended algebraic decision diagram (XADD)* — a practical data structure for efficiently and compactly manipulating the case representation — and apply XADD-based SVE to robotics localization and a tracking tasks. We conclude with a discussion of future extensions.

Discrete and Continuous Graphical Models

Factor Graph Representation

A discrete and continuous *graphical model* compactly represents a joint probability distribution $p(\mathbf{b}, \mathbf{x})$ over an assignment $(\mathbf{b}, \mathbf{x}) = (b_1, \ldots, b_n, x_1, \ldots, x_m)$ to respective random variables $(\mathbf{B}, \mathbf{X}) = (B_1, \ldots, B_n, X_1, \ldots, X_m)$.[2] Each b_i $(1 \leq i \leq n)$ is boolean s.t. $b_i \in \{0, 1\}$ and each x_j $(1 \leq j \leq m)$ is continuous s.t. $x_j \in \mathbb{R}$.

As a general representation of both directed and undirected graphical models, we use a *factor graph* (Kschischang, Frey, and Loeliger 2001) representing a joint probability $p(\mathbf{b}, \mathbf{x})$ as a product of a finite set of factors F, i.e.,

$$p(\mathbf{b}, \mathbf{x}) = \frac{1}{Z} \prod_{f \in F} \Psi_f(\mathbf{b}_f, \mathbf{x}_f). \tag{1}$$

Here, $\mathbf{b}_f$ and $\mathbf{x}_f$ denote the subset of variables that participate in factor f and $\Psi_f(\mathbf{b}_f, \mathbf{x}_f)$ is a non-negative, real-valued potential function that can be viewed as gauging the local compatibility of assignments $\mathbf{b}_f$ and $\mathbf{x}_f$. The functions Ψ_f may not necessarily represent probabilities and hence a normalization constant Z is often required to ensure $\sum_{\mathbf{b}} \int_{\mathbf{x}} p(\mathbf{b}, \mathbf{x}) d\mathbf{x} = 1$.

We will specify all of our algorithms on graphical models in terms of general factor graphs, but we note that *Bayesian networks* represent an important modeling formalism that we will use to specify our examples in this paper. For the Bayesian network directed graph in the introductory example, the joint distribution for n variables is represented as the product of all variables conditioned on their parent variables in the graph and can be easily converted to factor graph form, i.e.,

$$\begin{aligned} p(d, x_1, \ldots, x_k) &= p(d) \prod_{i=1}^{k} p(x_i|d) \\ &= \psi_d(d) \prod_{i=1}^{k} \Psi_{x_i}(x_i, d) \end{aligned} \tag{2}$$

where quite simply, $\Psi_d(d) := p(d)$ and $\Psi_{x_i}(x_i, d) := p(x_i|d)$. Here, $Z = 1$ and is hence omitted since joint distributions represented by Bayesian networks marginalize to 1 by definition.

Variable Elimination Inference

Given a joint probability distribution $p(\mathbf{b}, \mathbf{x})$ defined by a factor graph, our objective in this paper will be to perform two types of *exact, closed-form* inference:

(Conditional) Probability Queries: for query $\mathbf{q}$ and disjoint (possibly empty) evidence $\mathbf{e}$ drawn from a subset of $(\mathbf{b}, \mathbf{x})$, we wish to infer the *exact* form of $p(\mathbf{q}|\mathbf{e})$ as a *function* of $\mathbf{q}$ and $\mathbf{e}$. This can be achieved by the following computation, where for notational convenience we assume variables are renamed s.t. $\{\mathbf{b} \cup \mathbf{x}\} \setminus \{\mathbf{q} \cup \mathbf{e}\} = (b_1, \ldots, b_s, x_1, \ldots, x_t)$ and $\mathbf{q} = (b_{s+1}, \ldots, b_{s'}, x_{t+1}, \ldots, x_{t'})$:

$$\begin{aligned} &p(\mathbf{q}|\mathbf{e}) = \\ &= \frac{\sum_{b_1} \cdots \sum_{b_s} \int \cdots \int_{\mathbb{R}^t} \prod_{f \in F} \Psi_f(\mathbf{b}_f, \mathbf{x}_f)\, dx_1 \ldots dx_t}{\sum_{b_1} \cdots \sum_{b_{s'}} \int \cdots \int_{\mathbb{R}^{t'}} \prod_{f \in F} \Psi_f(\mathbf{b}_f, \mathbf{x}_f)\, dx_1 \ldots dx_{t'}} \end{aligned} \tag{3}$$

The $\frac{1}{Z}$ from (1) would appear in *both* the numerator and denominator and hence cancels.

(Conditional) Expectation Queries: for a continuous query random variable Q and disjoint evidence assignment $\mathbf{e}$ drawn from a subset of $(\mathbf{b}, \mathbf{x})$, we wish to infer the *exact value* of $\mathbb{E}[Q|\mathbf{e}]$.[3] This can be achieved by the following computation, where $p(q|\mathbf{e})$ can be pre-computed according to (3):

$$\mathbb{E}[Q|\mathbf{e}] = \int_{q=-\infty}^{\infty} [q \cdot p(q|\mathbf{e})]\, dq \tag{4}$$

Variable elimination (VE) (Zhang and Poole 1996) is a simple and efficient algorithm given in Algorithm 1 that exploits the distributive law to *efficiently* compute each elimination step (i.e., any $\sum_{b_i}$ or $\int_{x_j}$ in (3) and (4)) by first factorizing out all factors independent of the elimination variable; this prevents unnecessary multiplication of factors, hence minimizing the size and complexity of each elimination operation. If the factor representation is closed and computable under the VE operations of multiplication and marginalization then VE can compute any (conditional) probability or expectation query in (3) and (4). Closed-form, exact solutions for VE are well-known for the discrete and joint Gaussian cases; next we extend VE to expressive piecewise polynomial discrete/continuous graphical models.

Symbolic Variable Elimination in Discrete/Continuous Graphical Models

As discussed in Section , piecewise polynomial functions provide an expressive framework for representing dis-

cated variants of distributions appropriate, e.g., a range finder that exhibits Gaussian distributed measurement errors but only returns measurements in the range $[0, 10]$ may be well-suited to a truncated Normal distribution observation model.

[2]Notational comments: we sometimes abuse notation and treat vectors of random variables or assignments as a set, e.g., $(\mathbf{b}, \mathbf{x}) = \{b_1, \ldots, b_n, x_1, \ldots, x_m\}$. Also we often do not distinguish between a random variable (upper case) and its realization (lower case), e.g., $p(x_1) := p(X_1 = x_1)$.

[3]We do not discuss the expectation of $\{0, 1\}$ boolean random variables B since $\mathbb{E}[B|\mathbf{e}] = P(B = 1|e)$, which can already be computed in (3).

crete/continuous variable graphical models when all distributions have bounded support. In this section, we introduce a case notation and operations for piecewise polynomials, define factor graphs in terms of these case statements, and finally show that all VE operations, including definite integration w.r.t. nonlinear piecewise boundary constraints, can be computed exactly and in closed-form using a purely *symbolic* representation, hence the algorithm *Symbolic VE* (SVE).

Case Representation and Operators

For this section, we will assume all functions are represented in *case* form as follows:

$$f = \begin{cases} \phi_1 & f_1 \\ \vdots & \vdots \\ \phi_k & f_k \end{cases} \tag{5}$$

Here, the f_i may be polynomials of $\mathbf{x}$ with non-negative exponents. The ϕ_i are logical formulae defined over $(\mathbf{b}, \mathbf{x})$ that can consist of arbitrary conjunctions of (a) (negated) boolean variables in $\mathbf{v}$ and (b) inequalities $(\geq, >, \leq, <)$, equalities $(=)$, or disequalities $(\neq)$ where the left and right operands can be in the same form as the f_i. We assume that the set of conditions $\{\phi_1, \ldots, \phi_k\}$ disjointly and exhaustively partition $(\mathbf{b}, \mathbf{x})$ such that f is well defined for all $(\mathbf{b}, \mathbf{x})$. It is easy to verify that such a representation can represent the uniform and triangular distributions and arbitrarily approximate the truncated normal distribution required to specify $p(x_i|d)$ discussed in Section .

Unary operations such as scalar multiplication $c \cdot f$ (for some constant $c \in \mathbb{R}$) or negation $-f$ on case statements f are straightforward; the unary operation is simply applied to each f_i $(1 \leq i \leq k)$. Intuitively, to perform a *binary operation* on two case statements, we simply take the cross-product of the logical partitions of each case statement and perform the corresponding operation on the resulting paired partitions. Thus, we perform the "cross-sum" $\oplus$ of two (unnamed) cases as follows:

$$\begin{cases} \phi_1: & f_1 \\ \phi_2: & f_2 \end{cases} \oplus \begin{cases} \psi_1: & g_1 \\ \psi_2: & g_2 \end{cases} = \begin{cases} \phi_1 \wedge \psi_1: & f_1 + g_1 \\ \phi_1 \wedge \psi_2: & f_1 + g_2 \\ \phi_2 \wedge \psi_1: & f_2 + g_1 \\ \phi_2 \wedge \psi_2: & f_2 + g_2 \end{cases}$$

Likewise, we perform $\ominus$ and $\otimes$ by, respectively, subtracting or multiplying partition values (rather than adding them) to obtain the result. Some partitions resulting from the application of the $\oplus$, $\ominus$, and $\otimes$ operators may be inconsistent (infeasible); if we can detect this (e.g., via a nonlinear constraint solver), we may simply discard such partitions as they are irrelevant to the function value.

For variable elimination, we'll need to compute definite integrals — a fairly non-trivial operation that is discussed in its own section. But first we discuss maximization (needed for working with integral bounds) and restriction (needed to compute marginals for boolean variables).

Symbolic maximization is fairly straightforward to define if we note that the conditional nature of the case statements allows us to directly encode maximization:

$$\max\left(\begin{cases} \phi_1: & f_1 \\ \phi_2: & f_2 \end{cases}, \begin{cases} \psi_1: & g_1 \\ \psi_2: & g_2 \end{cases}\right) = \begin{cases} \phi_1 \wedge \psi_1 \wedge f_1 > g_1: & f_1 \\ \phi_1 \wedge \psi_1 \wedge f_1 \leq g_1: & g_1 \\ \phi_1 \wedge \psi_2 \wedge f_1 > g_2: & f_1 \\ \phi_1 \wedge \psi_2 \wedge f_1 \leq g_2: & g_2 \\ \phi_2 \wedge \psi_1 \wedge f_2 > g_1: & f_2 \\ \phi_2 \wedge \psi_1 \wedge f_2 \leq g_1: & g_1 \\ \phi_2 \wedge \psi_2 \wedge f_2 > g_2: & f_2 \\ \phi_2 \wedge \psi_2 \wedge f_2 \leq g_2: & g_2 \end{cases}$$

The key observation here is that case statements are closed under the max operation. While it may appear that this representation will lead to an unreasonable blowup in size, we note the XADD that we introduce later will be able to exploit the internal decision structure of this maximization to represent it much more compactly.

The two operations required for marginalization over boolean variables b are $\oplus$ and *restriction* of variable b in factor f to the value $x \in 0, 1$, written as $f|_b = x$. For $x = 1$ $(x = 0)$, $f|_b = x$ simply requires instantiating all variables b in f with x. For example, let

$$f = \begin{cases} \phi_1 \wedge b: & f_1 \\ \phi_2 \wedge \neg b: & f_2 \\ \phi_3: & f_3 \end{cases}$$

then the two possible restrictions of b yield the following results (where inconsistent case partitions have been removed):

$$f|_{b=1} = \begin{cases} \phi_1: & f_1 \\ \phi_3: & f_3 \end{cases} \qquad f|_{b=0} = \begin{cases} \phi_2: & f_2 \\ \phi_3: & f_3 \end{cases}.$$

Algorithm 1: VE(F, *order*)

input : F, *order*: a set of factors F, and a variable *order* for elimination
output: a set of factors after eliminating each $v \in$ *order*

```
begin
    // eliminate each v in the given order
    foreach v ∈ order do
        // multiply ⊗ all factors containing v
        // into f_v, put other factors in F_\v
        f_v ← 1; F_\v ← ∅
        foreach f ∈ F do
            if (f contains v)
            then f_v ← f_v ⊗ f
            else F_\v ← F_\v ∪ {f}
        // eliminate var; insert result into factor
        // set F along with F_\v
        if (var is boolean)
        then F ← F_\v ∪ {Σ_{v∈{0,1}} f_v}
        else F ← F_\v ∪ {∫_{v=-∞}^{∞} f_v dv}
    return F
end
```

Definite Integration of the Case Representation

One of the major technical contributions of this paper is the symbolic computation of the definite integration required to eliminate continuous variables in SVE. If we are computing $\int_{x_1=-\infty}^{\infty} f \, dx_1$ for f in (5), we can rewrite it in the following equivalent form

$$\int_{x_1=-\infty}^{\infty} \sum_i \mathbb{I}[\phi_i] \cdot f_i \, dx_1 \;=\; \sum_i \int_{x_1=-\infty}^{\infty} \mathbb{I}[\phi_i] \cdot f_i \, dx_1 \quad (6)$$

where $\mathbb{I}[\cdot]$ is an indicator function taking value 1 when its argument is true, 0 when it is false. Hence we can compute the integrals separately for each case partition (producing a case statement) and then $\sum$ the results using $\oplus$.

To continue with the integral for a single case partition, we introduce a concrete example. Let $f_1 := x_1^2 - x_1 x_2$ and $\phi_1 := [x_1 > -1] \wedge [x_1 > x_2 - 1] \wedge [x_1 \leq x_2^2] \wedge [x_1 \leq x_2 + 1] \wedge [x_2 > 0]$. In computing $\int_{x_1=-\infty}^{\infty} \mathbb{I}[\phi_1] \cdot f_1 \, dx_1$, the first thing we note is that the constraints involving x_1 in $\mathbb{I}[\phi_1]$ can be used to restrict the integration range for x_1. In fact, we know that the integrand is only non-zero for $\max(x_2 - 1, -1) < x_1 \leq \min(x_2^2, x_2 + 1)$. Using the max operation defined previously, we can write explicit functions in *piecewise polynomial case form* for the respective lower and upper bounds LB and UB:

$$LB := \begin{cases} x_2 - 1 > -1 : x_2 - 1 \\ x_2 - 1 \leq -1 : -1 \end{cases} \qquad UB \quad := \begin{cases} x_2 < x_3 + 1 : x_2 \\ x_2 \geq x_3 + 1 : x_3 + 1 \end{cases}$$

Now we can rewrite the integral as[4]

$$\mathbb{I}[x_2 > 0] \int_{x_1=LB}^{UB} (x_1^2 - x_1 x_2) \, dx_1. \quad (7)$$

Note here that $\mathbb{I}[x_2 > 0]$ is independent of x_1 and hence can factor outside the integral. With all indicator functions moved into the LB or UB or factored out, we can now compute the integral:

$$\mathbb{I}[x_2 > 0] \left[\frac{1}{3} x_1^3 - \frac{1}{2} x_1^2 x_2 \Big|_{x_1=LB}^{x_1=UB} \right]. \quad (8)$$

The question now is simply how to do this evaluation? Here we note that every expression (variables, constants, indicator functions, etc.) can be written as a simple case statements or as operations on case statements, even the upper and lower bounds as shown previously. So the evaluation is simply

$$\mathbb{I}[x_2 > 0] \otimes \left[\left(\frac{1}{3} UB \otimes UB \otimes UB \ominus \frac{1}{2} UB \otimes UB \otimes (x_2) \right) \right.$$
$$\left. \ominus \left(\frac{1}{3} LB \otimes LB \otimes LB \ominus \frac{1}{2} LB \otimes LB \otimes (x_2) \right) \right]. \quad (9)$$

Hence the result of the definite integration over a case partition of a polynomial function with piecewise nonlinear constraints is simply a case statement — this is somewhat remarkable given that all of the bound computations were symbolic. Furthermore, one might fear that high-order operations like $UB \otimes UB \otimes UB$ could lead to a case partition explosion, but we note this example simply has the effect of cubing the expressions in each partition of UB since all case partitions are disjoint.

However, we are not yet done, there is one final step that we must include for correctness. Because our bounds are symbolic, it may be the case for some assignment to $(\mathbf{b}, \mathbf{x})$ that $LB \geq UB$. In this case the integral should be zero since the constaints on x_1 could not be jointly satisfied. To enforce this symbolically, we simply need to $\otimes$ (9) by case statements representing the following inequalities for all pairs of upper and lower bounds:

$$\mathbb{I}[x_2 > x_2 - 1] \otimes \mathbb{I}[x_2 > -1] \otimes \mathbb{I}[x_3 + 1 > x_2 - 1] \otimes \mathbb{I}[x_3 + 1 > -1] \quad (10)$$

Of course, here $\mathbb{I}[x_2 + 1 > x_2 - 1]$ could be removed as a tautology, but the other constraints must remain.

This provides the solution for a single case partition and from (6), we just need to $\oplus$ the case statements from each case partition integral evaluation to obtain the full integral, *still in piecewise polynomial case form.*

Piecewise Polynomial Factor Graphs and Symbolic Variable Elimination (SVE)

We now proceed to define our factor graph from (1) using factors represented by case statements. The only restriction that we have on *piecewise polynomial factor graphs (PPFG)* requires knowledge of the variable elimination *order* to be used in Algorithm 1 and is stated as follows:

Restriction 1 (Case Factor Restriction for PPFGs)**.** *Given variable elimination order, for all PPFG factors $f(\mathbf{v})$, the $v \in \mathbf{v}$ occurring earliest in order may only occur linearly in any of the constraints ϕ_i of factor $f(\mathbf{v})$.*

In brief, this is to ensure that the bounds on v can be isolated in order to perform the definite integration from Section . We note that this constraint only affects *one* variable in each factor, hence *order* can be adapted as appropriate to accommodate a range of nonlinear modeling requirements. Furthermore, when all piecewise boundaries are linear constraints (which occurs often), this condition is always satisfied.

With the preceding machinery, generalizing VE to Symbolic VE (SVE) for PPFGs is straightforward. We need only replace all sums and products in VE with the case versions $\oplus$ and $\otimes$. Then the only operation we need to specify in the VE Algorithm 1 is to define the computation for the variable eliminations. For PPFGs, the two cases can be computed as follows:

Discrete Variable Elimination (line 14 of VE):

$$\sum_v f_v := f_v|_{v=1} \oplus f_v|_{v=1}.$$

Continuous Variable Elimination (line 15 of VE):

$$\int_{v=-\infty}^{\infty} f_v \, dv := \text{(see Section)}$$

.

This completes the definition of SVE. If the exact model could be represented as a PPFG, then SVE provides exact inference. Otherwise one can approximate most graphical models to arbitrary precision using PPFGs — once this is done, SVE will provide exact inference in this approximated model.

Extended ADDs (XADDs) for Case Statements

In practice, it can be prohibitively expensive to maintain a case statement representation of a value function with explicit partitions. Motivated by algebraic decision diagrams

[4]The careful reader will note that because the lower bounds were defined in terms of $>$ rather than $\leq$, we technically have an improper integral and need to take a limit. However, in taking the limit, we note that all integrands are continuous polynomials of order 0 or greater, so the limit exists and yields the same answer as substituting the limit value. Hence for polynomials, we need not be concerned about whether bounds are inclusive or not.

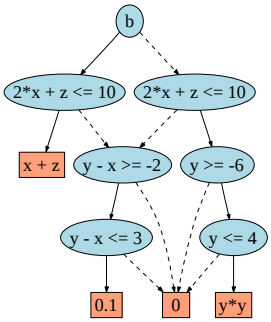

Figure 2: An XADD example from a case statement.

(ADDs) (Bahar et al. 1993), which maintain compact representations for finite discrete functions, we use an extended ADD (XADD) formalism introduced in (Sanner, Delgado, and de Barros 2011) and demonstrated in Figure 2.[5]

In brief we note that an XADD is like an algebraic decision diagram (ADD) (Bahar et al. 1993) except that (a) the decision nodes can have arbitrary inequalities, equalities, or disequalities (one per node) and (b) the leaf nodes can represent arbitrary functions. The decision nodes still have a fixed order from root to leaf and the standard ADD operations to build a canonical ADD (REDUCE) and to perform a binary operation on two ADDs (APPLY) still applies in the case of XADDs.

Of particular importance is that the XADD is a directed acyclic graph (DAG), and hence is often much more compact than a tree representation. For example, in Figure 2 the XADD of a function is provided. We note that not only is the XADD much more compact on account of its DAG, but that all unary and binary operations on XADDs can directly exploit this compact structure for efficiency.

It is fairly straightforward for XADDs to support all case operations required for SVE. Standard operations like unary multiplication, negation, $\oplus$, and $\otimes$ are implemented exactly as they are for ADDs. For the maximization operation, when two leaf nodes f and g are reached and they are general expressions, the additional decision node $f > g$ is introduced to represent the maximum. This may occasionally introduce out-of-order decisions which can be easily detected and repaired as discussed in (Sanner, Delgado, and de Barros 2011) .

Empirical Results

In this section we present proof-of-concept experiments for the robot localization graphical model provided in Section and a basic discrete/continuous tracking task in Figure 4. Our objectives in this section are to show that the previously defined exact inference methods can work with reasonably sized graphical models having multiple continuous and discrete variables with complex piecewise polynomial distributions and that SVE can exactly compute the highly complex, multi-modal queries in an exact functional form — a breakthrough in exact inference for such graphical models.

For the robot localization task, we plot the posterior over the distance $d \in D$ given various observations in Figure 3. What is noteworthy here is the interesting multi-modal nature of these plots. Recalling the discussion in the introduction, the range finder was modeled with various error modes; because of these models, the posterior plots demonstrate the different combinatorial possibilities that each measurement was accurate or noise with the various peaks in the distribution weighted according to these probabilities. Despite the multi-modal nature of these distributions, the conditional expectation of D can be computed exactly and is shown below each posterior. We note that in a Bayesian risk setting, where these posterior beliefs may be multiplied by some position-based risk function (e.g., one that increases with proximity to stairwells), it is important to have this true multi-modal posterior rather than an inaccurate unimodal approximation.

The tracking graphical model and (conditional) distributions are shown in Figure 4 and works as follows: an object has hidden state $x_i \in \mathbb{R}$ defining it's x position that is sampled for time steps $i = 1 \ldots t$. On every time step a noisy observation $o_i \in \mathbb{R}$ is received regarding the position of the object. Also at every time step, a variable b_i indicates whether the sensor is broken. A sensor fails with probability 0.2 and stays broken once it breaks. If the sensor is broken, the observation model is a simple uninformed triangular distribution, if it is not broken the observation model is an asymmetric triangular distribution that peaks at the true x position, but is biased to underestimate.

We show results for three queries $p(x_j|o_1)$ in the tracking model for predicting the state after j steps (up to 7) given an initial observation. Again, the results demonstrate the complex multi-modal nature of the distribution and the ability of the SVE inference algorithm to compute this in an exact closed-form. All of these queries completed in under 1 minute demonstrating that despite the complex symbolic manipulations involved, the SVE method should be reasonably scalable especially in graphical models with sparse structure like this sequential model.

Even though one can use other inference methods such as sampling to calculate the expectation, the interesting thing to observe here in Figure 3 and 5 is that using SVE we are able to derive the posterior model in a functional form. This is particularly helpful in cases where the objective is to evaluate the behavior of one variable with respect to another one when its value is not observed. With same analogy we are able to analyze the variable dependencies in a highly complex system with sophisticated distributions.

Related Work

Almost all prior work on continuous variable graphical models (with the exception of Kalman filtering (Welch and Bishop 1995) and other joint Gaussian models (Weiss and Freeman 1999)) has focused on *approximate* and/or *non-closed-form, exact* inference. Commonly used methods for continuous graphical models include discretization (which

[5]Our usage of the XADD differs from (Sanner, Delgado, and de Barros 2011) in that it was used there to represent *deterministic equations* over continuous variables, whereas here we use it to represent polynomial condition *probability densities*. (Sanner, Delgado, and de Barros 2011) did not discuss *definite integration* in XADDs, but the adaptation of this operation defined in Section from case statements to XADDs is straightforward.

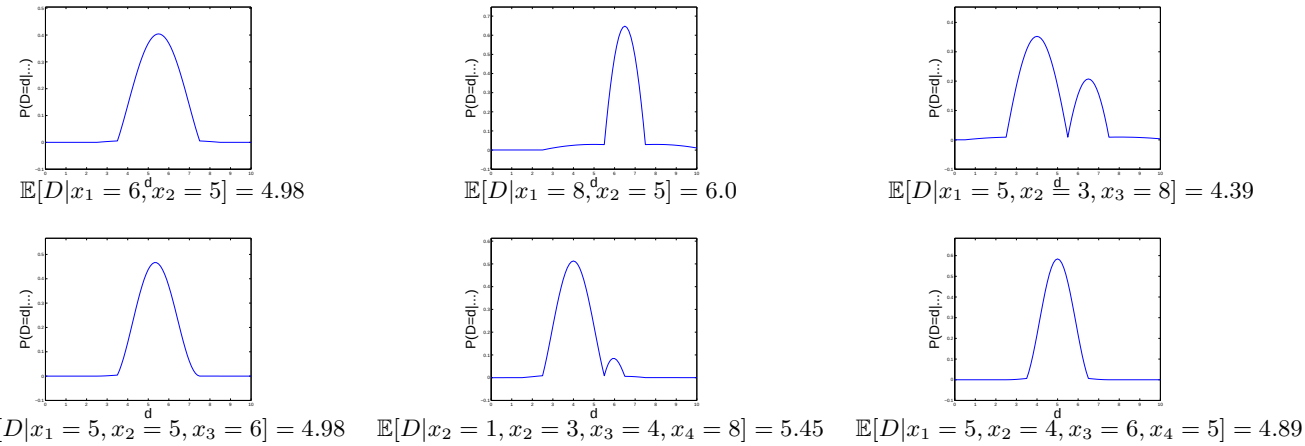

Figure 3: Queries for robot localization. The diagrams show $p(d|\mathbf{e})$ vs. d for the given evidence $\mathbf{e}$ shown below each diagram along with the exactly computed expectation $\mathbb{E}[D|\mathbf{e}]$ for this distribution.

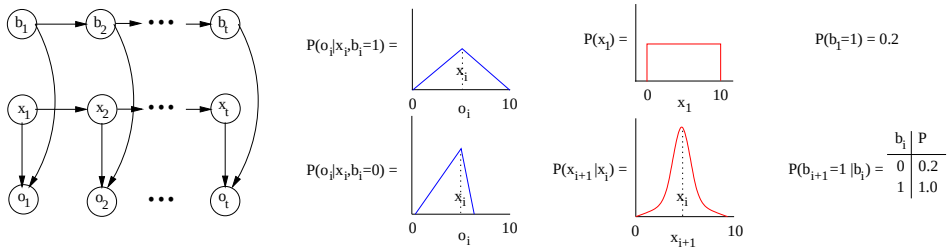

Figure 4: The tracking graphical model and all conditional probabilities.

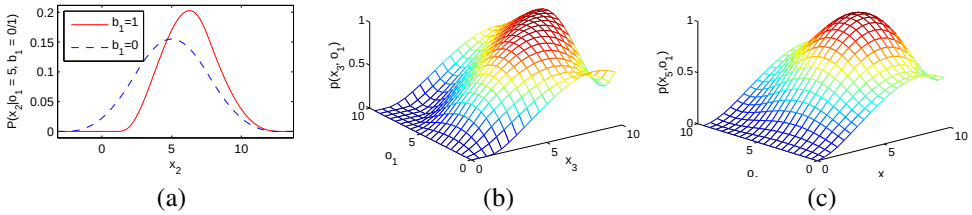

Figure 5: Plots of query and evidence variables for three queries in *tracking*: (a) $p(x_2|o_1 = 5, b_1 = 0)$ (blue dash) and $p(x_2|o_1 = 5, b_1 = 1)$ (red solid). (b) $p(x_3, o_1)$, (c) $p(x_5, o_1)$. o is on the lower left-edge axis, x is on the lower right, and the probability is the z axis.

is approximate) and numerical integration / (MCMC) sampling as exemplified in the BUGS tool (Lunn et al. 2000) (which provides an expressive language, but provides numerical results rather than a general functional form of a query result, e.g., as shown for an SVE query in Figure 5). In recent years, significant advances have been made using projected message passing algorithms such as variational inference (Jordan et al. 1999) and expectation propagation (Minka 2001) that can produce a functional form for a query result, albeit an approximation. However, it seems that little progress has been made in closed-form exact inference of results in a functional form for *expressive* non-Gaussian joint distributions — especially piecewise polynomial distributions with bounded support — which is the task addressed in this paper.

Conclusions and Future Work

We believe that SVE provides a significant step forward for exact, closed-form inference in discrete/continuous graphical models. Many real-world distributions have bounded support and can be arbitrarily approximated by polynomial functions, hence this work opens up the possibility of exact solutions (or arbitrary approximations thereof) in discrete/continuous graphical models for which previous exact, closed-form solutions were not available.

We crucially note that SVE is *not* limited to piecewise polynomial models, but that once more general expressions are introduced, we cannot guarantee that symbolic integrations will always be possible. Nonetheless, when all of the integrations in more general models can be computed symbolically (and much progress has been made in recent decades on automated symbolic integration), the SVE result will be *exact and closed-form*. Furthermore, if a certain function class is truly non-integrable, it may still be well-approximated by other integral function classes; this paves the way for the application of SVE to an extremely expressive range of discrete/continuous graphical models. Hence, we believe this paper is only the tip of the iceberg for SVE and its future applications.

References

Bahar, R. I.; Frohm, E.; Gaona, C.; Hachtel, G.; Macii, E.; Pardo, A.; and Somenzi, F. 1993. Algebraic Decision Diagrams and their applications. In *IEEE /ACM International Conference on CAD*.

Doucet, A.; De Freitas, N.; and Gordon, N., eds. 2001. *Sequential Monte Carlo methods in practice*. Springer.

Jordan, M. I.; Ghahramani, Z.; Jaakkola, T. S.; and Saul, L. K. 1999. An introduction to variational methods for graphical models. *Mach. Learn.* 37:183–233.

Kschischang, F. R.; Frey, B. J.; and Loeliger, H. A. 2001. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory* 47(2):498–519.

Lunn, D.; Thomas, A.; Best, N.; and Spiegelhalter, D. 2000. WinBUGS – a bayesian modelling framework: concepts, structure, and extensibility. *Statistics and Computing* 10:325–337.

Minka, T. 2001. Expectation propagation for approximate bayesian inference. In *Proceedings of UAI-01*, 362–36. San Francisco, CA: Morgan Kaufmann.

Sanner, S.; Delgado, K. V.; and de Barros, L. N. 2011. Symbolic dynamic programming for discrete and continuous state mdps. In *UAI-2011*.

Thrun, S.; Fox, D.; Burgard, W.; and Dellaert, F. 2000. Robust monte carlo localization for mobile robots. *Artificial Intelligence* 128(1-2):99–141.

Weiss, Y., and Freeman, W. T. 1999. Correctnes of belief propagation in gaussian graphical models of arbitrary topology. *Neural Computation* 13:2173–2200.

Welch, G., and Bishop, G. 1995. An introduction to the kalman filter. Technical report, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA.

Zhang, N. L., and Poole, D. 1996. Exploiting causal independence in bayesian network inference. *J. Artif. Intell. Res. (JAIR)* 5:301–328.