Probabilistic Inference for CART network

Christopher Meek
meek@microsoft.com

Bo Thiesson
thiesson@microsoft.com

April 2010

Technical Report
MSR-TR-2010-40

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
http://www.research.microsoft.com

1 Introduction

Graphical models are a widely used class of probabilistic models where independencies among variables are reflected in a graphical structure of the model. One of the key benefits of graphical models is the existence of inference algorithms that take advantage of independencies in the model in order to efficiently compute probabilities of interest. We consider the problem of inference in hybrid Bayesian networks—networks that contain both continuous and discrete variables.

Existing techniques for probabilistic inference in hybrid networks can be divided into methods which are exact (e.g., Lauritzen 1992; Lauritzen & Jensen 1999; Moral, Rumí, & Salmerón 2001; Cobb, Shenoy, & Rumí 2006) and methods which are approximate (e.g., Learner, Segal & Koller 2001; Rumí & Salmerón 2007). In this paper, we concentrate on exact probabilistic inference techniques.

Existing approaches to exact inference in hybrid networks are applicable to the class of conditional Gaussian (CG) distributions (Lauritzen & Wermuth, 1989) and require strong triangulation of the underlying Bayesian network. The key limitations of relying on conditional Gaussian (CG) distributions is that 1) no discrete variable can be a child of a continuous variable, 2) only linear relationships among continuous variables can be expressed, 3) continuous variables are restricted to Gaussians. In addition, the strong traingulation requirement can lead to intractabe inference problems even for simple networks such as a univariate binary switching state-space models (Lerner & Parr 2001).

In this paper, we consider probabilistic inference for an alternative family of distributions in which all of these limitations are removed but for which linear relationships among continuous variables cannot be parsimoniously expressed. This family contains the family of distributins defined by CART networks. A CART network is a Bayesian network in which the local distributions are CART models (Breiman, Friedman & Olshen, 1984), that is, each local distribution is either a classification or regression tree, depending on whether the dependent variable is discrete or continuous, respectively. This is a natural class of models to consider due to the fact that algorithms for learning Bayesian networks with local structure have been developed (Friedman & Goldszmidt 1996; Chickering, Heckerman & Meek 1997) and have been shown to yield significant improvements in modeling joint distributions.

In addition, there has been a variety of work on reducing the cost of inference in discrete Bayesian networks that aim to take advantage of contextual independencies in decision trees (e.g., Zhang & Poole 1996; Boutilier, Friedman, Goldszmidt & Koller 1996; Tung 2002). Our approach is most similar to the confactors of Poole and Zhang (2003) who represent a Bayesian network distribution by a set of multiplicative contextual factors for discrete variables. Poole and Zhang (2003) provide examples that demonstrate that one can improve computational efficiency by allowing for non-proper contextual factors and lazily multiplying confactors. Our work draws on this body of work, especially Poole and Zhang's lazy approach to multiplication. Our algorithm is a junction tree algorithm that allows one to efficiently compute all marginals si-

multaneously and take advantage of context specific independencies. Our work advances previous work in developing an exact inference method that handles models containing both continuous and discrete variables.

To accomplish the task of inference for CART networks we introduce region partitioned (RP) potentials and demonstrate how one can perform the marginalization and multiplication operations that are the basis of sum-product inference algorithms. We analyze the complexity of inference by providing bounds on the complexity of these operations during the course of inference. In the paper, we focus on computation of marginal probabilities for variables using a sum-product algorithm but we note that one could also use a max-product algorithm to do maximum *a posteriori* (MAP) inference.

The paper is organized as follows. In Section 2, we describe CART networks. Section 3 defines the RP potentials and the marginalization and multiplication operations used for inference. In Section 4, we describe a junction tree inference procedure for RP potentials and analyze the computational cost for the application of the technique to CART networks. Finally, in Section 5, we conclude with a discussion of CART networks, RP potentials and potential avenues for future research.

2 CART Networks

Let $\mathbf{X} = X_V = (X_v)_{v \in V}$ be a set of discrete and continuous random variables, and let $\mathcal{D}_V$ denote the domain of all possible values $x_V \in X_V$. A Bayesian Network for the variables X_V is a probabilistic graphical model defining a joint distribution $p(X_V)$. A directed acyclic graphical (DAG) structure defines conditional-independence assertions in the model, which imply a factorization

$$p(\mathbf{X}) = \prod_{v \in V} p(X_v | X_{pa(v)}) \tag{1}$$

into *local models* $p(X_v|X_{pa(v)})$, where $X_{pa(v)}$ is the set of parents for X_v in the DAG structure.

In a CART network, we will associate a decision (classification or regression) tree T_v with each local model in the Bayesian network. Splits in the tree are determined by split decisions on variables in the set of parents $X_{pa(v)}$, and leafs in the tree will hold marginal distributions for X_v. That is,

$$p(X_v | X_{pa(v)}) = \prod_{l_v \in \mathcal{L}_v} \left(f^{l_v}(X_v) \right)^{\chi^{l_v}(X_{pa(v)})} \tag{2}$$

where $f^{l_v}(X_v)$ is the distribution for X_v at leaf l_v in the tree T_v with leafs $\mathcal{L}_v$, and $\chi^{l_v}(X_{pa(v)})$ is an indicator function that associate an observation $x_{pa(v)}$ for $X_{pa(v)}$ with a particular leaf in the tree.

Here, the association of an observation with a leaf in the tree is determined by split decisions that evaluates the outcome according to a predicate on a single variable $X_j \in X_{pa(v)}$ at each internal node on the path from the root to the

leaf. The indicator is one if the conjunction of all split decisions on the path evaluate true, and zero otherwise. Let us define the marginal indicator function $\chi^{l_v}(X_a)$ for variables $X_a \subseteq X_{pa(v)}$ as the indicator that ignores all split decisions involving $X_{pa(v)} \setminus X_a$, and returns one if all split decisions on the path involving variables in X_a evaluate true, and zero otherwise. Thus, the indicator function that associates an observation with a leaf factorizes according to the marginal indicator functions

$$\chi^{l_v}(X_{pa(v)}) = \prod_{X_j \in X_{pa(v)}} \chi^{l_v}(X_j) \tag{3}$$

3 RP Potentials

The basic computational object in our inference framework is that of a *region partitioned* (RP) potential. In this section, we define the potential together with basic operations that allow us to utilize basic inference frameworks that exploit local structure in a network.

3.1 Definition

Let us consider a set of variables $X_U = X_\Gamma \cup X_\Delta$ of both continuous (X_Γ) and discrete (X_Δ) type, and let $\mathbb{R}_\Gamma$ denote the $|\Gamma|$-dimensional space of real numbers for the continuous variables and $\mathcal{I}_\Delta$ denote the Cartesian product of state spaces for the discrete variables. With $\mathcal{D}_U = \mathbb{R}_\Gamma \times \mathcal{I}_\Delta$, we define a *region* $r_U \subseteq \mathcal{D}_U$ as the product space of regions on the individual variables $X_v \in X_U$; for a continuous variable a region is an interval, and for a discrete variable a region is a subset of possible values for that variable.

A *region-partionend potential* (or RP potential) $\phi_U : \mathcal{D}_U \to \mathbb{R}^{\geq 0}$ is defined by a collection of an arbitrary number of *indicator-potential pairs* (a^i, b^i) with the property

$$\phi_U = \sum_i a_U^i b_U^i. \tag{4}$$

Here, $a_U : \mathcal{D}_U \to \{0, 1\}$ is a *region indicator* factoring into a product

$$a_U = \chi(X_U \in r_U) = \prod_{v \in U} \chi(X_v \in r_v), \tag{5}$$

where each indicator $\chi(X_v \in r_v)$ evaluates the outcome according to a region r_v on a single variable $X_v \in X_U$. $b_U : \mathcal{D}_U \to \mathbb{R}^{\geq 0}$ is a *region potential*, defined as a scaled joint distribution factoring into a product of integrable distributions on individual variables $X_v \in U$

$$b_U = w f(X_U) = w \prod_{v \in U} f(X_v). \tag{6}$$

We define a set of *effective variables* for a region indicator a_U (region potential b_U) as *eff*$(a_U) = X_W$ (*eff*$(b_U) = X_W$), where $X_W \subseteq X_U$ is a minimal

set of variables such that the region indicator (region potential) function does not depend on the values of $X_{U\setminus W}$. We say that a potential ϕ_U is a *defining potential* for a variable X_v if $\phi_U(X_{U\setminus v} = x_{U\setminus v})$—that is the functional obtained by choosing particular values for $X_{U\setminus v}$ from $\mathcal{D}_{U\setminus v}$—defines an integrable distribution. Finally, a set of RP potentials is *univalent* for a set of variables X_U if each variable in X_U has a unique defining potential.

In order to represent a CART model in an RP potential we define an indicator-potential pair (a, b) for each leaf in the tree. For the i^{th} leaf we define (a^i, b^i) where a^i is given by the product of indicators along the path to the leaf (the right-hand side of Equation 3) and b^i is the marginal distribution in the leaf $f^{l_v}(X_v)$ in Equation 2. Notice that $\mathit{eff}(a^i)$ (the set of parents) and $\mathit{eff}(b^i)$ (the target) are disjoint subsets of the variables, which is a simplification that will not hold for all of the RP potentials we consider in the following.

Example 1: The three decision trees shown in Figure 1 can be represented by the following three RP potentials

$$\phi_A = \left[\begin{array}{c} \big(\chi(X_A \in \mathcal{D}_A), f^1(X_A)\big) \end{array} \right]$$

$$\phi_{AB} = \left[\begin{array}{c} \big(\chi(X_A < 0), f^1(X_B)\big) \\ \big(\chi(0 \leq X_A < 2), f^2(X_B)\big) \\ \big(\chi(X_A \geq 2), f^3(X_B)\big) \end{array} \right]$$

$$\phi_{ABC} = \left[\begin{array}{c} \big(\chi(X_A < 3)\chi(X_B = t), f^1(X_C)\big) \\ \big(\chi(X_A < 0)\chi(X_B = f), f^2(X_C)\big) \\ \big(\chi(0 \leq X_A < 3)\chi(X_B = f), f^3(X_C)\big) \\ \big(\chi(X_A \geq 3), f^4(X_C)\big) \end{array} \right]$$

□

3.2 Basic Operations

We will now define a set of basic operations that will allow us to perform inference using RP potentials and we will discuss computational complexity for these operations.

3.2.1 Multiplication

Consider the two RP potentials $\phi_U = \sum_i a_U^i b_U^i$ and $\phi_W = \sum_j a_W^j b_W^j$. Multiplication is defined in the obvious way

$$\begin{aligned} \phi_U * \phi_W &= \left(\sum_i a_U^i b_U^i \right) * \left(\sum_j a_W^j b_W^j \right) \\ &= \sum_k a_{U\cup W}^k b_{U\cup W}^k \\ &= \phi_{U\cup W}, \end{aligned}$$

where $a^k_{U\cup W} = a^i_U a^j_W$ and $b^k_{U\cup W} = b^i_U b^j_W$.

Note that if the pair of potentials is univalent for a set of variables then the product is univalent for that set of variables; that is the resulting potential is a defining potential for each of the variable in the set. While univalency is not strictly required, as we shall see, it simplifies the inference procedure.

Example 1 (continued): Continuing from above, we can multiply the tree potentials to create the a new RP-potential

$\phi'_{ABC} = \phi_A * \phi_{AB} * \phi_{ABC} =$

$$\begin{bmatrix} \big(\chi(X_A<0)\chi(X_B=t), f^1(X_A)f^1(X_B)f^1(X_C)\big) \\ \big(\chi(X_A<0)\chi(X_B=f), f^1(X_A)f^1(X_B)f^2(X_C)\big) \\ \big(\chi(0\le X_A<2)\chi(X_B=t), f^1(X_A)f^2(X_B)f^1(X_C)\big) \\ \big(\chi(0\le X_A<2)\chi(X_B=f), f^1(X_A)f^2(X_B)f^3(X_C)\big) \\ \big(\chi(2\le X_A<3)\chi(X_B=t), f^1(X_A)f^3(X_B)f^1(X_C)\big) \\ \big(\chi(2\le X_A<3)\chi(X_B=f), f^1(X_A)f^3(X_B)f^3(X_C)\big) \\ \big(\chi(X_A\ge 3), f^1(X_A)f^3(X_B)f^4(X_C)\big) \end{bmatrix} \tag{7}$$

where we have removed (a, b)-pairs with inconsistent indicators. For example, the (a, b)-pair with $a = \chi(X_A < 0)\chi(X_A > 3)$ is removed. Notice that for the resulting RP potential ϕ'_{ABC}, the effective region indicators and the effective region potentials are defined on overlapping sets of variables. Also, notice that both the initial and resulting potentials are univalent. □

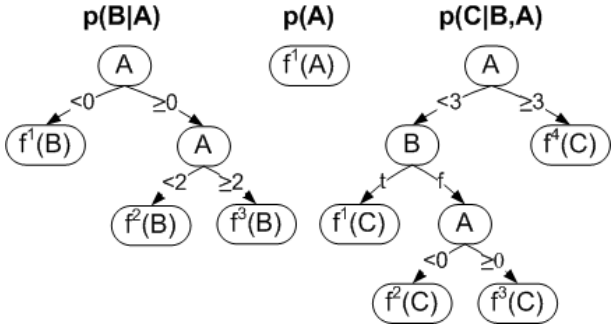

Figure 1: Decision trees for distributions $p(X_A)$, $p(X_b|X_A)$, and $p(X_c|X_b, X_a)$. (Notice that variables are only represented by their subscripts to make the figure easier to read.)

In practice, being lazy about combining potentials can be beneficial to efficiency (e.g., Madsen & Jensen 1999; Poole & Zhang 2003). In our approach, we take advantage of being lazy during multiplication. A *lazy RP potential* is a set of (non-lazy) RP potentials that will be combined by multiplication. In order to describe when we are lazy during computation, we distinguish between "lazy" and "non-lazy" RP potentials. We use ψ's for lazy RP potentials and ϕ's

for (non-lazy) RP potentials. We call the process of converting from a lazy to a (non-lazy) RP potential *expansion*. Note that we will generally expand lazy RP potentials when it becomes necessary to do so during marginalization.

Example 1 (continued): Continuing the example from above, the lazy RP potential $\psi'_{ABC} = \{\phi_A, \phi_{AB}, \phi_{ABC}\}$ is simply the set of the three individual potentials, which after expansion turns into its non-lazy counterpart ϕ'_{ABC}, shown in Equation 7. □

In order to efficiently compute the result of expansion (non-lazy multiplication) for two RP potentials it is necessary to do some precomputation. For a RP potential $\phi_U = (a_U^i, b_U^i)_i$ we term the collection of region indicators by $A_U = (a_U^i)_i$ a *region set*. Let $r(a_U) = \{X_U \in \mathcal{D}_U | a_U = 1\}$ denote the region specified by a particular region indicator $a_U \in A_U$. A region set A_U is a *refinement* of another region set D_U, denoted $A_U \preceq D_U$, if for every $a_U^i \in A_U$ there exist a $d_U^j \in D_U$ such that $r(a_U^i) \subseteq r(d_U^j)$. Consider two region sets D_U and E_U over the same domain. It is always possible to create a partition function A_U with the property that $A_U \preceq D_U$ and $A_U \preceq E_U$. In particular, Algorithm 1 computes one such refined partition function by iterating over all region indicators in the two input partition functions. The outcome of applying Algorithm 1 can be thought of as a simple overlay of regions for the two involved partition functions, as illustrated in Example 1. The algorithm also computes a mapping from elements of the new region set to the index of the elements of the input regions sets. This is important for efficient implementation of expansion.

Example 2: Consider the two region sets

$$D_{\{XY\}} = \begin{bmatrix} \chi(X < 2)\chi(Y \geq 3) \\ \chi(X \geq 2)\chi(Y \geq 2) \\ \chi(X < 2)\chi(Y < 3) \\ \chi(X \geq 2)\chi(Y < 2) \end{bmatrix}$$

$$E_{\{XY\}} = \begin{bmatrix} \chi(X < 1)\chi(Y \geq 1) \\ \chi(X \geq 1)\chi(Y \geq 1) \\ \chi(X < 3)\chi(Y < 1) \\ \chi(X \geq 3)\chi(Y < 1) \end{bmatrix}$$

each with four regions as illustrated in Figure 2. Simply overlaying the two visualizations of regions for $D_{\{XY\}}$ and $E_{\{XY\}}$ results in a visualization of regions for the resulting partition function $A_{\{XY\}}$, as created by Algorithm 1. The grey regions illustrate the mapping between an element in the new region set and elements in the old region sets. □

The cost of computing an expansion, assuming that Algorithm 1 is $\mathcal{O}(1)$, is $\mathcal{O}(|\phi_U|)$ where $|\phi_U|$ is the number of region-indicator pairs in the RP potential. As discussed below, we can precompute the region set A_U and the mappings g, h during initialization so it is reasonable to treat the cost of Algorithm 1 as constant in practice. Note that Algorithm 1 assumes that the input region

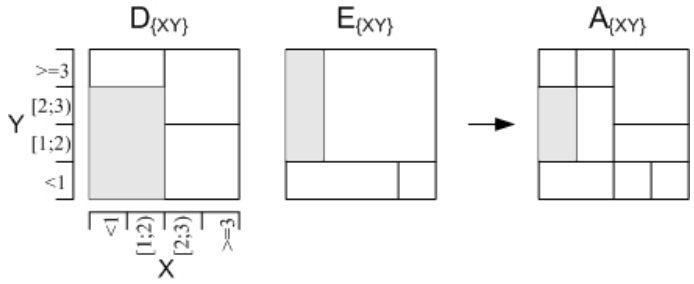

Figure 2: Refinement of two region sets $D_{\{XY\}}$ and $E_{\{XY\}}$ into $A_{\{XY\}}$.

Algorithm 1 Intersection of two region sets D_U and E_U, creating a refined region set A_U and two mapping functions. This algorithm assumes each input region set contains mutually exclusive regions

Input: D_U, E_U
Output: A_U, $g()$, $h()$
$A_U = \{\}$
for $d_U^i \in D_U$ **do**
  for $e_U^j \in E_U$ **do**
    if $r(d_U^i) \cap r(e_U^j) \neq \emptyset$ **then**
      $A_U = A_U \cup \{d_U^i e_U^j\}$
      set $g(d_U^i e_U^j) = i$
      set $h(d_U^i e_U^j) = j$
    end if
  end for
end for

sets contain mutually exlusive regions. This is the case for CART networks. If, however, this is not the case, Algorithm 1 can be used to obtain a refined region set that is exclusive; if A is non-exclusive then the region set returned by Algorithm1(A,A) is exclusive.

3.2.2 Marginalization

Consider the RP potential ϕ_U and its lazy counterpart $\psi_U = (\phi^j_{W^j})_j$. For $W \subseteq U$ we let $\phi_{U\downarrow W}$ (or $\psi_{U\downarrow W}$) denote marginalization of ϕ (or ψ) to X_W. We perform marginalization one variable at a time.

We define marginalization for lazy RP potentials in terms of marginalization of (non-lazy) RP potentials. We will therefore have to expand a lazy RP potential prior to marginalization. The procedure for preparing the lazy potential ψ_U for marginalizing out X_v is as follows: 1) create two lazy-potentials ψ^v_U, those in which X_v is an effective variable (in either the indicator or potential) and $\psi^{\hat{v}}_U$, those in which X_v is not an effective variable; 2) expand ψ^v_U into ϕ^v_U. Due

Algorithm 2 Multiplying two RP potentials utilizing precomputed mapping functions.

Input: ϕ_1, ϕ_2
Output: ϕ_U
Let A_1, A_2 be the region sets of ϕ_1, ϕ_2
$(A_U, g(), h()) =$ Algorithm1(A_1, A_2)
for $a^i \in A_U$ **do**
  $b^i = b_1^{g(a^i)} * b_2^{h(a^i)}$
end for

to the distributive law we have

$$\psi_{U\downarrow U\setminus v} = \psi_U^{\hat{v}} \phi_{U\downarrow U\setminus v}^{v}.$$

The computational cost of preparing the RP potential is a function of the number of potentials requiring expansion. Let $|K|$ denote the number of potentials in ψ_U^v. The overall complexity for the expansion operation is therefore the sum of the $|K| - 1$ expansions with complexity, as discussed in Section 3.2.1.

Marginalization of an RP potential

We now consider marginalizing a variable X_v out of a (non-lazy) RP potential ϕ_U. We restrict attention to marginalization of defining potentials for X_v in order to avoid integration issues for continuous variables. In particular, we want to avoid imposing arbitrary results when integrating with respect to a uniform distribution over unbounded intervals. We will see in Section 4.2 that this condition is always satisfied during the course of inference in CART networks.

For notational convenience we will abuse notation and use $\int_{X_v \in r_v \subseteq \mathcal{D}_v} \phi_U$ to denote marginalization for both discrete and continuous variables over some region. Using the definition from (4)

$$\phi_{U\downarrow U\setminus v} = \int_{X_v \in \mathcal{D}_v} \phi_U = \sum_i \int_{X_v \in \mathcal{D}_v} a_U^i b_U^i,$$

we see that each indicator-potential pair in the RP potential can be marginalized independently.

By using the factorizations in (5) and (6) we can derive the following computationally simple expression for the marginalization of each individual indicator-potential pair (a_U, b_U).

$$\begin{aligned} &\int_{X_v \in \mathcal{D}_v} a_U b_U \\ &= \int_{X_v \in \mathcal{D}_v} \prod_{u \in U} \chi(X_u \in r_u) * w \prod_{u \in U} f(X_u) \\ &= w \prod_{u \in U\setminus v} \chi(X_u \in r_u) \prod_{u \in U\setminus u} f(X_u) \int_{X_v \in r_v} f(X_v) \end{aligned} \tag{8}$$

where we in the last equation have used the equality

$$\int_{X_v \in \mathcal{D}_v} \chi(X_v \in r_v) f(X_v) = \int_{X_v \in r_v} f(X_v).$$

Note that if there is no indicator function for X_v then we integrate over the full domain $\mathcal{D}_V$. Clearly, RP potentials are closed under marginalization since the integral yields a scalar that can be combined with w resulting in a valid functional form for an RP potential.

Due to the fact that the potential is defining for X_v, the integral on the right-hand side of (8) is just a simple univariate integration (or summation) over a region, which can be easily computed (or looked up) for many standard distributions. The integral is particularly simple if $r_v = \mathcal{D}_v$, in which case it reduces to 1. Let $|I_v|$ denote the computational cost of integrating a single variable X_v over the region indicator r_v. For a continuous variable, this integration is typically found via two computations (or lookups) for the cumulative function with respect to the upper and lower bound for the region, respectively. For a discrete variable, $|I_v|$ is just the sum of states in the region. The integration is performed for each of the $|\phi_U|$ indicator-potential terms in ϕ_U. The overall computational complexity for the actual marginalization operation is therefore $\mathcal{O}(|I_v||\phi_U|)$.

Term-reduction

The result of marginalization described above contain as many indicator-potential pairs as the original ϕ_U. However, as the reduction in effective variables suggests, it is often the case that some pairs can be combined. In particular, if two indicator-potential pairs $(a_1, b_1), (a_2, b_2)$ are such that $a_1 = a_2$ and b_1 and b_2 only differ in their respective weight factors w_1 and w_2 then the two indicator-potential pairs can be replaced by a single pair $(a_1, (1+\frac{w_2}{w_1}) * b_1)$. The complexity of this operation is $\mathcal{O}(|\phi_U|)$. This can be accomplished by hashing into buckets of similar pairs while adjusting the weight of the representative potential in each bucket.

Example 3: Consider the lazy RP potential $\psi_{ABCD} = \{\phi_A, \phi_{AB}, \phi_{ABC}, \phi_{ACD}\}$, which in addition to the three potentials from Example 1 also contains the potential

$$\phi_{ACD} = \begin{bmatrix} \big(\chi(X_C < 0), f^1(X_D)\big) \\ \big(\chi(X_A < 1)\chi(X_C \geq 0), f^2(X_D)\big) \\ \big(\chi(X_A \geq 1\chi(X_C \geq 0), f^3(X_D)\big) \end{bmatrix}$$

For the marginalization $\psi_{ABCD\downarrow AB}$ let us first marginalize with respect to X_D. ϕ_{CD} is the only potential in which X_D is effective, so $\psi^{\hat{D}}_{ABCD} = \{\phi_A, \phi_{AB}, \phi_{ABC}\}$ and $\psi^{D}_{ABCD} = \phi_{ACD}$ (no expansion is necessary). Integrating ϕ_{ACD} with respect to X_D gives us

$$\phi_{ACD\downarrow AC} = \begin{bmatrix} (\chi(X_C < 0), 1) \\ (\chi(X_A < 1)\chi(X_C \geq 0), 1) \\ (\chi(X_A \geq 1\chi(X_C \geq 0), 1) \end{bmatrix}$$

For the further marginalizing of $\psi_{ABCD\downarrow ABC} = \{\phi_A, \phi_{AB}, \phi_{ABC}, \phi_{ACD\downarrow AC}\}$ with respect to X_C we have $\psi^{\hat{C}}_{ABCD\downarrow ABC} = \{\phi_A, \phi_{AB}\}$ and $\psi^{C}_{ABCD\downarrow ABC} = \{\phi_{ABC}, \phi_{ACD\downarrow AC}\}$. Expanding $\psi^{C}_{ABCD\downarrow ABC}$ leads to

$$\phi^{C}_{ABCD\downarrow ABC} = \begin{bmatrix} \big(\chi(X_A < 3)\chi(X_B = t)\chi(X_C < 0), f^1(X_C)\big) \\ \big(\chi(X_A < 0)\chi(X_B = f)\chi(X_C < 0), f^2(X_C)\big) \\ \big(\chi(0 \le X_A < 3)\chi(X_B = f)\chi(X_C < 0), f^3(X_C)\big) \\ \big(\chi(X_A \ge 3)\chi(X_C < 0), f^4(X_C)\big) \\ \big(\chi(X_A < 1)\chi(X_B = t)\chi(X_C \ge 0), f^1(X_C)\big) \\ \big(\chi(X_A < 0)\chi(X_B = f)\chi(X_C \ge 0), f^2(X_C)\big) \\ \big(\chi(0 \le X_A < 1)\chi(X_B = f)\chi(X_C \ge 0), f^3(X_C)\big) \\ \big(\chi(1 \le X_A < 3)\chi(X_B = t)\chi(X_C \ge 0), f^1(X_C)\big) \\ \big(\chi(1 \le X_A < 3)\chi(X_B = f)\chi(X_C \ge 0), f^3(X_C)\big) \\ \big(\chi(X_A \ge 3)\chi(X_C \ge 0), f^4(X_C)\big) \end{bmatrix}$$

By integrating this potential with respect to X_C we get (after term-reduction)

$$\phi^{C}_{ABCD\downarrow AB} = \begin{bmatrix} \big(\chi(X_A < 3)\chi(X_B = t), w^{1,<}\big) \\ \big(\chi(X_A < 0)\chi(X_B = f), w^{2,<} + w^{2,\ge}\big) \\ \big(\chi(0 \le X_A < 3)\chi(X_B = f), w^{3,<}\big) \\ \big(\chi(X_A \ge 3), w^{4,<} + w^{4,\ge}\big) \\ \big(\chi(X_A < 1)\chi(X_B = t), w^{1,\ge}\big) \\ \big(\chi(0 \le X_A < 1)\chi(X_B = f), w^{3,\ge}\big) \\ \big(\chi(1 \le X_A < 3)\chi(X_B = t), w^{1,\ge}\big) \\ \big(\chi(1 \le X_A < 3)\chi(X_B = f), w^{3,\ge}\big) \end{bmatrix}$$

where $w^{i,<}$ is the integration of $f^i(X_C)$ over the region $X_C < 0$ and $w^{i,\ge}$ is the integration of $f^i(X_C)$ over the region $X_C \ge 0$.

We finally have the marginalization $\psi_{ABCD\downarrow AB} = (\phi_A, \phi_{AB}, \phi^{C}_{ABCD\downarrow AB})$. □

4 Junction Tree Inference

In this section, we show how to use the (lazy) RP potentials and their associated multiplication and marginalization operations to perform probabilistic inference for CART networks, that is, compute the probability of a variable given the values for a subset of the remaining variables. In that respect, we define inference within the standard framework of a sum-product algorithm for inference inspired by Shenoy & Shafer (1990), and analyze the algorithm to provide computational bounds for the inference procedure.

Associated with a CART network is a graph. In order to perform inference, we triangulate the graph and form a junction tree. We denote a (maximal) clique in the junction tree by $C_i \in \mathcal{C}$ where $C_i \subseteq X_V$. We let N_i denote the set of indices for cliques that are adjacent to C_i in the junction tree, and we denote the separators between adjacent cliques $S_{i,j} = C_i \cap C_j$. The junction tree associates a lazy RP potential with every clique in the tree, and probabilistic inference is performed by passing messages between these clique potentials.

4.1 Initialization and Evidence

In this section we describe the process of initialization (representing the joint distribution of the CART network in the junction tree) and inserting evidence (representing the conditional distribution given values for a subset of variables).

In order to represent the joint distribution of the CART network with the junction tree we associate every local distribution in the CART network with a single clique in the junction tree. For each variable X_v, we transform the associated local model (2) in the CART network factorization (1) into a potential $\phi_{v \cup pa(v)}$, as discussed in detail in Section 3.1. Next we associate each of the local clique potentials $\phi_{v \cup pa(v)}$ with a clique in the junction tree. In order to accommodate the local distribution we require that $X_{v \cup pa(v)} \subseteq C_i$. Finally, we set the clique potential for each clique to be the lazy product of all the local CART potentials assigned to that clique. Notice that every clique is guarenteed to have at least one CART network assigned to it, because any clique in a junction tree has at least one variable that is unique to it.

At this point, if we take the product of the lazy RP potentials in the the cliques of the junction tree, we have

$$p(\mathbf{X}) = \prod_{C_i \in \mathcal{C}} \psi_{C_i}, \tag{9}$$

that is, after initialization we have represented the joint distribution for the CART network (1) in the junction tree.

We insert evidence $X_v = x_v$ into all clique potentials ψ_{C_i}, where $X_v \in C_i$. Let $\phi = \{a_i, b_i\}_i$ be any of the RP potentials in the lazy representation for one of these ψ_{C_i}. We first reduce the number of indicator-potential pairs in ϕ by removing any pair, where $\chi(X_v = x_v)a^i = 0$. Second, we simplify all b^i with $X_v \in \mathit{eff}(b^i)$ by reducing the set of effective variables with the update $b^i \leftarrow f(X_v = x_v)b^i / f(X_v)$, where $f(X_v = x_v)$ is a delta function when X_v is continuous and an indicator function when X_v is discrete.

Let $\mathbf{E} = \mathbf{e}$ denote all of the evidence that has been inserted into the clique potentials. If we at this point take the product of the clique potentials in the junction tree, we have

$$p(\mathbf{X}|\mathbf{E} = \mathbf{e}) = \prod_{C_i \in \mathcal{C}} \psi^*_{C_i}, \tag{10}$$

that is, after inserting evidence, the junction tree will represent the joint distribution given this evidence. Note that we use ψ^* to indicate the clique potentials after inserting evidence.

4.2 Message Passing

The marginal probabilities for any variable in the CART network given a set of evidence can be computed by marginalizing out all remaining variables from the joint posterior distribution in (10). However, the cost of this marginalization can be prohibitive if the size of the potential obtained by taking the product

of all of the clique potentials in the junction tree is large. The goal of utilizing the junction tree is to reduce the computational cost associated with computing marginals probabilities. The sum-product inference algorithm for a junction tree is defined in terms of a set of messages $m_{i\to j}(S_{i,j})$ between adjacent cliques C_i, C_j in the junction tree. The message between C_i and C_j is defined as

$$m_{i\to j}(S_{i,j}) = \int_{C_i \setminus S_{i,j}} \psi^*_{C_i} \prod_{n \in N_i \setminus \{j\}} m_{n\to i}(S_{n,i}). \tag{11}$$

To compute a marginal for a variable X_v we identify a clique C_i containing the variable and compute

$$p(X_v) \propto \int_{C_i \setminus X_v} \psi^*_{C_i} \prod_{n \in N_i} m_{n\to i}(S_{n,i}). \tag{12}$$

The standard algorithm computes each message $m_{i\to j}$ via (11), whenever the messages on the right-hand side of the equation have already been computed. It can be shown that this algorithm computes $2\left(|\mathcal{C}| - 1\right)$ messages; two messages associated with every pair of adjacent cliques in the junction tree.

Proposition 1 *If the collection of RP potentials associated with the junction tree JT is univalent then we can apply the RP potential junction tree inference procedure to JT.*

Proof: In order for the procedure to be correct every message required for inference must be well-defined. Marginalization of X_v for a potential ψ_U, as we have defined it, is only well-defined if ψ_U is a defining potential for X_v. Consider a message $m_{i\to j}$ for which the computation includes the marginalization of X_v. Since the collection of RP potentials in the junction tree is univalent there is a potential ψ_k that is defining for X_v. Each of the messages that depends on ψ_k must be a defining potential. Because the junction tree is a tree, at most one of the messages coming into $\mathcal{C}_i$ can be a defining potential for X_v. If none of the incoming messages are defining, then the local potential must be defining for X_v. Thus the set of potentials used to create the message is univalent and thus we marginalize X_v from a potential that is defining for X_v when creating the message. □

Note that the collection of potentials associated with a collection of CART models is univalent and thus we can apply the junction tree algorithm to perform inference in CART networks.

4.3 Computational Complexity

We consider the cost of our inference method using a non-lazy multiplication. We do this because quantifying the benefits of being lazy is analytically difficult.

In order to efficiently compute the product of two RP potentials we need to do some precomputation. In particular for each message, we precompute

the region set of the result of the multiplication and mappings from each of elements of the resulting region set to the index of the region-indicator pair of the input RP potential. This is described for the multiplication for two potentials in Algorithm 1 but can be generalized to the product of any number of RP potentials. It is possible to do this precomputation at the time of initialization for a junction tree because the structure of the junction tree and the potentials are known prior to the incorporation of evidence.

The cost of performing inference after initialization is a function of (1) incorporating evidence and (2) the cost of computing messages.

First, the cost of incorporating evidence is at most $\mathcal{O}(|\mathbf{e}||\mathcal{C}|)$ where $|\mathbf{e}|$ is the number of observed variables and $|\mathcal{C}|$ is the sum of the sizes of all of the RP potentials in the clique tree.

Second, the cost of computing messages is the cost of multiplying the incoming messages with the local potential and then marginalizing the result. As discussed in Sections 3.2.1 and 3.2.2, the cost of the multiplication (marginalization) procedures is a function of the size of the potential ϕ_U created by (used as input to) the procedures. The size of this potential is no larger than the naive tree width of the graphical model in which each continuous variable is discretized according to the finest set of intervals needed to respect any indicator function in any RP potential.

5 Conclusion and Further Work

In this paper, we have described a family of models, CART networks, that allow for rich dependencies among discrete and continuous variables. In order to develop an inference algorithm for CART networks, we introduced the RP potential and described the marginalization and multiplication operations for RP potentials. In addition, we analyzed the computational cost of a sum-product junction tree algorithm for CART networks.

There are a number of potential extensions to this work. First, while our CART models do relax several limitations of conditional Gaussian models, the regression trees do not allow us to model local linear dependencies among continuous variables. It would be useful to explore the possibility of doing inference when extending the regression trees to allow the regression for the target variable to depend on the value (as opposed to region) of its continuous parents. In that respect, the work on mixtures of truncated exponentials (MTEs) offers exact inference for a different, but more flexible, family of distributions than our CART networks. (See, e.g., Moral, Rumí, & Salmerón, 2001; Cobb, Shenoy, & Rumí, 2006). The MTEs allow dependencies among continuous variables, and can be used to approximate CART networks at some additional computational cost. It would be interesting to quantify both the approximation error and the additional computational cost. Second, it would be useful to investigate the computational complexity of MAP inference for CART networks. Finally, it is interesting to note that RP potentials can represent kernel density models. This opens the possibility of handling missing data in those models and might allow

one to develop interesting generalizations.

References

Boutilier, C., Friedman, N., Goldszmidt, M., & Koller, D. (1996). Context-specific independence in Bayesian networks. *Uncertainty in Artificial Intelligence* (pp. 115–123).

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Belmont, California: Wadsworth International Group.

Chickering, D., Heckerman, D., & Meek, C. (1997). A Bayesian approach to learning Bayesian networks with local structure. *Uncertainty in Artificial Intelligence* (pp. 80–89). Morgan Kaufman.

Cobb, B. R., Shenoy, P. P., & Rumí, R. (2006). Approximating probability density functions in hybrid bayesian networks with mixtures of truncated exponentials. *Statistics and Computing*, *16*, 293–308.

Friedman, N., & Goldszmidt, M. (1996). Learning Bayesian networks with local structure. *Learning in Graphical Models* (pp. 421–459). MIT Press.

Lauritzen, S., & Wermuth, N. (1989). Graphical models for associations between variables, some of which are qualitative and some quantitative. *Annals of Statistics*, *17*, 31–57.

Lauritzen, S. L. (1992). Propagation of probabilities, means and variances in mixed graphical association models. *Journal of the American Statistical Association*, *87*, 1098–1108.

Lauritzen, S. L., & Jensen, F. (1999). Stable local computation with conditional Gaussian distributions. *Statistics and Computing*, *11*, 191–203.

Lerner, U., & Parr, R. (2001). Inference in hybrid networks: Theoretical limits and practical algorithms. *Uncertainty in Artificial Intelligence* (pp. 310–318).

Lerner, U., Segal, E., & Koller, D. (2001). Exact inference in networks with discrete children of continuous parents. *Uncertainty in Artificial Intelligence* (pp. 319–328).

Madsen, A. L., & Jensen, F. V. (1999). LAZY propagation: A junction tree inference algorithm based on lazy evaluation. *Artificial Intelligence*, *113*, 203–245.

Moral, S., Rumí, R., & Salmerón, A. (2001). Mixtures of truncated exponentials in hybrid bayesian networks. *ECSQARU* (pp. 156–167).

Poole, D., & Zhang, N. L. (2003). Exploiting contextual independence in probabilistic inference. *Journal of Artificial Intelligence Research*, *18*, 2003.

Rumí, R., & Salmerón, A. (2007). Approximate probability propagation with mixtures of truncated exponentials. *Int. J. Approx. Reasoning*, *45*, 191–210.

Shenoy, P. P., & Shafer, G. (1990). Axioms for probability and belief-function propagation. *Uncertainty in Artificial Intelligence* (pp. 169–198). Morgan Kaufman.

Tung, L. (2002). A clique tree algorithm exploiting context-specific independence. Master's thesis, Department of Computer Science, University of British Columbia.

Zhang, N., & Poole, D. (1996). Exploiting contextual independence in Bayesian network inference. *Journal of Artificial Intelligence Research*, *5*, 301–328.