Collaborative and Confidential Junction Trees for Hybrid Bayesian Networks

Roberto Gheda

TU Delft

Collaborative and Confidential Junction Trees for Hybrid Bayesian Networks

by

Roberto Gheda

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday January 17, 2025 at 12:00

Student number:	5863503		
Project duration:	April 8, 2024 – January 17, 2025		
Thesis committee:	Prof. Lydia Y. Chen,	TU Delft,	supervisor
	Dr. Thiago Guzella,	ASML,	supervisor
	Dr. Carlo Lancia,	ASML,	supervisor
	Prof. Jie Yang,	TU Delft,	committee

This thesis is confidential and cannot be made public until July 17, 2025.

TU Delft

Preface

This thesis was a joint project between the TU Delft and ASML. I want to thank my supervisor from the Distributed Systems group, Prof. Lydia Chen, for the support and trust in giving me this meaningful opportunity. I want to thank my supervisors from ASML. In particular, I am thankful to Dr. Thiago Guzella for his warm and welcoming demeanor, and his consistent availability; and Dr. Carlo Lancia for his timely support and help. I want to thank both for their patience and guidance throughout the entire thesis journey. Furthermore, I would like to thank Abele Malan for his dedicated support and consistent availability. The expertise and help of all people mentioned above have been crucial for the completion of this project. I also want to thank my family and friends for their constant support and encouragement during the past years. Lastly, I want to thank Prof. Jie Yang for his approachability and flexibility in joining this committee.

Roberto Gheda
Delft, January 2025

Contents

1 Introduction **1**

2 Research Paper **3**

3 Background **15**
3.1 Bayesian Networks . . . 15
3.2 Inference on Bayesian Networks . . . 15
3.2.1 Variable Elimination . . . 16
3.2.2 Belief Propagation and Junction Trees . . . 16
3.3 Hybrid CLG Bayesian Networks . . . 17
3.3.1 Inference in Hybrid Bayesian Networks . . . 18
3.4 Distributed Bayesian Networks . . . 18
3.4.1 CCBNet . . . 18

4 Additional Experiments and Results **21**
4.1 Results on Continuous Data . . . 21
4.2 Results on Discrete Data . . . 22
4.3 Insights on the Hybrid Datasets . . . 22
4.3.1 Healthcare . . . 23
4.3.2 Sangiovese . . . 23
4.3.3 Mehra . . . 23

5 Conclusions **27**

1
Introduction

Bayesian Networks (BNs) are graphs that model a probability distribution by representing variables as nodes and dependencies between them as edges [1]. Compared to other machine learning approaches, BNs are capable of representing complex distributions while retaining interpretability. More importantly, Bayesian Networks can be designed by experts to represent their knowledge, i.e., not necessarily requiring data to be generated. These two characteristics make them relevant for several industries such as the ones where data is scarcely available.

In some scenarios, it might be required to leverage the knowledge of multiple BNs across multiple parties while retaining confidential information relevant to an individual party. This is the case for industries in which the manufacturers need to collaborate with customers to improve production efficiency while protecting trade secrets. CCBNet (Confidential Collaborative Bayesian Network) [2] is one of such studies.

While CCBNet supports representation of categorical variables, most real-world problems require dealing with a mixture of discrete and numerical data. The models that are capable of operating within this domain are called Hybrid Bayesian Networks [1, 3]. One of the most common classes of hybrid models is the set of *Conditional Linear Gaussian* (CLG) distribution models [4]. In this class, continuous nodes are Gaussian-shaped variables whose mean are linearly dependent on their parents. Furthermore, they cannot have any discrete child.

In this thesis project we aim at answering the following research questions:

1. How can we run Hybrid CLG inference in a distributed and privacy preserving fashion?
2. How can we reduce communication costs of collaborative inference over discrete variables?
3. How can we avoid revealing posterior of private variables in multi-party Bayesian Networks?

The thesis consists of three main parts. The first is a research paper that presents the main contributions presented in the proposed Hybrid CCJT framework and results. The second chapter provides extra insights into concepts relevant to the paper's content, like Bayesian Network inference procedure, Belief Propagation, inference on Hybrid CLG Bayesian Networks, and the other approaches to deal with Hybrid data. The third chapter present the additional experiments, followed by the conclusion chapter.

2

Research Paper

Collaborative and Confidential Junction Trees for Hybrid Bayesian Networks

Anonymous Authors[1]

Abstract

Bayesian Networks (BNs) are widely utilized across various industrial sectors to optimize processes, with an emerging focus on the collaboration across multiple parties. While most realistic scenarios require handling a mixture of categorical and continuous data simultaneously, the current state-of-the-art only supports collaborative inference on purely discrete models. The Junction Tree enables efficient and accurate inference on hybrid models but has not been implemented for confidential scenarios yet. To address this gap, we introduce `Hybrid CCJT`, an innovative framework for confidential multiparty inference in hybrid domains, offering: (i) a method to construct a collaborative, strongly-rooted junction tree for efficient and secure inference, (ii) a confidential-preserving inference protocol for Hybrid BNs, (iii) an optimized message-passing scheme that improves communication efficiency even in the purely discrete domain. Our extensive evaluation show that `Hybrid CCJT` improves the predictive accuracy of continuous target variables by an average of 32% in Mean Squared Error and reduce the communication cost up to 86-fold, against the best state-of-the-art baseline.

1. Introduction

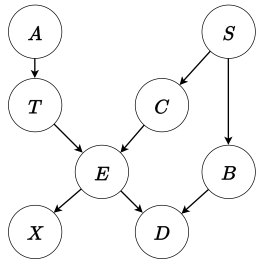
Bayesian Networks (BNs) (Pearl, 1988) have emerged as a powerful tool in numerous industrial domains for optimizing complex processes (Nannapaneni et al., 2016; McNaught & Chan, 2011), often requiring collaboration between parties. For instance, let us consider the use case of semiconductor manufacturing. Pursuing ever smaller size chips at a high yield (Ypma et al., 2020) entails cooperation between many specialized parties that must protect their trade secrets. Pursuing ever smaller size chips at a high yield (Ypma et al., 2020) entails cooperation between many specialized parties that must protect their trade secrets. In recent years, these probabilistic models have been explored in collaborative and confidential settings (Mălan, 2023), allowing stakeholders to extract insights and make decisions while protecting sensitive and proprietary knowledge.

While in most real-world applications, data often comprise a mix of categorical and continuous variables (Hertlein et al., 2020), the current state-of-the-art only supports confidential inference over purely discrete models (Mălan, 2023). Furthermore, the framework by Mălan lacks in scalability, as its communication costs grow exponentially depending on the parties' BNs size and structure. The current advancements in collaborative confidential inference for discrete networks are built upon variable elimination, which is a naive algorithm to take out exact inference in exponential time. Junction Trees provide a framework to make exact inference on larger instances tractable by decomposing the network in smaller ones. In hybrid domains, a Junction tree is said to be strongly-rooted if there exists a (strong) elimination order such that continuous nodes are eliminated from the graph before discrete ones (Madsen, 2008). If a Junction Tree is strongly-rooted it can be used to run accurate inference efficiently in hybrid BNs as well. However, Junction Trees have yet to be applied to collaborative confidential inference. Other studies in the field of distributed hybrid BNs disregarded model confidentiality constraints. For instance, Masegosa et al. (2016) provided a method to generate a centralized hybrid network from different parties' models. Albeit, structure and parameters of such generated network may leak confidential knowledge.

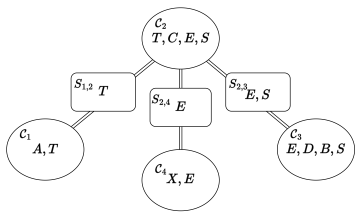
We propose `Hybrid CCJT`, the first framework that allows to run privacy-preserving multiparty inference over mixed data domains. `Hybrid CCJT` does not require a trusted third party, and protects confidentiality at both the levels of party models and data instances. The two key components of `Hybrid CCJT` are: (i) generation of a distributed strongly-rooted junction tree ; and (ii) a privacy-preserving inference protocol for Hybrid BNs. The novelty in the tree generation process lies in defining a variable elimination order that allows to perform accurate inference over hybrid models. Furthermore, we define an alignment procedure for discrete probability tables that allows marginalizing variables prior to message passing, improving scalability of

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

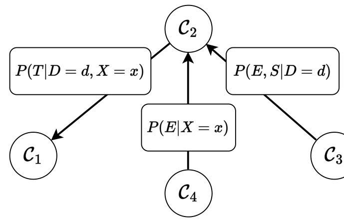
Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.



(a) The ASIA network. Having 8 variables in total, the cost of running inference directly on it is $\mathcal{O}(2^8)$.



(b) A Junction Tree defined over ASIA. The largets cliques are $\mathcal{C}_2, \mathcal{C}_3$ with 4 variables each. They bound the cost of inference to $\mathcal{O}(2^4)$.



(c) Belief is propagated toward clique $\mathcal{C}_1$ for query with evidence $X = x, D = d$.

Figure 1. Example of construction and inference on a Junction Tree.

communication costs. To execute inference queries confidentially, we propose two secret-sharing schemes to merge parties continuous and discrete knowledge while not disclosing computation results. We evaluate `Hybrid CCJT` against twelve different datasets spanning purely discrete, purely continuous, and hybrid domains. We compare it with non-hybrid confidential approaches, measuring significant improvements under both inference accuracy and communication costs.

In summary, we make the following contributions:

- We propose the first privacy-preserving inference framework on Hybrid Bayesian Networks by allowing parties to collaboratively run belief propagation without sharing private variables' posteriors.
- We design the first method to define a variable elimination order for strong marginalization in multiparty Hybrid Bayesian Networks by building a distributed strongly-rooted junction tree.
- We improve communication efficiency compared to the state-of-the-art by marginalizing all discrete private variables prior to message passing.
- We evaluate our method against twelve different datasets and analyze improvements compared to non-hybrid confidential methods. We measure 32% average improvement in mean squared error and up to 86-fold improvements in communication costs.

2. Background and Related Studies

2.1. Hybrid CLG Bayesian Networks

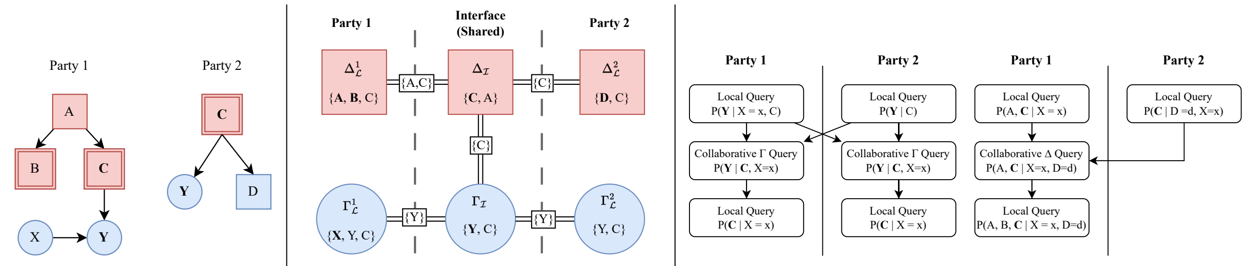
Bayesian Networks (BNs) (Pearl, 1988) are directed acyclic graphs whose nodes are random variables and whose edges correspond to direct influence of one node on another. The *conditional probability distribution* (CPD) of a variable given its parents -$P(x \mid \text{pa}(x))$- is called its *factor*. The table that summarizes all CPD for all variables is called CPD table. Traditionally, BNs only allow variables to be discretely valued (Koller & Friedman, 2009). However, such a requirement limits the representation quality for variables which are better represented by real-valued data (Salmerón et al., 2018). Moreover, exact inference in discrete BNs is NP-hard, while other continuous representations, such as *Conditional Linear Gaussian* (CLG) (Koller & Friedman, 2009), can perform exact inference with polynomial cost in the network size (Koller & Friedman, 2009).

Hybrid Bayesian Networks (Salmerón et al., 2018) allow to model probability distribution with both discrete (Δ) and continuous variables (Γ) simultaneously. One of the most common classes of hybrid models is the set of *Hybrid Conditional Linear Gaussian* (Hybrid CLG) distribution models (Lauritzen & Wermuth, 1989). In this class, continuous variables are Gaussian-shaped and cannot have any discrete child. The factor of a continuous variable $x \in \Gamma$ with discrete parents z_Δ and continuous parents z_Γ is given by:

$$f(x|\{z_\Delta, z_\Gamma\}) = \mathcal{N}(x; \alpha(z_\Delta) + \beta(z_\Delta)^T z_\Gamma, \sigma^2(z_\Delta))$$

where α and β are the coefficients that depend on the discrete state combination of z_Δ. If the state combination of z_Δ is fixed, x is Gaussian-shaped. If this is not the case, $f(x)$ is a mixture of $\mathcal{O}(2^{|\Delta|})$ Gaussian distributions. In general, even representing the correct marginal distribution in a hybrid CLG network require space that is exponential in the size of network (Koller & Friedman, 2009). Furthermore, even approximate inference for simple models structures such as polytrees is NP-hard in hybrid CLG networks (Lerner & Parr, 2013).

Inference. Lauritzen (1992; 2001) develops an algorithm to carry out accurate inference in Hybrid CLG BNs by leveraging a strong elimination order. That is, an order such that continuous nodes are eliminated from the graph before discrete ones (Madsen, 2008). On top of Lauritzen's works, Madsen (2008) builds an algorithm for running centralized lazy propagation (Madsen & Jensen, 1999). Lerner (2013)



(a) Parties' input hybrid BNs. (b) Collaborative Junction Tree. (c) Collaborative inference with evidence $X = x, D = d$.

Figure 2. Overview of `Hybrid CCJT`. Discrete variables are represented as red squares, continuous variables are represented as blue circles. Factor variables are highlighted in bold.

extended the algorithm of Lauritzen to allow inference for discrete children of continuous variables by approximating them using a softmax function. Other works on Hybrid CLG only allow for a subset of inference queries. For instance, Paskin (2003) leverages the Rao-Blackwell theorem to provide tractable approximated inference but does not allow queries with continuous evidence.

2.2. Junction Trees

Exact inference in discrete BNs is NP-hard as its cost grows exponentially in the number of variables in the network. A widely used technique to make inference on larger instances tractable is the Junction Tree (Lauritzen & Spiegelhalter, 1988). Similarly to tree decomposition, the original network is decomposed into a tree-like graph where each node - called *clique*- contains a sub-graph of the original BN. After resolving inference in the small cliques, these results can be combined via a message-passing algorithm called *belief propagation*, which is proven to converge in linear time on trees. Doing so, the cost of exact inference is bounded to the size of the largest clique.

In Figure 1a we showcase the popular ASIA network (Lauritzen & Spiegelhalter, 1988) as an example. Here, running Variable Elimination requires $\mathcal{O}(2^8)$ operations to carry-out exact inference. In contrast, the cost of inference on the Junction Tree in Figure 1b is bounded by the size of the largest clique. Thus, requiring only $\mathcal{O}(2^4)$ operations.

2.3. Distributed and Confidential Bayesian Networks

CCBNet (Mălan, 2023) is the current state-of-the-art in the field of distributed confidential BN inference. It is based on two protocols: `CABN` (Confidentially Augmented Bayesian Networks) and `SAVE` (Share Aggregation Variable Elimination). Briefly, `CABN` privately performs *alignment* of factors of common variables, while `SAVE` performs distributed inference based on Variable Elimination and a BN merging scheme inspired by (Del Sagrado & Moral, 2003) and (Feng et al., 2014).

Despite being a significant step forward in the Confidential BN literature, CCBNet falls short under several aspects. Namely, its communication costs have been shown to explode even for relatively small problems, it only supports discrete variables, and posterior probability values for some private variables from the peers are revealed to the party executing a query.

3. Hybrid CCJT

In this section, we first provide the preliminary of Junction Trees and then detail the collaborative BN architecture and the confidential inference protocol of `Hybrid CCJT`.

3.1. Preliminaries on Junction Trees

A Junction Tree of a Bayesian Network over variables $\mathcal{X}$ with set of factors Φ is a computational graph whose nodes c_i, also called *cliques*, are tuples $(X_i \subseteq \mathcal{X}, \Phi_i \subseteq \Phi)$. Edges, also called *separators*, are associated with a set of variables called *sepset* $S_{i,j} = X_i \cap X_j$. These edges connect the cliques to form a tree. In order to be valid, a Junction Tree must satisfy the following rules:

- *Family preservation*: Each factor $\phi \in \Phi$ must be associated with one cluster c_i such that Scope$[\phi] \subseteq X_i$.
- *Running-intersection property*: For every pair of cliques c_i, c_j, every clique on the path between c_i, c_j contains $X_i \cap X_j$.

Exact inference can be run on Junction Trees via message passing schemes, one of the most popular being the sum-product algorithm (Shenoy & Shafer, 1990). Let us define the potential of each clique c_i as $\phi(c_i) = \prod_{\phi_j \in \Phi_i} \phi_j$. This scheme requires cliques to send messages through the tree towards a root as:

$$\mu_{c_i \to c_j} = \sum_{v \notin S_{i,j}} \phi(c_i) \prod_{c_k \in \text{nb}(c_i) \setminus c_j} \mu_{c_k \to c_i} \qquad (1)$$

Where $\mu_{c_k \to c_i}$ is a message sent from clique c_k to clique c_i in form of a CPD table with scope $S_{k,i}$. $\sum$ represents the variable elimination operator, used to remove variables not in clique j before message passing.

Strongly-Rooted Junction Trees Since regular Junction Trees do not always have a strong elimination order (Madsen, 2008), they cannot be used to run inference correctly on Hybrid CLG models. Instead, a stricter structure called *Strongly-Rooted Junction Tree* (Lauritzen, 1992) is required. A Junction Tree is said to be strongly-rooted if it has a distinguished clique R, called *strong root*, such that for every couple of neighboring cliques (C, D), with C being closer to R than D, it holds that:

$$C \cap D \subseteq \Delta \quad \vee \quad D \setminus C \subseteq \Gamma$$

Lauritzen (1992; 2001) shows that this data structure can be used to compute exact marginals of all discrete variables (*strong marginalization*) and exact first and second moment of all continuous variables by approximating them as multivariate Gaussians (*weak marginalization*).

3.2. Method

`Hybrid CCJT` (**Hybrid** **C**ollaborative **C**onfidential **J**unction **T**rees) is a framework that allows to run privacy-preserving multiparty inference over mixed data domains. It is made of two main components: (i) Collaborative Junction Tree Networks, allows parties to generate a collaborative strongly-rooted junction tree for hybrid belief propagation, (ii) Confidential Inference Protocol, allows parties to jointly perform the exact inference as defined by Lauritzen (2001).

We assume that features from different parties have the same name only if they represent the same concept, and that the independence, across parties, of distinct parents for the same node reasonably approximates the ground truth. Thus, names identify the common nodes between models, serving as the contact points for graph fusion. Our **adversarial model** includes semi-honest parties that follow the protocol while trying to abuse gained information (Goldreich et al., 2005) but do not collude. No trusted third party exists. The goal is to protect all network parameters and only share structure/state-name information among parties modeling the same variables. Figure 2 shows an overview of the steps of `Hybrid CCJT`.

In order to achieve **confidentiality**, we construct a collaborative junction tree without sharing any information except the names of shared variables. We align discrete variables' factors by exponentiating their entries and privately computing normalization values of each column of the CPD tables. Then, the querying party takes care of such normalization at inference time, allowing peers to marginalize private variables in the scope of the common prior message passing. To merge continuous variables, we propose a merging scheme based on weak marginalization and use a secret sharing scheme for addition (Garcia & Jacobs, 2011) to combine parties' beliefs.

3.3. Collaborative Junction Tree Networks

Collaborative Junction Tree Networks is a protocol that allows to setup a collaborative strongly-rooted Junction Tree without sharing any confidential information. First, we construct cliques and separators of the tree, which enables to find a variable elimination order for strong marginalization. Afterward, the factors of common discrete variables are augmented confidentially, allowing parties to combine their beliefs at inference time.

The protocol requires every party to define a input Bayesian Network (either Hybrid CLG, Discrete or CLG) describing their knowledge and data (Figure 2a). Network structure and parameters can be defined by human experts or learned via structure and parameter learning methods (Scutari & Denis, 2021). Before running the protocol, we find *interface* variables, that are common variables between parties. We do so via *Private Set Intersection* (Morales et al., 2023). The collaborative Junction Tree is defined given each party's input network and the set of interface variables.

We first construct the junction tree for each party. For each party i we define two *local cliques*, one discrete ($\Delta^i_{\mathcal{L}}$) and one continuous ($\Gamma^i_{\mathcal{L}}$). Then, we define two *interface cliques*, one discrete ($\Delta_{\mathcal{I}}$) and one continuous ($\Gamma_{\mathcal{I}}$), over factors of all interface variables and their scope, serving as interface between parties' local cliques. Thus, each party has four cliques.

Next, we define the set of variables and factors to be associated with each clique (Figure 2b). Let $\Phi(X)$ denote the set of factors for each $x \in X$, where X is a set of variables. Each local clique contains all the variables and factors owned by each party minus the factors of the interface variables. Namely, every party i owns $\Delta^i_{\mathcal{L}} = (\Delta_i, \Phi(\Delta_i \setminus \mathcal{I}))$ and $\Gamma^i_{\mathcal{L}} = (\Gamma_i, \Phi(\Gamma_i \setminus \mathcal{I}))$. Interface cliques contain all variables in the scope of interface variables and factors of interface variables. Thus, $\Delta_{\mathcal{I}} = (\text{Scope}[\Delta \cap \mathcal{I}], \Phi(\Delta \cap \mathcal{I}))$ and $\Gamma_{\mathcal{I}} = (\text{Scope}[\Gamma \cap \mathcal{I}], \Phi(\Gamma \cap \mathcal{I}))$.

Cliques are then connected to form a tree. Discrete and continuous local cliques are connected with discrete and continuous interface cliques respectively. The two interface cliques are connected with each other with the separator being the set of *threshold variables*, the set of discrete variables with continuous children ($\Delta \cap \mathbf{pa}(\Gamma)$).

We merge discrete party beliefs using a weighted geometric mean inspired by (Del Sagrado & Moral, 2003). Hence, we define the remote clique potential as:

$$\phi(\Delta_{\mathcal{I}}) = \alpha \prod_{i \in \mathbb{P}} \phi(\Delta^i_{\mathcal{I}})^{\frac{w_i}{\sum_{j \in P} w_j}}, \tag{2}$$

Where w_i is the weight of party i, which represent confi-

Algorithm 1 Federated Hybrid Query

Input: Target T, Evidence $\mathcal{E}$, Party Q, Peers P

1: *FederatedContinuousInference*($\mathcal{E}^{\Gamma}$, {Q} $\cup$ P)
2: strong-marginals $\leftarrow$ *FederatedDiscreteInference*($\mathcal{E}^{\Delta}$)
3: Δ-marg $\leftarrow$ *VarElim*(thresholdVars $\cup$ T^{Δ}, {}, strong-marginals)
4: posterior $\leftarrow$ *WeakMarginalization*(Δ-marg, T^{Γ})
5: **return** posterior

Algorithm 2 Collaborative Continuous Inference

Input: Evidence $\mathcal{E}$, Parties P

1: **for** $p \in P$ **do**
2: $\quad p$.ovBelief $\leftarrow$ *posterior*(p.overlap, $\mathcal{E}$)
3: **end for**
4: ovBelief $\leftarrow$ *secretShare*($\bigcup_{p \in P}$ p.ovBelief)
5: **for** $p \in P$ **do**
6: $\quad p$.thresholdBelief $\leftarrow$ *canonicalPosterior*({}, ovBelief)
7: **end for**

dence in its BN and is publicly known. α is the column normalization factor that is applied to each table's column based on the potential alignment process. To implement this, parties allocate space in the interface variables' CPDs to account for parents managed by other peers with obfuscated names. Then, parties collaboratively compute column normalization factors α via homomorphic encryption (Cheon et al., 2017). This normalization is only applied during inference by the querying party, enabling peers to marginalize their private variables before message passing, where CPD entries are secret shared via a multiplication-based scheme (Kilbertus et al., 2018), enhancing both communication costs and privacy guarantees. Parties don't need to share any information to define $\Gamma_{\mathcal{I}}$, as our merging scheme for CLG variables does not require any factor alignment.

3.4. Confidential Inference Protocol

Confidential Inference Protocol has parties collaboratively run inference over the Junction Tree previously constructed. The protocol pseudocode is outlined in Algorithm 1. Belief is propagated towards a strong root through two steps: first, parties collaboratively run inference over continuous domain to compute marginals over threshold variables (line 1), then, they run collaborative discrete inference (line 2). Following that, the querying party can leverage the evidences of other parties to run local queries (ll. 3-5).

Confidential Continuous Inference During this step, we aim to merge parties local continuous evidence to find the strong marginals over threshold variables. The pseudocode of this procedure is outlined in Algorithm 2.

Algorithm 3 Collaborative Discrete Inference

Input: target T, evidence $\mathcal{E}^{\Delta}$, party Q, peers P

1: auxFacts $\leftarrow$ {}
2: **for** $p \in P$ **do**
3: $\quad$ partyFacts $\leftarrow$ $\bigcup_{cpd \in p.\text{CPDs}}$Factor(cpd)
4: $\quad$ partyFacts $\cup \leftarrow$ p.thresholdBelief
5: $\quad$ partyT $\leftarrow$ T $\cup$ overlapNodes(p)
6: $\quad$ auxFacts $\cup \leftarrow$ *VarElim*(partyT, $\mathcal{E}^{\Delta}$, partyFacts)
7: **end for**
8: auxFacts $\bigcup_{cpd \in \text{Q.cpds} \cap \text{overlapNodes(Q)}}$ $\leftarrow$ normFactor(cpd)
9: auxFacts $\bigcup_{cpd \in \text{Q.CPDs} \setminus \text{overlapNodes(Q)}}$ $\leftarrow$ Factor(cpd)
10: **return** *VarElim*($\Delta_Q \cup$ T, $\mathcal{E}^{\Delta}$, auxFacts)

Messages from $\Gamma^i_{\mathcal{L}}$ to $\Gamma_{\mathcal{I}}$ are derived by each party without interaction (ll. 1-3). When computing a message from $\Gamma_{\mathcal{I}}$ to $\Delta_{\mathcal{I}}$ we aim to find strong marginals over discrete parents of continuous variables. To do so, we merge the knowledge of continuous interface variables. Parties then marginalize all continuous variables to find strong marginals. In order to merge parties continuous knowledge, we propose a weighted merging scheme inspired by weak marginalization (Koller & Friedman, 2009) in which mean and variance of interface variables are updated as follows:

$$\boldsymbol{\mu} = \sum_{i \in P} w_i \mu_i \tag{3}$$

$$\boldsymbol{\Sigma} = \sum_{i \in P} (w_i \Sigma_i) + \sum_{i \in P} (w_i (\boldsymbol{\mu} - \mu_i)(\boldsymbol{\mu} - \mu_i)^T) \tag{4}$$

For the purpose of preserving confidentiality, we use additive secret-sharing with no trusted third party (Garcia & Jacobs, 2011) (l. 5). Each party randomly splits its secret value in as many shares as the number of parties and send it to each of them. Every party computes addition over the values it received. Finally, parties share the addition outcome with each other and compute the sum to find the final result. To compute Equation 3, each party secret shares $w_i \mu_i$. While to find Equation 4, each party secret shares $w_i \Sigma_i + w_i (\boldsymbol{\mu} - \mu_i)(\boldsymbol{\mu} - \mu_i)^T$.

Finding strong marginals requires integrating out all continuous variables while taking into account their evidences. When using canonical representation, this can be done by marginalizing out all continuous variables and exponentiating the marginalization outcome (ll. 6-8). Then, each entry of the CPD table is assigned the marginalization outcome corresponding to its state combination.

Confidential Discrete Inference Once marginals of threshold variables are computed, parties collaboratively calculate the message from $\Delta_{\mathcal{I}}$ to $\Delta^Q_{\mathcal{L}}$ to finalize strong marginalization. The pseudocode is outlined in Algorithm 3. Fort that, each non-querying party p_i find posteriors over their discrete domain Δ_i. Then, following Equation 1 and Equation 2, we find $\mu_{\Delta_{\mathcal{I}} \to \Delta^Q_{\mathcal{L}}}$:

$$\sum_{x \notin \mathbb{X}_q} \phi(\Delta_{\mathcal{I}}) \prod_{p_i \in \mathbb{C} \setminus \{Q\}} \mu_{\Delta^i_{\mathcal{L}} \to \Delta_{\mathcal{I}}} \tag{5}$$

$$= \sum_{x \notin \mathbb{X}_q} \alpha \prod_{p_i \in \mathbb{C}} \phi(\Delta^i_{\mathcal{I}})^{w_p} \prod_{p_i \in \mathbb{C} \setminus \{Q\}} \mu_{\Delta^i_{\mathcal{L}} \to \Delta_{\mathcal{I}}} \tag{6}$$

$$= \sum_{x \notin \mathbb{X}_Q} \alpha \phi(\Delta^Q_{\mathcal{I}})^{w_Q} \prod_{p_i \in \mathbb{C} \setminus \{Q\}} \phi(\Delta^i_{\mathcal{I}})^{w_p} \mu_{\Delta^i_{\mathcal{L}} \to \Delta_{\mathcal{I}}} \tag{7}$$

$$= \alpha \phi(\Delta^Q_{\mathcal{I}})^{w_q} \sum_{x \notin \mathbb{X}_q} \prod_{p_i \in \mathbb{C} \setminus \{Q\}} \phi(\Delta^i_{\mathcal{I}})^{w_p} \mu_{\Delta^i_{\mathcal{L}} \to \Delta_{\mathcal{I}}} \tag{8}$$

$$= \alpha \phi(\Delta^Q_{\mathcal{I}})^{w_Q} \prod_{p_i \in \mathbb{C} \setminus \{Q\}} \sum_{x \notin \mathbb{X}_q} \phi(\Delta^i_{\mathcal{I}})^{w_p} \mu_{\Delta^i_{\mathcal{L}} \to \Delta_{\mathcal{I}}} \tag{9}$$

where $\sum$ is the variable marginalization operator. The most important step is Equation 9. When multiplying CPD tables, we can marginalize variables that are not common in both tables prior to the product without affecting the result. This leads to a severe reduction of communication costs as each variables doubles the size of the shared CPD.

From Equation 9, it follows that each non-querying party i has to compute message $\mathrm{m}_i = \sum_{x \notin \mathbb{X}_q} \phi(\Delta^i_{\mathcal{I}})^{w_p} \mu_{\Delta^i_{\mathcal{L}} \to \Delta_{\mathcal{I}}}$. This can be done locally as $\phi(\Delta^i_{\mathcal{I}})^{w_p}$ is the outcome of discrete interface alignment known to party i and $\mu_{\Delta^i_{\mathcal{L}} \to \Delta_{\mathcal{I}}} = \sum_{x \notin \mathcal{I}} \phi(\Delta^i_{\mathcal{L}})$ can be computed locally with no interaction among parties (ll. 2-7).

Eventually, m_i is a factor over variables owned by the querying party Q. To protect the content of this messages, we use a secret sharing scheme for multiplication (Kilbertus et al., 2018). A secret value is split into shares distributed amongst parties. Parties perform the computation with their local share of each secret and all aggregate their results to reconstruct the answer.

Weak Marginalization Once discrete posteriors are computed, continuous ones can be found following the weak marginalization procedure (Koller & Friedman, 2009). Thus, for any continuous variables X that has at least one discrete parent we get:

$$\boldsymbol{\mu}_X = \sum_{s \in S} p(s) \mu_{X,s}$$

$$\boldsymbol{\Sigma}_X = \sum_{s \in S} p(s) \Sigma_{X,s} + \sum_{s \in S} p(s) (\boldsymbol{\mu}_X - \mu_{X,s})(\boldsymbol{\mu}_X - \mu_{X,s})^T$$

Where $p(s)$ is the probability of state combination s, S is the set of all possible state combinations of threshold variables, $\mu_{X,s}$ and $\Sigma_{X,s}$ are the parameters of X dependent on s.

3.5. Note on the Confidentiality

Below, we review how the different steps in our method enable it to maintain our confidentiality objective:

Table 1. Datasets stats.

Dataset	#Discrete nodes	#Continuous nodes	#Arcs	#Params
Healthcare	3	4	9	42
Sangiovese	1	14	55	259
Mehra	8	16	71	324423
Asia	8	-	8	18
Child	20	-	25	230
Alarm	37	-	46	509
Insurance	27	-	52	1008
Andes	223	-	338	1157
Link	724	-	1125	14211
Munin #2	1003	-	1244	69431

Junction Tree Construction First, we want to define a strong elimination order in the global network. We do so by only disclosing which variables are common between the parties. In fact, every party knows which of their variables to allocate in the local and interface cliques. This is sufficient to carry out inference as in our protocol, but no party knows anything about the content nor the structure of other parties' local cliques.

Early marginalization of private discrete variables We ensure that only the querying party handles the normalization of interface factors process. As shown in Equation 9, this allows each party to marginalize all variables that are not shared with the querying party prior to message passing. As such, the merging procedure will only reveal the posterior of the interface variables, which are owned by the querying party. Finally, this approach leads to a severe reduction of communication costs, which we extensively discuss in section 4.

Merging scheme for continuous variables Similarly to what we do for discrete variables, we aim to find the updated posteriors of common continuous variables while abiding our confidentiality assumptions. Parties merge the posterior of common variables via a secret sharing scheme for addition. This allows the querying party to find the merged posterior without having access to the peers' private posteriors, nor any parameter of the network.

4. Evaluation

We evaluate `Hybrid CCJT`'s predictive performance and communication costs on a total of twelve publicly available datasets (two of which are in Appendix A) whose data structures are either hybrid or discrete only. We compare it against the current state-of-the-art on different types of queries. Furthermore, we also include results on purely continuous data in Appendix A.

Evaluation Metrics Our experiments assess the average predictive performance for both discrete and continuous target variables, as well as the associated communication costs. For discrete variables, we evaluate the prediction quality using the Brier Score ($= \frac{1}{N} \sum_{t=1}^{N} \sum_{i=1}^{R} (f_{it} - o_{it})^2$

where N is the number of queries, R is the number of target variables state combinations and f and o are the predicted and reference probabilities). For continuous variables, we use the Mean Squared Error (MSE) of predicted means with respect to the reference values. Communication costs are calculated as the average number of CPD and CLG parameter values transmitted per query.

Dataset We consider the 10 data sets shown in Table 1. To define parties' input networks, we first generate a dataset of subnetworks by sampling from the original network. Then, we vertically split datasets and learn structure and parameters via 2-phase Restricted Maximization and Maximum Likelihood Estimator for conditional probabilities (for discrete variables) and least squares regression models (for CLG variables). Each vertical split has a different overlap ratio. That is, the ratio between the amount of common nodes and the amount of nodes in the global network.

Baselines We test our method against ten different datasets shown in Table 1 and compare `Hybrid CCJT`'s performance with two baselines:

- **CCBNet** (Mălan, 2023): the current state-of-the-art for distributed confidential inference in Bayesian Networks.

- **Δ-CCJT**: a simplified `Hybrid CCJT` variant with only the federated discrete inference as in Algorithm 3.

Since none of the baselines can handle continuous data, when such variables are present in the dataset, we discretize them with different degrees of coarseness ranging from 3 to 10 states per variable. Given the hardness of discrete exact inference (Koller & Friedman, 2009), a finer discretization implies significantly higher computational and communication costs, making it infeasible to run these algorithms with large number of states per variable. All variables are discretized using a quantile-based approach, ensuring that each discrete state contains an equal number of samples from the training set.

4.1. Results on Hybrid Data

Here, we consider three hybrid data sets, namely Healthcare, Sangiovese, and Mehra. We test `Hybrid CCJT` with different combinations of number of involved parties and overlap ratios for a total of 10 total experiment scenarios. For each of them, we run 1000 queries with discrete target variables and 1000 queries with continuous target variables. We summarize the results in Table 2.

Predictive Accuracy In all experiment scenarios considered, `Hybrid CCJT` outperforms all baselines in predictive accuracy of continuous target variables with an average 32% improvement of MSE compared with the best performing baseline. When targeting discrete variables, `Hybrid CCJT` is either the best performing solution or the second best performing solution with a performance gap always under 10^{-3} in terms of Brier score. The only experiment that does not fit this trend is Sangiovese with 4 parties and 30% overlap, where the deficit to the best model is 0.0015 (0.0125 versus 0.015). Recalling that the Sangiovese dataset contains only one discrete variable, we experimented that this returns a quasi-uniform distribution regardless of the set of continuous evidences. This explains why running hybrid inference on this data does not lead to any improvement rather than using a discretized version. As one would expect, the best performing baseline is the one with a finer discretization. Using a low number of discrete states leads to a drastic performance decay compared to our implementation with an MSE 26.9 times higher, and a Brier score 42.8% higher on average. In summary, `Hybrid CCJT` brings notable improvements when targeting continuous variables, proving the benefit of natively handling continuous data. Nonetheless, the performance of `Hybrid CCJT` matches the baselines when targeting discrete variables.

Communication Costs `Hybrid CCJT` demonstrates superior scalability in communication costs compared to all discretized baselines. Although Δ-CCJT achieves the lowest communication costs on the smaller Healthcare dataset, this advantage diminishes with larger datasets where we can start to appreciate `Hybrid CCJT` scalability. Under Sangiovese, which includes a higher number of continuous variables, the communication costs of Δ-CCJT with 10 states increase significantly faster than those of our method, reaching up to 40 times more communicated values per query. This happens because discrete CPD table take more space than regular continuous posteriors. Further highlighting the advantage of handling continuous data natively. While baselines with fewer states may reduce communication costs in certain experimental setups, this comes at the expense of a sharp decline in predictive performance. For example, on Sangiovese with 4 parties and 10% overlap, Δ-CCJT exhibits nearly double the error for discrete targets and 200 times the error for continuous ones, making `Hybrid CCJT` the most desirable choice overall. The current state-of-the-art model, CCBNet, brings the worst communication performance across all experiments. Its worst result averages almost 190k communicated values per query against the only 596 used by `Hybrid CCJT`. This shows how our collaborative discrete inference approach alone brings significant improvements under communication efficiency. We explore this more in detail in subsection 4.2.

Large data Mehra is the largest dataset among all the experimented ones, with 4 times the amount of parameters of the largest discrete dataset Munin. While `Hybrid CCJT` managed to complete all experiments, none of the discretized baselines managed to finish running within the timeout

Table 2. Results on hybrid data. Best result in bold. Second best Brier score underlined. Lower is better for all.

Dataset			**Healthcare**				**Sangiovese**			
Parties			2		4		2		4	
Overlap			10%	30%	10%	30%	10%	30%	10%	30%
Hybrid CCJT		Brier	0.0496	**0.036**	**0.0577**	**0.0856**	0.019	**0.0129**	0.02746	0.015
		MSE	**4.7e+06**	**4.6e+05**	**4.8e+06**	**1.4e+07**	**0.0033**	**0.0018**	**0.00041**	**0.0045**
		Comm	4.7	16.6	11.4	139.7	**43.5**	**87**	219	**596**
Δ-CCJT	3 States	Brier	0.0557	0.058	0.0651	0.128	0.0457	0.0138	0.0484	**0.0125**
		MSE	5.9e+06	4.9+05	5.4e+06	1.7e+07	0.044	0.021	0.083	0.0071
		Comm	4.6	**9.3**	8.3	**23.9**	44.6	164.8	**133.6**	2654
	5 States	Brier	0.0502	0.0578	0.0623	0.173	0.0243	0.0132	0.0476	0.013
		MSE	5e+06	5.2e+05	5.3e+06	1.6e+07	0.025	0.012	0.012	0.0218
		Comm	4.6	18.9	**5.2**	36.2	71	261.3	216.3	4174
	10 States	Brier	**0.0488**	0.044	0.0597	0.112	**0.0181**	**0.0129**	**0.0269**	0.013
		MSE	4.8e+06	4.9e+05	5e+06	1.6e+07	0.012	0.0036	0.0069	0.0049
		Comm	**4.0**	58.2	**5.2**	66.7	140.1	213.7	424.7	24013
CCBNet	3 States	Brier	0.0558	0.0568	0.0651	0.128	0.0496	0.0138	0.05769	0.0125
		MSE	5.8e+06	4.9+05	5.4e+06	1.7e+07	0.044	0.02	0.064	0.007
		Comm	38.7	15.2	17.0	34.3	196	793.6	19044	20271
	5 States	Brier	0.0502	0.0571	0.0623	0.172	0.0243	0.0132	0.0464	0.013
		MSE	5e+06	5.2e+05	5.3e+06	1.6e+07	0.025	0.011	0.011	0.0215
		Comm	12.6	28.2	9.6	161.1	52.8	808.9	4273.3	377341
	10 States	Brier	**0.0488**	0.044	0.0597	0.113	**0.0181**	**0.0129**	**0.0269**	0.013
		MSE	4.8e+06	4.9e+05	5e+06	1.6e+07	0.012	0.0037	0.0069	0.0049
		Comm	9.7	135.2	9.6	560.2	154.1	1381.9	4074.6	189650

Table 3. `Hybrid CCJT` results on the large hybrid dataset (Mehra).

Dataset		**Mehra**	
#Parties		8	
Overlap		10%	30%
Hybrid CCJT	Brier	0.00783	0.00772
	MSE	7.4e+11	4.5e+12
	Comm	186	6734

limit[1].This is due to the heavy computational requirements of aligning large CPD tables of discretized continuous variables. Despite the size and hardness of predictive accuracy under this model, `Hybrid CCJT` manages to achieve good Brier Score and MSE, while maintaining reasonable communication costs throughout the experiments. Specifically, with 30% overlap, `Hybrid CCJT` communicates less than a third of the values shared by Δ-CCJT with 10 states when dealing with half the parties in a much smaller dataset like Sangiovese.

4.2. Results on Discrete Data

Since scalability of communication costs is a significant issue for CCBNet (Mălan, 2023), we specifically emphasize the improvement of `Hybrid CCJT` under this aspect on purely discrete data. On datasets Child, Alarm and Insurance, we run experiments with different numbers of parties involved ranging from 2 to 8. Furthermore, we run experiments on larger datasets with up to 128 parties involved. For each experiment, we perform 2000 different queries. Results are showcased in Table 4. Since these datasets do not include any continuous variable, `Hybrid CCJT` degenerates into Δ-CCJT.

In smaller experiments, with two parties only, `Hybrid CCJT` reduces CCBNet communication costs by 8 times, from an average of 125 communicated values to an average of only 15. Then, improvement factors further increase when the number of parties involved is greater. In larger-scale experiments, CCBNet's communication costs increase significantly, reaching as high as 243k transmitted values in the Munin experiment. In contrast, `Hybrid CCJT` maintains a low communication overhead, transmitting only 735 values in the same experiment, achieving an improvement factor of 331. We did not measure any significant difference (≥ 0.001) in predictive accuracy in these experiments.

5. Conclusions

This work introduces `Hybrid CCJT`, a novel framework enabling privacy-preserving multiparty inference on Hybrid CLG Bayesian Networks. By addressing key limitations of existing methods, `Hybrid CCJT` facilitates secure collaborative inference while maintaining strict confidentiality of both party models and data. The proposed framework incorporates two pivotal components: a distributed strongly-

[1]Timeout limit of the experiment is 24 hours, on 512GB RAM.

Table 4. Results on discrete data: communication costs on discrete datasets. 10% overlap. Lower is better for all. Predictive accuracy difference is negligible ($< 10^{-3}$).

Dataset	#Parties	CCBNet	Hybrid CCJT	Improvement factor
Child	2	67	14	4.8x
	4	157	33	4.7x
	8	429	58	7.4x
Alarm	2	166	15	11.1x
	4	1959	40	49.0x
	8	1886	79	23.9x
Insurance	2	143	17	8.4x
	4	1835	43	42.7x
	8	473	42	11.3x
Andes	16	23175	6080	3.8x
Link	64	4455	459	9.7x
Munin #2	128	243474	735	331.3x

rooted junction tree for determining an elimination order and a privacy-preserving inference protocol that leverages such an elimination order.

Our evaluation demonstrates the efficacy of our method. For hybrid data, `Hybrid CCJT` improves the predictive accuracy of continuous target variables by an average of 32% in Mean Squared Error (MSE) compared to the best performing baseline. For discrete targets, it consistently achieves either the best or second-best results, with performance gaps below 10^{-3} in most cases. In addition, `Hybrid CCJT` outperforms existing baselines in communication costs, with up to 86-fold reductions in large hybrid datasets and 331-fold improvements in large-scale experiments on purely discrete data. Altogether, by natively handling continuous and discrete data, `Hybrid CCJT` offers better predictive quality and communication costs scalability compared to the state-of-the-art.

References

Cheon, J. H., Kim, A., Kim, M., and Song, Y. Homomorphic encryption for arithmetic of approximate numbers. In *Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I 23*, pp. 409–437. Springer, 2017.

Del Sagrado, J. and Moral, S. Qualitative combination of bayesian networks. *International Journal of Intelligent Systems*, 18(2):237–249, 2003.

Feng, G., Zhang, J.-D., and Liao, S. S. A novel method for combining bayesian networks, theoretical analysis, and its applications. *Pattern Recognition*, 47(5):2057–2069, 2014.

Garcia, F. D. and Jacobs, B. Privacy-friendly energy-metering via homomorphic encryption. In *Security and Trust Management: 6th International Workshop, STM 2010, Athens, Greece, September 23-24, 2010, Revised Selected Papers 6*, pp. 226–238. Springer, 2011.

Goldreich, O. et al. Foundations of cryptography–a primer. *Foundations and Trends® in Theoretical Computer Science*, 1(1):1–116, 2005.

Hertlein, N., Deshpande, S., Venugopal, V., Kumar, M., and Anand, S. Prediction of selective laser melting part quality using hybrid bayesian network. *Additive Manufacturing*, 32:101089, 2020.

Kilbertus, N., Gascón, A., Kusner, M., Veale, M., Gummadi, K., and Weller, A. Blind justice: Fairness with encrypted sensitive attributes. In *International Conference on Machine Learning*, pp. 2630–2639. PMLR, 2018.

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

Lauritzen, S. L. Propagation of probabilities, means, and variances in mixed graphical association models. *Journal of the American Statistical Association*, 87(420):1098–1108, 1992.

Lauritzen, S. L. and Jensen, F. Stable local computation with conditional gaussian distributions. *Statistics and Computing*, 11:191–203, 2001.

Lauritzen, S. L. and Spiegelhalter, D. J. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 50(2):157–194, 1988.

Lauritzen, S. L. and Wermuth, N. Graphical models for associations between variables, some of which are qualitative and some quantitative. *The annals of Statistics*, pp. 31–57, 1989.

Lerner, U. and Parr, R. Inference in hybrid networks: Theoretical limits and practical algorithms. *arXiv preprint arXiv:1301.2288*, 2013.

Lerner, U., Segal, E., and Koller, D. Exact inference in networks with discrete children of continuous parents. *arXiv preprint arXiv:1301.2289*, 2013.

Madsen, A. L. Belief update in clg bayesian networks with lazy propagation. *International Journal of Approximate Reasoning*, 49(2):503–521, 2008.

Madsen, A. L. and Jensen, F. V. Lazy propagation: a junction tree inference algorithm based on lazy evaluation. *Artificial Intelligence*, 113(1-2):203–245, 1999.

Mălan, A. Confidentiality-preserving collaborative bayesian networks. 2023.

Masegosa, A. R., Martınez, A. M., Langseth, H., Nielsen, T. D., Salmerón, A., Ramos-López, D., and Madsen, A. L. d-vmp: Distributed variational message passing. In *Conference on Probabilistic Graphical Models*, pp. 321–332. PMLR, 2016.

McNaught, K. and Chan, A. Bayesian networks in manufacturing. *Journal of Manufacturing Technology Management*, 22(6):734–747, 2011.

Morales, D., Agudo, I., and Lopez, J. Private set intersection: A systematic literature review. *Computer Science Review*, 49:100567, 2023.

Nannapaneni, S., Mahadevan, S., and Rachuri, S. Performance evaluation of a manufacturing process under uncertainty using bayesian networks. *Journal of Cleaner Production*, 113:947–959, 2016.

Paskin, M. Sample propagation. *Advances in Neural Information Processing Systems*, 16, 2003.

Pearl, J. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann, 1988.

Salmerón, A., Rumí, R., Langseth, H., Nielsen, T. D., and Madsen, A. L. A review of inference algorithms for hybrid bayesian networks. *Journal of Artificial Intelligence Research*, 62:799–828, 2018.

Scutari, M. and Denis, J.-B. *Bayesian networks: with examples in R*. Chapman and Hall/CRC, 2021.

Shenoy, P. P. and Shafer, G. Axioms for probability and belief-function propagation. In *Machine intelligence and pattern recognition*, volume 9, pp. 169–198. Elsevier, 1990.

Ypma, A., Koopman, A. C. M., and Middlebrooks, S. A. Methods and apparatus for obtaining diagnostic information, methods and apparatus for controlling an industrial process, January 21 2020. US Patent 10,539,882.

A. Results on Continuous Data

Table 5. Result on continuous data: mean squared errors.

Dataset		Ecoli70	Magic-Niab
Hybrid-CCJT		**0.070**	**0.287**
	3 States	1.411	0.521
Discretized BN	5 States	0.367	0.533
	10 States	0.368	N/A

Here, we measure the predictive accuracy of our method on purely continuous datasets, i.e., Ecoli70 and Magic-Niab and summarize the results in Table 5. Since this kind of datasets do not contain any categorical variable, we only run collaborative continuous inference as shown in Algorithm 2. Our baseline is the exact inference on discretized datasets with different levels of coarseness. Due to poor scalability of CCBNet and Δ-CCJT when dealing with discretized datasets, we use a centralized discrete network instead. This provides an upper bound on the predictive performance of a discretization-based approach. Under both experiments, we ran 10000 queries, with 4 parties and 10% overlap.

Across all experiments, `Hybrid CCJT` achieves the highest predictive accuracy. Under the Ecoli70 dataset, `Hybrid CCJT` attains an MSE of 0.07, which is five times lower than the optimal discrete counterpart and 20 times better than the non-optimal one. Under the Magic-Niab dataset, our model achieves an MSE of 0.287, outperforming the discretized counterpart, which has an MSE of 0.521. Furthermore, the discretized model failed to run inference in a reasonable amount of time with 10 states on Magic-Niab.

3
Background

3.1. Bayesian Networks

A Bayesian Network (BN) [5, 1] is a probabilistic graphical model represented as a directed acyclic graph (DAG) whose nodes are the random variables in the problem domain and whose edges correspond to direct influence of one variable on another [1]. The *conditional probability distribution* (CPD) of a variable given its parents -$P(x \mid \text{pa}(x))$- is called its *factor*. The table that summarizes all CPD for all variables is called CPD table. On the one hand, Bayesian Networks provide a semantics that enables a compact, declarative representation of a joint probability distribution, achieving better interpretability compared to other AI models such as deep neural networks [6]. On the other hand, many problems, including both exact inference and approximated inference, are proven to be NP-hard on such models [7, 8].

A model that is related to the Bayesian Network is the Markov Random Field (MRF). It is a probabilistic graphical model represented as an undirected graph [1]. A Bayesian Network can be transformed into a Markov Random Field via a process called moralization, which consists of removing directionality from all edges and connect all co-parents in the graph. An example of this process applied to the ASIA Bayesian Network is provided in Figures 3.1 and 3.2.

3.2. Inference on Bayesian Networks

Inference in Bayesian Networks involves computing the posterior distributions of certain variables given evidence about others [5]. This section introduces key concepts and methods relevant to inference in discrete Bayesian networks, including Conditional Probability Distribution (CPD) tables, Variable Elimination, Belief Propagation, and Junction Trees.

Given a Bayesian Network over variables $\mathcal{X}$, a set of observed variables $E \subset \mathcal{X}$ and a set of target variables $T \subseteq \mathcal{X} \setminus E$, The most common type of inference consists of computing the posterior

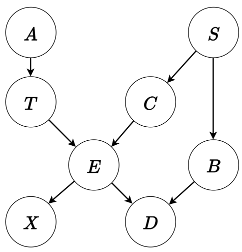


Figure 3.1: The ASIA Bayesian Network.

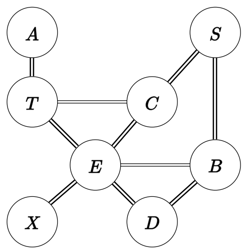


Figure 3.2: The ASIA Bayesian Network moralized.

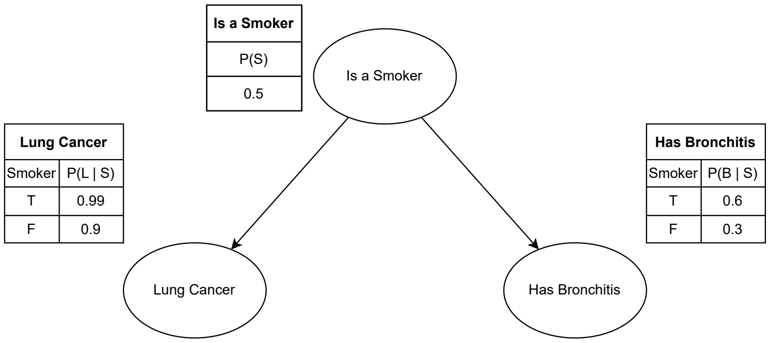


Figure 3.3: CPD tables taken from the ASIA network.

distribution $p(T|E)$ [3]. This, can be formulated as:

$$p(T|E) = \frac{p(T,E)}{p(E)} = \frac{\sum_{\mathcal{X} \in \Omega_{\mathcal{X} \setminus \{E \cup X_i\}}} p(\mathcal{X}, E)}{\sum_{\mathcal{X} \in \Omega_{\mathcal{X} \setminus E}} p(\mathcal{X}, E)} \tag{3.1}$$

Where $\Omega_{\mathcal{X}}$ is the set of all possible state combinations of variables in $\mathcal{X}$.

Most inference algorithms for Bayesian Networks require the usage of a *Conditional Probability Distribution Table*, also called CPD Table [1]. A CPD Table over a set of variables $\mathcal{X}$ contains the probability values of all state combinations for all variables in $\mathcal{X}$. An example of a CPD Table is shown in Figure 3.3. Since discrete variables have at least two possible different states, a CPD Table over $\mathcal{X}$ has size $\mathcal{O}(\Omega_{\mathcal{X}}) \sim \mathcal{O}(2^{|\mathcal{X}|})$. This gives an intuition on why both memory cost and computational cost can explode quickly when doing inference over these models.

3.2.1. Variable Elimination

Variable Elimination [1] is a fundamental algorithm for exact inference in Bayesian Networks. It computes the marginal probability of a target variable by systematically summing out or *marginalizing* the other variables in the network. The algorithm involves three key steps:

1. **Factorization**: Represent the joint probability distribution as a product of CPD tables.
2. **Summing Out Variables**: Sequentially eliminate variables not in the query or evidence by marginalizing them out, which involves summing over their values.
3. **Combination of Factors**: Multiply factors to produce intermediate results.

The order in which variables are eliminated significantly affects the computational efficiency of this algorithm. A poor elimination order can result in the creation of large intermediate factors, leading to exponential growth in computation time and memory requirements. Furthermore, representing the full joint probability distribution of a Bayesian Network requires $\mathcal{O}(2^N)$ space.

3.2.2. Belief Propagation and Junction Trees

Belief propagation [9] is an algorithm for inference in graphical models that leverages the network's structure to efficiently compute marginal probabilities. Belief propagation operates by passing messages between nodes in the network. Each node computes a local function based on incoming messages from its neighbors and sends updated messages back. However, in loopy networks (graphs with cycles), belief propagation becomes an iterative approximation algorithm, often called Loopy Belief Propagation.

A Junction Tree of a Bayesian Network over variables $\mathcal{X}$ with set of factors Φ is a computational graph whose nodes c_i, also called *cliques*, are tuples $(X_i \subseteq \mathcal{X}, \Phi_i \subseteq \Phi)$. Edges, also called *separators*, are associated with a set of variables called *sepset* $S_{i,j} = X_i \cap X_j$. These edges connect the cliques to form a tree. In order to be valid, a Junction Tree must satisfy the following rules:

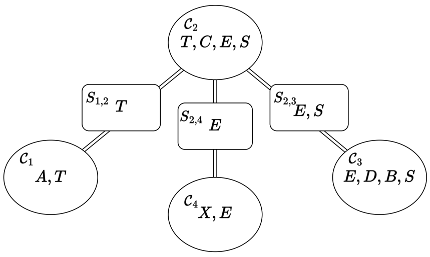


Figure 3.4: A Junction Tree constructed over the ASIA Bayesian Network.

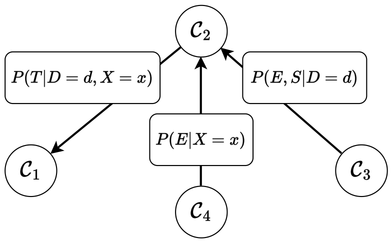


Figure 3.5: Belief Propagation toward clique $\mathcal{C}_1$ for query with evidence $X = x, D = d$.

- **Family preservation**: Each factor $\phi \in \Phi$ must be associated with one cluster c_i such that Scope$[\phi] \subseteq X_i$.

- **Running-intersection property**: For every pair of cliques c_i, c_j, every clique on the path between c_i, c_j contains $X_i \cap X_j$.

Junction Trees have the nice property that they can run belief propagation in linear time, even if the original network contained loops. One of the most popular message-passing schemes for Junction Trees is the sum-product scheme [10], where messages from node c_i to node c_k is computed as

$$\mu_{c_i \to c_j} = \sum_{v \notin S_{i,j}} \phi(c_i) \prod_{c_k \in \text{nb}(c_i) \setminus c_j} \mu_{c_k \to c_i}$$

Where $\sum$ represents the variable elimination operator, used to remove variables not in clique j before message passing. For tree-structured networks, belief propagation provides exact results in linear time [9]. While the cost of computing a message is still exponential, it remains bounded by the size of the largest clique in the tree.

As an example, let us consider the ASIA network (Figure 3.1) and the naively constructed Junction Tree (Figure 3.4). Here, running Variable Elimination requires $\mathcal{O}(2^8)$ operations to carry-out exact inference. In contrast, the cost of inference on the Junction Tree in Figure 3.4 is bounded by the size of the largest clique. Thus, requiring only $\mathcal{O}(2^4)$ operations.

Finding an optimal Junction Tree is an NP-hard problem. Although several heuristic algorithms exist, most of them require to moralize the original graph to identify patterns in it.

3.3. Hybrid CLG Bayesian Networks

So far we have seen how Bayesian Networks can be used to model and run inference over a set of discrete variables. However, some domains might require to model a problem as an union of discrete and continuous variables. In this scenario, *Hybrid Bayesian Networks* come in hand [3]. One of the most popular classes of hybrid models is the set of *Hybrid Conditional Linear Gaussian* (Hybrid CLG) distribution models [4]. In this class, continuous variables are Gaussian-shaped and cannot have any discrete child. The factor of a continuous variable $x \in \Gamma$ with discrete parents z_Δ and continuous parents z_Γ is given by:

$$f(x|\{z_\Delta, z_\Gamma\}) = \mathcal{N}(x; \alpha(z_\Delta) + \beta(z_\Delta)^T z_\Gamma, \sigma^2(z_\Delta))$$

where α and β are the coefficients that depend on the discrete state combination of z_Δ. If the state combination of z_Δ is fixed, x is Gaussian-shaped. If this is not the case, $f(x)$ is a mixture of $\mathcal{O}(2^{|\Delta|})$ Gaussian distributions. In general, even representing the correct marginal distribution in a hybrid CLG network require space that is exponential in the size of network [1]. Furthermore, even approximate inference for simple models structures such as polytrees is NP-hard in hybrid CLG networks [11].

3.3.1. Inference in Hybrid Bayesian Networks

Similarly to the discrete domain, we can define a query over a Hybrid Bayesian Network as:

$$p(x_i|\mathbf{x}_E) = \frac{p(x_i, \mathbf{x}_E)}{p(\mathbf{x}_E)} = \frac{\sum_{\Delta \in \Omega_{\Delta \setminus \{x_i\}}} \int_{\Gamma \in \Omega_{\Gamma \setminus \{x_i\}}} p(\mathbf{x}, \mathbf{x}_E) d\Gamma}{\sum_{\Delta \in \Omega_{\Delta}} \int_{\Gamma \in \Omega_{\Gamma}} p(\mathbf{x}, \mathbf{x}_E) d\Gamma} \tag{3.2}$$

Where $\mathbf{X}_E$ is the set of observed variables, Γ is the set of continuous variables and Δ is the set of discrete variables. In principle, the challenge we have to deal with when executing inference on Hybrid models, is that variables in $\Gamma \setminus \mathbf{X}_E$ need to be integrated out.

In order to make hybrid CLG inference tractable, most works focus on finding the first two moments of continuous distributions instead of the full marginal probability. The first attempt at developing an exact method over junction trees following this approach was introduced by Lauritzen [12] and later revised by Lauritzen and Jensen [13]. This algorithm is able to perform exact inference in hybrid BNs, as long as the joint distribution is a CLG. In order to achieve such result, we need to define a strong elimination order [14]. That is, an order such that continuous nodes are eliminated from the graph before discrete ones. Since regular Junction Trees do not always have a strong elimination order [14], they cannot be used to run inference correctly on Hybrid CLG models. Instead, a stricter structure called *Strongly-Rooted Junction Tree* [12] is required. A Junction Tree is said to be strongly-rooted if it has a distinguished clique R, called *strong root*, such that for every couple of neighboring cliques (C, D), with C being closer to R than D, it holds that: $(C \cap D \subseteq \Delta) \vee (D \setminus C \subseteq \Gamma)$. Meaning that no discrete variable is marginalized before any other CLG one in the factor.

Madsen's 2008 work [14] introduces an improved version of Lauritzen's algorithm where belief messages are now represented as set of potentials and resolve dependencies between continuous variables by running arc-reversal operations on the DAG and leveraging Lazy Propagation [15] rules to optimize inference computations. These two approaches require complex manipulation of variable potentials which cannot be done in a straight-forward manner without breaking confidentiality constraints. A significant disadvantage of CLG models is that they do not allow for categorical nodes to be children of continuous variables [4]. Lerner introduced *augmented CLG networks* in [11], where they build an algorithm on top of the work of Lauritzen [13] to run approximate inference in CLG models where continuous variables are allowed to have discrete children.
Other works only allow for a subset of inference queries. For instance, Paskin [16] leverages Rao-Blackwellization to provide tractable approximated inference but does not allow for queries with continuous evidence.

3.4. Distributed Bayesian Networks

Studies on applications of Belief Propagation on distributed domains include the publications by Xia et al. [17, 18] which focus on computation of message passing algorithms in concurrent systems without focusing on the privacy constraints imposed by our problem. Similarly, Stefanovitch et al. [19] also explores computational optimization rather than privacy in a multi-agent system setting with constraint communication and computational capabilities. Other works use Belief Propagation for privacy-safe collaborative filtering purposes [20, 21], but they require all parties to share the same set of nodes. Works by Jo et al. [22] and Xu et al. [23] explore distributed computation of belief propagation and moment sharing respectively. Although, these two methods require large communication costs due to their iterative nature. Finally, Kearns et al. [24] studied privacy-preserving applications of both Belief propagation and Gibbs sampling. Nonetheless, they don't provide a protocol to run such inference.

Masegosa et al. [25, 26] explored applications of hybrid *Conjugate Exponential Family* BNs [3] within distributed systems. They provided a method to generate a centralized hybrid network from different parties' models. Albeit, structure and parameters of such generated network may leak confidential knowledge.

3.4.1. CCBNet

CCBNet [2] is a pioneer work in the field of collaborative confidential inference for Bayesian Networks as well as the current state-of-the-art in its field. It is based on two protocols: `CABN` (Confidentially Augmented Bayesian Networks) and `SAVE` (Share Aggregation Variable Elimination). `CABN` is a protocol that augments probability distributions for features across parties into secret shares of their normalized combination. It follows a geometric mean merging procedure inspired by the works of Del Sagrado [27] and Feng [28]. `SAVE` is an inference protocol based on Variable Elimination. It requires parties to run said algorithm collaboratively by computing partial inference results over common variables and merge them via a secret sharing scheme for multiplication [29]. Despite being a significant step forward in the Confidential BN literature, CCBNet falls short under several aspects. Namely, its communication costs have been shown to explode even for relatively small problems, it only supports discrete variables, and posterior probability values for some private variables from the peers are revealed to the party executing a query.

4

Additional Experiments and Results

4.1. Results on Continuous Data

We evaluate the predictive accuracy of our method on purely continuous datasets, specifically Ecoli70 [30] and Magic-Niab [31]. The results are summarized in Table 4.1. Since these datasets do not include categorical variables, we exclusively perform collaborative continuous inference as described in Algorithm 3 in the paper.

Our baseline is exact inference applied to discretized datasets with varying levels of granularity. However, due to the limited scalability of CCBNet and our own purely discrete baseline Δ-CCJT when processing discretized data, we use a centralized discrete network instead. This serves as an upper bound for the predictive performance achievable with a discretization-based approach.

For all experiments, we execute 10,000 queries, involving four parties with a 10% overlap. We measure both the mean squared error (MSE), as well as the accuracy in Maximum A Posteriori (MAP) accuracy. With regards to the latter metric when used on the hybrid method, we measure wether the mean of the posterior of the targeted variable falls in the relative discrete bin.
Across all evaluations, Hybrid CCJT consistently delivers the highest predictive accuracy. For the Ecoli70 dataset, Hybrid CCJT achieves an MSE of 0.07, which is five times lower than the optimal discrete counterpart and 20 times better than the non-optimal version. On the Magic-Niab dataset, our method achieves an MSE of 0.287, outperforming the discretized model, which records an MSE of 0.521.

Furthermore, Hybrid CCJT consistently outperforms the baseline with respect with MAP accuracy. Additionally, the discretized model could not complete inference within a reasonable time when using 10 states on the Magic-Niab dataset.

Table 4.1: Results on continuous data: mean squared errors and maximum a posterior accuracy.

Dataset	**Ecoli70**						**Magic-Niab**			
Metric	Brier			MAP			Brier		MAP	
#States	3 States	5 States	10 States	3 States	5 States	10 States	3 States	5 States	3 States	5 States
Hybrid -CCJT	0.070			0.849	0.67	0.40	0.287		0.895	0.704
Discretized BN	1.411	0.367	0.368	0.619	0.604	0.389	0.521	0.533	0.0448	0.1876

Dataset		Child			Alarm			Insurance		Andes	Link	Munin #2
#Parties		2	4	8	2	4	8	2	4	16	64	128
Hybrid CCJT	Comm	14	33	58	15	40	79	17	43	6080	459	735
	Brier	0.023	0.031	0.038	0.010	0.022	0.041	0.028	0.051	0.046	0.0124	0.017
CCBNet	Comm	67	157	429	166	1959	1886	143	1835	23175	4455	243474
	Brier	0.023	0.031	0.038	0.011	0.022	0.041	0.028	0.052	0.046	0.126	0.016
CCBNetJ	Comm	18	36	31	51	74	61	23	57	305	2292	89008
	Brier	0.023	0.031	0.038	0.011	0.022	0.041	0.028	0.052	0.046	0.126	0.016

Table 4.2: Results on discrete data: communication costs and predictive accuracy (Brier score). Lower is better for all.

4.2. Results on Discrete Data

Given the significant scalability challenges posed by communication costs in CCBNet [2], we focus on evaluating the improvements achieved by Hybrid CCJT in this aspect when applied to purely discrete datasets. We further compare our model with CCBNetJ, a degenerate model of CCBNet introduced by [2] that stores the fully combined central CPDs for overlaps in one of the concerned parties, trading some safety for faster inference. We conduct experiments on the Child, Alarm, and Insurance datasets with varying numbers of parties, ranging from 2 to 8. Additionally, we test larger datasets with up to 128 parties. Each experiment involves 2,000 queries. The results are presented in Table 4.2. For these datasets, which contain no continuous variables, Hybrid CCJT simplifies to Δ-CCJT, as already discussed in the paper.

In smaller experiments involving only two parties, Hybrid CCJT reduces CCBNet's communication costs by a factor of 8, lowering the average number of communicated values from 125 to 15. The improvement becomes even more pronounced as the number of parties increases. In larger-scale experiments, such as the Munin dataset with up to 128 parties, CCBNet's communication costs escalate substantially, reaching up to 243k transmitted values. By contrast, Hybrid CCJT maintains a low communication overhead, transmitting only 735 values, representing a 331-fold improvement.

CCBNetJ manages to improve communication costs with respect with CCBNet. Bringing performance that are sometimes comparable with our model. Despite these considerations and the significant confidentiality trade-offs associated with this model, it fails to achieve scalability in very large discrete networks. In such cases, Hybrid CCJT continues to outperform the baseline by a substantial margin.

Importantly, we observe no significant differences in predictive accuracy throughout these experiments. Altogether, this shows how the Junction Tree approach is a desirable solution even when dealing with purely discrete scenarios due its superior scalability compared to approaches based on Variable Elimination.

4.3. Insights on the Hybrid Datasets

Inference complexity in purely discrete and purely continuous datasets is usually determined by factors such as the number of nodes, arcs, and states per variable. However, for hybrid models, it can be difficult to evaluate how computationally demanding or challenging the inference process will be. Gaining a clear understanding often requires deeper insights into the model's structure and behavior.

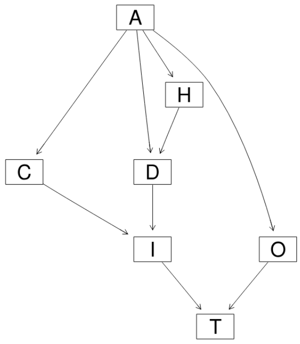


Figure 4.1: The Healthcare network.

4.3.1. Healthcare

The Healthcare dataset [32] (Figure 4.1) consists of only seven variables: three discrete (`A`, `H`, `C`) and four continuous. Despite its small size (7 nodes) it has more than twice the number of parameteres in the ASIA network (Figure 3.1). This highlights the increased complexity involved in representing a hybrid network compared to a purely discrete one.

The discrete variables in the Healthcare dataset have a maximum of three states, which helps keep the parameter count relatively manageable compared to other hybrid models. On the other hand, the continuous variables exhibit an average variance exceeding $5 \cdot 10^5$.

4.3.2. Sangiovese

The Sangiovese dataset [33] (Figure 4.2) has been used to assess the impact of several agronomic settings on the quality of Tuscan Sangiovese grapes. It has 15 variable of which only one is discrete (`Treatment`, with 16 different states). It has a total of 259 parameters, a relatively low number given the network size, gained thanks to the fact that it only has one discrete variable.

Thanks to the low number of discrete state combinations, and the low variance of each continuous variable, this network has an average variance of only slightly over 0.083.

4.3.3. Mehra

The Mehra dataset [34] (Figure 4.3) is a hybrid network used to model conditionality between air pollution, climate, and health data in several regions of England. Despite having only 24 variables, it is the largest network in terms of parameters by far, with more than four times the parameters of the Munin network (which contains more than 1000 nodes). This network gives a clear example of how computationally heavy a hybrid network can get with a small number of variables modeled.

The size of the network is given by both the amount of discrete variables (8) and the high amount of states per variable (up to 31). Along with high variances of Gaussian distribution modeled by the CLG node, this brings a in average variance of over $2.4 \cdot 10^{16}$. The highest value seen so far, which also explains the high results in MSE over this dataset.

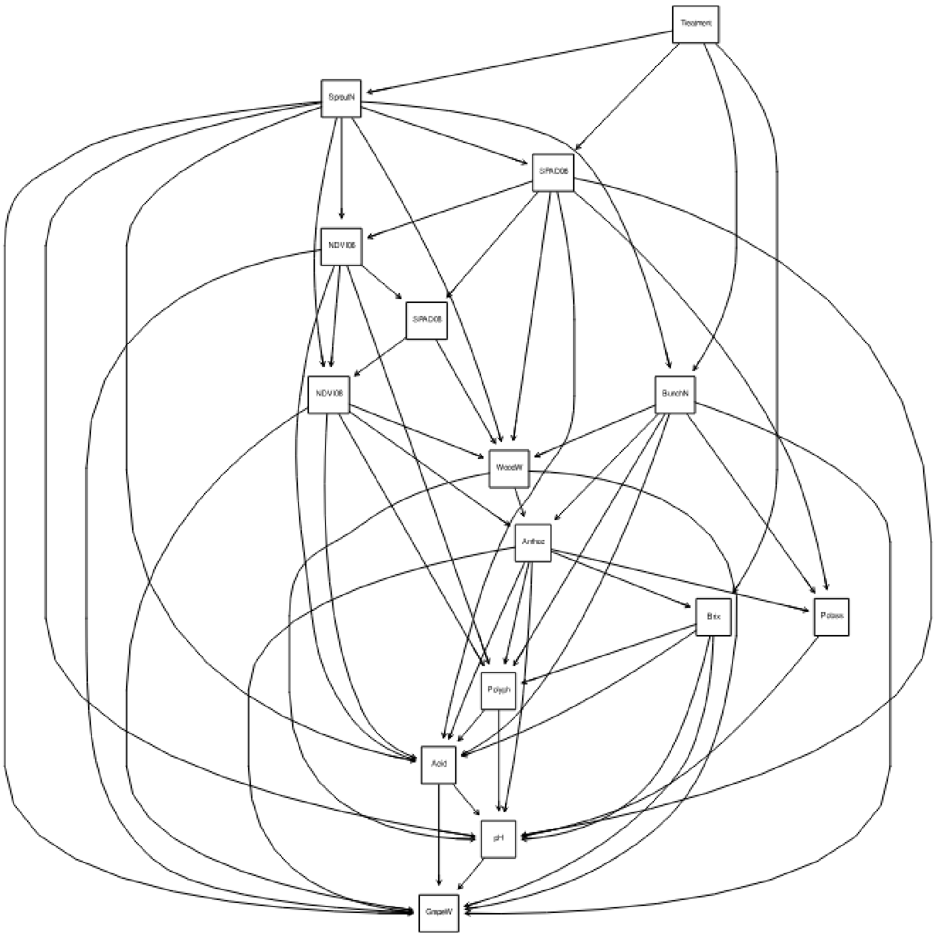


Figure 4.2: The Sangiovese network.

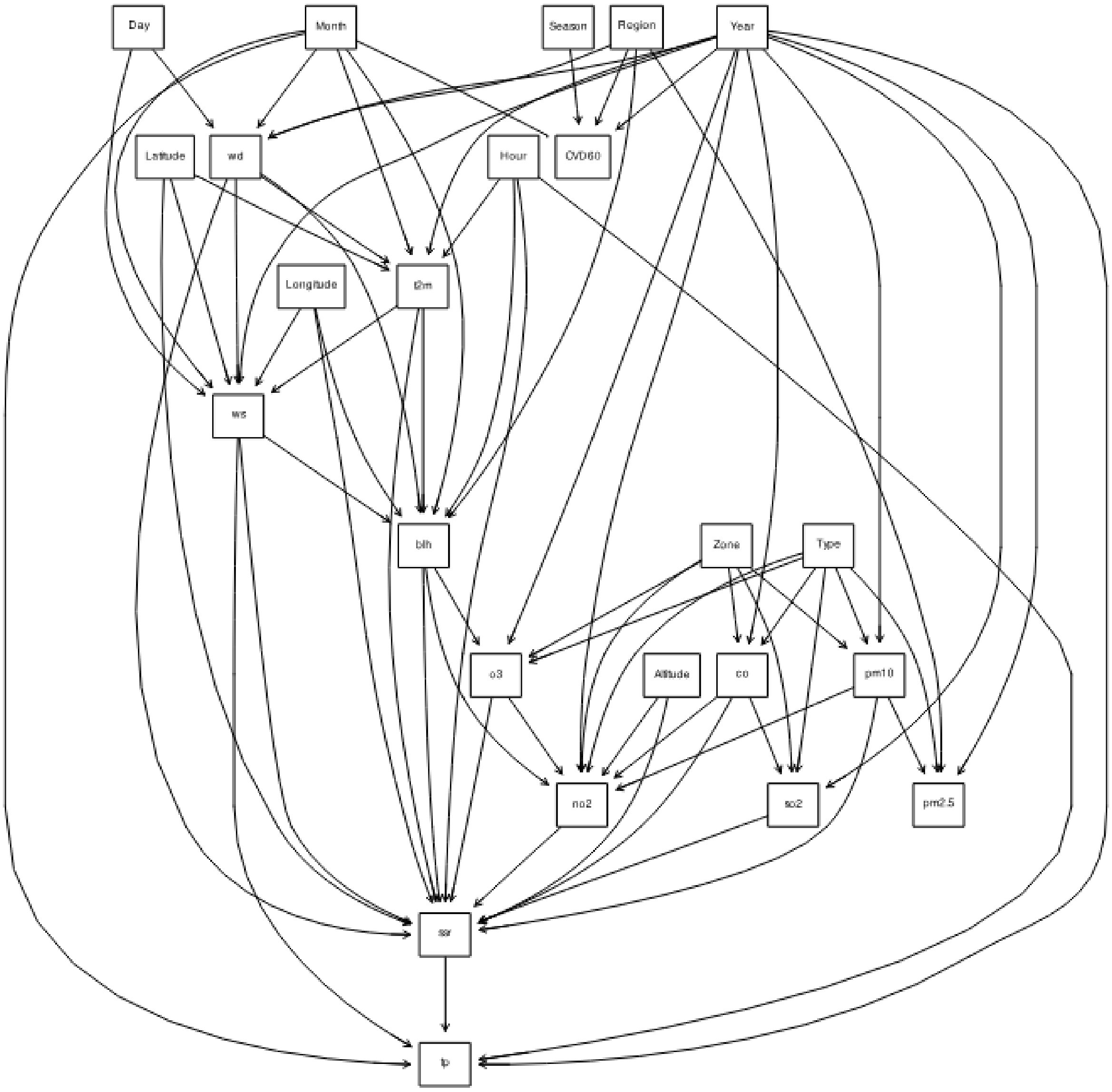


Figure 4.3: The Mehra network.

5

Conclusions

The answers to the initial research questions and solution identified by this thesis are:

1. How can we run Hybrid CLG inference in a distributed and privacy preserving fashion?

   Hybrid CLG inference can be run in a distributed and privacy-preserving fashion by leveraging the strongly-rooted Junction Tree properties. By defining a strong variable elimination order and a federated method for merging parties beliefs, we achieve accurate inference quality while maintaining parties' privacy on 12 data sets. When dealing with hybrid data, our method improves the predictive accuracy of continuous target variables by an average of 32% in Mean Squared Error (MSE) compared to the best performing baseline. Our method also maintain predictive performances comparable to the state-of-the-art when targeting discrete variables.

2. How can we reduce communication costs of collaborative inference over discrete variables?

   By leveraging properties of Belief Propagation and the Shafer-Shanoy message-passing scheme over trees, we marginalize overhead categorical variables before transmitting information between parties. This leads to a significant reduction in communication costs. Our method outperforms existing baselines in communication costs, with up to 40-fold reductions in large hybrid datasets and 331-fold improvements in large-scale experiments on purely discrete data.

3. How can we avoid revealing posterior of private variables in multi-party Bayesian Networks?

   By leveraging properties of Belief Propagation and the Shafer-Shanoy message-passing scheme over trees, we manage to marginalize private categorical variables before transmitting information between parties. This prevents the querying party from reverse computing posteriors of private variables owned by other parties.

Bibliography

[1] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[2] Abel Mălan. “Confidentiality-Preserving Collaborative Bayesian Networks”. In: (2023).

[3] Antonio Salmerón et al. “A review of inference algorithms for hybrid Bayesian networks”. In: *Journal of Artificial Intelligence Research* 62 (2018), pp. 799–828.

[4] Steffen L Lauritzen and Nanny Wermuth. “Graphical models for associations between variables, some of which are qualitative and some quantitative”. In: *The annals of Statistics* (1989), pp. 31–57.

[5] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann, 1988.

[6] Bojan Mihaljević, Concha Bielza, and Pedro Larrañaga. “Bayesian networks for interpretable machine learning and optimization”. In: *Neurocomputing* 456 (2021), pp. 648–665.

[7] Paul Dagum and Michael Luby. “Approximating probabilistic inference in Bayesian belief networks is NP-hard”. In: *Artificial intelligence* 60.1 (1993), pp. 141–153.

[8] Gregory F Cooper. “The computational complexity of probabilistic inference using Bayesian belief networks”. In: *Artificial intelligence* 42.2-3 (1990), pp. 393–405.

[9] Steffen L Lauritzen and David J Spiegelhalter. “Local computations with probabilities on graphical structures and their application to expert systems”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 50.2 (1988), pp. 157–194.

[10] Prakash P Shenoy and Glenn Shafer. “Axioms for probability and belief-function propagation”. In: *Machine intelligence and pattern recognition*. Vol. 9. Elsevier, 1990, pp. 169–198.

[11] Uri Lerner and Ron Parr. “Inference in hybrid networks: Theoretical limits and practical algorithms”. In: *arXiv preprint arXiv:1301.2288* (2013).

[12] Steffen L Lauritzen. “Propagation of probabilities, means, and variances in mixed graphical association models”. In: *Journal of the American Statistical Association* 87.420 (1992), pp. 1098–1108.

[13] Steffen L Lauritzen and Frank Jensen. “Stable local computation with conditional Gaussian distributions”. In: *Statistics and Computing* 11 (2001), pp. 191–203.

[14] Anders L Madsen. “Belief update in CLG Bayesian networks with lazy propagation”. In: *International Journal of Approximate Reasoning* 49.2 (2008), pp. 503–521.

[15] Anders L Madsen and Finn V Jensen. “Lazy propagation: a junction tree inference algorithm based on lazy evaluation”. In: *Artificial Intelligence* 113.1-2 (1999), pp. 203–245.

[16] Mark Paskin. “Sample propagation”. In: *Advances in Neural Information Processing Systems* 16 (2003).

[17] Yinglong Xia and Viktor K Prasanna. “Junction tree decomposition for parallel exact inference”. In: *2008 IEEE International Symposium on Parallel and Distributed Processing*. IEEE. 2008, pp. 1–12.

[18] Yinglong Xia and Viktor K Prasanna. “Distributed evidence propagation in junction trees on clusters”. In: *IEEE Transactions on Parallel and Distributed Systems* 23.7 (2011), pp. 1169–1177.

[19] Nicolas Stefanovitch et al. “Resource-aware junction trees for efficient multi-agent coordination”. In: (2011).

[20] Jun Zou, Arash Einolghozati, and Faramarz Fekri. “Privacy-preserving item-based collaborative filtering using semi-distributed belief propagation”. In: *2013 IEEE Conference on Communications and Network Security (CNS)*. IEEE. 2013, pp. 189–197.

[21] Jun Zou and Faramarz Fekri. “A belief propagation approach to privacy-preserving item-based collaborative filtering”. In: *IEEE Journal of Selected Topics in Signal Processing* 9.7 (2015), pp. 1306–1318.

[22] Saehan Jo, Jaemin Yoo, and U Kang. “Fast and scalable distributed loopy belief propagation on real-world graphs”. In: *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 2018, pp. 297–305.

[23] Minjie Xu et al. “Distributed Bayesian posterior sampling via moment sharing”. In: *Advances in neural information processing systems* 27 (2014).

[24] Michael Kearns, Jinsong Tan, and Jennifer Wortman. “Privacy-preserving belief propagation and sampling”. In: *Advances in Neural Information Processing Systems* 20 (2007).

[25] Andrés R Masegosa et al. “d-VMP: Distributed variational message passing”. In: *Conference on Probabilistic Graphical Models*. PMLR. 2016, pp. 321–332.

[26] Andrés R Masegosa et al. “Scaling up Bayesian variational inference using distributed computing clusters”. In: *International Journal of Approximate Reasoning* 88 (2017), pp. 435–451.

[27] José Del Sagrado and Serafin Moral. “Qualitative combination of Bayesian networks”. In: *International Journal of Intelligent Systems* 18.2 (2003), pp. 237–249.

[28] Guang Feng, Jia-Dong Zhang, and Stephen Shaoyi Liao. “A novel method for combining Bayesian networks, theoretical analysis, and its applications”. In: *Pattern Recognition* 47.5 (2014), pp. 2057–2069.

[29] Niki Kilbertus et al. “Blind justice: Fairness with encrypted sensitive attributes”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 2630–2639.

[30] Juliane Schäfer and Korbinian Strimmer. “A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics”. In: *Statistical applications in genetics and molecular biology* 4.1 (2005).

[31] Marco Scutari et al. “Multiple quantitative trait analysis using Bayesian networks”. In: *Genetics* 198.1 (2014), pp. 129–137.

[32] Marco Scutari and Jean-Baptiste Denis. *Bayesian networks: with examples in R*. Chapman and Hall/CRC, 2021.

[33] Alessandro Magrini, Stefano Di Blasi, and Federico Mattia Stefanini. “A conditional linear Gaussian network to assess the impact of several agronomic settings on the quality of Tuscan Sangiovese grapes”. In: *Biometrical Letters* 54.1 (2017), pp. 25–42.

[34] Claudia Vitolo et al. “Modeling air pollution, climate, and health data using Bayesian Networks: A case study of the English regions”. In: *Earth and Space Science* 5.4 (2018), pp. 76–88.