Second-Order Learning and Inference using Incomplete Data for Uncertain Bayesian Networks: A Two Node Example

Lance Kaplan[1] Federico Cerutti[2,3] Murat Şensoy[4] Kumar Vijay Mishra[1]
[1]CCDC Army Research Lab, Adelphi, MD, USA [2]University of Brescia, Brescia, Italy
[3]Cardiff University, Cardiff, UK [4]Blue Prism AI Labs, London, UK

***Abstract*—Efficient second-order probabilistic inference in uncertain Bayesian networks was recently introduced. However, such second-order inference methods presume training over complete training data. While the expectation-maximization framework is well-established for learning Bayesian network parameters for incomplete training data, the framework does not determine the covariance of the parameters. This paper introduces two methods to compute the covariances for the parameters of Bayesian networks or Markov random fields due to incomplete data for two-node networks. The first method computes the covariances directly from the posterior distribution of parameters, and the second method more efficiently estimates the covariances from the Fisher information matrix. Finally, the implications and effectiveness of these covariances is theoretically and empirically evaluated.**

I. Introduction

Situational understanding is paramount for informing decision makers. For example, observations about the health and social interactions of citizens in a region of interest can help to determine the probability that the distribution of certain aid will be effective in a disaster relief scenario. Probabilistic graphical models (PGMs) help handling the complexity of fusing such observations of interrelated random variables. The structure in PGMs encodes the conditional dependencies between the variables; we assume that such a structure is known. For instance, in our scenario, it is reasonable to believe that years of experience distributing aid throughout the world lead to proven socio-demographic models. Nevertheless, parameters within the models must be learned for each area of interest as the local population can exhibit specific social norms unique to that area. This means that machine learning is required to estimate the parameters from training data, and when such data are limited, the model parameters will not be known precisely (*epistemic uncertainty*), and this can lead to uncertainty on the probabilities inferred for variables of interest for decision making (*aleatoric uncertainty*).

The PGMs are represented as either Markov random fields (undirected graphs) parameterized by potentials composing the joint probability mass or density or as Bayesian networks parameterized by conditional probabilities. While model (epistemic) uncertainty may be incorporated in the structural and parameter learning, the inference process typically presumes precise conditional probabilities. A survey of Bayesian network research with imprecise conditional probability tables is provided in [1]. Valuation-based systems and credal networks have been investigated as a means to propagate impression of the conditional probabilities into imprecision of the inferred probabilities [2], [3]. From a Bayesian perspective, the likelihood of the training data leads to a posterior distribution for the parameters and recent methods have incorporated second-order probabilities, i.e., distributions on probabilities, to enable the propagation of these distributions into a distribution for the inferred state probabilities [4], [5]. These second-order inference methods assume that the distributions for the different conditional probabilities are statistically independent, which holds if all variables are observed over all instantiations of the network over a window of time.

It is impractical to assume that the training data is complete. It is expected that in the historical data, certain variables cannot be observed at certain times. While the expectation-maximization (EM) framework has classically been used to learn conditional probabilities for Bayesian networks with incomplete training data [6], [7], [8], to the best of our knowledge, there is no work to address the posterior distribution of conditional probabilities learned from incomplete training data. This paper provides a first glance by determining the covariance of the conditional probabilities for second-order inference over a binary two-node probabilistic network.

The rest of the paper is organized as follows. We provide the desiderata for Bayesian networks in the next section. In Section III, we determine the covariance matrix for the model parameters first using the direct covariance calculation using the posterior for the conditional probabilities (Section III-A) and then a more computationally efficient estimation through the inversion of the Fisher information matrix (FIM) (Section III-B). We consider both Bayesian network (directed) and Markov random field (undirected) interpretations of the joint probability mass and demonstrate the limits of incomplete data to reduce epistemic uncertainty in the inferred probabilities. Finally, in Section IV, we demonstrate through numerical experiments that both the posterior and Fisher analyses accurately characterize the confidence in the inferred probabilities, with the Fisher analysis being much less computationally demanding than the posterior analysis. We conclude in Section V.

II. Preliminaries

We summarize the concept of PGMs and Bayesian networks here.

A. Dirichlet Distribution

The probabilities $\boldsymbol{\theta} = \left(\theta_1, \ldots, \theta_{K-1}, 1 - \sum_{i=1}^{K-1} \theta_i\right)$ that a particular variable X will take on one of K state values in the alphabet $\mathbb{X}$ can be learned by observing a set of independent realizations of X. Note that $\boldsymbol{\theta} \in \mathcal{S}_K$ meaning that the K elements of $\boldsymbol{\theta}$ are all non-negative and sum up to one. Assuming a uniform (uninformative) prior for these probabilities, it is well known that the posterior distribution for these probabilities is Dirichlet distributed,

$$f_\beta(\boldsymbol{\theta}; \boldsymbol{\alpha}) = \frac{(\theta_1)^{\alpha_1 - 1} \cdots (\theta_{K-1})^{\alpha_{K-1}-1} \left(1 - \sum_{i=1}^{K-1} \theta_i\right)^{\alpha_K - 1}}{B(\boldsymbol{\alpha})}, \tag{1}$$

over $\mathcal{S}_K$ where $B(\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_1) \cdot \cdots \cdot \Gamma(\alpha_K)}{\Gamma(\alpha_1 + \cdots + \alpha_K)}$ is the K-dimensional beta function, $\Gamma(\cdot)$ is the Gamma function, and $\boldsymbol{\alpha} = (n_1 + 1, \cdots, n_K + 1)$ are the Dirichlet parameters representing the number of times a sample takes on one of the K possible values, i.e., n_i for $i = 1, \ldots, K$. Note that,

$$B(\boldsymbol{\alpha}) = \int_{\mathcal{S}_k} (\theta_1)^{\alpha_1 - 1} \cdots (\theta_{K-1})^{\alpha_{K-1}-1} (1 - \theta_1 - \cdots - \theta_{K-1})^{\alpha_K - 1} d\boldsymbol{\theta}. \tag{2}$$

The Dirichlet strength $S_\alpha = \alpha_1 + \cdots + \alpha_K$ represents the total number of observations to determine the distribution. It is inversely proportional to the epistemic uncertainty and represents the precision of the distribution. When $K = 2$, the Dirichlet distribution simplifies to the well known beta distribution. When viewing each θ_i in isolation, it is beta distributed with $\boldsymbol{\alpha} = (\alpha_i, S_\alpha - \alpha_i)$. The means and variances for θ_i are given by

$$m_i = \frac{\alpha_i}{S_\alpha} \quad \text{and} \quad \sigma_i^2 = \frac{m_i(1 - m_i)}{S_\alpha + 1}, \tag{3}$$

respectively.

B. Probabilistic Graphical Models

The PGMs encode the conditional statistical relations between variables to represent the joint probability mass function (pmf) of all the variables. The graphical structure simplifies the inference of latent variables in light of the evidence, i.e., the values of the observed variables. When the graphs are undirected, the pmfs can be represented as a product of potentials for each clique. For an undirected two-node graph X—Y, the potential is simply the values of the pmf that is completely parameterized by the parameters $\boldsymbol{\theta}_u = (\theta_{yx}, \theta_{y\bar{x}}, \theta_{\bar{y}x})$ representing the joint probabilities that $(X = x, Y = y)$, $(X = \bar{x}, Y = y)$ and $(X = x, Y = \bar{y})$, respectively. All other probabilities, joint or marginal, are derived as a function of these three parameters, e.g., $p_{\bar{y}\bar{x}} = h_{\bar{y}\bar{x}}(\boldsymbol{\theta}_u)$ such that $h_{\bar{y}\bar{x}}(\boldsymbol{\theta}_u) = 1 - \theta_{yx} - \theta_{y\bar{x}} - \theta_{\bar{y}x}$. In general, $h(\boldsymbol{\theta}_u)$ is the functions that computes $p_.$ from the three parameters $\boldsymbol{\theta}_u$ of the undirected model.

TABLE I

Various $h_.$ functions to compute probabilities $p_.$ from the model parameter $\boldsymbol{\theta}_r$ of the $X \to Y$ Bayesian network.

$h_{yx}(\boldsymbol{\theta}_r) = \theta_{y\vert x}\theta_x$	$h_y(\boldsymbol{\theta}_r) = \theta_{y\vert x}\theta_x + \theta_{y\vert\bar{x}}(1 - \theta_x)$
$h_{y\bar{x}}(\boldsymbol{\theta}_r) = \theta_{y\vert\bar{x}}(1 - \theta_x)$	$h_{\bar{y}}(\boldsymbol{\theta}_r) = (1 - \theta_{y\vert x})\theta_x + (1 - \theta_{y\vert\bar{x}})(1 - \theta_x)$
$h_{\bar{y}x}(\boldsymbol{\theta}_r) = (1 - \theta_{y\vert x})\theta_x$	$h_x(\boldsymbol{\theta}_r) = \theta_x$
$h_{\bar{y}\bar{x}}(\boldsymbol{\theta}_r) = (1 - \theta_{y\vert\bar{x}})(1 - \theta_x)$	$h_{\bar{x}}(\boldsymbol{\theta}_r) = 1 - \theta_x$
$h_{y\vert x}(\boldsymbol{\theta}_r) = \theta_{y\vert x}$	$h_{x\vert y}(\boldsymbol{\theta}_r) = \frac{\theta_{y\vert x}\theta_x}{\theta_{y\vert x}\theta_x + \theta_{y\vert\bar{x}}(1-\theta_x)}$
$h_{y\vert\bar{x}}(\boldsymbol{\theta}_r) = \theta_{y\vert\bar{x}}$	$h_{x\vert\bar{y}}(\boldsymbol{\theta}_r) = \frac{(1-\theta_{y\vert x})\theta_x}{(1-\theta_{y\vert x})\theta_x + (1-\theta_{y\vert\bar{x}})(1-\theta_x)}$

Bayesian networks are directed acyclic graphs with nodes representing random variables whose joint probability distributions is the product of the probability of each variable conditioned on the values of its parent variables. For two variables X and Y, the two Bayesian networks $X \to Y$ and $X \leftarrow Y$ are equivalent representations for the joint probability mass function of the binary valued variables. For the former, the parameters are uniquely specified by the three values $\boldsymbol{\theta}_r = (\theta_x, \theta_{y|x}, \theta_{y|\bar{x}})$. Likewise for the latter, the parameters are specified by $\boldsymbol{\theta}_l = (\theta_y, \theta_{x|y}, \theta_{x|\bar{y}})$. The function $h_.(\boldsymbol{\theta}_d)$ such that $d \in \{u, r, l\}$ computes a probability $p_.$ from the parameters for one of three network parameters as determined by d. For example,

$$p_{x|y} = h_{x|y}(\boldsymbol{\theta}_r) = \frac{\theta_{y|x}\theta_x}{\theta_{y|x}\theta_x + \theta_{y|\bar{x}}(1 - \theta_x)}.$$

Different $h_.$ functions—in the following we use h instead of $h_.$ for ease of reading—are provided in Table I for $\boldsymbol{\theta}_r$. The vector functions $\mathbf{h}_{d',d}(\boldsymbol{\theta}_d)$ simply transforms $\boldsymbol{\theta}_d$ into $\boldsymbol{\theta}_{d'}$ for $d', d \in \{u, r, l\}$.

Usually, it is assumed the parameters for the PGMs are known precisely. For these cases, the inferences are simply the computations of different h functions to compute the probabilities for the values of latent variables conditioned on the evidence formed by the observed variable values. Various inference methods such as variable elimination [9], belief propagation [10] and junction-tree [11] can precisely compute these probability values. Note that the more efficient belief propagation provides precise inference, but only over poly-tree network structures. At inference time, it is assumed the PGM parameters $\boldsymbol{\theta}_d$ are known. These parameters are either provided by a domain expert based upon his/her experience (i.e., epistemic domain knowledge) or learnt from training/historical data. Here, we assume that the domain knowledge to derive $\boldsymbol{\theta}_d$ is evidence driven by the data one way or the other.

In general, the parameters can be determined as the maximum a posterior (MAP) estimate from the observed training data $\mathbf{t}_o$ [6], i.e.,

$$\hat{\boldsymbol{\theta}}_d = \arg\max_{\boldsymbol{\theta}_d} \log\left((P(\mathbf{t}_o; \boldsymbol{\theta}_d) f(\boldsymbol{\theta}_d)\right), \tag{4}$$

where $f(\boldsymbol{\theta})$ is a prior distribution for the parameters that can naturally be modeled as Dirichlet distributed. For instance,

$$f(\boldsymbol{\theta}_r) = f_\beta(\boldsymbol{\theta}_x; (\alpha_x, \alpha_{\bar{x}})) f_\beta(\boldsymbol{\theta}_{y|x}; (\alpha_{y|x}, \alpha_{\bar{y}|x})) f_\beta(\boldsymbol{\theta}_{y|\bar{x}}; (\alpha_{y|\bar{x}}, \alpha_{\bar{y}|\bar{x}})). \tag{5}$$

For complete data, (4) is simple to compute as the logarithm term can be decomposed as the sum of the logarithm of the individual elements of $\boldsymbol{\theta}_d$. For incomplete data, however, latent variables do not enable the simple decomposition. The EM algorithm [6] provides an iterative framework to enable such simplified decompositions at each step. As such, $\mathbf{t}_o$ is augmented with the latent data $\mathbf{T}_\ell$ that is complicating the maximization process. Note that the possible values of $\mathbf{T}_\ell$ are $\mathbf{t}_\ell \in \mathbb{T}_\ell$. Given the estimate for the parameters at step t $\boldsymbol{\theta}_d^{(t)}$, the expectation step determines a Q-function as

$$Q(\boldsymbol{\theta}_d;\boldsymbol{\theta}_d^{(t)}) = \sum_{\mathbf{t}_\ell \in \mathbb{T}_\ell} \log\left(P(\mathbf{t}_o,\mathbf{t}_\ell;\boldsymbol{\theta}_d) f(\boldsymbol{\theta}_d)\right) P(\mathbf{t}_\ell|\mathbf{t}_o;\boldsymbol{\theta}_d^{(t)}). \tag{6}$$

The maximization step determines the update parameters as

$$\boldsymbol{\theta}_d^{(t+1)} = \arg\max_{\boldsymbol{\theta}_d} Q(\boldsymbol{\theta}_d;\boldsymbol{\theta}_d^{(t)}). \tag{7}$$

For the case of determining the parameters of the Bayesian network $X \rightarrow Y$, let C, $\mathcal{X}$ and $\mathcal{Y}$ represent the set of training observations for the values of X and Y together, the values of variable X alone, and the values of variable Y alone, respectively. For the latent-free observations, n_{yx} is the number of instantiations for which $X = x$ and $Y = y$, i.e., $n_{yx} = |\{(x_i, y_i) \in C : x_i = x, y_i = y\}|$. Similarly, $n_{y\bar{x}}$, $n_{\bar{y}x}$ and $n_{\bar{y}\bar{x}}$ are defined so that $n_{yx} + n_{y\bar{x}} + n_{\bar{y}x} + n_{\bar{y}\bar{x}} = |C|$. Likewise, n_x and $n_{\bar{x}}$ are the number of times $X = x$ and $X = \bar{x}$ for the X-only observations so that $n_x + n_{\bar{x}} = |\mathcal{X}|$. Finally, $n_y + n_{\bar{y}} = |\mathcal{Y}|$.

Now the Q function for $\boldsymbol{\theta}_r$ is

$$\begin{aligned} Q(\boldsymbol{\theta}_d;\boldsymbol{\theta}_d^{(t)}) = &\sum_{(x_i,y_i)\in C} \log\left(h_{y_i|x_i}(\boldsymbol{\theta}_r) h_{x_i}(\boldsymbol{\theta}_r)\right) + \sum_{x_i \in \mathcal{X}} \log\left(h_{x_i}(\boldsymbol{\theta}_r)\right) \\ &+ \sum_{y_i \in \mathcal{Y}} \sum_{x' \in \mathbb{X}} \log\left(h_{y_i|x'}(\boldsymbol{\theta}_r) h_{x'}(\boldsymbol{\theta}_r)\right) h_{x'|y_i}\left(\boldsymbol{\theta}_r^{(t)}\right) \\ &+ \sum_{x' \in \mathbb{X}} \left(\sum_{y' \in \mathbb{Y}} (\alpha_{y'|x'} - 1) \log(h_{y'|x'}(\boldsymbol{\theta}_r)) + (\alpha_{x'} - 1) \log(h_{x'}(\boldsymbol{\theta}_r)) \right). \end{aligned} \tag{8}$$

Maximization of (8) leads to the updates

$$\begin{aligned} \theta_x^{(t+1)} &= \frac{n_{yx} + n_{\bar{y}x} + n_x + n_y h_{x|y}(\boldsymbol{\theta}_r^{(t)}) + n_{\bar{y}} h_{x|\bar{y}}(\boldsymbol{\theta}_r^{(t)}) + (\alpha_x - 1)}{|C| + |\mathcal{X}| + |\mathcal{Y}| + \alpha_x + \alpha_{\bar{x}} - 2}, \\ \theta_{y|x}^{(t+1)} &= \frac{n_{yx} + n_y h_{x|y}(\boldsymbol{\theta}_r^{(t)}) + (\alpha_{y|x} - 1)}{n_{yx} + n_{\bar{y}x} + n_y h_{x|y}(\boldsymbol{\theta}_r^{(t)}) + n_{\bar{y}} h_{x|\bar{y}}(\boldsymbol{\theta}_r^{(t)}) + \alpha_{y|x} + \alpha_{\bar{y}|x} - 2}, \\ \theta_{y|\bar{x}}^{(t+1)} &= \frac{n_{y\bar{x}} + n_y h_{\bar{x}|y}(\boldsymbol{\theta}_r^{(t)}) + (\alpha_{y|\bar{x}} - 1)}{n_{y\bar{x}} + n_{\bar{y}\bar{x}} + n_y h_{\bar{x}|y}(\boldsymbol{\theta}_r^{(t)}) + n_{\bar{y}} h_{\bar{x}|\bar{y}}(\boldsymbol{\theta}_r^{(t)}) + \alpha_{y|\bar{x}} + \alpha_{\bar{y}|x} - 2}. \end{aligned} \tag{9}$$

The EM method simply starts with an initial parameter estimate $\boldsymbol{\theta}^{(0)}$ and iterates over (9) until convergence. We set $\alpha_x = \alpha_{\bar{x}} = 3$ and $\alpha_{y|x} = \alpha_{\bar{y}|x} = \alpha_{y|\bar{x}} = \alpha_{\bar{y}|\bar{x}} = 2$. These values are chosen so that the EM can always return a finite value and so the MAP estimate can correspond to the expectation of the posterior when using an uniform prior and $|\mathcal{X}| = |\mathcal{Y}| = 0$. Finally, we set the initial estimate as the MAP estimate when using only the C and $\mathcal{X}$ observations, i.e.,

$$\boldsymbol{\theta}_r^{(0)} = \left(\frac{n_{yx} + n_{\bar{y}x} + n_x + 2}{|C| + |\mathcal{X}| + 4}, \frac{n_{yx} + 1}{n_{yx} + n_{\bar{y}x} + 2}, \frac{n_{y\bar{x}} + 1}{n_{y\bar{x}} + n_{\bar{y}\bar{x}} + 2} \right). \tag{10}$$

Similarly, the EM can be used to determine the MAP estimates for $\boldsymbol{\theta}_l$ and $\boldsymbol{\theta}_u$. For equivalent priors, one can alternatively use the transformations $\mathbf{h}_{d,d'}$ to compute the MAP estimate for one representation using the estimates from an equivalent representation.

While the EM provides estimates for the model parameters, it does not provide an confidence in these values. It only provides a sense of the mean value for the parameters over the possible distribution of possible values. It is helpful to understand the spread of that distribution as well. This will allow the decision maker to assess risk better and whether or not to invest in gathering more training data to improve the abilities of the reasoning network model.

C. Second-Order PGMs

For complete training data, the posterior distribution for the PGM parameters are Dirichlet distributed. The natural uninformative prior for $\boldsymbol{\theta}_d$ is uniform over $\mathcal{S}_4$ so that $\alpha_{yx} = \alpha_{y\bar{x}} = \alpha_{\bar{y}x} = \alpha_{\bar{y}\bar{x}} = 1$. The transformations $\boldsymbol{\theta}_r = \mathbf{h}_{r,d}(\boldsymbol{\theta}_d)$ and $\boldsymbol{\theta}_l = \mathbf{h}_{l,d}(\boldsymbol{\theta}_d)$ leads to

$$f(\boldsymbol{\theta}_d) = 1 \Leftrightarrow f(\boldsymbol{\theta}_r) = \boldsymbol{\theta}_x(1 - \boldsymbol{\theta}_x) \Leftrightarrow f(\boldsymbol{\theta}_l) = \boldsymbol{\theta}_y(1 - \boldsymbol{\theta}_y) \tag{11}$$

as shown in Appendix A. Using this uninformative prior, it is straightforward to derive the posteriors for the parameters as

$$\begin{aligned} f(\boldsymbol{\theta}_u) &= f_\beta(\boldsymbol{\theta}_u;\boldsymbol{\alpha}_{YX}), \\ f(\boldsymbol{\theta}_r) &= f_\beta(\boldsymbol{\theta}_x;\boldsymbol{\alpha}_X) f_\beta(\boldsymbol{\theta}_{y|x};\boldsymbol{\alpha}_{Y|x}) f_\beta(\boldsymbol{\theta}_{y|\bar{x}};\boldsymbol{\alpha}_{Y|\bar{x}}) \\ f(\boldsymbol{\theta}_l) &= f_\beta(\boldsymbol{\theta}_y;\boldsymbol{\alpha}_Y) f_\beta(\boldsymbol{\theta}_{x|y};\boldsymbol{\alpha}_{X|y}) f_\beta(\boldsymbol{\theta}_{x|\bar{y}};\boldsymbol{\alpha}_{X|\bar{y}}), \end{aligned} \tag{12}$$

where $\boldsymbol{\alpha}_{YX} = [n_{yx} + 1, n_{y\bar{x}} + 1, n_{\bar{y}x} + 1, n_{\bar{y}\bar{x}} + 1]$, $\boldsymbol{\alpha}_{Y|x} = [n_{yx} + 1, n_{\bar{y}x} + 1]$, $\boldsymbol{\alpha}_{Y|\bar{x}} = [n_{y\bar{x}} + 1, n_{\bar{y}\bar{x}} + 1]$, $\boldsymbol{\alpha}_{X|y} = [n_{yx} + 1, n_{y\bar{x}} + 1]$, $\boldsymbol{\alpha}_{X|\bar{y}} = [n_{\bar{y}x} + 1, n_{\bar{y}\bar{x}} + 1]$, $\boldsymbol{\alpha}_X = [n_{yx} + n_{\bar{y}x} + 2, n_{y\bar{x}} + n_{\bar{y}\bar{x}} + 2]$, $\boldsymbol{\alpha}_Y = [n_{yx} + n_{y\bar{x}} + 2, n_{\bar{y}x} + n_{\bar{y}\bar{x}} + 2]$.

Second-order inference determines the distributions for the inferred probabilities given the posterior distribution for the model parameters. The second-order inference methods for Bayesian networks (see [4], [5]) assume the parameters are statically independent, which is true for the complete training data as can be verified for the two-node case by the posterior in (12). Both methods use the 'delta method' to approximate the distribution of the inferred probabilities by approximating its first two moments. In [4], variance elimination is used to compute partial derivatives, while in [5] the moments are propagated via an extension of belief propagation, which is much more efficient but is valid only for poly-tree networks. In any event, the first order inference process performs a transformation of the model parameters, e.g., $\theta_{x|y} = h_{x|y}(\boldsymbol{\theta}_r)$ (see Table I). The 'delta method' approximates the transformation through a first-order Tayor series approximation about the mean value of the model parameters so that in general

$$h(\boldsymbol{\theta}_d) \approx h(E[\boldsymbol{\Theta}_d]) + \nabla_{\boldsymbol{\theta}}^T h(E[\boldsymbol{\Theta}_d])(\boldsymbol{\theta}_d - E[\boldsymbol{\Theta}_d]), \tag{13}$$

so that

$$E[h(\boldsymbol{\Theta}_r)] = h(E[\boldsymbol{\Theta}_r]), \quad \mathrm{VAR}[h(\boldsymbol{\theta}_r)] = \nabla_{\boldsymbol{\theta}}^T h \mathbf{R} \nabla_{\boldsymbol{\theta}} h, \tag{14}$$

where $\mathbf{R}$ is the covariance matrix for the model parameters. The left side of (14) simply states that mean inferences are the standard first-order inferences operating on the mean values for model parameters. The new part of the second-order processing is simply the right side of (14) that approximates the variance for the distribution of the inferred probabilities. The primary distinction among the existing second-order inference methods, e.g., [4], [5], is in computing the gradient in (14).

For complete training data, the covariance matrix in (14) is diagonal where the variance terms are derived from the Dirichlet parameters in (12) via (3). For incomplete training data, the model parameters are not necessarily statically independent and $\mathbf{R}$ is not diagonal. The next section investigates the determination of $\mathbf{R}$ due to incomplete training data for purposes of second-order inferences via (14).

III. Second-Order Learning over Incomplete Data

This section discusses the determination of the covariance matrix for the model parameters in light of incomplete training data. The first subsection determines the covariance from the exact posterior distribution for the model parameters. However, the complexity of the method increases as the number of partial data increases. The second subsection estimates the covariance matrix from the inverse of the FIM.

A. Posterior Analysis

The likelihood of the i-th complete measurement $(x_i, y_i) \in \mathbb{X} \times \mathbb{Y}$ in C is $h_{y_i x_i}(\boldsymbol{\theta}_d)$. Note that the alphabets $\mathbb{X} = \{x, \bar{x}\}$ and $\mathbb{Y} = \{y, \bar{y}\}$. Similarly, the likelihoods of the i-th incomplete measurements in $\mathcal{X}$ and $\mathcal{Y}$ are $h_{x_i}(\boldsymbol{\theta}_d)$ and $h_{y_i}(\boldsymbol{\theta}_d)$, respectively, where $x_i \in \mathbb{X}$ and $y_i \in \mathbb{Y}$. Combining these likelihoods with the noninformative prior in (11), it is easy to formulate the posterior distribution for $\boldsymbol{\theta}_d$ for Markov and Bayesian network representations as

$$\begin{aligned}
f(\boldsymbol{\theta}_u) &= \frac{1}{Z}\theta_{yx}^{n_{yx}}\theta_{y\bar{x}}^{n_{y\bar{x}}}\theta_{\bar{y}x}^{n_{\bar{y}x}}(1-\theta_{yx}-\theta_{y\bar{x}}-\theta_{\bar{y}x})^{n_{\bar{y}\bar{x}}}(\theta_{yx}+\theta_{\bar{y}x})^{n_x} \\
&\quad \cdot(1-\theta_{yx}-\theta_{\bar{y}x})^{n_{\bar{x}}}(\theta_{yx}+\theta_{y\bar{x}})^{n_y}(1-\theta_{yx}-\theta_{y\bar{x}})^{n_{\bar{y}}}, \\
f(\boldsymbol{\theta}_r) &= \frac{1}{Z}\theta_x^{n_{yx}+n_{\bar{y}x}+n_x+1}(1-\theta_x)^{n_{y\bar{x}}+n_{\bar{y}\bar{x}}+n_{\bar{x}}+1}\theta_{y|x}^{n_{yx}}(1-\theta_{y|x})^{n_{\bar{y}x}} \\
&\quad \cdot\theta_{y|\bar{x}}^{n_{y\bar{x}}}(1-\theta_{y|\bar{x}})^{n_{\bar{y}\bar{x}}}(\theta_{y|x}\theta_x+\theta_{y|\bar{x}}(1-\theta_x))^{n_y} \\
&\quad \cdot(1-\theta_{y|x}\theta_x-\theta_{y|\bar{x}}(1-\theta_x))^{n_{\bar{y}}} \\
f(\boldsymbol{\theta}_l) &= \frac{1}{Z}\theta_y^{n_{yx}+n_{y\bar{x}}+n_y+1}(1-\theta_y)^{n_{\bar{y}x}+n_{\bar{y}\bar{x}}+n_{\bar{y}}+1}\theta_{x|y}^{n_{yx}}(1-\theta_{x|y})^{n_{y\bar{x}}} \\
&\quad \cdot\theta_{x|\bar{y}}^{n_{\bar{y}x}}(1-\theta_{x|\bar{y}})^{n_{\bar{y}\bar{x}}}(\theta_{x|y}\theta_y+\theta_{x|\bar{y}}(1-\theta_y))^{n_x} \\
&\quad \cdot(1-\theta_{x|y}\theta_y-\theta_{x|\bar{y}}(1-\theta_y))^{n_{\bar{x}}},
\end{aligned} \tag{15}$$

where Z is the normalizing constant so that all densities integrate to one. Note that Z is the same for all representations as they are equivalent distributions via a change-of-variable.

One can compute the exact moments for inferred marginal variable values conditioned on the observations or the other variable. This is done by exploiting the binomial expansion property. For instance, the mean and covariance for $\boldsymbol{\theta}_r$ can be computed as

$$E[\boldsymbol{\theta}_r] = \left(\frac{g_{1,0,0}}{g_{0,0,0}}, \frac{g_{0,1,0}}{g_{0,0,0}}, \frac{g_{0,0,1}}{g_{0,0,0}}\right) \tag{16}$$

$$[\mathbf{R}]_{i,j} = \frac{g_{\delta_{i,1}+\delta_{j,1},\delta_{i,2}+\delta_{j,2},\delta_{i,3}+\delta_{j,3}}}{g_{0,0,0}} - E[\boldsymbol{\theta}_r]_i E[\boldsymbol{\theta}_r]_j, \tag{17}$$

where as shown in Appendix B, $g_{\kappa,k,\ell}$ are computed as

$$\begin{aligned}
g_{\kappa,k,\ell} = \sum_{i=0}^{n_y}\sum_{j=0}^{n_{\bar{y}}}\binom{n_y}{i}\binom{n_{\bar{y}}}{j}\Big[&B(t_x+\kappa+i+j, t_{\bar{x}}+n_y+n_{\bar{y}}-i-j)\cdot \\
&\cdot B(n_{yx}+1+i+k, n_{\bar{y}x}+1+j)\cdot \\
&\cdot B(n_{y\bar{x}}+1+n_y-i+\ell, n_{\bar{y}\bar{x}}+1+n_{\bar{y}}-j)\Big],
\end{aligned} \tag{18}$$

$$t_x = n_{yx}+n_{\bar{y}x}+n_x+2, \quad t_{\bar{x}} = n_{y\bar{x}}+n_{\bar{y}\bar{x}}+n_{\bar{x}}+2,$$

and $\delta_{i,j}$ is the Kronecker delta function that is zero unless $i = j$ where it is one. Note that $Z = g_{0,0,0}$. Similarly, one can compute the mean and covariances for $\boldsymbol{\theta}_l$ and $\boldsymbol{\theta}_u$. From (18), the computational complexity to compute the moments of $\boldsymbol{\theta}_r$ grows as $O(|\mathcal{Y}|^2)$. For larger Bayesian networks, the binomials become multinomials and the number of different multinomials grows exponentially with the number of variables. As a result, the exact moment computations becomes prohibitive and a faster method to estimate the covariance matrix is needed.

B. Fisher Information

It is known that under mild regularity conditions for the likelihood and the prior, the posterior distribution is asymptotically normal with mean given by the maximum likelihood estimate and covariance equal to the inverse FIM [12], which is a result of the Bernstein-von Mises theorem. As a result, it is reasonable to approximate the distribution of the model parameters using the output of the EM algorithm for the mean and the Fisher information to approximate the covariance. As least, these approximations of the moments will align with those computed from the true posterior for a large number of training samples.

Given that the observations are statistically independent of each other, the log-likelihood for the collection of uninstantiated observations $\mathbf{T}$ for the $X \rightarrow Y$ model is

$$\begin{aligned}
L(\mathbf{T};\boldsymbol{\theta}_r) &= \sum_{(X_i,Y_i)\in C}\log(P(X_i,Y_i;\boldsymbol{\theta}_r)) + \sum_{X_i\in\mathcal{X}}\log(P(X_i;\boldsymbol{\theta}_r)) \\
&\quad + \sum_{Y_i\in\mathcal{Y}}\log(P(Y_i;\boldsymbol{\theta}_r)), \\
&= \sum_{(X_i,Y_i)\in C}\log(h_{Y_iX_i}(\boldsymbol{\theta}_r) + \sum_{X_i\in\mathcal{X}}\log(h_{X_i}(\boldsymbol{\theta}_r)) + \sum_{Y_i\in\mathcal{Y}}\log(h_{Y_i}(\boldsymbol{\theta}_r)).
\end{aligned} \tag{19}$$

The Fisher information matrix (FIM) is

$$\mathbf{F} = E_{\mathbf{T}}\left[\nabla_{\boldsymbol{\theta}_r}L(\mathbf{T};\boldsymbol{\theta}_r)\left(\nabla_{\boldsymbol{\theta}_r}L(\mathbf{T};\boldsymbol{\theta}_r)\right)^T\right]. \tag{20}$$

As shown in Appendix C, the FIM for incomplete training data is

$$\mathbf{F} = \begin{pmatrix} \frac{|C|+|\mathcal{X}|}{\theta_x(1-\theta_x)} & 0 & 0 \\ 0 & \frac{|C|\theta_x}{\theta_{y|x}(1-\theta_{y|x})} & 0 \\ 0 & 0 & \frac{|C|(1-\theta_x)}{\theta_{y|\bar{x}}(1-\theta_{y|\bar{x}})} \end{pmatrix} + \frac{|\mathcal{Y}|}{h_y(\boldsymbol{\theta}_r)(1-h_y(\boldsymbol{\theta}_r))}\mathbf{v}\mathbf{v}^T, \tag{21}$$

where

$$\mathbf{v} = \nabla_{\boldsymbol{\theta}_r}\left(h_y(\boldsymbol{\theta}_r)\right) = \begin{bmatrix} \theta_{y|x}-\theta_{y|\bar{x}} & \theta_x & 1-\theta_x \end{bmatrix}^T. \tag{22}$$

The covariance is estimated as the inverse of the FIM. Using the matrix inversion lemma to determine the inverse of the FIM leads to

$$\mathbf{R} = \mathbf{D} - \frac{1}{\gamma(\boldsymbol{\theta})}\mathbf{D}\mathbf{v}\mathbf{v}^T\mathbf{D}, \tag{23}$$

where

$$\mathbf{D} = \begin{pmatrix} \frac{\theta_x(1-\theta_x)}{|C|+|\mathcal{X}|} & 0 & 0 \\ 0 & \frac{\theta_{y|x}(1-\theta_{y|x})}{|C|\theta_x} & 0 \\ 0 & 0 & \frac{\theta_{y|\bar{x}}(1-\theta_{y|\bar{x}})}{|C|(1-\theta_x)} \end{pmatrix}, \tag{24}$$

is a diagonal matrix representing the covariance due to solely the latent-free and the x-only incomplete training observations and

$$\gamma(\boldsymbol{\theta}) = \mathbf{v}^T\mathbf{D}\mathbf{v} + \frac{h_y(\boldsymbol{\theta}_r)(1-h_y(\boldsymbol{\theta}_r))}{|\mathcal{Y}|}. \tag{25}$$

The estimation of the covariance via (23) requires the model parameters $\boldsymbol{\theta}_r$ which are unknown *a priori*. In practice, the estimates from the EM algorithm $\hat{\boldsymbol{\theta}}_r$ are used to determine $\mathbf{R}$. Because the EM method that iterates over (9) is biased, we use a modified version of $\mathbf{D}$ in the simulations so that the covariance estimate via the FIM fits the true covariance for the posterior for complete data. To this end,

$$\mathbf{D} = \begin{pmatrix} \frac{\theta_x(1-\theta_x)}{|C|+|\mathcal{X}|+5} & 0 & 0 \\ 0 & \frac{\theta_{y|x}(1-\theta_{y|x})}{|C|\theta_x+3} & 0 \\ 0 & 0 & \frac{\theta_{y|\bar{x}}(1-\theta_{y|\bar{x}})}{|C|(1-\theta_x)+3} \end{pmatrix}, \tag{26}$$

and the covariance is estimated via (21), where $\mathbf{v}$ and $\mathbf{D}$ are given by (22) and (26), respectively. The mean value for the model parameters are obtained from MAP estimates via EM.

The covariance matrix for the model parameters $\boldsymbol{\theta}_l$ and $\boldsymbol{\theta}_u$ are similarly computed. Using the reparameterization property of the FIM and taking the inverse, leads to the following transformation between the covariance for $\boldsymbol{\theta}_d$ and for $\boldsymbol{\theta}_{d'}$,

$$\mathbf{R}_{\boldsymbol{\theta}_d} = \mathbf{J}_{\mathbf{h}_{d,d'}} \mathbf{R}_{\boldsymbol{\theta}_{d'}} \mathbf{J}^T_{\mathbf{h}_{d,d'}}, \tag{27}$$

where $\mathbf{J}_{\mathbf{h}_{d,d'}}$ is the Jacobian for the transformation $\mathbf{h}_{d,d'}(\boldsymbol{\theta}_{d'})$. Setting $d' = r$ and $d = l$ or $d = u$ allows for the computation of the covariances for $\boldsymbol{\theta}_l$ and $\boldsymbol{\theta}_u$, respectively.

The complexity to compute the FIM is due to the expectation over the possible variable values. For the two-node network, simple closed form expressions result. For more general network structures, the expectation will require numerical computations. Nevertheless, the computational complexity for the FIM is independent of the sizes of the latent-free $|C|$ and incomplete $|\mathcal{X}|$, and $|\mathcal{Y}|$ training data. This is in contrast to the exact computation of covariance from the posterior. We expect that as the size of the network grows, the FIM approximation remains feasible while the posterior method become intractable.

C. Discussion about Second-Order Inference with Incomplete Data Training

The variances of various inferences can be determined using the 'delta method'. In this subsection, we investigate the reduction of the variance for inferences due to the incomplete training data. To this end, we use the standard form for $\mathbf{D}$ in (24). Here, we consider inferring the probability that $X = x$ given the evidence that $Y = y$ using the $\boldsymbol{\theta}_r$ parameters. To this end, the second-order inference computes the mean and variance of the inference probability via (14) using $h_{y|x}(\boldsymbol{\theta}_r)$ as given in Table I so that

$$\nabla^T_{\boldsymbol{\theta}_r} h_{x|y} = \frac{\begin{pmatrix} \theta_{y|x}\theta_{y|\bar{x}} & \theta_{y|\bar{x}}\theta_x(1-\theta_x) & -\theta_{y|x}\theta_x(1-\theta_x) \end{pmatrix}}{\left(\theta_{y|x}\theta_x + \theta_{y|\bar{x}}(1-\theta_x)\right)^2}. \tag{28}$$

For a $|C| > 0$ with no incomplete data, i.e., $|\mathcal{X}| = |\mathcal{Y}| = 0$, the insertion of (28) into (14) results (after many simplifying steps) into a baseline variance of

$$V^{(C)}_{x|y} = \frac{h_{x|y}(\boldsymbol{\theta}_r)(1 - h_{x|y}(\boldsymbol{\theta}_r))}{h_y(\boldsymbol{\theta}_r)|C|}. \tag{29}$$

Clearly, an infinite amount of latent-free data will drive the epistemic uncertainty about $p_{x|y}$ to zero. On the other hand, adding an infinite amount of Y-only data while keeping $|C|$ fixed and $|\mathcal{X}| = 0$ does not change the variance. Such partial data does not lower the epistemic uncertainty for $p_{x|y}$. In contrast, incorporation of an infinite amount of X-only data when not using any Y-only data does lower the variance to

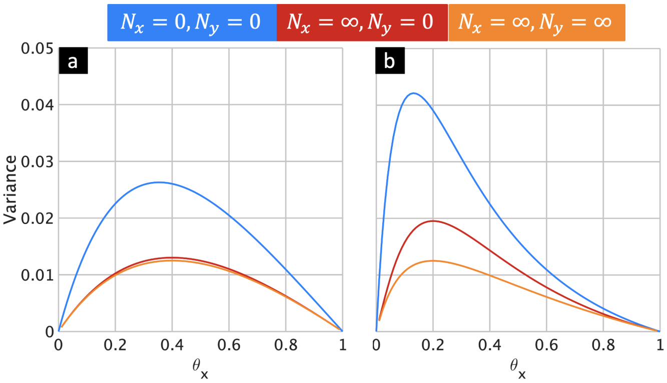

Fig. 1. Reduction in variance using an infinite amount of Y-only or Y- and X-only training data as a function of θ_x: (a) Nearly independent variables and (b) strong dependency between variables.

$$\mathrm{VAR}[\theta_{x|y}] = V^{(C)}_{x|y}\left(1 - h_y(\boldsymbol{\theta}_r)\frac{h_{x|y}(\boldsymbol{\theta}_r)(1 - h_{x|y}(\boldsymbol{\theta}_r))}{\theta_x(1-\theta_x)}\right). \tag{30}$$

The infinite amount of partial X-only does lower the uncertainty about the inference, but the value of such data exhibits diminishing returns as, in general, it cannot drive the uncertainty to zero. Now, if both an infinite number of partial X- and Y-only data ia added to the complete data, the variance reduces further to

$$\mathrm{VAR}[\theta_{y|x}] = V^{(C)}_{y|x}\left(\frac{h_{x|\bar{y}}(\boldsymbol{\theta}_r)h_{\bar{x}|\bar{y}}(\boldsymbol{\theta}_r)h_{\bar{y}}(\boldsymbol{\theta}_r)}{h_{x|y}(\boldsymbol{\theta}_r)h_{\bar{x}|y}(\boldsymbol{\theta}_r)h_y(\boldsymbol{\theta}_r) + h_{x|\bar{y}}(\boldsymbol{\theta}_r)h_{\bar{x}|\bar{y}}(\boldsymbol{\theta}_r)h_{\bar{y}}(\boldsymbol{\theta}_r)}\right). \tag{31}$$

Figure 1 demonstrates the limits for the reduction in variance due to incomplete data for two cases where $|C| = 20$. Figure 1(a) shows the case where the X and Y variables are nearly independent with $\theta_{y|x} = .6$ and $\theta_{y|\bar{x}} = .4$ as θ_x varies from 0 to 1. In contrast, Figure 1(b) shows the case where two variables have strong dependency with $\theta_{y|x} = .8$ and $\theta_{y|\bar{x}} = .2$. The baseline variance for only the complete data is shown in blue. In both cases, addition of X-only data does decrease the variance; the decrease is more significant as θ_x moves away from its extreme value and shows higher aleatoric uncertainty. The additional value of the Y-only data is less significant for the nearly independent case where such partial data provide little insight into the values of the latent X variable.

IV. Simulations

First, we compare the effectiveness of the variance of the inferred probabilities to capture the uncertainty about the estimate of its mean. To this end, we generated 1000 realizations of two-node networks where the ground truth joint probabilities are selected uniformly over the simplex $\mathcal{S}_4$. For each network both latent-free and incomplete training data is generated. The mean and covariances are computed either from the true posterior as in (16)-(17) or the EM iterations in (9) lead to the mean estimates for the $\boldsymbol{\theta}_r$ parameters and the covariance matrix is derived from the biased FIM (where $\mathbf{D}$ is given by (26)). Finally, the variances of the inferred probabilities are extracted via (14). As in the previous section, the simulations focus on the inference of $p_{x|y}$. To characterize the quality of the variances, we compute the

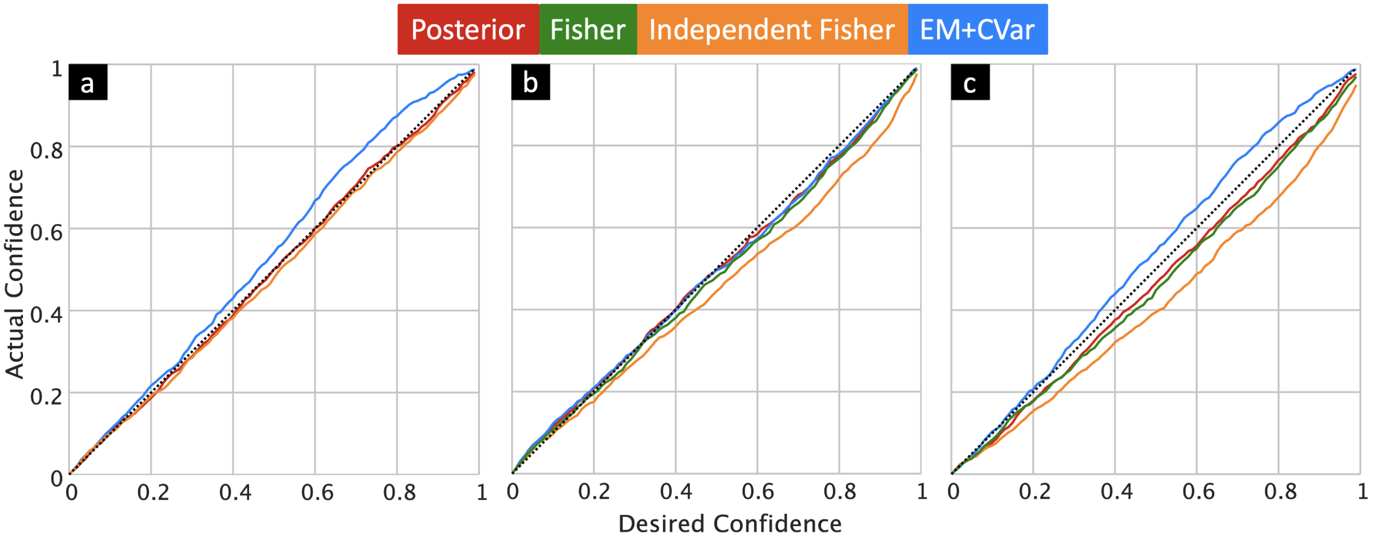

Fig. 2. DeCBoD plots for various methods to determine the distribution of the $y_{x|y}$ inference for various incomplete training data when $|C| = 20$: (a) $|\mathcal{X}| = 100$ and $|\mathcal{Y}| = 0$, (b) $|\mathcal{X}| = 0$ and $|\mathcal{Y}| = 100$, and (c) $|\mathcal{X}| = 100$ and $|\mathcal{Y}| = 100$. Best close to the diagonal.

desired confidence bound divergence (DeCBoD) by comparing the desired confidence bound strength to the ratio of times the ground truth falls within the bound over the 1000 realizations. The bounds are set by assuming the distribution of $p_{x|y}$ is beta with the computed mean and variance. More details about the DeCBoD calculations is given in [5].

Figure 2 illustrates the DeCBoD for the moments extracted from *Posterior* and *Fisher* methods for latent-free data with $|C| = 20$ augmented by various types of incomplete data. To illustrate the need to consider the correlations between model parameters, the *Independent Fisher* method treats the FIM derived covariance matrix as diagonal by zeroing out the off diagonal term. The final method (*EM+CVar*) considers extracting the covariance matrix from the latent-free C data only while extracting the mean from the EM over all the data. In this case, the variances for inferences are larger than the EM error. Overall, both the *Posterior* and *Fisher* methods faithfully provides bounds whose actual confidence is consistent with the desired strength. The *Independent Fisher* method underestimates the error by providing tighter bounds than desired when $|\mathcal{Y}| > 0$ making the true covariance non-diagonal. The *EM+CVar* method usually provides looser bounds than desired. The exception is when $|\mathcal{Y}| = 0$ as the X-only incomplete data does not inform the $p_{x|y}$ inference as discovered in Section III-C.

Figures 3-5 compare the variances of $p_{x|y}$ due to the *Posterior* and *Fisher* methods with and without the augmentation of the incomplete data. The posterior and Fisher variances are correlated with each other, but the correlation weakens for larger variances. For the most part, the variances are smaller without the incomplete data augmentation as long as X-only data augments the complete data. Occasionally, the extra incomplete data increases the variances when it changes the mean values to be closer to 0.5. When $|\mathcal{X}| = 0$, the incomplete data does not decrease the posterior variance. The *Fisher* variance on average does not increase or decrease with the augmented data. The change in variance is due to changes in the mean in the EM method.

V. Conclusions

In this paper, we investigated the implication of incomplete BNs. Specifically, the paper develops a method to extract the mean and variance of the model parameters from the true posterior. Furthermore, we derive a computationally efficient approximation of the parameter covariances from the FIM. This leads to closed form expressions for the variances of inferred probabilities, which informs about the effectiveness and limitations of incomplete training data. Simulations validate the effectiveness of the true and approximative covariances to capture the uncertainty of the inferred probabilities. Furthermore, the simulations illustrate when and when not partial measurements reduce the variances of an inference.

It is expected that FIM will provide a scalable method to compute the model parameters for larger PGMs. Future work will formulate the computational engine to estimate the covariance for arbitrary Bayesian networks via the FIM when incorporating incomplete data. Furthermore, we will investigate Laplace approximations for even more computational efficiency. We hope that this effort will lead to better theoretical understanding of what types of incomplete training data are effective or not in reducing the variance of a particular inference based upon the structure of the Bayesian network.

Appendix A
Derivation of Priors for $\boldsymbol{\theta}_r$ and $\boldsymbol{\theta}_l$

First, the transformations $\mathbf{h}_{r,u}(\cdot)$ and $\mathbf{h}_{l,u}(\cdot)$ are one-to-one because their inverse functions are $\mathbf{h}_{u,r}(\cdot)$ and $\mathbf{h}_{u,l}(\cdot)$, respectively. Given $f(\boldsymbol{\theta}_u) = 1$, by the 'change of variables' property for distributions,

$$f(\boldsymbol{\theta}_r) = \det(\mathbf{J}_{\mathbf{h}_{d,r}}) \quad \text{and} \quad f(\boldsymbol{\theta}_r) = \det(\mathbf{J}_{\mathbf{h}_{d,l}}),$$

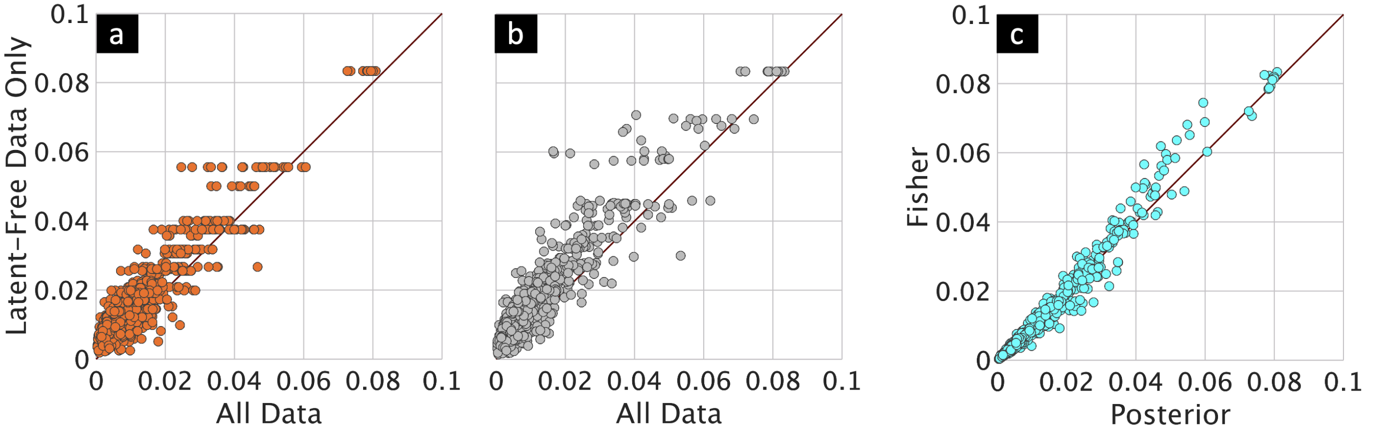

Fig. 3. Scatter plots comparing variances obtained by various methods when $|\mathcal{C}| = 20$, $|\mathcal{X}| = 100$, and $|\mathcal{Y}| = 0$: (a) Posterior of latent-free data versus all data, (b) Fisher of latent-free data versus all data, and (c) Fisher versus posterior for all data.

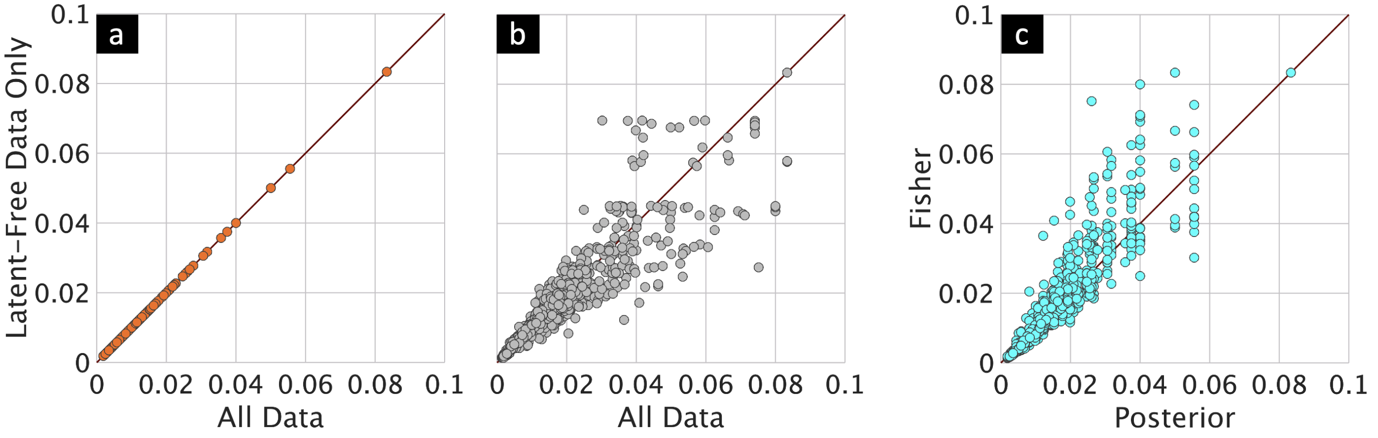

Fig. 4. Scatter plots comparing variances obtained by various methods when $|\mathcal{C}| = 20$, $|\mathcal{X}| = 0$, and $|\mathcal{Y}| = 100$: (a) Posterior of latent-free data versus all data, (b) Fisher of latent-free data versus all data, and (c) Fisher versus posterior for all data.

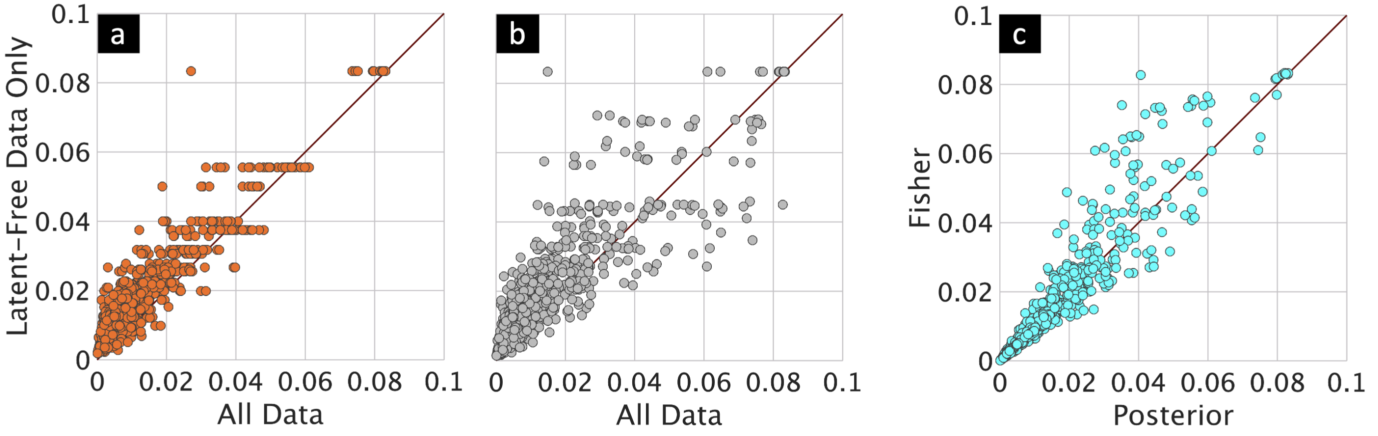

Fig. 5. Scatter plots comparing variances obtained by various methods when $|\mathcal{C}| = 20$, $|\mathcal{X}| = 100$, and $|\mathcal{Y}| = 100$: (a) Posterior of latent-free data versus all data, (b) Fisher of latent-free data versus all data, and (c) Fisher versus posterior for all data.

where $\mathbf{J_h} = [\nabla_{\boldsymbol{\theta}} h_1 \cdots \nabla_{\boldsymbol{\theta}} h_K]^T$ is the Jacobian for the vector function $\mathbf{h}$. Since

$$\mathbf{J}_{\mathbf{h}_{u,r}} = \begin{pmatrix} \theta_{y|x} & \theta_x & 0 \\ -\theta_{y|\bar{x}} & 0 & 1-\theta_x \\ 1-\theta_{y|x} & -\theta_x & 0 \end{pmatrix} \quad \text{and} \tag{32}$$

$$\mathbf{J}_{\mathbf{h}_{u,l}} = \begin{pmatrix} \theta_{x|y} & \theta_y & 0 \\ 1-\theta_{x|y} & -\theta_y & 0 \\ -\theta_{x|\bar{y}} & 0 & 1-\theta_y \end{pmatrix} \tag{33}$$

then $\det(\mathbf{J}_{\mathbf{h}_{u,r}}) = \theta_x(1-\theta_x)$ and $\det(\mathbf{J}_{\mathbf{h}_{u,l}}) = \theta_x(1-\theta_x)$. □

Appendix B
Derivation of $g_{\kappa,k,\ell}$

$$\begin{aligned} g_{k,\ell} &= Z \int (\theta_x)^{\kappa} (\theta_{y|x})^{k} (\theta_{y|\bar{x}})^{\ell} f(\boldsymbol{\theta}_r) d\theta_x d\theta_{y|x} d\theta_{y|\bar{x}} \\ &= \int \theta_x^{n_{yx}+n_{\bar{y}x}+n_x+\kappa+1} (1-\theta_x)^{n_{y\bar{x}}+n_{\bar{y}\bar{x}}+n_{\bar{x}}+1} \theta_{y|x}^{n_{yx}+k} (1-\theta_{y|x})^{n_{\bar{y}x}} \\ &\quad \cdot \theta_{y|\bar{x}}^{n_{y\bar{x}}+\ell} (1-\theta_{y|\bar{x}})^{n_{\bar{y}\bar{x}}} (\theta_{y|x}\theta_x + \theta_{y|\bar{x}}(1-\theta_x))^{n_y} \\ &\quad \cdot ((1-\theta_{y|x})\theta_x + (1-\theta_{y|\bar{x}})(1-\theta_x))^{n_{\bar{y}}} d\theta_x d\theta_{y|x} d\theta_{y|\bar{x}}. \end{aligned} \tag{34}$$

Now expanding the polynomials associated to p_y and $p_{\bar{y}}$ via the binomial expansion property leads to

$$\begin{aligned} g_{k,\ell} = \sum_{i=0}^{n_y} \sum_{j=0}^{n_{\bar{y}}} \binom{n_y}{i} \binom{n_{\bar{y}}}{j} \Big[\int \theta_x^{t_x+\kappa+i+j-1} (1-\theta_x)^{t_{\bar{x}}+n_y-i+n_{\bar{y}}-j-1} d\theta_x \\ \cdot \int \theta_{y|x}^{n_{yx}+k+i} (1-\theta_{y|x})^{n_{\bar{y}x}+j} d\theta_{y|x} \int \theta_{y|\bar{x}}^{n_{y\bar{x}}+\ell+n_y-i} (1-\theta_{y|\bar{x}})^{n_{\bar{y}\bar{x}}+n_{\bar{y}}-j} d\theta_{y|\bar{x}} \Big]. \end{aligned} \tag{35}$$

Given that the three integrals represent beta functions (see (2)), (18) follows. □

Appendix C
Derivation of the FIM for Incomplete Training Data

Given that each instantiation of the network values is statistically independent of the others, the FIM for the likelihood given in (19) can be expressed as

$$\begin{aligned} \mathbf{F} =& |\mathcal{C}| \cdot E[\nabla_{\boldsymbol{\theta}_r} \log(h_{YX}(\boldsymbol{\theta_r})) \nabla_{\boldsymbol{\theta}_r}^T \log(h_{YX}(\boldsymbol{\theta_r}))] \\ &+ |\mathcal{X}| \cdot E[\nabla_{\boldsymbol{\theta}_r} \log(h_X(\boldsymbol{\theta_r})) \nabla_{\boldsymbol{\theta}_r}^T \log(h_X(\boldsymbol{\theta_r}))] \\ &+ |\mathcal{Y}| \cdot E[\nabla_{\boldsymbol{\theta}_r} \log(h_Y(\boldsymbol{\theta_r})) \nabla_{\boldsymbol{\theta}_r}^T \log(h_Y(\boldsymbol{\theta_r}))] \end{aligned} \tag{36}$$

$$\begin{aligned} & E[\nabla_{\boldsymbol{\theta}_r} \log(h_{YX}) \nabla_{\boldsymbol{\theta}_r}^T \log(h_{YX})] = \\ & \quad \sum_{(x',y')\in\mathbb{X}\times\mathbb{Y}} h_{y'x'} \nabla_{\boldsymbol{\theta}_r} \log(h_{y'x'}) \nabla_{\boldsymbol{\theta}_r}^T \log(h_{y'x'}) \\ & \quad = \frac{1}{\theta_{y|x}\theta_x} \begin{pmatrix} \theta_{y|x}^2 & \theta_{y|x}\theta_x & 0 \\ \theta_{y|x}\theta_x & \theta_x^2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\ & \quad + \frac{1}{\theta_{y|\bar{x}}(1-\theta_x)} \begin{pmatrix} \theta_{y|\bar{x}}^2 & 0 & -\theta_{y|\bar{x}}(1-\theta_x) \\ 0 & 0 & 0 \\ -\theta_{y|\bar{x}}(1-\theta_x) & 0 & (1-\theta_x)^2 \end{pmatrix} \\ & \quad + \frac{1}{(1-\theta_{y|x})\theta_x} \begin{pmatrix} (1-\theta_{y|x})^2 & -(1-\theta_{y|x})\theta_x & 0 \\ -(1-\theta_{y|x})\theta_x & \theta_x^2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\ & \quad + \frac{1}{(1-\theta_{y|\bar{x}})(1-\theta_x)} \begin{pmatrix} (1-\theta_{y|\bar{x}})^2 & 0 & (1-\theta_{y|\bar{x}})(1-\theta_x) \\ 0 & 0 & 0 \\ (1-\theta_{y|\bar{x}})(1-\theta_x) & 0 & (1-\theta_x)^2 \end{pmatrix} \\ & \quad = \begin{pmatrix} \frac{1}{\theta_x(1-\theta_x)} & 0 & 0 \\ 0 & \frac{\theta_x}{\theta_{y|x}(1-\theta_{y|x})} & 0 \\ 0 & 0 & \frac{(1-\theta_x}{\theta_{y|\bar{x}}(1-\theta_{y|\bar{x}})} \end{pmatrix} \end{aligned} \tag{37}$$

$$\begin{aligned} & E[\nabla_{\boldsymbol{\theta}_r} \log(h_X) \nabla_{\boldsymbol{\theta}_r}^T \log(h_X)] = \sum_{x'\in\mathbb{X}} h_{x'} \nabla_{\boldsymbol{\theta}_r} \log(h_{x'}) \nabla_{\boldsymbol{\theta}_r}^T \log(h_{x'}) \\ & \quad = \frac{1}{\theta_x(1-\theta_x)} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \end{aligned} \tag{38}$$

$$\begin{aligned} & E[\nabla_{\boldsymbol{\theta}_r} \log(h_Y) \nabla_{\boldsymbol{\theta}_r}^T \log(h_Y)] = \sum_{y'\in\mathbb{Y}} h_{y'} \nabla_{\boldsymbol{\theta}_r} \log(h_{y'}) \nabla_{\boldsymbol{\theta}_r}^T \log(h_{y'}) \\ & \quad = \frac{1}{h_y(1-h_y)} \nabla_{\boldsymbol{\theta}_r} h_y \nabla_{\boldsymbol{\theta}_r}^T h_y. \end{aligned} \tag{39}$$

Insertion of (37)-(39) into (36) and defining $\mathbf{v}$ via (22) leads to (21). □

Acknowledgements

We wish to thank Dr. Magdalena Ivanovska for her valuable comments to improve the paper. This research was sponsored by the U.S. CCDC Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. K. V. M. acknowledges support from the National Academies of Sciences, Engineering, and Medicine via Army Research Laboratory Harry Diamond Distinguished Postdoctoral Fellowship. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] J. Rohmer, "Uncertainties in conditional probability tables of discrete Bayesian belief networks: A comprehensive review," *Engineering Applications of Artificial Intelligence*, vol. 88, p. 103384, 2020.

[2] P. P. Shenoy, "A valuation-based language for expert systems," *Int. Journal of Approximate Reasoning*, vol. 3, no. 2, pp. 383–411, 1989.

[3] M. Zaffalon and E. Fagiuoli, "2U: An exact interval propagation algorithm for polytrees with binary variables," *Artificial Intelligence*, vol. 106, no. 1, pp. 77–107, 1998.

[4] T. Van Allen, A. Singh, R. Greiner, and P. Hooper, "Quantifying the uncertainty of a belief net response: Bayesian error-bars for belief net inference," *Artificial Intelligence*, vol. 172, no. 4-5, pp. 483–513, 2008.

[5] L. Kaplan and M. Ivanovska, "Efficient belief propagation in second-order Bayesian networks for singly-connected graphs," *International Journal of Approximate Reasoning*, vol. 93, pp. 132–152, 2018.

[6] F. V. Jensen and T. D. Nielsen, *Bayesian Networks and Decision Graphs*. New York: Springer Science & Business Media, 2007.

[7] S. L. Lauritzen, "The EM algorithm for graphical association models with missing data," *Computational Statistics & Data Analysis*, vol. 19, no. 2, pp. 191 – 201, 1995.

[8] G. Pavlin, A.-L. Jousselme, J. P. de Villiers, P. C. Costa, K. Laskey, F. Mignet, and A. de Waal, "Online system evaluation and learning of data source models: A probabilistic generative approach," in *ISIF/IEEE International Conference on Information Fusion*, 2019.

[9] R. Dechter, "Bucket elimination: A unifying framework for probabilistic inference," in *Proceedings of the Twelfth international conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1996, pp. 211–219.

[10] J. Pearl, "Fusion, propagation, and structuring in belief networks," *Artificial Intelligence*, vol. 29, no. 3, pp. 241–288, Sep. 1986.

[11] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Foundations and Trends® in Machine Learning*, vol. 1, no. 1–2, pp. 1–305, Jan. 2008.

[12] A. M. Walker, "On the asymptotic behaviour of posterior distributions," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 31, no. 1, pp. 80–88, 1969.