PROCEEDINGS OF SPIE

Decision support using SAR and LiDAR machine learning target classification and Bayesian belief networks

Bogart, Christopher, Solorzano, Lidia, Lam, Stephen

SPIE.

Decision support using SAR and LiDAR machine learning target classification and Bayesian belief networks

Christopher Bogart, Lidia Solorzano, and Stephen Lam

Expedition Technology, Inc, 13865 Sunrise Valley Drive, Suite 350, Herndon, VA USA, 20171

ABSTRACT

EXP's IC3D system classifies SAR or LiDAR derived 3D point clouds using a deep representation learning approach, producing as output a vector of categorical posterior probabilities of target classifications. Such posterior probabilities are suitable observational inputs to a Bayesian belief network (BBN), such as the EXP Shadow Compass system. In concert with conditional probabilities of intermediate events depending on the observation states, the Bayesian network computes posterior probabilities for events conditionally dependent on those intermediate event states. We demonstrate this approach by computing posterior event probabilities for sample analyst scenarios with intermediate event conditional probabilities specified by analysts. Future work could include extending the Bayesian network approach to discovery of the network topology from analyst data.

Keywords: Bayesian inference, decision support, data fusion, SAR, LiDAR

1. INTRODUCTION

In recent years there has been much work in the computer vision community on using deep learning techniques as a means of identifying objects of interest in images. These techniques quantitatively identify the strength of inference for identified objects using such metrics as average precision for each type (class) of object. Point cloud deep learning is a subset of machine learning for computer vision where the images are of point clouds, representing three dimensional objects in the scene. Our current work uses the results of deep learning on point clouds derived from LiDAR or SAR as imaging techniques. The topic of this paper is work done by Expedition Technology, Incorporated (EXP) using Bayesian Inference as an interpretive and data fusion technique in concert with point cloud machine learning, taking the probability scores of object detection in point clouds as the input to child nodes of a Bayesian belief network.

The identification of objects in point cloud images with deep learning in and of itself may only be one piece of information within a larger analyst picture. The idea explored in this paper is to posit the identification of objects in point cloud data in IC3D (the Intelligent Classification for 3D system developed by EXP) as representing contextual information regarding a process that relates to other processes, conditioned on the outcome of object identification. The association of these dependent processes requires further interpretation to form a meaningful picture of a question of interest to an analyst.

For instance, if unusual activity in an area leads to a preponderance of certain types of vehicles in point clouds of the area, this may indicate that an unusual event, such as a natural disaster or economic or geopolitical event, is in progress in that area. But reaching that conclusion may require the fusion of image data with other human and machine created interpretations of data from other sources. For example, this could include weather data, reports from the media, and first responders. In many real-world scenarios, both the reliability of such information, and how the various components of information relate to one another, may have some degree of quantifiable uncertainty and weighting.

One technique for fusing data from disparate sources with a degree of uncertainty in the reliability of the information from each source, is to interpret such data as representing multiple probabilistic processes of events given data, with conditional probabilistic dependencies between each process. This sort of data fusion problem fits neatly into the Bayesian inference domain.

Further author information: (Send correspondence to Christopher Bogart)
Christopher Bogart: E-mail: cbogart@exptechinc.com

Specifically, we treat the outputs of point cloud deep learning classification as the inputs to a Bayesian belief network. The process of machine learning results in detection of objects in the point cloud scene, and identification of the class of object (e.g., ground vehicle, aircraft, etc.) with an associated confidence score. The confidence score is readily convertible to a probability of the object being of each class. This probability, then, serves as input evidence to nodes on a Bayesian belief network.

We emphasize that our use of Bayesian inference in the current work remains decoupled from the point cloud machine learning process that generates the object probabilities. Bayesian methods have been incorporated directly into deep learning, such as hyperparameter search[1] and sampling in Bayesian neural networks.[2] However, these techniques are not the focus of the current work.

Bayesian inference with network graphs is not a new technique algorithmically but has found application in fields as diverse as medicine, psychology, environmental science, and sports. Coupled with modern machine learning techniques for point cloud image inference, Bayesian inference has the potential to serve as a framework for integration of point cloud machine learning with disparate data sources to produce a quantitative model in terms of joint probability of a hypothesized outcome.

2. POINT CLOUD INFERENCE

The network architecture we used for point cloud learning on LiDAR and SAR images was based on the VoxelNet architecture originally developed by Apple.[3] This is a "representation learning" architecture based on convolutional neural networks (CNN). Specific extensions applied by EXP to our network architecture included (see Ref.[4] has more details):

- Using focal loss[5] during training to compensate for the known class imbalance in the point cloud data set.

- Optimizing the network architecture using the DARTS[6] neural architecture search.

- Prototyped using an extension of CenterNet[7] to three dimensions to enhance anchor-free object detection –this enhancement was not implemented for the current work.
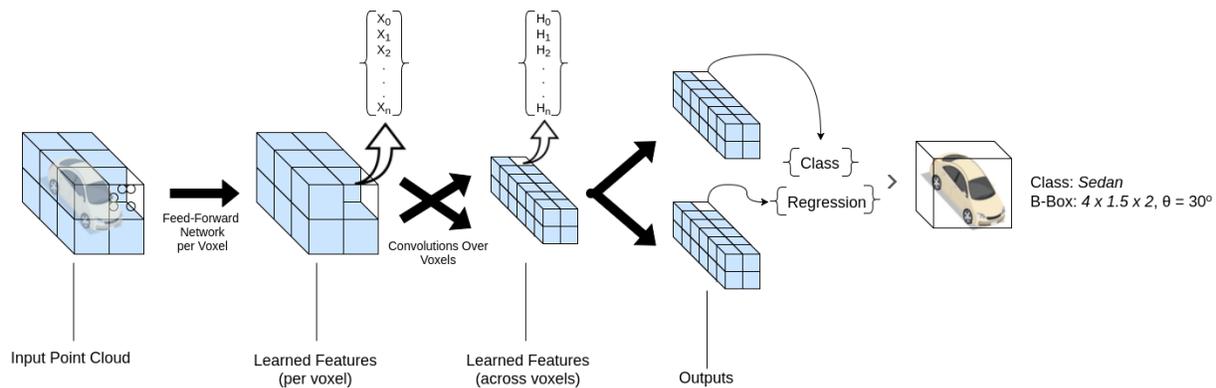

Figure 1. Overview of the IC3D neural network architecture

The point cloud dataset used in this work was provided by the US Government and consisted of 41 LiDAR point clouds. The original data contained objects that were unlabeled as to what object they represented. To support supervised training of the network, we manually labeled each object with one of seven object classes known to be present in the scenes.

For training, the point clouds were split into train, validation, and test sets with 25 point clouds in train, 9 in validation, and the remaining 7 in test. The distribution of classes, as shown in Fig.2, displayed some class imbalance. Further details of the training process can be found in[4] and its references. Once the network is trained, inference proceeds as follows. A specific point cloud is processed by the neural network, producing
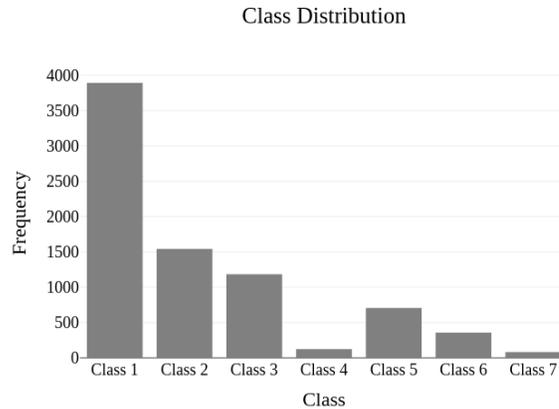
Figure 2. Histogram of object classes

as output a regression map and probability map for each class of object, as well as for an additional "class" representing background. The first step is to create a mask to identify locations in the scene where there is likely to be an object, using the background map to identify regions not belonging to the background class. This background exclusion mask is then applied to each of the other class probability maps. For each object, the class yielding the highest probability score is identified as the object class.

To supply appropriate information for the BBN, each detection is converted to a Bernoulli random variable using the produced probability map. By selecting the probability for the class of interest $p(a)$, and computing its compliment, $1 - p(a)$, the detection is now a Bernoulli random variable with $p = p(a)$ and $q = 1 - p(a)$. Details of the probability map will be discussed in the next section.

3. MERGING WITH BAYESIAN INFERENCE

A Bayesian belief network (BBN)[8,9] is a network (specifically, a directed graph) of parent, intermediate, and child nodes, with the state of each node described by a discrete or continuous probability distribution, and with conditional probabilistic dependencies between connected nodes. Connected nodes can be relative parents or relative children, depending on the direction of conditional probability (child nodes depend conditionally on the state of their parents). Nodes in the network that are not connected can be treated as conditionally independent.

Figure 3 shows an example of a BBN. In this example, nodes P1 and P2 represent alternative hypotheses, nodes I1, I2, and I3 represent intermediate evidential states, and nodes C1, C2, C3, C4 represent nodes whose state is described by externally derived evidence. The states of the intermediate nodes I1, I2, I3 conditionally depend on the states of their parent nodes P1 and P2, as specified in conditional probability tables (CPTs) for each intermediate node. Similarly, the states of C1, C2, C3, and C4 are conditionally dependent on the states of their respective intermediate (relative parent) nodes, as well as the external evidence. The arrows in the diagram represent the flow of external or "diagnostic" evidence, from relative child to relative parent node, in the network – there are also flows of "causal" evidence from relative parent to relative child within the network, representing the effects of conditional probability, not shown on the diagram.

The goal of a BBN is, given the state of the child nodes (which may depend on an external process), to compute the posterior probability of the states of the intermediate and parent nodes, considering conditional dependencies between adjacent nodes. In our example, we want to compute the posterior probability of the alternative hypotheses P1 and P2.

In principle this computation could be done with successive application of Bayes Theorem to the nodes of the network. However, this procedure for a general non-polytree BBN is known to be NP-hard to compute.[10] In practice, solutions of non-polytree BBNs are found using one of several approximation algorithms. These algorithms take advantage of the fact that, while immediately adjacent (connected) nodes in the network are

conditionally dependent on one another, more distant (unconnected) nodes may be regarded as conditionally independent. Approximation algorithms used for this purpose include Markov Chain Monte Carlo (MCMC),[11] which is suited to BBNs with nodes described by either discrete or continuous probability distributions, but computationally intensive. Other approximate algorithms are more suited to, or restricted to, nodes described by continuous probability distributions; these are beyond the scope of the current paper.
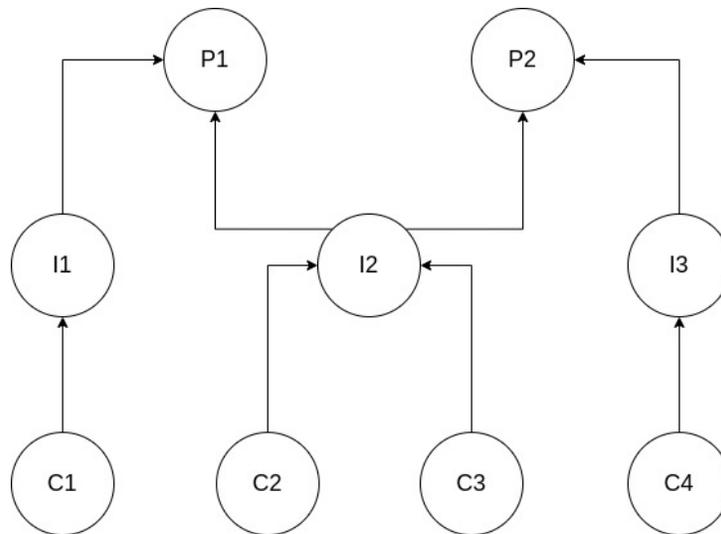

Figure 3. Example Bayesian Belief network.

For BBNs that are polytrees, the solution to the network can be computed exactly using an algorithm originally due to Pearl[12] known as belief propagation (BP). The BP algorithm is a discrete node algorithm – the state of each node in the network must be described with a discrete probability distribution. Examples include the Bernoulli distribution and the categorical distribution. The BP algorithm is highly computationally efficient ($O(N)$ in the number of nodes N), but it is limited to a BBN that is a polytree. EXP's Shadow Compass system implements both Pearl's BP method for polytree networks, , and MCMC for non-polytree networks, composed of Bernoulli or categorical nodes.

In the current work, the external evidence for the state of the child nodes is derived from the results of IC3D object inference. The child nodes represent observational probability of several objects of Class X being present. The Bayesian network captures the significance of that observation in the context of the scenario being considered. Our example scenario will be constructed such that belief propagation can be applied (i.e., as a polytree).

The process of converting object inference scores to Bayesian observational probabilities is as follows:

First, the individual object inference scores from IC3D are softmax scores for each of the object classes, computed as described in the Sec.2. The attainable values of these scores in training and inference can vary by object class, due to variance in the number of examples of each class in the training set. As in Section 2, the softmax scores form a vector of size k where k corresponds to the number of classes that any detected object could be classified as.

Next, assuming we are concerned with one specific class (say Class X), we use the probability score for that object as the probability of success in a Bernoulli "trial" where success is defined as "object of Class X is present". If we wanted to consider several classes jointly for each object, we would convert the softmax scores for each class on an object to a categorical "trial" probability over the classes. For a given inference run there are n detections D and we are interested in converting $D_j for j = 1, 2, ..., n$ to a Bernoulli random variable $I_j \sim Bern(p_j)$ where p_j is the probability of seeing our class of interest X in the jth detection

$$D_j = Softmax(z)$$

$$p_j(x) = D_j(z_x)$$
$$I_j \sim Bern(p_j(x))$$

where size of $z = k$ and x is our class of interest.

Finally, for a child node in the Bayesian network representing the K successes of Bernoulli trials of "detecting object of Class X", each with a different probability, we compute the cumulative probability of K successes out of N detections using a Poisson Binomial Distribution.

We obtain the cumulative distribution function by applying the Discrete Fourier Transform to the Poisson Binomial Characteristic function using the technique developed by Hong.[13] For a set of independent non-identically distributed random variables $I_j \sim Bern(p_j)$ for $j = 1, ..., n$. The CDF of a Poisson Random Variable $F_N(k)$ where $N = \sum_{j=1}^{n} I_j$ is

$$F_N(k) = \sum_{m=0}^{k} \xi_m = \frac{1}{n+1} \sum_{l=0}^{n} \sum_{k=0}^{m} \exp(-i\omega lk)x_l \qquad (1)$$

where $x_l = \prod_{j=1}^{n}[1 - p_j + p_j \exp(i\omega l)]$ and $\omega = \frac{2\pi}{n+1}$. Here x_l is the characteristic function of $F_n(k)$ and ξ_k represents the PMF of N, $p_n(k)$

For a vector of object classes, treating each object as a categorical trial, for N objects, we would compute cumulative probability using Poisson Multinomial Distribution. The data set used in this work was found to have sufficient training exemplars for only one of the classes, such that the aggregate probability scores were sufficiently greater than chance. As such scores are needed to drive a BBN posterior probability computation, we did not consider the categorical case at present.

4. EXAMPLE ANALYST SCENARIO AND RESULTS

Figure 4 is an example of an analyst scenario using a Bayesian network. This scenario focuses on an analyst examining the reason for vehicle traffic in an industrial area. The analyst is interested in the likelihood that traffic is caused by rush hour or an unusual event (e.g., car accident, fallen trees and/or powerlines). We use Class 1, the most populated class from our data set, as the input class – for purposes of this scenario, we treat this class as representing small passenger vehicles.

Before Pearl Belief propagation can be used to calculate the posterior probabilities of the main hypothesis (rush hour traffic) and alternate hypothesis (unusual event), the analyst must set prior and conditional probabilities of each state of the nodes to the network. To add interpretive meaning to these states and their prior and conditional probabilities, we define the states of the parent hypotheses as True and False (T, F), the intermediate nodes are defined by Routine and Unusual (R, U) states and the point cloud inference-based traffic density indicators are defined by (High, Low) states. We posit that the prior and conditional probability values shown would have been derived from analyst assessments of disparate data sources such as historical traffic studies, analysis of media and social media accounts of traffic effects, or historical weather reports. Prior and Conditional probabilities are outlined in Tables 1–5
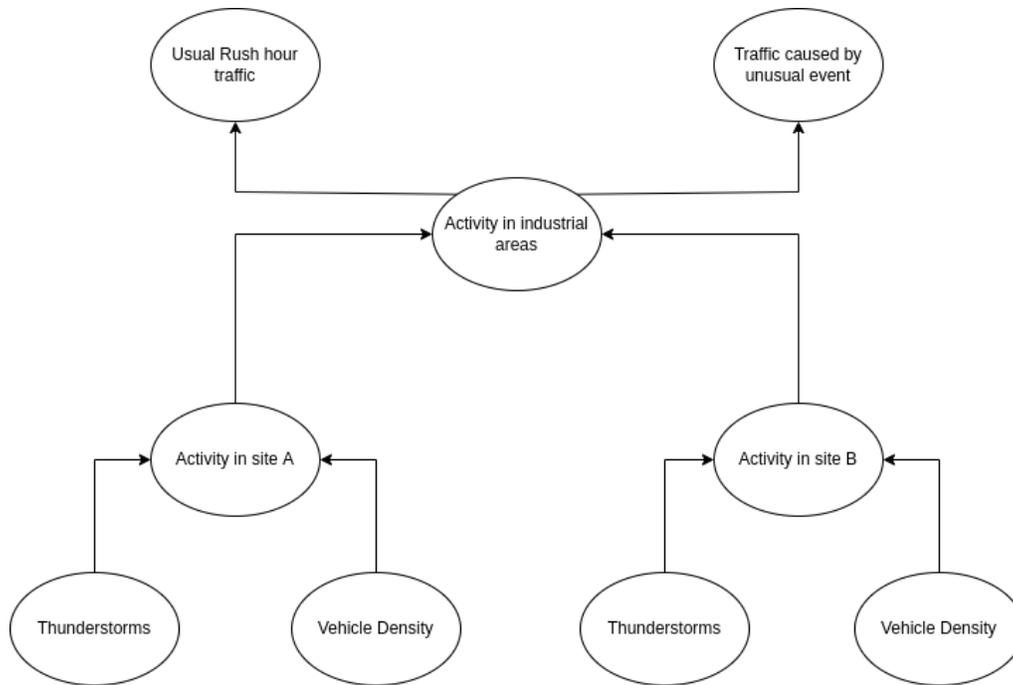
Figure 4. Example of a scenario translated to a Bayesian Network

Table 1. Prior probabilities for network main and alternate parent hypotheses

Parent	Prior Probability (True, False)
Unusual Rush Hour Traffic	(0.80, 20)
Traffic caused by Unusual Event	(0.20, 0.80)

Table 2. Conditional probabilities for Activity in Industrial Areas

Condition	Conditional Probability (R, U)
Rush Hour = T, Event = T	(0.01, 0.99)
Rush Hour = T, Event = T	(0.99, 0.01)
Rush Hour = F, Event = F	(0.01, 0.99)
Rush Hour = F, Event = F	(0.70, 0.30)

Table 3. Conditional probabilities for Activity in Sites A and B

Condition	Site A Probability (R, U)	Site B Probability (R, U)
Activity in Industrial Areas = R	(0.80, 0.20)	(0.99, 0.01)
Activity in Industrial Areas = U	(0.20, 0.80)	(0.01, 0.99)

Table 4. Conditional probabilities for Vehicle Density in Sites A and B

Condition	Site A Probability (H, L)	Site B Probability (H, L)
Activity in Site A = R	(0.85, 0.15)	(0.80, 0.20)
Activity in Site A = U	(0.20, 0.80)	(0.01, 0.99)

Table 5. Conditional probabilities for Thunderstorms in Sites A and B

Condition	Site A Probability (H, L)	Site B Probability (H, L)
Activity in Site A = R	(0.10, 0.90)	(0.25, 0.75)
Activity in Site A = U	(0.95, 0.05)	(0.99, 0.01)

Data that will inform the posterior probabilities of the model come from the set of indicators (leaf nodes) for Sites A and B. First, the point clouds representing the two industrial sites go through IC3D inference and the softmax scores for Class 1 are used to represent the set of Bernoulli random variables that parameterize the Poisson Binomial distribution for each location. The observational probabilities for vehicle density in Sites A and B are $F_{N_{SiteA}}(K > 130)$ and $F_{N_{SiteB}}(K > 130)$. Thunderstorms in Sites A and B are not point cloud related indicators so the analyst must manually enter observational probabilities based on a weather service. Once observational probabilities are set, posterior probabilities are calculated using Pearl Belief Propagation. The results are shown in Tabs. 6 and 7

Table 6. Observational Probabilities for Indicators

Indicator	Observational Probability (H, L)
Site A Thunderstorms	(0.70, 0.30)
Site B Thunderstorms	(0.99, 0.01)
Site A Vehicle Density	(1.00, 0.00)
Site B Vehicle Density	(0.01, 0.99)

Table 7. Posterior Probabilities for Internal and Root Nodes

Node	Posterior Probability
Rush Hour (T, F)	(0.70, 0.30)
Unusual Event (T, F)	(0.61, 0.39)
Industrial Activity (R, U)	(0.23, 0.77)
Activity in Site A (R, U)	(0.47, 0.53)
Activity in Site B (R, U)	(0.19, 0.81)

The analyst observed that thunderstorms are extremely likely to have occurred in Site B and likely to have occurred in Site A. Point cloud inference results show that there is high vehicle density in Site A and low vehicle density in Site B. The posterior probabilities computed by propagation through the network show these effects:

- a posterior belief that there is unusual vehicle activity at Site B of 0.81 (the posterior belief in routine activity being 0.19)

- a posterior belief of unusual vehicle activity at Site A of 0.52

- a joint posterior belief of unusual vehicle activity in industrial areas of 0.77

- a decreased belief in the hypothesis of usual rush hour traffic, compared to the prior probability

- an increased belief in the hypothesis of an unusual traffic event compared to the prior probability of such an event

Though the posterior probability of traffic due to normal rush hour conditions is still greater than that of an unusual event, the latter is enhanced enough to prompt additional observation or concern that there is an unusual event. The typical response of an analyst may consist of

- adjusting prior probabilities of this model to reflect the new evidence and rerunning the model with additional point cloud data, if available, to refine the base and alternative hypothesis estimates

- constructing additional Bayesian models considering other available evidence to support or refute the hypotheses

5. DISCUSSION

This work demonstrates using Bayesian networks with nodes fed by Bernoulli distributed probabilistic indicators derived from point cloud machine learning object inferences. The BBN is a means of establishing meaning and context for object identifications and fusing them with data and insights from other information modalities, both machine and human driven. Our future work will include developing more complex scenarios, in concert with evolving LiDAR and SAR point cloud data sets and the evolution of IC3D machine learning techniques. Due to limitations on class probability scores due to number of training examples we were unable to present results with point cloud machine learning viewed as categorical decisions, but with richer point cloud data sets, our methods can be easily extended to such cases. In the Bayesian portion of our setup, future work may include extending the current human constructed Bayesian network architectures, and setting of conditional probabilities, to include automated methods of constructing these graphs.

ACKNOWLEDGMENTS

REFERENCES

[1] Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, M. M. A., Prabhat, and Adams, R. P., "Scalable bayesian optimization using deep neural networks," *arXiv preprint arXiv:1502.05700* (2015).

[2] Shridhar, K., Laumann, F., and Liwicki, M., "A comprehensive guide to bayesian convolutional neural network with variational inference," *arXiv preprint arXiv:1901.02731* (2019).

[3] Zhou, Y. and Tuzel, O., "Voxelnet: End-to-end learning for point cloud based 3d object detection," in [*In Proceedings of the IEEE conference on computer vision and pattern recognition*], 4490–4499 (2018).

[4] Moses, A., Jakkampudi, S., Danner, C., and Biega, D., "Localization-based active learning (local) for object detection in 3d point clouds," in [*Defense + Commercial Sensing*], *Proc. SPIE* (2022).

[5] Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P., "Focal loss for dense object detection," in [*Proceedings of the IEEE international conference on computer vision*], 2980–2988 (2017).

[6] Liu, H., Simonyan, K., and Yang, Y., "Darts: Differentiable architecture search," in [*International Conference On Learned Representations*], (2019).

[7] Zhou, X., Wang, D., and Krähenbühl, P., "Objects as points," *arXiv preprint arXiv:1904.07850* (2019).

[8] Koller, D. and Friedman, N., [*Probabilistic Graphical Models*], MIT Press (2019).

[9] Krieg, M., "A tutorial on bayesian belief networks," in [*Technical Report DSTO-TN-0403*], *Defense Science and Technology Organization* (2001).

[10] Cooper, G. F., "The computational complexity of probabilistic inference using bayesian belief networks," *Artificial Intelligence* **42**(2), 393–405 (1990).

[11] Gelman, A., Carlin, J., Venhatri, A., Dunson, D., Ruben, D., and Stern, H. S., [*Bayesian Data Analysis*], Chapman and Hall/CRC, 3rd ed. ed. (2013).

[12] Kim, J. H. and Pearl, J., "A computational model for causal and diagnostic reasoning in inference systems," *IJCAI'83*, 190–193, Morgan Kaufmann Publishers Inc. (1983).

[13] Hong, Y., "On computing the distribution function for the poisson binomial distribution," *Computational Statistics & Data Analysis* **59**, 41–51 (2013).