

Using Test Plans for Bayesian Modeling [★]

Rainer Deventer¹, Joachim Denzler¹, Heinrich Niemann¹, and Oliver Kreis²

¹ Chair for pattern recognition

² Chair of manufacturing technology

Friedrich–Alexander–Universität Erlangen–Nürnberg, 91058 Erlangen, Germany

Abstract. When modeling technical processes, the training data regularly come from test plans, to reduce the number of experiments and to save time and costs. On the other hand, this leads to unobserved combinations of the input variables. In this article it is shown, that these unobserved configurations might lead to un-trainable parameters. Afterwards a possible design criterion is introduced, which avoids this drawback. Our approach is tested to model a welding process. The results show, that hybrid Bayesian networks are able to deal with yet unobserved in- and output data.

Keywords: Bayesian Network, Modeling, Manufacturing process, Laser-beam welding

1 Introduction

Modeling of technical processes is applied in many parts of industrial everyday life, e.g. in model based control and quality management. Particularly in the domain of manufacturing, there are two, closely intertwined problems. We have to tackle with few, in most of the cases incomplete, data. The reason is the way tests are executed. Each test causes expenses for material and for the staff, carrying out the experiments. This cost pressure leads to test plans [1, 10]. The main idea is to make less experiments, knowing that the used test plan is not able to reveal all possible interdependencies. Particularly interdependencies between multiple parameters are neglected. Therefore only in seldom cases all possible combinations of variables are tested.

Modeling with Bayesian networks(BN) has many advantages, e.g. the possibility of structure learning and to deal with hidden variables. But one of the drawbacks in Bayesian modeling is the sensitivity to missing combinations, if parameters, representing these combinations, are part of the network. In the training process, these parameters are trained as usual, that is depending on the frequency of the occurring cases. When the output for a yet unrepresented example has to be predicted, this results in a faulty output.

As the usage of discrete nodes causes this phenomenon, a possible solution would be the restriction to continuous nodes, with the disadvantage, that only

[★] This work was funded by the “German Research Association” (DFG), Collaborative research center (SFB) 396, project-parts C1 and C3. Only the authors are responsible for the content of this article

linear functions could be modeled. Additionally it is not possible to treat each random variable as continuous random variable. The other possibility is to rely purely on a-priori knowledge. But in most of the cases it will be hard to get exact a-priori knowledge. This article shows the critical structures, to be deduced from the test plan, and discusses possible workarounds. The method is applied to the problem of modeling a welding process with 6 input parameters and one output parameter. Some of the parameters are discrete, others are continuous, thus a hybrid Bayesian network[2, 7, 8] is used for modeling. This enables us to model also nonlinearities. In our example two different methods are used. For one input parameter, the squared value is used as additional input. On the other side it is possible to represent a variable both as discrete and continuous random variable. Using this method also steep slopes can be modeled, e.g. the failure of the welding process. The combination of discrete and continuous nodes allows also the approximation of nonlinear functions by multiple Taylor series [3].

This article is structured as follows. Section 2 gives a brief introduction to hybrid Bayesian networks, section 3 deals with modeling of manufacturing data derived from test plans. Afterwards, in section 4, the data to be modeled are introduced. Section 4 is followed by the applied model and the obtained results. In comparison to other modeling methods, like neural networks and classification trees, we are able to predict both input and output signals with the same model, as the Bayesian network represents a joint distribution of all in- and output parameters, so that arbitrary variables are predicted by marginalization. The article finishes with a conclusion which contains an outlook to further research.

2 Bayesian Networks

Bayesian networks represent a multivariate distribution $P(X_1, X_2, \dots, X_n)$ of random variables X_1, X_2, \dots, X_n . In this article P denotes a distribution of discrete random variables, p is used for continuous ones. Using the Bayes rule, the probability of a configuration x_1, \dots, x_n , i.e. an instantiation $X_i = x_i$ of every random variable, can be calculated as a product of conditional probabilities [15]

$$P(x_1, x_2, \dots, x_n) = P(x_1) \prod_{i=2}^n P(x_i | x_{i-1}, \dots, x_1) . \quad (1)$$

In many cases, X_i does not depend on all random variables X_1, \dots, X_{i-1} , but only on a subset $\mathbf{Pa}(X_i) \subseteq \{X_1, \dots, X_{i-1}\}$, called the parents $\mathbf{Pa}(X_i)$ of X_i . Using these independencies the chain rule (1) rewrites to

$$P(x_1, x_2, \dots, x_n) = P(x_1) \prod_{i=2}^n P(x_i | \mathbf{pa}(X_i)) , \quad (2)$$

where $\mathbf{pa}(X_i)$ denotes the instantiation of $\mathbf{Pa}(X_i)$. Usually the dependencies between random variables are represented in a acyclic graph, with the random variables as nodes and directed edges from the parents $\mathbf{Pa}(X_i)$ to X_i . As an

example, figure 1 might be used. The parents of node H are $\{X_1, X_2, X_3, X_4\}$, thus there are edges $X_i \rightarrow H$ from X_i to H . It is assumed, that H is a hidden node, which are drawn in shaded manner in this article. To distinguish discrete nodes from continuous ones, the former are drawn as circle or ellipse, the latter as square or rectangle. The next section discusses so called hybrid Bayesian networks, where discrete and continuous nodes are used at the same time.

2.1 Hybrid Bayesian Networks

At the beginning of the development of BNs, only networks with discrete nodes were used. That means that discretization is needed for all continuous variables. Additionally, a great number of parameters is required, to describe exactly a BN with discrete nodes. If only continuous nodes are regarded, it is possible to use a Gaussian network instead, where normal distributions are associated with every random variable, whose mean is calculated as linear combination of its predecessor's values. I.e. the distribution p of a random variable X with parents \mathbf{Y} is

$$p(x|\mathbf{y}) = \mathcal{N}(\mu_{X_0} + \mathbf{w}_X \mathbf{y}, \sigma_X) \quad (3)$$

with \mathcal{N} as the one-dimensional normal distribution. When $\mathbf{y} = 0$ the mean of the normal distribution is μ_{X_0} , \mathbf{w}_X is the weight vector between \mathbf{Y} and X . Of course, it is possible to regard X also as a multidimensional random variable, but, for the purpose of the article, it is sufficient to use a one-dimensional distribution. If not only continuous variables are used, or if non-linearities are required, these needs are met by hybrid BNs as described in [7, 8, 11, 14].

The set of nodes of a hybrid BN contains both discrete and continuous nodes. Discrete nodes, having only discrete predecessors, are handled as usual. I.e. each node X stores the conditional probabilities $P(X|\mathbf{Pa}(X))$ in a table, which is used for calculation of joint and marginal distributions. Major changes are made for continuous nodes, having both discrete and continuous predecessors. As in Gaussian networks, the values of continuous nodes are still assumed to be normal distributed, but this time a mixture of normal distribution is used with $P(\mathbf{x}_p)$, the probability of the parents having configuration \mathbf{x}_p , as the mixing coefficients. As defined in [4], a configuration for a set of nodes is a set of states with exactly one state for each variable. Therefore, there are different means $\mu_{X_0}[\mathbf{x}_p]$, weights $w_X[\mathbf{x}_p]$ and standard deviations $\sigma_X[\mathbf{x}_p]$ for every possible configuration. The distribution of node X , given that the continuous parents \mathbf{Y} have value \mathbf{y} and configuration \mathbf{x}_p for the discrete parents, is

$$p(x|\mathbf{y}, \mathbf{x}_p) = P(\mathbf{x}_p) \mathcal{N}(\mu_{X_0}[\mathbf{x}_p] + \mathbf{w}_X[\mathbf{x}_p] \mathbf{y}, \sigma_X[\mathbf{x}_p]) \cdot \quad (4)$$

If a continuous node has no discrete parents, there is only one possible configuration, and the equation is reduced to the pure Gaussian case. It remains the problem, whether discrete nodes are allowed to have continuous parents. Some authors, e.g. Lauritzen [7] [8] and Olesen [14] assume, that there are no such nodes allowed, which simplifies training of BNs. At the moment there are two main approaches, discussed e.g. in [12], to deal with continuous predecessors of

discrete nodes. In [9] the junction tree algorithm, used for inference in BNs, is expanded for BNs with continuous nodes as predecessors of discrete nodes.

3 Modeling of Manufacturing Processes

In section 2.1 an introduction to the parameterization of hybrid Bayesian networks is given. The reader should keep in mind, that for every possible configuration \mathbf{x}_p of the parent nodes there is a set of parameters to be adapted. For a discrete node X this is usually a table with conditional probabilities $P(x|\mathbf{x}_p)$, for continuous nodes the weights $\mathbf{w}[\mathbf{x}_p]$, means $\mu[\mathbf{x}_p]$ and dispersions $\sigma[\mathbf{x}_p]$ have to be stored. This section gives a brief overview about test plans, mainly to show that the used test plan leads to special restrictions on the used structure.

Test plans are used to save time and money for a lot of tests, which might be redundant. The reason might be, that also the influence of parameter combinations is explored, which have no effect on the output parameter, according to the engineer, executing the test. Imagine a simple, linear process with four input parameters X_1, \dots, X_4 , and one output Y . Suppose that combinations of three different variables have no influence on the output of the experiment. In this case, the test plan in table 1 can be used, where the input of X_4 is calculated as the product of X_1 to X_3 . The symbols '-' and '+' represent a low respectively

Table 1. Example of a simple test plan

X_1	X_2	X_3	$X_4 = X_1 \cdot X_2 \cdot X_3$	Y
-	-	-	-	-
-	-	+	+	+
-	+	-	+	+
-	+	+	-	-
+	-	-	+	+
+	-	+	-	-
+	+	-	-	-
+	+	+	+	+

a high instantiation of the random variables. The product \cdot is defined, so that the product of two different values is negative, the product of two equal values is positive. As X_4 is the product of X_1, \dots, X_3 , there is no way to distinguish between the influence of X_4 and the combined influence of X_1, \dots, X_3 , which is no drawback, as it was supposed that $X_1 \cdot X_2 \cdot X_3$ has no influence on the outcome of the experiment. On the other hand, 50% of the experiments are saved. As test plans are applied regularly to explore the interdependencies within a process, it is necessary to discuss the effect of test plans in Bayesian modeling. Simple BNs to represent the influence of X_i on Y are depicted in figures 1 and 2. The hidden node H on the left hand side has as parameter a table with all

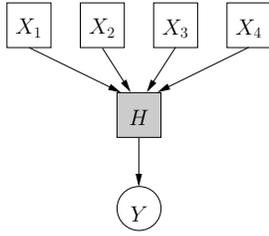


Fig. 1. Critical node H

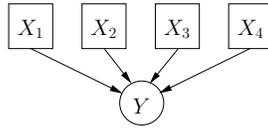


Fig. 2. Critical node Y

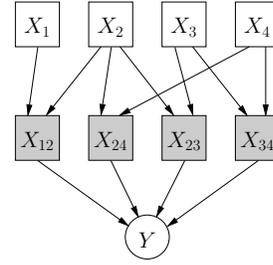


Fig. 3. Robust model

conditional probabilities $P(h|x_1, x_2, x_3, x_4)$, including e.g. $P(h|---)$, which represents a never tested combination.

The same problem occurs in figure 2, where the parameters $\mu_Y[x_1, x_2, x_3, x_4]$ and $\sigma_Y[x_1, x_2, x_3, x_4]$ can not be trained. Of course a pure continuous model will solve the problem, but in this case only linear models can be represented.

Models, as depicted in figure 3, might help. They show the following features:

- There is no node representing the influence of a combination of 3 variables. When designing the test plan, it is concluded, that these combinations have no influence on Y .
- For all nodes with discrete parents all configurations of the parents are observed.

Thus the preconditions for a robust model are fulfilled. The reader should notice that this consideration provides also a criterion for structure learning, which can easily be tested. The variance analysis, usually used to evaluate tests, provides further hints for the structure of the Bayesian network.

The principles discussed in this section are applied to develop a model of the welding process discussed in section 4. The resulting model, together with the result, is presented in section 5.

4 Welding

The welding process is part of a shortened process chain for the manufacturing of complex, hollow bodies, that consists of the processes hydroforming, cutting, and joining by laser beam welding. Two blanks are inserted in a hydroforming tool, formed by pressing fluid between the blanks and cutted by shear cutting. The next process is the welding of the flange by a welding robot that is integrated in the forming tool. For a description of the complete process chain see [5, 6].

The first input parameter of the welding process is the type of weld (Confer figure 4 for a complete list of the used parameters). Lap edge joint and lap seam joint were investigated. Also, the number of welds is from interest. To achieve a closed impermeable weld, the beginning of the weld has to be welded again at

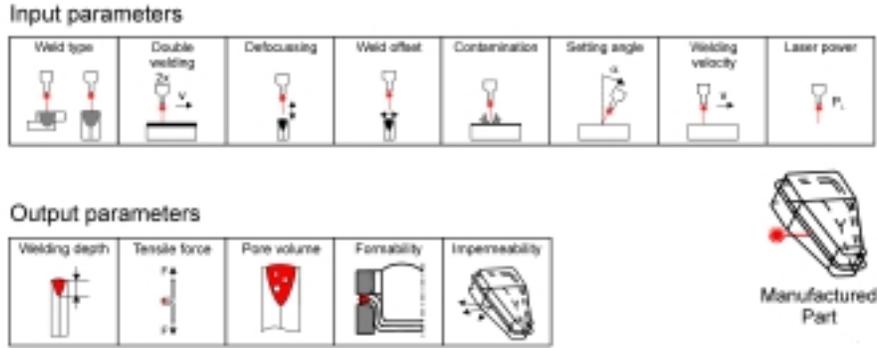


Fig. 4. In- and output parameters of welding process

the end of the process. This double welding may have a negative effect on the welding quality. Furthermore, the necessary accuracy of the welding robot has to be examined by determining the influence of defocussing and of a weld offset. The results of these experiments are depicted in figure 5. Since the welding process takes place just after the hydroforming and cutting, the blanks are contaminated with hydroforming media and lubricant. The effect of this contamination has to be determined. To ensure a constant welding velocity, a setting angle of 30° must be applied in the corners of the flange. The last and surely most important input factor is the welding velocity. The effect of the velocity is displayed in figure 6. The output parameters of the welding process are welding depth, tensile force, pore volume, formability and impermeability of the weld. In this article, only the dependency of the tensile force on the input parameter is modeled.

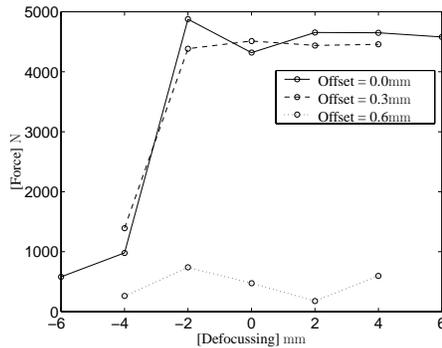


Fig. 5. Tensile force F depending on the offset

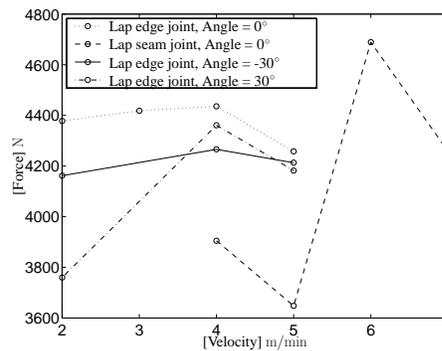


Fig. 6. Tensile force F depending on the velocity

5 Results

The principles discussed in section 3 are applied to model the welding process, introduced in section 4. A variance analysis shows, that the contamination has nearly no effect on the tensile force and that both, the angle, and the type of joint, have an influence on the tensile force. Thus, there is no node for the contamination.

When developing the structure of the BN, the problem occurs, that there are no data available for the lap seam joint, together with an angle of $+/- 30^\circ$. So the problem of untrainable parameter occurs. To avoid the risk of failure, the two unobserved configurations are mapped with equal probability to neighboured configurations. E.g. the lap seam joint with an angle = 30° is mapped to the lap-edge joint with an angle of 30° , and the lap seam joint with 0° . This mapping is done by the deterministic node H_2 . This means, that the conditional probability table, which determines the behaviour of H_2 is not changed during training.

Figure 6 shows, that the tensile force depends nonlinearly on the velocity v . To enable the model to learn this nonlinearity, an additional node, representing the squared velocity, is added. The mean of node F_{H1} and the weights of the edges $v \rightarrow F_{H1}$, and $v^2 \rightarrow F_{H1}$, are initialised, so that F is approximated by a regression polynomial of second order. The used model is depicted in figure 7, the obtained results for an angle of 0° are shown in figure 8. The tensile force

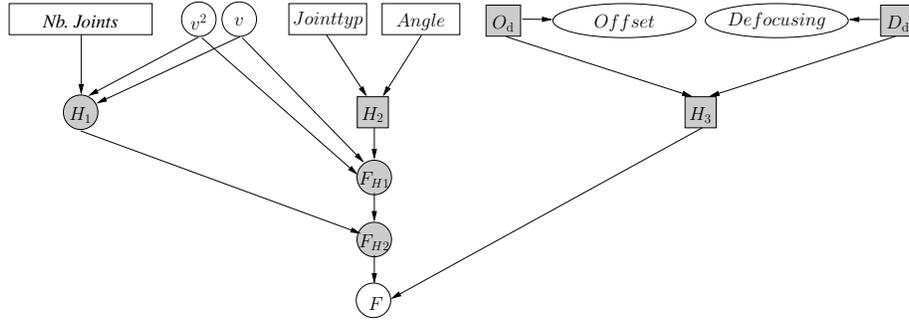


Fig. 7. Model for the tensile force

depends also on the number of joints. Node H_1 represents the difference of the tensile force for a second joint. That is $H_1 = 0$ for one joint and larger than 0 for a second joint. The results of F_{H1} and H_1 are added, i.e. both links $H_1 \rightarrow F_{H2}$ and $F_{H1} \rightarrow F_{H2}$ are initialised to 1, so that F_{H2} represents the tensile force for intact seams.

The failure of the seam is caused by the offset or defocussing being larger than a threshold. This threshold is represented by the two means of *Offset*

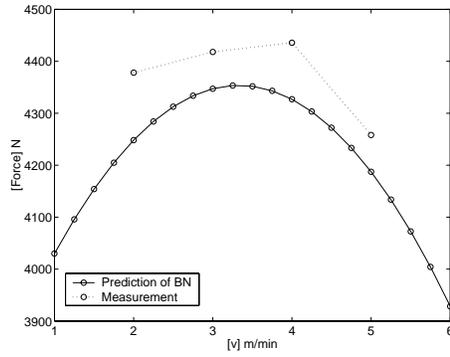


Fig. 8. Tensile force F depending on the velocity

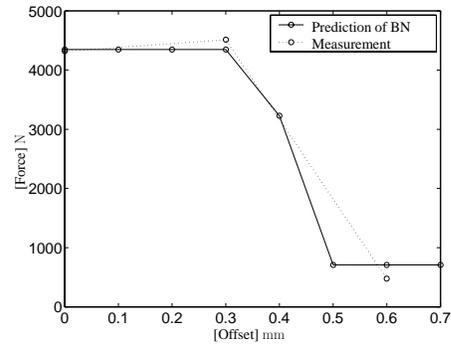


Fig. 9. Tensile force F depending on the offset

and *Defocusing*, triggered by the nodes O_d and D_d respectively. Again, a good initialisation is essentially.

The discrete node H_3 has two states, representing an intact or defective joint, which triggers the node F . In case of an intact joint the tensile force is determined by F_{H2} , otherwise the node F predicts approximately the mean of the tensile force of all defective joints. For a comparison of the predicted tensile force, depending on the offset and on the defocussing, see figures 9 and 10. The results in figure 10 are best for an offset of 0 mm, as the largest part of the experiments are executed with that offset.

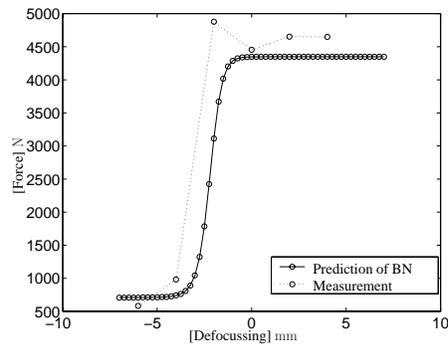


Fig. 10. Tensile force depending on defocussing

To test our model, the Bayesian network was trained with the EM algorithm, which is already implemented in the BN-toolbox [13], which was used for the experiments described in this article.

As training data, we got 48 blocks with 6 examples each. To test, whether our model is able to make predictions also for unseen configurations, we trained our net with 47 blocks. Afterwards we compare the predictions with the measured values of the remaining block. To calculate the predictions, all the remaining, observed, variables are entered as evidence and the mean of the marginal distribution is taken as prediction. For continuous random variables (v, F), we used the relative error

$$e_r^c = \frac{|v_m - v_p|}{v_m} 100\% \quad (5)$$

as quality criterion. In equation (5) v_m denotes the measured value, and v_p the predicted one. For discrete random variables (Number of joints, joint type, angle), the error is defined as quotient between the number of misclassifications n_m and the total number of classifications n_t .

$$e_r^d = \frac{n_m}{n_t} 100\% . \quad (6)$$

The results are given in table 2. For the offset and defocusing exact predictions

Table 2. Relative error

Variable	Number of joints	Velocity	Angle	Joint type	Tensile force
e_r	37.7%	22.1%	50.4%	15.5 %	17.04%

can not be expected. Only predictions for the equivalence class, e.g. failure caused by offset or not, can be made. For the offset there are 3 misclassifications in the 48 blocks tested. For the defocusing 2 blocks are not correctly classified.

6 Conclusion

The usage of test plans is widely spread in manufacturing. Even simple test plans result in unobserved configurations, as the number of experiments grows exponentially with the number of variables.

This article has shown, that unskilful modeling might lead to a complete failure of the model. In contrary, when all configurations of discrete parents are observed, this results in a stable model. This principle is applied to the modeling of a welding process. The results show, that the discussed model is able to deal with evidences, not seen before, e.g. our model is able to predict the tensile force of yet unobserved velocities and offsets. In comparison to neural networks, a Bayesian network is also able to predict input variables, when the output is given. The price for this advantage is a higher effort for modeling, even if there are lot of structure learning algorithms available.

References

1. Larry B. Barrentine. *Introduction to Design of Experiments: A Simplified Approach*. ASQ Quality Press, Milwaukee, 1999.
2. R. Cowell. Advanced inference in bayesian networks. In M. I. Jordan, editor, *Learning in graphical models*, pages 27 – 49. MIT Press, Cambridge, Massachusetts, 1999.
3. R. Deventer, J. Denzler, and H. Niemann. Non-linear modeling of a production process by hybrid Bayesian Networks. In Werner Horn, editor, *ECAI 2000 (Berlin)*, pages 576–580. IOS Press, August 2000.
4. F. V. Jensen. *An introduction to Bayesian networks*. UCL Press, 1996.
5. O. Kreis and Ph. Hein. Manufacturing system for the integrated hydroforming, trimming and welding of sheet metal pairs. *Journal of Materials Processing Technology*, 115:49–54, 2001.
6. Oliver Kreis. *Integrierte Fertigung - Verfahrensintegration durch Innenhochdruck-Umformen, Trennen und Laserstrahlschweißen in einem Werkzeug sowie ihre tele- und multimediale Präsentation*. PhD thesis, Universität Erlangen-Nürnberg, Meisenbach, Bamberg, 2002. <http://www.integrierte-fertigung.de>.
7. S. L. Lauritzen. Propagation of probabilities, means, and variances in mixed graphical association models. *Journal of the American Statistical Association*, Vol. 87(420):1098 – 1108, December 1992.
8. S. L. Lauritzen and F. Jensen. Stable Local Computation with Conditional Gaussian Distributions. Technical Report R-99-2014, Aalborg University, Department of Mathematical Sciences, September 1999.
9. Uri Lerner, Eran Segal, and Daphne Koller. Exact inference in networks with discrete children of continuous parents. In *Uncertainty in Artificial Intelligence: Proceedings of the Seventeenth Conference (UAI-2001)*, pages 319–328, San Francisco, CA, 2001. Morgan Kaufmann Publishers.
10. Douglas C. Montgomery. *Design and Analysis of Experiments*. Wiley, New York, fifth edition edition, 2000.
11. Kevin P. Murphy. Inference and Learning in Hybrid Bayesian Networks. Technical Report CSD-98-990, University of California, Computer Science Division (EECS), January 1998.
12. Kevin P. Murphy. A variational approximation for bayesian networks with discrete and continuous latent variables. In Kathryn Blackmond Laskey and Henri Prade, editors, *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, San Francisco, 1999. Morgan Kaufmann Publishers Inc.
13. Kevin P. Murphy. The Bayes Net Toolbox for Matlab. *Computing Science and Statistics*, 33, 2001.
14. K. G. Olesen. Causal probabilistic networks with both discrete and continuous variables. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 3(15):275 –279, 1993.
15. J. Pearl. *Probabilistic reasoning in intelligent systems*. Morgan Kaufmann Publishers, Inc., 1988. ISBN 0-934613-73-7.