Learning Conditional Distributions Using Mixtures of Truncated Basis Functions

Inmaculada Pérez-Bernabé[1](✉), Antonio Salmerón[1], and Helge Langseth[2]

[1] University of Almería, 04120 Almería, Spain
{iperez,antonio.salmeron}@ual.es
[2] Norwegian University of Science and Technology, 7491 Trondheim, Norway
helgel@idi.ntnu.no

Abstract. Mixtures of Truncated Basis Functions (MoTBFs) have recently been proposed for modelling univariate and joint distributions in hybrid Bayesian networks. In this paper we analyse the problem of learning conditional MoTBF distributions from data. Our approach utilizes a new technique for learning joint MoTBF densities, then propose a method for using these to generate the conditional distributions. The main contribution of this work is conveyed through an empirical investigation into the properties of the new learning procedure, where we also compare the merits of our approach to those obtained by other proposals.

Keywords: Mixtures of truncated basis functions · Hybrid bayesian networks · Joint density · Conditional density

1 Introduction

Mixtures of truncated basis functions (MoTBFs) [2] have recently been proposed as a general framework for handling hybrid Bayesian networks, i.e., Bayesian networks where discrete and continuous variables coexist. Previous hybrid models as the so-called mixtures of truncated exponentials (MTEs) [7] and mixtures of polynomials (MoPs) [10] can be regarded as particular cases of MoTBFs.

Part of the success of MoTBFs is due to the fact that they can model hybrid Bayesian networks with no structural restrictions, unlike the conditional Gaussian (CG) model [6], where discrete variables are not allowed to have continuous parents. Furthermore, MoTBFs are closed under addition, multiplication, and integration, which facilitates the use of efficient inference methods like the Shenoy-Shafer architecture [9] or the *variable elimination* algorithm [12].

The problem of learning MoTBFs from data has been studied considerably already (see, e.g., [3,5]). However, even though a Bayesian network model populated with MoTBF distributions requires the specification of both marginal and conditional MoTBF distributions, only limited attention has been given to learning the *conditional* MoTBF distributions directly from data [1,11]. In this paper we first extend previous work on learning marginal MoTBF distributions [5] to also learn joint densities. These are in turn employed to generate the required conditional MoTBFs.

S. Destercke and T. Denoeux (Eds.): ECSQARU 2015, LNAI 9161, pp. 397–406, 2015.
DOI: 10.1007/978-3-319-20807-7_36

The remainder of the paper is organized as follows: The MoTBF model is introduced in Sect. 2. Next, techniques for learning marginal and joint MoTBF densities from data is described in Sect. 3, where we also detail how we define the conditional distributions. The main part of this work is given in Sect. 4, where our proposal is validated through a series of experiments. Finally, we give some conclusions in Sect. 5.

2 The MoTBF Model

The MoTBF framework is based on the abstract notion of real-valued *basis functions* $\psi(\cdot)$, which include both polynomial and exponential functions as special cases. Let X be a continuous variable with domain $\Omega_X \subset \mathbb{R}$ and let $\psi_i : \Omega_X \mapsto \mathbb{R}$, for $i = 0, \ldots, k$, define a collection of real basis functions. We say that a function $f : \Omega_X \mapsto \mathbb{R}_0^+$ is an MoTBF potential of level k wrt. $\Psi = \{\psi_0, \psi_1, \ldots, \psi_k\}$ if f can be written as

$$f(x) = \sum_{i=0}^{k} c_i \, \psi_i \, (x),$$

where c_i are real numbers [2]. The potential is a density if $\int_{\Omega_X} f(x) \, \mathrm{d}x = 1$.

In this paper we will restrict our attention to the MoP framework, meaning that $\psi_i(x) = x^i$.

When there are more than one variable, we can use a joint MoP to capture the probability density function over the variables. Let $\mathbf{X}$ be a d-dimensional continuous variable, $\mathbf{X} = (X_1, \ldots, X_d)$ with domain $\Omega_{\mathbf{X}} \subset \mathbb{R}^d$. A function $f : \Omega_{\mathbf{X}} \mapsto \mathbb{R}^+$ is said to be an MoP potential of level k if it can be written as

$$f(\mathbf{x}) = \sum_{\ell_1=0}^{k} \cdots \sum_{\ell_d=0}^{k} c_{\ell_1, \ell_2, \ldots, \ell_d} \prod_{i=1}^{d} x_i^{\ell_i}, \tag{1}$$

or if there is a partition of $\Omega_{\mathbf{X}}$ into hypercubes where f can be written as in Eq. 1 for each part.

3 Learning MoPs from Data

We will now investigate how to learn MoP distributions for a given set of random variables. We start by looking at how to learn univariate MoP distributions from data, before we extend that approach to learning joint MoP distributions, and finally discuss how one can obtain conditional distribution functions.

3.1 Univariate MoPs

The learning of univariate MoTBFs from data was explored in [5], and we will briefly summarize that approach here in the special case of MoPs. The estimation procedure relies on the empirical cumulative distribution function (CDF) as a representation of the data $\mathcal{D} = \{x_1, \ldots, x_N\}$. The empirical CDF is defined as

$$G_N(x) = \frac{1}{N} \sum_{\ell=1}^{N} \mathbf{1}\{x_\ell \leq x\}, \quad x \in \Omega_X \subset \mathbb{R},$$

where $\mathbf{1}\{\cdot\}$ is the indicator function.

The algorithm in [5] approximates the empirical CDF by a function whose derivative is an MoTBF, using least squares. In our case, the role of the basis functions is taken by the polynomials, and since the integral of a polynomial is itself a polynomial, the target function is of the form $F(x) = \sum_{i=0}^{k} c_i\, x^i$, defined on an interval $\Omega_X = [a, b] \subset \mathbb{R}$. The optimization problem thus becomes

$$\begin{aligned} \textbf{minimize} \; & \sum_{\ell=1}^{N} \left(G_N(x_\ell) - F(x_\ell)\right)^2 \\ \textbf{subject to} \; & \frac{\mathrm{d}F(x)}{\mathrm{d}x} \geq 0 \quad \forall x \in \Omega_X, \\ & F(a) = 0 \text{ and } F(b) = 1. \end{aligned} \tag{2}$$

The probability density function (PDF) is found by simple differentiation of the estimated CDF. The constraints of the optimization program ensures that the result is a legal density; the first requirement ensures that the PDF is non-negative over the domain, the others ensure it integrates to one. Furthermore, [5] remarks that the solution obtained by solving program in Eq. 2 is a consistent estimator of the true CDF in terms of the mean squared error for all $x \in \Omega_X$.

Note that the optimization program is convex, and can be efficiently solved in theory. However, the infinite number of constraints introduced by requiring that $\frac{\mathrm{d}F(x)}{\mathrm{d}x} \geq 0$ for *all* $x \in \Omega_X$ complicates the implementation on a computer. In practice, we therefore only check that the constraint is fulfilled for a limited set of points spread across Ω_X.

In learning situations where we have lots of data (N is large), the solution of the program can be slow. In such cases we rather define a *grid* on Ω_X, where the grid is selected so that the number of observations is the same between each pair of consecutive grid-points. Then, the grid-points will play the role of the evaluation points in the objective function.

The level k of the estimated MoP can be decided using a multitude of different model selection techniques. For the results presented in this paper we have searched greedily for k, and chosen the value that maximized the BIC score [8]. This choice is motivated by [3], who showed that the estimators based on Eq. 2 are consistent in terms of the mean squared error for all $x \in \Omega_X$.

3.2 Joint MoPs

During the definition of the conditional distributions (described in Sect. 3.3), we will investigate the use of joint MoP densities to define conditional distributions. We therefore proceed by extending the program in Eq. 2 to arbitrarily dimensional random vectors. The procedure is very similar to the univariate case. The data now consists of d-dimensional observations, $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, $\mathbf{x} \in \Omega_{\mathbf{X}} \subset \mathbb{R}^d$. We continue to use $\mathbf{1}\{\cdot\}$ to denote the indicator function, and the say that the event $\mathbf{x}_\ell \leq \mathbf{x}$ is true if and only if $\mathbf{x}_{\ell,i} \leq \mathbf{x}_i$ for each dimension $i = 1, \ldots, d$. For notational convenience we use $\Omega_{\mathbf{X}}^- \in \mathbb{R}^d$ to denote the minimal point of $\Omega_{\mathbf{X}}$ (obtained by choosing the minimum of $\Omega_{\mathbf{X}}$ in each dimension), and let $\Omega_{\mathbf{X}}^+ \in \mathbb{R}^d$ be the corresponding maximal point. Then, the empirical CDF is defined as

$$G_N(\mathbf{x}) = \frac{1}{N} \sum_{\ell=1}^{N} \mathbf{1}\{\mathbf{x}_\ell \leq \mathbf{x}\}, \quad \mathbf{x} \in \Omega_{\mathbf{X}} \subset \mathbb{R}^d.$$

Our goal is to find a representation of the empirical CDF of the form

$$F(\mathbf{x}) = \sum_{\ell_1=0}^{k} \cdots \sum_{\ell_d=0}^{k} c_{\ell_1,\ell_2,\ldots,\ell_d} \prod_{i=1}^{d} x_i^{\ell_i},$$

leading us to the optimization problem

$$\begin{aligned} \textbf{minimize} \quad & \sum_{\ell=1}^{N} \left(G_N(\mathbf{x}_\ell) - F(\mathbf{x}_\ell)\right)^2 \\ \textbf{subject to} \quad & \frac{\partial^d F(\mathbf{x})}{\partial x_1, \ldots, \partial x_d} \geq 0 \quad \forall \mathbf{x} \in \Omega_{\mathbf{X}}, \\ & F\left(\Omega_{\mathbf{X}}^-\right) = 0 \text{ and } F\left(\Omega_{\mathbf{X}}^+\right) = 1. \end{aligned} \tag{3}$$

The solution to this problem is the parameter-set that defines the joint CDF, and the density can be obtained simply by differentiation of the joint CDF. As in the univariate case, the problem is a quadratic optimization problem, that can be solved efficiently. When the amount of data and/or the dimensionality get large, we have used the same strategy wrt. grid-points for the joint density as we did when estimating the univariate PDFs.

The top of Fig. 1 shows the MoP density generated by solving the optimization program in Eq. 3. The model was learned from a database of 1000 observation generated from a bivariate standard normal distribution (i.e., with correlation-coefficient $\rho = 0$). In the bottom part of Fig. 1 we can see the model learned from same distributions but with correlation $\rho = 0.99$.

3.3 Conditional Distributions

The last piece of the puzzle is to learn the conditional density functions for a variable X with parents $\mathbf{Z}$, that will be used to populate the Bayesian network structure. Using the minimization program in Eq. 3, we can learn both $f(x, \mathbf{z})$ and $f(\mathbf{z})$, hence by the definition of a conditional probability density it seems natural to define $f(x|\mathbf{z})$ as

$$f(x|\mathbf{z}) \leftarrow \frac{f(x, \mathbf{z})}{f(\mathbf{z})}, \tag{4}$$

where both $f(\mathbf{z})$ and $f(x, \mathbf{z})$ are MoPs. Unfortunately, though, MoPs are not closed under division [2], thus $f(x|\mathbf{z})$ defined by Eq. 4 will not lead to a legal MoP-representation of a conditional density. An alternative was therefore pursued by [2], where the influence the parents $\mathbf{Z}$ have on X was encoded only through the partitioning of the domain of $\mathbf{Z}$ into hyper-cubes. Then, specific distributions for X that are valid as long as $\mathbf{Z}$ is inside a specific hypercube was learned from data.

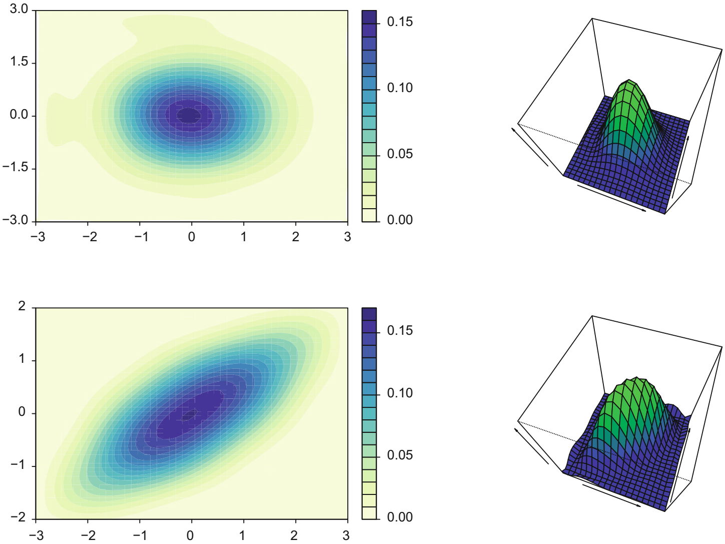


Fig. 1. The contour and the perspective plots of the result of learning a MoP from $N = 1000$ samples drawn from bivariate standard normal distributions with $\rho = 0$ (top) and $\rho = 0.99$ (bottom).

Here, however, we will follow an alternative strategy similar to the one pursued in [11]. The idea is to learn representations for $f(x, \mathbf{z})$ and $f(\mathbf{z})$, then utilize Eq. 4 to calculate $f(x|\mathbf{z})$. As already noted, this will not result in an MoP, and the next step is therefore to approximate this representation into an MoP by some means. Varando et al. [11] investigated two schemes: *i*) To use the representation in Eq. 4 to generate samples from the conditional distribution of x given $\mathbf{Z}$ and learn the MoP representation from the generated dataset; *ii*) to use numerical techniques to approximate the fraction directly (specifically, both Taylor series and Lagrange interpolation were considered). In our work we first learn an MoP representation for $f(x, \mathbf{z})$ using the program in Eq. 3, then *calculate* $f(\mathbf{z}) = \int_{\Omega_x} f(x, \mathbf{z})\mathrm{d}x$ directly from the learned joint. Note that since $f(x, \mathbf{z})$ is a MoP the integral can easily be performed analytically. Next, the conditional distribution defined through Eq. 4 is our target, leading to the following optimization program:

$$\begin{aligned} \textbf{minimize} \quad & \sum_{\ell=1}^{N} \left(\frac{f(x_\ell, \mathbf{z}_\ell)}{f(\mathbf{z}_\ell)} - f(x_\ell|\mathbf{z}_\ell) \right)^2 \\ \textbf{subject to} \quad & f(x|\mathbf{z}) \geq 0 \quad \forall (x, \mathbf{z}) \in (\Omega_X \times \Omega_{\mathbf{z}})\,. \end{aligned} \tag{5}$$

The solution to this problem is a parameter-set that defines an un-normalized conditional PDF (that is, we have no guarantee that $\int_{\Omega_x} f(x|\mathbf{z}) \mathrm{d}x = 1$ for all $\mathbf{z} \in \Omega_{\mathbf{z}}$). Hence, the procedure is finalized by partially normalizing the distribution [10]. The program is quadratic, and can therefore be solved efficiently.

We note that while the programmes in Eqs. 2 and 3 are defined to obtain the CDFs, the programme in Eq. 5 works directly with the PDF. The reason for the programmes in Eqs. 2 and 3 to work with the cumulative distribution functions is that the defined $G_N(\cdot)$ function is a more robust data-representation than, say, a histogram [5], and as $G_N(\cdot)$ represents the empirical CDF the result of these programs are also CDFs. On the other hand, the program in Eq. 5 does not work directly with representations of the data, but rather defines the target function through Eq. 4. Therefore, the objects under study by this program are PDFs.

4 Experimental Analysis

In this section, we compare the proposal given in Sect. 3 with the methods described in [5] (where the conditioning variables are discretized) and in [11] (where B-splines are used) for learning conditional MoPs from data.

We consider two different scenarios concerning two continuous variables, X and Y. In the first one, $Y \sim \mathcal{N}(\mu = 0, \sigma = 1)$ and $X|\{Y = y\} \sim \mathcal{N}(\mu = y, \sigma = 1)$. In the second scenario, $Y \sim$ Gamma(rate = 10, shape = 10) and $X|\{Y = y\} \sim$ Exp(rate = y). For each scenario, we generated 10 data-sets of samples $\{X_i, Y_i\}_{i=1}^{N}$, where the size is chosen as $N = 25$, 500, 2500, 5000. The effectiveness of the tested methods was measured by computing the mean square error for each set of samples. The results are showed in Tables 1 and 2.

The results in Table 1 indicate that the most accurate results for scenario 1 are achieved by the B-spline approach [11]. The worst results by far are obtained by the approach that discretizes the conditioning variables [5]. Both the proposed approach and the B-spline approach yield errors close to zero in most cases.

The results for scenario 2 are reported in Table 2. In this case, the most accurate results in terms of mean square error are provided by the MoTBF approach. Again, the method in [5] obtains the worst results overall.

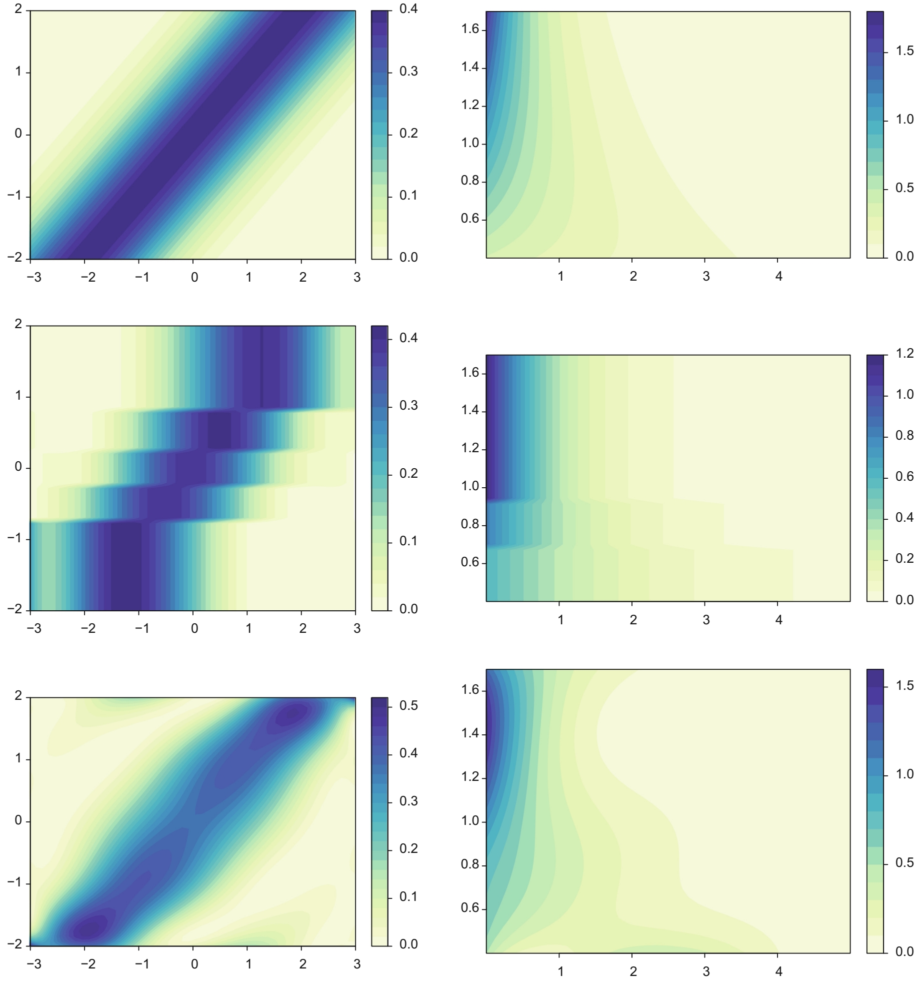
The results are consistent with the plots in Fig. 2, where the MoTBF approach (bottom row in the figure) presented in this paper is able to resemble the shape of the exact conditional distribution (top row), specially in the non Gaussian scenario, while the method in [5] (middle row) is penalized by the fact that the estimated model is piecewise constant along the Y axis. The plots in Fig. 2 show the results obtained when learning from $N = 5000$ samples.

Table 1. Average MSE between the different methods to obtain MoP approximations and the true conditional densities for each set of 10 samples, where $Y \sim \mathcal{N}(0,1)$ and $X|Y \sim \mathcal{N}(y,1)$.

N	$f_{X\|Y}(x\|y)$	*Split Method*[5]	*MoTBF Algorithm*	*B-Splines Method*[11]
25	y=-0.6748	0.1276	0.0848	**0.0103**
	y=0.00	0.1254	0.0936	**0.0089**
	y=0.6748	0.1279	0.1416	**0.0105**
500	y=-0.6748	0.0256	0.0453	**0.0025**
	y=0.00	0.0317	0.0117	**0.0009**
	y=0.6748	0.0246	0.0411	**0.0020**
2500	y=-0.6748	0.0031	0.0019	**0.0006**
	y=0.00	0.0064	0.0010	**0.0002**
	y=0.6748	0.0058	0.0024	**0.0006**
5000	y=-0.6748	0.0019	0.0018	**0.0006**
	y=0.00	0.0074	0.0009	**0.0002**
	y=0.6748	0.0019	0.0020	**0.0006**

Table 2. Average MSE between the different methods to obtain MoP approximations and the true conditional densities for each set of 10 samples, where $Y \sim Gamma(rate = 10, shape = 10)$ and $X|Y \sim Exp(y)$.

N	$f_{X\|Y}(x\|y)$	*Split Method*[5]	*MoTBF Algorithm*	*B-Splines Method*[11]
25	y=0.7706	0.4054	**0.0083**	0.0131
	y=0.9684	0.4703	**0.0081**	0.0225
	y=1.1916	0.5473	**0.0229**	0.0374
500	y=0.7706	0.0158	0.0037	**0.0012**
	y=0.9684	0.0048	0.0034	**0.0022**
	y=1.1916	0.0118	**0.0039**	0.0057
2500	y=0.7706	0.0064	**0.0025**	**0.0025**
	y=0.9684	0.0080	**0.0024**	0.0043
	y=1.1916	0.0029	**0.0046**	0.0074
5000	y=0.7706	**0.0013**	0.0021	0.0015
	y=0.9684	0.0091	**0.0015**	0.0022
	y=1.1916	**0.0026**	0.0029	0.0032



(a) $Y \sim \mathcal{N}(0, 1)$ $X|Y \sim \mathcal{N}(y, 1)$.

(b) $Y \sim Gamma(10, 10)$ $X|Y \sim Exp(y)$.

Fig. 2. For the two scenarios (in columns), true conditional density (top row), the MoP produced by the method introduced in [5] (middle row) and the MoP obtained by the proposal in this paper (bottom row).

5 Concluding Remarks

In this paper we have extended the learning algorithm for univariate MoTBFs in [5] to multivariate and conditional densities. The advantage of the proposal described here with respect to the B-spline approach in [11] is that there is no need to split the domain of any variable. This is a fundamental issue in order to keep the complexity of inference in hybrid Bayesian networks under control. We note that while in theory high order polynomials may be required to model the distributions, the use of the BIC-score [8] leads to low-order polynomials being selected in practice [4,5].

The experimental analysis suggests that our proposal is competitive with the B-spine approach in a range of commonly used distributions. Even if the conditional distribution functions yielded by the method is this paper are not proper conditional densities, evidence so far indicates they are accurate approximations, which in practice allows the method to be used as a means of representing the parameters of a Bayesian network. This paves the way to envisioning structural learning algorithms for hybrid Bayesian networks parameterized by MoTBFs.

Finally, we note that even though the paper develops a learning method for MoPs, the techniques employed here can easily be extended to be applicable for MoTBFs in general.

Acknowledgments. This research has been partly funded by the Spanish Ministry of Economy and Competitiveness, through project TIN2013-46638-C3-1-P and by Junta de Andalucía through project P11-TIC-7821 and by ERDF funds. A part of this work was performed within the AMIDST project. AMIDST has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 619209.

References

1. Langseth, H., Nielsen, T.D., Rumí, R., Salmerón, A.: Maximum likelihood learning of conditional MTE distributions. In: Sossai, C., Chemello, G. (eds.) ECSQARU 2009. LNCS, vol. 5590, pp. 240–251. Springer, Heidelberg (2009)
2. Langseth, H., Nielsen, T.D., Rumí, R., Salmerón, A.: Mixtures of truncated basis functions. Int. J. Approximate Reasoning **53**, 212–227 (2012)
3. Langseth, H., Nielsen, T.D., Salmerón, A.: Learning mixtures of truncated basis functions from data. In: Proceedings of the Sixth European Workshop on Probabilistic Graphical Models (PGM 2012), pp. 163–170 (2012)
4. Langseth, H., Nielsen, T.D., Rumí, R., Salmerón, A.: Inference in hybrid Bayesian networks with mixtures of truncated basis functions. In: Proceedings of the Sixth European Workshop on Probabilistic Graphical Models (PGM 2012), pp. 171–178 (2012)
5. Langseth, H., Nielsen, T.D., Pérez-Bernabé, I., Salmerón, A.: Learning mixtures of truncated basis functions from data. Int. J. Approximate Reasoning **55**, 940–956 (2014)
6. Lauritzen, S.: Propagation of probabilities, means and variances in mixed graphical association models. J. Am. Stat. Assoc. **87**, 1098–1108 (1992)
7. Moral, S., Rumí, R., Salmerón, A.: Mixtures of truncated exponentials in hybrid Bayesian networks. In: Benferhat, S., Besnard, P. (eds.) ECSQARU 2001. LNCS (LNAI), vol. 2143, p. 156. Springer, Heidelberg (2001)
8. Schwarz, G.: Estimating the dimension of a model. Ann. Stat. **6**, 461–464 (1978)
9. Shenoy, P., Shafer, G.: Axioms for probability and belief function propagation. In: Shachter, R., Levitt, T., Lemmer, J., Kanal, L. (eds.) Uncertainty in Artificial Intelligence 4, pp. 169–198. North Holland, Amsterdam (1990)
10. Shenoy, P., West, J.: Inference in hybrid Bayesian networks using mixtures of polynomials. Int. J. Approximate Reasoning **52**, 641–657 (2011)

11. Varando, G., López-Cruz, P.L., Nielsen, T.D., Bielza, C., Larrañga, P.: Conditional density approximations with mixtures of polynomials. Int. J. Intell. Syst. **30**, 236–264 (2015)
12. Zhang, N., Poole, D.: Exploiting causal independence in Bayesian network inference. J. Artif. Intell. Res. **5**, 301–328 (1996)