

Mixtures of Truncated Basis Functions

Brief introduction to MoTBFs

Mixtures of truncated basis functions (MoTBFs) [1] have been proposed as a general framework for handling hybrid Bayesian networks, i.e., Bayesian networks where discrete and continuous variables coexist. As special cases, the framework includes the so-called mixtures of truncated exponentials (MTEs) [2] and mixtures of polynomials (MoPs) [3, 4].

One of the advantages of MoTBFs is that they allow for hybrid Bayesian networks with no structural restrictions on the relations between the continuous and discrete variables; this is in contrast to conditional Gaussian (CG) models [5], where discrete variables are not allowed to have continuous parents. Restricting arc directions is problematic, for instance in situations where the network structure is given a causal interpretation; the fact that MoTBFs have no such restriction make them suitable for this kind of analysis. Furthermore, MoTBFs are closed under addition, multiplication, and integration, which facilitates the use of exact probabilistic inference methods like the Shenoy-Shafer architecture [6] or the variable elimination algorithm [7].

Methods for learning MoTBFs from data have previously been studied and covers algorithms for learning both marginal MoTBF densities [1, 8] as well as conditional MoTBF densities from data [9–11]. To address situations where data is limited, [12] proposed a methodology for integrating prior knowledge when learning univariate and conditional MoTBFs from data. The underlying idea of the integration is to represent the prior knowledge as an MoTBF density that is later combined with the density learned from data, thus resulting in a new MoTBF density. Together, the learning algorithms for MoTBF-based Bayesian networks facilitate learning in data-rich domains as well as domains where limited quantitative data is counterbalanced by qualitative domain knowledge.

The MoTBF framework [1] is based on the abstract notion of real-valued *basis function*, which includes both polynomial and exponential functions as special cases.

More formally, let $\mathbf{X}$ be a mixed n -dimensional random vector. Let $\mathbf{Y} = (Y_1, \dots, Y_d)$ and $\mathbf{Z} = (Z_1, \dots, Z_c)$ be the discrete and continuous parts of $\mathbf{X}$, respectively, with $c + d = n$. Let $\Psi = \{\psi_i(\cdot)\}_{i=0}^{\infty}$ with $\psi_i : \mathbb{R} \rightarrow \mathbb{R}$ define a collection of real basis functions. We say that a function $\hat{f} : \Omega_{\mathbf{X}} \mapsto \mathbb{R}_0^+$ is a *mixture of truncated basis functions* potential to level k wrt. Ψ if one of the following two conditions holds:

- f can be written as

$$f(\mathbf{x}) = f(\mathbf{y}, \mathbf{z}) = \sum_{i=0}^k \prod_{j=1}^c a_{i,\mathbf{y}}^{(j)} \psi_i(z_j), \quad (1)$$

where $a_{i,\mathbf{y}}^{(j)}$ are real numbers.

- There is a partition $\Omega_{\mathbf{X}}^1, \dots, \Omega_{\mathbf{X}}^m$ of $\Omega_{\mathbf{X}}$ for which the domain of the continuous variables, $\Omega_{\mathbf{Z}}$, is divided into hyper-cubes and such that f is defined as

$$f(\mathbf{x}) = f_{\ell}(\mathbf{x}) \quad \text{if } \mathbf{x} \in \Omega_{\mathbf{X}}^{\ell},$$

where each f_{ℓ} , $\ell = 1, \dots, m$ can be written in the form of Equation~1.

Typically, a univariate MoTBF for a variable X does not rely on a partitioning of Ω_X .

To see the relationship between MoTBFs and MoPs [3], we can instantiate the basis functions as polynomials (i.e., $\psi_i(x) = x^i$, for $i = 0, \dots, k$) in which case the MoTBF model reduces to an MoP model. For example, with polynomial basis functions, a univariate MoTBF of level k for a variable X is given by

$$f(x) = \sum_{i=0}^k \theta_i \psi_i(x).$$

Similarly, by having exponential basis functions the MoTBF model implements an MTE model [2]. For ease of exposition, we shall in the remainder of this paper assume polynomial basis functions unless explicitly stated otherwise.

An MoTBF potential is a *density* if $\sum_{\mathbf{y} \in \Omega_{\mathbf{Y}}} \int_{\Omega_{\mathbf{Z}}} f(\mathbf{y}, \mathbf{z}) d\mathbf{z} = 1$. Similarly, we say that an MoTBF $f(\mathbf{y}, \mathbf{z})$ is a *conditional density* for $\mathbf{Z}' \subseteq \mathbf{Z}$ and $\mathbf{Y}' \subseteq \mathbf{Y}$ given $(\mathbf{Z} \setminus \mathbf{Z}')$ and $(\mathbf{Y} \setminus \mathbf{Y}')$ if

$$\sum_{\mathbf{y}' \in \Omega_{\mathbf{Y}'}} \int_{\Omega_{\mathbf{Z}'}} f(\mathbf{y}', \mathbf{y}'', \mathbf{z}', \mathbf{z}'') d\mathbf{z}' = 1,$$

for all $\mathbf{z}'' \in \Omega_{\mathbf{Z} \setminus \mathbf{Z}'}$ and $\mathbf{y}'' \in \Omega_{\mathbf{Y} \setminus \mathbf{Y}'}$. Following [1] we furthermore assume that the influence that a set of continuous parent variables $\mathbf{Z}$ have on their child variable X is encoded only through the partitioning of $\Omega_{\mathbf{Z}}$ into hyper-cubes, and not directly in the functional form of $f(x|\mathbf{z})$ inside the hyper-cube $\Omega_{\mathbf{Z}}^j$. That is, for a partitioning $\mathcal{P} = \{\Omega_{\mathbf{Z}}^1, \dots, \Omega_{\mathbf{Z}}^m\}$ of $\Omega_{\mathbf{Z}}$, the conditional MoTBF is defined for $\mathbf{z} \in \Omega_{\mathbf{Z}}^j$, $1 \leq j \leq m$, as

$$f_k^{(j)}(x|\mathbf{z} \in \Omega_{\mathbf{Z}}^j) = \sum_{i=0}^k \theta_i^{(j)} \psi_i^{(j)}(x). \quad (2)$$

In the remainder of this paper we shall assume that a conditional MoTBF density includes only a single ‘head’ variable, i.e., $|\mathbf{Z}' \cup \mathbf{Y}'| = 1$.

Learning univariate MoTBFs from data

[10] present a method for learning univariate MoTBF distributions from data. This method is also implemented in the **MoTBFs** package and is briefly described here together with its extension to both conditional and joint distributions. The estimation procedure relies on the empirical cumulative distribution function (CDF) as a representation of the data, which, for a sample $D = \{x_1, \dots, x_N\}$, is defined as

$$G_N(x) = \frac{1}{N} \sum_{\ell=1}^N \mathbf{1}\{x_{\ell} \leq x\}, \quad x \in \mathbb{R}, \quad (3)$$

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

The algorithm developed by [10] fits a potential, whose derivative is an MoTBF, to the empirical CDF using least squares. As an example, if we use polynomials as basis functions, $\Psi = \{1, x, x^2, x^3, \dots\}$, the parameters of the CDF, denoted as $c_0, \dots, c_k$, are estimated by solving the optimization problem

$$\begin{aligned} & \underset{c_0, \dots, c_k}{\text{minimize}} && \sum_{\ell=1}^N \left(G_N(x_{\ell}) - \sum_{i=0}^k c_i x_{\ell}^i \right)^2 \\ & \text{subject to} && \sum_{i=1}^k i c_i x^{i-1} \geq 0 \quad \forall x \in \Omega, \\ & && \sum_{i=0}^k c_i \alpha^i = 0 \text{ and } \sum_{i=0}^k c_i \beta^i = 1, \end{aligned} \quad (4)$$

where the constraints ensure that the obtained parameter estimates define a valid CDF, with α and β being the minimum and maximum values of the data sample, respectively. More precisely, the first constraint

guarantees that the corresponding density is non-negative and the last two restrictions ensure that it integrates to 1. The latter is equivalent to stating that the CDF should be equal to 0 at the minimum and equal to 1 at the maximum. Note that the estimated function is not actually a density, but a CDF instead. An MoTBF density can be obtained by simply taking the derivative of the CDF.

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

In learning scenarios involving a large amount of data (i.e., when N is large), solving the program can be time consuming. In such cases we define a *grid* on Ω_X , that is selected so that the number of observations is the same between each pair of consecutive grid-points. The grid-points are used to evaluate the objective function instead of the sample points.

The level k of the estimated MoP can be decided using different model selection techniques. For the results presented in this paper we have performed a greedy search, choosing the value for k maximizing the Bayesian information criterion (BIC) [13]:

$$\text{BIC}(f, D) = \sum_{\ell=1}^N \log f(x_\ell) - \frac{k+2}{2} \log N. \quad (5)$$

This choice is motivated by [10], who showed that the estimators based on Equation 4 are consistent in terms of the mean squared error for all $x \in \Omega_X$.

Learning conditional MoTBFs from data

In a conditional MoTBF density, the continuous parent variables $\mathbf{Z}$ only influence the child variable X through the partitioning of the domain of $\mathbf{Z}$ into hyper-cubes, and not directly in the functional form of $f(x|\mathbf{z})$ inside each hyper-cube [1]. Thus, learning a conditional MoTBF basically consists in finding a partitioning of the domain of the parent variables and using the procedure above for estimating the density of the child variable for each of these partitions.

This procedure is formally described by [10], where the domain of the parent variables, $\Omega_{\mathbf{Z}}$, is incrementally split as long as the BIC score improves. Equal frequency binning is used to determine candidate split points. After splitting a variable Z , the algorithm fits a univariate MoTBF density for each induced sub-partition $\Omega_{\mathbf{Z}}^1$ and $\Omega_{\mathbf{Z}}^2$. A candidate partition $\Omega_{\mathbf{Z}'}$ is only accepted if the BIC score is improved, i.e., if

$$\text{BIC-Gain}(\Omega'_{\mathbf{Z}}, Z) = \text{BIC}(f', D) - \text{BIC}(f, D) > 0,$$

where f' is the conditional MoTBF potential defined over the candidate partition.

Learning joint MoTBFs from data

The procedure for learning joint densities is an extension of the program in Equation 4 to random vectors of arbitrary dimension [11]. In the multivariate case, the sample is a set of d -dimensional observations, $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, $\mathbf{x} \in \Omega_{\mathbf{X}} \subset \mathbb{R}^d$. We 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 $i = 1, \dots, d$. 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

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

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

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

obtained by solving the optimization problem

$$\begin{aligned}
& \textbf{minimize} && \sum_{\ell=1}^N (G_N(\mathbf{x}_\ell) - F(\mathbf{x}_\ell))^2 \\
& \textbf{subject to} && \frac{\partial^d F(\mathbf{x})}{\partial x_1, \dots, \partial x_d} \geq 0 \quad \forall \mathbf{x} \in \Omega_{\mathbf{X}}, \\
& && F(\Omega_{\mathbf{X}}^-) = 0 \text{ and } F(\Omega_{\mathbf{X}}^+) = 1.
\end{aligned} \tag{6}$$

The solution to this problem is the parameter-set that defines the joint CDF, and the density can be obtained by differentiation of the joint CDF. As in the univariate case, it is a quadratic optimization problem, that can be solved efficiently if the objective function is only evaluated on a set of grid-points.

Incorporating prior knowledge

There are real world situations, where the amount of available data is insufficient for accurate density estimation. The **MoTBFs** package includes implementations oriented to face these kinds of situations by allowing prior knowledge to be taken into account during density estimation. In Bayesian statistics [14], prior information is encoded as prior probability distributions over the parameters. As an example, consider the case of a random variable representing the body temperature of a patient in a hospital, and assume that the variable is normally distributed with mean μ and standard deviation σ . Prior knowledge could be provided in the form of a prior distribution on μ by establishing that $\mu \sim \mathcal{N}(37, 0.1)$. However, in the case of MoTBF distributions, the parameters do not have a meaning in general. Therefore, there is no clear way in which a practitioner could provide prior information on any of the parameters, although some information could still be specified. For instance, in the body temperature example, the practitioner could choose not to give prior information on any single parameter, but instead provide a full distribution of the variable, reflecting his or her prior knowledge when no data is available. Such prior information could, e.g., include that the body temperature follows

a normal distribution with mean 37 and standard deviation 0.5. This is the approach followed by [12], where prior knowledge is encoded as an MoTBF distribution over the random variable, which is then later combined with the MoTBF density learned from data.

In order to represent such a prior distribution as an MoTBF, a sample is drawn from the prior – e.g., $\mathcal{N}(37, 0.1)$ in the example above – and the sample is then used to learn an MoTBF using the methods previously described in this paper. Alternatively, one may also rely on direct translation schemes as described in, e.g., [15, 16].

Given a prior MoTBF density f_{prior} over a variable X , [12] obtain a posterior density f_{post} by making a linear combination of the prior density and the density f_{data} learned from data, expressed as

$$f_{post} = w_p f_{prior} + w_d f_{data} \quad ,$$

where w_p is the weight of f_{prior} and w_d is the weight of f_{data} , with $0 \leq w_p \leq 1$, $0 \leq w_d \leq 1$ and $w_p + w_d = 1$. The weights w_p and w_d reflect the way in which they explain the observed data. They are computed by measuring the difference between the log-likelihood produced by each one of the densities (f_{prior} and f_{data}) and the expected log-likelihood that would be produced by a randomly generated MoTBF. See [12] for a detailed explanation of the procedure.

Probabilistic inference

In a Bayesian network, probabilistic inference is the task of computing the posterior distribution of any variable X given that some other variables $\mathbf{Y}$ have been observed to take some value $\mathbf{y}$. Therefore, the goal of probabilistic inference is to compute the density $f(x|\mathbf{y})$, where the value $\mathbf{y}$ is fixed.

An easy approach to estimate this density is by *forward sampling* [17]. The idea is to draw a sample of configurations of the variables in the Bayesian network by simulating each variable using its conditional

distribution following a top-down order. Prior to sampling, all the densities in the network are restricted to the value $\mathbf{Y} = \mathbf{y}$ and when a variable is to be sampled, its conditional distribution is restricted to the values already obtained for its parents. Sampling is always possible since the variables are sampled following the topological ordering of the network. Once the sample has been obtained, $f(x|\mathbf{y})$ is estimated as a univariate MoTBF as we described before.

References

1. Langseth, H., Nielsen, T. D., Rumí, R., & Salmerón, A. (2012). Mixtures of truncated basis functions. *International Journal of Approximate Reasoning*, 53, 212–227.
2. Moral, S., Rumí, R., & Salmerón, A. (2001). Mixtures of truncated exponentials in hybrid Bayesian networks. In *ECSQARU'01. Lecture notes in artificial intelligence* (Vol. 2143, pp. 135–143).
3. Shenoy, P. P., & West, J. C. (2011). Inference in hybrid Bayesian networks using mixtures of polynomials. *International Journal of Approximate Reasoning*, 52, 641–657.
4. López-Cruz, P. L., Bielza, C., & Larrañaga, P. (2012). Learning mixtures of polynomials from data using B-spline interpolation. In A. Cano, M. Gómez-Olmedo, & T. D. Nielsen (Eds.), *Proceedings of the 6th european workshop on probabilistic graphical models (PGM'12)* (pp. 211–218).
5. Lauritzen, S. L. (1992). Propagation of probabilities, means and variances in mixed graphical association models. *Journal of the American Statistical Association*, 87, 1098–1108.
6. Shenoy, P. P., & Shafer, G. (1990). Axioms for probability and belief function propagation. In R. D. Shachter, T. S. Levitt, J. F. Lemmer, & L. N. Kanal (Eds.), *Uncertainty in artificial intelligence 4* (pp. 169–198). North Holland, Amsterdam.
7. Zhang, N. L., & Poole, D. (1996). Exploiting causal independence in Bayesian network inference. *Journal of Artificial Intelligence Research*, 5, 301–328.
8. Langseth, H., Nielsen, T. D., & Salmerón, A. (2012). Learning mixtures of truncated basis functions from data. In *Proceedings of the sixth european workshop on probabilistic graphical models (PGM'2012)* (pp. 163–170).
9. Langseth, H., Nielsen, T. D., Rumí, R., & Salmerón, A. (2009). Maximum likelihood learning of conditional MTE distributions. *ECSQARU 2009. Lecture Notes in Computer Science*, 5590, 240–251.
10. Langseth, H., Nielsen, T. D., Pérez-Bernabé, I., & Salmerón, A. (2014). Learning mixtures of truncated basis functions from data. *International Journal of Approximate Reasoning*, 55, 940–956.
11. Pérez-Bernabé, I., Salmerón, A., & Langseth, H. (2015). Learning conditional distributions using mixtures of truncated basis functions. *ECSQARU'2015. Lecture Notes in Artificial Intelligence*, 9161, 397–406.
12. Pérez-Bernabé, I., Fernández, A., Rumí, R., & Salmerón, A. (2016). Parameter learning in hybrid Bayesian networks using prior knowledge. *Data Mining and Knowledge Discovery*, 30, 576–604.
13. Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6, 461–464.
14. Bernardo, J. M., & Smith, A. F. (2009). *Bayesian theory* (Vol. 405). Wiley. com.
15. Langseth, H., Nielsen, T. D., Rumí, R., & Salmerón, A. (2010). Parameter estimation and model selection for mixtures of truncated exponentials. *International Journal of Approximate Reasoning*, 51, 485–498.
16. Cobb, B., Shenoy, P. P., & Rumí, R. (2006). Approximating probability density functions with mixtures of truncated exponentials. *Statistics and Computing*, 16, 293–308.
17. Henrion, M. (1988). Propagating uncertainty by logic sampling in Bayes' networks. In J. F. Lemmer & L. N. Kanal (Eds.), *Uncertainty in artificial intelligence* (Vol. 2, pp. 317–324). North-Holland (Amsterdam).