Learning Mixtures of Truncated Basis Functions from Data

Langseth, Helge; Nielsen, Thomas Dyhre; Salmerón, Antonio

Published in:
Proceedings of the Sixth European Workshop on Probabilistic Graphical Models

Publication date:
2012

Document Version
Early version, also known as pre-print

Learning Mixtures of Truncated Basis Functions from Data

Helge Langseth
Department of Computer and Information Science
The Norwegian University of Science and Technology
Trondheim (Norway)
helgel@idi.ntnu.no

Thomas Dyhre Nielsen
Department of Computer Science
Aalborg University
Aalborg (Denmark)
tdn@cs.aau.dk

Antonio Salmerón
Department of Statistics and Applied Mathematics
University of Almería, Almería (Spain)
antonio.salmeron@ual.es

Abstract

In this paper we describe a new method for learning hybrid Bayesian network models from data. The method utilizes a kernel density estimator, which is in turn "translated" into a *mixture of truncated basis functions*-representation using a convex optimization technique. We argue that these estimators approximate the maximum likelihood estimators, and compare our approach to previous attempts at learning hybrid Bayesian networks from data. We conclude that while the present method produces estimators that are slightly poorer than the state of the art (in terms of log likelihood), it is significantly faster.

1 Introduction

In domains involving both discrete and continuous variables, Bayesian networks with mixtures of truncated exponentials (MTEs) (Moral et al., 2001) and mixtures of truncated polynomials (MOPs) (Shenoy and West, 2011) have received increasing interest over the last few years. A recent addition to the fold is the *mixtures of truncated basis functions* (MoTBFs) framework (Langseth et al., 2012), which offers a unified theory for MTEs and MoPs. The MoTBFs framework allows discrete and continuous variables to co-exist in a Bayesian network without any structural constraints, and since the family of MoTBFs is closed under addition, multiplication, and integration, inference in an MoTBF network can be performed efficiently using the Shafer-Shenoy architecture (Shafer and Shenoy, 1990).

The problem of learning MoTBF models from data has been only scarcely considered, with the main body of work relating to MTEs (Romero et al., 2006; Langseth et al., 2009, 2010); we are not aware of published contributions focusing on the MoP framework. Romero et al. (2006) used a kernel estimator to represent the data distribution, and thereafter fitted an MTE to the kernel using regression. Langseth et al. (2009, 2010) attempted to find maximum likelihood parameters directly but since the maximum likelihood equations have no analytic solution in general, they instead proposed an iterative scheme utilizing Newton's method and Lagrange multipliers. This resulted in better estimators (in terms of likelihood on the training-set as well as on a hold-out test-set), but at the cost of a steep increase in the computational complexity.

We present a new parameter estimation method, which aims at *approximating* the maximum likelihood parameters of an MoTBF network with known structure. We compare our results to those of Langseth et al. (2009, 2010), and find that although the new method finds parameters that are slightly poorer (in terms of likelihood), it is more than an order of magnitude faster than previous techniques.

The rest of the paper is organized as follows:

We start with an introduction to the MoTBF framework in Section 2. The simplified problem of learning univariate MoTBFs from data is considered in Section 3, and we discuss learning conditional distributions in Section 4. We report on some experiments in Section 5, and finally we conclude in Section 6.

2 The MoTBF model

The MoTBF framework is based on the abstract notion of real-valued *basis functions* $\psi(\cdot)$, which includes both polynomial and exponential functions as special cases. The first building-block of the framework is the marginal distribution: Let X be a continuous variable[1] with domain $\Omega_X \subseteq \mathbb{R}$ and let $\psi_i : \mathbb{R} \to \mathbb{R}$, for $i = 0, \ldots, k$, define a collection of real basis functions. We say that a function $g_k : \Omega_X \mapsto \mathbb{R}_0^+$ is a *mixture of truncated basis functions* (MoTBF) potential of level k wrt. $\Psi = \{\psi_0, \psi_1, \ldots, \psi_k\}$ if g_k can be written as

$$g_k(x) = \sum_{i=0}^{k} a_i \, \psi_i(x), \tag{1}$$

where a_i are real numbers. The potential is a density if $\int_{\Omega_X} g_k(x)\, dx = 1$. Note that as opposed to the MTE and MoP definitions, a marginal MoTBF potential does not employ interval refinement to improve its expressive power.

Next, we turn to the MoTBF definition of conditional distributions, which mirrors the corresponding definition for MTEs. Thus, the influence a set of continuous parent variables $\mathbf{Z}$ has on their child variable X is encoded only through the partitioning of the domain of $\mathbf{Z}$, $\Omega_{\mathbf{Z}}$, into hyper-cubes, and not directly in the functional form of $g_k^{(\ell)}(x|\mathbf{z})$ inside each hypercube $\Omega_{\mathbf{Z}}^{\ell}$. More precisely, for a partitioning $\mathcal{P} = \{\Omega_{\mathbf{Z}}^1, \ldots, \Omega_{\mathbf{Z}}^m\}$ of $\Omega_{\mathbf{Z}}$, the conditional MoTBF is defined for $\mathbf{z} \in \Omega_{\mathbf{Z}}^j$, $1 \leq j \leq m$, as

$$g_k^{(j)}(x|\mathbf{z} \in \Omega_{\mathbf{Z}}^j) = \sum_{i=0}^{k} a_{i,j} \, \psi_i(x). \tag{2}$$

Finally, the joint MoTBF distribution over $\mathbf{x} = (x_1, \ldots, x_n)$ is found using the usual factorization, $g_{\boldsymbol{k}}(\mathbf{x}) = \prod_{i=1}^{n} g_{k_i}(x_i|\mathrm{pa}\,(x_i))$, where the marginals and conditional distributions are defined using Equations (1) and (2), respectively.

Langseth et al. (2012) describe a "translation" procedure for efficiently finding an MoTBF approximation of any density function. The approximation procedure assumes that the basis functions Ψ are both *legal* and *orthonormal*: If $\mathcal{Q}$ is the set of all linear combinations of the members of a set of basis functions $\Psi = \{\psi_i(\cdot)\}_{i=0}^{\infty}$, then Ψ is said to be a *legal* set of basis functions if the following conditions hold:

- ψ_0 is constant in its argument.
- If $\phi_i \in \mathcal{Q}$ and $\phi_j \in \mathcal{Q}$, then $(\phi_i \cdot \phi_j) \in \mathcal{Q}$.
- For any pair of real numbers s and t, there exists a function $\phi \in \mathcal{Q}$ such that $\phi(s) \neq \phi(t)$.

When considering orthonormal basis functions, we focus on the space $L^2[a, b]$ of quadratically integrable real functions over the finite interval $\Omega = [a, b]$. For two functions ϕ_i and ϕ_j defined on Ω we define the inner product as

$$\langle \phi_i, \phi_j \rangle = \int_{\Omega} \phi_i(x)\phi_j(x)dx,$$

and say that two functions are orthonormal if $\langle \phi_i, \phi_j \rangle = \delta_{ij}$, where δ_{ij} is the Kronecker delta. A set of non-orthonormal basis functions can easily be orthonormalized using, for instance, the Gram-Schmidt procedure.

To set the scene, we let $f(x)$ be the (target) density, and let $g_k(x|\boldsymbol{\theta})$ be an MoTBF of order k. The key idea of Langseth et al. (2012) is to use *generalized Fourier series* to find "optimal" values for $\boldsymbol{\theta} = (\theta_0, \ldots, \theta_k)$, that is, choosing $\hat{\theta}_i = \langle f, \psi_i \rangle$. It can easily be shown that while the generalized Fourier approximation up to degree k is guaranteed to minimize the L^2 distance

[1] In this paper we will often refer to MoTBF potentials defined only over continuous variables. In such cases, we understand, unless the contrary is specified, that all the claims about such potentials are extensible to those potentials also containing discrete variables in their domains, simply by having the claims hold for each configuration of the discrete variables.

$\int_x \left(f(x) - g_k(x|\hat{\boldsymbol{\theta}})\right)^2 dx$, g_k is not always positive, and is thus not a density approximation. A convex optimization scheme (initialized with the generalized Fourier series coefficients) was therefore employed to obtain parameters that guarantee that $g_k(x)$ is a density, and at the same time minimize an upper bound of the KL divergence $D\left(\, f \,\|\, g_k \,\right)$ (Langseth et al., 2012). It was also shown that the approximation can be made arbitrarily tight, simply by increasing k.

3 Learning univariate distributions

While Langseth et al. (2012) defined their translation procedure as a means to create MoTBF approximations of *known* distributions, this paper will utilize the translation for *learning* hybrid BNs from data. The top-level algorithm is to (1) approximate the data using a kernel-density, and (2) approximate the kernel density with an MoTBF parameterization. We discuss each step below, and start by looking at univariate (marginal) distributions. We will move on to conditional distributions in Section 4.

Assume that $f(x)$ is the (unknown) density, which generated the univariate sample $\mathcal{D} = \{x_1, \ldots, x_N\}$. Next, let $h_{\mathcal{D}}\left(\cdot|t_w\right)$ be a kernel density estimator based on the samples $\mathcal{D}$ using kernel function t_w with bandwidth w. We define the kernel density estimator s.t. w approaches zero as $N \to \infty$. Now, the soundness of the approach rests upon the following proposition:

Proposition 1. *Let $\tilde{\boldsymbol{\theta}}_N$ be chosen to minimize $D\left(\, h_{\mathcal{D}}\left(x|t_w\right) \,\|\, g_k(x|\tilde{\boldsymbol{\theta}}_N) \,\right)$. Then $\tilde{\boldsymbol{\theta}}_N$ converges to the maximum likelihood estimator of $\boldsymbol{\theta}$ as $N \to \infty$.*

Sketch of proof:
First we note that since

$$D\left(\, h_{\mathcal{D}}\left(x|t_w\right) \,\|\, g_k(x|\boldsymbol{\theta}) \,\right) = \int_x h_{\mathcal{D}}(x|t_w) \log\left(\frac{h_{\mathcal{D}}(x|t_w)}{g_k(x|\boldsymbol{\theta})}\right) dx,$$

minimizing $D\left(\, h_{\mathcal{D}}\left(x|t_w\right) \,\|\, g_k(x|\boldsymbol{\theta}) \,\right)$ wrt. $\boldsymbol{\theta}$ is equivalent to maximizing $\mathbb{E}_{h_{\mathcal{D}}}[\log g_k(X|\boldsymbol{\theta})]$ wrt. $\boldsymbol{\theta}$; the expectation is taken wrt. X, which is assumed to have density function $h_{\mathcal{D}}\left(\cdot|t_w\right)$. Next, since the bandwidth of $h_{\mathcal{D}}\left(\cdot|t_w\right)$ decreases to 0 as $N \to \infty$, we have that

$$h_{\mathcal{D}}\left(x|t_w\right) \to \frac{1}{N}\sum_{\ell=1}^{N} \delta(x - x_\ell)$$

as $N \to \infty$, where $\delta(\cdot)$ is Dirac's delta. Therefore, $\mathbb{E}_{h_{\mathcal{D}}}[\log g_k(X|\boldsymbol{\theta})] \to \int_x \sum_\ell \frac{1}{N}\delta(x - x_\ell) \cdot \log g_k(x|\boldsymbol{\theta}) dx = \frac{1}{N}\sum_\ell \log g_k(x_\ell|\boldsymbol{\theta})$. It follows that the parameters that minimize the KL divergence are asymptotically also those that maximize the likelihood.

The MoTBF density $g_k(x|\boldsymbol{\theta})$ uses $k+2$ parameters: The interval of support (2 values) and the k "free" θ-values (θ_0 is fixed to make sure the function integrates to one; recall that an MoTBF density is defined without interval refinement). Thus, we can choose between models g_ℓ and g_m using an (approximate) BIC-score: The (approximate) ML parameters are found, and each model is penalized according to complexity. In the experiments reported in Section 5 we have used a greedy approach starting from g_0 and stopping as soon as an approximation g_k is better (in terms of BIC) than both g_{k+1} and g_{k+2}.[2] Our greedy procedure is exemplified in Figure 1, where an MoTBF is learned from 50 samples from a standard Gaussian distribution (shown as crosses on the x-axis). The kernel density approximation is drawn with a dashed line, and the MoTBF approximations from g_0 up to g_{10} are shown; g_5 is the best in terms of BIC-score, and is drawn with double line-width.

To fully specify the learning of univariate MoTBF distributions from data, we need to further analyze the use of kernel density approximations as an intermediate representation between the data and the learned MoTBF. The use of kernel estimators for learning MTEs was first proposed by Romero et al. (2006), and further analyzed by Langseth et al. (2010). The Epanechnikov kernel was found to offer the most consistent results in the case of MTE learning (Langseth et al., 2010), but we nevertheless use

[2] Computationally more demanding procedures can also be devised, e.g., to compare all subsets of basis functions from a fixed set $\{\psi_0, \psi_1, \ldots, \psi_k\}$.

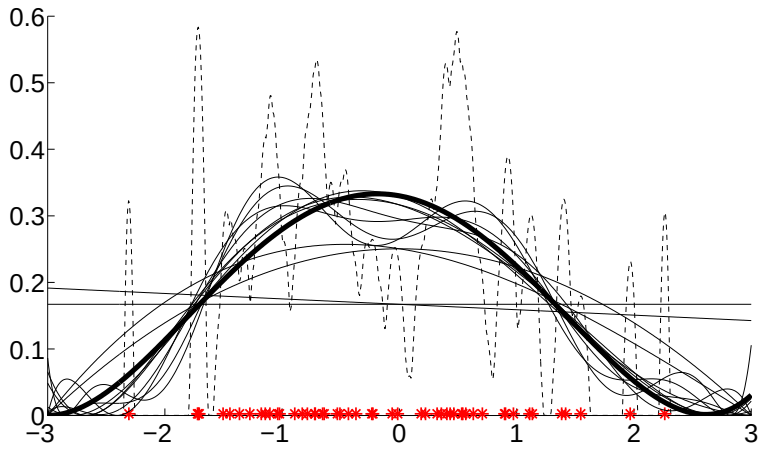

Figure 1: BIC-based learning: 50 samples from a standard Gaussian (crosses on the x-axis) are evaluated. The density estimator (thin dashed line) is the target of the MoTBF translations. g_0 up to g_{10} are shown; g_5 is the best in terms of BIC-score, and is drawn with double line-width.

the Gaussian kernel in our work to speed up the implementation. Previous attempts used Silverman's rule of thumb when selecting the bandwidth, $t_w^S \approx 1.06 \cdot \hat{\sigma} \cdot N^{-1/5}$, where $\hat{\sigma}$ is the empirical standard deviation of the dataset. By following that procedure, we get the results shown in Figure 2 (left-hand part of figure): a kernel density estimator is fitted to 50 samples from a standard Gaussian distribution, and the kernel density (drawn with the thin line) is then used as a starting point for the BIC-based MoTBF learning. The learned MoTBF representation is defined using 3 basis functions. Visually, the MoTBF approximation is quite poor (it does not resemble the standard Gaussian drawn with a dashed line), and we argue that the reason for the poor result is that using t_w^S is an unfortunate bandwidth choice, as it in principle leads us to smoothing the data *twice*: once when employing the kernel density, and once when the MoTBF is fitted to the kernel density. Rather, we want the kernel density to be a faithful representation of the data. To illustrate the effect, the right-hand part of Figure 2 shows the result of using the scaled bandwidth $t_w^S/25$. For this bandwidth, the BIC-score is optimized using 5 basis functions. The results are visually more appealing, and this is underlined when calculating the log likelihood of a hold-out set, giving -1541.02 and -1464.86 for the two bandwidths, respectively. We have investigated this further by examining a range of different datasets, both small and large, as defined by Langseth et al. (2010). For each dataset, we have learned an MoTBF representation using the BIC score for model selection and with a set of bandwidths defined by $t_w \leftarrow t_w^S/\alpha$, where $\alpha \in \{1, 2, 5, 10, 25, 50\}$ is the bandwidth scale. Table 1 lists the results of the experiment in terms of the number of basis functions that are selected as well as the obtained log likelihood on a hold-out dataset. The results appear to be robust across the data sets as long as the bandwidth is "sufficiently small". We have therefore used a fixed value of $t_w^S/10$ in the following, unless stated otherwise.

4 Learning conditional distributions

Recall that for a conditional MoTBF $f(x|\mathbf{z})$, the variables $\mathbf{Z}$ influence X only through the partitioning of $\Omega_{\mathbf{Z}}$, see Equation (2). Learning a conditional MoTBF therefore amounts to finding

- a partitioning of $\Omega_{\mathbf{Z}}$, and
- a (univariate) MoTBF for each hyper-cube in the partitioning.

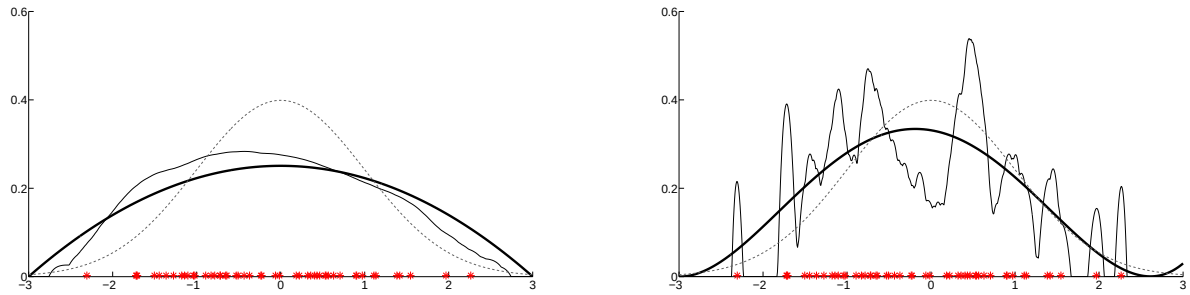

Figure 2: 50 samples from a Gaussian distribution are learned using a kernel density. The kernel density is shown in a thin line, where the bandwidth is the Silverman's rule of thumb (left figure) and one twenty-fifth of Silverman's rule of thumb (right). The learned MoTBF representations are defined using 3 and 5 basis functions in the left and right hand plots, respectively, and drawn in a thick line. Visually, the right-hand figure gives a better fit (compare to the standard Gaussian density drawn with the dashed line), and this is also the case when evaluated using log likelihood of a hold-out set (-1541.020 and -1465.585, respectively).

The algorithm for learning conditionals MoTBFs proceeds by iterating over the two steps above.

4.1 Finding an MoTBF for a fixed partitioning

For a given hyper-cube $\Omega^l_{\mathbf{Z}} \in \mathcal{P}$ we start by approximating the conditional empirical distribution with a conditional kernel density estimate. For ease of exposition, consider a variable Y with parent X for which we have a data sample $\mathcal{D} = \{\boldsymbol{d}_1, \ldots, \boldsymbol{d}_N\}$, where $\boldsymbol{d}_i = (x_i, y_i)$. We now define the conditional kernel density estimate for Y given X as

$$h_{\mathcal{D}}(y|x, t_{w_y}, t_{w_x}) = \frac{\sum_{i=1}^N h_{y_i}(y|t_{w_y}) h_{x_i}(x|t_{w_x})}{\sum_{i=1}^N h_{x_i}(x|t_{w_x})},$$

where $h_{x_i}(x|t_{w_x})$ is a kernel density estimator based on x_i only and with bandwidth t_{w_x}; $h_{y_i}(y|t_{w_y})$ is defined similarly. Given a conditional kernel density estimator $h_{\mathcal{D}}(y|x, t_{w_y}, t_{w_x})$ and a partitioning $\mathcal{P}$ of Ω_X, we approximate $h_{\mathcal{D}}(y|x, t_{w_y}, t_{w_x})$ with an MoTBF potential $f(y|x)$ (see Equation (2)) by following the procedure of Langseth et al. (2012). Thus, for all $\Omega^l_X \in \mathcal{P}$, we seek

$$f(y|x \in \Omega^l_X) \sim h_{\mathcal{D}}(y|x \in \Omega^l_X, t_{w_y}, t_{w_x}) = \int_x h_{\mathcal{D}}(y|x, t_{w_y}, t_{w_x}) h_{\mathcal{D}}(x|x \in \Omega^l_X, t_{w_x}) dx,$$

where the integral can be approximated by $\sum_{i=1}^n h_{\mathcal{D}}(y|x_i, t_{w_y}, t_{w_x}) h_{\mathcal{D}}(x_i|x_i \in \Omega^l_X, t_{w_x})$ using data samples $x_1, \ldots, x_n$ from $\mathcal{D}$ belonging to Ω^l_X. That is, for a fixed partitioning of Ω_X learning a conditional MoTBF potential reduces to estimating a univariate MoTBF potential (as described in Section 3) for each partition $\Omega^l_X \in \mathcal{P}$.

4.2 Finding a partitioning of the conditioning variables

In order to find a partitioning of $\Omega_{\mathbf{Z}}$ we employ a myopic strategy, where we in each step consider a bisection of an existing partition along each $Z \in \mathbf{Z}$. That is, for each partition $\Omega^l_{\mathbf{Z}} \in \mathcal{P}$ the algorithm evaluates the potential gain of splitting the partition along $Z \in \mathbf{Z}$. After scoring the candidate partitions the algorithm selects the highest scoring partition $\Omega^{\text{BS}}_{\mathbf{Z}}$ and splitting variable Z_{BS}, and learns MoTBF representations of the two induced sub-partitions $\Omega^{\text{BS},1}_{\mathbf{Z}}$ and $\Omega^{\text{BS},2}_{\mathbf{Z}}$. To guide the selection of a candidate partition $\Omega'_{\mathbf{Z}}$ we consider the potential improvement in BIC score resulting from splitting that partition:

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

where f' is the conditional MoTBF potential defined over the partitioning $\{\mathcal{P} \setminus \Omega'_{\mathbf{Z}}\} \cup$

Distribution	Scaler α	$N = 50$		$N = 1000$	
		#BF	Logllik	#BF	Loglik
MTE	1	4	-2430.82	5	-2349.78
	2	6	-2349.76	9	-2315.24
	5	6	-2349.76	12	-2308.68
	10	6	-2349.76	12	-2308.68
	25	6	-2349.76	9	-2315.24
	50	6	-2349.76	9	-2315.24
Beta(.5,.5)	1	5	180.06	7	228.97
	2	5	180.06	7	228.97
	5	3	80.92	7	228.97
	10	3	80.92	7	228.97
	25	5	180.06	5	206.18
	50	1	0.00	3	151.94
χ^2_8	1	2	-2887.23	4	-2803.42
	2	4	-2801.10	7	-2736.99
	5	5	-2747.73	10	-2711.65
	10	5	-2747.73	10	-2711.65
	25	5	-2747.73	10	-2711.65
	50	5	-2747.73	10	-2711.65
Gauss(0,1)	1	3	-1541.02	6	-1428.35
	2	5	-1464.86	7	-1420.33
	5	5	-1464.86	7	-1420.33
	10	5	-1464.86	7	-1420.33
	25	5	-1464.86	7	-1420.33
	50	3	-1541.02	5	-1434.28
Log-Norm(0,1)	1	8	-1434.51	7	-1393.15
	2	8	-1434.51	8	-1390.39
	5	8	-1434.51	8	-1378.77
	10	5	-1523.29	9	-1378.77
	25	5	-1523.29	8	-1390.39
	50	5	-1523.29	8	-1390.39

Table 1: The effect of the chosen bandwidth wrt. log likelihood of a test-set. In general, we see that results for "large" bandwidths ($\alpha = 1$) are poor due to "double smoothing". Additionally, results using large scalers ($\alpha = 50$) are also sometimes unsatisfactory; typically due to numerical instabilities in the solution method due to the peakedness of the kernel approximation, which in turn leads to numerically unstable calculations.

$\{\Omega_{\mathbf{Z}}^{Z,1}, \Omega_{\mathbf{Z}}^{Z,2}\}$. In principle, when scoring the model f' one would need to find the basis functions (and the corresponding parameters) maximizing this score. This will, however, be computationally difficult, and instead we lower-bound the improvement in BIC score by using the same set of basis functions as was used for the parent partition $\Omega'_{\mathbf{Z}}$. It should be noted that for the calculation of the improvement in BIC score, we only need to consider the parts of the score relating to the partition $\Omega'_{\mathbf{Z}}$, since the contributions from the partitions for which f and f' agree cancel out; this property also supports an efficient caching scheme for BIC-Gain. The overall procedure for learning conditional MoTBFs is summarized in Algorithm 1.

Algorithm 1 Learning conditional MoTBFs.

$\mathcal{P} \leftarrow \{\Omega_{\mathbf{Z}}\}$
repeat
  $(\Omega_{\mathbf{Z}}^{\mathrm{BS}}, Z_{\mathrm{BS}}) \leftarrow$
    $\arg\max_{\Omega'_{\mathbf{Z}} \in \mathcal{P}, Z \in \mathbf{Z}} \text{BIC-Gain}(\Omega'_{\mathbf{Z}}, Z)$
  if BIC-Gain$(\Omega_{\mathbf{Z}}^{\mathrm{BS}}, Z_{\mathrm{BS}}) > 0$ **then**
    Learn MoTBF potentials for
      $\Omega_{\mathbf{Z}}^{\mathrm{BS},1}$ and $\Omega_{\mathbf{Z}}^{\mathrm{BS},2}$.
    $\mathcal{P} \leftarrow \left\{\mathcal{P} \setminus \Omega_{\mathbf{Z}}^{\mathrm{BS}}\right\} \cup \left\{\Omega_{\mathbf{Z}}^{\mathrm{BS},1}, \Omega_{\mathbf{Z}}^{\mathrm{BS},2}\right\}$.
  else
    terminate.
  end if
until false

5 Experiments

In this section we will report on two small experimental studies undertaken to compare the merits of the proposed method to its most immediate competitors. Firstly, we will compare the new method of learning marginal MoTBF densities to the results obtained by Romero et al. (2006) and Langseth et al. (2010), as reported by Langseth et al. (2010). Datasets, each containing 1000 training examples, generated from five different distributions were used. Table 2 reports the log likelihood of each dataset using the obtained estimator of each of the techniques. We have used the polynomials as basis functions, meaning that ψ_ℓ in Equations (1) and (2) is the (scaled and stretched) Legendre polynomial of order ℓ. The number of basis functions was chosen so that the number of free parameters corresponds to the number of parameters used by Langseth et al. (2010); recall that the MoTBF distribution g_k on $\Omega_{\mathbf{Z}}$ is specified using $k + 2$ parameters. Note that where the methods by Romero et al. (2006) and Langseth et al. (2010) divide the support of the distribution into sub-intervals and fit one model per interval, the current approach does not use interval refinement. This may harm the fit of the MoTBF distributions, when the gold-standard distribu-

Dataset	#par	Romero et al. (2006)	Langseth et al. (2010)	Current approach
MTE	12	-2556.68	-2263.13	-2272.71
Beta	12	39.42	160.69	159.74
χ^2_8	24	-2766.86	-2685.76	-2710.35
Gaussian	12	-1565.28	-1420.34	-1387.73
Log-Normal	24	-1636.99	-1398.30	-1373.73

Table 2: The obtained log likelihood of the training data when learning from 1000 samples from different distributions.

tion is not continuous. In general, though, the results of the new method seem to be better than those by Romero et al. (2006), and comparable to those by Langseth et al. (2010).

The speedup from the direct maximum likelihood approach (Langseth et al., 2010) to our approach is above a factor 10. The main contribution to the speed increase is that while the previous technique was based on an iterative scheme, where each potential solution (living in a very complicated likelihood-landscape) needed to be evaluated using a computationally expensive procedure, the current approach casts the learning problem as a convex optimization problem, with much cheaper evaluations.

Next, we compare the predictive performance of the method in Langseth et al. (2010) to ours. We used the same training data as reported in Table 2, but this time used the (approximate) BIC score for model selection. The chosen models were examined by calculating the log likelihood of a separate dataset of 1000 cases.

Dataset	Langseth et al. (2010)	Current approach
MTE	-2285.25	-2308.68
Beta	249.45	228.97
χ^2_8	-2702.67	-2711.65
Gaussian	-1430.88	-1420.33
Log-Normal	-1358.38	-1378.77

Table 3: Test-set log likelihood of estimators after learning models using the BIC score.

The results in Table 3 indicate that the method by Langseth et al. (2010) is slightly better than our procedure, but the speedup of the current approach is more than a factor 15. The main source of the extra speed-increase is that the relatively costly initialization of the MoTBF technique (finding and representing the orthonormal basis functions) needs not be performed each time a new candidate model is evaluated.

Finally, we exemplify the learning of conditional distributions by generating data from a model where X is standard Gaussian, and $Y|\{X = x\} \sim \mathcal{N}(x/2, 1)$. Datasets containing 50, 500, 2500 and 5000 cases were generated, and given to Algorithm 1. The resulting conditional distributions are shown in Figure 3, with the parent on the x-axis and the child on the y-axis. For the smallest dataset of 50 cases, the BIC-score only gave support for a single split-point, inserted at the midpoint of the support for X. As the size of the training-sets increases, the BIC score selects more and more refined models, allowing itself to use more parameters to represent the correlation between X and Y as it is more and more clearly manifested in the training data. Notice that the algorithm uses more effort on refining the model where the bulk of the data is found (i.e., around $x \approx 0$).

6 Conclusions

In this paper we have examined a new technique for (approximately) learning maximum likelihood parameters of a hybrid Bayesian network from data. The main idea is to find a kernel density estimate of the data and utilize an effective "translation" procedure designed for approximating any marginal or conditional distribution function (in this case a kernel density) by an MoTBF distribution. Although the method was found to be slightly worse than state-of-the-art techniques in terms of the log-likelihood score, the speed-up over previous methods is substantial, and we are currently investigating how the method scales to larger domains.

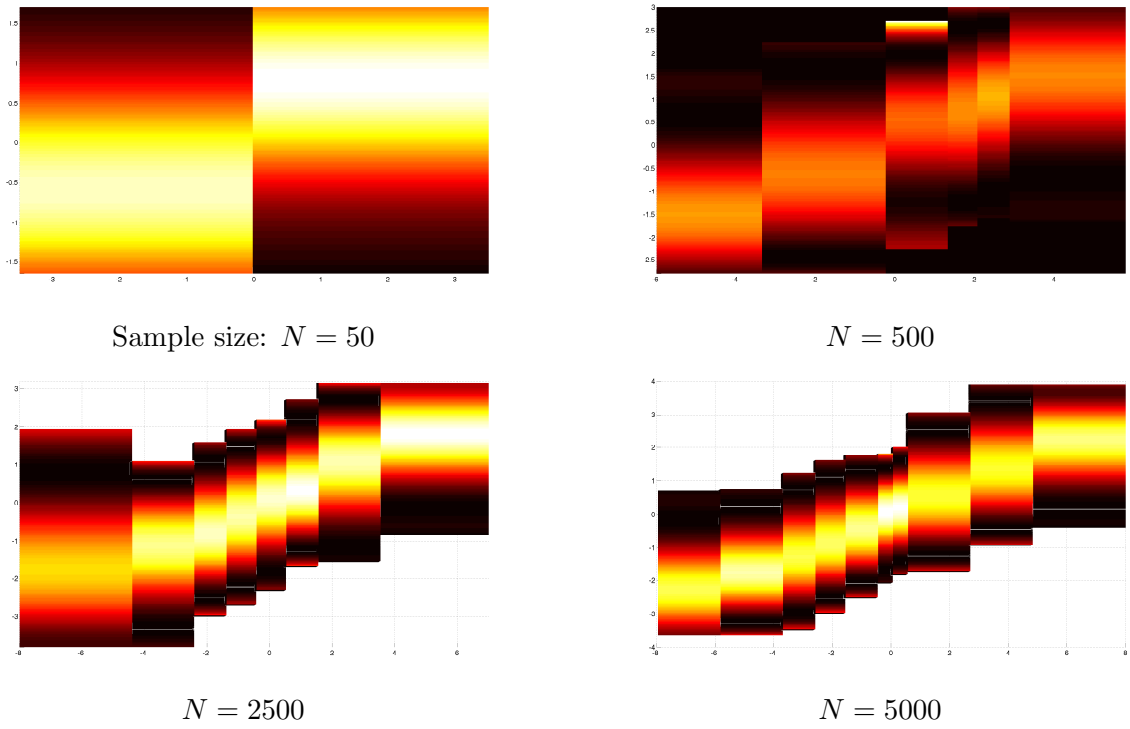

Figure 3: Learning a conditional linear Gaussian distribution using BIC score. Note how finer model granularity is selected as the size of training-set grows, and how the discretization effort is is kept to the area with the bulk of the data.

Acknowledgments

This work has been supported by a Senior Grant in the frame of the CALL UCM-EEA-ABEL-02-2009 of the Abel Extraordinary Chair (NILS Project), and by the Spanish Ministry of Science and Innovation, through projects TIN2010-20900-C04-02,03 (entitled Data mining with PGMs: New algorithms and applications) and by ERDF (FEDER) funds.

References

Langseth, H., Nielsen, T., Rumí, R., and Salmerón, A. (2009). Maximum likelihood learning of conditional MTE distributions. *ECSQARU 2009. Lecture Notes in Computer Science*, 5590:240–251.

Langseth, H., Nielsen, T., Rumí, R., and Salmerón, A. (2010). Parameter estimation and model selection for mixtures of truncated exponentials. *International Journal of Approximate Reasoning*, 51:485–498.

Langseth, H., Nielsen, T., Rumí, R., and Salmerón, A. (2012). Mixtures of truncated basis functions. *International Journal of Approximate Reasoning*, 53:212–227.

Moral, S., Rumí, R., and Salmerón, A. (2001). Mixtures of truncated exponentials in hybrid Bayesian networks. In *ECSQARU'01. Lecture Notes in Artificial Intelligence*, volume 2143, pages 135–143.

Romero, V., Rumí, R., and Salmerón, A. (2006). Learning hybrid Bayesian networks using mixtures of truncated exponentials. *International Journal of Approximate Reasoning*, 42:54–68.

Shafer, G. R. and Shenoy, P. P. (1990). Probability propagation. *Annals of Mathematics and Artificial Intelligence*, 2:327–352.

Shenoy, P. and West, J. (2011). Inference in hybrid Bayesian networks using mixtures of polynomials. *International Journal of Approximate Reasoning*, 52:641–657.