Parameter Estimation in Mixtures of Truncated Exponentials

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

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

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

Abstract

Bayesian networks with mixtures of truncated exponentials (MTEs) support efficient inference algorithms and provide a flexible way of modeling hybrid domains. On the other hand, estimating an MTE from data has turned out to be a difficult task, and most prevalent learning methods treat parameter estimation as a regression problem. The drawback of this approach is that by not directly attempting to find the parameters that maximize the likelihood, there is no principled way of e.g. performing subsequent model selection using those parameters. In this paper we describe an estimation method that directly aims at learning the maximum likelihood parameters of an MTE potential. Empirical results demonstrate that the proposed method yields significantly better likelihood results than regression-based methods.

1 Introduction

In domains involving both discrete and continuous variables, Bayesian networks with mixtures of truncated exponentials (MTE) (Moral et al., 2001) have received increasing interest over the last few years. Not only do MTE distributions allow discrete and continuous variables to be treated in a uniform fashion, but since the family of MTEs is closed under addition and multiplication, inference in an MTE network can be performed efficiently using the Shafer-Shenoy architecture (Shafer and Shenoy, 1990).

Despite its appealing approximation and inference properties, data-driven learning methods for MTE networks have received only little attention. In this context, focus has mainly been directed towards parameter estimation, where the most prevalent methods look for the MTE parameters minimizing the mean squared error w.r.t. a kernel density estimate of the data (Romero et al., 2006).

Although the least squares estimation procedure can yield a good MTE model in terms of generalization properties, there is no guarantee that the estimated parameters will be close to the maximum likelihood (ML) parameters. This has a significant impact when considering more general problems such as model selection and structural learning. Standard score functions for model selection include e.g. the Bayesian information criterion (BIC) (Schwarz, 1978), which is a form of penalized log-likelihood. However, the BIC score assumes ML parameter estimates, and since there is no justification for treating the least squares parameter estimates as ML parameters, there is in turn no theoretical foundation for using a least squared version of the BIC score.[1]

In this paper we propose a new parameter es-

[1]Learning the general form of an MTE can also be posed as a model selection problem, where we look for the number of exponential terms as well as appropriate split points. Hence, the problem also appears in this simpler setting.

timation method for univariate MTE potentials that directly aims at estimating the ML parameters for an MTE density with predefined structure (detailed below). The proposed method is empirically compared to the least squares estimation method described in (Romero et al., 2006), and it is shown that it offers a significant improvement in terms of likelihood.

The method described in this paper should be considered as a first step towards a general maximum likelihood-based approach for learning Bayesian networks with MTE potentials. Thus, we shall only hint at some of the difficulties (complexity-wise) that are involved in learning general conditional MTE potentials, and instead leave this topic as well as structural learning as subjects for future work.

2 Preliminaries

Throughout this paper, random variables will be denoted by capital letters, and their values by lowercase letters. In the multi-dimensional case, boldfaced characters will be used. The domain of the variable $\mathbf{X}$ is denoted by $\Omega_{\mathbf{X}}$. The MTE model is defined by its corresponding potential and density as follows (Moral et al., 2001):

Definition 1 (MTE potential) *Let $\mathbf{X}$ be a mixed n-dimensional random vector. Let $\mathbf{W} = (W_1, \ldots, W_d)$ and $\mathbf{Z} = (Z_1, \ldots, Z_c)$ be the discrete and continuous parts of $\mathbf{X}$, respectively, with $c + d = n$. We say that a function $f : \Omega_{\mathbf{X}} \mapsto \mathbb{R}_0^+$ is a* Mixture of Truncated Exponentials (MTE) potential *if for each fixed value $\mathbf{w} \in \Omega_{\mathbf{W}}$ of the discrete variables $\mathbf{W}$, the potential over the continuous variables $\mathbf{Z}$ is defined as:*

$$f(\mathbf{z}) = a_0 + \sum_{i=1}^{m} a_i \exp\left\{ \sum_{j=1}^{c} b_i^{(j)} z_j \right\}, \qquad (1)$$

for all $\mathbf{z} \in \Omega_{\mathbf{Z}}$, where a_i, $i = 0, \ldots, m$ and $b_i^{(j)}$, $i = 1, \ldots, m$, $j = 1, \ldots, c$ are real numbers. We also say that f is an MTE potential if there is a partition $D_1, \ldots, D_k$ of $\Omega_{\mathbf{Z}}$ into hypercubes and in each D_i, f is defined as in Eq. (1).

An MTE potential is an *MTE density* if it integrates up to 1.

In the remainder of this paper we shall focus on estimating the parameters for a univariate MTE density. Not surprisingly, the proposed methods also immediately generalize to the special case of conditional MTEs having only discrete conditioning variables.

3 Estimating Univariate MTEs from Data

The problem of estimating a univariate MTE density from data can be divided into three tasks: *i*) partitioning the domain of the variable, *ii*) determining the number of exponential terms, and *iii*) estimating the parameters for a given partition of the domain and a fixed number of exponential terms. At this point we will concentrate on the estimation of the parameters, assuming that the split points are known, and that the number of exponential terms is fixed.

We start this section by introducing some notation: Consider a random variable X with density function $f(x)$ and assume that the support of $f(x)$ is divided into M intervals $\{\Omega_i\}_{i=1}^{M}$. Focus on one particular interval Ω_m. As a target density for $x \in \Omega_m$ we will consider an MTE with 2 exponential terms:

$$f(x|\boldsymbol{\theta}_m) = k_m + a_m e^{b_m x} + c_m e^{d_m x}, \ x \in \Omega_m. \qquad (2)$$

This function has 5 free parameters, namely $\boldsymbol{\theta}_m = (k_m, a_m, b_m, c_m, d_m)$. For notational convenience we may sometimes drop the subscript m when clear from the context.

3.1 Parameter Estimation by Maximum Likelihood

Assume that we have a sample $\mathbf{x} = \{x_1, \ldots, x_n\}$ and that n_m of the n observations are in Ω_m. To ensure that the overall parameter-set is a maximum likelihood estimate for $\boldsymbol{\Theta} = \cup_m \boldsymbol{\theta}_m$, it is required that

$$\int_{x \in \Omega_m} f(x|\boldsymbol{\theta}_m)\, dx = n_m/n. \qquad (3)$$

Given this normalization, we can fit the parameters for each interval Ω_m separately, i.e., the parameters in $\boldsymbol{\theta}_m$ are optimized independently of those in $\boldsymbol{\theta}_{m'}$. Based on this observation, we shall only describe the learning procedure for *a fixed interval* Ω_m, since the generalization to the whole support of $f(x)$ is immediate.

Assume now that the target density is as given in Eq. (2), in which case the likelihood function for a sample $\mathbf{x}$ is

$$L(\boldsymbol{\theta}_m|\mathbf{x}) = \prod_{i=1}^{n} \left\{ k_m + a_m e^{b_m x_i} + c_m e^{d_m x_i} \right\}. \quad (4)$$

To find a closed-form solution for the maximum likelihood parameters, we need to differentiate Eq. (4) wrt. the different parameters and set the results equal to zero. To exemplify, we perform this exercise for b_m, and obtain

$$\begin{aligned} \frac{\partial L(\boldsymbol{\theta}_m|\mathbf{x})}{\partial b_m} &= \sum_{i=1}^{n} \left\{ \frac{\partial L(\boldsymbol{\theta}_m|x_i)}{\partial b_m} \prod_{j \neq i} L(\boldsymbol{\theta}_m|x_j) \right\} \\ &= a_m b_m \sum_{i=1}^{n} e^{b_m x_i} \left\{ \prod_{j \neq i} \left(k_m + a_m e^{b_m x_j} \right.\right. \\ &\qquad \left.\left. + c_m\, e^{d_m\, x_j} \right) \right\}. \end{aligned} \quad (5)$$

Unfortunately, Eq. (5) is non-linear in the unknown parameters $\boldsymbol{\theta}_m$. Furthermore, both the number of terms in the sum as well as the number of terms inside the product operator grows as $O(n)$; thus, the maximization of the likelihood becomes increasingly difficult as the number of observations rise.

Alternatively, one might consider maximizing the logarithm of the likelihood, or more specifically a *lower bound* for the likelihood using Jensen's inequality. By assuming that $a_m > 0$ and $c_m > 0$ we have

$$\begin{aligned} \log\left(L(\boldsymbol{\theta}_m|\mathbf{x})\right) &= \sum_{i=1}^{n} \log\left(k_m + a_m\, \exp(b_m\, x_j)\right. \\ &\qquad \left. + c_m\, \exp(d_m\, x_j)\right) \\ &\geq \sum_{i=1}^{n} \log\left(k_m\right) + \sum_{i=1}^{n} \log\left(a_m\, \exp(b_m\, x_j)\right) \\ &\qquad + \sum_{i=1}^{n} \log\left(c_m\, \exp(d_m\, x_j)\right) \\ &= n\left[\log(k_m) + \log(a_m) + \log(c_m)\right] \\ &\qquad + (b_m + d_m) \sum_{i=1}^{n} x_i, \end{aligned} \quad (6)$$

and the idea would then be to maximize the lowerbound of Eq. (6) to push the likelihood upwards (following the same reasoning underlying the EM algorithm (Dempster et al., 1977) and variational methods (Jordan et al., 1999)). Unfortunately, though, restricting both a_m and c_m to be positive enforces too strict a limitation on the expressiveness of the distributions we learn.

Instead, an approximate solution can be obtained by solving the likelihood equations by numerical means. The proposed method for maximizing the likelihood is based on the observation that maximum likelihood optimization for MTEs can be seen as a constrained optimization problem, where constraints are introduced to ensure that both $f(x|\boldsymbol{\theta}_m) \geq 0$, for all $x \in \Omega_m$, and that Eq. (3) is fulfilled. A natural framework for solving this is the Lagrange multipliers, but since solving the Lagrange equations are inevitably at least as difficult as solving the unconstrained problem, this cannot be done analytically. In our implementation we have settled for a numerical solution based on Newton's method; this is described in detail in Section 3.1.2. However, it is well-known that Newton's method is quite sensitive to the initialization-values, meaning that if we initialize a search for a solution to the Lagrange equations from a parameter-set far from the optimal values, it will not necessarily converge to a useful solution. Thus, we need a simple and robust procedure for initializing Newton's method, and this is described next.

3.1.1 Naïve Maximum Likelihood in MTE Distributions

The general idea of the optimization is to iteratively update the parameter estimates until convergence. More precisely, this is done by iteratively tuning *pairs* of parameters, while the other parameters are kept fixed. We do this in a round-robin manner, making sure that all parameters are eventually tuned. Denote by $\hat{\boldsymbol{\theta}}_m^t = \left(k^t, a^t, b^t, c^t, d^t\right)$ the parameter values after iteration t of this iterative scheme. Algorithm 3.1 is a top-level description of this procedure, where steps 3 and 4 correspond to the optimization of the shape-parameters and steps

5 and 6 distribute the mass between the five terms in the MTE potential (the different steps are explained below).

Algorithm 3.1 ML estimation

1: Initialize $\hat{\theta}_m^0$; $t \leftarrow 0$.
2: **repeat**
3: $\quad (a', b') \leftarrow \arg\max_{a,b} L(k^t, a, b, c^t, d^t \,|\, \mathbf{x})$
4: $\quad (c', d') \leftarrow \arg\max_{c,d} L(k^t, a', b', c, d \,|\, \mathbf{x})$
5: $\quad (k', a') \leftarrow \arg\max_{k,a} L(k, a, b', c', d' \,|\, \mathbf{x})$
6: $\quad (k', c') \leftarrow \arg\max_{k,c} L(k, a', b', c, d' \,|\, \mathbf{x})$
7: $\quad (k^{t+1}, a^{t+1}, b^{t+1}, c^{t+1}, d^{t+1},) \leftarrow (k', a', b', c, d')$
8: $\quad t \leftarrow t + 1$
9: **until** convergence

For notational convenience we shall define the auxiliary function $p(s,t) = \int_{x \in \Omega_m} s \exp(tx)\, dx$; $p(s,t)$ is the integral of the exponential function over the interval Ω_m. Note, in particular, that $p(s,t) = s \cdot p(1,t)$, and that $p(1,0) = \int_{x \in \Omega_m} dx$ is the length of the interval Ω_m. The first step above is initialization. In our experiments we have chosen b^0 and d^0 as $+1$ and -1 respectively. The parameters k^0, a^0, and c^0 are set to ensure that each of the three terms in the integral of Eq. (3) contribute with equal probability mass, i.e.,

$$\begin{aligned} k^0 &\leftarrow \frac{n_m}{3n \cdot p(1,0)}, \\ a^0 &\leftarrow \frac{n_m}{3n \cdot p(1,b^0)}, \\ c^0 &\leftarrow \frac{n_m}{3n \cdot p(1,d^0)}. \end{aligned}$$

Iteratively improving the likelihood under the constraints is actually quite simple as long as the parameters are considered in pairs. Consider Step 3 above, where we optimize a and b under the constraint of Eq. (3) while keeping the other parameters (k^t, c^t, and d^t) fixed. Observe that if Eq. (3) is to be satisfied after this step we need to make sure that $p(a', b') = p(a^t, b^t)$. Equivalently, there is a functional constraint between the parameters that we enforce by setting $a' \leftarrow p(a^t, b^t)/p(1, b')$. Optimizing the value for the pair (a, b) is now simply done by line-search, where only the value for b is considered:

$$b' = \arg\max_b L(k, \frac{p(a^t, b^t)}{p(1,b)}, b, c^t, d^t | \mathbf{x}).$$

Note that at the same time we choose $a' \leftarrow p(a^t, b^t)/p(1, b')$. A similar procedure is used in Step 4 to find c' and d'.

Steps 5 and 6 utilize the same idea, but with a different normalization equation. We only consider Step 5 here, since the generalization is immediate. For this step we need to make sure that $\int_{x \in \Omega_m} k + a \exp(b' x)\, dx = \int_{x \in \Omega_m} k^t + a^t \exp(b' x)\, dx$, for any pair of parameter candidates (k, a). By rephrasing, we find that this is obtained if we insist that $k' \leftarrow k^t - p(a' - a^t, b')/p(1, 0)$. Again, the constrained optimization of the pair of parameters can be performed using line-search in one dimension (and let the other parameter be adjusted to keep the total probability mass constant).

Note that Steps 3 and 4 do not move "probability mass" between the three terms in Eq. (2), these two steps only fit the shape of the two exponential functions. On the other hand, Steps 5 and 6 assume the shape of the exponentials fixed, and proceed by moving "probability mass" between the three terms in the sum of Eq. (2).

3.1.2 Refining the Initial Estimate

The parameter estimates returned by the line-search method can be further refined by using these estimates to initialize a nonlinear programming problem formulation of the original optimization problem. In this formulation, the function to be maximized is again the log-likelihood of the data, subject to the constraints that the MTE potential should be nonnegative, and that

$$g_0(\mathbf{x}, \boldsymbol{\theta}) \equiv \int_{x \in \Omega_m} f(x \,|\, \boldsymbol{\theta})\, dx - \frac{n_m}{n} = 0.$$

Ideally the nonnegative constraints should be specified for all $x \in \Omega_m$, but since this is not feasible we only encode that the function should be nonnegative in the endpoints e_1 and e_2 of the interval (we shall return to this issue later).

Thus, we arrive at the following formulation:

$$\begin{aligned}
&\text{Maximize } \log L(\boldsymbol{\theta} \mid \mathbf{x}) = \sum_{i=1}^{n} \log L(\boldsymbol{\theta} \mid x_i) \\
&\text{Subject to } g_0(\mathbf{x}, \boldsymbol{\theta}) = 0, \\
&\qquad\qquad f(e_1 \mid \boldsymbol{\theta}) \geq 0, \\
&\qquad\qquad f(e_2 \mid \boldsymbol{\theta}) \geq 0,
\end{aligned}$$

To convert the two inequalities into equalities we introduce slack variables:

$$f(x \mid \boldsymbol{\theta}) \geq 0 \Leftrightarrow f(x \mid \boldsymbol{\theta}) - s^2 = 0, \text{ for some } s \in \mathbb{R};$$

we shall refer to these new equalities using $g_1(e_1, \boldsymbol{\theta}, s_1)$ and $g_2(e_2, \boldsymbol{\theta}, s_2)$, respectively. We now have the following equality constrained optimization problem:

$$\begin{aligned}
&\text{Maximize } \log L(\boldsymbol{\theta} \mid \mathbf{x}) = \sum_{i=1}^{n} \log L(\boldsymbol{\theta} \mid x_i) \\
&\text{Subject to } \mathbf{g}(\mathbf{x}, \boldsymbol{\theta}) = \begin{bmatrix} g_0(\mathbf{x}, \boldsymbol{\theta}) \\ g_1(e_1, \boldsymbol{\theta}, s_1) \\ g_2(e_2, \boldsymbol{\theta}, s_2) \end{bmatrix} = 0.
\end{aligned}$$

This optimization problem can be solved using the method of Lagrange multipliers. That is, with the Lagrangian function $l(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{s}) = \log L(\boldsymbol{\theta} \mid \mathbf{x}) + \lambda_0 g_0(x, \boldsymbol{\theta}) + \lambda_1 g_1(x, \boldsymbol{\theta}, s_1) + \lambda_2 g_2(x, \boldsymbol{\theta}, s_2)$ we look for a solution to the equalities defined by

$$A(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{s}) = \nabla l(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{s}) = 0.$$

Such a solution can be found numerically by applying Newton's method. Specifically, by letting $\boldsymbol{\theta}' = (\boldsymbol{\theta}, s_1, s_2)^{\mathrm{T}}$, the Newton updating step is given by

$$\begin{bmatrix} \boldsymbol{\theta}'_{t+1} \\ \boldsymbol{\lambda}_{t+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\theta}'_{t} \\ \boldsymbol{\lambda}_{t} \end{bmatrix} - \nabla A(\mathbf{x}, \boldsymbol{\theta}'_t, \boldsymbol{\lambda}_t)^{-1} A(\mathbf{x}, \boldsymbol{\theta}'_t, \boldsymbol{\lambda}_t),$$

where $\boldsymbol{\theta}'_t$ and $\boldsymbol{\lambda}_t$ are the current estimates and

$$A(\mathbf{x}, \boldsymbol{\theta}'_t, \boldsymbol{\lambda}_t) = \begin{bmatrix} \nabla_{\boldsymbol{\theta}'} l(\mathbf{x}, \boldsymbol{\theta}', \boldsymbol{\lambda}) \\ \mathbf{g}(\mathbf{x}, \boldsymbol{\theta}') \end{bmatrix};$$

$$\nabla A(\mathbf{x}, \boldsymbol{\theta}'_t, \boldsymbol{\lambda}_t) = \begin{bmatrix} \nabla^2_{\boldsymbol{\theta}'\boldsymbol{\theta}'} l(\mathbf{x}, \boldsymbol{\theta}', \boldsymbol{\lambda}) & \nabla \mathbf{g}(\mathbf{x}, \boldsymbol{\theta}') \\ \nabla \mathbf{g}(\mathbf{x}, \boldsymbol{\theta}')^{\mathrm{T}} & 0 \end{bmatrix}.$$

As initialization values, $\boldsymbol{\theta}_0$, we use the maximum likelihood estimates returned by the line-search method described in Section 3.1, and in order to control the step size during updating, we employ the Armijo rule (Bertsekas, 1996). For the test results reported in Section 4, the Lagrange multipliers were initialized (somewhat arbitrarily) to 1 and the slack variables were set to $\sqrt{f(e_1 \mid \boldsymbol{\theta}_0)}$ and $\sqrt{f(e_2 \mid \boldsymbol{\theta}_0)}$, respectively.

Finally, it should be emphasized that the above search procedure may lead to $f(x \mid \boldsymbol{\theta})$ being negative for some x. In the current implementation we have addressed this problem rather crudely: simply terminate the search when negative values are encountered. Moreover, due to numerical instability, the search is also terminated if the determinant for the system is close to zero ($< 10^{-9}$) or if the condition number is large ($> 10^9$). Note that by terminating the search before convergence, we have no guarantees about the solution. In particular, the solution may be worse than the initial estimate. In order to overcome this problem, we always store the best parameter estimates found so far (including those found by line search) and return these estimates upon termination.

3.2 Parameter Estimation by Least Squares

Least squares (LS) estimation is based on finding the values of the parameters that minimize the mean squared error between the fitted model and the empirical density of the sample. In earlier work on MTE parameter estimation (Rumí et al., 2006), the empirical density was estimated using a histogram. In order to avoid the lack of smoothness, especially when data is scarce, (Romero et al., 2006) proposed to use kernels to approximate the empirical density instead of histograms.

As the LS method does not directly seek to maximize the likelihood of the model, the resulting LS parameters are not guaranteed to be close to the ML parameters. This difference was confirmed by our preliminary experiments, and has resulted in a few modifications to the LS method presented in (Rumí et al., 2006; Romero et al., 2006): *i*) Instead of using Gausian kernels, we used Epanechnikov kernels, which tended to provide better ML estimates in our preliminary experiments. *ii*) Since the smooth kernel density estimate assigns positive probability mass, p^*, outside the truncated region (called the boundary bias (Simonoff, 1996)), we reweigh the kernel density with $1/(1-p^*)$. *iii*) In order to reduce the effect of low probability areas, the summands in the mean squared error are weighted according to the empirical density at the corresponding points.

3.2.1 The Weighted LS Algorithm

In what follows we denote by $\mathbf{y} = \{y_1, \ldots, y_n\}$ the values of the empirical kernel for sample $\mathbf{x} = \{x_1, \ldots, x_n\}$, and with reference to the target density in Eq. (2), we assume initial estimates for a_0, b_0 and k_0 (we will later discuss how to get these initial estimates). With this outset, c and d can be estimated by minimizing the *weighted mean squared error* between the function $c \exp\{dx\}$ and the points $(\mathbf{x}, \mathbf{w})$, where $\mathbf{w} = \mathbf{y} - a_0 \exp\{b_0 \mathbf{x}\} - k_0$. Specifically, by taking logarithms, the problem reduces to linear regression:

$$\ln\{w\} = \ln\{c \exp\{dx\}\} = \ln\{c\} + dx,$$

which can be written as $w^* = c^* + dx$; here $c^* = \ln\{c\}$ and $w^* = \ln\{w\}$. Note that we here assume that $c > 0$. In fact the data $(\mathbf{x}, \mathbf{w})$ is transformed, if necessary, to fit this constraint, i.e., to be convex and positive. This is achieved by changing the sign of the values $\mathbf{w}$ and then adding a constant to make them positive. We then fit the parameters taking into account that afterwards the sign of c should be changed and the constant used to make the values positive should be subtracted.

A solution to the regression problem is then defined by

$$(c^*, d) = \arg\min_{c^*, d} \sum_{i=1}^{n} (w_i^* - c^* - dx_i)^2 y_i,$$

which can be described analytically:

$$c^* = \frac{\left(\sum_{i=1}^n w_i x_i y_i\right) - d\left(\sum_{i=1}^n x_i y_i\right)^2}{\left(\sum_{i=1}^n x_i y_i\right)}$$

$$d = \frac{\left(\sum_{i=1}^n w_i y_i\right)\left(\sum_{i=1}^n x_i y_i\right) - \left(\sum_{i=1}^n y_i\right)\left(\sum_{i=1}^n w_i x_i y_i\right)}{\left(\sum_{i=1}^n x_i y_i\right)^2 - \left(\sum_{i=1}^n y_i\right)\left(\sum_{i=1}^n x_i^2 y_i\right)}.$$

Once a, b, c and d are known, we can estimate k in $f^*(x) = k + ae^{bx} + ce^{dx}$, where $k \in \mathbb{R}$ should minimize the error

$$E(k) = \sum_{i=1}^{n} \frac{(y_i - ae^{bx_i} - ce^{dx_i} - k)^2 y_i}{n}.$$

This is achieved for

$$\hat{k} = \frac{\sum_{i=1}^n (y_i - ae^{bx_i} - ce^{dx_i}) y_i}{\sum_{i=1}^n y_i}.$$

Here we are assuming a fixed number of exponential terms. However, as the parameters are not optimized globally, there is no guarantee that the fitted model minimizes the *weighted mean squared error*. This fact can be somewhat corrected by determining the contribution of each term to the reduction of the error as described in (Rumí et al., 2006).

The initial values a_0, b_0 and k_0 can be arbitrary, but "good" values can speed up convergence. We consider two alternatives: *i*) Initialize the values by fitting a curve ae^{bx} to the modified sample by exponential regression, and compute k as before. *ii*) Force the empiric density and the initial model to have the same derivative. In the current implementation, we try both initializations and choose the one that minimizes the squared error.

4 Experimental Comparison

In order to compare the behaviour of both approaches we have used 6 samples of size 1000 taken from the following distributions: an MTE density defined by two regions, a beta distribution *Beta*(0.5, 0.5), a standard normal distribution, a χ^2 distribution with 8 degrees of freedom, and a log-normal distribution *LN*(0, 1).

	MTE	Beta	χ^2	Normal 2 splits	Normal 4 splits	Log-normal
ML	−2263.37	160.14	−2695.02	−1411.79	−1380.45	−1415.06
LS	−2307.21	68.26	−2739.24	−1508.62	−1403.46	−1469.21
Original LS	−2338.46	39.68	−2718.99	−1570.62	−1406.23	−1467.24

	MTE	Beta	χ^2	Normal 2 splits	Normal 4 splits	Log-normal
ML	−2263.13	160.69	−2685.76	−1420.34	−1392.28	−1398.3
LS	−2321.18	60.29	−2742.80	−1509.11	−1468.11	−2290.17
Original LS	−2556.68	39.42	−2766.86	−1565.28	−1438.67	−1636.99

Table 1: Comparison of ML vs. LS in terms of likelihood. In the upper table the split points were found using the method described in (Rumí et al., 2006), and in the lower table they were defined by the extreme points and the inflexion points of the exact density.

For the MTE, beta and normal distributions, we have used two split points, whereas for the log-normal and the χ^2 distributions, the number of splits points was set to 4. We have also run the experiment with four split points for the standard normal distribution.

The plots of the fitted models, together with the original density as well as the empirical histograms, are displayed in Figure 1. The split points used for these figures were selected using the procedure described in (Rumí et al., 2006).

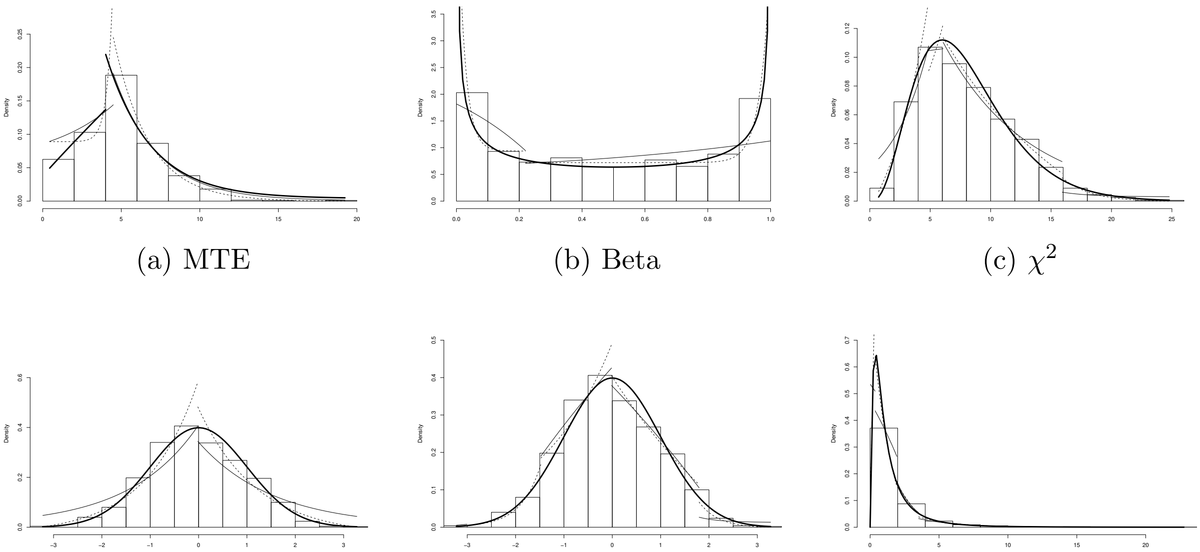
Table 1 shows the likelihood of the different samples for the models fitted using the direct ML approach, the modified LS method, and the original LS method described in (Rumí et al., 2006). The two sub-tables correspond to the split points found using the method described in (Rumí et al., 2006) and split points found by identifying the extreme points and the inflexion points of the of the true density, respectively.

From the results we clearly see that the ML-based method outperforms the LS method in terms of likelihood. This is hardly a surprise, as the ML method is actively using likelihood maximization as its target, whereas the LS methods do not. On the other hand, the LS and Original LS seem to be working at comparable levels. Most commonly (in 8 out of 12 runs), LS is an improvement over its original version, but the results for the Log-normal distribution (with the split-points selected according to the inflection points) cloud this picture; here the Original LS achieves a likelihood which is 10^{283} times as high as the one found by the LS method.

5 Conclusions and Future Work

In this paper we have introduced *maximum likelihood* learning for MTEs. Finding Maximum Likelihood parameters is interesting not only in its own right, but also as a vehicle to do more advanced learning: With maximum likelihood parameters we could, for instance, use the BIC criteria (Schwarz, 1978) to choose the number of exponential terms required to approximate the distribution function properly. We are currently in the process of evaluating this with the goal of avoiding overfitting during learning.

We are also considering to use a maximum likelihood-approach to learn *the location of the split-points*. Consider a sample $\mathbf{x} = \{x_1, \ldots, x_n\}$ where all samples are in the interval $\Omega_m = [\alpha, \beta)$, and assume the sample is sorted. A *brute force* approach to learning split-points could be to first fit MTE distributions on the intervals $[\alpha, (x_i + x_{i+1})/2)$ and $[(x_i + x_{i+1})/2, \beta)$, for each $i = 1, \ldots, n-1$, and calculate the likelihood of the data using the learned ML parameters. Then, one would choose the split-point, which gives the highest likelihood. Unfortunately, the complexity of this approach is squared in the sample size; we are currently investigating a number of simple heuristics to speed up the process. We have also started working on ML-based learning of *conditional distributions*, starting from the ideas published in (Moral et al., 2003). However, accurately locating the split-points for a conditional MTE is even more difficult than when learning marginals distributions; locating the split-



(a) MTE (b) Beta (c) χ^2

(d) Gaussian, 2 splits (e) Gaussian, 4 splits (f) Log-Normal

Figure 1: The plots show the results of samples from different distributions. The gold-standard distribution is drawn with a thick line, the MTE with Lagrange-parameters are given with the dashed line, and the results of the LS approach are given with the thin, solid line. The empiric distributions of each sample is shown using a histogram.

points for a variable X will not only influence the approximation of the distribution of X itself, but also the distributions for all the children of X.

Acknowledgments

This work has been supported by the Spanish Ministry of Education and Science, through project TIN2007-67418-C03-02.

References

D.P. Bertsekas. 1996. *Constrained optimization and Lagrange multiplier methods*. Academic Press Inc.

A.P. Dempster, N.M. Laird, and D.B. Rubin. 1977. Maximun likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1 – 38.

Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Laurence K. Saul. 1999. An introduction to variational methods for graphical models. *Machine Learning*, 37:183–233.

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

S. Moral, R. Rumí, and A. Salmerón. 2003. Approximating conditional MTE distributions by means of mixed trees. In *ECSQARU'03. Lecture Notes in Artificial Intelligence*, volume 2711, pages 173–183.

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

R. Rumí, A. Salmerón, and S. Moral. 2006. Estimating mixtures of truncated exponentials in hybrid Bayesian network. *Test*, 15:397–421.

Gideon Schwarz. 1978. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464.

Glenn R. Shafer and Prakash P. Shenoy. 1990. Probability Propagation. *Annals of Mathematics and Artificial Intelligence*, 2:327–352.

J.S. Simonoff. 1996. *Smoothing methods in Statistics*. Springer.