Estimating mixtures of truncated exponentials from data

Serafín Moral
Dpt. Computer Science
and Artificial Intelligence.
University of Granada
Granada, Spain

Rafael Rumí and Antonio Salmerón
Dept. of Statistics
and Applied Mathematics
University of Almeria
Almeria, Spain

Abstract

The MTE (mixture of truncated exponentials) model allows to deal with Bayesian networks containing discrete and continuous variables simultaneously. One of the features of this model is that standard propagation algorithms can be applied. In this paper, we study the problem of estimating these models from data. We propose an iterative algorithm based on least squares approximation. The performance of the algorithm is tested both with artificial and actual data.

Keywords: Mixtures of Truncated Exponentials, Least Squares Estimation, Hybrid Bayesian Networks, Learning

1 Introduction

MTE (Rumi et al., 2001) were introduced as a way of dealing with mixed networks (Bayesian networks containing discrete and continuous variables simultaneously). This problem was deeply studied before, but the only general solution is the discretisation of the continuous variables (Christofides et al., 1999; Dougherty et al., 1995; Koller and Kozlov, 1997) which are then treated as if they were discrete, and therefore the results obtained are not exact. Exact propagation can be carried out over mixed networks when the model is a conditional Gaussian distribution (Lauritzen, 1992; Olesen, 1993), but in this case, discrete variables are not allowed to have continuous parents. This restriction was overcome in Koller et al. (1999) using a mixture of exponentials to represent the distribution of discrete nodes with continuous parents, but the price to pay is that propagation cannot be carried out using exact algorithms: Monte Carlo methods are used instead.

MTE models provide the advantages of the traditional methods and the added feature that discrete variables with continuous parents are allowed. Exact standard propagation algorithm can be performed over them, as well as Monte Carlo algorithms, but its real power shows as an alternative to discretisation. Discretisation can be seen as approximating an arbitrary density by a mixture of uniforms; more accurate approximations can be obtained using exponentials instead of uniforms (the uniform distribution is a particular case of the MTE).

In this paper we propose an iterative algorithm, based on least squares approximation, to estimate MTE models from a database. The method described here is valid for univariate MTE distributions, but can be extended to estimate conditional distributions as well.

The paper is organised as follows: In section 2 we describe the MTE model, and the notation used throughout the paper. The estimation algorithm is presented in section 3, and examples of its performance are shown in section 4. Section 5 is devoted to the extension of the algorithm to estimate conditional distributions, and the paper ends with conclusions in section 6.

2 Preliminaries

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 set of possible values of a variable $\mathbf{X}$ is denoted by $\Omega_{\mathbf{X}}$. The MTE model is defined as follows:

Definition 1 (MTE density) *Let $\mathbf{X}$ be a mixed n-dimensional random variable. Let $\mathbf{Y} = (Y_1, \ldots, Y_d)$ 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 $f : \Omega_{\mathbf{X}} \mapsto I\!R_0^+$ is a* mixture of truncated exponentials density (MTE density) *if one of the next two conditions holds:*

i. *f can be written as*

$$f(\mathbf{x}) = f(\mathbf{y}, \mathbf{z}) = a_0 + \sum_{i=1}^{m} a_i \exp\left\{ \sum_{j=1}^{d} b_i^{(j)} y_j + \sum_{k=1}^{c} b_i^{(d+k)} z_k \right\} \quad (1)$$

for all $\mathbf{x} \in \Omega_{\mathbf{X}}$, where a_i, $i = 0, \ldots, m$ and $b_i^{(j)}$, $i = 1, \ldots, m$, $j = 1, \ldots, n$ are real numbers.

ii. *There is a partition $\Omega_1, \ldots, \Omega_k$ of $\Omega_{\mathbf{X}}$ verifying that the domain of the continuous variables, $\Omega_{\mathbf{Z}}$, is divided into hypercubes and such that f is defined as*

$$f(\mathbf{x}) = f_i(\mathbf{x}) \quad \text{if} \quad \mathbf{x} \in \Omega_i \ ,$$

where each f_i, $i = 1, \ldots, k$ can be written in the form of equation (1).

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

Three basic operations can be performed, *restriction*, *marginalisation* and *product*, over MTE potentials, and the result will be another MTE potential, this is, the class of MTE potentials is closed under these operations (Rumi et al., 2001), and therefore closed for Shenoy-Shafer propagation (Shenoy and Shafer, 1990).

3 Estimation Algorithm

In this section we will focus on estimating marginal (univariate) MTE distributions (corresponding to root nodes in a Bayesian network), as a previous step for defining conditional distributions associate with non-root nodes in a Bayesian Network.

The starting point of the algorithm is a density $f(x)$ for which we want to get the MTE density that best approximates it. Learning from data is a particular case of this, since the empirical histogram can be considered as target density.

As we can see in Def. 1, an MTE density can be defined in several parts and in each of these we can have an arbitrary number of exponential terms.

3.1 Splitting the domain

The way in which the domain is split is determined by the properties of the exponential function, which is the core of the MTE model. Since the exponential function $\exp\{x\}$ is concave and increases all over its domain, the partition must be such that in each one of the sub-intervals, the density $f(x)$ that we want to approximate (or the empirical histogram of the data) does not show changes from concavity to convexity or increase/decrease. In any case, an upper limit must be imposed to the number of sub-intervals, to avoid an excessive increase in the complexity of the learnt model.

3.2 Determining the number of exponential terms

The learning scheme we have designed allows to incorporate new exponential terms in each sub-intervals as long as the accuracy of the estimated model is increased. However, we have imposed an upper limit of two exponential terms (plus a constant) for each sub-interval.

The reason for using just two exponentials is the way in which the domain is split: If in each sub-interval there are not changes in terms of increase/decrease or concavity/convexity, usually two terms can accurately fit almost any curve. Of course the accuracy would increase using more terms, but the rate difficulty/improvement may not be worthy.

Therefore, on each interval we will estimate an independent term and no more than two exponentials, that is, the learnt MTE will be like this:

$$f^*(x) = K + a\exp\{bx\} + c\exp\{dx\} \ .$$

3.3 Fitting the model in each sub-interval

Let's see now how to estimate this MTE on a given interval D_j. We start off with two vectors, $\mathbf{x}$ and $\mathbf{y}$. In $\mathbf{x}$ we have a set of points within interval D_j, and in $\mathbf{y}$ the value of density f (or the empirical histogram) on each point of $\mathbf{x}$, i.e., $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (f(x_1), \ldots, f(x_n))$. We organise the information in this way to approach the problem just as if it were a problem of *exponential regression*. In an exponential regression we have a vector of points $(\mathbf{x}, \mathbf{y})$, and we try to fit to these points a function of the form

$$y = f^*(x) = a\exp\{bx\}$$

minimising the mean square error. To achieve this we take logarithms, so that

$$\ln\{y\} = \ln\{a\}\exp\{bx\} = \ln\{a\} + bx \ .$$

Therefore, we can write

$$y^* = a^* + bx \ ,$$

with $a^* = \ln\{a\}$ and $y^* = \ln\{y\}$, that is, just a linear regression whose solution is

$$(y^* - \bar{y^*}) = \frac{S_{xy^*}}{S_x^2}(x - \bar{x}) \ .$$

But we said before that our MTE has an independent term, which we calculate as follows. We have $y = f^*(x) = a\exp\{bx\} + c\exp\{dx\} + K$ and we want to get the $K \in \mathbb{R}$ that minimises the error function

$$E = \sum_{i=1}^{n} \frac{(y_i - f^*(x_i))^2}{n} \ .$$

Substituting $f^*(x)$ by its value we obtain

$$E = \sum_{i=1}^{n} \frac{(y_i - a\exp\{bx_i\} - c\exp\{dx_i\} - K)^2}{n} \ .$$

In order to find the minimum we derive

$$\frac{\partial E}{\partial K} = \sum_{i=1}^{n} \frac{-2(y_i - a\exp\{bx_i\} - c\exp\{dx_i\} - K)}{n} \ ,$$

and after solving the equation $\frac{\partial E}{\partial K} = 0$ we get:

$$K = \frac{1}{n}\sum_{i=1}^{n}(y_i - a\exp\{bx_i\} - c\exp\{dx_i\}) \ , \quad (2)$$

and to be sure that it is a minimum,

$$\frac{\partial^2 E}{\partial K^2}(K) = 2 > 0 \ .$$

We will follow an iterative algorithm to estimate the parameters of the MTE. First we learn one exponential, and afterwards the other, so that we will introduce the second exponential term only if the error decreases.

We need some initial values of a, b, and K (later we will consider how to initialise these parameters), and we take as initial values for c and d; $c = d = 0$, so we begin just with one exponential and the independent term.

Algorithm MTE-learn(a,b,K)

```
1. $c = d = 0$.
2. $\mathbf{w} = \mathbf{y} - a\exp\{b\mathbf{x}\} - K$.
3. Obtain $c$ and $d$ making an exponential regression $\mathbf{w} = c\exp\{d\mathbf{x}\}$.
4. Compute the mean squared error $E$. If it is lower than the last one computed keep $c$ and $d$ as calculated, otherwise leave them as they were before step 3.
5. $\mathbf{w} = \mathbf{y} - c\exp\{d\mathbf{x}\} - K$.
6. Obtain $a$ and $b$ making an exponential regression $\mathbf{w} = a\exp\{b\mathbf{x}\}$.
7. Compute the mean squared error $E$. If it is lower than the last one computed we keep $a$ and $b$ as calculated, otherwise leave them as they were before step 6.
```

8. Compute the independent term K, as shown in Eq. (2).
9. Compute the mean squared error E. If it is lower than the last one computed, keep K as calculated, otherwise leave it as it was before step 8.
10. Repeat from step 2 to step 9 until a fixed number of iterations is reached or until the parameters remain unchanged.

Some remarks must be done on this algorithm: When we have one exponential and estimate the parameters of the other, for instance, c and d, we get the exponential that best approximates this new pair $(\mathbf{x}, \mathbf{w})$, but we may be over-estimating the data, that is, maybe the best we can do is not to introduce this exponential, so, we calculate a coefficient, H, minimising the error that multiplies this exponential in order to obtain the actual influence of this term, that is, we make

$$\mathbf{y} = a \exp\{b\mathbf{x}\} + Hc \exp\{d\mathbf{x}\} + K \ .$$

The value of H that minimises the error is

$$H = \frac{\sum_{i=1}^{n}(y_i - a\exp\{bx_i - K\})(\exp\{dx_i\})}{\sum_{i=1}^{n} c \exp\{2dx_i\}} \ ,$$

and so we make $c = H * c$, and we do the same with a.

Also, in steps 2 and 5 we create a new vector $\mathbf{w}$. The vector $\mathbf{y}$ is always positive, but this new vector may be negative, so we need to transform the data before solving the exponential regression on steps 3 and 6. What we do is to add a constant to $\mathbf{w}$ in order to make every $w_i > 0$. After solving the regression we should undo the transformation, but we do not need to do so, since in step 8 we calculate the independent term that minimises the error.

The initial values for c and d are fixed to zero so that we do not introduce a new exponential term unless it causes a reduction in the error. The initial values of a, b and K can be anyone, but we suggest two different methods to calculate them:

Exponential regression, i.e., obtain a and b from $\mathbf{y} = a\exp\{b\mathbf{x}\}$ and K as explained in (2), or using a method that takes into account the derivative of the function. We want to get $f(x) \approx a\exp\{bx\} + K$, so the derivative of both functions should be the same: $f'(x) = ab\exp\{bx\}$. We do not work with the density, but with the pairs $(\mathbf{x}, \mathbf{y})$, so $f'(x)$ means the envelopes of the lines joining the points of $(\mathbf{x}, \mathbf{y})$. The method is as follows:

1. Construct a line from $(x_{(i-1)*m}, y_{(i-1)*m})$ to (x_{i*m}, y_{i*m}) for $i = 1, \ldots, n/m$.
2. Take as $x_i^* = \frac{x_{(i-1)*m} + x_{i*m}}{2}$ for $i = 1, \ldots n/m$.
3. Take as y_i^* the envelope of the between $(x_{(i-1)*m}, y_{(i-1)*m})$ and (x_{i*m}, y_{i*m}).
4. Solve an exponential regression $\mathbf{y}^* = a^* \exp\{b\mathbf{x}^*\}$ where $a^* = ab$, and obtain the initial values for a and b.
5. Obtain K as explained in (2).

This is the iterative algorithm we propose to estimate an MTE density from data. In fact, what we get doing this is not a density, but perhaps up to a normalisation factor.

4 Experiments

In order to test the ability of the MTE distribution to model common situations, and in order to check the accuracy of the algorithm, we have carried out experiments to fit some known distributions as well as real-world data.

4.1 Known distributions

We will see how we can represent three different distributions, very common in practice, with an MTE density.

4.1.1 Uniform distribution

As we said before, one of the advantages of using MTE is that is a generalisation of discretisation, in the sense that we can see a discretisation as a mixture of uniforms, so, if an MTE can represent a uniform distribution, it can represent a discretisation as well.

The uniform density is

$$f(x) = \begin{cases} \dfrac{1}{l_2 - l_1} & x \in (l_1, l_2) \ , \\ 0 & otherwise \ . \end{cases}$$

This distribution is exactly obtained when this method is applied. We do not have to split the domain, and in the first iteration we get the exact values for the parameters, that is, $a = b = c = d = 0$, and $K = \frac{1}{l_2 - l_1}$.

4.1.2 Exponential distribution

The exponential density is :

$$f(x) = \lambda \exp\{-\lambda x\} \qquad x > 0 \ .$$

The first thing we have to do before applying algorithm MTE-learn to this density is to define the domain our MTE will have, since it must be finite. We select an interval $(0, l)$ where l is such that $P(X > l) \approx 0$. Therefore we have only one interval in the domain, and the algorithm obtains the exact value for the parameters.

4.1.3 Normal distribution

We will focus on the standard normal distribution, $\mathcal{N}(0, 1)$, since we can get any $\mathcal{N}(\mu, \sigma)$ from this one. In order to apply the algorithm, we divide the domain of the normal density in four intervals, $(-4, -1), (-1, 0), (0, 1)$ and $(1, 4)$, on which the density does not have changes of concavity and increase/decrease. Applying the algorithm to these intervals we get:

Interval $(-4, -1)$:

$$\begin{aligned} a &= 2.67632782323753 \\ b &= 2.0995654834596547 \\ c &= -4.307200936472499\text{E} - 4 \\ d &= 0.0022590818669798686 \\ K &= 0.0013430276452163264 \end{aligned}$$

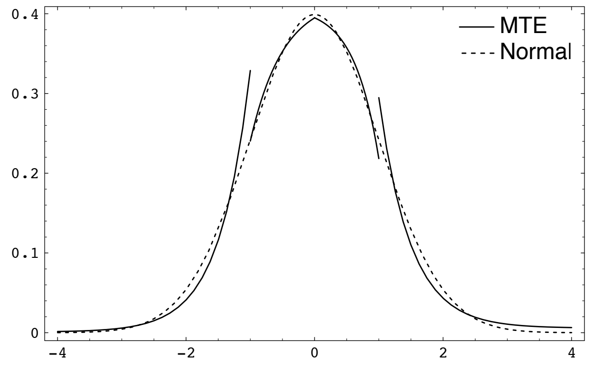

Figure 1: Comparison of the actual density and the learnt MTE density.

Interval $(-1, 0)$:

$$\begin{aligned} a &= -0.025332983750353668 \\ b &= -1.9593439331969362 \\ c &= 0 \\ d &= 0 \\ K &= 0.4201166819186963 \end{aligned}$$

Interval $(0, 1)$:

$$\begin{aligned} a &= -0.013883101566178647 \\ b &= 2.618433856043035 \\ c &= 0 \\ d &= 0 \\ K &= 0.4087927677026328 \end{aligned}$$

Interval $(1, 4)$:

$$\begin{aligned} a &= 2.2257931326098874 \\ b &= -2.04177256076009 \\ c &= -0.0018154157845694822 \\ d &= 0.004952177870949646 \\ K &= 0.007669343990692532 \end{aligned}$$

As we can see in Figure 1, the differences between the $N(0, 1)$ and this MTE density defined in four pieces are minimal. In order to prove that it is a good approximation, we simulated 100 values from the standard normal distribution and other 100 values from the learnt MTE distribution, and we performed a Kolmogorov-Smirnov test for two samples with the hypothesis that both samples come from the same distribution.

The test reported a p-Value of 0.6994 with two-sided alternative hypotheses, which supports the hypothesis that both samples come from the same distribution.

We next describe the method used to obtain the sample from the learnt MTE density.

Simulating a sample from an MTE density: If the coefficients of the exponentials are positive, i.e., $a > 0$, $c > 0$ and $K > 0$, in (Rumi et al., 2001) it was explained how to simulate values from an MTE density. If any of the coefficients is negative, we have to use another method, (Bignami and Matteis, 1971). To simulate a value from

$$f^*(x) = a \exp\{bx\} + c \exp\{dx\} + K \ ,$$

where a, c or K are negative, we transform $f^*(x)$ into a sum of densities

$$f^*(x) = w_1 f_1(x) + w_2 f_2(x) + w_3 f_3(x) \ ,$$

where some of the w_i are negative. Let's suppose it is w_2. We can write

$$f^*(x) = (w_1 + w_3)\left(\frac{w_1}{w_1+w_3} f_1(x) + \frac{w_3}{w_1+w_3} f_3(x)\right) + w_3 f_3(x) = (w_1+w_3) g(x) + w_3 f_3(x) \ .$$

To simulate a value we have to repeat these three steps until the acceptance condition in step 3 is met.

1. Simulate a value x_* from the density $g(x)$: This value will come from $f_1(x)$ with probability $\frac{w_1}{w_1+w_3}$ and from $f_3(x)$ with probability $\frac{w_3}{w_1+w_3}$.
2. Generate a random number, r.
3. If $r \leq \frac{f^*(x_*)}{w_1 f_1(x_*) + w_3 f_3(x_*)}$ accept x_* as a value from $f^*(x)$. Otherwise, repeat from step 1.

4.2 Real-world data

When learning a Bayesian Network from data, we do not have the explicit density of the continuous variables, but a sample of values of the variable. In this section we will see how algorithm MTE-learn can deal with this situation.

Let $(z_1, \ldots, z_n)$ be a sample from a continuous variable Z. We saw in Section 3 that the algorithm requires two vectors, $\mathbf{x} = (x_1, \ldots, x_m)$ and $\mathbf{y} = (y_1, \ldots, y_m)$, where $\mathbf{x}$ are values of the variable, and each y_i represents the value of the density of the variable in x_i. Therefore, before applying the algorithm, we must transform the data in order to obtain these two vectors to which apply it. We do it as follows:

1. Divide the domain in m sub-intervals,
$$\Omega_X = \bigcup_{i=1}^{n} I_i \ .$$
2. Compute the frequency for each interval I_j, n_j, $j = 1, \ldots, m$.
3. Take as z_i the midpoint of the interval I_i.
4. Take as $y_i = \frac{n_i/n}{length(I_i)}$

Doing this, we get two vectors $\mathbf{z}$ and $\mathbf{y}$ from $\mathbf{x}$, as required by the algorithm. We have used it to estimate a density for two continuous variables extracted from an agricultural survey performed about the greenhouses in the province of Almería (Spain): consumption and harvest. To test the accuracy of estimation, we used a χ^2 test instead of a Kolmogorov-Smirnov test since some values are repeated due to rounding the data when answering to the questions in the survey.

4.2.1 Consumption

We have a sample of 471 values of this variable, ranging from 36.1 to 1583.79. In order to obtain the empirical histogram, we transformed it choosing intervals of length 40. The fitted model is shown in Figure 2. We can see two parts very well differentiated, as increase/decrease and concavity/convexity is concerned, resulting in a split of the domain into

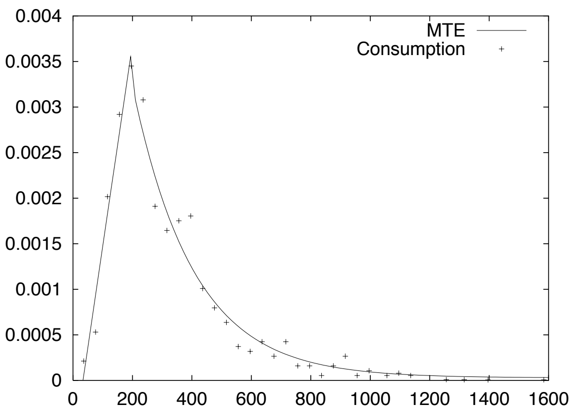

Figure 2: Approximation of the empirical histogram of variable Consumption by an MTE model

two intervals: $(36.1, 196.1)$ and $(196.1, 1583.79)$, and in each of them we estimate a MTE function:

Interval $(36.1, 196.1)$:

$$\begin{aligned} a &= -1.00481152209664 \\ b &= -2.2123865768185145\text{E}-5 \\ c &= -4.649576063595301\text{E}-16 \\ d &= -3.414123099380927\text{E}-10 \\ K &= 1.00406169448145 \end{aligned}$$

Interval $(196.1, 1583.79)$:

$$\begin{aligned} a &= 0.008503182179895076 \\ b &= -0.004880418989153087 \\ c &= 7.840998789880702\text{E}-8 \\ d &= -7.618752837976\text{91E}-8 \\ K &= 2.8252423901716575\text{E}-5 \end{aligned}$$

To prove that our MTE density is a good approximation of the actual density, we used a χ^2 test to determine whether the original sample can be said to come from the learnt MTE density or not. At a significance level $\alpha = 0.05$, the rejecting region is $(0, 0.484) \cup (11.20, \infty)$, and the value of the test statistic obtained was 3.172402699364092, so we can conclude that the test supports the hypothesis that the original sample comes from the learnt MTE density.

It is interesting to point out that in the first interval, the MTE density is almost a straight line. That tells us that we can also approximate this kind of functions with a MTE.

4.2.2 Harvest

Similarly to the previous one, from this variable we have again a sample of 471 values, ranging from 0.45 to 26. Again we have two intervals very well defined in which to split the domain, as we can see in Figure 3 :

Interval $(0.45, 4.45)$:

$$\begin{aligned} a &= 0.010826612517872319 \\ b &= 0.6255472893484579 \\ c &= -5.416676195266647\text{E}-4 \\ d &= 0.015687129036406677 \\ K &= 0.0053790556847500455 \end{aligned}$$

Interval $(4.45, 26)$:

$$\begin{aligned} a &= 0.23314178683678177 \\ b &= -0.18400083794081074 \\ c &= 0.14954712040375723 \\ d &= -0.22459486950530835 \\ K &= -0.0031378040872907897 \end{aligned}$$

Performing the same test as in the case of consumption, we get a value for the χ^2 statistic of 5.704724703434343, supporting the hypothesis that the original sample comes from the learnt MTE density.

5 Conditional Distributions

We have seen how algorithm MTE-learn works in the univariate case. However, in Bayesian Networks most of the probabilistic potentials that appear in the factorisation of the joint distribution are conditional distributions. Therefore, it is fundamental to solve the conditional case in order to make the MTE model really appropriate for Bayesian networks. In this section we roughly describe how it is possible to tackle the conditional case taking as basis the univariate case.

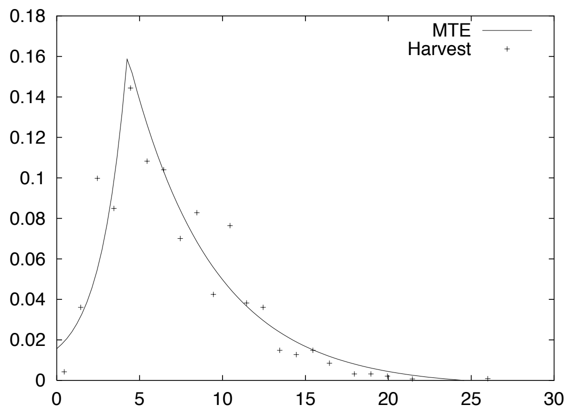

Figure 3: Approximation of the empirical histogram of variable Harvest by an MTE density

A way of defining a conditional distribution is

$$f(X|Y) = \frac{f(X,Y)}{f(Y)} \quad ,$$

but the MTE class is not closed under divisions, so we cannot face the problem in this way.

Another way to think about it just consider the conditional density as an MTE,

$$f(x|y) = a_0 + \sum_{i=1}^{m} a_i \exp\{b_i x + c_i y\} \quad ,$$

and then estimate the parameters in a similar way as in the univariate case. However, here the number of parameters is higher, which makes the task more difficult. Furthermore, restrictions about the values of the parameters should be made to guarantee that the function is actually a conditional density.

If all the parents of a continuous node are discrete, the problem is rahter easy: it is enough to estimate an MTE density for each configuration of the parents. If the parents are continuous, discretising them we could estimate a density for each configuration, and so the whole distribution would be a conditional distribution.

The question of how the continuous parents of a continuous node should be discretised is not trivial, and deserves a deep study. We think that a way of approaching the problem is to construct a *mixed probability tree* (Rumi et al., 2001) to represent each conditional distribution. A mixed probability tree is similar to a classification tree (Breiman et al., 1984), but instead of classes, each leaf contains an MTE density, that can be learnt using algorithm MTE-learn. Each internal node represents a parent node and the branches are splits of the domain.

6 Conclusions

We have presented in this paper a way to learn MTE densities from data, and we have checked its accuracy with both known distributions and real-world data. Finally we have outlined how we could learn a complete MTE network (including conditional distributions) from data.

In future works we plan to continue with the study of the MTE distributions, looking for a satisfactory discretization of the parents in a conditional distribution, so that we can define them properly, and trying to find out the conditions that must hold an MTE to be considered a conditional MTE.

Acknoledgments

This work has been supported by the Spanish Ministry of Science and Technology, through projects TIC2001-2973-C05-01,02 and by UAL-CAJAMAR, project E-729/01.

References

A. Bignami and A. De Matteis. 1971. A note on sampling from combinations of distributions. *J. Inst. Maths Applics*, 8:80–81.

L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. 1984. *Classification and regression trees.* Chapman & Hall/CRC.

A. Christofides, B. Tanyi, D. Whobrey, and N. Christofides. 1999. The optimal discretization of probability density functions. *Computational Statistics and Data Analysis*, 31:475 – 486.

J. Dougherty, R. Kohavi, and M. Sahami. 1995. Supervised and unsupervised discretization of continous features. In A. Prieditis y S. Russell, editor, *Machine Learning: Proceedings of the Twelfth International Conference*. Morgan Kaufmann, San Francisco.

D. Koller and A.V. Kozlov. 1997. Nonuniform dynamic discretization in hybrid networks. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, pages 314–325.

D. Koller, U. Lerner, and D. Anguelov. 1999. A general algorithm for approximate inference and its application to hybrid bayes nets. In K.B. Laskey and H. Prade, editors, *Proc. of the 15th Conference on Uncertainty in Artificial Intelligence*, pages 324–333. Morgan and Kauffman.

S.L. Lauritzen. 1992. Propagation of probabilities, means and variances in mixed graphical association models. *Journal of the American Statistical Association*, 87:1098–1108.

K.G. Olesen. 1993. Causal probabilistic networks with both discrete and continuous variables. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 15, pages 275–293.

R. Rumi, S. Moral, and A. Salmeron. 2001. Mixtures of truncated exponentials in hybrid bayesian networks. In Salem Benferhat and Philippe Besnard, editors, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, volume 2143 of *Lecture Notes in Artificial Intelligence*, pages 156–167. Springer.

P. P. Shenoy and G. Shafer. 1990. Axioms for probability and belief functions propagation. In R.D. Shachter, T.S. Levitt, J.F. Lemmer, and L.N. Kanal, editors, *Uncertainty in Artificial Intelligence, 4*, pages 169–198. North Holland, Amsterdam.