MPE inference in conditional linear gaussian networks

Salmerón, Antonio; Rumí, Rafael; Langseth, Helge; Madsen, Anders Læsø; Nielsen, Thomas Dyhre

Published in:
Symbolic and Quantitative Approaches to Reasoning with Uncertainty

DOI (link to publication from Publisher):
10.1007/978-3-319-20807-7_37

Publication date:
2015

Document Version
Early version, also known as pre-print

Citation for published version (APA):
Salmerón, A., Rumí, R., Langseth, H., Madsen, A. L., & Nielsen, T. D. (2015). MPE inference in conditional linear gaussian networks. In S. Destercke, & T. Denoeux (Eds.), *Symbolic and Quantitative Approaches to Reasoning with Uncertainty: 13th European Conference, ECSQARU 2015, Compiègne, France, July 15-17, 2015. Proceedings* (pp. 407-416). Springer. Lecture Notes in Computer Science, No. 9161
https://doi.org/10.1007/978-3-319-20807-7_37

MPE inference in Conditional Linear Gaussian Networks

Antonio Salmerón[1], Rafael Rumí[1], Helge Langseth[2], Anders L. Madsen[3,4], and Thomas D. Nielsen[4]

[1] University of Almería, ES-04120 Almería, Spain,
{rrumi,antonio.salmeron}@ual.es
[2] Norwegian University of Science and Technology, NO-7491 Trondheim, Norway,
helgel@idi.ntnu.no
[3] Hugin Expert A/S, DK-9000 Aalborg, Denmark,
anders@hugin.com
[4] Aalborg University, DK-9220 Aalborg, Denmark,
tdn@cs.aau.dk
http://www.amidst.eu

Abstract. Given evidence on a set of variables in a Bayesian network, the most probable explanation (MPE) is the problem of finding a configuration of the remaining variables with maximum posterior probability. This problem has previously been addressed for discrete Bayesian networks and can be solved using inference methods similar to those used for finding posterior probabilities. However, when dealing with hybrid Bayesian networks, such as conditional linear Gaussian (CLG) networks, the MPE problem has only received little attention. In this paper, we provide insights into the general problem of finding an MPE configuration in a CLG network. For solving this problem, we devise an algorithm based on bucket elimination and with the same computational complexity as that of calculating posterior marginals in a CLG network. We illustrate the workings of the algorithm using a detailed numerical example, and discuss possible extensions of the algorithm for handling the more general problem of finding a maximum a posteriori hypothesis (MAP).

Keywords: MPE inference, Conditional Linear Gaussian networks, hybrid Bayesian networks

1 Introduction

Probabilistic graphical models provide a well-founded and principled approach for performing inference in complex domains endowed with uncertainty. A probabilistic graphical model is a framework consisting of two parts: a qualitative component in the form of a graphical model encoding conditional independence assertions about the domain being modeled as well as a quantitative component consisting of a collection of local probability distributions adhering to the independence properties specified in the graphical model. Collectively, the two components provide a compact representation of the joint probability distribution over the domain being modeled.

Given a Bayesian network where a subset of the variables is observed, we may, e.g., query the network for the posterior marginal distributions of the remaining variables or for a maximum a posteriori probability configuration for a subset of the variables. If this subset is a proper subset of the non-observed variables, then the problem is referred to as a maximum a posteriori (MAP) hypothesis problem [10]. On the other hand, if the variables of interest correspond to the complement of the observation set, then the problem is referred to as that of finding the most probable explanation (MPE) [2, 6]; MPE can therefore be considered a specialization of MAP.

For Bayesian networks containing only discrete variables, there has been a substantial amount of work on devising both exact and approximate algorithms for performing MAP and MPE inference. However, for hybrid Bayesian networks, with both discrete and continuous variables, these types of inference problems have received only little attention [12]. In this paper we consider the problem of performing MPE inference in conditional linear Gaussian networks [7]. We propose an MPE algorithm based on bucket-elimination, which has the same computational complexity as that of standard inference for posterior marginals [8]. In contrast to the proposal in [12], we study the effect of entering evidence and also avoid the use of piece-wise defined functions by using an auxiliary tree structure keeping track of the functions used in previous calculations. The algorithm is illustrated using a detailed numerical example.

2 Preliminaries

Bayesian networks (BNs) [11, 1, 5] are a particular type of probabilistic graphical model that has enjoyed widespread attention in the last two decades. Attached to each node, there is a conditional probability distribution given its parents in the network, so that in general, for a BN with N variables $\mathbf{X} = \{X_1, \ldots, X_N\}$, the joint distribution factorizes as $p(\mathbf{X}) = \prod_{i=1}^{N} p(X_i|Pa(X_i))$, where $Pa(X_i)$ denotes the set of parents of X_i in the network. A BN is called *hybrid* if some of its variables are discrete while some others are continuous.

We will use lowercase letters to refer to values or configurations of values, so that x denotes a value of X and boldface $\mathbf{x}$ is a configuration of the variables in $\mathbf{X}$. Given a set of observed variables $\mathbf{X}_E \subset \mathbf{X}$ and a set of variables of interest $\mathbf{X}_I \subset \mathbf{X} \backslash \mathbf{X}_E$, *probabilistic inference* consists of computing the posterior distribution $p(x_i|\mathbf{x}_E)$ for each $i \in I$. If we denote by $\mathbf{X}_C$ and $\mathbf{X}_D$ the set of continuous and discrete variables not in $\{\mathbf{X}_i\} \cup \mathbf{X}_E$, and by $\mathbf{X}_{C_i}$ and $\mathbf{X}_{D_i}$ the set of continuous and discrete variables not in $\mathbf{X}_E$, the goal of inference can be formulated as computing

$$p(x_i|\mathbf{x}_E) = \left[\sum_{\mathbf{x}_D \in \Omega_{\mathbf{X}_D}} \int_{\mathbf{x}_C \in \Omega_{\mathbf{X}_C}} p(\mathbf{x}, \mathbf{x}_E) \mathrm{d}\mathbf{x}_C \right] \Big/ \left[\sum_{\mathbf{x}_{D_i} \in \Omega_{\mathbf{X}_{D_i}}} \int_{\mathbf{x}_{C_i} \in \Omega_{\mathbf{X}_{C_i}}} p(\mathbf{x}, \mathbf{x}_E) \mathrm{d}\mathbf{x}_{C_i} \right],$$

where $\Omega_{\mathbf{X}}$ is the set of possible values of a set of variables $\mathbf{X}$ and $p(\mathbf{x}, \mathbf{x}_E)$ is the joint distribution in the BN instantiated according to the observed values $\mathbf{x}_E$.

A particularly complex kind of inference in BNs is the so-called *maximum a posteriori (MAP)* problem. For a set of target variables $\mathbf{X}_I \subseteq \mathbf{X} \setminus \mathbf{X}_E$, the goal of MAP inference is to compute

$$\mathbf{x}_I^* = \arg \max_{\mathbf{x}_I \in \Omega_{\mathbf{X}_I}} p(\mathbf{x}_I | \mathbf{X}_E = \mathbf{x}_E), \tag{1}$$

where $p(\mathbf{x}_I | \mathbf{X}_E = \mathbf{x}_E)$ is obtained by first marginalizing out from the joint distribution $p(\mathbf{x})$ the variables not in $\mathbf{X}_I$ and not in $\mathbf{X}_E$. A related problem is *MPE* that stands for finding the *most probable explanation* to an observation $\mathbf{X}_E = \mathbf{x}_E$. It is a particular case of MAP, where $\mathbf{X}_I = \mathbf{X} \setminus \mathbf{X}_E$. Both MAP and MPE belong to the class of problems known as *abductive inference* [4].

2.1 Conditional Linear Gaussian Networks

A *Conditional Linear Gaussian Network* is a hybrid Bayesian network where the joint distribution is a conditional linear Gaussian (CLG) [7]. In the CLG model, the conditional distribution of each discrete variable $X_D \in \mathbf{X}$ given its parents is a multinomial, whilst the conditional distribution of each continuous variable $Z \in \mathbf{X}$ with discrete parents $\mathbf{X}_D \subseteq \mathbf{X}$ and continuous parents $\mathbf{X}_C \subseteq \mathbf{X}$, is given by

$$p(z | \mathbf{X}_D = \mathbf{x}_D, \mathbf{X}_C = \mathbf{x}_C) = \mathcal{N}(z; \alpha(\mathbf{x}_D) + \boldsymbol{\beta}(\mathbf{x}_D)^{\mathsf{T}} \mathbf{x}_C, \sigma(\mathbf{x}_D)) \,, \tag{2}$$

for all $\mathbf{x}_D \in \Omega_{\mathbf{X}_D}$ and $\mathbf{x}_C \in \Omega_{\mathbf{X}_C}$, where α and $\boldsymbol{\beta}$ are the coefficients of a linear regression model of Z given its continuous parents; this model can differ for each configuration of the discrete variables $\mathbf{X}_D$.

After fixing any configuration of the discrete variables, the joint distribution of any subset $\mathbf{X}_C \subseteq \mathbf{X}$ of continuous variables is a multivariate Gaussian. Hence, the parameters of the multivariate Gaussian can be obtained from the ones in the CLG representation. For a set of n continuous variables $Z_1, \ldots, Z_n$ with a conditionally specified joint density $p(z_1, \ldots, z_n) = \prod_{i=1}^n f(z_i | z_{i+1}, \ldots, z_n)$, where the k-th factor, $1 \leq k \leq n$, is such that

$$p(z_k | z_{k+1}, \ldots, z_n) = \mathcal{N}(z_k; \mu_{z_k | z_{k+1}, \ldots, z_n}, \sigma_{z_k}) \,,$$

it holds that the joint is $p(z_1, \ldots, z_n) = \mathcal{N}(z_1, \ldots, z_n; \boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ is the n-dimensional vector of means and $\boldsymbol{\Sigma}$ is the covariance matrix of the multivariate distribution over random variables $Z_1, \ldots, Z_n$ and both $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are derived from the parameters in Eq. (2) [9].

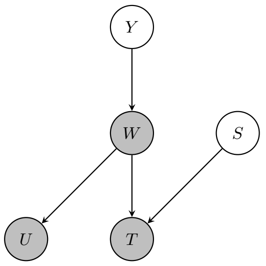
3 MPE Inference in CLG Networks

MPE inference can be carried out by adapting generic inference algorithms like *Bucket Elimination* [3]. The choice of bucket elimination as the underlying inference scheme for our proposal is motivated by its simplicity and flexibility, as well as the fact that it has been successfully employed in the MPE problem for discrete variables. The bucket elimination algorithm computes the MPE using local computations. A *bucket* containing probability functions is kept for each variable. Initially, an ordering of the variables in the network is established, and each conditional distribution in the network is assigned to the bucket corresponding to the variable in its domain holding the highest rank. Afterwards, the buckets are processed in a sequence opposite to the initial ordering of the variables. Each bucket is processed by combining all the functions it contains and by marginalizing the main variable in that bucket by maximization. The details of the algorithm are given in Alg. 1.

```
Function Elim-MPE($\mathbf{X}$,$P$,$\sigma$,$\mathbf{x}_E$)
Input: The set of variables in the network, $\mathbf{X} = \{X_1, \ldots, X_N\}$. The distributions in
       the network $P = \{p_1, \ldots, p_N\}$. An ordering, $\sigma$, of the variables in $\mathbf{X}$. Evidence
       $\mathbf{X}_E = \mathbf{x}_E$.
Output: $\mathbf{x}^{mpe}$, the configuration for which the posterior density reaches its maximum,
        and $mpe$, the density value at that point.
begin
    Initialization:
    Partition $P$ into buckets $B_1, \ldots, B_N$, where $B_i$ contains the conditional distributions in $P$ whose highest index variable is $X_i$.
    Backward phase:
    for $p \leftarrow N$ to 2 do
        if $X_p \in \mathbf{X}_E$ then
            Replace $X_p$ by $\mathbf{x}_{E_p}$ in each $h \in B_p$, and insert the resulting $h$ in the bucket
            corresponding to its highest ranked variable according to ordering $\sigma$.
        end
        else
            $h^p \leftarrow \max_{x_p} \prod_{h \in B_p} h$
            Insert $h_p$ in the bucket corresponding to its highest ranked variable.
        end
    end
    Forward phase:
    for $p \leftarrow 1$ to $n$ do
        Let $h^{R(x_1,\ldots,x_p)}$ denote the restriction of each function $h \in B_p$ to the values
        $(x_1, \ldots, x_p)$.
        $x_p^{mpe} \leftarrow \arg\max_{x_p} \prod_{h \in B_p} h^{R(x_1,\ldots,x_p)}$.
    end
    return $\mathbf{x}^{mpe} = \{x_1^{mpe}, \ldots, x_N^{mpe}\}$ and $mpe = \max_{x_1} \prod_{h \in B_1} h$ .
end
```

Algorithm 1: The Bucket elimination algorithm for computing the MPE as described in [3].

Example 1. Consider the network in Fig. 1 and the ordering $\langle Y, S, W, T, U \rangle$. According to such ordering, the initial setting of the buckets would be $B_Y = \{P(Y)\}$, $B_S = \{P(S)\}$, $B_W = \{f(w|Y)\}$, $B_T = \{f(t|w, S)\}$ and $B_U = \{f(u|w)\}$. The *backward* phase in Alg. 1 conveys the processing of the buckets as follows. The first bucket to be processed is B_U. It is done by maximizing out u from $f(u|w)$. As $f(u|w) = \mathcal{N}(u; w, 1)$, the maximum is reached at the mean, which means that U is maximized out by replacing u in $f(u|w)$ by w, which results in a function $h^U(w) = \frac{1}{\sqrt{2\pi}}$. Hence, the obtained function is in fact a constant, that is shifted to bucket B_W. The next bucket to handle is B_T, where T is removed from $f(t|w, S)$ by replacing t by the mean of the conditional distribution, resulting again in a constant function $h^T(w, S) = \frac{1}{\sqrt{2\pi}}$. After this calculation, h^T is stored in B_W, which is itself processed by multiplying $f(w|Y), h^T(w, S)$ and $h^U(w)$ and maximizing out W from the result. Since h^T and h^U are constant, we just have to maximize $f(x|Y)$ and multiply by the constants afterwards. The result is $h^W(Y, S) = (\frac{1}{\sqrt{2\pi}})^3$, that is stored in B_S. Bucket B_S contains $P(S)$ and $h^W(Y, S)$, whose product is equal to $0.1(\frac{1}{\sqrt{2\pi}})^3$ when $S = 0$ and $0.9(\frac{1}{\sqrt{2\pi}})^3$ when $S = 1$. Hence, maximizing with respect to S yields $h^S(Y) = 0.9(\frac{1}{\sqrt{2\pi}})^3$, that is sent to bucket B_Y. The MPE configuration is actually obtained in the *forward* phase of the algorithm, where the bucket processing step is traced back.



$P(Y) = (0.5, 0.5)$
$P(S) = (0.1, 0.9)$
$f(w|Y = 0) = \mathcal{N}(w; -1, 1)$
$f(w|Y = 1) = \mathcal{N}(w; 2, 1)$
$f(t|w, S = 0) = \mathcal{N}(t; -w, 1)$
$f(t|w, S = 1) = \mathcal{N}(t; w, 1)$
$f(u|w) = \mathcal{N}(u; w, 1)$

Fig. 1. A hybrid Bayesian network with two discrete and three continuous (shaded) variables.

The example above shows how maximizing out continuous variables is an easy task if the continuous variables are always removed first, as it just amounts to replacing the variable being removed by its mode (which in the Gaussian case is equal to its mean). The price to pay is that, in the worst case, a function containing all the discrete variables would be created, as is the case of $h^W(Y, S)$. It is an undesirable event, as the size of a probability function of discrete variables is exponential in the number of variables. This complexity blow-up can be avoided in many cases by allowing orderings for constructing the buckets where discrete and continuous variables can be arranged with no restrictions. But then a new problem arises, as the maximization operation becomes more complex. Assume, for instance, that we reach a point where Y is maximized out before W in Fig. 1. This amounts to computing

$$h^Y(w) = \max_y\{P(Y = y)f(w|Y = y)\} = \max\{0.5\mathcal{N}(w; -1, 1), 0.5\mathcal{N}(w; 2, 1)\}.$$

Therefore, h^Y is not a function with a single analytical expression, but it is piecewise defined instead. We show in the next section how it is possible to avoid piece-wise representations of the result of maximizing out discrete variables. Instead, we will keep lists of the functions that take place in the max operation.

In other words, the max operation is carried out in a lazy way. The counterpart is that the forward phase in Alg. 1 requires us to keep track of the operations carried out over the potentials in the backward phase. We propose to use a tree structure to keep track of the functions involved in intermediate calculations as illustrated in Fig. 2 and which corresponds to Example 1.

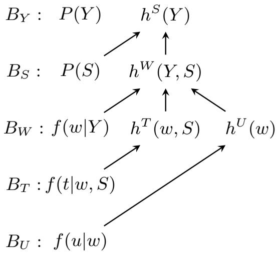


Fig. 2. Tree structure keeping track of the functions involved in the intermediate calculations performed during the backward phase of the bucket elimination algorithm.

3.1 Entering Evidence

If a variable is observed, no bucket is created for it. Instead, the variable is replaced by its observed value in every function where it appears. Assume a continuous variable X that is observed taking on value $X = x_0$. If the parents of X are $Y_1, \ldots, Y_n$, replacing variable X by value x_0 in its conditional density results in a function

$$\phi(y_1, \ldots, y_n) = \frac{1}{\sigma_x\sqrt{2\pi}} \exp\left\{-\frac{(x_0 - (\beta_0 + \sum_{i=1}^n \beta_i y_i))^2}{2\sigma_x^2}\right\}. \qquad (3)$$

Eventually, function ϕ will be passed to the bucket corresponding to one of its parents, where it will be multiplied by the parent's density prior to maximization. Let Y_j be such a parent of X. Its conditional density can be written as

$$f(y_j|Pa(Y_j)) = \frac{1}{\sigma_{y_j}\sqrt{2\pi}} \exp\left\{-\frac{(y_j - \mu_{y_j|pa(y_j)})^2}{2\sigma_{y_j}^2}\right\}. \qquad (4)$$

Maximizing the product of the functions in Eqs.(3) and (4) with respect to y_j is equivalent to maximizing the sum of their respective logarithms. It is obtained by solving the equation

$$\frac{\partial}{\partial y_j}\left(-\frac{(x_0 - (\beta_0 + \sum_{i=1}^n \beta_i y_i))^2}{2\sigma_x^2} - \frac{(y_j - \mu_{y_j|pa(y_j)})^2}{2\sigma_{y_j}^2}\right) = 0, \qquad (5)$$

which simply amounts to maximizing a quadratic function.

4 A numerical example

In this section we illustrate our proposal through a detailed example. Consider the CLG network illustrated in Fig. 1, where the discrete variables Y and S are assumed to be binary with states 0 and 1. Assume now that the continuous variable U is instantiated to 1 and we seek an MPE configuration over the remaining variables.

For performing MPE inference in this network we proceed with bucket elimination using the order $\langle W, T, S, Y\rangle$. Thus, the buckets are initialized as $B_Y = \{P(Y), f(w|Y)\}$, $B_S = \{P(S),\ f(t|w,S)\}$, $B_T = \{1\}$, $B_W = \{f(u=1|w)\}$, and $B_U = \{1\}$. The first bucket to be processed is B_Y, which involves maximizing Y from $P(Y)f(w|Y)$ and passing the result to bucket B_W.

$$h_1^Y(w) = \max_y P(y)f(w|y) = \max[P(Y=0)f(w|Y=0), P(Y=1)f(w|Y=1)],$$

where the super-script Y means that the potential contains two pieces indexed by Y; each of them corresponds to a scaled normal distribution (see Fig. 3). From an operational point of view, we use a list to store the components of $h_1^Y(w)$.

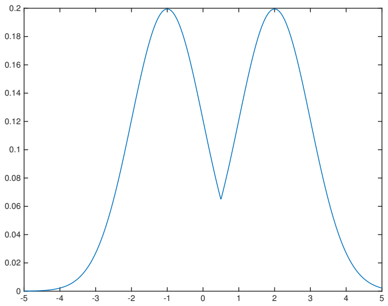


Fig. 3. The potential $h_1(w)$ obtained by maximizing Y out of $P(Y)f(w|Y)$.

The next bucket to process is B_S from which S should be eliminated. This operation produces the potential

$$\begin{aligned} h_2^S(t,w) = \max_s P(s)f(t|w,s) = \max[&P(S=0)f(t|w,S=0),\\ &P(S=1)f(t|w,S=1)], \end{aligned}$$

which is passed to B_T; again, the super-script S indicates that $h_2^S(t,w)$ is a list with as many elements as states of S. When processing B_T, we maximize out T:

$$h_3(w) = \max_t h_2(w,t) = \max_t \max_s P(s)f(t|w,s) = \max_s P(s) \max_t f(t|w,s),$$

which produces a potential containing a contribution for each state of S. By following the arguments from Example 1, $f(t|w, S=i)$ is maximized at the conditional means $-w$ (for $S=0$) and w (for $S=1$), thus

$$h_3(w) = (\sqrt{2\pi})^{-1} \max[P(S=0)\sigma^{-1}_{T,S=0}, P(S=1)\sigma^{-1}_{T,S=1}],$$

which is a scalar value and constant wrt. W; since $h_3(w)$ contains only one element we omit the super-script index previously used. Based on the CLG specification above, we find that $h_3(w) = (\sqrt{2\pi})^{-1} \max[0.1 \cdot 1, 0.9 \cdot 1] = 0.9(\sqrt{2\pi})^{-1}$, which is passed to B_W.

Finally, we eliminate W based on the potentials $B_W = \{h_1(w), h_3(w), f(U = 1|w)\}$, but since $h_3(w)$ is constant wrt. w we can disregard it during maximization (algorithmically, we can also detect this from the network structure using d-separation analysis):

$$\begin{aligned} h_4^Y &= \max_w [f(U=1|w)h_1(w)] \\ &= \max_w [f(U=1|w) \max[P(Y=0)f(w|Y=0), P(Y=1)f(w|Y=1)]] \\ &= \max[\max_w f(U=1|w)P(Y=0)f(w|Y=0), \\ &\qquad \max_w f(U=1|w)P(Y=1)f(w|Y=1)]]. \end{aligned}$$

The two maximizations over w can easily be solved analytically (see the discussion in Section 3.1), since $\log(f(U=1|w)P(Y=i)f(w|Y=i))$ is quadratic wrt. w, for $i=0,1$. That is, $\log(f(U=1|w)P(Y=i)f(w|Y=i))$ is maximized when

$$\frac{\partial}{\partial w}\left(-\frac{1}{2}(1-\beta_U w)^2 - \frac{1}{2}(w - \mu_{W,Y=i})^2\right) = 0,$$

which is achieved for $w^{\mathrm{mpe}}_{Y=i} = (\beta_U + \mu_{W,Y=i})/(\beta_U^2 + 1)$; here β_U is the regression coefficient for U wrt. w, $\mu_{W,Y=i}$ is the mean of W given $Y=i$ and the constant 1 in $(1-\beta_U w)^2$ corresponds to the observed value of U. Using the numerical specification above, we get $w^{\mathrm{mpe}}_{Y=0} = 0$ and $w^{\mathrm{mpe}}_{Y=1} = 1.5$.

In order to find a full MPE configuration over all the variable (and thereby also a single MPE value for W), we need to retrace the maximizing arguments for the variables on which the current potential depends (a tree structure like the one displayed in Fig. 2 can be used). This set of variables can be identified from the functional arguments for the potential in question together with the variables that index the list structure of this potential (given above by the super-script indexes). Specifically, for h_4^Y we see that the potential depends on Y only, hence we look for the value y^{mpe} of Y maximizing $P(Y)f(w^{\mathrm{mpe}}_Y|Y)$ (corresponding to $h_1^Y(w^{\mathrm{mpe}}_Y)$) and we get $y^{\mathrm{mpe}} = 1$ since $0.5 \cdot \mathcal{N}(1.5; 2, 1) > 0.5 \cdot \mathcal{N}(0; -1, 1)$. We thus also have $w^{\mathrm{mpe}} = 1.5$.

Next we proceed backwards in the elimination ordering and look for an MPE value for T. This is achieved by considering the maximizing arguments for h_3, which is the potential obtained when maximizing out T. From the discussion above we see that these maximizing arguments can immediately be identified as the conditional means of $f(t|w^{\mathrm{mpe}}, S=i))$ and we therefore find that $t^{\mathrm{mpe}}_{S=0} = -1.5$ and $t^{\mathrm{mpe}}_{S=1} = 1.5$. Lastly, we consider S and from the maximizing argument for $h_2^S(t, w)$ (obtained when maximizing out S) with t and w being fixed to their MPE values ($t_S^{\mathrm{mpe}} = 1.5$ and $w^{\mathrm{mpe}} = 1.5$), we get that $s^{\mathrm{mpe}} = 1$, since $0.1 \cdot \mathcal{N}(-1.5; 1.5, 1) < 0.5 \cdot \mathcal{N}(1.5; 1.5, 1)$, and thus $t^{\mathrm{mpe}} = 1.5$.

As a final comment, we would like to reemphasize that the MPE inference scheme as proposed in this paper, and illustrated above, follows the same structure as standard algorithms for performing, say marginal, inference in CLG networks. Thus, the algorithms share the same computational complexity. In particular, in the example above we see that the elimination order is able to exploit the conditional independencies in the model structure, and we therefore avoid the computational blow-up of having to consider all combinations of the discrete variables, cf. the discussion in Section 3. Furthermore, when identifying MPE configurations for the continuous variables we see that these configurations can easily be identified as either corresponding to the conditional means of the densities involved or they can be found by maximizing a quadratic function.

5 Conclusion and future work

In this paper we have discussed the MPE problem in conditional linear Gaussian networks. The behavior of the proposed algorithm was illustrated with the help of a small example model, successfully calculating the most probable explanation over the variables in the domain. The run-time complexity of the proposed algorithm is identical to that of standard probabilistic inference in CLG networks, and all maximization operations can be done efficiently using analytic solutions. The key contributor to the complexity is maintaining the list of Gaussian components representing the densities of the unobserved continuous variables.

Our next step is to extend our results to the maximum a posteriori (MAP) problem. This is significantly more difficult than the MPE problem, as we will have to do both summation and maximization operations over the discrete variables. Consider again the model in Fig. 1, and assume we are interested in the MAP configuration over Y and T. Eliminating S (by summation) will result in a *mixture* of Gaussians potential, while eliminating T (by maximization) results in a *maximum* of Gaussians potential; the two potentials should later be combined. Maintaining these two separate types of potentials is inconvenient, as they are not closed under the required operations, something that is highly unsatisfactory from a computational point of view.

We are currently investigating a technique to approximate the max-potentials using sum-potentials, see Fig. 4, which will enable us to do the calculations using a single data structure. We are looking into the quality of the generated approximations, and we are also working towards an implementation of the approximate inference technique. We are also studying strategies for selecting optimal variable orders for computing the buckets.

Acknowledgments. This work was performed as part of the AMIDST project. AMIDST has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 619209.

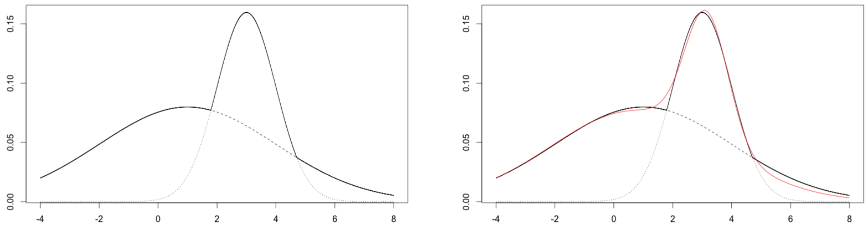


Fig. 4. Left part: Two Gaussian distributions (dashed lines) are shown together with their point-wise maximization (solid line). Right part: The max-potential is approximated by a mixture of Gaussians drawn using solid red line.

References

1. Robert G. Cowell, A.P̃hilip Dawid, Steffen L. Lauritzen, and David J. Spiegelhalter. *Probabilistic Networks and Expert Systems.* Statistics for engineering and information science. Springer, 1999. ISBN 0-387-98767-3.
2. A. Philip Dawid. Applications of a general propagation algorithm for a probabilistic expert system. *Statistics and Computing*, 2:25–36, 1992.
3. R. Dechter. Bucket elimination: a unifiying framework for reasoning. *Artificial Intelligence*, 113:41–85, 1999.
4. J.A. Gámez. Abductive inference in Bayesian networks: a review. In J.A. Gámez, S. Moral, and A. Salmerón, editors, *Advances in Bayesian Networks*, pages 101–117. Springer, 2004.
5. Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques.* MIT Press, 2009.
6. Johan Kwisthout. Most probable explanations in Bayesian networks: Complexity and tractability. *International Journal of Approximate Reasoning*, 52:1452–1469, 2011.
7. S.L. Lauritzen and N. Wermuth. Graphical models for associations between variables, some of which are qualitative and some quantitative. *The Annals of Statistics*, 17:31–57, 1989.
8. Uri Lerner and Ronald Parr. Inference in hybrid networks: Theoretical limits and practical algorithms. In *In UAI*, pages 310–318, 2001.
9. J.D. Nielsen, J.A. Gámez, and A. Salmerón. Modelling and inference with Conditional Gaussian probabilistic decision graphs. *International Journal of Approximate Reasoning*, 53:929–945, 2012.
10. James D. Park. Map complexity results and approximation methods. In Adnan Darwiche and Nir Friedman, editors, *UAI*, pages 388–396. Morgan Kaufmann, 2002.
11. Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers Inc., San Mateo, CA., 1988.
12. W. Sun and K.C. Chang. Study of the most probable explanation in hybrid Bayesian networks. In *Signal Processing, Sensor Fusion, and Target Recognition XX. Proc. of SPIE*, volume 8050, 2011.