Scalable MAP inference in Bayesian networks based on a Map-Reduce approach

Darío Ramos-López DRAMOSLOPEZ@UAL.ES
Antonio Salmerón ANTONIO.SALMERON@UAL.ES
Rafel Rumí RRUMI@UAL.ES
Department of Mathematics
University of Almería
Almería (Spain)

Ana M. Martínez ANA@CS.AAU.DK
Thomas D. Nielsen TDN@CS.AAU.DK
Department of Computer Science
Aalborg University
Aalborg (Denmark)

Andrés R. Masegosa ANDRES.MASEGOSA@IDI.NTNU.NO
Helge Langseth HELGEL@IDI.NTNU.NO
Department of Computer and Information Science
Norwegian University of Science and Technology
Trondheim (Norway)

Anders L. Madsen ALM@HUGIN.COM
Hugin Expert A/S and Department of Computer Science
Aalborg University
Aalborg (Denmark)

Abstract

Maximum a posteriori (MAP) inference is a particularly complex type of probabilistic inference in Bayesian networks. It consists of finding the most probable configuration of a set of variables of interest given observations on a collection of other variables. In this paper we study scalable solutions to the MAP problem in hybrid Bayesian networks parameterized using conditional linear Gaussian distributions. We propose scalable solutions based on hill climbing and simulated annealing, built on the Apache Flink framework for big data processing. We analyze the scalability of the solution through a series of experiments on large synthetic networks.

Keywords: MAP inference; hybrid Bayesian networks; scalable algorithms; Apache Flink.

1. Introduction

Today, omnipresent sensors are continuously providing streaming data on the environments in which they operate. For instance, a typical monitoring and analysis system may use streaming data generated by sensors to monitor the status of a particular device and to make predictions about its future behaviour, or diagnostically infer the most likely system configuration that has produced the observed data. Sources of streaming data with even a modest updating frequency can produce extremely large volumes of data, thereby making efficient and accurate data analysis and prediction difficult. One of the main challenges is related to handling uncertainty in the data, where principled methods and algorithms for dealing with uncertainty in massive data applications are required.

Probabilistic graphical models (PGMs) provide a well-founded and principled approach for performing inference and belief updating in complex domains endowed with uncertainty. In particular, Bayesian networks (BNs) (Jensen and Nielsen, 2007; Kjærulff and Madsen, 2013; Pearl, 1988) have enjoyed widespread attention in the last decades.

In this paper we focus on scenarios where data come in streams at high speed, and where a quick response is required. Furthermore, we assume data to consist of observations of both discrete and continuous variables, and that we are interested in querying a Bayesian network in order to obtain the most probable configuration of specific *variables of interest* given observations of particular *evidence variables*. As an example consider the problem of situation awareness while piloting a plane. The system would constantly be receiving input in the form of data streams consisting of discrete variables (flaps/ slats on or off, landing gear up or down, etc) as well as continuous variables (ground speed, altitude, outside temperature, engine power, etc). For each observation in the stream, the system would have to determine the most likely status of the pilot's situation awareness, defined in terms of variables such as the current flight situation (landing, take off, cruise, ...) and the degree of the pilot's awareness about the current situation (ranging from 0 to 1).

The process of answering such queries is called *MAP (maximum a posteriori) inference* and is of special relevance and difficulty (Park, 2002; de Campos et al., 2001; Fu et al., 2013). MAP inference is known to be NP-hard even for simple network structures like poly-trees and naïve Bayes networks (de Campos, 2011; Kwisthout, 2011). This makes approximate algorithms even more necessary for handling practical situations, and current state-of-the-art algorithms range from algorithms based on simplifying the problem structure and refining the solution afterwards (Choi and Darwiche, 2009) to search-based algorithms guided by heuristic functions (Yuan and Hansen, 2009; Marinescu et al., 2014, 2015). Common for these approaches is that they are restricted to domains where the variables are discrete, and they are therefore not applicable for scenarios like the piloting example above.

The only specific works related to MAP inference in hybrid BNs we have found in the literature are those of Sun and Chang (2011) and Weiss and Freeman (2010). However, they focus on problems where the maximum is computed over all non-observed variables, which is a special type of the MAP problem called the *most probable explanation (MPE)*. Furthermore, convergence to the global solution is only guaranteed in the pure Gaussian case, replying on the unimodality of the Gaussian density; the method may only reach a local optimum in multimodal settings. Also restricted to MPE is the method proposed by Salmerón et al. (2015), where a lazy approach is taken in order to keep inference complexity comparable with regular marginal inference.

In this paper, we propose scalable approaches to the MAP problem in hybrid Bayesian networks. The proposals are based on hill climbing and simulated annealing and have been implemented on top of Apache Flink (Alexandrov et al., 2014), a big data processing architecture compatible with Map/Reduce schemes (Dean and Ghemawat, 2008). The remainder of the paper is organized as follows. Basic definitions and notation is given in Section 2. Our proposal is described in Section 3 and is experimentally analyzed in Section 4. The paper ends with conclusions in Section 5.

2. Preliminaries

A BN consists of a directed acyclic graph, where the nodes correspond to random variables and the arcs represent dependencies among them. Attached to each node in the graph, there is a conditional probability distribution for the corresponding variable given its parents. By the chain rule, a BN with N variables $\mathbf{X} = \{X_1, \ldots, X_N\}$ defines a joint probability distribution that factorizes as

$p(\mathbf{X}) = \prod_{i=1}^{N} p(X_i|Pa(X_i))$, where $Pa(X_i)$ are the parents of X_i in the graph. A BN is called *hybrid* if it has both discrete and continuous variables. We will use lowercase letters to refer to values or configurations of values, so that x corresponds to a value of the variable X and boldface $\mathbf{x}$ refers to a configuration of the variables in $\mathbf{X}$. The set of all possible configurations of a random vector $\mathbf{X}$ is denoted by $\Omega_{\mathbf{X}}$.

In a *Conditional Linear Gaussian (CLG) Network* (Lauritzen and Wermuth, 1989), the conditional distribution of each discrete variable $X_D \in \mathbf{X}$ given its parents is a multinomial distribution, 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 as a normal density by

$$p(z|\mathbf{X}_D = \mathbf{x}_D, \mathbf{X}_C = \mathbf{x}_C) = \mathcal{N}(z; \alpha_{\mathbf{x}_D} + \boldsymbol{\beta}_{\mathbf{x}_D}^{\mathsf{T}} \mathbf{x}_C, \sigma_{\mathbf{x}_D}), \tag{1}$$

for all $\mathbf{x}_D \in \Omega_{\mathbf{X}_D}$ and $\mathbf{x}_C \in \Omega_{\mathbf{X}_C}$; $\alpha_{\mathbf{x}_D}$ and $\boldsymbol{\beta}_{\mathbf{x}_D}$ are the coefficients of a linear regression model of Z (there is one such set of coefficients for each configuration of the discrete parent variables). Thus, the conditional mean of Z is a linear function of its continuous parents, while its standard deviation, σ_D, only depends on the discrete parents. Finally it should be noted that a CLG network does not allow continuous variables having discrete children.

2.1 The MAP inference problem

Generally, for the *maximum a posteriori* (MAP) inference problem we look for a configuration of a particular subset of variables $\mathbf{X}_I$ having maximum posterior probability given observation $\mathbf{x}_E$ of another set of variables $\mathbf{X}_E$: $\mathbf{x}_I^* = \arg\max_{\mathbf{x}_I \in \Omega_{\mathbf{X}_I}} p(\mathbf{x}_I|\mathbf{x}_E) = \arg\max_{\mathbf{x}_I \in \Omega_{\mathbf{X}_I}} p(\mathbf{x}_I, \mathbf{x}_E)$. If $\mathbf{X}_E \cup \mathbf{X}_I \subset \mathbf{X}$, the variables in the set $\mathbf{X} \setminus (\mathbf{X}_E \cup \mathbf{X}_I)$ should be marginalized out before doing the maximization:

$$\mathbf{x}_I^* = \arg \max_{\mathbf{x}_I \in \Omega_{\mathbf{X}_I}} \sum_{\mathbf{x}_D \in \Omega_{\mathbf{X}_D \setminus (\mathbf{X}_I \cup \mathbf{X}_E)}} \int_{\mathbf{x}_C \in \Omega_{\mathbf{X}_C \setminus (\mathbf{X}_I \cup \mathbf{X}_E)}} p(\mathbf{x}, \mathbf{x}_E) \mathrm{d}\mathbf{x}_C. \tag{2}$$

In the special case where $\mathbf{X} = \mathbf{X}_E \cup \mathbf{X}_I$, the inference problem is also referred to as finding the *most probable explanation* (MPE) (Gámez, 2004).

Finding an exact solution to the general MAP problem has been shown to be $\mathrm{NP}^{\mathrm{PP}}$ complete (Park, 2002). Furthermore, if N equals the number of variables in the model, approximating MAP with an error-bound growing slower than exponentially in N is NP-hard (Park, 2002; Roth, 1996). The complexity of the problem motivates the study of approximate algorithms, where the term "approximate" here refers to the fact that the given solutions are not guaranteed to be optimal.

3. Scalable approximate MAP in CLG networks

The MAP problem is defined as the optimization problem in Equation (2), where the optimization is carried out over $p(\mathbf{x}_I, \mathbf{x}_E)$. This distribution is obtained from $p(\mathbf{x})$ after marginalizing out all the variables that are not included in $\mathbf{X}_I \cup \mathbf{X}_E$. We will develop our proposal for doing MAP inference in CLG networks by first analyzing the problem for discrete models, and then extend these to CLG models. Afterwards we will consider how to parallelize the calculations required for the MAP inference.

3.1 MAP in discrete Bayesian network models

Assume first that the model only contains *discrete* variables. In this case, the MAP problem can be approximately solved using well known optimization techniques like *hill climbing* (Park and Darwiche, 2001) or *simulated annealing* (de Campos et al., 2001). The idea underlying both schemes is to explore the space of possible solutions to the MAP problem (that is, configurations over $\mathbf{X}_I$) by iteratively moving from a configuration to one of its "neighbours", hopefully following a path leading towards the most probable configuration. There are usually two ways of moving from one configuration of the discrete variables to the next, namely by selecting an entirely new configuration (potentially changing the value of all the involved variables) or by only partially modifying the current configuration. These approaches are known as *global* and *local search*, respectively (Park and Darwiche, 2001).

The details of the *hill climbing* method are given in Algorithm 1. It starts from an initial configuration over the variables of interest, $\mathbf{X}_I = \mathbf{x}_I$, and iteratively moves towards the optimum, making sure never to move to a configuration worse than the current one while exploring the search space. The exploration is determined by the call to **GenerateConfiguration** in Line 2, where the movement in the search-space is defined; we shall return to this procedure in Section 3.2. The internals of this procedure are outlined in Algorithm 3. It requires a pre-defined *stopping criterion*; some popular choices are to establish a maximum number of iterations, a maximum number of consecutive non-improving iterations, or a target probability threshold. In practice, we found that setting the maximum number of iterations to a relatively small number produces good results. Also this algorithm is initiated by a guess about the configuration over the variables of interest.

The algorithm requires that the probability $p(\mathbf{x}_I, \mathbf{x}_E)$ is calculated in Line 3. It can be computed using Equation (2), but that approach would require handling the joint distribution $p(\mathbf{x}, \mathbf{x}_E)$, which is often computationally intractable. We therefore propose to approximate it using *importance sampling*. Let us denote by $\mathbf{X}^*$ variables that are not observed and are not part of the variables of interest, i.e. $\mathbf{X}^* = \mathbf{X} \setminus (\mathbf{X}_I \cup \mathbf{X}_E)$, and let f^* denote a density function called the *sampling distribution*. f^* can be selected arbitrarily, as long as it allocates strictly positive probability mass to any configuration of $\mathbf{X}^*$ where p has support, $f^*(\mathbf{x}) > 0 \ \forall \mathbf{x} \in \Omega_{\mathbf{X}^*}$. We follow the evidence weighting scheme of Salmerón et al. (2015) and choose f^* to be the product of the conditional distributions in the network instantiated to $\mathbf{x}_I$ and $\mathbf{x}_E$. Now, we can write $p(\mathbf{x}_I, \mathbf{x}_E)$ as:

$$\begin{aligned} p(\mathbf{x}_I, \mathbf{x}_E) &= \sum_{\mathbf{x}^* \in \Omega_{\mathbf{X}^*}} p(\mathbf{x}_I, \mathbf{x}_E, \mathbf{x}^*) = \sum_{\mathbf{x}^* \in \Omega_{\mathbf{X}^*}} \frac{p(\mathbf{x}_I, \mathbf{x}_E, \mathbf{x}^*)}{f^*(\mathbf{x}^*)} f^*(\mathbf{x}^*) \\ &= \mathbb{E}_{f^*}\left[\frac{p(\mathbf{x}_I, \mathbf{x}_E, \mathbf{x}^*)}{f^*(\mathbf{x}^*)}\right] \approx \frac{1}{n} \sum_{i=1}^{n} \frac{p\left(\mathbf{x}_I, \mathbf{x}_E, \mathbf{x}^{*(i)}\right)}{f^*\left(\mathbf{x}^{*(i)}\right)}, \end{aligned} \tag{3}$$

where $\left\{\mathbf{x}^{*(1)}, \ldots, \mathbf{x}^{*(n)}\right\}$ is a sample of size n drawn from f^*.

Algorithm 2 details the *simulated annealing* algorithm for obtaining a MAP estimate. This technique is a well known optimization procedure designed to avoid local maxima during the search process. This aspect of the search is controlled by the *temperature* parameter T, initialized in Line 2. The algorithm uses the standard stopping criterion of terminating when the system is sufficiently *cold* ($T < \varepsilon$). The three free parameters T, α, and ϵ must be set before the algorithm can run. The values in Line 2 are the ones used in the experiments reported in this paper. Similar to Algorithm 1, this algorithm is also initiated by an initial configuration over $\mathbf{X}_I$. We retain the new configuration

```
HC_MAP($B$,$\mathbf{x}_I$,$\mathbf{x}_E$,$r$)
Input: A BN $B$ defining a joint distribution $p(\mathbf{x})$ over variables $\mathbf{X}$, evidence $\mathbf{X}_E = \mathbf{x}_E$
       (optional), an initial configuration of the variables of interest $\mathbf{X}_I = \mathbf{x}_I$, and the
       number $r$ of variables whose value will be changed to generate a new configuration.
Result: An approximate MAP estimate.
1  while stopping criterion not satisfied do
2      $\mathbf{x}_I^* \leftarrow$ GenerateConfiguration($B$, $\mathbf{x}_I$, $\mathbf{x}_E$, $r$)
3      if $p(\mathbf{x}_I^*, \mathbf{x}_E) \geq p(\mathbf{x}_I, \mathbf{x}_E)$ then
4          /* Improvement, so accept the new solution */
5          $\mathbf{x}_I \leftarrow \mathbf{x}_I^*$
6      end
7  end
8  return $\mathbf{x}_I$
```

Algorithm 1: *Hill climbing* algorithm for approximate MAP.

$\mathbf{x}_I^*$ if it is better than the currently chosen one or if we randomly choose to do so (Line 7). Note how the temperature T guides the search, from an almost completely random behaviour when $T \gg 1$ to almost completely greedy when $T \ll 1$. Again, we require the calculation of $p(\mathbf{x}_I, \mathbf{x}_E)$ (used in Line 7 of Algorithm 2), and again we use importance sampling to approximately calculate this value (see Equation (3)).

```
SA_MAP($B$,$\mathbf{x}_I$,$\mathbf{x}_E$,$r$)
Input: A BN $B$ with variables $\mathbf{X}$ and joint distribution $p(\mathbf{x})$, evidence $\mathbf{X}_E = \mathbf{x}_E$ (optional),
       an initial configuration of the variables of interest $\mathbf{X}_I = \mathbf{x}_I$, and the number $r$ of
       variables whose value will be changed to generate a new configuration.
Result: An approximate MAP estimate.
1  /* Default values of the temperature parameters */
2  $T \leftarrow 1\,000$; $\alpha \leftarrow 0.90$; $\varepsilon > 0$
3  while $T \geq \varepsilon$ do
4      $\mathbf{x}_I^* \leftarrow$ GenerateConfiguration($B$, $\mathbf{x}_I$, $\mathbf{x}_E$, $r$)
5      /* Accept $\mathbf{x}_I^*$ if its prob. increases or decreases $< T \cdot \ln(1/\tau)$
           */
6      Simulate a random number $\tau \sim \mathcal{U}(0,1)$
7      if $p(\mathbf{x}_I^*, \mathbf{x}_E) > p(\mathbf{x}_I, \mathbf{x}_E)/(T \cdot \ln(1/\tau))$ then
8          $\mathbf{x}_I \leftarrow \mathbf{x}_I^*$
9      end
10     /* Cool down the temperature */
11     $T \leftarrow \alpha \cdot T$
12 end
13 return $\mathbf{x}_I$
```

Algorithm 2: A *simulated annealing* algorithm for finding a MAP estimate.

3.2 Hybrid models

In the case of hybrid models, we need to extend the above methods to also accommodate continuous variables. Notice how Algorithms 1 and 2 above can both be used for CLG models as long as $i)$ we can approximately calculate $p(\mathbf{x}_I^*, \mathbf{x}_E)$, and $ii)$ we define the procedure **GenerateConfiguration** used by both algorithms to work with hybrid domains. The first point is not problematic, as the importance sampling scheme of Equation (3) can be immediately extended to hybrid domains. The definition of the search operator **GenerateConfiguration** is what we will discuss next.

The first step is to recognize an important feature of the MPE configuration of CLG networks. Recall that discrete variables can never have continuous parents in a CLG network. Therefore, we can define a topological ordering of the nodes in the DAG so that the discrete variables always appear before the continuous ones in that order. As a consequence, if there is no evidence ($\mathbf{X}_E = \emptyset$) it can easily be shown that the most probable configuration[1] of the set of continuous variables $\mathbf{X}_C$ can be calculated for a given configuration of all the discrete variables $\mathbf{X}_D = \mathbf{x}_D$. For any complete configuration of $\mathbf{X}_D$, the continuous variables represent a single multivariate Gaussian distribution, and the MPE configuration is found by visiting the continuous variables in topological order and selecting the mean of each variable conditioned on the values chosen for the parents, see also Equation (1). Furthermore, since the multivariate Gaussian distribution marginalized onto a subset of its variables is obtained by simply dropping irrelevant variables, it follows that this procedure also works for obtaining a MAP configuration over the continuous variables given a full configuration over $\mathbf{X}_D$.

Furthermore, this property also holds whenever the parents of any observed continuous variable are either discrete or observed, that is, when

$$Pa(\mathbf{X}_C \cap \mathbf{X}_E) \subseteq \mathbf{X}_D \cup \mathbf{X}_E. \tag{4}$$

However, if an unobserved variable Z_1 is the parent of an observed variable Z_2 the procedure cannot be applied; when allocating the value of Z_1 we are not taking the value of Z_2 into account. In those situations, the topological order of the continuous variables can be modified by using the so-called *arc reversal* operation (Madsen, 2008; Cinicioglu and Shenoy, 2009), resulting in a new DAG where the observed continuous variables ($\mathbf{X}_C \cap \mathbf{X}_E$) appear before the unobserved ones ($\mathbf{X}_C \setminus \mathbf{X}_E$) in the order. Unfortunately, this procedure can be computationally demanding in practice, and we propose a more efficient approximate technique. Our particular proposal within the context of CLG networks is to restrict the search space taking advantage of the property of CLG networks commented above. More precisely, we propose to randomly generate new values for (at most) the discrete variables and the continuous variables not satisfying Equation (4). The values of the remaining continuous variables are then chosen in order to maximize the density of the resulting configuration. Note that we define a *full* configuration over all the discrete variables, yet we only return the variables of interest. The details are given in Algorithm 3, where the parameter r determines if we do global or local search.

1. Note that we here slightly abuse terminology to ease the following presentation. Since all configurations of the variables in a hybrid model have probability zero, we here use the term "most probable configuration" to denote the configuration with the largest probability *density*.

```
GenerateConfiguration($B$, $\mathbf{x}_I$, $\mathbf{x}_E$, $r$)
Input: A BN $B$ with $N$ variables $\mathbf{X} = \mathbf{X}_D \cup \mathbf{X}_C$ and joint distribution $p(\mathbf{x})$, evidence
       $\mathbf{X}_E = \mathbf{x}_E$ (optional), an initial configuration of the variables of interest $\mathbf{X}_I = \mathbf{x}_I$,
       and the number $r$ of discrete variables whose value will be changed to generate a new
       configuration
Result: A new configuration of the variables in $\mathbf{X}_I$.
1  Let $\mathbf{x}^*$ be the part of configuration $\mathbf{x}_I$ corresponding to discrete variables
2  for $j \leftarrow 1$ to $r$ do
3      // Change the value of $r$ different discrete variables
4      Randomly draw a discrete variable $X$ from $(\mathbf{X}_D \cap \mathbf{X}_I) \setminus \mathbf{X}_E$ without replacement
5      Select a valid state $x$ of variable $X$ randomly and uniformly
6      Change the value of $X$ to $x$ in $\mathbf{x}^*$
7  end
8  Append $\mathbf{x}_E$ to $\mathbf{x}^*$
9  foreach $X \in \mathbf{X}_D \setminus (\mathbf{X}_I \cup \mathbf{X}_E)$ in topological order do
10     Simulate a value $x$ for $X$ using its conditional distribution instantiated to $\mathbf{x}^*$
11     Append $x$ to $\mathbf{x}^*$
12 end
13 foreach $X \in \mathbf{X}_C \setminus \mathbf{X}_E : Pa(X) \nsubseteq \mathbf{X}_D \cup \mathbf{X}_E$ in topological order do
14     // Handle continuous variables with 'difficult' evidence
           patterns
15     Simulate a value $x$ for $X$ from its conditional distribution instantiated to $\mathbf{x}^*$
16     Append $x$ to $\mathbf{x}^*$
17 end
18 foreach $X \in \mathbf{X}_C \setminus \mathbf{X}_E : Pa(X) \subseteq \mathbf{X}_D \cup \mathbf{X}_E$ in topological order do
19     // Change the value of the 'simple' continuous variables
20     Let $x = \mu_{X|\mathbf{x}^*}$, the mean of $X$ conditional on $\mathbf{x}^*$
21     Append $x$ to $\mathbf{x}^*$
22 end
23 Remove from $\mathbf{x}^*$ all the values corresponding to variables not in $\mathbf{X}_I$
24 return $\mathbf{x}^*$
```

Algorithm 3: Procedure to move from one configuration of variables in a CLG BN to another one.

3.3 Scalable implementation

The algorithms described above have been implemented as a part of of the AMIDST toolbox (Masegosa et al., 2015, 2016), which is a Java library for scalable probabilistic machine learning [2]. The implementation has been developed on top of Apache Flink (Alexandrov et al., 2014), a framework for big data processing.

Since hill climbing and simulated annealing are iterative algorithms, a simple manner of parallelizing the execution is as follows. First, draw a certain number of samples from the BN according to the probability distribution of the variables in the network given the evidence. This can

2. The toolbox can be downloaded from `http://amidst.github.io/toolbox/`.

be achieved using forward sampling, i.e., simulating from root to leaves (Salmerón et al., 2015) until a configuration compatible with the evidence is found. Then use each sample as the initial guess for any of the optimization algorithms (hill climbing or simulated annealing), which can be run in parallel to obtain M estimates of the MAP: $\mathbf{x}_j^* = \mathrm{optimizationAlgorithm}(\mathbf{x}_j), j = 1, 2, \ldots, M$ which are candidates to optimal solution. Finally, pick the best found solution, $\mathbf{x}^* = \arg\max_{1 \leq j \leq M} p(\mathbf{x}_j^*, \mathbf{x}_E)$, as the MAP estimate.

Figure 1 shows the flow of computation of a Map/Reduce scheme for performing scalable MAP Inference. Notice how several instantiations of the selected search algorithm are run for the input evidence, based on different starting points. Multiple instantiations of the local variations can also be executed. The output is the MAP estimate $\mathbf{x}_I^*$ given each observation $\mathbf{x}_E$, which is computed as the maximum of the local solutions obtained in each computing unit. Both Algorithm 1 and Algorithm 2 are anytime algorithms, thus minimizing synchronization costs.

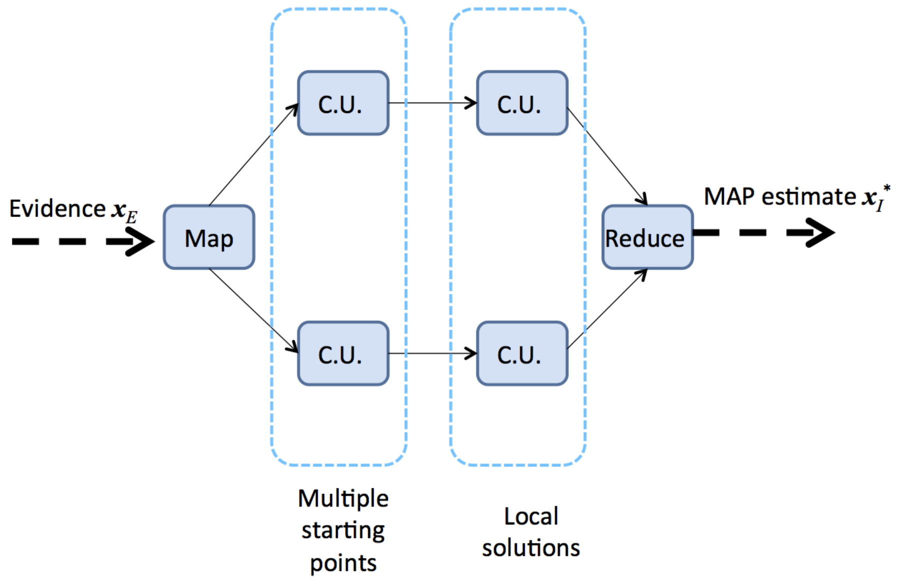

Figure 1: Illustration of the Map/Reduce scheme for MAP Inference. For each observation received as input, the optimization is carried out from different starting points that are distributed among the available computing units (C.U. in the figure). The final estimate is calculated as the maximum of the local solutions.

4. Experimental evaluation

A number of experiments were carried out in order to assess the scalability of the proposed methods. They have been tested when working on top of Apache Flink in two different environments: using a single multi-core node and using a multi-node cluster. For the former scenario, the experiments were run on a dual-processor AMD Opteron 2.8 GHz server with 32 cores and 64 GB of RAM, running Linux Ubuntu 14.04.1 LTS. For the latter, we performed experiments with ad-hoc clusters built within Amazon Web Services (AWS), making use of the Amazon Elastic Cloud Computing

(EC2) and the Elastic MapReduce (EMR) technologies. In each case, there was a master node managing a number of worker nodes. All the nodes employed were of type “m4.xlarge”, each with 16 GB of RAM and 4 vCPUs (virtual CPUs) having a Intel Xeon E5-2676 v3 (Haswell), 2.4 GHz, processor. In our experiments, configurations with 1, 2, 4, 8, 16, and 32 cores/worker nodes were tested.

The setting of the experiments was the following. We randomly generated hybrid BNs with 50, 100, and 200 variables, where half of the variables were categorical and half were continuous. The evidence consisted of observations of 70% of the variables (randomly selected), whereas 10% of the variables were chosen as variables of interest. Thus, 20% of the variables should be marginalized out. For both simulated annealing and hill climbing, 250 initial configurations of the variables of interest were sampled from the BNs. The algorithms were then run for 200 iterations, using 200 auxiliary samples for estimating the probabilities of each partial assignment ($n = 200$ in Equation (3)). The experiments were repeated 5 times and the average runtimes were collected.

The execution times are plotted in Figure 2 for the multi-core scenario (left) and for the multi-node scenario (right), along with the speed-up of each method. For both situations, the speed-up was computed as the execution time with 1 core (resp. node) divided by the execution time with the actually used number of cores (resp. nodes), thus giving a measure of the speed gain due to the increase of available resources. Due to space limitations, we only show results for 200 variables BNs, but the results are qualitatively similar for smaller networks. It can be seen that the different methods exhibit good scalability properties, reaching a maximum speed-up of above 20 in the multi-node scenario and of approximately 17 for the multi-core setup.

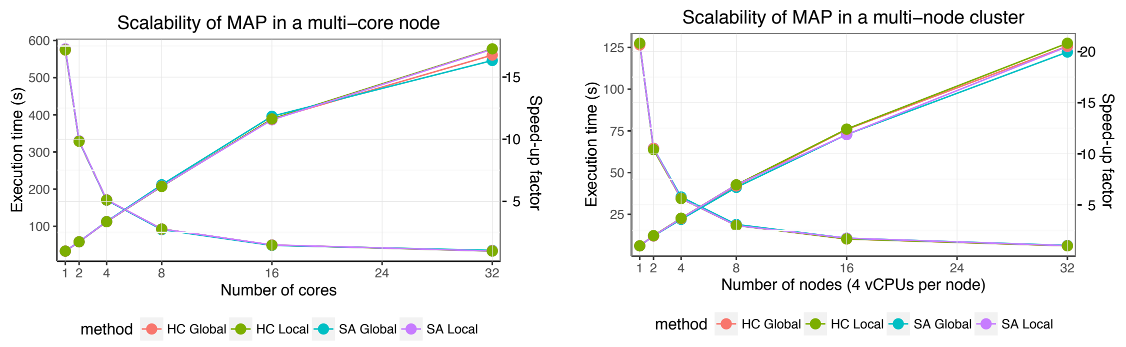

Figure 2: MAP inference in Bayesian networks with 200 variables. Single multi-core node (left) and multi-node cluster (right). In each plot, the mean execution time to obtain the MAP estimate is shown on the left scale and the corresponding speed-up is shown on the right scale. For the local version of the algorithms, the parameter r was set to 3.

Finally, the scalability in terms of precision was also analyzed by fixing the number of starting points per core, so that the total number of starting points is proportional to the number of available computing units. The execution times are therefore approximately the same independently of the number of computing resources. This analysis was performed with a randomly generated Bayesian network consisting of 200 variables, 20% of which were set as variables of interest and another 50%

was set as evidence variables. The different algorithms were run 10 times, with different starting points, and the log-probabilities of the resulting MAP estimates were computed.

The log-probabilities were calculated using Equation (3) with n sufficiently large (starting from 200,000). To ensure an accurate estimation, the log-probabilities were independently estimated three times and the relative standard error was computed, as the standard error divided by the absolute value of the mean. If it was greater than 1%, then the value of n was increased by a factor of 4 and the log-probabilities were reestimated. This process was repeated until the relative standard error was lower than the set threshold, by which time the final probability was computed as the average of the three last estimates.

The distributions of the log-probabilities of the MAP estimates are depicted in Figure 3 for the different methods. The figures show an improvement in the quality of the estimates as more computing resources are used. This effect becomes particularly clear for the median values, which are increasing for all methods and at the same time the variance is steadily reduced.

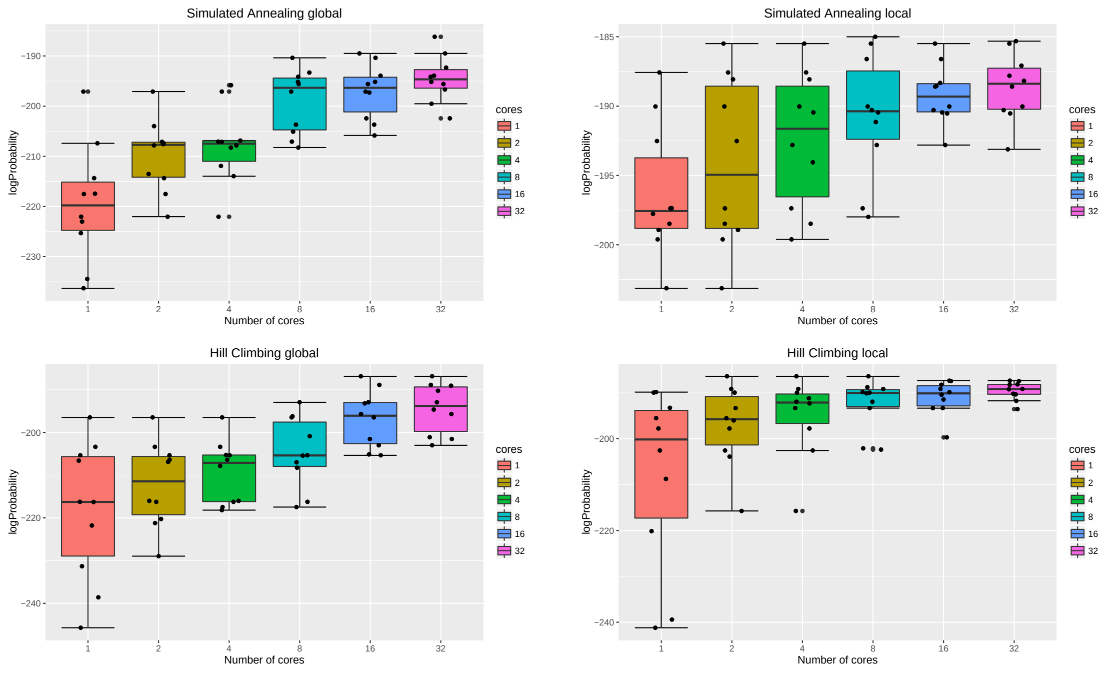

Figure 3: Estimated log-probabilities of the MAP configurations found by each algorithm.

5. Conclusion

In this paper we have studied the problem of performing MAP inference in hybrid BNs from the perspective of scalability. Our analysis focused on CLG networks, which constitute the core of the literature on hybrid BNs. We have described solutions based on simulated annealing and hill climbing, showing how both schemes can be appropriately implemented using MapReduce-based designs. Both solutions have been implemented on top of Apache Flink and tested on a multi-core server as well as on the Amazon Cluster through AWS. The experimental results demonstrate the scalability of our proposals, both in terms of speed-up and accuracy improvements.

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.

References

A. Alexandrov, R. Bergmann, S. Ewen, J. Freytag, F. Hueske, A. Heise, O. Kao, M. Leich, U. Leser, V. Markl, F. Naumann, M. Peters, A. Rheinländer, M. Sax, S. Schelter, M. Höger, K. Tzoumas, and D. Warneke. The Stratosphere platform for big data analytics. *The International Journal on Very Large Data Bases*, 23(6):939–964, 2014.

A. Choi and A. Darwiche. Relax then compensate: On max-product belief propagation and more. In *Proceedings of the Twenty-Third Annual Conference on Neural Information Processing Systems (NIPS)*, pages 351–359, 2009.

E. N. Cinicioglu and P. P. Shenoy. Arc reversals in hybrid Bayesian networks with deterministic variables. *International Journal of Approximate Reasoning*, 50:763–777, 2009.

C. de Campos. New complexity results for MAP in Bayesian networks. In T. Walsh, editor, *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pages 2100–2106, 2011.

L. M. de Campos, J. A. Gámez, and S. Moral. Partial abductive inference in Bayesian belief networks by simulated annealing. *International Journal of Approximate Reasoning*, 27:263–283, 2001.

J. Dean and S. Ghemawat. MapReduce: Simplified data processing on large clusters. *Communications of the ACM*, 51(1):107, 2008.

Q. Fu, H. Wang, and A. Banerjee. Bethe-ADMM for tree decomposition based parallel MAP inference. In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence (UAI2013)*, 2013.

J. A. Gámez. Abductive inference in Bayesian networks: a review. In J. Gámez, S. Moral, and A. Salmerón, editors, *Advances in Bayesian Networks*, pages 101–117. Springer, 2004.

F. Jensen and T. Nielsen. *Bayesian networks and decision graphs*. Springer Publishing Company, Incorporated, second edition, 2007.

U. Kjærulff and A. L. Madsen. *Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis*. Springer, 2013.

J. Kwisthout. Most probable explanations in Bayesian networks: Complexity and tractability. *International Journal of Approximate Reasoning*, 52:1452–1469, 2011.

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.

A. L. Madsen. Belief update in CLG Bayesian networks with lazy propagation. *International Journal of Approximate Reasoning*, 49:503–521, 2008.

R. Marinescu, R. Dechter, and A. Ihler. AND / OR search for marginal MAP. In *Uncertainty in Artificial Intelligence (UAI)*, pages 563–572, 2014.

R. Marinescu, R. Dechter, and A. Ihler. Pushing forward marginal MAP with best-first search. In *IJCAI International Joint Conference on Artificial Intelligence*, pages 696–702, 2015.

A. Masegosa, A. Martínez, H. Borchani, D. Ramos-López, T. Nielsen, H. Langseth, A. Salmerón, and A. Madsen. AMIDST: Analysis of massive data streams. In *Proceedings of the 27th Benelux Conference on Artificial Intelligence (BNAIC 2015)*, 2015.

A. Masegosa, A. Martínez, and H. Borchani. Probabilistic graphical models on multi-core CPUs using Java 8. *IEEE Computational Intelligence Magazine*, 11:41–54, 2016.

J. Park. MAP complexity results and approximation methods. In A. Darwiche and N. Friedman, editors, *UAI*, pages 388–396. Morgan Kaufmann, 2002.

J. Park and A. Darwiche. Approximating MAP using local search. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence (UAI–01)*, pages 403–410. Morgan Kaufmann Publishers, 2001.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Mateo, CA., 1988.

D. Roth. On the hardness of approximate reasoning. *Artificial Intelligence*, 82:273 – 302, 1996.

A. Salmerón, D. Ramos-López, H. Borchani, A. Masegosa, A. Fernández, H. Langseth, A. Madsen, and T. Nielsen. Parallel importance sampling in conditional linear Gaussian networks. *CAEPIA'2015. Lecture Notes in Artificial Intelligence*, 9422:36–46, 2015.

A. Salmerón, R. Rumí, H. Langseth, A. L. Madsen, and T. D. Nielsen. MPE inference in conditional linear Gaussian networks. *ECSQARU'2015. Lecture Notes in Artificial Intelligence*, 9161:407–416, 2015.

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.

Y. Weiss and W. Freeman. On the optimality of solutions of the Max-Product belief-propagation algorithm in arbitrary graphs. *IEEE Transactions on Information Theory*, 47(2):736–744, 2010.

C. Yuan and E. Hansen. Efficient computation of jointree bounds for systematic MAP search. In *IJCAI International Joint Conference on Artificial Intelligence*, pages 1982–1989, 2009.