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

Darío Ramos-López[1], Antonio Salmerón[1], Rafael Rumí[1], Ana M. Martínez[2], Thomas D. Nielsen[2], Andrés R. Masegosa[3], Helge Langseth[3], Anders L. Madsen[2,4]

[1]Department of Mathematics, University of Almería, Spain

[2]Department of Computer Science, Aalborg University, Denmark

[3] Department of Computer and Information Science, The Norwegian University of Science and Technology, Norway

[4] Hugin Expert A/S, Aalborg, Denmark

Outline

1. Motivation
2. MAP in CLG networks
3. Scalable MAP
4. Experimental results
5. Conclusions

Outline

1. Motivation
2. MAP in CLG networks
3. Scalable MAP
4. Experimental results
5. Conclusions

Motivation

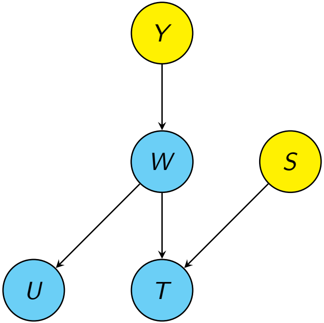
- Aim: Provide scalable solutions to the MAP problem.
- Challenges:
  - Data coming in streams at high speed, and a quick response is required.
  - For each observation in the stream, the most likely configuration of a set of variables of interest is sought.
  - MAP inference is highly complex.
  - Hybrid models come along with specific difficulties.

Context

- The AMiDST project: Analysis of MassIve Data STreams
  http://www.amidst.eu
- Large number of variables
- Queries to be answered in real time
- Hybrid Bayesian networks (involving discrete and continuous variables)
  - Conditional linear Gaussian networks

Outline

Conditional Linear Gaussian networks



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

Querying a Bayesian network

- Belief update: Computing the posterior distribution of a variable:

$$p(x_i|\boldsymbol{x}_E) = \frac{\displaystyle\sum_{\boldsymbol{x}_D} \int_{\boldsymbol{x}_C} p(\boldsymbol{x}, \boldsymbol{x}_E) \mathrm{d}\boldsymbol{x}_C}{\displaystyle\sum_{\boldsymbol{x}_{D_i}} \int_{\boldsymbol{x}_{C_i}} p(\boldsymbol{x}, \boldsymbol{x}_E) \mathrm{d}\boldsymbol{x}_{C_i}}$$

- Maximum a posteriori (MAP): For a set of target variables $\boldsymbol{X}_I$, seek

$$\boldsymbol{x}_I^* = \arg\max_{\boldsymbol{x}_I} p(\boldsymbol{x}_I|\boldsymbol{X}_E = \boldsymbol{x}_E)$$

  where $p(\boldsymbol{x}_I|\boldsymbol{X}_E = \boldsymbol{x}_E)$ is obtained by first marginalizing out from $p(\boldsymbol{x})$ the variables not in $\boldsymbol{X}_I$ and not in $\boldsymbol{X}_E$

- Most probable explanation (MPE): A particular case of MAP where $\boldsymbol{X}_I$ includes all the unobserved variables

Querying a Bayesian network

- Belief update: Computing the posterior distribution of a variable:

$$p(x_i|\boldsymbol{x}_E) = \frac{\displaystyle\sum_{\boldsymbol{x}_D} \int_{\boldsymbol{x}_C} p(\boldsymbol{x}, \boldsymbol{x}_E)\mathrm{d}\boldsymbol{x}_C}{\displaystyle\sum_{\boldsymbol{x}_{D_i}} \int_{\boldsymbol{x}_{C_i}} p(\boldsymbol{x}, \boldsymbol{x}_E)\mathrm{d}\boldsymbol{x}_{C_i}}$$

- Maximum a posteriori (MAP): For a set of target variables $\boldsymbol{X}_I$, seek

$$\boldsymbol{x}_I^* = \arg\max_{\boldsymbol{x}_I} p(\boldsymbol{x}_I|\boldsymbol{X}_E = \boldsymbol{x}_E)$$

where $p(\boldsymbol{x}_I|\boldsymbol{X}_E = \boldsymbol{x}_E)$ is obtained by first marginalizing out from $p(\boldsymbol{x})$ the variables not in $\boldsymbol{X}_I$ and not in $\boldsymbol{X}_E$

- Most probable explanation (MPE): A particular case of MAP where $\boldsymbol{X}_I$ includes all the unobserved variables

Outline

MAP by Hill Climbing

HC_MAP(B,$\boldsymbol{x}_I$,$\boldsymbol{x}_E$,r)
while *stopping criterion not satisfied* **do**
  $\boldsymbol{x}_I^* \leftarrow$ **GenerateConfiguration**(B, $\boldsymbol{x}_I$, $\boldsymbol{x}_E$, r)
  if $p(\boldsymbol{x}_I^*, \boldsymbol{x}_E) \geq p(\boldsymbol{x}_I, \boldsymbol{x}_E)$ **then**
    $\boldsymbol{x}_I \leftarrow \boldsymbol{x}_I^*$
  end
end
return $\boldsymbol{x}_I$

MAP by Hill Climbing

HC_MAP$(B, \boldsymbol{x}_I, \boldsymbol{x}_E, r)$
while *stopping criterion not satisfied* **do**
  $\boldsymbol{x}_I^* \leftarrow$ **GenerateConfiguration**$(B, \boldsymbol{x}_I, \boldsymbol{x}_E, r)$
  if $p(\boldsymbol{x}_I^*, \boldsymbol{x}_E) \geq p(\boldsymbol{x}_I, \boldsymbol{x}_E)$ **then**
    $\boldsymbol{x}_I \leftarrow \boldsymbol{x}_I^*$
  end
end
return $\boldsymbol{x}_I$

- Max. number of non-improving iterations
- Target prob. threshold
- Max. number of iterations

MAP by Hill Climbing

HC_MAP$(B, \boldsymbol{x}_I, \boldsymbol{x}_E, r)$
while *stopping criterion not satisfied* **do**
  $\boldsymbol{x}_I^* \leftarrow$ **GenerateConfiguration**$(B,\ \boldsymbol{x}_I,\ \boldsymbol{x}_E,\ r)$
  if $p(\boldsymbol{x}_I^*, \boldsymbol{x}_E) \geq p(\boldsymbol{x}_I, \boldsymbol{x}_E)$ **then**
    $\boldsymbol{x}_I \leftarrow \boldsymbol{x}_I^*$
  end
end
return $\boldsymbol{x}_I$

- Never move to a worse configuration

MAP by Hill Climbing

HC_MAP$(B, \mathbf{x}_I, \mathbf{x}_E, r)$
while *stopping criterion not satisfied* **do**
  $\mathbf{x}_I^* \leftarrow$ **GenerateConfiguration**$(B,\ \mathbf{x}_I,\ \mathbf{x}_E,\ r)$
  if $\boxed{p(\mathbf{x}_I^*, \mathbf{x}_E)} \geq \boxed{p(\mathbf{x}_I, \mathbf{x}_E)}$ **then**
    $\mathbf{x}_I \leftarrow \mathbf{x}_I^*$
  end
end
return

- Estimated using importance sampling:

$$p(\mathbf{x}_I, \mathbf{x}_E) = \sum_{\mathbf{x}^* \in \Omega_{\mathbf{x}^*}} p(\mathbf{x}_I, \mathbf{x}_E, \mathbf{x}^*) = \sum_{\mathbf{x}^* \in \Omega_{\mathbf{x}^*}} \frac{p(\mathbf{x}_I, \mathbf{x}_E, \mathbf{x}^*)}{f^*(\mathbf{x}^*)} f^*(\mathbf{x}^*)$$

$$= \mathbb{E}_{f^*}\left[\frac{p(\mathbf{x}_I, \mathbf{x}_E, \mathbf{x}^*)}{f^*(\mathbf{x}^*)}\right] \approx \frac{1}{n}\sum_{i=1}^{n} \frac{p\left(\mathbf{x}_I, \mathbf{x}_E, \mathbf{x}^{*(i)}\right)}{f^*\left(\mathbf{x}^{*(i)}\right)},$$

MAP by Hill Climbing

HC_MAP$(B, \boldsymbol{x}_I, \boldsymbol{x}_E, r)$
while *stopping criterion not satisfied* **do**
  $\boldsymbol{x}_I^* \leftarrow$ **GenerateConfiguration**$(B,\ \boldsymbol{x}_I,\ \boldsymbol{x}_E,\ r)$
  if $p(\boldsymbol{x}_I^*, \boldsymbol{x}_E) \geq p(\boldsymbol{x}_I, \boldsymbol{x}_E)$ **then**
    $\boldsymbol{x}_I \leftarrow \boldsymbol{x}_I^*$
  end
end
return $\boldsymbol{x}_I$

- We'll see later

MAP by Simulated Annealing

SA_MAP$(B, \boldsymbol{x}_I, \boldsymbol{x}_E, r)$
$T \leftarrow 1\,000$; $\alpha \leftarrow 0.90$; $\varepsilon > 0$
while $T \geq \varepsilon$ **do**
    $\boldsymbol{x}_I^* \leftarrow$ **GenerateConfiguration**$(B, \boldsymbol{x}_I, \boldsymbol{x}_E, r)$
    Simulate a random number $\tau \sim \mathcal{U}(0,1)$
    if $p(\boldsymbol{x}_I^*, \boldsymbol{x}_E) > p(\boldsymbol{x}_I, \boldsymbol{x}_E)/(T \cdot \ln(1/\tau))$ **then**
        $\boldsymbol{x}_I \leftarrow \boldsymbol{x}_I^*$
    end
    $T \leftarrow \alpha \cdot T$
end
return $\boldsymbol{x}_I$

MAP by Simulated Annealing

SA_MAP$(B, \mathbf{x}_I, \mathbf{x}_E, r)$
$T \leftarrow 1\,000$; $\alpha \leftarrow 0.90$; $\varepsilon > 0$
while $T \geq \varepsilon$ **do**
  $\mathbf{x}_I^* \leftarrow$ **GenerateConfiguration**$(B, \mathbf{x}_I, \mathbf{x}_E, r)$
  Simulate a random number $\tau \sim \mathcal{U}(0,1)$
  if $p(\mathbf{x}_I^*, \mathbf{x}_E) > p(\mathbf{x}_I, \mathbf{x}_E)/(T \cdot \ln(1/\tau))$ **then**
    $\mathbf{x}_I \leftarrow \mathbf{x}_I^*$
  end
  $T \leftarrow \alpha \cdot T$
end
return $\mathbf{x}_I$

- Default values of the temperature parameters
  - $T \gg 1$: almost completely random
  - $T \ll 1$: almost completely greedy

MAP by Simulated Annealing

SA_MAP$(B, \boldsymbol{x}_I, \boldsymbol{x}_E, r)$
$T \leftarrow 1\,000$; $\alpha \leftarrow 0.90$; $\varepsilon > 0$
while $T \geq \varepsilon$ **do**
  $\boldsymbol{x}_I^* \leftarrow$ **GenerateConfiguration**$(B, \boldsymbol{x}_I, \boldsymbol{x}_E, r)$
  Simulate a random number $\tau \sim \mathcal{U}(0,1)$
  if $p(\boldsymbol{x}_I^*, \boldsymbol{x}_E) > p(\boldsymbol{x}_I, \boldsymbol{x}_E)/(T \cdot \ln(1/\tau))$ **then**
    $\boldsymbol{x}_I \leftarrow \boldsymbol{x}_I^*$
  end
  $T \leftarrow \alpha \cdot T$
end
return $\boldsymbol{x}_I$

- Accept $\boldsymbol{x}_I^*$ if its prob. increases or decreases $< T \cdot \ln(1/\tau)$

MAP by Simulated Annealing

SA_MAP$(B,\mathbf{x}_I,\mathbf{x}_E,r)$
$T \leftarrow 1\,000$; $\alpha \leftarrow 0.90$; $\varepsilon > 0$
while $T \geq \varepsilon$ **do**
  $\mathbf{x}_I^* \leftarrow$ **GenerateConfiguration**$(B, \mathbf{x}_I, \mathbf{x}_E, r)$
  Simulate a random number $\tau \sim \mathcal{U}(0,1)$
  if $p(\mathbf{x}_I^*, \mathbf{x}_E) > p(\mathbf{x}_I, \mathbf{x}_E)/(T \cdot \ln(1/\tau))$ **then**
    $\mathbf{x}_I \leftarrow \mathbf{x}_I^*$
  end
  $T \leftarrow \alpha \cdot T$
end
return $\mathbf{x}_I$

- Cool down the temperature

Generating a new configuration of variables

Only discrete variables

- The new values are chosen at random

Generating a new configuration of variables

Hybrid models

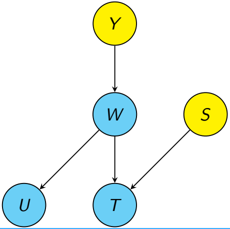
- We take advantage of the properties of the CLG distribution

Generating a new configuration of variables

Hybrid models

- We take advantage of the properties of the CLG distribution

A variable whose parents are discrete or observed



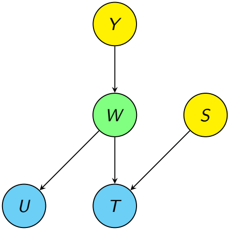
$$P(Y) = (0.5, 0.5)$$
$$P(S) = (0.1, 0.9)$$
$$f(w|Y=0) = \mathcal{N}(w; -1, 1)$$
$$f(w|Y=1) = \mathcal{N}(w; 2, 1)$$
$$f(t|w, S=0) = \mathcal{N}(t; -w, 1)$$
$$f(t|w, S=1) = \mathcal{N}(t; w, 1)$$
$$f(u|w) = \mathcal{N}(u; w, 1)$$

Generating a new configuration of variables

Hybrid models

- We take advantage of the properties of the CLG distribution

A variable whose parents are discrete or observed



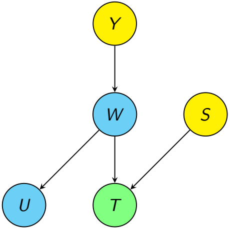
Return its conditional mean:

$$f(w|Y=0)=\mathcal{N}(w;-1,1)$$
$$f(w|Y=1)=\mathcal{N}(w;2,1)$$

Hybrid models

- We take advantage of the properties of the CLG distribution

A variable with unobserved continuous parents

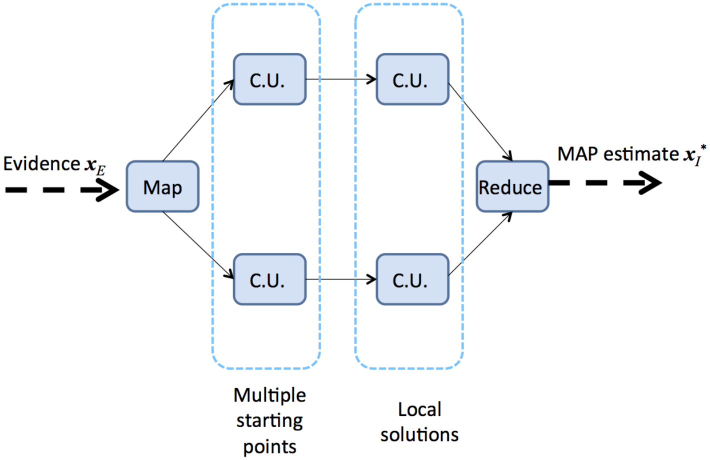


Simulate a value using

its conditional distribution:

$$f(t|w, S = 0) = \mathcal{N}(t; -w, 1)$$
$$f(t|w, S = 1) = \mathcal{N}(t; w, 1)$$

Scalable implementation



Outline

Experimental analysis

Purpose

Analyze the scalability in terms of

- Speed
- Accuracy

Experimental analysis

Purpose

Analyze the scalability in terms of

- Speed
- Accuracy

Experimental setup

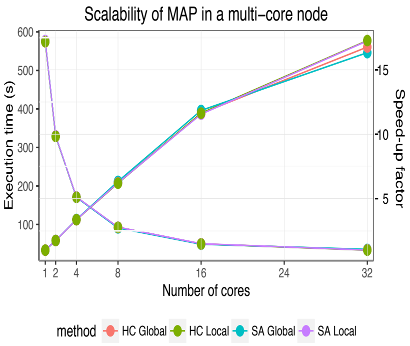
- Synthetic networks with 200 variables (50% discrete)
- 70% of the variables observed at random
- 10% of the variables selected as target $\Rightarrow$ 20% to be marginalized out

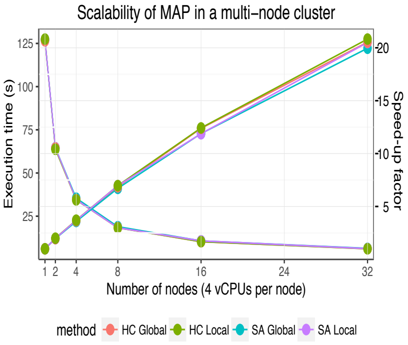
Experimental analysis

Computing environment

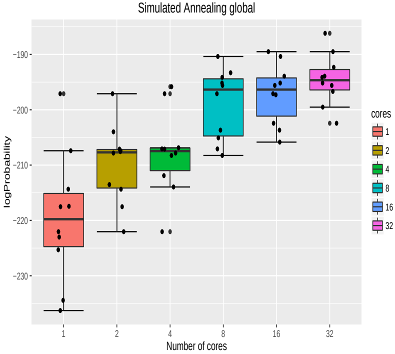
- AMIDST Toolbox with Apache Flink
- Multi-core environment based on a dual-processor AMD Opteron 2.8 GHz server with 32 cores and 64 GB of RAM, running Ubuntu Linux 14.04.1 LTS
- Multi-node environment based on Amazon Web Services (AWS)

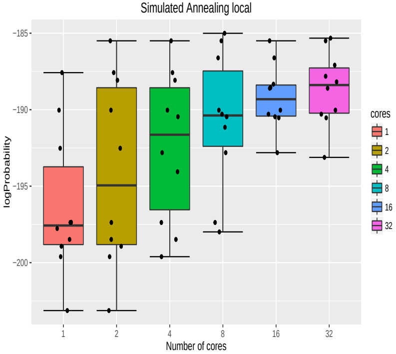
Scalability: run times





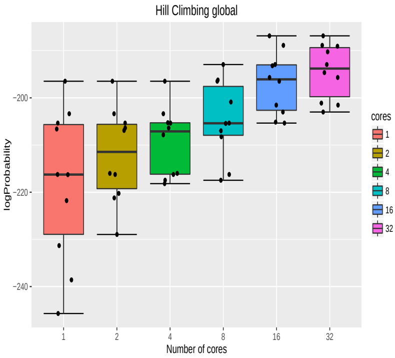
Scalability: accuracy (Simulated Annealing)

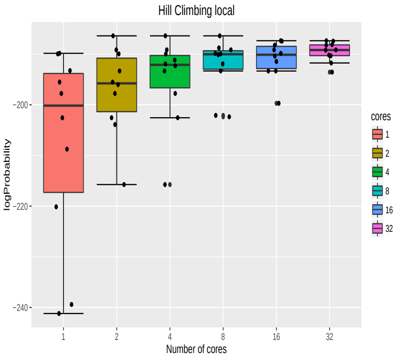




Estimated log-probabilities of the MAP configurations found by each algorithm

Scalability: accuracy (Hill Climbing)





Estimated log-probabilities of the MAP configurations found by each algorithm

Outline

1. Motivation
2. MAP in CLG networks
3. Scalable MAP
4. Experimental results
5. Conclusions

Conclusions

- Scalable MAP for CLG models in terms of accuracy and run time
- Available in the AMIDST Toolbox
- Valid for multi-cores and cluster systems
- MapReduce-based design on top of Apache Flink

Thank you for your attention

You can download our open source Java toolbox:

AMiDST TOOLBOX

`http://www.amidsttoolbox.com`

Acknowledgments: This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 619209