Generating Random WMC Instances

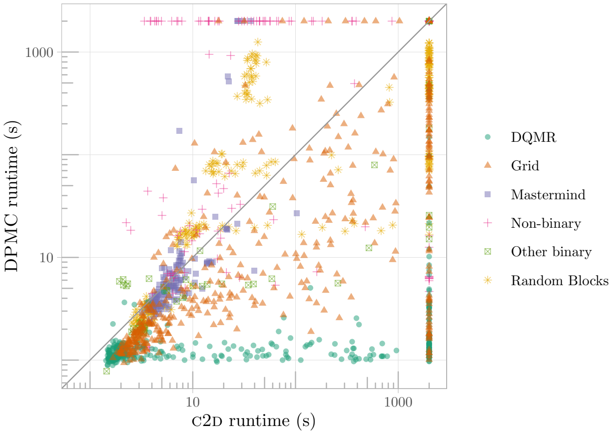
An Empirical Analysis with Varying Primal Treewidth

Paulius Dilkas

National University of Singapore

AIAI Seminar

Which Algorithm Is Better? It Depends on the Data



Data from Dilkas and Belle (2021): various Bayesian networks encoded using the method by Darwiche (2002)

The Problem: Weighted Model Counting (WMC)

- A generalisation of propositional model counting (#SAT)
- Applications:
  - graphical models
  - probabilistic programming
  - neural-symbolic artificial intelligence

Example

$w(x) = 0.3,\ w(\neg x) = 0.7,$
$w(y) = 0.2,\ w(\neg y) = 0.8$

$\mathrm{WMC}(x \lor y) = w(x)w(y) + w(x)w(\neg y) + w(\neg x)w(y) = 0.44$

(Some of the) WMC Algorithms

- Cachet (Sang et al. 2004)
  - a SAT solver with clause learning and component caching
- c2d (Darwiche 2004)
  - knowledge compilation to d-DNNF
- d4 (Lagniez and Marquis 2017)
  - knowledge compilation to decision-DNNF
- miniC2D (Oztok and Darwiche 2015)
  - knowledge compilation to decision sentential decision diagrams
- DPMC (Dudek, Phan and Vardi 2020)
  - dynamic programming with algebraic decision diagrams and tree decomposition based planning

Tree Decompositions and Primal Treewidth

Formula in CNF:

$$\phi = (x_4 \vee \neg x_3 \vee x_1) \wedge (\neg x_2 \vee x_4) \wedge (\neg x_1 \vee x_2 \vee x_4)$$

Tree Decompositions and Primal Treewidth

Formula in CNF:

$$\phi = (x_4 \vee \neg x_3 \vee x_1) \wedge (\neg x_2 \vee x_4) \wedge (\neg x_1 \vee x_2 \vee x_4)$$

Its primal graph:

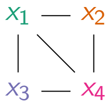
$$\begin{tikzcd} x_1 \arrow[r, dash] \arrow[d, dash] \arrow[dr, dash] & x_2 \arrow[d, dash] \\ x_3 \arrow[r, dash] & x_4 \end{tikzcd}$$

Tree Decompositions and Primal Treewidth

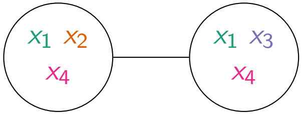
Formula in CNF:

$$\phi = (x_4 \vee \neg x_3 \vee x_1) \wedge (\neg x_2 \vee x_4) \wedge (\neg x_1 \vee x_2 \vee x_4)$$

Its primal graph:



Its minimum-width tree decomposition:

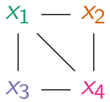


Tree Decompositions and Primal Treewidth

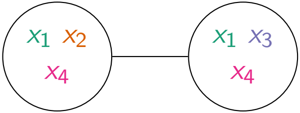
Formula in CNF:

$$\phi = (x_4 \vee \neg x_3 \vee x_1) \wedge (\neg x_2 \vee x_4) \wedge (\neg x_1 \vee x_2 \vee x_4)$$

Its primal graph:



Its minimum-width tree decomposition:



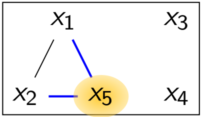
$\therefore$ the primal treewidth of ϕ is 2

From Random SAT to Random WMC

We introduce parameter $\rho \in [0, 1]$ that biases the probability distribution towards adding variables that would introduce fewer new edges to the primal graph.

Example partially-filled formula:
$(\neg x_5 \vee x_2 \vee x_1) \wedge (x_5 \vee ? \vee ?)$

Its primal graph:



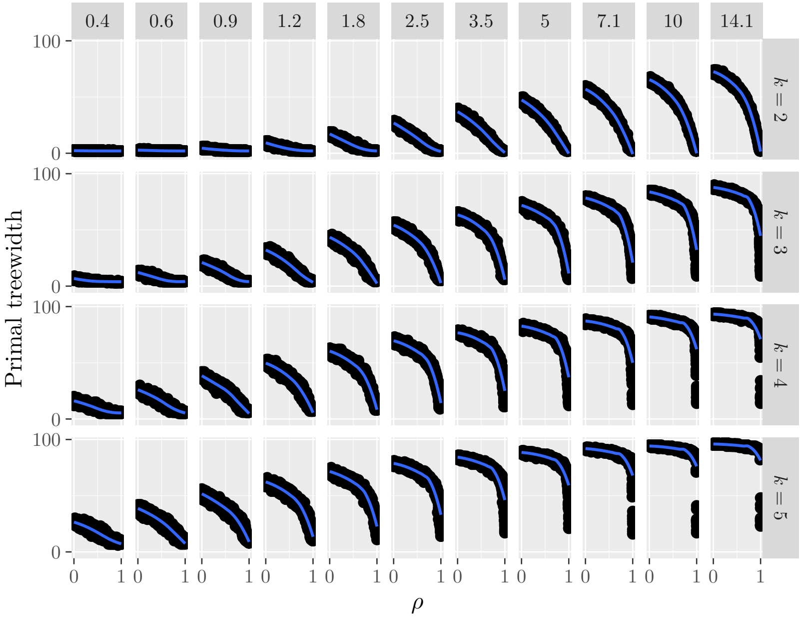
The probability distribution for the next variable

Base probability of each variable being chosen:

$$\frac{1-\rho}{4}.$$

Both x_1 and x_2 get a bonus probability of $\rho/2$ for each being the endpoint of one out of the two neighbourhood edges.

The Relationship Between ρ and Primal Treewidth



Peak Hardness w.r.t. Density

Let μ denote the density, i.e., the number of clauses divided by the number of variables.

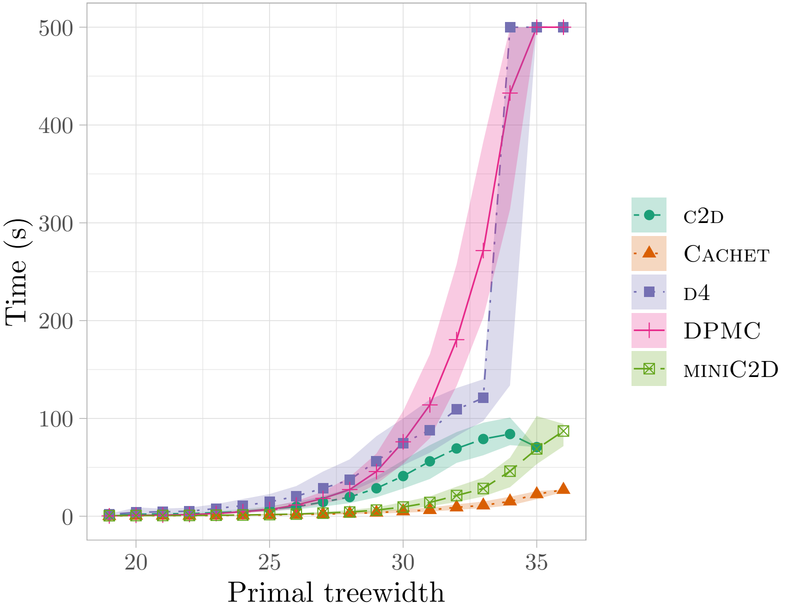
- CACHET is known to peak at $\mu = 1.8$ (Sang et al. 2004)
- Bayardo Jr. and Pehoushek (2000) show some #SAT algorithms to peak at $\mu = 1.2$ and $\mu = 1.9$

Peak Hardness w.r.t. Density

Let μ denote the density, i.e., the number of clauses divided by the number of variables.

- CACHET is known to peak at $\mu = 1.8$ (Sang et al. 2004)
- Bayardo Jr. and Pehoushek (2000) show some #SAT algorithms to peak at $\mu = 1.2$ and $\mu = 1.9$
- In our experiments:
  - DPMC peaks at $\mu = 2.2$
  - all other algorithms peak at $\mu = 1.9$

Hardness w.r.t. Primal Treewidth (when $\mu = 1.9$)

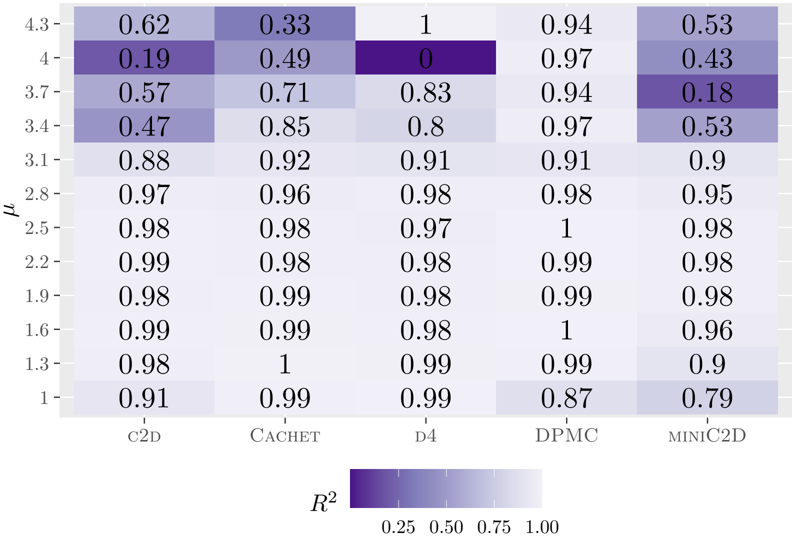


Is The Relationship Exponential?

Let us fit the model $\ln t \sim \alpha w + \beta$, i.e., $t \sim e^{\beta}(e^{\alpha})^{w}$, where t is runtime, and w is primal treewidth

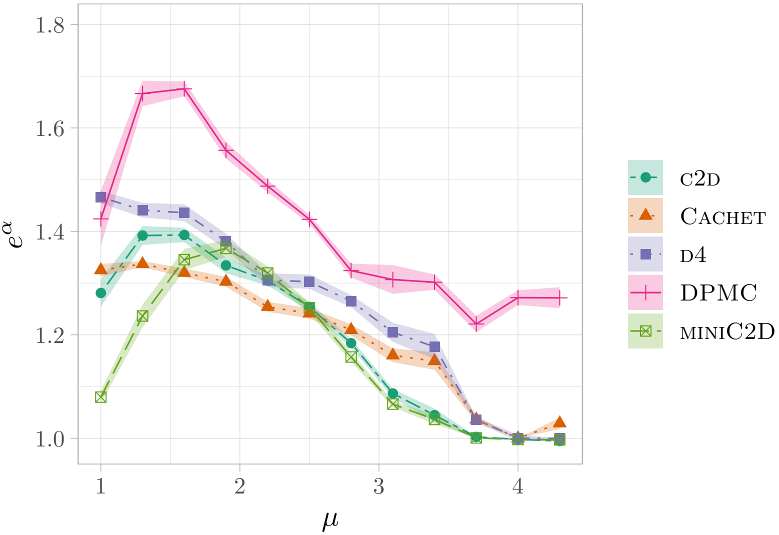
Is The Relationship Exponential?

Let us fit the model $\ln t \sim \alpha w + \beta$, i.e., $t \sim e^{\beta}(e^{\alpha})^{w}$, where t is runtime, and w is primal treewidth



Is The Relationship Exponential?

Let us fit the model $\ln t \sim \alpha w + \beta$, i.e., $t \sim e^{\beta}(e^{\alpha})^{w}$, where t is runtime, and w is primal treewidth



Summary

- Introduced a random model for WMC instances
- Observations:
  - All algorithms scale exponentially w.r.t. primal treewidth
  - The running time of DPMC peaks at a higher density and scales worse w.r.t. primal treewidth
- Future work:
  - A theoretical relationship between ρ and primal treewidth
  - Non-k-CNF instances
  - Algorithm portfolios for WMC