Modular Exact Inference for Discrete Probabilistic Programs

Steven Holtzen and Guy Van den Broeck and Todd Millstein
University of California, Los Angeles
{sholtzen,guyvdb,todd}@cs.ucla.edu

UCLA Samueli Computer Science

1. Motivation

- PPLs are extremely powerful and flexible, but this makes inference hard
  - PPLs have a *scaling problem:* focus primarily on small programs
- ***Our goal:*** Focus on **discrete programs,** make a high-performance *exact* inference algorithm for this specialized setting
  - Discreteness is *very common*, many programs have discrete parts (text, graphs, computer networks, …)
  - Discreteness is *challenging for many methods*
    - Many methods rely on differentiability
    - Low-probability observations
  - Exact inference preferable to approximate
    - Does not propagate errors
    - Suitable for high-consequence decisionmaking

2. Method

- Key idea: *factorize the inference computation* (see Figure 1a)

$$\underbrace{0.1}_{x=T}\cdot\underbrace{0.2}_{y=T}\cdot\underbrace{0.4}_{z=T}+\underbrace{0.1}_{x=T}\cdot\underbrace{0.8}_{y=F}\cdot\underbrace{0.5}_{z=T}+\underbrace{0.9}_{x=F}\cdot\underbrace{0.3}_{y=T}\cdot\underbrace{0.4}_{z=T}+\underbrace{0.9}_{x=F}\cdot\underbrace{0.7}_{y=F}\cdot\underbrace{0.5}_{z=T}$$

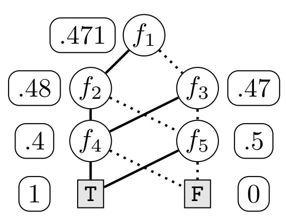
Versus…

$$\underbrace{0.1}_{x=T}\cdot\Big(\underbrace{0.2}_{y=T}\cdot\underbrace{0.4}_{z=T}+\underbrace{0.8}_{y=F}\cdot\underbrace{0.5}_{z=T}\Big)+\underbrace{0.9}_{x=F}\cdot\Big(\underbrace{0.3}_{y=T}\cdot\underbrace{0.4}_{z=T}+\underbrace{0.7}_{y=F}\cdot\underbrace{0.5}_{z=T}\Big).$$

- Finding and exploiting these factorization opportunities can be hard!
- We do it with binary decision diagrams (BDDs)

```
let x = flip₁ 0.1 in
let y = if x then flip₂ 0.2 else
    flip₃ 0.3 in
let z = if y then flip₄ 0.4 else
    flip₅ 0.5 in z
```

(a) Example Dice program.



(b) Compiled BDD with weighted model counts.

Figure 1: Illustration of compiling a Dice program that exploits factorization.

3. Experiments

- Show Dice can perform exact inference on *extremely large programs*
  - For instance, a 1.9 megabyte program with over 100k random variables
- Compared Dice against **Psi** and **Ace** (specialized Bayesian network solver)
- Three main experiments:
  1. Common Baselines
  2. Single-marginal Bayesian network inference
  3. All-marginal Bayesian network inference

Benchmark	Psi (ms)	DP (ms)	Dice (ms)	# Paths	BDD Size
Grass	167	57	**14**	95	15
Burglar Alarm	98	10	**13**	250	11
Coin Bias	94	23	**13**	4	13
Noisy Or	81	152	**13**	1640	35
Evidence1	48	32	**13**	9	5
Evidence2	59	28	**13**	9	6
Murder Mystery	193	75	**10**	16	6

Table 1: *Baselines.* Comparison of inference algorithms (times are milliseconds). The total time for Dice is reported under the "Dice" column, and the total size of the final compiled BDD is reported in the "BDD Size" column.

Benchmark	Psi (ms)	DP (ms)	Dice (ms)	# Parameters	# Paths
Cancer	772	46	**13.0**	10	1.1×10^{3}
Survey	2477	152	**13.0**	21	1.3×10^{4}
Alarm	✗	✗	**25.0**	509	1.0×10^{36}
Insurance	✗	✗	**75.0**	984	1.2×10^{40}
Hepar2	✗	✗	**54.0**	48	2.9×10^{69}
Hailfinder	✗	✗	**618**	2656	2.0×10^{76}
Pigs	✗	✗	**72**	5618	7.3×10^{492}
Water	✗	✗	**2876**	1.0×10^{4}	3.2×10^{54}
Munin	✗	✗	**1998**	8.1×10^{5}	2.1×10^{1622}

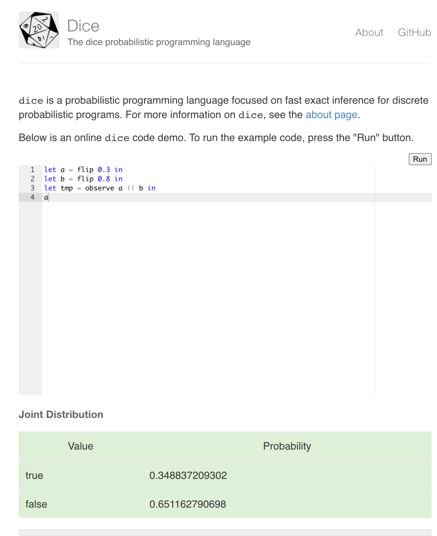
Table 1: *Single Marginal Inference.* Comparison of inference algorithms (times are milliseconds). A "✗" denotes a timeout at 2 hours of running. The total time for Dice is reported under the "Dice" column, and the total size of the final compiled BDD is reported in the "BDD Size" column.

Benchmark	Dice (ms)	Ace (ms)	BDD Size
Alarm	**159**	422	4.3×10^{5}
Hailfinder	1280	**522**	2.1×10^{5}
Insurance	**222**	492	2.3×10^{5}
Hepar2	**163**	495	5.4×10^{5}
Pigs	11243	**985**	2.6×10^{5}
Water	3320	**605**	6.8×10^{4}
Munin	4021194	**3500**	2.2×10^{7}

Table 1: *All marginals.* A comparison between Dice and Ace on the all-marginal discrete Bayesian network inference task.

4. Conclusion

- Github: https://github.com/SHoltzen/dice
- Webpage: http://dicelang.cs.ucla.edu/
- Steven Holtzen, Guy Van den Broeck, and Todd Millstein. 2020. Scaling Exact Inference for Discrete Probabilistic Programs. Proc. ACM Program. Lang. 4, OOPSLA, Article 140 (November 2020), 37 pages. https://doi.org/10.1145/3428208
- Paper link: https://arxiv.org/abs/2005.09089



5. Acknowledgments

This work is partially supported by NSF grants #IIS-1943641, #IIS-1956441, #CCF-1837129, DARPA grant #N66001-17-2-4032, a Sloan Fellowship, and gifts by Intel and Facebook research. The authors would like to thank Jon Aytac and Philip Johnson-Freyd for feedback on paper drafts.

NSF DARPA Alfred P. Sloan FOUNDATION intel